Dell EMC Avamar for SQL Server

User Guide

19.1

Dell Inc.

July 2021
Rev. 04

Notes, cautions, and warnings

NOTE: A NOTE indicates important information that helps you make better use of your product.

CAUTION: A CAUTION indicates either potential damage to hardware or loss of data and tells you how to avoid the problem.

WARNING: A WARNING indicates a potential for property damage, personal injury, or death.

© 2001 - 2021 Dell Inc. or its subsidiaries. All rights reserved. Dell, EMC, and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be trademarks of their respective owners.

Contents

Tables 7

Figures 8

Preface 9

Chapter 1: Introduction 13
Architecture 13
Avamar components 13
How Avamar connects to SQL Server 14
Failover Cluster Instances 14
AlwaysOn availability groups 16
Database mirroring 21
Data Domain system support 21
Server virtualization 21
Backup 22
Backup types 22
Backups in AlwaysOn availability groups 24
Mixed backup storage 24
Multistreaming 25
Database log truncation 25
Backups with other tools 25
Restore 25
Restore to the original location 26
Restore to a new database in the original instance 26
Restore to a different instance on the original server 26
Restore to an instance on a different server 26
Restore to a file 26
Restore to an AlwaysOn availability group 27
Restore of a database with an intact log file 28
Restore of system databases 28
Tail-log backup 29
Point-in-time restore 30
SQL Server recovery operations 30
Table level recovery 31
Disaster recovery 31

Chapter 2: Installation 32
Preparing to install the SQL Server plug-in 32
System requirements 32
Preparing a stand-alone server 36
Preparing a cluster 37
Downloading the software 37
Verifying the environment 38
Installing the Avamar Plug-in for SQL Server 39

Installing the Avamar client software....40
Installation road maps....40
Installing the Avamar Client for Windows....40
Installing the Avamar Plug-in for SQL Server....41
Silent installation of Itempoint for SQL....42
Registering the client....42
Configuring the cluster client in a failover cluster....42
Configuring the cluster client for an AlwaysOn availability group....43
Configuring the cluster client for an AlwaysOn availability group on Amazon Web Services and Azure....44
Upgrading the Avamar client software....46
Upgrading on a stand-alone server....46
Upgrading in a cluster....46
Uninstalling the Avamar client software....47
Uninstall road map....47
Uninstalling the cluster client in a failover cluster....47
Uninstalling the cluster client for an AlwaysOn availability group....48

Chapter 3: Backup....49
Perform an on-demand SQL backup by using AUI....49
Configure the SQL server policy....50
Scheduling backups using the AUI Policy wizard....51
Creating a dataset....51
Creating a backup policy....52
Enabling a scheduled backup for a backup policy....52
Monitoring backups....52
Cancel backups....53
Verifying backups....53
Enforcement of backups to Data Domain....53

Chapter 4: Application Consistent SQL Virtual Machine Image Backup....54
Managing advanced policies....54
Prerequisites....54
Add an advanced group policy....54
Configure a source....55
Configure the SQL server policy....55
Configure members....56
Configure a proxy for the image backup....57
Remove an advanced policy....57
Edit an advanced policy....57
View advanced policy details....58
Migrate existing Backup Policy to an Advanced Policy....58
View logs....58

Chapter 5: Restore....59
Restore requirements....59
Software requirements for restore....59
Tail-log backup and point-in-time restore requirements....60
Requirements to restore secondary database files....60

Requirements to restore the Report Server database....61
SQL Server write permission requirements....61
Determine the restore size for an SQL server database....61
Determine the space required to restore the system database....61
Determine the space required to restore a single SQL server database....61
Determine the space required to restore multiple SQL server databases....62
Restoring a database to the original client....62
Restoring a database to a different client....63
Restore features available by using Avamar Administrator....64
Finding a backup....64
Restoring to the original location....69
Restoring to a new database in the original instance....70
Restoring to a different instance....71
Restoring to a file....72
Restoring system databases....78
Restoring to an AlwaysOn availability group....83
Restoring a database with an intact log file....85
Setting restore options....86
Performing table level recovery....92
Restore only on primary replica....93
Table Level restore....93
Monitor restores....94
Cancel restores....95

Chapter 6: Disaster Recovery....96
Preparing for disaster recovery....96
Performing disaster recovery....96

Appendix A: SQL Plug-in Options....97
How to set plug-in options....97
SQL plug-in backup options....97
Backup options....100
SQL plug-in restore options....104
SQL TLR plug-in options....107
SQL TLR plug-in browse options....107
SQL TLR plug-in restore options....108

Appendix B: Command Line Interface....109
Overview of the SQL Server plug-in CLI....109
Command reference....109
Synopsis....109
Operations....109
Options....110
Specifying command line options....122
Password encoding....123
Performing command line operations in a failover cluster....123
Performing command line operations in an AlwaysOn environment....123
CLI examples....124
Example browse commands....124

Example backup commands.....125
Example restore commands.....128
avsqltlr command reference.....133
avsqltlr synopsis.....133
avsqltlr options.....133
Monitoring CLI activities.....133

Appendix C: Troubleshooting.....134
Backup problems and solutions.....134
Restore problems and solutions.....135
Upgrade problems and solutions.....135

Appendix D: Cluster Configuration Tool CLI.....136
Cluster Configuration Tool CLI Overview.....136
Plugins.....136
Operations.....137
SQL plugin configure options.....137
SQL plugin remove options.....138
SQLAlwayson plugin configure options.....138
SQLAlwayson plugin remove options.....139
Cluster Configuration Tool CLI examples.....140

Appendix E: ItemPoint for Microsoft SQL Server Notes.....143
Supported environments for ItemPoint for Microsoft SQL Server.....143
Limitations for ItemPoint for Microsoft SQL Server.....144

Glossary.....146

Tables

1 Typographical conventions.... 10
2 Recovery operation options.... 30
3 Minimum and recommended hardware requirements.... 32
4 Steps to backup data.... 49
5 Enter path to retrieve history for text box.... 66
6 Advanced options.... 82
7 Recovery operation options.... 89
8 Backup plug-in options.... 97
9 Backup options for the avsql command.... 100
10 Restore plug-in options.... 104
11 Restore options for the SQL TLR plug-in.... 107
12 Restore options for the SQL TLR plug-in.... 108
13 Operations for the avsql command.... 110
14 Common avsql options.... 111
15 Account options for the avsql command.... 111
16 Logging options for the avsql command.... 112
17 Browse option for the avsql command.... 113
18 Backup options for the avsql command.... 114
19 Restore options for the avsql command.... 118
20 avsqltlr option.... 133
21 Plugin descriptions.... 136
22 Operations for the av_cluster_config_console command.... 137
23 SQL Configure Options.... 137
24 SQL remove option.... 138
25 SQLAlwayson configure options.... 139
26 SQLAlwayson remove options.... 139
27 Component requirements.... 143

Figures

1 Avamar architecture with SQL Server 13
2 Avamar architecture with SQL Server in an active/passive cluster 15
3 Avamar architecture with SQL Server in an active/active cluster 16
4 Backup workflow on a primary replica in an AlwaysOn environment 17
5 Backup workflow on a secondary replica in an AlwaysOn environment 18
6 Restore workflow to only the primary replica in an AlwaysOn environment 19
7 Restore workflow to the primary and secondary replicas in an AlwaysOn environment 20
8 Types of database backups 22
9 Backup, Restore and Manage window 65
10 Backup, Restore and Manage window 68
11 Restore Command Line Options dialog box 86
12 Select Source Data Wizard for Differential Restore 94

Preface

As part of an effort to improve the product lines, revisions of the software and hardware are periodically released. Therefore, some functions that are described in this document might not be supported by all versions of the software or hardware currently in use. The product release notes provide the most up-to-date information on product features.

Contact the technical support professional when a product does not function correctly or does not function as described in this document.

NOTE: This document was accurate at publication time. To find the latest version of this document, go to Online Support (https://www.dell.com/support).

Purpose

This guide describes how to install Avamar in a Microsoft SQL Server database environment, and how to back up and restore SQL Server databases.

Audience

The information in this guide is primarily intended for:

- System administrators who are responsible for installing software and maintaining servers and clients on a network
- Microsoft SQL Server administrators who are responsible for backing up and maintaining Microsoft SQL Servers

Persons using this guide should have current practical experience with the following topics:

- Operating system shell commands on the SQL Server platform (root permission required)
- The specific version of Microsoft SQL Server currently deployed at the site

Revision history

The following table presents the revision history of this document:

Revision	Date	Description
04	July, 2021	Updated the following sections: ● Installing the Avamar Plug-in for SQL Server ● Silent installation of Itempoint for SQL
03	April, 2021	Updated the following sections: ● Copy-only backup ● Backups in AlwaysOn availability groups ● Requirements for the var directory ● Configuring the cluster client in a failover cluster ● Configuring the cluster client for an AlwaysOn availability group ● Configuring the cluster client for an AlwaysOn availability group Amazon Web Services and Azure
02	November, 2019	Updated the "Configuring the cluster client for an AlwaysOn availability group" section.
01	May 20, 2019	GA release of Avamar 19.1

Related documentation

The following publications provide additional information:

- *E-LAB Navigator* at https://elabnavigator.emc.com/eln/elnhome
- *Avamar Release Notes*
- *Avamar Administration Guide*
- *Avamar for Windows Server User Guide*
- *Avamar and Data Domain System Integration Guide*
- *Avamar Operational Best Practices Guide*
- *Avamar for Hyper-V VSS User Guide*
- *Avamar for VMware User Guide*

Typographical conventions

These type style conventions are used in this document.

Table 1. Typographical conventions

Bold	Used for names of interface elements, such as names of windows, dialog boxes, buttons, fields, tab names, key names, and menu paths (what the user specifically selects or clicks)
Italic	Used for full titles of publications that are referenced in text
`Monospace`	Used for: ● System code ● System output, such as an error message or script ● Pathnames, filenames, prompts, and syntax ● Commands and options
Monospace italic	Used for variables
`Monospace bold`	Used for user input
[ ]	Square brackets enclose optional values
|	Vertical bar indicates alternate selections - the bar means "or"
{ }	Braces enclose content that the user must specify, such as x or y or z
...	Ellipses indicate nonessential information that is omitted from the example

Where to get help

The Avamar support page provides access to licensing information, product documentation, advisories, and downloads, as well as how-to and troubleshooting information. This information may resolve a product issue before contacting Customer Support.

To access the Avamar support page:

1. Go to https://www.dell.com/support.
2. Type a product name in the **Enter a Service Tag, Serial Number, Service Request, Model, or Keyword** search box.
3. Select the product from the list that appears. When you select a product, the **Product Support** page loads automatically.
4. (Optional) Add the product to the **My Products** list by clicking **Add to My Saved Products** in the upper right corner of the **Product Support** page.

Documentation

The Avamar product documentation provides a comprehensive set of feature overview, operational task, and technical reference information. To supplement the information in product administration and user guides, review the following documents:

- Release notes provide an overview of new features and known limitations for a release.
- Technical notes provide technical details about specific product features, including step-by-step tasks, where necessary.
- White papers provide an in-depth technical perspective of a product or products as applied to critical business issues or requirements.

Knowledgebase

The Knowledgebase contains applicable solutions that you can search for either by solution number (for example, KB000xxxxxx) or by keyword.

To search the Knowledgebase:

1. Go to https://www.dell.com/support.
2. Under the **Support** tab, click **Knowledge Base**.
3. Type either the solution number or keywords in the search box. Optionally, you can limit the search to specific products by typing a product name in the search box and then selecting the product from the list that appears.

Online Communities

Go to Community Network at https://www.dell.com/community/ for peer contacts, conversations, and content on product support and solutions. Interactively engage online with customers, partners, and certified professionals for all products.

Live chat

To engage Customer Support by using live interactive chat, click **Join Live Chat** on the **Service Center** panel of the Avamar support page.

Service Requests

For in-depth help from Customer Support, submit a service request by clicking **Create Service Requests** on the **Service Center** panel of the Avamar support page.

NOTE: To open a service request, you must have a valid support agreement. Contact a sales representative for details about obtaining a valid support agreement or with questions about an account.

To review an open service request, click the **Service Center** link on the **Service Center** panel, and then click **View and manage service requests**.

Enhancing support

It is recommended to enable ConnectEMC and Email Home on all Avamar systems:

- ConnectEMC automatically generates service requests for high priority events.
- Email Home sends configuration, capacity, and general system information to Customer Support.

Comments and suggestions

Comments and suggestions help to continue to improve the accuracy, organization, and overall quality of the user publications. Send comments and suggestions about this document to DPAD.Doc.Feedback@emc.com.

Please include the following information:

- Product name and version
- Document name, part number, and revision (for example, 01)
- Page numbers

- Other details to help address documentation issues

1

Introduction

Topics:

- Architecture
- Backup
- Restore
- Table level recovery
- Disaster recovery

Architecture

You can use Avamar to back up and restore data in a variety of Microsoft SQL Server environments.

Avamar components

Required Avamar software components in a SQL Server environment include the Avamar Client for Windows, the Avamar Plug-in for SQL Server, and Avamar Administrator.

The following figure illustrates a basic system architecture, including required Avamar software components, when you use Avamar to back up a SQL Server environment.

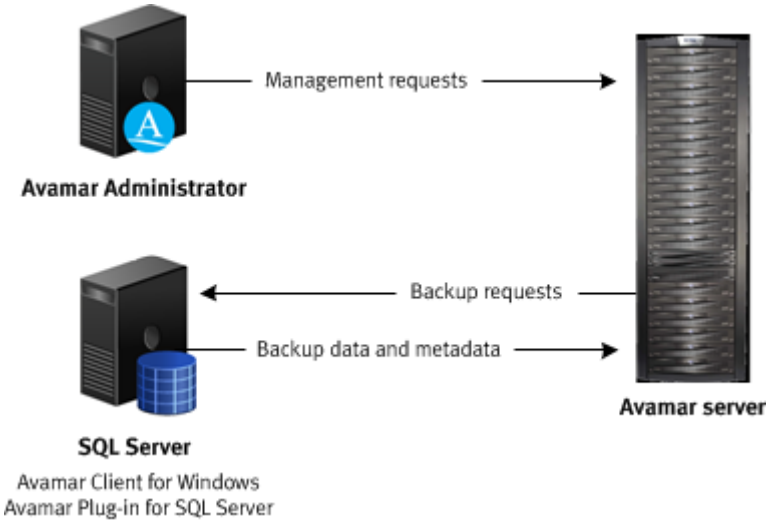

Figure 1. Avamar architecture with SQL Server

Install the Avamar Client for Windows and the Avamar Plug-in for SQL Server on the computer that is running SQL Server. Install Avamar Administrator on either the computer that is running SQL Server or on a separate computer, as shown in the figure.

Avamar Client for Windows

The Avamar Client for Windows installation includes the Avamar Plug-in for Windows File System and the Avamar agent, which is required for the Avamar Plug-in for SQL Server. You can use the Windows File System plug-in to back up operating system and SQL Server binary files, which are required for disaster recovery.

Avamar Plug-in for SQL Server

The SQL Server plug-in enables you to back up and restore SQL Server instances and databases.

Avamar Administrator

Avamar Administrator is a graphical management console software application for remote administration of an Avamar system from a supported Windows or Linux client computer. You can configure, perform, monitor, and manage backups and restores using Avamar Administrator. The *Avamar Administration Guide* provides complete instructions for installing and using Avamar Administrator.

How Avamar connects to SQL Server

The Avamar Plug-in for SQL Server uses the SQL Server virtual device interface (VDI) to back up and restore databases and transaction logs.

Backup process overview

1. The SQL Server plug-in creates one or more VDI devices, and then executes the necessary SQL commands to back up the database to the VDI devices.

   The number of VDI devices that the SQL Server plug-in creates depends on the number of data streams that you configure for a backup. Each data stream corresponds to a VDI device.

2. The Avamar `avtar` program is spawned in a standard input/output (stdio) mode.
3. The SQL Server plug-in reads data read from the VDI device and writes data to `avtar` standard input.
4. The `avtar` program reads and sends the data to either the Avamar server or a configured Data Domain system.

Restore process overview

During a restore, the SQL Server plug-in reads data from standard output and writes the data to the VDI device, which communicates with SQL Server to restore the databases.

Authentication

Avamar connects to SQL Server for backup or restore by using either Windows authentication or SQL Server authentication.

With Windows authentication, Avamar connects to SQL Server by using the Windows system service account (NT AUTHORITY\SYSTEM), which has privileges in SQL Server. The account must have the sysadmin server-level role.

With SQL Server authentication, Avamar connects to SQL Server by using a SQL Server login account. The account must have the sysadmin server-level role. You must select the mixed authentication mode for the Database Engine when you configure SQL Server.

If you do not specify an authentication method, then the SQL Server plug-in uses NT authentication and logs in with the Windows system service account. This account has the public and sysadmin server-level roles in SQL Server by default in SQL Server 2008 and 2008 R2. In SQL Server 2012, 2014, and 2016, you must add the account to the SQL Server administrators group.

Adding sysadmin server-level role in SQL Server Management Studio on page 37 provides more instructions on adding sysadmin server-level role in SQL Server Management Studio.

Failover Cluster Instances

You can install SQL Server in a Windows Server Failover Clustering (WSFC) cluster with two or more nodes.

With Failover Cluster Instances (FCIs), SQL Server databases and log files are on storage that is shared between the nodes. If one of the nodes fails, then the applications *fail over* to and run on another node. When the failed node comes back online, you can *fail back* the applications from the other node.

SQL Server can run as either active/passive or active/active in a cluster.

Active/passive cluster environments

In an active/passive configuration, SQL Server runs on the active node, which is the primary node. The passive node is a standby node. SQL Server does not run on the passive node.

If a failure occurs on the primary active node, then SQL Server fails over to the passive node. The passive node becomes the active node. When the primary active node comes back online, you can fail back SQL Server from the standby node to the primary active node. The passive node returns to its standby role.

Avamar can perform both on-demand and scheduled backups of SQL Server data while SQL Server is running on either the primary active node or on a standby node.

The following figure illustrates an active/passive cluster environment with SQL Server and Avamar.

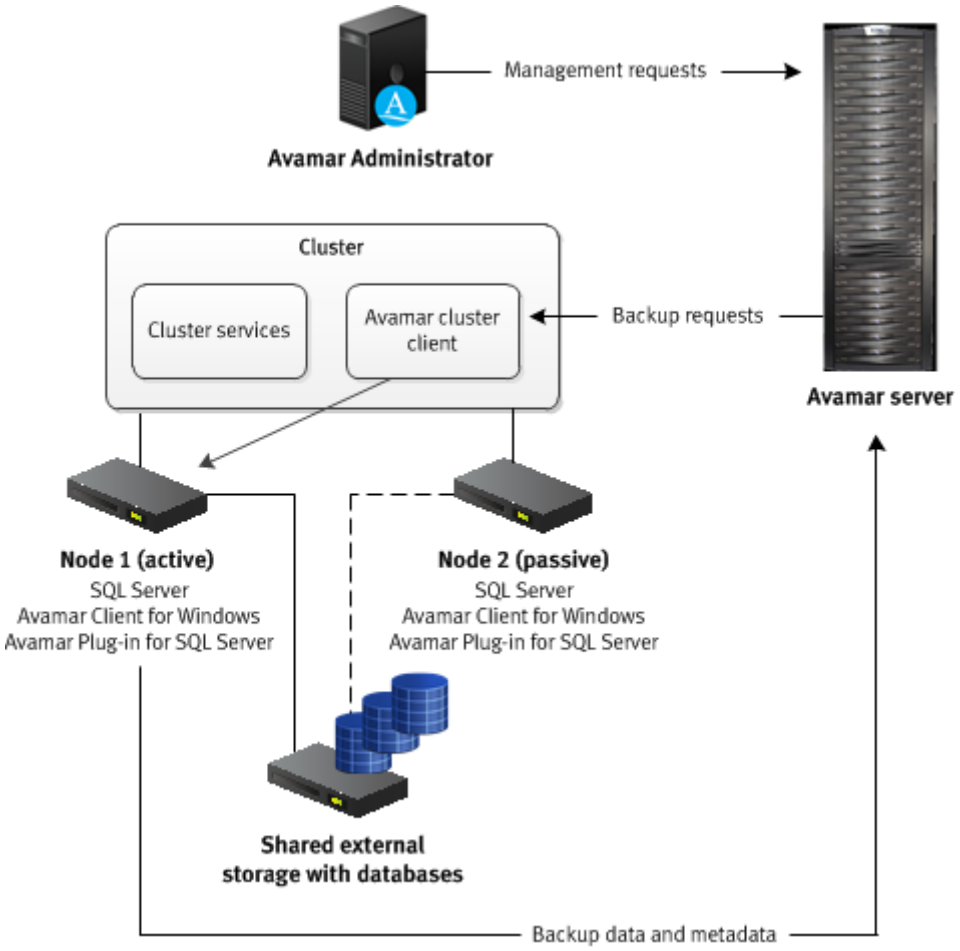

Figure 2. Avamar architecture with SQL Server in an active/passive cluster

Install the Avamar Client for Windows and the Avamar Plug-in for SQL Server on each node in the cluster. Then run the Cluster Configuration Tool to configure the Avamar cluster client.

The Avamar server sends backup requests to the Avamar cluster client, which passes the request to the SQL Server plug-in on the active node. The plug-in on the active node sends the backup data and metadata to the Avamar server.

On Windows Server 2012 R2, the Avamar Plug-in for SQL Server supports storage of SQL Server databases and log files in an active/passive cluster on shared disks, clustered shared volumes (CSVs), and shared virtual hard disk (VHDX) files. Shared VHDX files must be on either a CSV or on a Server Message Block (SMB) 3.0 file share.

NOTE: For a SQL cluster on CSV, the CSV disk does not display as part of the SQL Cluster in the **Avamar Windows Cluster Configuration** wizard. This is because the powershell commands do not provide the associated disk for the SQL cluster in a CSV environment. To configure a client for the SQL cluster, a non-csv disk needs to be assigned to the SQL Server role.

Active/active cluster environments

In an active/active configuration, SQL Server runs on both nodes. The SQL Server installation on each node manages its own separate databases.

If a failure occurs on one of the nodes, then the SQL Server installation on the other node assumes responsibility for managing the databases for both nodes. When the failed node comes back online, you can return the management responsibility for the databases to the original SQL Server installation on each node.

Avamar can perform both on-demand and scheduled backups of SQL Server data regardless of which active node is managing the data at the time of the backup.

The following figure illustrates an active/active cluster environment with SQL Server and Avamar.

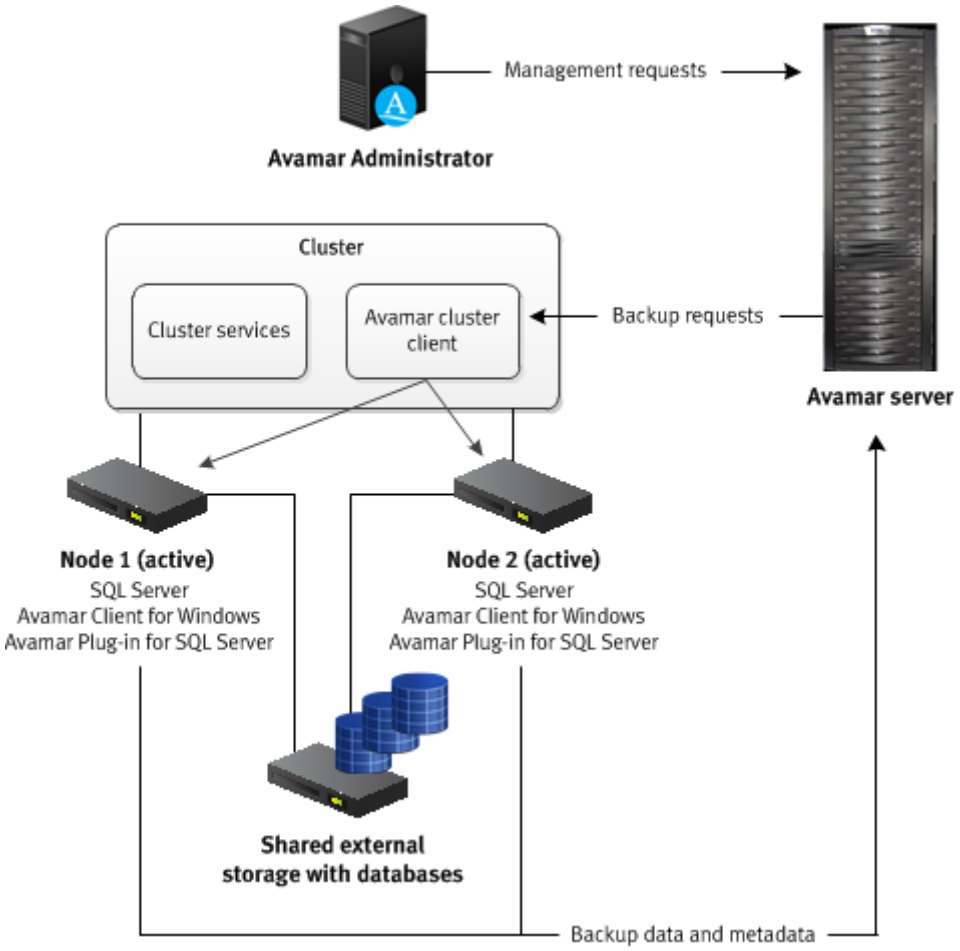

Figure 3. Avamar architecture with SQL Server in an active/active cluster

You install the Avamar Client for Windows and the Avamar Plug-in for SQL Server on each node in the cluster. Then you run the Cluster Configuration Tool to configure the Avamar cluster client. When you perform a backup or restore, you select the cluster client as the client to back up or the target client for the restore.

The Avamar server sends backup requests to the Avamar cluster client, which passes the request to the SQL Server plug-in on both active nodes. The plug-ins on the active nodes back up the SQL Server data that they are managing to the Avamar server.

AlwaysOn availability groups

You can configure high availability of groups of databases by using AlwaysOn availability groups (AGs) in SQL Server 2012, 2014, and 2016.

With AGs, SQL Server is installed in a WSFC cluster, but the data is not stored on a shared drive. Instead, the data is stored on each node, and SQL Server synchronizes the data from the primary version of the database to any secondary versions on other nodes.

You can group user databases together in an availability group. Databases in an availability group must use the full recovery model. All databases in an availability group fail over together from one node to another.

A SQL Server instance on a cluster node that hosts an availability group is called an *availability replica*. Each availability replica of an availability group must reside on a different node of the same cluster. There are two types of availability replicas: one primary replica and one or more secondary replicas. The primary replica handles read/write activity from clients and sends transaction log updates to the secondary replicas. Each secondary replica applies the transaction log updates to its databases.

During failover of an AG, the target secondary replica assumes the primary role and becomes the new primary replica. The target secondary replica brings its databases online as the primary databases, and client applications can connect to them. When the former primary replica is available, it assumes the secondary role and becomes a secondary replica.

NOTE: The Avamar Plug-in for SQL Server does not support backups of AlwaysOn availability groups on an availability replica that is hosted by a SQL Server Failover Cluster Instance (FCI). To back up databases in such an environment, you must install the Avamar client software on the node with the primary replica as a stand-alone client, and then perform backups on only that node.

Architecture and workflow of backups in AlwaysOn availability groups

Avamar can perform both on-demand and scheduled backups of databases in either the primary replica or a secondary replica for an availability group.

Install the Avamar Client for Windows and the Avamar Plug-in for SQL Server on each node in the cluster. Then run the Cluster Configuration Tool to configure the Avamar cluster client for the availability group listener. When you perform a backup or restore, select the cluster client for the availability group listener as the client to back up or the target client for the restore.

Backups on the primary replica

The following figure illustrates the data workflow of SQL Server plug-in backups on the primary replica.

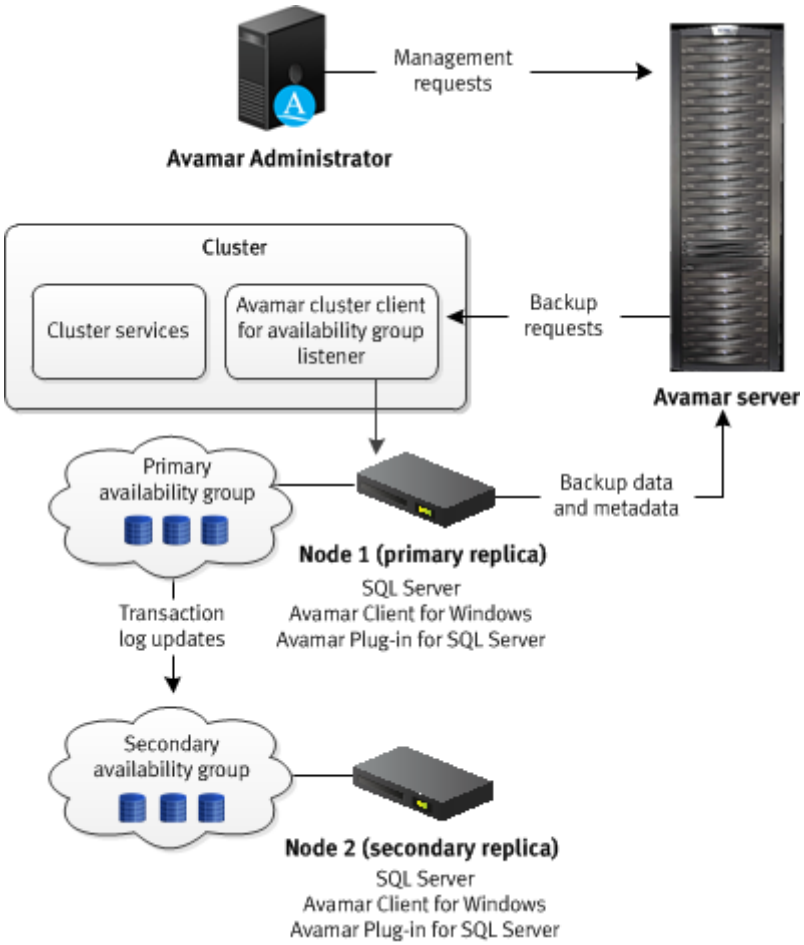

Figure 4. Backup workflow on a primary replica in an AlwaysOn environment

The Avamar server sends a backup request to the Avamar cluster client for the availability group listener, which passes the request to the primary replica. The SQL Server plug-in on the node with the primary replica performs the backup and sends the backup data and metadata to the Avamar server.

Backups on the secondary replica

The following figure illustrates the data workflow of SQL Server plug-in backups on a secondary replica.

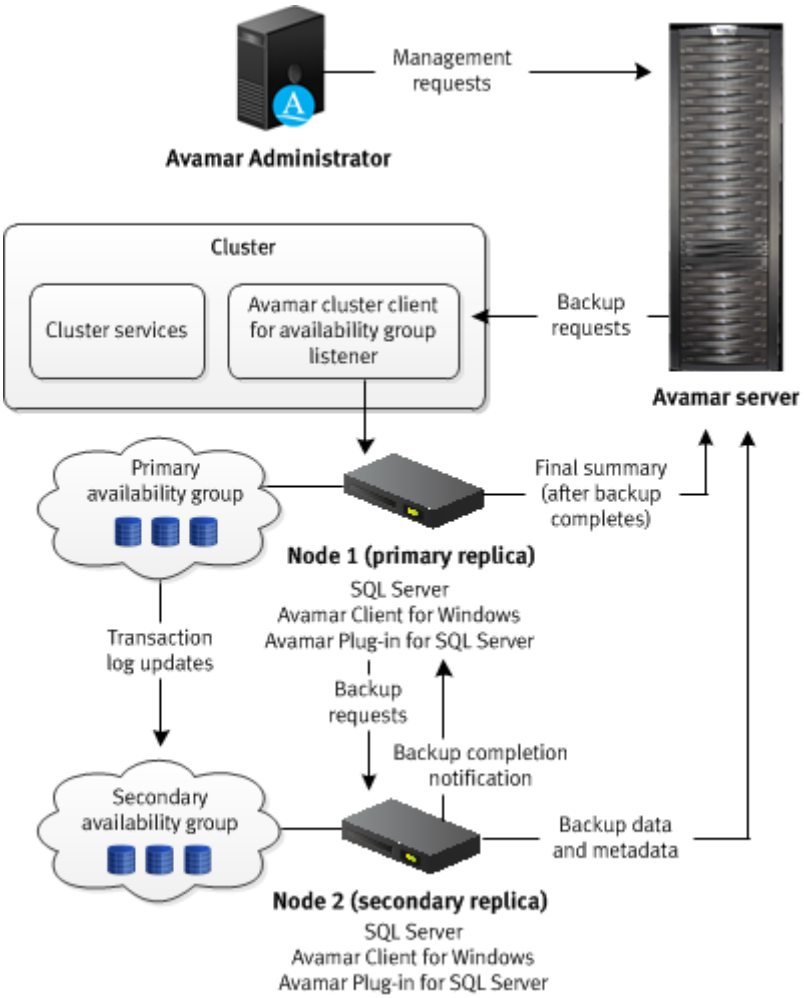

Figure 5. Backup workflow on a secondary replica in an AlwaysOn environment

The Avamar server sends a backup request to the Avamar cluster client for the availability group listener, which passes the request to the primary replica. The SQL Server plug-in on the node with the primary replica determines the secondary replica on which to perform the backup, and forwards the backup request to the SQL Server plug-in on the node with the secondary replica. The plug-in on the node with the secondary replica performs the backup and then sends the backup data to the Avamar server. The plug-in on the secondary replica then sends the metadata to the Avamar server and notifies the plug-in on the primary replica whether the backup completed successfully. The primary replica sends the final summary to the Avamar server.

Architecture and workflow of restores in AlwaysOn availability groups

You can use the backup from either the primary replica or a secondary replica to restore the databases in an AlwaysOn availability group. You can restore to only the primary replica, or restore to both the primary and secondary replicas.

Restore to only the primary replica

The following figure illustrates the process to restore to only the primary replica in an AlwaysOn availability group.

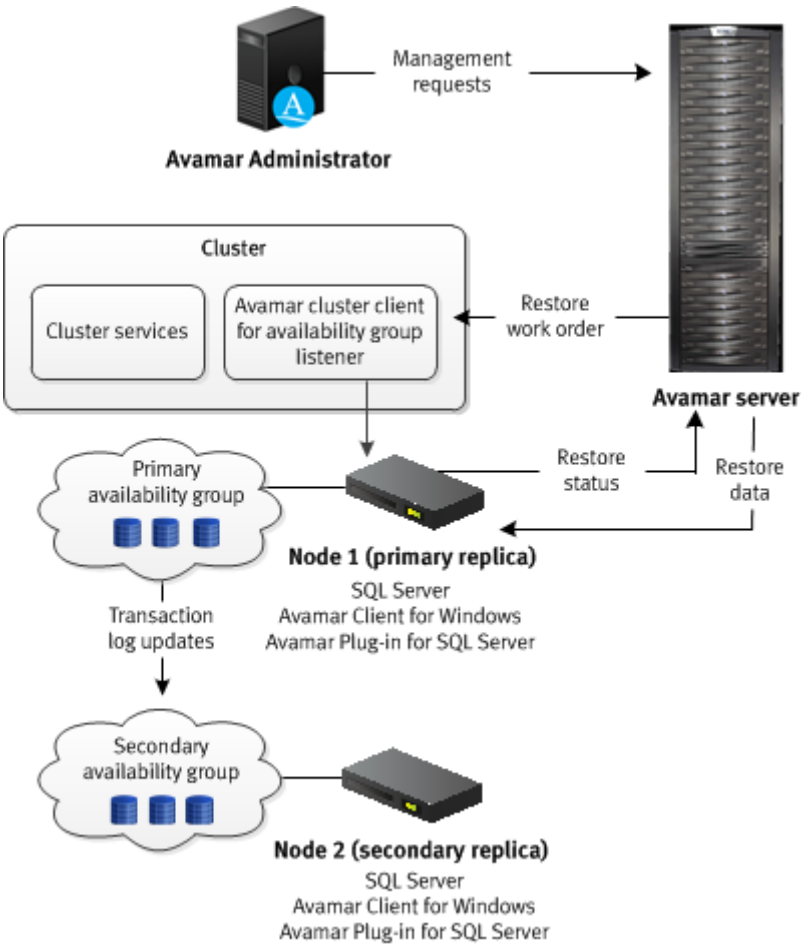

Figure 6. Restore workflow to only the primary replica in an AlwaysOn environment

The Avamar server sends a restore work order to the cluster client for the availability group listener, which passes the work order to the SQL Server plug-in on the node with the primary replica. The plug-in on the primary replica removes the databases that you are restoring from the availability group and restores the databases. After the restore completes, the plug-in on the primary replica adds the restored databases to the availability group and sends information about the restore to the Avamar server.

If there is a corresponding database on the secondary replicas when you restore a database only on the primary replica, then the database on the secondary replicas is in a restoring state. To restore the databases on the secondary replicas as part of the availability group, manually prepare and restore the databases, and join them to the availability group on the secondary replica.

You can also set the database on a secondary replica online without rejoining it to the availability group by restoring the database with the `RECOVERY` recovery operation (`RESTORE database WITH RECOVERY`). The SQL Server documentation on the Microsoft TechNet website provides details.

Restore to both the primary and secondary replicas

The following figure illustrates the process to restore to both the primary and secondary replicas.

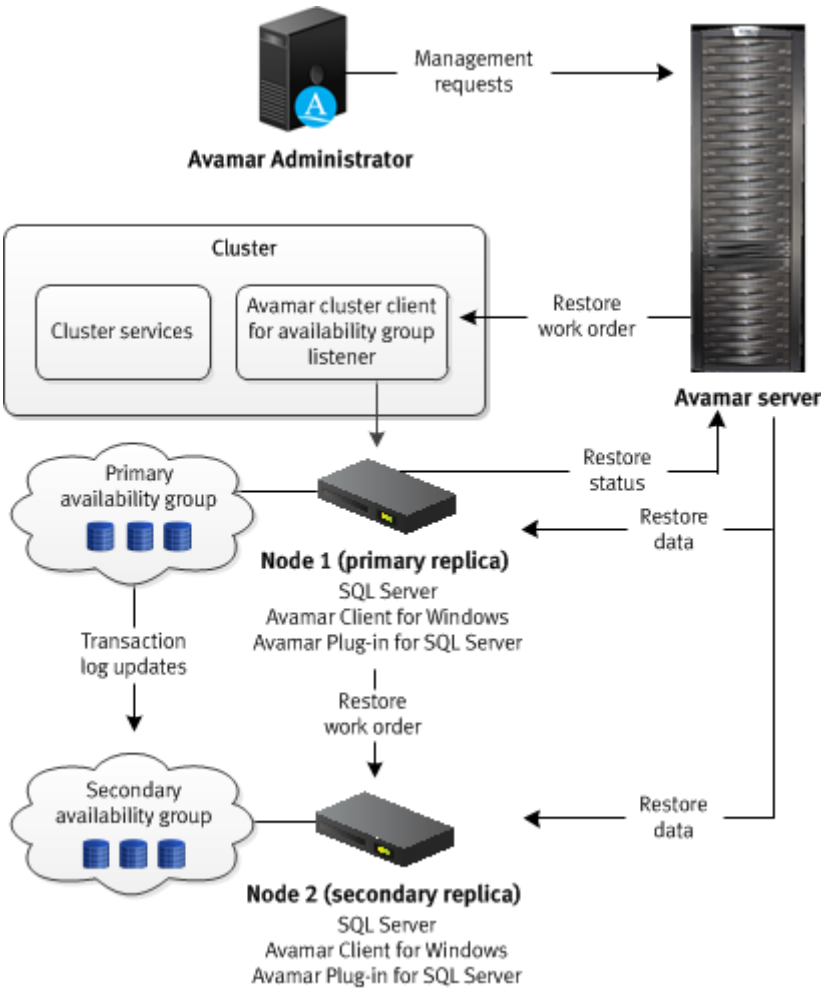

Figure 7. Restore workflow to the primary and secondary replicas in an AlwaysOn environment

During the restore, the Avamar server sends a restore work order to the SQL Server plug-in on the node with the primary replica. The plug-in on the node with the primary replica then sends the restore work order to the plug-in on the nodes with the secondary replicas. The plug-ins on the primary and secondary replicas remove the databases that you are restoring from the availability group and restore the databases.

After the restore completes, the plug-in on the node with the primary replica adds the restored databases to the availability group. Then the plug-ins on the nodes with the secondary replicas join the restored databases to the availability groups. When the entire restore process completes on all replicas, the plug-in on the node with the primary replica sends information about the restore to the Avamar server.

Hybrid IT environments with AlwaysOn availability groups and Microsoft Azure

The Avamar Plug-in for SQL Server can perform both on-demand and scheduled backups of databases in AlwaysOn availability groups in hybrid IT environments with Microsoft Azure.

The backup can occur on either the primary replica or the secondary replica of the availability group, but the replica on which the backup occurs must be on-premise. Avamar cannot back up databases on a replica on a Microsoft Azure virtual machine.

The Avamar installation process is the same in a hybrid IT environment as in a traditional AlwaysOn availability group environment. Install the Avamar Client for Windows and the Avamar Plug-in for SQL Server on each node in the cluster, including the Microsoft Azure nodes. Then run the Cluster Configuration Tool to configure the Avamar cluster client for the availability group listener.

When you perform a backup or restore, select the availability group listener as the client to back up or the target client for the restore. Ensure that backups occur on an on-premise replica by using SQL Server and Avamar settings.

If the primary replica is on-premise, restore data to only the primary replica. Do not attempt to restore to both the primary and secondary replicas. If there is a corresponding database on the secondary replicas when you restore a database only on the primary replica, then the database on the secondary replicas is in a restoring state. To restore the databases on the secondary replicas as part of the availability group, manually prepare and restore the databases, and join them to the availability group on the secondary replica.

If the primary replica is on a Microsoft Azure virtual machine, then restore the database to operating system files and then use SQL Server tools to restore the database to the availability group.

Database mirroring

Database mirroring maintains multiple copies of a single database that must reside on different SQL Server instances. Typically, these server instances reside on computers in different locations.

The principal server serves the database to clients, while the mirror servers serve as standby servers.

Implement mirroring settings individually for each database.

Mirroring works only with databases that use the full recovery model. The simple and bulk-logged recovery models do not support database mirroring, and you cannot mirror the master, msdb, tempdb, or model databases.

You can use the Avamar Plug-in for SQL Server to back up mirrored SQL Server databases. However, several conditions apply:

- The SQL Server version must be 2008 or greater.
- Perform backups of only the principal database, not the mirrors.
- Use only the Avamar Plug-in for SQL Server to perform backups. If you use other backup products in addition to the Avamar Plug-in for SQL Server, then log chain breaks may occur.
- Backup and database administrators must use extra care. Some mirror operations cause log chain breaks that cannot be detected.
- When database mirroring is established, either initially or as the result of failover and failback, then you must manually perform a new full backup. Otherwise, incremental and differential backups that occur after the establishment of database mirroring are not valid for restore. After you perform the full backup in this case, you cannot perform point-in-time recoveries to a point in time before the full backup.
- To restore a database, you must break the SQL mirror.

Data Domain system support

You can store backups on either the Avamar server or a Data Domain® system. Backup metadata is stored on the Avamar server.

Before you can store backups on a Data Domain system, add the Data Domain system to the Avamar configuration by using Avamar Administrator. Then select the Data Domain system in the plug-in options when you perform an on-demand backup or when you create a dataset for a scheduled backup. You can also use the command line interface (CLI) to perform backups to a Data Domain system.

The steps to restore backups are the same whether you restore from the Avamar server or a Data Domain system. The restore process determines the location of the backup and restores the backup.

Avamar also supports Data Domain Cloud Tier. DD Cloud Tier moves data from Data Domain to the cloud. From the Avamar Administrator, you can configure cloud tiering to move Avamar backups from Data Domain to the cloud, and perform seamless recovery of these backups.

The *Avamar and Data Domain System Integration Guide* provides more information about Data Domain systems in an Avamar environment, including detailed steps to add a Data Domain system to the Avamar configuration.

Server virtualization

You can install SQL Server in a server virtualization environment such as VMware or Microsoft Hyper-V. There are multiple ways that you can install and use Avamar to back up and restore SQL Server data in a server virtualization environment.

The *Avamar for Hyper-V VSS User Guide* and *Avamar for VMware User Guide* provide details on additional system requirements, as well as installation and configuration procedures.

Backup

When you perform a backup with the Avamar Plug-in for SQL Server, you can back up either all SQL Server data on a specific server, one or more instances, or one or more databases.

You cannot use the Avamar Plug-in for SQL Server to back up individual filegroups, files, or tables. To back up individual files, use the Avamar Client for Windows.

The SQL Server plug-in can back up both user databases and system databases such as the master, msdb, and model databases. Backup of system databases in a SQL Server replication environment, such as the publication, distribution, and subscription databases, is also supported.

The SQL Server plug-in does not support backup of either the Resource or tempdb database because Microsoft SQL Server does not support backup and restore of those system databases.

Avamar can back up databases that use any of the three recovery models: simple, full, or bulk-logged. However, the recovery model may determine the type of backup that you can perform of the database.

NOTE: The Avamar AUI is only supported in stand-alone environments.

Backup types

The Avamar Plug-in for SQL Server supports full, differential, transaction log (incremental), and copy-only database backups.

The following figure illustrates the different types of data that are included in each of the backup types.

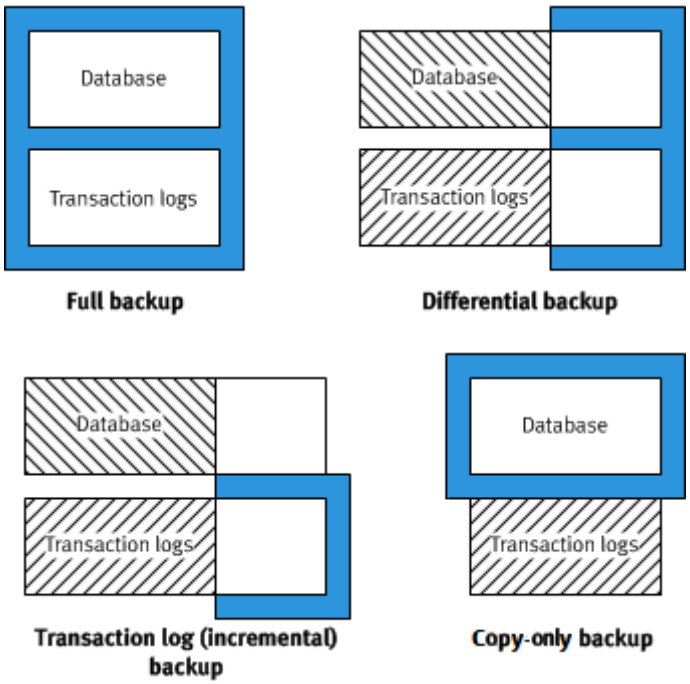

Figure 8. Types of database backups

Full backups

Full backups include the entire database, including all objects, system tables, and data. As the backup operation progresses, it copies the transaction logs. The timing of the transaction log backup ensures that you can recover the complete database to the state it was in when the backup finished.

Differential backups

Differential backups include data that changed after the last full backup. The backup operation also copies relevant portions of the transaction logs.

When you select a differential backup to restore a database, the restore process restores the last full backup, followed by the differential backups performed after the full backup. Because a differential backup only saves changes to data, the backup is smaller and faster than a full backup. Therefore, you can perform a differential backup more often than a full backup.

Differential backups are used primarily in heavily utilized systems where you must bring a failed database back online as quickly as possible. Differential backups are smaller than full backups, and so have less effect on the system where they run.

Transaction log (incremental) backups

By default, transaction log backups only back up the transaction logs. Transaction logs are serial records of all database modifications.

The logs are used in recovery operations to update the database with complete transactions and roll back incomplete transactions. When you back up a transaction log, the backup stores all changes after the last transaction log backup. Transaction log backups record the state of the transaction log at the start of the backup (unlike full and differential backups, which record the state of the data at the end of the backup).

When a transaction log backup is complete, the log truncates to remove any transactions that are committed to the database. When restoring the database, restore the data to the state it was in at the end of the last full or differential backup, and then sequentially restore the transaction log backups in order.

To fully restore data from a transaction log backup, at least one full backup must exist. To ensure data integrity, the Avamar Plug-in for SQL Server software always checks for the presence of a full backup on the server. If there is a full backup, the transaction log backup proceeds (that is, the backup includes only transaction logs). If there is no full backup and you select the **Force full backup** option (the default setting), then the Avamar Plug-in for SQL Server software forces a full backup to ensure data integrity.

Copy-only backups

Copy-only backups are special-purpose full backups that are created independently of the sequence of full, differential, and transaction log backups. After you perform a copy-only backup, the next differential or transaction log backup is based on the last full backup, not the new copy-only backup.

Copy-only backups are supported for all recovery models, and for performing backups at the database, instance and stripe levels. Restoring from a copy-only backup is the same as restoring from a full backup.

Copy-only backups cannot be used as a base for differential backups and do not truncate the transaction log. If you perform a copy-only backup on a database without a current full backup, the Avamar Plug-in for SQL Server software promotes the next differential or transaction log backup to a full backup. Promotion to full is indicated in the Activity Window of the Avamar Administrator with the status **Completed with Promotion**. When only some of the databases in the backup are promoted to full, the status will still be indicated as **Completed with Promotion**.

Previous versions of Avamar identify copy-only backups as type `Unknown`. Copy-only backup is not supported in AlwaysOn Availability environments.

Supported backup types for different recovery models

The type of database backup that you can perform depends on the recovery model of the database.

You can perform full backups of all databases, regardless of the recovery model (simple, full, or bulk-logged).

If the database uses the simple recovery model, then you cannot perform a transaction log backup of the database. Databases with the simple recovery model do not support transaction log backups. System databases such as the master and msdb databases typically use the simple recovery model.

Microsoft SQL Server also does not allow differential backups of the master database. You can only create full backups of the master database.

If a transaction log backup includes databases with the simple recovery model and databases with other recovery models, then you can select how Avamar handles the databases with the simple recovery model. You can exclude the databases with the simple recovery model and log either a warning or an error message in the log, or you can automatically perform a full backup of the databases. When you perform the backup, the **For simple recovery model databases** option controls this behavior.

Databases in AlwaysOn availability groups must use the full recovery model.

If you change the recovery model of a database, perform a full backup before you try to perform a differential or transaction log backup.

Backups in AlwaysOn availability groups

Avamar can perform both on-demand and scheduled backups of databases in either the primary replica or a secondary replica for an availability group. You can increase performance of the primary replica when you perform backups on a secondary replica.

Settings in SQL Server and in Avamar enable you to specify a preference for the replica on which the backup occurs:

- Always on the primary replica
- Always on a secondary replica
- On a secondary replica if possible, otherwise on the primary replica
- No preference defined

In a hybrid IT environment with Microsoft Azure and AlwaysOn availability groups, use these settings to ensure that the backup occurs on an on-premise replica. Avamar cannot back up databases on a replica on a Microsoft Azure virtual machine.

If you do not specify a preference, then other factors contribute to the selection of the replica for backup, including the backup priority, operational status, and connection status of each replica.

If you perform the backup on a secondary replica, then the replica must meet the following requirements:

- The replica must be in a synchronized state.
- The replica role setting in SQL Server must allow either read-only connections or all connections.
- If there are multiple secondary replicas, then set the priority for which secondary replica to use for the backup.

If you select the option to perform the backup only on a secondary replica and no secondary replicas that meet the requirements are available, then the backup fails with an error.

You can perform full, differential, and transaction log backups on the primary replica. You can perform only full and transaction log backups on a secondary replica. If you try to perform a differential backup on a secondary replica, then the backup fails with an error message.

SQL Server does not support system databases in an availability group. To back up system databases with Avamar, you must perform a separate backup. During the backup, select the server name of the physical node as the client to back up, and then select the system databases.

NOTE: Copy-only backup is not supported in AlwaysOn Availability environments.

Mixed backup storage

You can store backups on either the Avamar server or a Data Domain system. However, the full backup for a client and all subsequent transaction log (incremental) and differential backups must be stored on either the Avamar server or a single Data Domain system.

Avamar does not support the following scenarios:

- Full backup on a Data Domain system, and transaction log or differential backups on the Avamar server
- Full backup on the Avamar server, and transaction log or differential backups on a Data Domain system
- Full backup on one Data Domain system, and transaction log or differential backups on another Data Domain system

NOTE: An exception to this restriction is if you perform a tail-log backup during a restore. The tail-log backup is stored on the Avamar server even if the other backups for the client are stored on a Data Domain system.

As a result, if you change the server on which backups for a client are stored, then you must perform a full backup before any further transaction log or differential backups.

If you change the backup storage from one Data Domain system to another Data Domain system, you can restore to a point in time between the two full backups.

Multistreaming

Multistreaming enables you to improve backup and restore performance by backing up and restoring SQL Server data by using multiple parallel data streams.

You can either back up multiple databases in parallel with one stream per database, or back up a single database with multiple parallel streams.

If you use multiple data streams to send backup data for a single database to the Avamar server or Data Domain system, then the backup for the database is stored as multiple files. As a result, the restore uses the same number of streams that you use for the backup.

You can specify a maximum of 10 streams for each backup, and the minimum size of a stream.

Database log truncation

The Avamar Plug-in for SQL Server truncates the database transaction log after an incremental (transaction log) backup of the database.

You can truncate the database transaction log by performing an incremental backup automatically after a full backup in SQL Server 2008 and later.

You also can prevent truncation of database transaction logs after backups.

The **Truncate database log** option controls whether truncation occurs. To automatically perform an incremental backup after a full backup, select the **Force incremental backup after full backup** option.

When the backup process successfully truncates the log file, `LOG_BACKUP` appears in the header of the database.

NOTE: Truncation does not reduce the physical size of a log file. To reduce the physical size of a log file, shrink the log file. The Microsoft TechNet website provides information on log shrinking in SQL Server.

Backups with other tools

Issues can occur when you perform backups with other tools in addition to backups with the Avamar Plug-in for SQL Server.

Backups with multiple tools can break the backup chain for all of the tools, especially if the backups are truncating the database transaction log. Consider the following scenario:

1. You perform a transaction log backup of a database with the SQL Server plug-in. The backup process truncates the log.
2. The next day you perform transaction log backup with SQL Server management tools. The backup process truncates the log.
3. The next time that a transaction log backup with the SQL Server plug-in occurs, the backup is missing the transactions from the last Avamar backup through the SQL Server backup.

Exclusive backups with the Avamar Plug-in for SQL Server are recommended to avoid breaks in the backup chain.

Restore

The Avamar Plug-in for SQL Server supports a variety of restore options to accommodate a wide range of data recovery needs.

The SQL Server plug-in performs only offline restores of SQL Server data. Online restore is not supported.

NOTE: The Avamar AUI is only supported in stand-alone environments.

Restore to the original location

There are two options when you use the Avamar Plug-in for SQL Server to restore a SQL Server instance, database, or filegroup to its original location:

- Perform a standard restore with a tail-log backup and recovery.
- Use the SQL Server `REPLACE` option to completely overwrite the database.

A standard restore with a tail-log backup is perhaps the most common restore procedure. During this procedure, a tail-log backup is created to capture transactions that are not in a backup. Then Avamar restores the database from the most recent full backup and any differential or transaction log backups.

You may need to use the SQL Server `REPLACE` option for a restore, for example, if a previous database restore exited with the following SQL Server error in the Avamar SQL restore log:

```
One or more devices or files already exist.
Reissue the statement using the WITH REPLACE
option to overwrite these files and devices.
```

NOTE: When you use the Avamar plug-in option for the SQL Server `REPLACE` option, it adds an SQL `WITH REPLACE` clause statement to the restore Transact-SQL command. Use of this statement overrides a SQL Server safety check to prevent you from accidentally overwriting a different database or file. The *Microsoft Transact-SQL Reference Manual* provides more information about the safety check in the `RESTORE` command section.

Restore to a new database in the original instance

You can use the Avamar Plug-in for SQL Server to restore a backup of a database to its original instance, but with a new name that creates a database in the instance.

When you restore to a new database in the original instance, you can perform a tail-log backup and recovery to recover transactions that were not in the backup. You also can specify the path for the database and log files.

If the database uses the full recovery model, then you can restore to either a specific date and time or to a named mark in the transaction log.

Restore to a different instance on the original server

You can use the Avamar Plug-in for SQL Server to restore an instance, database, filegroup, or file to a different instance on the original server.

When you restore to a different instance on the original server, you cannot perform a tail-log backup. However, you can specify the path for the database and log files.

If you are restoring a database, you can leave the original database name or restore the database with a new name. You can also restore to either a specific date and time or to a named mark in the transaction log.

Restore to an instance on a different server

You can use the Avamar Plug-in for SQL Server to restore an instance, database, filegroup, or file to an instance on a different server.

When you restore to an instance on a different server, you cannot perform a tail-log backup. However, you can specify the path for the database and log files.

If you are restoring a database, you can leave the original database name or restore the database with a new name. You also can restore to either a specific date and time or to a named mark in the transaction log.

Restore to a file

If the Avamar Plug-in for SQL Server is not installed on the target server, or you want to use the standard SQL Server restore tools for other features, then you can restore user or system databases to operating system files. You can then use SQL Server

tools, such as SQL Server Management Studio, a Transact-SQL `RESTORE` command, or the Microsoft SQL Server `sqlcmd` utility, to restore the databases.

The SQL Server plug-in restores the backup as one or more files to the specified destination in the following path:

`destination\client\instance\database\file`

where:

- *destination* is the destination for the files that you specified in the **Set Destination** dialog box.
- *client* is the name of the computer on which SQL Server is installed.
- *instance* is the name of the SQL Server instance from the backup.
- *database* is the name of the database from the backup.
- *file* is the name of the file.

  A single backup may include multiple files, depending on the number of streams in the backup. The file name for each file is composed of the backup type and the stream number:

  - f-0 indicates a full backup.
  - d-*n* indicates a differential backup.
  - i-*n* indicates a transaction log (incremental) backup.

  where *n* is the sequential number of the differential or incremental backup after the preceding full backup.

For example, a full backup with two streams results in two files: `f-0.stream0` and `f-0.stream1`.

Restore to an AlwaysOn availability group

When you back up a database in an AlwaysOn availability group, you can restore the database to the following locations:

- To the original availability group
- To a new database in the original availability group
- To a different availability group
- To a different instance that does not use availability groups
- To a file

You can restore an entire database, a filegroup, or a file. You also can perform point-in-time restore of databases that use the full recovery model.

These restore options are available regardless of whether the backup was on the primary replica or a secondary replica.

Restore to the original availability group

When you restore to the original availability group, the restore process can automatically restore the databases on both the primary replica and the secondary replicas.

You can also restore databases only on the primary replica. When you restore a database only on the primary replica, the corresponding database on the secondary replicas is in a restoring state. To restore the databases on the secondary replicas as part of the availability group, manually prepare and restore the databases, and join them to the availability group on the secondary replicas.

You can also set the databases on a secondary replica online without rejoining them to the availability group by restoring the databases with the `RECOVERY` recovery operation. The SQL Server documentation on the Microsoft TechNet website provides details.

Restore to a new database in the original availability group

You can restore a database to the original AlwaysOn availability group, but with a new name that creates database in the availability group.

When you restore to a new database in the original instance, the restore process restores the database to only the primary replica. You must then join the new database to the availability group.

To restore the database on the secondary replicas as part of the availability group, manually restore the database and join the database to the availability group on each secondary replica.

Restore to a different availability group

When you restore a database to an availability group on a different server, the restore process restores the database to only the primary replica.

To restore the database on the secondary replicas as part of the availability group, manually restore the database and join the database to the availability group on each secondary replica.

NOTE: You cannot restore a database to a different availability group in the same cluster. Restore fails with an error message that indicates the existence of the database.

Restore to a different instance that does not use availability groups

You can restore a database to a different SQL Server instance that does not use AlwaysOn availability groups. The steps are the same whether you restore from a backup of databases on a stand-alone server, in a cluster, or in an availability group.

Restore of a database from an availability group to a file

You can restore a database from a backup of an availability group to operating system files. This type of restore may be necessary if the Avamar Plug-in for SQL Server is not installed on the target server, or you want to use the standard SQL Server restore tools for other features. The steps are the same whether you restore from a backup of databases on a stand-alone server, in a cluster, or in an availability group.

Restore to hybrid IT environments with Microsoft Azure

In a hybrid IT environment with AlwaysOn availability groups and Microsoft Azure, the steps to restore a database depend on the location of the primary replica.

If the primary replica is on-premise, restore data to only the primary replica. Do not attempt to restore to both the primary and secondary replicas. If there is a corresponding database on the secondary replicas when you restore a database only on the primary replica, then the database on the secondary replicas is in a restoring state. To restore the databases on the secondary replicas as part of the availability group, manually prepare and restore the databases, and join them to the availability group on the secondary replica.

If the primary replica is on a Microsoft Azure virtual machine, then restore the database to operating system files and then use SQL Server tools to restore the database to the availability group.

Restore of a database with an intact log file

If a database becomes corrupt or is otherwise lost but an intact database log file is available, you can restore the database and use the log file to recover transactions after the most recent Avamar backup.

To restore the database, perform a transaction log backup by using a Transact-SQL command. Then, restore the database from the most recent Avamar backup by using Avamar Administrator. Finally, restore the transaction log backup by using Transact-SQL commands.

Restore of system databases

Restore of only system databases is rare but may be necessary if one or more system databases are damaged. It is more likely that you must restore system databases when you restore user databases. When you select both system and user databases for restore, the system databases restore first.

When you restore system databases, the Avamar Plug-in for SQL Server can automatically restore the databases in the correct order and manage SQL Server services. However, you can also restore individual system databases and manually manage the services.

NOTE: The Avamar Plug-in for SQL Server can back up and restore system databases in a SQL Server replication environment, including the publication, distribution, and subscription databases. The SQL Server plug-in can automatically restore the replication system databases in the correct order and manage SQL Server services. The "Back Up and Restore

Replicated Databases" topic in the SQL Server documentation on the MSDN website provides details on backup strategies and the steps to manually restore databases in different types of replication environments.

Automatic restore of system databases

When you restore multiple system databases, Avamar automatically restores the databases in the correct order: master, msdb, and model.

Avamar can also automatically manage the stop and restart of the necessary SQL Server services during the restore. For example:

- When you restore the master database, Avamar can automatically stop the SQL Server instance, including dependent services such as the SQL Server agent service and the Analysis Service, and restart the instance in single-user mode before the restore. After the restore, Avamar automatically restarts the instance.
- When you restore the msdb database, Avamar can automatically stop the SQL Server agent service, and then restart it when the restore is complete.

Manual restore of system databases

When you restore system databases manually, you must manage the services and restore the databases in the correct order.

1. Shut down the SQL Server instance and then restart the instance in single-user mode.
2. Restore the master database.
3. Restart the SQL Server service.
4. Stop the SQL Server Agent service.
5. Restore the msdb and model databases.
6. Restart the SQL Server Agent service.

Tail-log backup

With a tail-log backup, Avamar backs up the tail of the transaction log during the restore process to capture the log records that are not in a backup. After the database restore, Avamar uses the tail-log backup to recover the transactions that were not in the backup.

To perform a tail-log backup, the database must be online and using either the full or bulk-logged recovery model. As a result, you cannot perform a tail-log backup of system databases such as the master and msdb databases because those databases use the simple recovery model.

When you are restoring a user-defined filegroup or secondary data file and you perform a tail-log backup, you must select the most recent backup as the backup from which to restore. Otherwise, the restore fails and an error message is written to the log file.

You can perform a tail-log backup when you are restoring an instance, database, filegroup, or file to its original location without the SQL `WITH REPLACE` option.

You can also perform a tail-log backup when you are restoring a database to the original instance but with a new database name.

If you are performing a point-in-time restore and the point in time that you are restoring to is after the most recent transaction log backup, then you must perform a tail-log backup.

A tail-log backup is also necessary if you restore a file from a user-defined filegroup to its original location.

Do not perform a tail-log backup if you are performing a redirected restore to a different SQL Server instance.

NOTE: If the tail-log backup fails to complete, then the restore cannot take place. Review the log file to determine the cause of the failure. Correct the problem, and then restart the restore. Keep in mind that if you clear the **Tail-log backup** checkbox to prevent the tail-log backup from occurring, then the restore includes only the transactions up to the selected backup. Any transactions in the tail of the log are lost.

Point-in-time restore

You can restore a database with the full recovery model to a specific date and time or to a named mark in the transaction log.

You cannot perform a point-in-time restore of system databases such as the master and msdb databases because those databases use the simple recovery model.

To restore to a specific point in time, you must provide the transaction date and time or named mark to which to recover. This information is available in the SQL Server transaction log. The SQL Server documentation on the Microsoft TechNet website provides details on how to access transaction log information.

The point in time to which you are restoring must be after the finish time for the most recent full backup. In addition, if the point in time is before the start time of the most recent transaction log (incremental) backup, then a tail-log backup is not necessary. However, a tail-log backup is required if the point in time is after the most recent transaction log backup.

When you specify the point in time for restore, do not specify the start time of the selected transaction log backup if it is not the last backup in the backup sequence. Otherwise, the restore fails, and a tail-log backup does not occur even if you select the **Tail-log backup** checkbox.

SQL Server recovery operations

You can control the recovery operation that occurs after the restore by using restore options in the Avamar Plug-in for SQL Server.

Table 2. Recovery operation options

Recovery operation	Description
`RECOVERY`	The database is fully recovered and online after the restore. This operation is the default setting.
`NORECOVERY`	The database remains in a restoring state after the restore. This feature enables you to perform additional manual restore tasks, such as applying additional SQL transaction log files.
`STANDBY`	The database is in standby (read-only) mode after the restore. This mode enables you to bring up a database for read-only access between transaction log restores. You may need to use this option with either warm standby server situations or special recovery situations in which it is useful to inspect the database between log restores. This option also creates a file with recovery changes. You can use the file to revert the recovery changes, if required.

You can specify the recovery operation in the following restore scenarios:

- You are restoring an instance, database, filegroup, or file to its original location.
- You are restoring a database to the original instance but with a new database name.
- You are restoring an instance, database, filegroup, or file to a different instance on either the original server or a different server.
- You are restoring one or more system databases. Keep in mind the following points, however:
  - If you are restoring the master or model database, then you must select the `RECOVERY` option. Do not use either the `NORECOVERY` or `STANDBY` options.
  - If you are restoring the msdb database, then you can select any of the recovery operation options. However, if you select `NORECOVERY` or `STANDBY`, then all databases become inaccessible until the restore of the msdb database is complete.
  - If you are restoring the system databases, then you must select the `RECOVERY` option. Do not use either the `NORECOVERY` or `STANDBY` options.

Table level recovery

You can restore individual tables from a backup of an SQL database by performing a table level recovery.

Table level recovery (TLR) from an SQL database uses the Avamar Plug-in for SQL TLR with ItemPoint for Microsoft SQL Server, which is installed during installation of the Avamar Plug-in for SQL TLR. Single-stream as well as multiple streams are supported.

During a recovery operation when the Avamar Plug-in for SQL TLR is selected, the selected database backup is mounted as a virtual mounted drive. Once the database backup is mounted, ItemPoint for Microsoft SQL Server is used to perform the table level recovery.

Disaster recovery

To ensure sufficient preparation for disaster recovery of a SQL Server environment, you must perform ongoing backups of the Windows server and all system and user databases. Disaster Recovery on page 96 provides high-level procedures for preparing for and performing disaster recovery.

2

Installation

Topics:

- Preparing to install the SQL Server plug-in
- Installing the Avamar Plug-in for SQL Server
- Installing the Avamar client software
- Upgrading the Avamar client software
- Uninstalling the Avamar client software

Preparing to install the SQL Server plug-in

Review the system requirements for the Avamar Plug-in for SQL Server, and ensure that the environment meets the requirements before you install Avamar client software. The Avamar Config Checker for Microsoft Windows provides an additional automated verification of the environment. You also must download the Avamar Client for Windows and the Avamar Plug-in for SQL Server installation packages from the Avamar server.

System requirements

The environment must meet client compatibility requirements before you install Avamar client software.

See the *E-lab Navigator* at https://elabnavigator.emc.com/eln/modernHomeDataProtection for client compatibility requirements and supported operating systems and application versions.

The Avamar file system client and the plug-ins that you install on the host must have the same version number.

Installation of SQL Server 2008 or SQL Server 2008 R2 in a Windows Server 2012 cluster requires SQL Server 2008 SP3 or later, or SQL Server 2008 R2 SP1.

On Windows Server 2012, the Avamar Plug-in for SQL Server does not support backup and restore of SQL Server data on Server Message Block (SMB) 3.0 file shares. However, on Windows Server 2012 R2, the SQL Server plug-in supports storage of SQL Server databases and log files in an active/passive cluster on shared virtual hard disk (VHDX) files on an SMB 3.0 file share.

The Avamar Plug-in for SQL Server does not support backup and restore of the Windows Internal Database, which is also referred to as SQL Server Embedded Edition.

The environment must also meet other requirements, including hardware requirements, security requirements, and SQL Server feature and configuration requirements.

Hardware requirements

The following table lists the hardware requirements for the Avamar Plug-in for SQL Server.

Table 3. Minimum and recommended hardware requirements

Requirement	Minimum and recommended
RAM	512 MB (2 GB recommended).
Hard drive space	2 GB permanent hard drive space for software installation. The Avamar Plug-in for SQL Server software also requires an additional 12 MB of permanent hard drive space for each 64 MB of physical RAM. This space is necessary for local cache files.

Table 3. Minimum and recommended hardware requirements (continued)

Requirement	Minimum and recommended
Network interface	10BaseT minimum; 100BaseT or higher recommended, configured with the latest drivers for the environment.

Security requirements

When Avamar connects to SQL Server for backup or restore, you can use either Windows (NT) authentication or SQL Server authentication.

With Windows authentication, Avamar connects to SQL Server by using the Windows system service account (NT AUTHORITY\SYSTEM), which has privileges in SQL Server. The account must have the sysadmin server-level role.

NOTE: To use the Windows authentication method, select **NT authentication** in the plug-in options for the Avamar Plug-in for SQL Server.

With SQL Server authentication, Avamar connects to SQL Server by using a SQL Server login account. The account must have the sysadmin server-level role. You must select the mixed authentication mode for the Database Engine when you configure SQL Server.

If you do not specify an authentication method, then the SQL Server plug-in uses NT authentication and logs in with the Windows system service account. This account has the public and sysadmin server-level roles in SQL Server by default in SQL Server 2008 and 2008 R2. In SQL Server 2012, 2014, and 2016, you must add the account to the SQL Server administrators group.

Adding sysadmin server-level role in SQL Server Management Studio

Steps

1. In SQL Server Management Studio, expand the **Security** node and then the **Logins** node for the instance in the Object Explorer.
2. Right-click the user to whom you want to assign the **sysadmin** server role and select **Properties**.
   The **Login Properties** dialog box appears.
   NOTE: For Windows authentication, select the **NT AUTHORITY\SYSTEM** user account.
3. Select the **Server Roles** page from the list in the left pane.
4. In the right pane, select the checkbox next to the **sysadmin** server role.
5. Click **OK**.

SQL Server requirements

The SQL Server installation and environment must meet the requirements for the Avamar Plug-in for SQL Server.

SQL Server feature installation requirements

The Avamar Plug-in for SQL Server requires installation of SQL Server Management Objects (SMO). To install SMO, install the Client Tools SDK when you install SQL Server. Alternatively, install SMO from the SQL Server feature pack.

You also must install Management Tools and the SQL Client Connectivity SDK when you install SQL Server.

Database name requirements

Use only valid file name characters in database names. Avoid any of the following characters, which are known to interfere with proper operation of the SQL Server plug-in: asterisk (*), forward slash (/), backward slash (\), colon (:), semicolon (;), question mark (?), right angle bracket (>), left angle bracket (<), vertical bar (|), or number (#).

Do not end the database name with a period (.). If you end a database name with a period and then you restore the database, you receive an access violation when you try to open the folder for the restored database.

Transaction log marking requirements

The Avamar Plug-in for SQL Server cannot successfully back up and restore databases in which database transaction log marks contain leading or trailing white space.

System database requirements

The SQL Server plug-in can back up both user databases and system databases such as the master, msdb, and model databases.

The SQL Server Resource database is a read-only database that contains copies of all system objects that ship with SQL Server. The Avamar Plug-in for SQL Server cannot back up the Resource database, `mssqlsystemresource.mdf`, or its log file, `mssqlsystemresource.ldf`, because SQL Server cannot back up the Resource database. To back up Resource database files, use the Avamar Client for Windows.

The tempdb database is a temporary system database that is re-created every time you restart a SQL Server instance. The SQL Server plug-in cannot back up the tempdb system database because SQL Server does not support backup of the tempdb database.

The Avamar Plug-in for SQL Server can back up and restore system databases in a SQL Server replication environment, including the publication, distribution, and subscription databases. The SQL Server plug-in can automatically restore the replication system databases in the correct order and manage SQL Server services. The "Back Up and Restore Replicated Databases" topic in the SQL Server documentation on the MSDN website provides details on backup strategies and the steps to manually restore databases in different types of replication environments.

Alias requirements

A SQL Server alias is a name that you configure on the client computer that points to a server, instance, or database on either the local server or on a different computer. You can use an alias to connect with a certain network protocol, such as TCP/IP, Named Pipes, or Shared Memory.

An alias can improve the convenience and speed of connecting to SQL Server. However, improper use of an alias can result in connectivity issues that are difficult to isolate and troubleshoot.

If you plan to use a SQL Server alias, review the requirements and configuration steps on the Microsoft website for setting up an alias. Also, review best practices for using and troubleshooting aliases. These best practices are available in the blogs and Support knowledgebase articles on the Microsoft website.

AlwaysOn availability group requirements

In a SQL Server AlwaysOn environment, you can back up databases on either the primary or secondary replica for an availability group. There must be an availability group listener for each availability group. The following sections list additional requirements in an AlwaysOn environment.

Requirements for backups on a secondary replica

If you perform the backup on a secondary replica, then the replica must meet the following requirements:

- The replica must be in a synchronized state.
- The replica role setting in SQL Server must allow either read-only connections or all connections:
  - When you create a replica in SQL Server Management Studio, select either **Read-intent only** or **Yes** from the **Readable secondary** list on the **Replicas** tab in the **Add Replica Wizard**.
  - When you edit the configuration for an availability group in SQL Server Management Studio, select either **Read-intent only** or **Yes** from the **Readable secondary** list on the **Availability Replica Properties** dialog box.
  - Specify either `READ_ONLY` or `ALL` for the `SECONDARY_ROLE` option when you issue the `CREATE AVAILABILITY GROUP` or `ALTER AVAILABILITY GROUP` Transact-SQL command.
- If there are multiple secondary replicas, then set the priority for which secondary replica to use for the backup:
  - When you create a replica in SQL Server Management Studio, specify a value between 1 and 100 in the **Backup Priority** field in the **Add Replica Wizard**.
  - When you edit the configuration for an availability group in SQL Server Management Studio, specify a value between 1 and 100 in the **Backup Priority** field on the **Availability Replica Properties** dialog box.

- Specify a numeric value for the `BACKUP_PRIORITY` option when you issue the `CREATE AVAILABILITY GROUP` or `ALTER AVAILABILITY GROUP` Transact-SQL command.

A value of 1 indicates the lowest priority, and a value of 100 indicates the highest priority.

System databases in an availability group

SQL Server does not support system databases in an availability group. To back up system databases with Avamar, you must perform a separate backup. During the backup, select the server name of the physical node as the client to back up, and then select the system databases.

Requirements for the var directory

The var directory for the SQL AlwaysOn Cluster Plug-in must meet the following requirements:

- Must have greater than 2 GB of available space.
- For a shared location:
  - The full path to the shared location must be available on all nodes in the cluster where the Availability group is selected.
  - During registration, the shared \var folder must be specified as a UNC path.
  - Mounted shared locations which are available only to the primary replica are not supported.
  - The account running the avagent service must have modify permission level (including read/write access) to the shared var folder.

Hybrid IT environments with Microsoft Azure and AlwaysOn availability groups

In a hybrid IT environment with Microsoft Azure and AlwaysOn availability groups, ensure that the backup occurs on an on-premise replica. Avamar cannot back up databases on a replica on a Microsoft Azure virtual machine.

AlwaysOn availability groups and FCIs

The Avamar Plug-in for SQL Server does not support backups of AlwaysOn availability groups on an availability replica that the SQL Server Failover Cluster Instance (FCI) hosts. To back up databases in such an environment, you must install the Avamar client software on the node with the primary replica as a stand-alone client, and then perform backups on only that node.

Database mirroring requirements

You can use the Avamar Plug-in for SQL Server to back up SQL Server databases that are mirrored.

The following conditions apply when you use the SQL Server plug-in in an environment with database mirroring:

- The SQL Server version must be 2008 or greater.
- Perform backups of only the principal database, not the mirrors.
- Use only the Avamar Plug-in for SQL Server to perform backups. If you use other backup products in addition to the Avamar Plug-in for SQL Server, then log chain breaks may occur.
- Backup and database administrators must use extra care. Some mirror operations cause log chain breaks that cannot be detected.
- When database mirroring is established, either initially or as the result of failover and failback, then you must manually perform a new full backup. Otherwise, incremental and differential backups that occur after the establishment of database mirroring are not valid for restore. After you perform the full backup in this case, you cannot perform point-in-time recoveries to a point in time before the full backup.
- To restore a database, you must break the SQL mirror.

Transparent Data Encryption requirements

When you protect Avamar backups of SQL Server 2008 or later databases with Transparent Data Encryption (TDE), the backups intentionally do not include the Database Encryption Key (DEK).

Including the DEK defeats one of the primary reasons for using TDE, which is to encrypt exported data such as backups.

When you perform an Avamar backup of TDE-protected databases, manually back up the DEK. The article “Understanding Transparent Data Encryption (TDE),” available on the Microsoft TechNet website, provides additional details.

FILESTREAM requirements

The SQL Server FILESTREAM feature enables you to configure a database to store binary large object (BLOB) data as files on the file system instead of in the database. The Avamar Plug-in for SQL Server supports backups of SQL Server databases with FILESTREAM data for SQL Server 2008 and later.

Backing up and restoring a database with FILESTREAM data by using either Avamar Administrator or the `avsql` command line interface (CLI) are the same as backing up and restoring other databases.

Known issues with FILESTREAM support

You might encounter the following issues when you back up or restore a SQL Server database with FILESTREAM data.

FILESTREAM data restores to the database folder during redirected restore

FILESTREAM data restores to the folder with the database files, even if the FILESTREAM data was originally in a different folder than the database files, when the following conditions are true:

- You are restoring the database to a different location than the original location at the time of the backup.
- You specify the new location by using the **Alternate database location** box on the **Restore Command Line Options** dialog box.

To avoid this issue, specify the restore location for both the database files and the FILESTREAM data in the **Set Destination** dialog box.

Restore fails when the folder with FILESTREAM data is open

Restore of a database with FILESTREAM data fails when the folder with FILESTREAM data is open in a program such as Windows Explorer.

To avoid this issue, close the program that is accessing the folder with the FILESTREAM before you perform the restore. However, if the FILESTREAM folder is open and the restore fails, then perform the restore again and select the WITH REPLACE option to force the restore and bring the database online.

FILESTREAM datafiles appear as Rows Data in Avamar Administrator

FILESTREAM datafiles appear with a description of **Rows Data** on the **Restore** tab of the **Backup, Restore and Manage** window in Avamar Administrator. The description in Avamar Administrator is different than the **Filestream data** description that appears in SQL Server Management Studio.

SQL Server Analysis Service (SSAS) requirements

The Avamar Plug-in for SQL Server cannot back up SSAS databases. However, you can back up the SSAS databases by using SQL Server Management Studio (SSMS) or the XMLA Query Editor. Then use the Avamar Client for Windows to back up the flat file that results from the SSMS or XMLA query backup.

The Microsoft website provides instructions on using SSMS or an XMLA query to back up SSAS databases.

Considerations for the number of databases

The Avamar Management Console Server (MCS) may not be able to display all databases when you browse for data to back up or restore on a SQL Server installation with many databases.

When this issue occurs, a Browse Timeout message appears and enables you to either set a new time limit in seconds or to view partial results. If you view partial results, no entries appear. The Avamar Plug-in for SQL Server does not support this option.

Preparing a stand-alone server

You must perform SQL Server installation and configuration steps before you install Avamar client software on a stand-alone server.

Steps

1. Install and configure Microsoft SQL Server.

2. Install Microsoft .NET Framework 4. This software is required to install the Avamar Plug-in for SQL Server. Search the Microsoft Download Center for “Microsoft .NET Framework 4" to find downloads and additional information.
3. If you are using Microsoft SQL Server 2012, 2014, or 2016, then add the Windows system service account to the SQL Server administrators group. Adding sysadmin server-level role in SQL Server Management Studio on page 33 provides instructions.

Preparing a cluster

You must perform SQL Server installation and configuration steps before you install Avamar client software in a cluster.

Steps

1. Install and prepare the cluster.
2. If you are installing SQL Server 2008 or SQL Server 2008 R2 in a Windows Server 2012 cluster, enable the COM-based `MsClust.dll` library on each node.

   If you do not enable this library on the cluster node, then the SQL Server setup on the node fails with an error that cluster service verification failed.

   To enable the library, enable the Failover Cluster Automation Server feature by using one of the following methods on each node in the cluster:

   - In Server Manager, expand **Remote Server Administration Tools** > **Feature Administration Tools** > **Failover Clustering Tools**, and then select the **Failover Cluster Automation Server** checkbox.
   - Run the following Windows PowerShell cmdlet from an elevated command prompt:

     ```
     add-windowsfeature RSAT-Clustering-AutomationServer
     ```

3. If you are installing SQL Server 2008 or SQL Server 2008 R2 in a Windows Server 2012 cluster, install SQL Server 2008 SP3 or later, or SQL Server 2008 R2 SP1 or later.

   Otherwise, a “Windows Server 2003 FILESTREAM Hotfix Check” error appears when you install and configure SQL Server in the cluster.

   To ensure that you are installing the service pack version of SQL Server, you may need to perform one of the following steps:

   - Download the correct service pack package, and run the service pack setup on each of the cluster nodes to pre-patch the nodes with setup binaries.

     The setup process closes automatically after you install setup support files. You must reopen the setup wizard to perform the SQL Server installation.

     NOTE: If the SQL Support Files from SQL Server 2008 or SQL Server 2008 R2 are currently installed on the node, uninstall SQL Server before you run the service pack setup to install latest SQL Support Files.
   - Create a slipstream package that includes the service pack, and use that to install the service pack version of SQL Server on each node. Microsoft knowledgebase article 955392, “How to update or slipstream an installation of SQL Server 2008,” which is available on the Microsoft website, provides instructions on creating a slipstream package.
4. Install and configure Microsoft SQL Server on each node.
5. Install Microsoft .NET Framework 4 on each node. This software is required to install the Avamar Plug-in for SQL Server. Search the Microsoft Download Center for “Microsoft .NET Framework 4" to find downloads and additional information.

   Microsoft .NET Framework 4 is pre-installed on Windows Server 2012.
6. If you are using Microsoft SQL Server 2012, 2014, or 2016, then add the Windows system service account to the SQL Server administrators group. Adding sysadmin server-level role in SQL Server Management Studio on page 33 provides instructions.

Downloading the software

Download the installation packages for the Avamar Client for Windows, the Avamar Plug-in for SQL Server, and the Avamar Config Checker from the Avamar server. Then save the installation packages to a temporary folder.

Steps

1. Log in to the computer that is running SQL Server as an administrator.

2. Log in to the *application_name* server as an administrator.
3. Open a web browser and type the following URL:
   `https://Avamar_server/dtlt/home.html`
   where *Avamar_server* is the DNS name or IP address of the Avamar server.
   The **Avamar Web Restore** page appears.
4. Click **Downloads.**
   The **Downloads** list appears.
5. Click **+** next to the operating system headings until the applicable software installation packages appear.
6. Click the Avamar Client for Windows installation package, `AvamarClient-windows-x86_64-version.msi`
   where *version* is the Avamar client version.
7. Click the link for the SQL Server plug-in installation package, `AvamarSQL-windows-x86_64-version.exe`.
   where *version* is the SQL Server plug-in version.

   NOTE: Ensure that you select the correct installation package for the operating system.

8. Save the SQL Server plug-in installation package to a temporary folder.
9. Click the Avamar Config Checker installation package, `Avamar_ConfigChecker_win_x64.zip`.

Verifying the environment

Use the Avamar Config Checker for Microsoft Windows to verify that you correctly configured the Windows environment for Avamar backup and recovery. The Config Checker checks the configuration for problems that can lead to installation, backup, or recovery failures. These failures can affect the operating system for the application host or the application itself.

About this task

You can run the Avamar Config Checker either before or after you install Avamar software on the client computer.

The Config Checker supports only English language operating systems.

Steps

1. Unzip the Avamar Config Checker installation package. To install the software, run the setup program.
2. Start the Config Checker:
   a. Open the **Start** screen.
   b. Select **Avamar Config Checker**.
3. Click **Next** on the welcome page.
   The **Avamar Application and User Settings** page appears.
4. Select the version number from the **Avamar version** list.
5. In the application list, select the checkbox next to the applications on the client computer. Specify credentials, if required.
6. Click **Next**.
7. Review the summary information.
8. Click **Run Tests**.
   When the verification completes, the **Config Check Results** window appears.
9. Save or open the results in HTML format.

   NOTE: The Config Checker does not automatically save the results. If you do not save the results, you must rerun the Config Checker to view them.

10. Click **Finish** to exit the Config Checker.
11. Review the HTML result file, and correct all failed checks.
12. Rerun the Config Checker to ensure that all the checks are successful.
    The *Avamar Config Checker for Microsoft Windows Technical Note*, available on Online Support at https://support.EMC.com, provides troubleshooting information and details about the results that appear for each application.

Installing the Avamar Plug-in for SQL Server

The SQL Server plug-in enables you to back up and restore SQL Server data.

About this task

A setup wizard leads you through the steps to install the plug-in. If UAC is enabled on the client computer, then you must start the setup wizard by using administrator privileges. Otherwise, the software does not install correctly. This procedure provides one method to bypass UAC. The Microsoft documentation provides other methods and additional information.

The plug-in automatically installs in the same folder as the Avamar Client for Windows.

Steps

1. Log in to the computer that is running SQL Server as an administrator.
2. Go to the temporary folder that contains the Avamar installation files that you downloaded.
3. Start the setup wizard:
   - If UAC is disabled, double-click the installation file (.exe) to open it.
   - If UAC is enabled, open a command prompt as an administrator, change directory to the location of the installation package, and then run the following command:

     ```
     msiexec /i AvamarSQL-windows-x86_64-<version>.msi
     ```

     where *version* is the Avamar plug-in version.

     NOTE: Specify the `.msi` file extension when you use CLI.

   The installation wizard opens at the welcome page.
4. (Optional) To enable support for table lever recovery (TLR) select **TLR using ItemPoint for SQL**. Click **Next**.

   This step installs both the Avamar Plug-in for SQL TLR and the ItemPoint software.

   NOTE: Installing the Avamar Plug-in for SQL TLR will also automatically install ItemPoint for Microsoft SQL Server. TLR using Itempoint for SQL is selected by default in Avamar 18.1. You can unselect the option if you do not wish to install TLR using Itempoint for SQL.

   The **Ready to Install Avamar Backup Plug-in for SQL Server** page appears.
5. Click **Install**.

   NOTE: The SQL Server plug-in automatically installs to the same folder as the Avamar Client for Windows.
6. When the installation completes, click **Finish**.

Installing the Avamar client software

To install the Avamar client software, install the Avamar Client for Windows and the plug-in on the client computer. Then register the client with the Avamar server. In a cluster environment, perform these steps on each node, and then configure the cluster client.

Installation road maps

The steps to install the Avamar software on a client computer depend on whether the environment is a stand-alone server or a cluster.

Stand-alone installation road map

To install the Avamar client software on a stand-alone server, install the Windows client and the plug-in on the server, and then register the client with the Avamar server.

Steps

1. Install the Avamar Client for Windows on the computer that is running SQL Server.
2. Install the Avamar Plug-in for SQL Server on the computer that is running SQL Server.
3. Register the computer that is running SQL Server as a client with the Avamar server.

Cluster installation road map

To install the Avamar client software in a cluster, install the Windows client and the plug-in on each node, register each node, and then configure the Avamar cluster client.

Steps

1. Install the Avamar Client for Windows in the same folder on each node in the cluster.
2. Install the Avamar Plug-in for SQL Server in the same folder on each node in the cluster.
3. Register each node in the cluster with the Avamar server.
4. Use the Cluster Configuration Tool to configure the Avamar cluster client.

   In a failover cluster, run the tool on the active node in the cluster. In an AlwaysOn availability group environment, run the tool on the node that is hosting the primary replica.

Installing the Avamar Client for Windows

The Avamar Client for Windows includes the Avamar agent and the Avamar Plug-in for Windows File System. The Avamar agent is required for backups and restores with application plug-ins. You can use the Windows File System plug-in to back up operating system and application binary files, which are required for disaster recovery.

About this task

A setup wizard leads you through the steps to install the Windows client. If the User Account Control (UAC) feature is enabled on the client computer, then you must start the setup wizard by using administrator privileges. Otherwise, the software does not install correctly. This procedure provides one method to bypass UAC. The Microsoft documentation provides other methods and additional information.

Steps

1. Log in to the computer that is running SQL Server as an administrator.
2. Go to the temporary folder that contains the Avamar installation files that you downloaded.
3. Start the setup wizard:
   - If UAC is disabled, double-click the installation file to open it.

- If UAC is enabled, open a command prompt as an administrator, change directory to the location of the installation package, and then type one of the following command:

  `msiexec /i AvamarClient-windows-x86_64-`*`version`*`.msi`

  where *version* is the Avamar client version.

The welcome page appears.

4. Click **Next**.
   The **End-User License Agreement** page appears.
5. Review the license agreement.
6. Select the checkbox to accept the terms in the license agreement, and click **Next.**
   The **Custom Setup** page appears.
7. Ensure that **Avamar Client User Interface** is selected for installation.
8. (Optional) To specify a folder for the Avamar client installation, click **Browse** and select a location.
9. Click **Next** on the **Custom Setup** page.
   The **Ready to Install Avamar for Windows** page appears.
10. Click **Install.**
11. When the installation completes, click **Finish.**
12. In a cluster environment, repeat these steps to install the Windows client on each node.

Installing the Avamar Plug-in for SQL Server

The SQL Server plug-in enables you to back up and restore SQL Server data.

About this task

A setup wizard leads you through the steps to install the plug-in. If UAC is enabled on the client computer, then you must start the setup wizard by using administrator privileges. Otherwise, the software does not install correctly. This procedure provides one method to bypass UAC. The Microsoft documentation provides other methods and additional information.

The plug-in automatically installs in the same folder as the Avamar Client for Windows.

Steps

1. Log in to the computer that is running SQL Server as an administrator.
2. Go to the temporary folder that contains the Avamar installation files that you downloaded.
3. Start the setup wizard:
   - If UAC is disabled, double-click the installation file (.exe) to open it.
   - If UAC is enabled, open a command prompt as an administrator, change directory to the location of the installation package, and then run the following command:

     `msiexec /i AvamarSQL-windows-x86_64-<version>.msi`

     where *version* is the Avamar plug-in version.

     NOTE: Specify the `.msi` file extension when you use CLI.

   The installation wizard opens at the welcome page.
4. (Optional) To enable support for table lever recovery (TLR) select **TLR using ItemPoint for SQL**. Click **Next**.
   This step installs both the Avamar Plug-in for SQL TLR and the ItemPoint software.

   NOTE: Installing the Avamar Plug-in for SQL TLR will also automatically install ItemPoint for Microsoft SQL Server. TLR using Itempoint for SQL is selected by default in Avamar 18.1. You can unselect the option if you do not wish to install TLR using Itempoint for SQL.

   The **Ready to Install Avamar Backup Plug-in for SQL Server** page appears.
5. Click **Install**.

   NOTE: The SQL Server plug-in automatically installs to the same folder as the Avamar Client for Windows.

6. When the installation completes, click **Finish**.

Silent installation of Itempoint for SQL

Steps

1. Open **Command Prompt** and run the commands in the next step.
2. You can select Itempoint to be installed or not installed depending on the parameter that is passed to the Avamar SQL application during silent installation.
   - To install Itempoint for SQL, run the following command:

     ```
     msiexec /i AvamarSQL-windows-x86_64-<version>.msi /q
     ```

   - If you do not want to install Itempoint for SQL, run the following command:

     ```
     msiexec /i AvamarSQL-windows-x86_64-<version>.msi /q /ComponentArgs
     "EMCItemPoint":"noipinstall"
     ```

   NOTE: Specify the `.msi` file extension when you use CLI.

Registering the client

Before you can back up or restore data on a server, you must register the server as a client with the Avamar server. Specify the name of the Avamar server, the Avamar domain for the client, and the port on the Avamar server for client/server communication. Avamar domains enable you to group clients and limit backup, restore, and monitoring privileges for the group to certain accounts.

About this task

In a cluster environment, register all nodes in the cluster as Avamar clients.

Steps

1. Log in to the server.
2. Right-click the Avamar client system tray icon, and select **Manage** > **Activate Client**.
   The **Activate Client Setup** dialog box appears.
3. In the **Administrator Server Address** box, type the DNS name for the Avamar server.
4. In the **Administrator Server Port** box, specify the port on the Avamar server for client/server communication.
5. In the **Client Domain** box, type the name of the Avamar domain for the client.

   The default domain is **`clients`**. Do not use a slash (/) as the first character when you type a subdomain. If you use a slash, an error occurs and you cannot register the client.
6. Click **Activate.**
   A confirmation message appears.
7. Click **OK**.

Configuring the cluster client in a failover cluster

The Avamar cluster client in a failover cluster enables you to back up and restore SQL Server data on shared storage in the cluster, regardless of which node is managing the data at the time of the backup or restore. The Avamar Cluster Configuration Tool provides the steps to configure the Avamar cluster client for the SQL Server plug-in in a failover cluster.

Steps

1. Log in to the active node in the cluster as a domain administrator. The account must also be a member of the local Administrators group on each cluster node.
2. Start the Cluster Configuration Tool by opening the **Start** screen and select **Cluster Configuration Tool**.
   The welcome page appears.
3. Click **Next**.
   The **Plug-Ins** page appears.
4. Select **SQL** and click **Next**.
   The **Cluster Nodes** page appears with a list of nodes and their status.

5. Ensure that the environment meets the following requirements:
   - The status for each SQL Server node is Up.
   - The installation status of the Windows client software for each node is Installed.
   - The installation status of the SQL Server plug-in on each node is Installed.
6. Click **Next**.
   The **Operations** page appears.
7. Select **Configure a new cluster client for all nodes**, and then click **Next**.
   The **Prerequisites** page appears. A check mark next to a prerequisite indicates that the environment meets the prerequisite.
8. Ensure that the environment meets all prerequisites on the **Prerequisites** page.

   If the environment does not meet a prerequisite, then exit the wizard, resolve the issue, and restart the wizard.
9. Select the IP version that the environment uses, and then click **Next**.
   The **SQL Settings** page appears.
10. Select the cluster role for the cluster client from the **Cluster role for cluster client** list.
11. Specify whether to automatically bring the cluster client online and activate it with the Avamar server by selecting or clearing the **Bring the cluster client online and activate it with the Avamar server** check box.
12. Click **Next**.
    The **Server Settings** page appears.
13. Specify the settings for the Avamar server:
    a. Type either the DNS name of the Avamar server in the **Name** box or the IP address in the **IPv4/IPv6** address box.
    b. Type the name of the Avamar domain in the Avamar client domain for the cluster client box.

       To specify a domain at the root level, type ***/domain***, where *domain* is the domain name. To specify a subdomain, type ***/domain/subdomain***, where *domain* is the domain name and *subdomain* is the subdomain name.
    c. Type the data port for Avamar client/server communication in the **Port number** box.

       NOTE: Port 28001 is the default port that the Avamar client uses to communicate with the Avamar server.
14. Click **Next**.
    The **Client Settings** page appears.
15. Type the name of the shared network folder or volume in the **Cluster client's var directory** box, or click **Browse** to select a folder.

    The `var` directory should reside on the cluster disk, and is used to store the cluster client configuration and log files. All nodes in the cluster must have write access to directory.

    NOTE: Select Cluster Shared Volumes (CSV) that the cluster owns instead of a remote pathname on the network.
16. Type the name of the shared network folder or volume in the **Cluster client's SYSDIR directory** box, or click **Browse** to select a folder.

    The SYSDIR directory must have more than 2 GB of available space.
17. Review the configuration settings, and then click **Configure**.
    The **Progress** page provides the status of the configuration. When the configuration is complete, the **Results** page appears.
18. Click **Close**.

Configuring the cluster client for an AlwaysOn availability group

The Avamar cluster client for an AlwaysOn availability group enables you to back up and restore SQL Server databases in an availability group. The Avamar Cluster Configuration Tool walks you through the steps to configure the Avamar cluster client for the SQL Server plug-in in an AlwaysOn availability group.

Steps

1. Log in to the node that is hosting the primary replica as a domain administrator. The account must also be a member of the local Administrators group on each cluster node.
2. Start the Cluster Configuration Tool by opening the **Start** screen and select **Cluster Configuration Tool**.
   The welcome page appears.
3. Click **Next**.
   The **Plug-Ins** page appears.
4. Select **SQL AlwaysOn** and click **Next**.

The **Cluster Nodes** page appears with a list of nodes and their status.

5. Ensure that the environment meets the following requirements:
   - The status for each SQL Server node is Up.
   - The installation status of the Windows client software for each node is Installed.
   - The installation status of the SQL Server plug-in on each node is Installed.
6. Click **Next**.
   The **Operations** page appears.
7. Select **Configure a new cluster client for all nodes**, and then click **Next**.
   The **Prerequisites** page appears. A check mark next to a prerequisite indicates that the environment meets the prerequisite.
8. Ensure that the environment meets all prerequisites on the **Prerequisites** page.

   If the environment does not meet a prerequisite, then exit the wizard, resolve the issue, and restart the wizard.
9. Select the IP version that the environment uses, and then click **Next**.
   The **SQL AlwaysOn Settings** page appears.
10. Select the cluster role for the cluster client from the **Cluster role for cluster client** list.
    The name of the cluster client appears in the **Cluster client name** box.

    NOTE: Configure an availability group listener for each availability group. Do not configure a cluster client for an availability group that does not have a listener.
11. Specify whether to automatically bring the cluster client online and activate it with the Avamar server by selecting or clearing the **Bring the cluster client online and activate it with the Avamar server** check box.
12. Click **Next**.
    The **Server Settings** page appears.
13. Specify the settings for the Avamar server:
    a. Type either the DNS name of the Avamar server in the **Name** box or the IP address in the **IPv4/IPv6** address box.
    b. Type the name of the Avamar domain in the Avamar client domain for the cluster client box.

       To specify a domain at the root level, type ***/domain***, where *domain* is the domain name. To specify a subdomain, type ***/domain/subdomain***, where *domain* is the domain name and *subdomain* is the subdomain name.
    c. Type the data port for Avamar client/server communication in the **Port number** box.

       NOTE: Port 28001 is the default port that the Avamar client uses to communicate with the Avamar server.
14. Click **Next**.
    The **Client Settings** page appears.
15. Type the name of the shared network folder or volume in the **Cluster client's var directory** box, or click **Browse** to select a folder.

    The `var` directory should reside on the cluster disk, and is used to store the cluster client configuration and log files. All nodes in the cluster must have write access to directory.

    NOTE: Select Cluster Shared Volumes (CSV) that the cluster owns instead of a remote pathname on the network.
16. Type the name of the shared network folder or volume in the **Cluster client's SYSDIR directory** box, or click **Browse** to select a folder.

    The SYSDIR directory must have more than 2 GB of available space.
17. Click **Next**.
    The **Summary** page appears.
18. Review the configuration settings, and then click **Configure**.
    The **Progress** page provides the status of the configuration. When the configuration is complete, the **Results** page appears.
19. Click **Close**.

Configuring the cluster client for an AlwaysOn availability group on Amazon Web Services and Azure

You can configure the Avamar cluster client for SQL Server in an AlwaysOn availability group, which resides on Amazon Web Services or Microsoft Azure. The Avamar cluster client for an AlwaysOn availability group enables you to back up and restore SQL Server databases in an availability group. The Avamar Cluster Configuration Tool walks you through the steps to configure the Avamar cluster client for the SQL Server plug-in in an AlwaysOn availability group.

Steps

1. Log in to the node that is hosting the primary replica as a domain administrator. The account must also be a member of the local Administrators group on each cluster node.
2. Start the Cluster Configuration Tool by opening the **Start** screen and select **Cluster Configuration Tool**.
   The welcome page appears.
3. Click **Next**.
   The **Plug-Ins** page appears.
4. Select **SQL AlwaysOn** and click **Next**.
   The **Cluster Nodes** page appears with a list of nodes and their status.
5. Ensure that the environment meets the following requirements:
   - The status for each SQL Server node is Up.
   - The installation status of the Windows client software for each node is Installed.
   - The installation status of the SQL Server plug-in on each node is Installed.
6. Click **Next**.
   The **Operations** page appears.
7. Select **Configure a new cluster client for all nodes**, and then click **Next**.
   The **Prerequisites** page appears. A check mark next to a prerequisite indicates that the environment meets the prerequisite.
8. Ensure that the environment meets all prerequisites on the **Prerequisites** page.

   If the environment does not meet a prerequisite, then exit the wizard, resolve the issue, and restart the wizard.
9. Select the IP version that the environment uses, and then click **Next**.
   The **SQL AlwaysOn Settings** page appears.
10. Select the cluster role for the cluster client from the **Cluster role for cluster client** list.
    The name of the cluster client appears in the **Cluster client name** box.

    NOTE: Configure an availability group listener for each availability group. Do not configure a cluster client for an availability group that does not have a listener.
11. Clear the **Bring the cluster client online and activate it with the Avamar server** checkbox.
12. Click **Next**.
    The **Server Settings** page appears.
13. Specify the settings for the Avamar server:
    a. Type either the DNS name of the Avamar server in the **Name** box or the IP address in the **IPv4/IPv6** address box.
    b. Type the name of the Avamar domain in the Avamar client domain for the cluster client box.

       To specify a domain at the root level, type ***/domain***, where *domain* is the domain name. To specify a subdomain, type ***/domain/subdomain***, where *domain* is the domain name and *subdomain* is the subdomain name.
    c. Type the data port for Avamar client/server communication in the **Port number** box.

       NOTE: Port 28001 is the default port that the Avamar client uses to communicate with the Avamar server.
14. Click **Next**.
    The **Client Settings** page appears.
15. Type the name of the shared network folder or volume in the **Cluster client's var directory** box, or click **Browse** to select a folder.

    The `var` directory should reside on the cluster disk, and is used to store the cluster client configuration and log files. All nodes in the cluster must have write access to directory.

    NOTE: Select Cluster Shared Volumes (CSV) that the cluster owns instead of a remote pathname on the network.
16. Type the name of the shared network folder or volume in the **Cluster client's SYSDIR directory** box, or click **Browse** to select a folder.

    The SYSDIR directory must have more than 2 GB of available space.
17. Click **Next**.
    The **Summary** page appears.
18. Review the configuration settings, and then click **Configure**.
    The **Progress** page provides the status of the configuration. When the configuration is complete, the **Results** page appears.
19. Click **Close**.
20. On the primary node, launch Microsoft Failover Cluster Manager.
21. Select the **Cluster instance** > **Roles** > **AAG group** under **Configuration**.

22. Under **Resources**, select **EMC Avamar Backup Agent resource for XXX**.
23. Right-click **Properties** and select **Startup parameters**.
24. Remove the flag `--netbind=`.
25. Edit the flag `--sysdir="C:\Program Files\avs\aag\etc"` (or the existing local folder).
26. Click **Save** and **Close**.
27. On all the SQL nodes, create the `SYSDIR` folder, if it does not exist.
28. On each SQL AlwaysOn availability group node, create the `avagent.cmd` file in the `var` (`C:\Program Files\avs\AAG\var`) folder with the `--netbind=`*<localhostIP>* option .
29. On the primary SQL node, bring the resource **EMC Avamar Backup Agent resource for XXX** online.
30. On all the SQL secondary nodes, copy the `cid.bin` file from the primary `var` folder (`C:\Program Files\avs\AAG\var`) to all the secondary nodes **var** folder.

Upgrading the Avamar client software

Upgrading steps depend on whether the installation is on a stand-alone server or in a cluster.

Upgrading on a stand-alone server

Steps

1. Uninstall the current version of the Avamar client and plug-in:
   a. Uninstall the ItemPoint software, if it has been installed.
   b. Uninstall the earlier version of the Avamar Plug-in for SQL Server.
   c. Uninstall the earlier version of the Avamar Client for Windows.
2. Install the new version of the Avamar Client for Windows.
3. Install the new version of the Avamar Plug-in for SQL Server.
4. Perform a full backup of all SQL Server data with the `Force Incremental After Full` option cleared, to ensure that you have a current full backup with the new version.

   Restore of data from a backup with an earlier release may fail, especially when you restore from a differential or incremental backup. The *E-LAB Navigator* at https://elabnavigator.emc.com/eln/modernHomeDataProtection provides details on supported versions and any version compatibility issues.

Upgrading in a cluster

When you upgrade Avamar client and plug-in software in a cluster, you must uninstall the current Avamar clients and plug-ins from each node, and then install the new version.

Prerequisites

- Ensure that the environment meets all system requirements for the new version.
- Ensure that no backups are in progress or scheduled to take place during the upgrade. Otherwise, Avamar client software files may be locked during the upgrade and you may need to restart the computer after the upgrade.

Steps

1. Uninstall the current version of the Avamar client and plug-in:
   a. Use the earlier version of the Cluster Configuration Tool to uninstall the Avamar cluster client.
   b. Uninstall the ItemPoint software, if it has been installed.
   c. Uninstall the earlier version of the Avamar Plug-in for SQL Server.
   d. Uninstall the earlier version of the Avamar Client for Windows.

   The plug-in guide for the earlier version provides instructions for each of these steps.
2. Install the new version of the Avamar client and plug-in:
   a. Install the Avamar Client for Windows in the same folder on each node in the cluster.
   b. Install the Avamar Plug-in for SQL Server in the same folder on each node in the cluster.

c. Register each node in the cluster with the Avamar server.
d. Configure the Avamar Cluster Client.

3. Perform a full backup of all SQL Server data with the `Force Incremental After Full` option cleared, to ensure that you have a current full backup with the new version.

   Restore of data from a backup with an earlier release may fail, especially when you restore from a differential or incremental backup. The *E-LAB Navigator* at https://elabnavigator.emc.com/eln/modernHomeDataProtection provides details on supported versions and any version compatibility issues.

Uninstalling the Avamar client software

When you uninstall Avamar plug-in and client software from a client computer, scheduled backups no longer occur for the client. You cannot restore backups to the client after you uninstall the software.

About this task

When you uninstall the Avamar client software, you can keep or delete the backups for the client:

- To keep the backups for the client so that you can restore the backups to a different client, retire the client by using Avamar Administrator.
- To delete the backups for the client, delete the client by using Avamar Administrator.

Retire or delete the client either before or after you uninstall the plug-in. The *Avamar Administration Guide* provides more information.

Uninstall road map

To uninstall Avamar client and plug-in software from a stand-alone server, uninstall the plug-in and Windows client by using standard Windows uninstall features. In a cluster, first uninstall the Avamar cluster client. Then uninstall the plug-in and Windows client on each node.

Steps

1. (Cluster only) Uninstall the Avamar cluster client.
2. Uninstall the Avamar Plug-in for SQL Server by using **Programs and Features**.
3. Uninstall the Avamar Client for Windows by using **Programs and Features**.
4. (Cluster only) Repeat step 2 and step 3 on each node.

Uninstalling the cluster client in a failover cluster

To uninstall the Avamar cluster client in a failover cluster, run the Cluster Configuration Tool on an active node in the cluster, and select the removal option.

Steps

1. Log in to the active node in the cluster as a domain administrator. The account must be a member of the local Administrators group on each cluster node.
2. Start the Cluster Configuration Tool by opening the **Start** screen and selecting **Cluster Configuration Tool**
   The welcome page appears.
3. Click **Next**.
   The **Plug-ins** page appears.
4. Select **SQL** and click **Next**.
   The **Cluster Nodes** page appears with a list of nodes and their status.
5. Ensure that the status of each SQL Server node is Up, and then click **Next**.
   The **Operations** page appears.
6. Select **Remove the cluster client from all nodes**, and then click **Next**.
   The **Prerequisites** page appears. A check mark next to a prerequisite indicates that the environment meets the prerequisite.
7. Ensure that the environment meets all prerequisites on the page, and click **Next**.
   The **Uninstall SQL** page appears.

8. Select the cluster role, service, or group that contains the cluster client from the **Cluster role/service/group for cluster client** list, and then click **Next**.
9. Select the cluster client from the **Cluster client name** list.
10. Select the shared volume for the cluster client from the **Shared volume for the cluster client** list.
11. Click **Next**.
    The **Summary** page appears.
12. Review the summary information, and then click **Uninstall**.
    The **Progress** page provides the status of the uninstall. When the uninstall is complete, the **Results** page appears.
13. Click **Close**.

Uninstalling the cluster client for an AlwaysOn availability group

To uninstall the Avamar cluster client from an AlwaysOn availability group, run the Cluster Configuration Tool on the primary replica, and select the removal option.

Steps

1. Log in to the cluster node with the primary replica as a domain administrator. The account must also be a member of the local Administrators group on each cluster node.
2. Start the Cluster Configuration Tool by opening the **Start** screen and selecting **Cluster Configuration Tool**
    The welcome page appears.
3. Click **Next**.
    The **Plug-ins** page appears.
4. Select **SQL AlwaysOn** and click **Next**.
    The **Cluster Nodes** page appears with a list of nodes and their status.
5. Ensure that the status of each SQL Server node is Up, and click **Next**.
    The **Operations** page appears.
6. Select **Remove the cluster client from all nodes**, and then click **Next**.
    The **Prerequisites** page appears. A check mark next to a prerequisite indicates that the environment meets the prerequisite.
7. Ensure that the environment meets all prerequisites on the page, and click **Next**.
    The **Uninstall Settings** page appears.
8. Select the cluster role or service that contains the cluster client from the **Cluster role/service for cluster client** list, and then click **Next**.
    The **Summary** page appears.
9. Review the summary information, and then click **Uninstall**.
    The **Progress** page provides the status of the uninstall. When the uninstall is complete, the **Results** page appears.
10. Click **Close**.

3

Backup

Topics:

- Perform an on-demand SQL backup by using AUI
- Configure the SQL server policy
- Scheduling backups using the AUI Policy wizard
- Monitoring backups
- Cancel backups
- Verifying backups
- Enforcement of backups to Data Domain

Perform an on-demand SQL backup by using AUI

An on-demand backup is a user-initiated backup of specific data that you select. Perform an on-demand backup for the first backup of the client immediately after you install the Avamar client software, and before system maintenance, software installations, or software upgrades.

Steps

1. In the AUI navigation pane on the left, click ≫, and then click **Asset Management**.
   The **Asset Management** window is displayed.
2. Select the domain that contains the host of the SQL server.
   A list of Avamar clients is displayed in the pane under the domains list.
3. In the list of clients, select the computer that is running the SQL Server.

   Keep the following points in mind when you select a client:

   - To back up databases in an AlwaysOn availability group, select the cluster client for the availability group listener.
   - To back up databases on shared storage in a failover cluster, select the cluster client for the virtual server.
   - You can only view clients in the domain for the login account. To view all clients, log in to the root domain.
4. Click the **Backup** tab.
   The Backup wizard is displayed. In the **Plugin** pane, a list of plug-ins on the client is displayed.
5. In the **Plugins** pane, perform the following steps:
   a. Select the *Windows SQL* plug-in.
   b. Browse to and select one or more databases for the backup.

   Table 4. Steps to backup data

Data to select for backup	Selection steps
All data on the client	In the **Plugin** pane, select **Windows SQL**.
Specific instances or databases on the client	To browse to the instances or databases: i. Expand the **Windows SQL** plug-in node. ii. Select the SQL Server instance that contains the databases to back up. A list of the databases for that instance is displayed in the right pane. iii. To back up all databases in the instance, select the instance.

   c. Click **NEXT**.

   The **Basic Configuration** window is displayed.
6. Select the backup retention policy settings:

- To automatically delete this backup from the Avamar server after a specific amount of time, select **Retention period**. Specify the number of days, weeks, months, or years for the retention period.
- To automatically delete this backup from the Avamar server on a specific calendar date, select **End date** and browse to that date on the calendar.
- To keep this backup for as long as this client remains active in the Avamar server, select **No end date**.

7. From the **Avamar encryption method** list, select the encryption method to use for data transfer between the client and the Avamar server during the backup.

   The encryption technology and bit strength for a client/server connection depends on several factors, including the client operating system and Avamar server version. The *Avamar Product Security Guide* provides additional information.

8. From the **Optionally select a proxy to perform backup** list, select the proxy.

   The default setting is **Automatic**, which enables the Avamar server to choose the best proxy for this operation.

9. Click **NEXT**.

   The **More Options** window is displayed.

10. Set the plug-in options.

    SQL plug-in backup options provides more information about the basic backup options.

11. (Optional) Toggle the **Show Advanced Options** switch to view advanced configuration options.

    SQL plug-in backup options provides more information about the advanced backup options.

12. Click **FINISH**.

Configure the SQL server policy

A policy contains all the information necessary to perform backups of groups of instances.

About this task

Policy objects contain three child objects:

- Schedule—A policy that controls the frequency and the start and end time each day for backups of clients. A schedule is a persistent and reusable Avamar policy that can be named and attached to multiple policies.
- Retention—A policy that defines how long the backup is stored in the backup appliance.
- Dataset—A policy that defines a set of files, directories, and file systems for each supported platform that are included or excluded in backups across a group of clients. A dataset is a persistent and reusable Avamar policy that can be named and attached to multiple policies.

From the **Overview** page, perform the following steps.

To access the **Source** page of the **Policy** wizard, in the AUI navigation pane on the left, click ≫, click **Advanced Policy** , and then click **ADD** to create a policy.

Steps

1. In the **Advanced Policy Type** field, select **SQL Server** .
2. To add a backup group, click +, and then select a backup group type from the list.
3. In the **Schedule** column, select a schedule for each backup group.

   To ensure that backups occur on an ongoing basis, scheduled backups run automatically. You can schedule backups to run daily, weekly, or monthly.

   Alternatively, you can add a schedule by selecting **Create** in the drop-down list. The **Create Schedule** wizard is displayed and opens to the **Basic Configuration** page.

4. In the **Dataset** column, select a dataset for each backup group.

   A dataset specifies the data to include in a scheduled backup and the options to use for the backup. Create at least one dataset for scheduled backups on a client or group of clients. To segregate client data, create multiple datasets.

   Alternatively, you can add a dataset by selecting **Create** in the drop-down list. The **Create DataSet** wizard is displayed.

5. In the **Retention** column, select a retention policy for each backup group.

Retention is the time setting to automatically delete backups on an Avamar server. Retention can be set to permanent for backups that should not be deleted from an Avamar server.

Alternatively, you can add a retention policy by selecting **Create** in the drop-down list. The **Add Retention Policy** wizard is displayed.

6. Click **NEXT**.

Scheduling backups using the AUI Policy wizard

Scheduled backups run automatically to ensure that backups occur on an ongoing basis. You can schedule backups to run daily, weekly, or monthly.

About this task

You can schedule backups by using the Policy wizard to create a backup policy.

Perform the following steps within the **Policy** wizard. The *Avamar Administration Guide* provides more information about policies, datasets, schedules, and retention settings.

Steps

1. Assign members to the new backup policy.
2. Assign a dataset to the new backup policy.
   To create a dataset, use the Policy wizard or select **Settings** > **Dataset** > **Add**.
3. Assign a schedule to the new backup policy.
   To create a schedule, use the Policy wizard or select **Settings** > **Schedule** > **Add**.
4. Assign a retention policy to the new backup policy.
   To create a retention policy, use the Policy wizard or select **Settings** > **Retention** > **Add**.
5. Enable scheduling for the backup policy.

Creating a dataset

A dataset specifies the data to include in a scheduled backup and the options to use for the backup. Create at least one dataset for scheduled backups on a client or group of clients. Create multiple datasets to segregate client data.

Steps

1. In the AUI navigation pane on the left, click ≫, and then click **Settings**.
   The **Setting** pane is displayed.
2. Click the **Dataset** tab.
3. Click **ADD**.
   The **Create Dataset** window is displayed.
4. In the **Dataset Name** field, type a name for the dataset.
   The name can include alphanumeric characters (A-Z, a-z, 0–9) and the following special characters: period (.), hyphen (-), and underscore (_). Do not use Unicode characters or the following special characters: ` ~ ! @ # $ % ^ & * ( ) = + [ ] { } | \ / ; : ' " < > , ?
5. From the **Plugins** list, select the **Windows SQL** plug-in.
6. Click the **Options** tab, and perform the following steps:
   a. To view advanced options, select the **Show Advanced Options** check box.
   b. Set the plug-in options. SQL plug-in backup options provides details on each option.
7. Click the **Source Data** tab and then select the data to include in the dataset:
   - To include all SQL server data on a client, select **SQL**.
   - To specify instances or databases on a client, perform the following steps:
     a. Select **Select Files and/or Folders**.
     b. In the **File/Folder Path** field, type the file path.

c. Click **ADD**.

8. Click **SUBMIT**.

Creating a backup policy

A backup policy is a collection of Avamar clients that use the same dataset, schedule, and retention settings to implement scheduled backups.

About this task

Member clients must all be in the same Avamar domain. When you create a backup policy, you define the dataset, schedule, and retention settings that apply for scheduled backups. These settings comprise the backup policy, which controls backup behavior for all members of the backup policy unless you override these settings at the client level.

The *Avamar Administration Guide* provides information about creating and editing backup policies, schedules, or retention settings.

Enabling a scheduled backup for a backup policy

Scheduled backups occur only for enabled backup policies. Backup policies are disabled by default unless you select the **Enabled** check box on the first page of the **New Policy** wizard. If you did not enable the backup policy when you created it, use the menu options in the **Policy** window to enable backups.

Steps

1. In the AUI navigation pane on the left, click ≫, and then click **Backup Policy**.
   The **Policy** page is displayed.
2. In the domain tree, select a domain or subdomain for the backup policy.
   To select a subdomain for the backup policy, toggle the **Include Sub-domain** switch to on.
3. Select a backup policy from the list.
4. To enable a backup policy, click **MORE ACTIONS** > **Enable Policy**.
5. To disable a backup policy, click **MORE ACTIONS** > **Disable Policy**.

Monitoring backups

You can monitor and view status information for backup and restore operations by using the **Activity Monitor**.

About this task

To access the **Activity Monitor**, open the navigation pane, and then click **Activity**. The **Activity Monitor** appears with a list of all activities.

NOTE: The AUI **Activity Monitor** window has been optimized for at least 1366 pixels-wide screens. Display issues might occur for smaller screens. To properly display the AUI, ensure that your display is at least 1366 pixels wide.

The **Activity Monitor** provides you with options to filter the information that appears:

- Filter activities by duration—By default, the **Activity Monitor** displays the most recent 5,000 client activities. To select a different duration, in the **Filter activities by duration** drop-down list, select **Last 24 hours** or **Last 72 hours**.
- Filter activities by domain—By default, the **Activity Monitor** displays all activities regardless of domain. To display only the activities for a specific domain, in the **Filter activities by domain** drop-down list, select a domain or subdomain.
- Filter activities by status—By default, the **Activity Monitor** displays all activities regardless of status.

  To display only activities with a specific status, at the top of the **Activity Monitor**, select one of the following options:

  - Canceled
  - Completed
  - Completed with Exceptions
  - Failed
  - Running

- Waiting

To filter activities by client, start time, plug-in, or type, click in their respective column.

The **Activity Monitor** displays the date and time that an activity began, and the total number of bytes examined during an activity.

To view activity details, expand the **Details** pane, by clicking .

Cancel backups

You can cancel a backup any time before it completes. The cancellation might take 5 minutes or longer. The backup might complete before the cancellation finishes.

Steps

1. In the AUI navigation pane on the left, click , and then click **Activity**.
   The **Activity Monitor** appears with a list of activities.
2. Select the backup from the list.
3. Click **CANCEL**.
   A confirmation dialog box is displayed.
4. Click **YES**.

Verifying backups

You can use the **VerifyOnly restore** option to verify a backup without performing a restore.

About this task

The verify-only restore feature supports the following configurations:

- Database-level backup
- Multiple databases, using database-level backup
- Instance-level backup
  - Instance level restore validates the system database backups also, without performing the SQL Service restart in single user mode.
- Availability Group (AG) database or level backup
  - Verify-only restore can be performed by using redirected restore. AG restores to the same client are not supported.

If the backup verification succeeds, the restore is marked as successful. If the backup verification is unsuccessful, the restore is marked as failed, or if the backup being verified is an AG backup, and instance backup, or involves multiple databases, the restore is marked as "partial success / failure."

Enforcement of backups to Data Domain

If the Avamar server is configured to enforce backups to a Data Domain system, the server rejects backups that are not destined for the Data Domain. This enforcement covers backups that you configure through the Avamar Administrator and the AUI, as well as from command-line interfaces and other tools.

These backups must have additional flags that indicate the storage target. The *Avamar and Data Domain System Integration Guide* provides more information about backup enforcement and the related client version requirements. Backup enforcement is disabled by default.

4

Application Consistent SQL Virtual Machine Image Backup

Topics:

- Managing advanced policies
- Prerequisites
- Add an advanced group policy
- Remove an advanced policy
- Edit an advanced policy
- View advanced policy details
- Migrate existing Backup Policy to an Advanced Policy
- View logs

Managing advanced policies

Configure an advanced policy to manage application-consistent SQL virtual machine backups, including automatic SQL discovery, automatic backup policy management, and automatic client SQL plug-in installation and registration. In an advanced policy, you can also enable the Cloud DR function for SQL virtual machine backups.

Prerequisites

Review the following prerequisites before setting up an advanced policy for application-consistent SQL virtual machine protection.

- Both administrator users and local users with the administrator role can perform advanced application-consistent SQL virtual machine backups. However, due to the introduction of User Account Control (UAC) in Microsoft Windows 7 and later versions, a local user with the administrator role must additionally turn off **Admin Approval Mode** in the UAC settings for the admin group on the guest operating system. Otherwise, the **Advanced Policy Builder** fails to install/upgrade/register client agents on SQL virtual machines.
- If the vCenter inventory name of the Virtual Machine (VM) is same as the guest computer name, change the mcserver.xml setting to `<entry key="allow_duplicate_client_names" value="true" />`. If you do not change the setting, the registration might fail during Avamar client registration with MCS.

Add an advanced group policy

Steps

1. To start the Avamar Web User Interface, open a web browser and type the following URL:

   `https://Avamar_server/aui`

   where *Avamar_server* is the DNS name or IP address of the Avamar server.

   NOTE: If your environment does not meet HTTPS certificate validation requirements, the certificate validation fails and an error message appears asking if you want to continue to download packages. Ignoring certificate validation might cause security issues.

   a. In the **Avamar Username** field, type a username with administrative privileges.
   b. In the **Avamar Password** field, type the password for the administrative user.
   c. Select **Avamar** as the **Auth Type**.

d. Click **Log In**.

2. In the AUI navigation pane on the left, click ≫, and then click **Advanced Policy** .
   The **Advanced Policy** window is displayed.
3. In the domain tree, select a domain or subdomain for the client.
4. To add an advanced policy, click **ADD**.
   The **Advanced Policy** wizard is displayed.
5. Name the advanced policy.
6. To enable the advanced policy, clear the **Disable this Advanced Policy** check box.
7. Click **NEXT**.

Configure a source

About this task

From the **Source** page, perform the following steps.

To access the **Source** page of the **Policy** wizard, in the AUI navigation pane on the left, click ≫, click **Advanced Policy** , click +, and then click **Source**.

Steps

1. In the **vCenter** field, select a target vCenter or add a target vCenter.

   NOTE: If the vCenter is not in the list, ensure that the domain under which the policy is being created can access the domain where the vCenter was originally added.

2. To create a vCenter, click +:
   The **New vCenter Client** wizard is displayed.
   a. Complete the following information in the wizard.
      - Client information, including client type and client name or IP address
      - vCenter information, including username, password, and port number.
      - Optional information
   b. Review the summary.
   c. Click **ADD**.
3. On the **Advanced Policy Builder** page, click **NEXT**.

Configure the SQL server policy

A policy contains all the information necessary to perform backups of groups of instances.

About this task

Policy objects contain three child objects:

- Schedule—A policy that controls the frequency and the start and end time each day for backups of clients. A schedule is a persistent and reusable Avamar policy that can be named and attached to multiple policies.
- Retention—A policy that defines how long the backup is stored in the backup appliance.
- Dataset—A policy that defines a set of files, directories, and file systems for each supported platform that are included or excluded in backups across a group of clients. A dataset is a persistent and reusable Avamar policy that can be named and attached to multiple policies.

From the **Overview** page, perform the following steps.

To access the **Source** page of the **Policy** wizard, in the AUI navigation pane on the left, click ≫, click **Advanced Policy** , and then click **ADD** to create a policy.

Steps

1. In the **Advanced Policy Type** field, select **SQL Server** .

2. To add a backup group, click +, and then select a backup group type from the list.
3. In the **Schedule** column, select a schedule for each backup group.

   To ensure that backups occur on an ongoing basis, scheduled backups run automatically. You can schedule backups to run daily, weekly, or monthly.

   Alternatively, you can add a schedule by selecting **Create** in the drop-down list. The **Create Schedule** wizard is displayed and opens to the **Basic Configuration** page.

4. In the **Dataset** column, select a dataset for each backup group.

   A dataset specifies the data to include in a scheduled backup and the options to use for the backup. Create at least one dataset for scheduled backups on a client or group of clients. To segregate client data, create multiple datasets.

   Alternatively, you can add a dataset by selecting **Create** in the drop-down list. The **Create DataSet** wizard is displayed.

5. In the **Retention** column, select a retention policy for each backup group.

   Retention is the time setting to automatically delete backups on an Avamar server. Retention can be set to permanent for backups that should not be deleted from an Avamar server.

   Alternatively, you can add a retention policy by selecting **Create** in the drop-down list. The **Add Retention Policy** wizard is displayed.

6. Click **NEXT**.

Configure members

About this task

From the **Members** page, perform the following steps.

> **NOTE:** Pushing clients to a group of target virtual machines simultaneously might cause a performance impact. To improve performance, adjust the Avamar throttling settings for vCenter and Esxi.
>
> Access the following Avamar throttling settings at `/usr/local/avamar/bin/vabm/vabm.cfg`:
>
> - `vabm_esxi_throttling` allows the maximum number of installation jobs to run simultaneously for each Esxi.
> - `vabm_vcenter_throttling` allows the maximum number of vCenter requests to run simultaneously for each vCenter
> - `vabm_cache_timeout` allows the maximum minutes of detection cache. The detection result of operating system and plugin version is cached in Avamar for a specific time.

Steps

1. To automatically install, upgrade, or register client agents on these virtual machines:
   a. Ensure that the UpgradeClientDownloads package has been installed on the Avamar server.
   b. Select the **Install/Upgrade/Register client agents on these VMs automatically** check box.
2. To force register the client agent from the previous Avamar server to the target Avamar server, select **Force register client agents to this Avamar**.

   The target client is force registered to the current Avamar server.
3. In the **Rule to apply** field, select an option.

   If no rule is listed in the drop-down list, ensure that the domain under which the policy is being created can access the domain where the rule was originally created. One rule maps to one policy. Ensure that the rule is not already used for another advanced policy.
4. Type the username for the virtual machine's guest operating system.

   When selecting a username for the virtual machine, consider the following information:
   - If the target virtual machine UAC is enabled, use the administrator account or the administrative account with Admin Approval Mode turned off.
   - If the target virtual machine UAC is disabled, use the administrator account or the administrative account.

NOTE: If you have not added the object BUILTIN\users to the SQL login, the advanced policy cannot automatically add the sysadmin role to the NT AUTHORITY\SYSTEM object. To use NT authentication for SQL backup, manually add the sysadmin role before proceeding. For more information, see the SQL server documentation.

5. Type the password of the virtual machine.
6. To manually push the client agents on these virtual machines, click **RETRY INSTALLATION**.
7. To display the list of members and view information for a virtual machine, click **REFRESH**.

   This action scans all the virtual machines under the configured vCenter, evaluates the rule matched members, and then detects the member information. This process might take some time to complete.
8. Click **NEXT**.

Configure a proxy for the image backup

About this task

From the **Members** page, perform the following steps. To access the **Source** page of the **Policy** wizard, in the AUI navigation pane on the left, click ≫, click **Advanced Policy** , click +, and then click **Proxies**.

Steps

1. To enable **Auto proxy mapping**, select the checkbox.
2. Select a proxy from the list.
3. To start the plugin installation and registration, click **SUBMIT**.

Remove an advanced policy

Steps

1. In the AUI navigation pane on the left, click ≫, and then click **Advanced Policy** .
   The **Advanced Policy** window appears.
2. In the domain tree, select a domain or subdomain for the client.
3. Select the solution that you want to remove, and then click the **Delete** button.

Edit an advanced policy

About this task

Do not re-configure individual policies that were created as part of an advanced policy.

Steps

1. In the AUI navigation pane on the left, click ≫, and then click **Advanced Policy** .
   The **Advanced Policy** window appears.
2. In the domain tree, select a domain or subdomain for the client.
3. Select the policy that you want to edit, and then click the **Edit** button.

   The **Advanced Policy** wizard appears.

   NOTE: When editing an advanced policy, you cannot change the policy name or select a different vCenter.

View advanced policy details

Steps

1. In the AUI navigation pane on the left, click ≫, and then click **Advanced Policy** .
   The **Advanced Policy** window appears and lists the configured policies.
2. Review the policy details.
3. To view detailed information for advanced policy members, click the **VIEW VM LIST** button.

Migrate existing Backup Policy to an Advanced Policy

Steps

1. Ensure that the old policy contains SQL Server virtual machines.

   Advanced policy is only for SQL Server virtual machines. If the old policy does not contain only SQL Server virtual machines, split the policy.
2. Design the rule to match the virtual machines in the old policy.

   For example, you can add new tag for these virtual machines and use tags in your rule.
3. Create an advanced policy with the rule that your created in the previous step.
4. Push the agent manually.
5. Move the client to the advanced policy directory.
6. If required, delete the old policy.

View logs

Steps

1. In the AUI navigation pane on the left, click ≫, and then click **Advanced Policy** .
   The **Advanced Policy** window appears and lists the configured policies.
2. To view the logs for a policy, select a policy from the list, and then click 👁.
   The **VM list** window appears.
3. Select a virtual machine from the list, and then click **View Logs**.
4. To view system logs, browse to `/usr/local/avamar/var/vabm/log/daemon/`.
5. To view a detailed log for each virtual machine, browse to `/usr/local/avamar/var/vabm/log/<vm-uuid>/<vm-uuid.log>`.

5

Restore

Topics:

- Restore requirements
- Determine the restore size for an SQL server database
- Restoring a database to the original client
- Restoring a database to a different client
- Restore features available by using Avamar Administrator
- Table Level restore
- Monitor restores
- Cancel restores

Restore requirements

To ensure a successful restore, ensure that the environment meets the necessary requirements.

The Avamar Plug-in for SQL Server performs only offline restores of SQL Server data. Online restore is not supported. The Avamar Plug-in for SQL Server does not support the restore of individual files, except by restoring partial files from an SQL backup into the same instance and database when the database is online during the restore.

To perform table level recovery (TLR), the Avamar Plug-in for SQL TLR is required. When the Avamar Plug-in for SQL TLR is installed, the ItemPoint for Microsoft SQL Server is also installed.

Software requirements for restore

To restore an instance, database, filegroup, or file to SQL Server by using the Avamar Plug-in for SQL Server, ensure that the software in the environment meets the necessary requirements.

- The following software must be running on both the source and destination systems:
  - Microsoft SQL Server
  - Avamar Client for Windows
  - Avamar Plug-in for SQL Server

  If the Avamar Plug-in for SQL Server is not installed on the target server, or you want to use the standard SQL Server restore tools for features that the Avamar Plug-in for SQL Server does not provide, then you can restore a SQL Server database from an Avamar backup to operating system files. You can then use SQL Server tools to restore the database.
- To perform table level recovery, the Avamar Plug-in for SQL TLR must be installed.
- To restore a system database, the target SQL Server installation for the restore must have the same SQL Server version and service pack as the SQL Server installation on which the backup occurred. Otherwise, restore fails. The "You cannot restore system database backups to a different build of SQL Server" article on the Microsoft Support website provides details.
- To restore a user database, the target SQL Server installation for the restore must have the same or a newer SQL Server version and service pack as the SQL Server installation on which the backup occurred. Restore fails when you attempt to restore from a newer SQL Server version to an earlier SQL Server version. For example, you can restore a backup of a SQL Server 2008 user database to a SQL Server 2012 instance. However, restore of a SQL Server 2012 user database to a SQL Server 2008 instance is not supported. This compatibility requirement is enforced by Microsoft SQL Server and is not an Avamar limitation.

  NOTE: Restore of a user database to a newer SQL Server version (for example, restore from SQL Server 2008 to SQL Server 2012) should work in most cases that are based on SQL Server version compatibility guidelines. However, not all restore scenarios with the SQL Server plug-in have been validated.
- The destination server must be registered with the same Avamar server as the source.
- If the SQL Server installation is in a failover cluster, then you configured the Avamar cluster client for all SQL Server cluster nodes.

- If you enabled AlwaysOn availability groups, then you configured the Avamar cluster client for the availability group listener.

Tail-log backup and point-in-time restore requirements

Review the requirements to perform a tail-log backup or point-in-time restore to ensure successful completion of the restore.

- To perform a tail-log backup during the restore process, the database must be online and using either the full or bulk-logged recovery model. To perform a point-in-time restore, the database must be using the full recovery model. As a result, you cannot perform either a tail-log backup or a point-in-time restore of system databases such as the master and msdb databases because those databases use the simple recovery model.
- To restore to a specific point in time, you must provide the transaction date and time or named mark to which to recover from the SQL Server transaction log. The SQL Server documentation on the Microsoft website provides details on how to access transaction log information.
- The point in time that you are restoring to must be after the finish time for the most recent full backup. In addition, if the point in time is before the start time of the most recent transaction log (incremental) backup, then a tail-log backup is not required. However, a tail-log backup is required if the point in time is after the most recent transaction log backup.
- When you specify the point in time for restore, do not specify the start time of the selected transaction log backup if it is not the last backup in the backup sequence. Otherwise, the restore fails and a tail-log backup does not occur even if you select the **Tail-log backup** option.

Requirements for point-in-time restore after changing the backup storage

You can restore to a point in time between two full backups when you change the backup storage from one Data Domain system to another Data Domain system.

To restore to a point in time between a full backup on the first Data Domain system and the first full backup to the new Data Domain system, select the **Force incremental backup** checkbox when you perform the first full backup to the new Data Domain system. Otherwise, clear the **Force incremental backup** checkbox.

When you select the **Force incremental backup** checkbox, a restore error appears because full backup on one Data Domain system and incremental (transaction log) backup on another Data Domain system is not supported. As a result, the backup process creates a single backup set that includes both the new full backup and the forced incremental backup.

To restore the database to a point in time between the full backup on the first Data Domain system and the full backup on the second Data Domain system:

1. Restore the full backup from the Data Domain system to a file.
2. Restore the forced incremental (transaction log) backup from the second Data Domain system to a file.
3. Use those files to restore the database to the necessary point in time by using SQL Server tools such as SQL Server Management Studio or the Microsoft SQL Server `sqlcmd` utility. The Microsoft website also provides full details on how to use SQL Server Management Studio to restore a database backup to a specific point in time.

Requirements to restore secondary database files

When you restore the secondary data files of a database to the original location, the restore must meet the following requirements.

- You must select the most recent backup of the database for the restore.
- You must perform a tail-log backup.
- You must perform the restore with a single restore operation.

You cannot restore additional secondary data files after you perform a restore with a tail-log backup.

To restore the secondary data files of a database with two different restore operations, restore the backup to a file and then use SQL Server management tools to perform the restores.

Requirements to restore the Report Server database

Stop the SQL Server Reporting Services before you restore the Report Server database. Otherwise, the Report Server database does not restore correctly.

SQL Server write permission requirements

SQL Server must have write permissions to the location to which you restore data. Otherwise, the restore fails with an `Access is denied` error.

Determine the restore size for an SQL server database

Use the following procedures to determine the disk space requirements of individual databases in the backup set as well as the total space requirements. These procedures determine the restore size for a database without downloading the complete backup content. The backup header is downloaded to fetch the sizing information.

NOTE: Performing this operation might result in a log entry that the operation was externally canceled and the `Restore interrupted` warning message appears. This warning can be ignored.

Determine the space required to restore the system database

To determine the space required to restore the database, use the --print-restore-size flag.

About this task

NOTE: To determine the restore size for the system database, use the --restoresystem flag.

Determine the space required to restore a single SQL server database

Before restoring a single SQL server database "DB1", determine the disk space requirements by using the CLI.

About this task

At a command prompt, type the following command:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin --ap=password --path=/SQL/
SQLServer1 --print-restore-size --hostname-sql=SQLServer1 --labelnum=28 "(local)/DB1"
```

Output similar to the following appears:

```
==================================================
Size of the Database (local)\DB1: 3.000 MB
Size of the Database (local)\DB1_log: 1.000 MB
--------------------------------------------------------------------------------------------
Total require restore size: 4.000 MB
==================================================
```

In this example, DB1 is the name of the database that is being restored.

Determine the space required to restore multiple SQL server databases

Before restoring multiple SQL server databases, determine the disk space requirements by using the CLI.

About this task

At a command prompt, type the following command:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin --ap=password --path=/SQL/
SQLServer1 --print-restore-size --hostname-sql=SQLServer1 --labelnum=28 "(local)/DB2"
"(local)/DB3"
```

Output similar to the following appears:

```
==================================================
 Size of the Database (local)\DB2: 2.489 GB
 Size of the Database (local)\DB2_log: 984.4 MB
 Size of the Database (local)\DB3: 3.000 MB
 Size of the Database (local)\DB3_log: 1.000 MB
-------------------------------------------------------------------------------------
 Total require restore size: 3.454 GB
 ==================================================
```

Restoring a database to the original client

You can restore SQL databases to the original directory on the original client.

Steps

1. In the AUI navigation pane on the left, click ≫, and then click **Asset Management**.
   The **Asset Management** window is displayed.
2. In the domain tree, select the domain that contains the SQL server.
   You cannot view clients outside the domain for the login account. To view all clients, log in to the root domain.
   A list of Avamar clients is displayed in the pane below the domains list.
3. From the list of clients, select the client that runs the SQL server.
4. (Optional) To locate backups by date:
   a. In the right pane, click **VIEW MORE**.
   b. Click **SEARCH**.
   c. In the **From** and **To** fields, specify the date range.
   d. Click **RETRIEVE**.
   e. In the list of backups, select a backup.
   The list of backups for the date range is displayed.
5. Click the **Restore** tab.
   The **Restore** wizard is displayed and the **Destination Client** pane is displayed on the right.
6. In the **Destination Client** pane, perform the following steps:
   a. Select **Restore to original client**.
   b. Click **NEXT**.
   The **Backup Content** pane is displayed.
7. In the **Backup Content** pane, perform the following steps:
   a. In the left pane, select the folder from the tree.
      The **Backup Content** pane displays a list of databases within the backup.
   b. In the right pane, select the databases that you want to restore.
   c. Click **NEXT**.
   The **Destination Location** pane is displayed.
8. In the **Destination Location** pane, perform the following steps:

a. Select **Restore everything to its original location**.
b. Click **Next**.

The **More Options** pane is displayed.

9. In the **More Options** pane, set the plug-in options:

   (Optional) Toggle the **Show Advanced Options** switch to view advanced configuration options.

   SQL plug-in restore options provides more information about the basic restore options.

   SQL Restore Options provides more information about the advanced restore options.

10. Click **NEXT**.

    The **Summary** pane is displayed.

11. Review the provided information, and then click **FINISH**.

Restoring a database to a different client

You can restore SQL databases to a different client.

Prerequisites

Steps

1. In the AUI navigation pane on the left, click ≫, and then click **Asset Management**.
   The **Asset Management** window is displayed.
2. In the domain tree, select the domain that contains the SQL server.

   You cannot view clients outside the domain for the login account. To view all clients, log in to the root domain.

   A list of Avamar clients is displayed in the pane below the domains list.
3. From the list of clients, select the client that runs the SQL server.
4. (Optional) To locate backups by date:
   a. In the right pane, click **VIEW MORE**.
   b. Click **SEARCH**.
   c. In the **From** and **To** fields, specify the date range.
   d. Click **RETRIEVE**.
   e. In the list of backups, select a backup.

   The list of backups for the date range is displayed.
5. Click the **Restore** tab.

   The **Restore** wizard is displayed and the **Destination Client** pane is displayed on the right.
6. In the **Destination Client** pane, perform the following steps:
   a. Select **Restore everything to a different client**.
   b. In the domain tree, select the domain for the client.
   c. In the list of clients, select the destination client.
      The client is displayed in the **Destination Client** field.
   d. Click **NEXT**.

      The **Backup Content** pane is displayed.
7. In the **Backup Content** pane, perform the following steps:
   a. In the left pane, select the folder from the tree.

      The **Backup Content** pane displays a list of databases within the backup.

   b. In the right pane, select the databases that you want to restore.
   c. Click **NEXT**.

   The **Destination Location** pane is displayed.
8. In the **Destination Location** pane, perform the following steps:
   a. Select **Restore everything to a different location**.
   b. In the **Restore Location** field, type the file path of the target location where you want to restore the data.

For example: E:/dbrestore_dir/

c. Click **NEXT**.

The **More Options** pane is displayed.

9. In the **More Options** pane, set the plug-in options:

   (Optional) Toggle the **Show Advanced Options** switch to view advanced configuration options.

   SQL plug-in restore options provides more information about the basic restore options.

   SQL Restore Options provides more information about the advanced restore options.

10. Click **NEXT**.

    The **Summary** pane is displayed.

11. Review the provided information, and then click **FINISH**.

Restore features available by using Avamar Administrator

This section describes the restore features that are available by using Avamar Administrator.

Finding a backup

The first step to restore data is to find the backup with the data that you want to restore. You can find Avamar client backups by searching either for a specific date or for specific content.

About this task

Locate backups by date when one or more of the following situations apply:

- You have saved all data for the client in a single backup set.
- The exact pathname or name of the data to restore is unknown.
- The backup that you want to restore is before a specific date or event. For example, you know the approximate date when data was lost or corrupted. in which you can search for a backup before that date.
- The specific types of backups are known. For example, scheduled disaster recovery backups are running every Wednesday and Saturday night and full volume backups daily. When rebuilding a server, select the disaster recovery backup with the date closest to the event that caused the loss of data.

Locate backups by the content of the backup when one or more of the following situations apply:

- You have backed up data on the client in separate backup sets.
- You want to view multiple versions of the same file so that you can decide the version to restore.
- The date of the backup or the content of a backup is unknown, but you know the name of the data to restore.

NOTE: Avamar generally supports the use of specific supported international characters in directory, folder, and filenames. However, proper display of international language characters is contingent on the client computer's Java locale and installed system fonts being compatible with the original language. If you browse backups that were created with international characters and a compatible font is not installed, then any characters that the system cannot resolve appear as rectangles. This action is a normal limitation of that particular situation and does not affect the ability to restore these directories, folders, or files. The *Avamar Release Notes* provide additional international language support information.

Finding a backup by date

Steps

1. In Avamar Administrator, click the **Backup & Restore** launcher link button.
   The **Backup, Restore and Manage** window appears.
2. Click the **Restore** tab.
   The upper left pane contains a list of domains.
3. Select the domain that contains the client.

You cannot view clients outside the domain for the login account. To view all clients, log in to the root domain.

A list of Avamar clients appears in the pane under the domains list.

4. From the list of clients, select the computer that is running SQL Server. Keep the following points in mind when you select a client:
   - If you are restoring databases in an AlwaysOn availability group, then select the cluster client for the availability group listener.
   - If you are restoring databases on shared storage in a failover cluster, then select the cluster client for the virtual server.
   - You cannot view clients outside the domain for the login account. To view all clients, log in to the root domain.
5. Click the **Restore** tab.
6. Click the **By Date** tab.
7. Select the backup date from the calendar. Valid backups occurred on dates with a yellow highlight.
   A list of backups on that date appears in the **Backups** table next to the calendar.

   If a backup is the second or later full backup and you selected the **Force incremental backup after full backup** checkbox for the backup, then three backups appear in the **Backups** table:

   - The first backup in the list is the forced incremental backup. You can use this backup to restore from the full backup through the forced incremental backup.
   - The backup with `forced_incremental` appended to the backup label is the forced incremental backup. You can use this backup to restore from the previous full backup through the forced incremental backup.
   - The backup with `full_backup_only` appended to the backup label is the full backup.
8. Select the backup to restore from the **Backups** table.

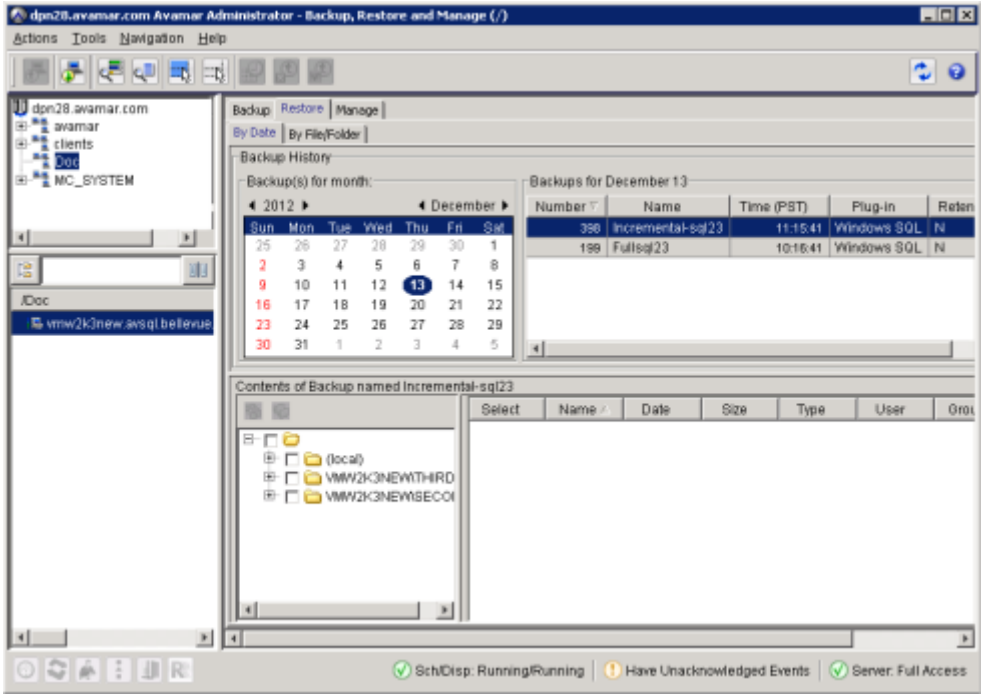

Figure 9. Backup, Restore and Manage window

Keep in mind the following points when you select the backup:

- In the **Type** column in the lower-right pane, f-0 indicates a full backup, d-*n* indicates a differential backup, and i-*n* indicates a transaction log (incremental) backup.
- If you are restoring from a transaction log or differential backup, select the backup that corresponds to the date and time to which you want to recover.

  During the restore process, Avamar automatically restores any necessary data from the full backup, then restores and applies the intervening backup files as necessary. In other words, you do not need to select the full backup in addition to the transaction log or differential backup.

- Select the most recent backup if you plan to perform a tail-log backup and recover to a point in time since that last backup. A tail-log backup includes only transactions that have not been included yet in a backup.

9. In the two bottom panes of the **Backup, Restore and Manage** window, select the data to restore:

- To restore everything listed for the instance, select the checkbox next to the instance in the folder tree in the lower-left pane.
- To restore a database and its logs, expand the node for the instance in the folder tree in the lower-left pane, and then select the checkbox next to the database.
- To restore a filegroup, expand the node for the instance in the folder tree in the lower-left pane, select the database in the lower-left pane, and then select the checkbox next to the files in the filegroup in the lower-right pane.

  If there are multiple files in the filegroup, ensure that you select the checkbox next to each file to ensure that you restore the entire filegroup.

  The name of the filegroup to which a file belongs appears in the **Filegroup** column of the lower-right pane.

  If you are restoring from a transaction log or differential backup and you plan to restore the files to a different instance, ensure that you select the checkbox next to all files in all filegroups. You cannot restore individual files from a transaction log or differential backup to a different instance.

NOTE: You cannot accurately determine the exact size of a restored database until after the restore operation completes. As a result, database sizes that appear in Avamar Administrator when you perform a restore may be smaller than when you perform a backup.

10. Continue with the restore as described in the following topics:
    - Restoring to the original location on page 69
    - Restoring to a new database in the original instance on page 70
    - Restoring to a different instance on page 71
    - Restoring to a file on page 72
    - Restoring system databases on page 78

Finding a backup by content

Steps

1. In Avamar Administrator, click the **Backup & Restore** launcher link button.
   The **Backup, Restore and Manage** window appears.
2. Click the **Restore** tab.
   The upper left pane contains a list of domains.
3. Select the domain that contains the client.

   You cannot view clients outside the domain for the login account. To view all clients, log in to the root domain.

   A list of Avamar clients appears in the pane under the domains list.
4. From the list of clients, select the computer that is running SQL Server. Keep the following points in mind when you select a client:
   - If you are restoring databases in an AlwaysOn availability group, then select the cluster client for the availability group listener.
   - If you are restoring databases on shared storage in a failover cluster, then select the cluster client for the virtual server.
   - You cannot view clients outside the domain for the login account. To view all clients, log in to the root domain.
5. Click the **Restore** tab.
6. Click the **By File/Folder** tab.
7. In the **Enter path to retrieve history for** text box, specify the path to the instance or database by using one of the methods in the following table.

Table 5. Enter path to retrieve history for text box

Method	Steps to specify the path
Browse to the instance or database	a. Click **Browse**. The **Select File or Folder** dialog box appears. b. Select the client from the **Clients** tree in the left pane. c. In the middle pane, expand the **Windows SQL** plug-in node. d. Under the **Windows SQL** plug-in node, select the SQL Server instance that contains the databases for restore.

Table 5. Enter path to retrieve history for text box (continued)

Method	Steps to specify the path
	A list of the databases for that instance appears in the right pane of the **Select File or Folder** dialog box. e. To select all databases in an instance, select the checkbox next to the instance in the middle pane. Or, to select an individual database, select the checkbox next to the database in the right pane. f. Click **OK**.
Type the path to the instance or database	Type the full path to the client folder or file in the **Enter path to retrieve history for** text box using one of the following formats: ● To restore the local instance, type **`(local)`**. ● To restore a database in the local instance, type **`(local)/`*`database/`***. ● To restore a named instance, type ***`client\instance/`***. ● To restore a database in a named instance, type ***`client\instance/database/`***. ● To restore a database if there is only one instance on the client and it is not the local instance, type ***`client/database/`***. where *client* is the name of the computer that is running SQL Server, *instance* is the name of the named instance, and *database* is the name of the database.

8. Click **Retrieve**.
   The **Version History** table lists all versions and sizes of the content in backups for the client.
9. Select the version in the **Version History** table.
   All backups for the client that contain the version appear in the **Backups** table next to the **Version History** table.
10. Select the backup to restore from the **Backups** table.

    Keep in mind the following points when you select the backup:

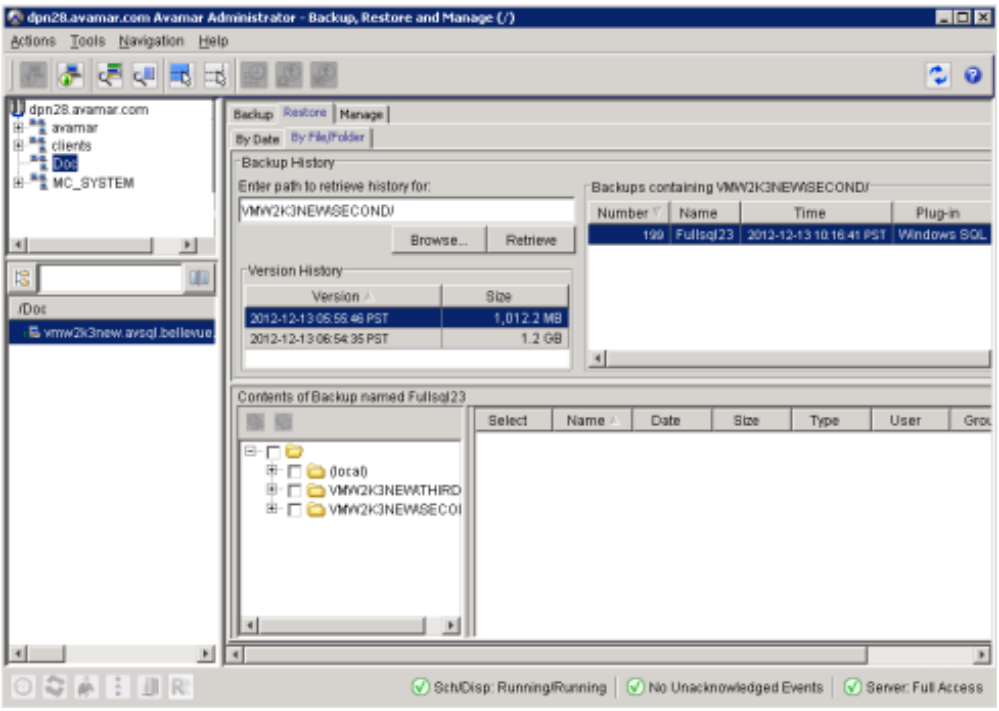

Figure 10. Backup, Restore and Manage window

- In the **Type** column in the lower-right pane, f-0 indicates a full backup, d-*n* indicates a differential backup, and i-*n* indicates a transaction log (incremental) backup.
- If you are restoring from a transaction log or differential backup, select the backup that corresponds to the date and time to which you want to recover.

  During the restore process, Avamar automatically restores any necessary data from the full backup, then restores and applies the intervening backup files as necessary. In other words, you do not need to select the full backup in addition to the transaction log or differential backup.

- Select the most recent backup if you plan to perform a tail-log backup and recover to a point in time since that last backup. A tail-log backup includes only transactions that have not been included yet in a backup.

11. In the two bottom panes of the **Backup, Restore and Manage** window, select the data to restore:
    - To restore everything listed for the instance, select the checkbox next to the instance in the folder tree in the lower-left pane.
    - To restore a database and its logs, expand the node for the instance in the folder tree in the lower-left pane, and then select the checkbox next to the database.
    - To restore a filegroup, expand the node for the instance in the folder tree in the lower-left pane, select the database in the lower-left pane, and then select the checkbox next to the files in the filegroup in the lower-right pane.

      If there are multiple files in the filegroup, ensure that you select the checkbox next to each file to ensure that you restore the entire filegroup.

      The name of the filegroup to which a file belongs appears in the **Filegroup** column of the lower-right pane.

      If you are restoring from a transaction log or differential backup and you plan to restore the files to a different instance, ensure that you select the checkbox next to all files in all filegroups. You cannot restore individual files from a transaction log or differential backup to a different instance.

    NOTE: You cannot accurately determine the exact size of a restored database until after the restore operation completes. As a result, database sizes that appear in Avamar Administrator when you perform a restore may be smaller than when you perform a backup.

12. Continue with the restore as described in the following topics:
    - Restoring to the original location on page 69
    - Restoring to a new database in the original instance on page 70
    - Restoring to a different instance on page 71
    - Restoring to a file on page 72

- Restoring system databases on page 78

Restoring to the original location

You can restore a SQL Server instance, database, filegroup, or file to its original location.

About this task

NOTE: If you are restoring to the original location in an AlwaysOn availability group, use the instructions in Restoring to an AlwaysOn availability group on page 83.

There are two restore options when you restore a SQL Server instance, database, or filegroup to its original location.

- A standard restore with a tail-log backup is the most common restore procedure. During this procedure, a tail-log backup occurs to capture transactions that are not yet in a backup. Then the database is restored from the most recent full backup and any differential or transaction log backups.
- A restore with the SQL Server `REPLACE` option that completely overwrites the database might be necessary, for example, if a previous database restore exited with the following SQL Server error in the Avamar SQL restore log:

  ```
  One or more devices or files already exist.
  Reissue the statement using the WITH REPLACE
  option to overwrite these files and devices.
  ```

  NOTE: When you select the Avamar option to use the SQL Server `REPLACE` option, it adds an SQL `WITH REPLACE` clause statement to the restore Transact-SQL command. This statement overrides a SQL Server safety check to prevent you from accidentally overwriting a different database or file. The Microsoft Transact-SQL documentation provides more information in the `RESTORE` command section.

Steps

1. Ensure that the environment meets the guidelines in Restore requirements on page 59.
2. In Avamar Administrator, click the **Backup & Restore** launcher link button.
   The **Backup, Restore and Manage** window appears.
3. Find the backup to restore:
   - Finding a backup by date on page 64
   - Finding a backup by content on page 66

   The backup to restore is selected in the **Backups** table.
4. Select **Actions** > **Restore Now**.
   The **Restore Options** dialog box appears.
5. Ensure that **Windows SQL** appears in the **Restore Plug-in** list.
6. From the **Avamar encryption method** list, select the encryption method to use for data transfer between the Avamar server and the client during the restore.

   The encryption technology and bit strength for a client/server connection depend on several factors, including the client operating system and Avamar server version. The *Avamar Product Security Guide* provides additional information.
7. Leave the default selection of **Restore to the original location**.
8. Click **More Options**.
   The **Restore Command Line Options** dialog box appears.
9. Select the restore and recovery options:
   - To create a tail-log backup and perform a direct restore with recovery, leave the **Use SQL REPLACE option** checkbox clear and the **Tail-log backup** checkbox selected.
   - To perform a direct restore with `REPLACE`, select the **Use SQL REPLACE option** checkbox and clear the **Tail-log backup** checkbox.
10. If you are restoring in an AlwaysOn availability group and you want to restore on only the primary replica, select the **Restore only on primary replica** checkbox.

    Leave the checkbox clear to automatically restore to both the primary replica and any secondary replicas.
11. From the **Encryption method from Data Domain system** list, select the encryption method for data transfer during the restore if the backup was stored on a Data Domain system.
12. To restore system databases, set the plug-in options as discussed in System database restore options on page 90.
13. (Optional) Select **Show Advanced Options**, and set the advanced options as discussed in the following topics:

- Recovery operation options on page 89
- Authentication options on page 91
- Point-in-time recovery options on page 91

14. Disregard the redirected restore options, which are only necessary when you restore to a different location.
15. Click **OK** on the **Restore Command Line Options** dialog box.
16. Click **OK** on the **Restore Options** dialog box.
    The following status message appears: `Restore initiated`.
17. Click **OK**.

Next steps

- If you perform a tail-log backup and the tail-log backup fails to complete, then the restore cannot take place. Review the log file to determine the cause of the failure. Correct the problem, and then restart the restore.

  Keep in mind that if you clear the **Tail-log backup** checkbox to prevent the tail-log backup from occurring, then the restore includes only the transactions up to the selected backup. You may lose any transactions in the tail of the log.

- After the restore completes successfully, perform a full backup of the database and clear the **Force incremental backup after full backup** checkbox in the plug-in options for the backup. If the checkbox is selected when a full backup occurs after a restore, then the transaction log backup that occurs automatically after the full backup fails.

Restoring to a new database in the original instance

You can restore a database to the original SQL Server instance on the same SQL Server client but with a new database name.

Steps

1. Ensure that the environment meets the guidelines in Restore requirements on page 59.
2. In Avamar Administrator, click the **Backup & Restore** launcher link button.
   The **Backup, Restore and Manage** window appears.
3. Find the backup to restore:
   - Finding a backup by date on page 64
   - Finding a backup by content on page 66

   The backup to restore is selected in the **Backups** table.
4. Select **Actions** > **Restore Now**.
   The **Restore Options** dialog box appears.
5. Ensure that **Windows SQL** appears in the **Restore Plug-in** list.
6. From the **Avamar encryption method** list, select the encryption method to use for data transfer between the Avamar server and the client during the restore.

   The encryption technology and bit strength for a client/server connection depend on several factors, including the client operating system and Avamar server version. The *Avamar Product Security Guide* provides additional information.
7. Leave the default selection of **Restore to the original location**.
8. Click **More Options**.
   The **Restore Command Line Options** dialog box appears.
9. Select the **Show Advanced Options** checkbox.
   Several advanced options appear in red.
10. Leave the **Use SQL REPLACE option** checkbox clear.
11. Specify whether to perform a tail-log backup during the restore by selecting or clearing the **Tail-log backup** checkbox.
12. If you are restoring in an AlwaysOn availability group and you want to restore the database with the new name on only the primary replica, select the **Restore only on primary replica** checkbox.
13. From the **Encryption method from Data Domain system** list, select the encryption method for data transfer during the restore if the backup was stored on a Data Domain system.
14. In the **New database name** box, specify the new database name.
15. In the **Alternate database location** box, specify the path to which to restore the database files, or leave the box blank to restore the files to the original location.
16. Select the location to which to restore the log files for the database:
    - To restore the log files to the same location as the database, select **Same as alternate database location** from the **Alternate log location** list.

- To restore the log files to a different location than the database, select **Different location than database** from the **Alternate log location** list, and then specify the path for the log files in the **Path to alternate log location** box.

17. (Optional) Set other plug-in options as discussed in the following topics:
    - Recovery operation options on page 89
    - Authentication options on page 91
    - Point-in-time recovery options on page 91
18. Disregard the system databases options, which are only necessary when you restore a system database.
19. Click **OK** on the **Restore Command Line Options** dialog box.
20. Click **OK** on the **Restore Options** dialog box.
    The following status message appears: `Restore initiated`.
21. Click **OK**.

Next steps

- If you perform a tail-log backup and the tail-log backup fails to complete, then the restore cannot take place. Review the log file to determine the cause of the failure. Correct the problem, and then restart the restore.

  Keep in mind that if you clear the **Tail-log backup** checkbox to prevent the tail-log backup from occurring, then the restore includes only the transactions up to the selected backup. You may lose any transactions in the tail of the log.

- After the restore completes successfully, perform a full backup of the database and clear the **Force incremental backup after full backup** checkbox in the plug-in options for the backup. If the checkbox is selected when a full backup occurs after a restore, then the transaction log backup that occurs automatically after the full backup fails.

Restoring to a different instance

You can restore one or more instances, databases, filegroups, or files to a different SQL Server instance on either the same SQL Server client or a different SQL Server client.

Steps

1. Ensure that the environment meets the guidelines in Restore requirements on page 59.
2. In Avamar Administrator, click the **Backup & Restore** launcher link button.
   The **Backup, Restore and Manage** window appears.
3. Find the backup to restore:
   - Finding a backup by date on page 64
   - Finding a backup by content on page 66

   The backup to restore is selected in the **Backups** table.
4. Select **Actions** > **Restore Now**.
   The **Restore Options** dialog box appears.
5. Ensure that **Windows SQL** appears in the **Restore Plug-in** list.
6. From the **Avamar encryption method** list, select the encryption method to use for data transfer between the Avamar server and the client during the restore.

   The encryption technology and bit strength for a client/server connection depend on several factors, including the client operating system and Avamar server version. The *Avamar Product Security Guide* provides additional information.
7. Specify the destination client in the **Restore Destination Client** box:
   - To restore to the original client, leave the default setting of the original client domain and name.
   - To restore to a different client, click **Browse** and then browse to the client.
8. Select **Restore to a different SQL Server instance or location**.
9. Select the instance on the destination client for the restore:
   a. Next to the **Select the SQL instance on the destination client where the items should be restored** box, click **Browse**.
      The **Browse for Restore Destination** dialog box appears.
   b. Select **Windows SQL**.
   c. In the right pane, select the checkbox next to the instance.
   d. Click **OK**.
10. Set the destination file path for the database and log files to restore:
    a. Click **Set Destination**.

The **Set Destination** dialog box appears.

b. To specify a path for a single file, select the row in the table. Or, to specify the same path for multiple files in the list, press **Shift** and select the rows.

c. Click **Browse**.
The **Browse for File, Folder, or Directory** dialog box appears.

d. Select **Windows File System**.

e. In the right pane, browse to and select the checkbox for the folder to which to restore the selected files.

f. Click **OK** to return to the **Set Destination** dialog box.

g. Repeat step b through step f for the remaining rows in the **Set Destination** dialog box.

h. Click **OK** to return to the **Restore Options** dialog box.

11. Click **More Options**.
The **Restore Command Line Options** dialog box appears.

12. Select the **Show Advanced Options** checkbox.
Several advanced options appear in red.

13. Leave the **Use SQL REPLACE option** checkbox clear.

14. Clear the **Tail-log backup** checkbox to disable tail-log backup. Tail-log backup is not supported when you restore to a different SQL Server instance.

15. If you are restoring to an AlwaysOn availability group in a different SQL Server instance, select the **Restore only on primary replica** checkbox to restore the database on only the primary replica.

16. From the **Encryption method from Data Domain system** list, select the encryption method for data transfer during the restore if the backup was stored on a Data Domain system.

17. Disregard the **System Databases** options. Do not restore system databases to a different instance.

18. (Optional) To restore the database with a new name, type the new name in the **New database name** box.

19. Disregard the remaining **Redirected Restore** settings (**Alternate database location**, **Alternate log location**, and **Path to alternate log location**). You already specified these settings on the **Set Destination** dialog box.

20. (Optional) Set other plug-in options as discussed in the following topics:
    - Recovery operation options on page 89
    - Authentication options on page 91
    - Point-in-time recovery options on page 91

21. If the target server for the restore has version 6.1 or earlier of the SQL Server plug-in, then you may need to decrease the buffer size that SQL Server uses to read and write backup images. The buffer size increased in version 7.0 of the SQL Server plug-in to improve backup and restore performance.

    To decrease the buffer size to the size before version 7.0, set the `--max-transfer-size` attribute to `65536` (64 KB):

    a. Click **More** on the **Restore Command Line Options** dialog box.

    b. Type **`--max-transfer-size`** in the **Enter Attribute** box.

    c. Type **`65536`** in the **Enter Attribute Value** box.

    d. Click **+**.

22. Click **OK** on the **Restore Command Line Options** dialog box.

23. Click **OK** on the **Restore Options** dialog box.
The following status message appears: `Restore initiated`.

24. Click **OK**.

Next steps

After the restore completes successfully, perform a full backup of the database and clear the **Force incremental backup after full backup** checkbox in the plug-in options for the backup. If the checkbox is selected when a full backup occurs after a restore, then the transaction log backup that occurs automatically after the full backup fails.

Restoring to a file

You can restore SQL Server user and system databases from an Avamar backup to operating system files.

About this task

You may want to perform this type of restore in the following situations:

- The Avamar Plug-in for SQL Server is not installed on the target server.

- You want to use the standard SQL Server restore tools for features that the SQL Server plug-in does not provide.

You can use either the Avamar Plug-in for SQL Server or the Avamar Plug-in for the Windows File System to restore a database to a file. You can then use SQL Server tools to restore the data to SQL Server.

Restoring to a file with the SQL Server plug-in

You can restore an instance or database to operating system files by using the Avamar Plug-in for SQL Server.

Steps

1. Ensure that the environment meets the guidelines in Restore requirements on page 59.
2. In Avamar Administrator, click the **Backup & Restore** launcher link button.
   The **Backup, Restore and Manage** window appears.
3. Find the backup to restore:
   - Finding a backup by date on page 64
   - Finding a backup by content on page 66

   The backup to restore is selected in the **Backups** table.
4. Select **Actions** > **Restore Now**.
   The **Restore Options** dialog box appears.
5. Ensure that **Windows SQL** appears in the **Restore Plug-in** list.
6. From the **Avamar encryption method** list, select the encryption method to use for data transfer between the Avamar server and the client during the restore.

   The encryption technology and bit strength for a client/server connection depend on several factors, including the client operating system and Avamar server version. The *Avamar Product Security Guide* provides additional information.
7. Specify the destination client in the **Restore Destination Client** box:
   - To restore to the original client, leave the default setting of the original client domain and name.
   - To restore to a different client, click **Browse** and then browse to the client.
8. Select **Restore SQL Server backup as files to the file system**.
9. Set the destination file path for the database and log files to restore:
   a. Click **Set Destination**.
      The **Set Destination** dialog box appears.
   b. Click **Browse**.
      The **Browse for File, Folder, or Directory** dialog box appears.
   c. Select **Windows File System**.
   d. In the right pane, browse to and select the checkbox for the folder to which to restore the selected files.
   e. Click **OK** to return to the **Set Destination** dialog box.
   f. Click **OK** to return to the **Restore Options** dialog box.
10. Click **More Options**.
    The **Restore Command Line Options** dialog box appears.
11. From the **Encryption method from Data Domain system** list, select the encryption method for data transfer during the restore if the backup was stored on a Data Domain system.
12. If the restore includes system databases, select the **Restore system databases** checkbox.
13. Disregard the remaining options, which do not apply when you restore to a file.
14. Click **OK** on the **Restore Command Line Options** dialog box.
15. Click **OK** on the **Restore Options** dialog box.
    The following status message appears: `Restore initiated`.
16. Click **OK**.

Results

The backup restores as one or more files to the specified destination in the path *`destination\client\instance\database\file`*, where:

- *destination* is the destination for the files that you specified in the **Set Destination** dialog box.
- *client* is the name of the computer on which SQL Server is installed.
- *instance* is the name of the SQL Server instance from the backup.
- *database* is the name of the database from the backup.

- *file* is the name of the file.

A single backup can include multiple files, depending on the number of streams in the backup. The file name for each file is composed of the backup type and the stream number:

- `f-0` for full backups
- `d-n` for differential backups
- `i-n` for transaction log (incremental) backups

where *n* is the sequential number of the differential or incremental backup since the preceding full backup. For example, a full backup with two streams results in two files: `f-0.stream0` and `f-0.stream1`.

Next steps

- Ensure that the SQL backup format files that you restored are accessible to SQL Server. You may need to make the data visible to SQL Server or copy the data.
- Manually restore the database by using SQL Server tools.

Restoring to a file with the Windows File System plug-in

You can restore an instance or database to operating system files by using the Avamar Plug-in for the Windows File System.

Steps

1. Ensure that the environment meets the guidelines in Restore requirements on page 59.
2. In Avamar Administrator, click the **Backup & Restore** launcher link button.
   The **Backup, Restore and Manage** window appears.
3. Find the backup to restore:
   - Finding a backup by date on page 64
   - Finding a backup by content on page 66

   The backup to restore is selected in the **Backups** table.
4. Select **Actions** > **Restore Now**
   The **Restore Options** dialog box appears.
5. Select **Windows File System** from the **Restore Plug-in** list.
6. From the **Avamar encryption method** list, select the encryption method to use for data transfer between the Avamar server and the client during the restore.

   The encryption technology and bit strength for a client/server connection depend on several factors, including the client operating system and Avamar server version. The *Avamar Product Security Guide* provides additional information.
7. Specify the destination client in the **Restore Destination Client** box:
   - To restore to the original client, leave the default setting of the original client domain and name.
   - To restore to a different client, click **Browse** and then browse to the client.
8. Select **Restore everything to a different location**.
9. Set the destination file path for the database and log files to restore:
   a. Click **Set Destination**.
      The **Set Destination** dialog box appears.
   b. Click **Browse**.
      The **Browse for File, Folder, or Directory** dialog box appears.
   c. Select **Windows File System**.
   d. In the right pane, browse to and select the checkbox for the folder to which to restore the selected files.
   e. Click **OK** to return to the **Set Destination** dialog box.
   f. Click **OK** to return to the **Restore Options** dialog box.
10. (Optional) Click **More Options** and set plug-in options for the restore. The *Avamar for Windows Server User Guide* provides details on the available plug-in options.
11. Click **OK** on the **Restore Options** dialog box.
    The following status message appears: `Restore initiated`.
12. Click **OK**.

Results

The backup restores as one or more files to the specified destination in the path `destination\client\instance\database\file`, where:

- *destination* is the destination for the files that you specified in the **Set Destination** dialog box.
- *client* is the name of the computer on which SQL Server is installed.
- *instance* is the name of the SQL Server instance from the backup.
- *database* is the name of the database from the backup.
- *file* is the name of the file.

A single backup can include multiple files, depending on the number of streams in the backup. The file name for each file is composed of the backup type and the stream number:

- `f-0` for full backups
- `d-n` for differential backups
- `i-n` for transaction log (incremental) backups

where *n* is the sequential number of the differential or incremental backup since the preceding full backup. For example, a full backup with two streams results in two files: `f-0.stream0` and `f-0.stream1`.

Next steps

- Ensure that the SQL backup format files that you restored are accessible to SQL Server. You may need to make the data visible to SQL Server or copy the data.
- Manually restore the database by using SQL Server tools.

Restoring a database with SQL Server tools

After you use either the SQL Server plug-in or the Windows file system plug-in to restore a SQL Server backup to a file, you can use SQL Server tools to restore a database to SQL Server. Supported methods include SQL Server Management Studio, a Transact-SQL `RESTORE` command, and the `sqlcmd` utility.

Restoring a database with SQL Server Management Studio

You can restore a database from a SQL formatted backup file to SQL Server by using the user interface in SQL Server Management Studio. The Microsoft website provides full details on how to use SQL Server Management Studio to restore a database backup.

About this task

This procedure provides details on using SQL Server Management Studio for SQL Server 2008 to restore a database from SQL formatted backup files. The steps for other SQL Server versions may be different.

Steps

1. Restore the database backup to a file by using the instructions in one of the following topics:
   - Restoring to a file with the SQL Server plug-in on page 73
   - Restoring to a file with the Windows File System plug-in on page 74
2. Ensure that the SQL backup format files that you restored are accessible to SQL Server. You may need to make the data visible to SQL Server or copy the data.
3. Restore the full backup (`f-0` file) to SQL Server:
   a. Open the **Restore Database** window.
      - If the database already exists, then right-click the database in the **Object Explorer** and select **Tasks** > **Restore** > **Database**.
      - If the database has been lost, then right-click the **Databases** node in the **Object Explorer** and select **Restore Database**.
   b. On the **General** page of the **Restore Database** window, select **From device**.
   c. Click the **...** button.
      The **Specify Backup** dialog box appears.
   d. Click **Add**.
      The **Locate Backup File** dialog box appears.

e. Select the folder in which the full backup files are located.
f. From the **Files of type** list, select **All files(*)**.
g. Select the full backup (`f-0`) file.
h. Click **OK**.
i. If there are multiple full backup files from multi-streaming (such as `f-0.stream0`, `f-0.stream1`, `f-0.stream2`, and so on), then repeat step d through step h to add each file.
j. Click **OK** on the **Specify Backup** dialog box.
k. On the **General** page of the **Restore Database** window, select the checkboxes next to the backup files to restore.
l. In the left pane, click **Options** to open the **Options** page.
m. In the **Restore the database files as** list, select each file and click the **...** button to specify the location to which to restore the files.
n. For **Recovery state**, select **RESTORE WITH NORECOVERY**.
o. Click **OK** to begin the restore.

4. Restore the differential (`d-n`) or transaction log (`i-n`) files in order from the oldest to the most recent:
   a. In the **Object Explorer**, right-click the database and select **Tasks** > **Restore** > **Database**.
   b. On the **General** page of the **Restore Database** window, select **From device**.
   c. Click the **...** button.
   The **Specify Backup** dialog box appears.
   d. Click **Add**.
   The **Locate Backup File** dialog box appears.
   e. Select the folder in which the differential or transaction log backup files are located.
   f. From the **Files of type** list, select **All files(*)**.
   g. Select the differential (`d-n`) or transaction log (`i-n`) backup file, where *n* is the sequential number of the differential or incremental backup since the preceding full backup.
   h. Click **OK**.
   i. If there are multiple differential or transaction log backup files from multi-streaming (such as `d-3.stream0`, `d-3.stream1`, `d-3.stream2`, or `i-6.stream0`, `i-6.stream1`, `i-6.stream2`, and `i-6.stream3`), then repeat step d through step h to add each file.
   j. Click **OK** on the **Specify Backup** dialog box.
   k. On the **General** page of the **Restore Database** window, select the checkboxes next to the backup files to restore.
   l. In the left pane, click **Options** to open the **Options** page.
   m. In the **Restore the database files as** list, select each file and click the **...** button to specify the location to which to restore the files.
   n. For **Recovery state**, select **RESTORE WITH NORECOVERY** for all except the most recent backup file. When you restore the most recent backup file, select **RESTORE WITH RECOVERY**.
   o. Click **OK** to begin the restore.
5. If the database is not already listed in SQL Server Management Studio, then refresh the list or connect to the database.

Next steps

After the restore completes successfully, perform a full backup of the database and clear the **Force incremental backup after full backup** checkbox in the plug-in options for the backup. If the checkbox is selected when a full backup occurs after a restore, then the transaction log backup that occurs automatically after the full backup fails.

Restoring a database with a Transact-SQL RESTORE command

You can issue a Transact-SQL `RESTORE` command to restore a database from a SQL formatted backup file to SQL Server. The Microsoft website provides full details on how to issue a Transact-SQL command, including details on the available options for the `RESTORE` command.

Steps

1. Restore the database backup to a file by using the instructions in one of the following topics:
   - Restoring to a file with the SQL Server plug-in on page 73
   - Restoring to a file with the Windows File System plug-in on page 74
2. Ensure that the SQL backup format files that you restored are accessible to SQL Server. You may need to make the data visible to SQL Server or copy the data.
3. Restore the full backup (`f-0` file) to SQL Server by using a Transact-SQL command similar to the following command:

```
RESTORE DATABASE dbname
FROM DISK = 'drive:\folder\f-0.stream0'
WITH
MOVE 'dbname_data' TO 'drive:\folder\dbname.mdf',
MOVE 'dbname_logfile' TO 'drive:\folder\dbname.ldf',
NORECOVERY
```

where:

- *dbname* is the name of the database to restore.
- The `FROM DISK` clause specifies the backup file from which to restore, and *drive:\folder\f-0.stream0* is the path and file name of the full backup file.
- The `MOVE` clauses specify the path and file name for the restored files:
  - *dbname_data* is the name of the database data file in the backup.
  - *drive:\folder\dbname.mdf* is the path and file name for the restored database file.
  - *dbname_logfile* is the name of the database log file in the backup.
  - *drive:\folder\dbname.ldf* is the path and file name for the restored database log file.
- The `NORECOVERY` option specifies that the database will remain in the restoring state, which enables you to restore additional backups before setting the database online.

4. Restore all but the most recent differential (`d-n`) or transaction log (`i-n`) backup files, starting with the oldest file, by using a Transact-SQL command similar to the command in the previous step. However, replace the *f-0.stream0* file name in the `FROM DISK` clause with the file name for the differential or transaction log backup file.
5. Restore the most recent differential or transaction log backup file by using a Transact-SQL command similar to the following command:

```
RESTORE DATABASE dbname
FROM DISK = 'drive:\folder\i-7.stream0'
WITH
MOVE 'dbname_data' TO 'drive:\folder\dbname.mdf',
MOVE 'dbname_logfile' TO 'drive:\folder\dbname.ldf',
RECOVERY
```

where:

- *dbname* is the name of the database to restore.
- The `FROM DISK` clause specifies the backup file from which to restore, and *drive:\folder\i-7.stream0* is the path and file name of the most recent transaction log backup file.
- The `MOVE` clauses specify the path and file name for the restored files:
  - *dbname_data* is the name of the database data file in the backup.
  - *drive:\folder\dbname.mdf* is the path and file name for the restored database file.
  - *dbname_logfile* is the name of the database log file in the backup.
  - *drive:\folder\dbname.ldf* is the path and file name for the restored database log file.
- The `RECOVERY` option sets the database online after the restore.

6. If the database is not already listed in SQL Server Management Studio, then refresh the list or connect to the database.

Next steps

After the restore completes successfully, perform a full backup of the database and clear the **Force incremental backup after full backup** checkbox in the plug-in options for the backup. If the checkbox is selected when a full backup occurs after a restore, then the transaction log backup that occurs automatically after the full backup fails.

Restoring a database with the sqlcmd utility

The `sqlcmd` utility is a command line utility for entering Transact-SQL statements and scripts. You can restore a database from a backup file by specifying the Transact-SQL `RESTORE` statements with the `sqlcmd` utility. The Microsoft website provides full details on using the `sqlcmd` utility.

NOTE: You can use the Microsoft SQL Server `osql` utility, but Microsoft recommends the `sqlcmd` utility.

Restoring a single database from a single full backup file

```
sqlcmd -S server -E
1> restore database dbname
2> from disk = 'drive:\folder\f-0.stream0'
3> with recovery;
4> go
```

where:

- *server* is the server that is running SQL Server, and optionally, the instance, to which to restore the backup.
- *dbname* is the database to restore.
- *drive:\folder\f-0.stream0* is the path and file name of the backup file from which to restore the database.

Determining the number and names of files in the database to restore

```
sqlcmd -S server -E
1> restore filelistonly
2> from disk = 'drive:\folder\f-0.stream0'
3> go
1> restore database dbname
2> from disk = 'drive:\folder\f-0.stream0'
3> with norecovery,
4> move 'dbname_data' to 'drive:\dbname.mdf',
5> move 'dbname_log' to 'drive:\dbname.ldf'
6> go
```

Restoring a database to a point in time by using STOPAT syntax

```
RESTORE DATABASE dbname
FROM disk= 'drive:\folder\f-0.stream0'
WITH NORECOVERY, STOPAT = 'Apr 25, 2014 12:00 AM'
go
RESTORE LOG dbname
FROM disk= 'drive:\folder\i-1.stream0'
WITH RECOVERY, STOPAT = 'Apr 25, 2013 12:00 AM'
go
```

where:

- *dbname* is the database to restore.
- *drive:\folder\f-0.stream0* is the path and file name of the full backup file from which to restore the database.
- *drive:\folder\i-1.stream0* is the path and file name of the transaction log backup file from which to restore.

Performing a full backup after the restore

After the restore completes successfully, perform a full backup of the database and clear the **Force incremental backup after full backup** checkbox in the plug-in options for the backup. If the checkbox is selected when a full backup occurs after a restore, then the transaction log backup that occurs automatically after the full backup fails.

Restoring system databases

It is rare that you need to restore only system databases. One possible scenario for restoring only system databases is if damage occurs to one or more system databases. When you restore system databases, the Avamar Plug-in for SQL Server can automatically restore the databases in the correct order and manage SQL Server services. If necessary, however, you can restore individual system databases and manually manage the services.

About this task

NOTE: System databases in a replication environment include the publication, distribution, and subscription databases. The steps to restore these databases and other system databases in a SQL Server replication environment depend on the replication configuration. The SQL Server plug-in can automatically restore the databases in the correct order and manage SQL Server services. The steps to manually restore databases in a replication environment are beyond the scope of this

guide. Review the "Back Up and Restore Replicated Databases" topic in the SQL Server documentation on the MSDN website for detailed steps to manually restore system databases in a replication environment.

Restoring system databases automatically to the original location

When you restore multiple system databases to the original location, the Avamar Plug-in for SQL Server automatically restores the databases in the correct order. The SQL Server plug-in can also automatically manage the stop and restart of the necessary SQL Server services during the restore.

Prerequisites

- Ensure that the environment meets the guidelines in Restore requirements on page 59.
- Close all instances of SQL Server Management Studio, and disable any other possible connections to the system databases. If there are other connections to the system databases, then Avamar may not be able to restore the master database.

Steps

1. In Avamar Administrator, click the **Backup & Restore** launcher link button.
   The **Backup, Restore and Manage** window appears.
2. Find the backup and select the system databases to restore:
   - Finding a backup by date on page 64
   - Finding a backup by content on page 66
3. Select **Actions** > **Restore Now**.
   The **Restore Options** dialog box appears.
4. Ensure that **Windows SQL** appears in the **Restore Plug-in** list.
5. From the **Avamar encryption method** list, select the encryption method to use for data transfer between the Avamar server and the client during the restore.

   The encryption technology and bit strength for a client/server connection depend on several factors, including the client operating system and Avamar server version. The *Avamar Product Security Guide* provides additional information.
6. Leave the default selection of **Restore to the original location**.

   If you plan to restore the system databases to a file, follow the steps in Restoring to a file on page 72 instead of the steps in this procedure.
7. Click **More Options**.
   The **Restore Command Line Options** dialog box appears.
8. Select the **Use SQL REPLACE option** checkbox.
9. Clear the **Tail-log backup** checkbox.
10. From the **Encryption method from Data Domain system** list, select the encryption method for data transfer during the restore if the backup was stored on a Data Domain system.
11. Select the **Restore system databases** checkbox.
12. Select the **Manage SQL services automatically during restore** checkbox.
13. (Optional) Select the **Show Advanced Options** checkbox, and set other plug-in options as discussed in the following topics:
    - Recovery operation options on page 89
    - Authentication options on page 91
14. Disregard the redirected restore options, which are only necessary when you are restoring to a different location.
15. Disregard the point-in-time recovery options, which are only supported for databases that use the full recovery model.
16. Click **OK** on the **Restore Command Line Options** dialog box.
17. Click **OK** on the **Restore Options** dialog box.
    The following status message appears: `Restore initiated`.
18. Click **OK**.

Restoring system databases manually to the original location

When you restore system databases to the original location manually, you must manage the services and restore the databases in the correct order.

About this task

NOTE: The steps to restore system databases in a SQL Server replication environment depend on the replication configuration and are beyond the scope of this guide. Review the "Back Up and Restore Replicated Databases" topic in the SQL Server documentation on the MSDN website for detailed steps to manually restore system databases in a replication environment.

Steps

1. Shut down the SQL Server instance, and ensure that dependent services stop, such as the SQL Server Agent service and the Analysis Service.
2. Close all instances of SQL Server Management Studio, and disable any other possible connections to the system databases.

   If there are other connections to the system databases, then Avamar may not be able to restore the master database.
3. Start the SQL Server instance in single-user mode by running the `sqlservr.exe` application with the `-m` and `-c` options:
   - To start the default instance in single-user mode, open a command prompt and type the following command:

     ```
     cd \MSSQLPATH\Binn
     sqlservr.exe -m -c
     ```

   - To start a named instance in single-user mode, open a command prompt and type the following command:

     ```
     cd \MSSQLPATH\Binn
     sqlservr.exe instance -m -c
     ```

   where *\MSSQLPATH\Binn* is the path to the `Binn` folder for the instance, and *instance* is the name of the instance to start.
4. Wait for the `Recovery Complete` message to appear on the console.

   If you log in to the SQL Server computer as a local or domain administrator with the SQL services running under the Local System account, then the `sqlservr.exe` command may fail to correctly start SQL services in single-user mode. If that occurs, complete step 4.a on page 80 through step 4.e on page 80 instead of running `sqlservr.exe` from the command line. Otherwise, go to step 5 on page 80.

   Complete the following steps if SQL services do not start in single-user mode correctly:

   a. Shut down the SQL service. If SQL Server is installed on a standalone server, then use the Windows Services console. If SQL Server is installed in a cluster, use Cluster Manager.
   b. Right-click the SQL service in the Windows **Services** console, and then click **Properties**.
   c. In the **Start parameters** box, type **`-m -c`**.
   d. Click **Start** to start the service.
   e. Click **OK** to close the **Properties** dialog box.
5. Restore the master database to the original location:
   a. Ensure that the environment meets the guidelines in Restore requirements on page 59.
   b. In Avamar Administrator, click the **Backup & Restore** launcher link button.
      The **Backup, Restore and Manage** window appears.
   c. Find the backup and select the master database to restore:
      - Finding a backup by date on page 64
      - Finding a backup by content on page 66
   d. Select **Actions** > **Restore Now**
      The **Restore Options** dialog box appears.
   e. Ensure that **Windows SQL** appears in the **Restore Plug-in** list.
   f. From the **Avamar encryption method** list, select the encryption method to use for data transfer from the Avamar server to the client during the restore.
      The encryption technology and bit strength for a client/server connection depend on several factors, including the client operating system and Avamar server version. The *Avamar Product Security Guide* provides additional information.
   g. Leave the default selection of **Restore to the original location**.

h. Click **More Options**.
The **Restore Command Line Options** dialog box appears.
i. Select the checkbox next to the **Use SQL REPLACE option** and **Restore system databases** options.
j. Clear the **Tail-log backup** checkbox.
k. (Optional) Select **Show Advanced Options**, and set the authentication options as discussed in Authentication options on page 91.
l. Disregard the remaining restore options, which do not apply when you restore the master database.
m. Click **OK** on the **Restore Command Line Options** dialog box.
n. Click **OK** on the **Restore Options** dialog box.
The following status message appears: `Restore initiated.`
o. Click **OK**.

After you restore the master database, the SQL Server service is stopped automatically.

6. Restart the SQL Server service:
   - To start the default instance of SQL Server, open a command prompt and type **`net start MSSQLServer`**.
   - To start a named instance of SQL Server, open a command prompt and type **`net start MSSQL$`*`instance`***, where *instance* is the name of the instance.
7. Restore the msdb and model databases:
   a. Ensure that the environment meets the guidelines in Restore requirements on page 59.
   b. In Avamar Administrator, click the **Backup & Restore** launcher link button.
   The **Backup, Restore and Manage** window appears.
   c. Find the backup and select the msdb and model databases to restore:
      - Finding a backup by date on page 64
      - Finding a backup by content on page 66
   d. Select **Actions** > **Restore Now**
   The **Restore Options** dialog box appears.
   e. Ensure that **Windows SQL** appears in the **Restore Plug-in** list.
   f. From the **Avamar encryption method** list, select the encryption method to use for data transfer from the Avamar server to the client during the restore.
   The encryption technology and bit strength for a client/server connection depend on several factors, including the client operating system and Avamar server version. The *Avamar Product Security Guide* provides additional information.
   g. Leave the default selection of **Restore to the original location**.
   h. Click **More Options**.
   The **Restore Command Line Options** dialog box appears.
   i. Select the checkbox next to the **Use SQL REPLACE option** and **Restore system databases** options.
   j. Clear the **Tail-log backup** checkbox.
   k. (Optional) Select **Show Advanced Options**, and set the authentication options as discussed in Authentication options on page 91.
   l. Disregard the remaining restore options, which do not apply when you restore the msdb and model databases.
   m. Click **OK** on the **Restore Command Line Options** dialog box.
   n. Click **OK** on the **Restore Options** dialog box.
   The folowing status message appears: `Restore initiated.`
   o. Click **OK**.
8. If required, restart the SQL Server Agent service:
   - To start the default instance of SQL Server Agent, open a command prompt and type **`net start SQLSERVERAGENT`**.
   - To start a named instance of SQL Server, open a command prompt and type **`net start SQLAGENT$`*`instance`***, where *instance* is the name of the instance.

Restoring system databases to a different instance

When you restore system databases to a different instance, the restore process overwrites the system databases in the target instance. You must restore the master database before you restore other system databases. You can restore the model and msdb databases either separately or at the same time.

Prerequisites

- The target SQL Server installation for the restore must have the same SQL Server version and service pack as the SQL Server installation on which the backup occurred.
- The name of the target instance for the restore is the same as the original SQL Server instance for the backup.

- The user account and domain for authentication is the same on the target SQL Server instance as the original SQL Server instance for the backup.
- Close all instances of SQL Server Management Studio on the target SQL Server installation, and disable any other possible connections to the system databases. If there are other connections to the system databases, then Avamar may not be able to restore the master database.
- Ensure that there are no `avsql` or `avtar` processes running on the target SQL Server installation.

About this task

You must restore the master and model database files to the same file system path on the target server as the file system path on the original server for the backup. You can restore the msdb database files to a different file system location. To restore the msdb database files to a different file system location, restore the msdb database in a separate restore operation than the model database, and select the target file system location for the restored files.

To change the file system location of the master and model database files, restore the database files to the same file system location in the target instance, and then manually move the files by using the steps in the "Move System Databases" article on the MSDN website at http://msdn.microsoft.com/en-us/library/ms345408%28v=sql.110%29.aspx.

Steps

1. Restore the master database to the target instance:
   a. In Avamar Administrator, click the **Backup & Restore** launcher link button.
   The **Backup, Restore and Manage** window appears.
   b. Find the backup and select the master database to restore:
      - Finding a backup by date on page 64
      - Finding a backup by content on page 66
   c. Select **Actions** > **Restore Now**.
   The **Restore Options** dialog box appears.
   d. Ensure that **Windows SQL** appears in the **Restore Plug-in** list.
   e. From the **Avamar encryption method** list, select the encryption method to use for data transfer between the Avamar server and the client during the restore.
   f. Specify the destination client in the **Restore Destination Client** box, or click **Browse** to browse to the client.
   g. Select **Restore to a different SQL Server instance or location**.
   h. Next to the **Select the SQL instance on the destination client where the items should be restored** box, click **Browse**, select **Windows SQL**, and then select the checkbox next to the target instance.
   i. Click **OK**.
   j. On the **Restore Options** dialog box, click **More Options**.
   The **Restore Command Line Options** dialog box appears.
   k. Select the **Show Advanced Options** checkbox.
   Several advanced options appear in red.
   l. Specify the options as described in the following table.

   Table 6. Advanced options

Option	Selection
Use SQL REPLACE option	Select the checkbox.
Tail-log backup	Clear the checkbox to disable tail-log backup. Tail-log backup is not supported when you restore to a different instance.
Encryption method from Data Domain system	Select the encryption method for data transfer during the restore if the backup was stored on a Data Domain system.
Restore system databases	Select the checkbox.
Manage SQL services automatically during restore	Select the checkbox.

   m. Disregard the remaining options.
   n. Click **OK** on the **Restore Command Line Options** dialog box.
   o. Click **OK** on the **Restore Options** dialog box.
   The following status message appears: `Restore initiated.`

p. Click **OK**.

2. Restore the msdb database to the target instance by using the same steps and options that you used to restore the master database.

   NOTE: You can restore both the msdb and model databases at the same time if you restore them to the same file system path on the target server as the path on the original server. If you want to restore the msdb database files to a different file system path on the target server, then restore the msdb and model databases with separate restore operations, and specify the file system location for the msdb database files by clicking **Set Destination** on the **Restore Options** dialog box and selecting the location.

3. If you did not restore the model database in the previous step, then restore the model database by using the same steps and options that you used to restore the master and msdb databases.

Restoring to an AlwaysOn availability group

When you back up databases in an AlwaysOn availability group, you can restore the databases to the following locations:

- The original location in the original availability group
- A new database in the original availability group
- A different availability group
- A SQL Server instance that does not use availability groups
- Operating system files

In a hybrid IT environment with AlwaysOn availability groups and Microsoft Azure, the steps to restore a database depend on the location of the primary replica:

- If the primary replica is on-premise, restore data to only the primary replica. Do not attempt to restore to both the primary and secondary replicas. If there is a corresponding database on the secondary replicas when you restore a database only on the primary replica, then the database on the secondary replicas is in a restoring state. To restore the databases on the secondary replicas as part of the availability group, manually prepare and restore the databases, and join them to the availability group on the secondary replica.
- If the primary replica is on a Microsoft Azure virtual machine, then restore the database to operating system files and then use SQL Server tools to restore the database to the availability group.

Restoring to the original availability group

When you restore to the original location in an AlwaysOn availability group, the restore process can automatically restore the databases on both the primary replica and the secondary replicas. You can also restore databases only on the primary replica.

Restoring automatically to both primary and secondary replicas

About this task

To automatically restore the databases to both the primary replica and the secondary replicas, perform the steps in Restoring to the original location on page 69. During the restore, set the following values:

- Select the client for the availability group listener when you search for a backup to restore.
- Set the plug-in options with the following values:
  - Clear the **Restore only on primary replica** checkbox.
  - Leave the **System Databases** options clear. SQL Server does not support system databases in an availability group.
  - (Optional) Perform a tail-log backup or use SQL `REPLACE` to force a restore even if the database already exists.
  - Disregard the **Redirected Restore** options.
  - (Optional) Specify authentication information.
  - (Optional) Perform a point-in-time recovery.

Restoring to only the primary replica

Steps

1. Perform the steps in Restoring to the original location on page 69. During the restore, set the following values:

- Select the client for the availability group listener when you search for a backup to restore.
- Leave the **Restore only on primary replica** checkbox selected in the plug-in options.
- Leave the **System Databases** options clear. SQL Server does not support system databases in an availability group.
- (Optional) Perform a tail-log backup or use SQL `REPLACE` to force a restore even if the database already exists.
- Disregard the **Redirected Restore** options.
- (Optional) Specify authentication information.
- (Optional) Perform a point-in-time recovery.

After you restore a database on only the primary replica, the corresponding database on the secondary replicas is in a restoring state.

2. (Optional) Restore the databases on the secondary replicas by manually preparing and restoring the databases, and then joining them to the availability group on the secondary replicas.

Restoring to a new database in the original availability group

You can restore a database in an AlwaysOn availability group to its original instance but with a new name, creating a new database in the instance.

About this task

To automatically restore the databases to both the primary replica and the secondary replicas, perform the steps in Restoring to a new database in the original instance on page 70. During the restore, set the following values:

- Select the client for the availability group listener when you search for a backup to restore.
- Set the plug-in options with the following values:
  - Leave the **Use SQL REPLACE option** checkbox clear.
  - (Optional) Perform a tail-log backup.
  - Clear the **Restore only on primary replica** checkbox.
  - (Optional) Specify recovery operation options.
  - Leave the **System Databases** options clear. SQL Server does not support system databases in an availability group.
  - Specify a name for the new database in the **New database name** box.
  - (Optional) Specify a new path for the database files in the **Alternate database location** box.
  - (Optional) Select the location for the restored database log files.
  - (Optional) Specify authentication information.
  - (Optional) Perform a point-in-time recovery.

Restoring to a different availability group

You can restore databases from one availability group to an availability group in a different cluster.

Steps

1. Perform the steps in Restoring to a different instance on page 71. During the restore, set the following values:
   - Select the client for the original availability group listener when you search for a backup to restore.
   - Leave the **Use SQL REPLACE** option checkbox clear in the plug-in options.
   - Clear the **Tail-log backup** checkbox. This restore scenario does not support tail-log backup.
   - Leave the **Restore only on primary replica** checkbox selected.
   - (Optional) Specify recovery operation options.
   - Leave the **System Databases** options clear. SQL Server does not support system databases in an availability group.
   - Disregard the **Redirected Restore** options. You already specified these settings on the **Set Destination** dialog box.
   - (Optional) Specify authentication information.
   - (Optional) Perform a point-in-time recovery.
2. After the restore completes, join the new database to the availability group.

   When you add the database to the availability group, the process automatically creates and synchronizes the databases on the secondary replicas.

Restoring to a SQL Server instance without availability groups

When you back up databases in an availability group, you can restore the databases to a different SQL Server instance that does not use availability groups, either on the same server or on a different server.

About this task

The steps are the same whether you restore from a backup of databases on a stand-alone server, on shared storage in a failover cluster, or in an availability group. Restoring to a different instance on page 71 provides details.

Restoring to operating system files

When you back up databases in an availability group, you can restore the databases to operating system files.

About this task

The steps are the same whether you restore from a backup of databases on a stand-alone server, on shared storage in a failover cluster, or in an availability group. Restoring to a file on page 72 provides details.

Restoring a database with an intact log file

If a database becomes corrupt or is otherwise lost but an intact database log file is available, you can restore the database and use the log file to recover transactions after the most recent backup.

Steps

1. Perform a transaction log backup of the intact database log by issuing the following Transact-SQL command:

   BACKUP LOG *dbname* TO DISK = '*drive:\folder\filename*' WITH NO_TRUNCATE

   where *dbname* is the name of the database and *drive:\folder\file* is the path to the folder and file name to which to save the backup.

2. In Avamar Administrator, find the most recent backup of the database by using the instructions in Finding a backup by content on page 66.
3. In Avamar Administrator, restore the backup to its original location by using the instructions in Restoring to the original location on page 69.

   On the **Restore Command Line Options** dialog box during the restore, select the **Show Advanced Options** checkbox, and then configure the settings as follows:

   - Select the **Use SQL REPLACE option** checkbox.
   - Clear the **Tail-log backup** checkbox.
   - (Optional) If the backup is stored on a Data Domain system, select the encryption method for data transfer from the Data Domain system to the client during the restore by using the **Encryption method from Data Domain system** list.
   - Select **NORECOVERY** from the **Recovery operation** list.
   - Disregard the remaining options, which do not apply to this restore.

4. Restore the transaction log backup of the intact database log by issuing the following Transact-SQL command:

   RESTORE LOG *dbname* FROM DISK = '*drive:\folder\filename*' WITH RECOVERY

   where *dbname* is the name of the database and *drive:\folder\file* is the path and file name for the transaction log backup file.

5. If the database does not appear in SQL Server Management Studio, then refresh the list or connect to the database.

Next steps

After the restore completes successfully, perform a full backup of the database and clear the **Force incremental backup after full backup** checkbox in the plug-in options for the backup. If the checkbox is selected when a full backup occurs after a restore, then the transaction log backup that occurs automatically after the full backup fails.

Setting restore options

You set plug-in options during the restore by using the **Restore Command Line Options** dialog box.

To view all plug-in options, including advanced options, select the **Show Advanced Options** checkbox. The advanced options appear in red.

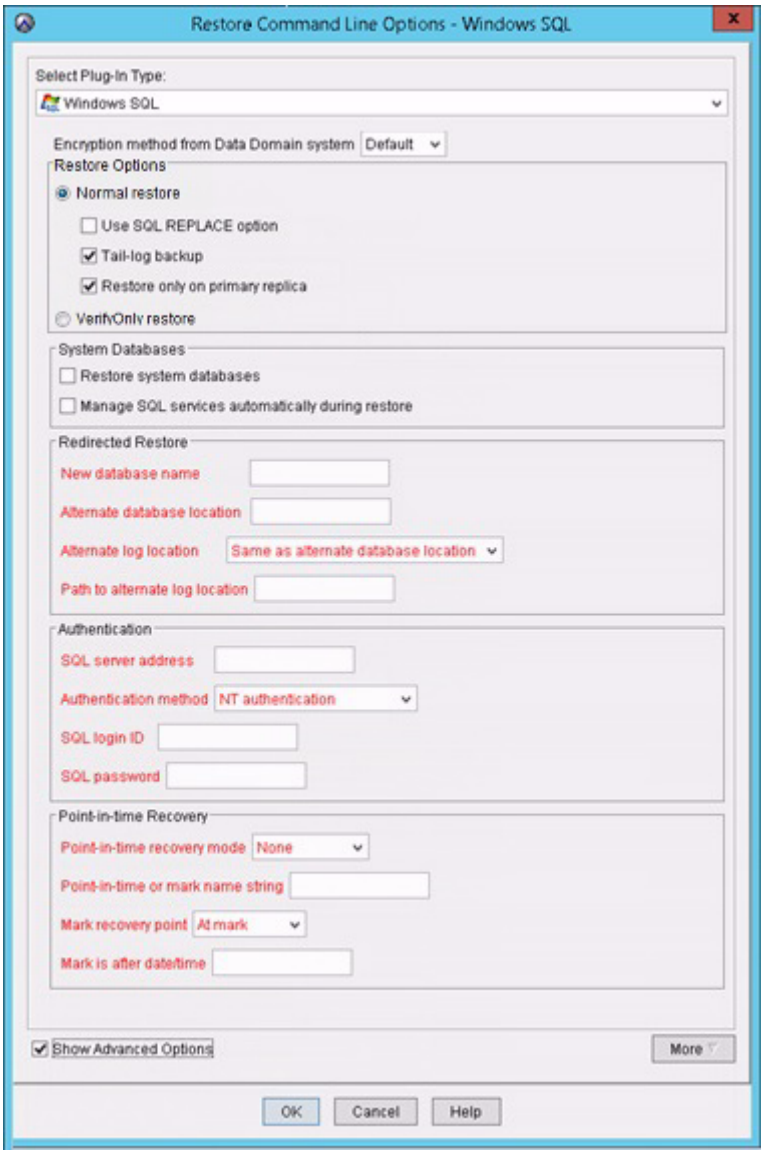

Figure 11. Restore Command Line Options dialog box

General restore options

General options on the **Restore Command Line Options** dialog box include the **Encyrption method from Data Domain system** option, **Normal restore** selection, which also allows you to select the **Use SQL REPLACE** option checkbox, the **Tail-log backup** checkbox, and the **Restore only on primary replica** checkbox, **VerifyOnly** selection, and the **Enable debugging messages** checkbox.

Encryption method from Data Domain system

If the backup is stored on a Data Domain system, the **Encryption method from Data Domain system** option specifies the encryption method to use for data transfer between the Data Domain system and the client during the restore.

Normal restore

Performs a normal restore of the databases. Selecting this option allows you to also select the **Use SQL REPLACE** option checkbox, the **Tail-log backup** checkbox, and the **Restore only on primary replica** checkbox.

Use SQL REPLACE option

When you select the **Use SQL REPLACE option** checkbox on the **Restore Command Line Options** dialog box, an SQL `WITH REPLACE` clause statement is added to the restore Transact-SQL command. The statement specifies that SQL Server will create any necessary database and related files even if another database or file exists with the same name.

NOTE: This option overrides a SQL Server safety check to prevent you from accidentally overwriting a different database or file. The Microsoft Transact-SQL documentation provides more information in the `RESTORE` command section.

Select the checkbox only when you are restoring an instance, database, filegroup, or file to its original location and you need to force the overwrite of the original data. You may need to force the overwrite if a previous database restore exited with the following SQL Server error in the Avamar SQL restore log:

```
One or more devices or files already exist. Reissue the statement using the WITH REPLACE
option to overwrite these files and devices.
```

Use this option also when you restore system databases.

Tail-log backup

When you select the **Tail-log backup** checkbox on the **Restore Command Line Options** dialog box, Avamar backs up the tail of the transaction log during the restore to capture the log records that are not in a backup. The restore process then uses the tail-log backup after the database restore to recover the transactions that were not in the backup.

To perform a tail-log backup, the database must be online and using either the full or bulk-logged recovery model. As a result, you cannot perform a tail-log backup of system databases such as the master and msdb databases because those databases use the simple recovery model.

When you are restoring a user-defined filegroup or secondary data file and you perform a tail-log backup, you must select the most recent backup as the backup from which to restore. Otherwise, the restore fails and an error message is written to the log file.

You can perform a tail-log backup when you are restoring an instance, database, filegroup, or file to its original location without the SQL `WITH REPLACE` option. In other words, if you select the **Use SQL REPLACE option** checkbox when you are restoring to the original location, then leave the **Tail-log backup** checkbox clear.

You can also perform a tail-log backup when you are restoring a database to the original instance but with a new database name.

If you are performing a point-in-time restore and the point in time to which you are restoring is after the most recent transaction log backup, then you must perform a tail-log backup.

A tail-log backup is also required if you are restoring a file from a user-defined filegroup to its original location.

Do not perform a tail-log backup if you are performing a redirected restore to a different SQL Server instance.

NOTE: If the tail-log backup fails to complete, then the restore cannot take place. Review the log file to determine the cause of the failure. Correct the problem, and then restart the restore. If you clear the **Tail-log backup** checkbox to prevent the tail-log backup, then the restore includes only the transactions up to the selected backup. You may lose any transactions in the tail of the log.

Restore only on primary replica

The **Restore only on primary replica** checkbox controls whether databases restore to only the primary replica or to the primary replica and all secondary replicas in an AlwaysOn availability group.

When you clear the checkbox, the restore process automatically restores the databases to the original location on both the primary replica and the secondary replicas. Clear the checkbox only when you restore to the original location in the original availability group and you want to restore to all replicas.

When you select the checkbox and restore the databases on only the primary replica, the corresponding database on the secondary replicas is in a restoring state. To restore the databases on the secondary replicas as part of the availability group, you can manually prepare and restore the databases, and then join them to the availability group on the secondary replica.

You can also set the database on a secondary replica online without rejoining it to the availability group by restoring the database with the `RECOVERY` recovery operation. The SQL Server documentation available on the Microsoft website provides details.

Select the checkbox in the following restore scenarios:

- You are restoring to a different location in the original availability group.
- You are restoring to a different availability group.
- You are restoring to an on-premise primary replica in a hybrid IT environment with Microsoft Azure.

You can also select the checkbox if you want to restore to only the primary replica in the original availability group.

Individual restore of an incremental or differential backup

This feature is supported for standalone, cluster, and AG environments configured with single or multiple SQL databases.

Overwrite default recoveryplan checkbox in the Avamar Administrator.

Using the Avamar Administrator

To restore individual incremental or differential backup streams:

1. Ensure that the environment meets the guidelines in Restore requirements on page 59.
2. In Avamar Administrator, click the **Backup & Restore** launcher link button.

   The **Backup, Restore and Manage** window appears.
3. Find the backup to restore:
   - Finding a backup by date on page 64
   - Finding a backup by content on page 66

   The backup to restore is selected in the **Backups** table.
4. Select **Actions** > **Restore Now**.

   The **Restore Options** dialog box appears.
5. Ensure that **Windows SQL** appears in the **Restore Plug-in** list.
6. In the **Recovery operation** field, select one of the following options:
   - To determine the correct checkpoint up to restore during a restore operation, select **WITH NO Recovery**.
   - To determine the correct checkpoint up to restore and view and read the data that recovered during a restore operation, select **WITH STANDBY**.
   - To place the database in normal mode and restore the individual backup stream, select **WITH Recovery**.
   - To restore the last checkpoint and restore the individual backup stream, select **WITH Recovery**
7. If the **WITH STANDBY** option was selected, in the **Standby file location** field, type the file name location.
8. Select the **Overwrite defaul recovery plan** checkbox.
9. Click **OK**.

Using the CLI

To restore individual incremental or differential backup streams, use the `--overwrite-default-recoveryplan` flag.

When restoring individual streams of incremental or differential backups, consider the following options:

- To determine the correct checkpoint up to restore during a restore operation, use the `WITH NORECOVERY` flag.
- To determine the correct checkpoint up to restore and view and read the data that recovered during a restore operation, use the `WITH STANDBY` flag.
- To place the database in normal mode, restore the individual backup stream by using the `WITH RECOVERY` flag.
- To restore the last checkpoint, use the restore the individual backup stream by using the `WITH RECOVERY` flag.

AG Considerations

For AG environments, individual restore operations can only be performed on the primary node.

If the AG database is removed from AG while performing an individual restore operation:

1. On the primary node, search for the required checkpoint by performing an incremental or differential restore.
2. Manually join the database to the AG.

   You can use either of the following methods:

   - Use the SQL server.
   - Perform a restore operation in the Avamar Administrator:
     a. In the **Recovery operation** field, select **RECOVERY**.
     b. Select the **Overwrite defaul recovery plan** checkbox.
     c. Disable the **Restore primary only** checkbox.
     d. Click **OK**.

VerifyOnly restore

This options performs a verify-only restore, which verifies the backup but does not restore it.

Enable debugging messages

When you select the **Enable debugging messages** option, maximum information is written to log files during the operation. If you select the checkbox, the operation generates very large log files. Use this option only for debugging purposes.

Recovery operation options

The **Recovery operation** list and **Standby file location** box on the **Restore Command Line Options** dialog box enable you to control the recovery operation that occurs after the restore.

Table 7. Recovery operation options

Recovery operation	Description
`RECOVERY`	The database is fully recovered and online after the restore. This feature is the default setting.
`NORECOVERY`	The database remains in a restoring state after the restore. This feature enables you to perform additional manual restore tasks, such as applying additional SQL transaction log files.
`STANDBY`	The database is in standby (read-only) mode after the restore. This feature enables you to bring up a database for read-only access between transaction log restores, and can be used with either warm standby server situations or special recovery situations in which it is useful to inspect the database between log restores. This option also creates a file with recovery changes. If required, you can use the file to revert the recovery changes. The size of the file depends on the volume of undo actions from uncommitted transactions. Specify the path to this file in the Standby file location box. Use the following format: *`drive:\folder\subfolder`* where *drive* is the drive letter on the destination client, and *folder\subfolder* is the path on the drive letter in which to create the file. If a file with recovery changes exists in the specified location, then SQL Server overwrites it. If you do not specify a path, then Avamar creates the file in the *`C:\Program Files\avs\var`* folder, where *C:\Program Files\avs* is the Avamar installation folder.

Table 7. Recovery operation options (continued)

Recovery operation	Description
	Do not use STANDBY when a database upgrade is necessary. You may need to perform a database upgrade when restoring backup sets from an earlier version of SQL Server.

You can specify the recovery operation in the following restore scenarios:

- You are restoring an instance, database, filegroup, or file to its original location.
- You are restoring a database to the original instance but with a new database name.
- You are restoring an instance, database, filegroup, or file to a different instance on either the original server or a different server.
- You are restoring one or more system databases. Keep in mind the following points, however:
  - If you are restoring the master or model database, then you must select the RECOVERY option. Do not use either the NORECOVERY or STANDBY options.
  - If you are restoring the msdb database, then you can select any of the recovery operation options. However, if you select NORECOVERY or STANDBY, then all databases become inaccessible until the restore of the msdb database is complete.
  - If you are restoring the system databases, then you must select the RECOVERY option. Do not use either the NORECOVERY or STANDBY options.

You do not need to specify the recovery operation when you are restoring to an operating system file.

If you specify either the NORECOVERY or STANDBY recovery operation, then you can bring the database back online after any manual operations by using one of the following methods:

- Restore the database again by using the Avamar Plug-in for SQL Server, but select RECOVERY from the **Recovery operation** list.
- Issue a **RESTORE DATABASE *dbname* WITH RECOVERY** command by using SQL Server Management Studio, where *dbname* is the name of the database to restore.

System database restore options

The **Restore system databases** and **Manage SQL services automatically during restore** checkboxes on the **Restore Command Line Options** dialog box enable you to correctly restore system databases, such as the master, msdb, and model databases.

When you restore an entire instance, or if you specifically select system databases for restore, then select the **Restore system databases** checkbox to ensure that the system databases restore. If you leave the checkbox clear, then the system databases do not restore.

The **Manage SQL services automatically during restore** option automatically stops and restarts SQL services during the restore:

- When you restore the master database, this option automatically stops the SQL Server instance, including dependent services such as the SQL Server agent service and the Analysis Service. The option also restarts the instance in single-user mode before the restore. After the restore, the instance automatically restarts.
- When you restore the msdb database, this option automatically stops the SQL Server agent service, and then restarts it when the restore is complete.

When you select both system and user databases for restore, the system databases restore first. You must select the **Manage SQL services automatically during restore** checkbox to ensure that all system databases restore in the proper order and with the necessary service stops and restarts.

Redirected restore options

The redirected restore options on the **Restore Command Line Options** dialog box enable you to control the database name and file locations when you restore a database to the original instance but with a new name.

About this task

You also can specify the file locations by clicking **Set Destination** on the **Restore Options** dialog box. If you specify the file locations in the **Restore Options** dialog box, then you do not need to specify them on the **Restore Command Line Options** dialog box.

You can also specify a new database name when you are restoring a database to a different instance but with a new name.

Steps

1. In the **Restore Command Line Options** dialog box, select the **Show Advanced Options** checkbox.
   Several advanced options, including the point-in-time recovery options, appear in red.
2. (Optional) If you are restoring a single database with a new name, specify the new name in the **New Database** name box.
3. To restore the database files to a different path than the original path, type the full path of the new database file (`*.mdf`) location on the client in the **Alternate database location** box. An example path is `C:\temp`.
4. Select the path to which to restore the database log file (`*.ldf`):
   - To restore the database log file to the same location as the database files (as specified in the **Alternate database location** box), select **Same as alternate database location** from the **Alternate log location** list.
   - To restore the database log file to a different location than the database files, select **Different location** from the **Alternate log location** list, and then type the path to the location on the client in the **Path to alternate log location** box. An example path is `C:\temp\logs`.

Authentication options

Authentication options enable you to specify whether Avamar uses Windows (NT) authentication or SQL Server authentication to connect to SQL Server when you restore an instance, database, filegroup, or file to either its original location or to a different location.

About this task

If you do not specify an authentication method, then the SQL Server plug-in uses NT authentication and logs in with the NT AUTHORITY\SYSTEM account.

You do not need to specify authentication options when you are restoring to an operating system file because it is not necessary to connect to SQL Server during this type of restore.

Steps

1. In the **Restore Command Line Options** dialog box, select the **Show Advanced Options** checkbox.
   Several advanced options, including the authentication options, appear in red.
2. In the **SQL server address** box, specify the hostname or IP address of the server to connect to.
3. From the **Authentication method** list, select whether to use NT authentication or SQL Server authentication.
4. If you select SQL Server authentication, specify the login ID and password for the SQL Server account in the **SQL login ID** and **SQL password** boxes, respectively.

Point-in-time recovery options

If you are restoring a database that uses the full recovery model to either its original location or to a different location, then you can restore to either a specific date and time or to a named mark in the transaction log.

About this task

You cannot perform a point-in-time restore of system databases such as the master and msdb databases because those databases use the simple recovery model.

To restore to a specific point in time, you must provide the transaction date and time or named mark to which to recover from the SQL Server transaction log. The SQL Server documentation on the Microsoft website provides details on how to access transaction log information.

The point in time to which you are restoring must be after the finish time for the most recent full backup. In addition, if the point in time is before the start time of the most recent transaction log (incremental) backup, then a tail-log backup is not required. However, a tail-log backup is required if the point in time is after the most recent transaction log backup.

When you specify the point in time for restore, do not specify the start time of the selected transaction log backup if it is not the last backup in the backup sequence. Otherwise, the restore fails and a tail-log backup does not occur even if you select the Tail-log backup option.

Steps

1. In the **Restore Command Line Options** dialog box, select the **Show Advanced Options** checkbox.

   Several advanced options, including the point-in-time recovery options, appear in red.
2. From the **Point-in-time recovery mode** list, select whether to recover to a point in time or a named mark:
3. In the **Point-in-time or mark name string** box, specify either the point in time or the named mark to which to recover:
   - To recover to a specific point in time, specify the date and time in yyyy-mm-ddThh:mm:ss format. For example, 2013-10-15T14:15:45 is October 15, 2013 at 2:15:45 p.m.
   - To recover to a named mark, specify the mark.
4. If you specified a mark, select whether to include the mark in the recovery:
   - To specify that the log record immediately before the mark is the recovery point, select **Before mark** from the **Mark recovery point** list. In other words, the recovery rolls forward to the mark and excludes the marked transaction.
   - To specify that the marked transaction is the recovery point, select **At mark** from the **Mark recovery point** list. In other words, the recovery rolls forward to the mark and includes the marked transaction.
5. If you specified a mark and named marks are not unique in the transaction log, then use the **Mark is after date/time** box to locate the mark to which to recover. The recovery process stops at the first mark with the specified name, exactly at or after the specified date and time. Specify the date and time in yyyy-mm-ddThh:mm:ss format.

Performing table level recovery

You can restore individual tables from a backup of an SQL database by performing a table level recovery in conjunction with ItemPoint for Microsoft SQL Server.

Steps

1. In Avamar Administrator, click the **Backup & Restore** launcher link button.
   The **Backup, Restore, and Manage** window appears.
2. Find a full backup that contains the items to restore:
   - Finding a backup by date on page 64
   - Finding a backup by content on page 66

   While selecting differential backups to perform table level recovery, select only the full backup and the required differential backup. Selecting multiple differential backups is not supported.
3. Select the **Browse for Granular Restore** icon in the lower-left pane of the **Restore** tab.
   The **Select Destination Client** dialog box appears.
4. Select the target machine onto which the database will be mounted for TLR.
   - To mount the database on the original machine, select **Restore everything to its original location**.
   - To mount the database on another machine, select **Restore everything to a different location**.

   NOTE: TLR can be performed on physical nodes only. In the case of a failover of a cluster or an AlwaysOn availability group, restore must be redirected to a physical nodes that has TLR installed.
5. Click **OK.**
   A message will indicate that the mounting operation may take a long time to complete.
6. Click **OK** to proceed with the mount operation.
   The **Restore Browse Options** dialog will open.
7. At the **Restore Browse Options** dialog, identify the following:
   a. For **Drive letter or mount path**, enter the drive letter that will be used to mount the SQL backup data or enter a forward slash (**/**) or backslash (****) to mount the drive using the first available drive letter, beginning with **z:**.

This option is not case sensitive. If the letter entered here has a drive already mounted on it and that drive is not controlled by the Avamar Plug-in for SQL TLR, the mount operation will fail. If the Avamar Plug-in for SQL TLR has already mounted a drive on this letter, the existing drive will be unmounted and the new one will be mounted.

b. For **Amount of time to leave AvFS mounted**, select when to automatically dismount the drive. If the amount of time needs to be extended during the recovery, use the `avsqltlr` command.

8. On the target machine, open ItemPoint for Microsoft SQL Server from the Microsoft Windows **Start** menu:
   a. Select **Programs**
   b. Select the `EMC ItemPoint for Microsoft SQL Server` folder.
   c. Select the `EMC ItemPoint for Microsoft SQL Server` program.

   Following instructions in the ItemPoint for Microsoft SQL Server documentation to perform a table level recovery.

Restore only on primary replica

The **Restore only on primary replica** checkbox controls whether databases restore to only the primary replica or to the primary replica and all secondary replicas in an AlwaysOn availability group.

When you clear the checkbox, the restore process automatically restores the databases to the original location on both the primary replica and the secondary replicas. Clear the checkbox only when you restore to the original location in the original availability group and you want to restore to all replicas.

When you select the checkbox and restore the databases on only the primary replica, the corresponding database on the secondary replicas is in a restoring state. To restore the databases on the secondary replicas as part of the availability group, you can manually prepare and restore the databases, and then join them to the availability group on the secondary replica.

You can also set the database on a secondary replica online without rejoining it to the availability group by restoring the database with the `RECOVERY` recovery operation. The SQL Server documentation available on the Microsoft website provides details.

Select the checkbox in the following restore scenarios:

- You are restoring to a different location in the original availability group.
- You are restoring to a different availability group.
- You are restoring to an on-premise primary replica in a hybrid IT environment with Microsoft Azure.

You can also select the checkbox if you want to restore to only the primary replica in the original availability group.

Table Level restore

The SQL plug-in supports table level restore (TLR) of full, transaction log, and differential backups.

Steps

1. In the AUI navigation pane on the left, click ≫, and then click **Asset Management**.
   The **Asset Management** window appears.
2. In the **Domain** pane, select **clients**, and then select the desired client.
3. (Optional) To locate backups by date:
   a. In the right pane, click **VIEW MORE**.
   b. Click **SEARCH**.
   c. In the **From** and **To** fields, specify the date range.
   d. Click **RETRIEVE**.
   e. In the list of backups, select a backup.

   The list of backups for the date range is displayed.
4. In the **Backup Content** pane, move the **TLR** slider to the right.
5. In the left pane, select the drive letter on which to mount the backups, and then select the database in the right pane. Click **NEXT**.

   NOTE: TLR can be performed on physical nodes only. In the case of a failover of a cluster or an AlwaysOn availability group, restore must beredirected to a physical node that has TLR installed.

6. In the **More Options** pane:

a. For **Drive letter or mount path**, enter the drive letter that will be used to mount the SQL backup data or enter a forward slash (/) or backslash (\) to mount the drive using the first available drive letter, beginning with `Z:`.

   This option is not case sensitive. If there is already a drive mounted on the letter specified, and that drive is not controlled by the Avamar Plug-in for SQL TLR, the mount operation will fail. If the Avamar Plug-in for SQL TLR has already mounted a drive on this letter, the existing drive will be unmounted and the new one will be mounted.

b. For **Amount of time to leave AvFS mounted**, select when to automatically dismount the drive. If the amount of time needs to be extended during the recovery, use the `avsqltlr` command.

c. Move the **Show Advanced options** slider to the right, and then select the **Enable debugging messages** check box in order to obtain detailed logs.

7. Click **NEXT**, and then click **FINISH**.
   The mount operation might take several minutes to complete. When the mount is complete, the message `Retrieving backup contents of client completed. The mount disk avaliable now on target client` appears.
8. On the SQL server, open **Itempoint for MS SQL Server**.
9. For **Source Type**, select `Folder`, and then select the drive on which the backups are mounted.
10. Click **NEXT**.
11. Select the files that you want to restore:
    - To restore full or incremental backups, select all files.
    - To restore differential backups, remove the incremental files (\i-n files).

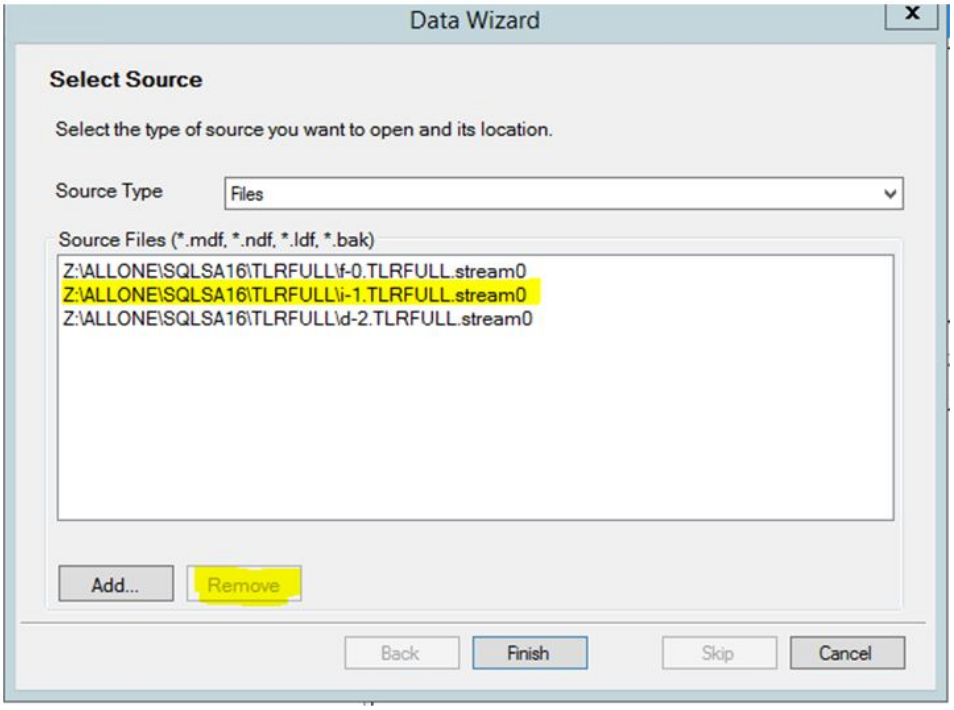

Figure 12. Select Source Data Wizard for Differential Restore

12. Click **FINISH**.
13. Select the target SQL instance.
14. Drag and drop, or copy and paste, the tables from the source database to the target database.
15. Close **Itempoint for MS SQL Server**.

Monitor restores

You can monitor and view status information for backup and restore operations in the **Activity Monitor**.

About this task

To access the **Activity Monitor**, open the navigation pane, and then click **Activity**. The **Activity Monitor** appears with a list of all activities.

NOTE: The AUI **Activity Monitor** window has been optimized for at least 1366 pixels-wide screens. Display issues might occur for smaller screens. To properly display the AUI, ensure that your display is at least 1366 pixels wide.

The **Activity Monitor** provides you with options to filter the information that appears:

- Filter activities by duration—By default, the **Activity Monitor** displays the most recent 5,000 client activities. To select a different duration, in the **Filter activities by duration** drop-down list, select **Last 24 hours** or **Last 72 hours**.
- Filter activities by domain—By default, the **Activity Monitor** displays all activities regardless of domain. To display only the activities for a specific domain, in the **Filter activities by domain** drop-down list, select a domain or subdomain.
- Filter activities by status—By default, the **Activity Monitor** displays all activities regardless of status.

  To display only activities with a specific status, at the top of the **Activity Monitor**, select one of the following options:

  - Canceled
  - Completed
  - Completed with Exceptions
  - Failed
  - Running
  - Waiting

To filter activities by client, start time, plug-in, or type, click in their respective column.

The **Activity Monitor** displays the date and time that an activity began, and the total number of bytes examined during an activity.

To view activity details, expand the **Details** pane, by clicking .

Cancel restores

You can cancel a restore any time before it completes. The cancellation might take 5 minutes or longer. The restore might complete before the cancellation finishes.

Steps

1. In the AUI navigation pane on the left, click , and then click **Activity**.
   The **Activity Monitor** appears with a list of activities.
2. Select the restore from the list.
3. Click **CANCEL**.

   A confirmation dialog box is displayed.
4. Click **YES**.

6

Disaster Recovery

Topics:

- Preparing for disaster recovery
- Performing disaster recovery

Preparing for disaster recovery

Steps

1. Ensure that you have onsite and offsite copies of the installation disks for the operating system, SQL Server, and any software updates.
2. Perform full Windows server backups by using the Avamar Client for Windows. Include System State and all critical and noncritical disks in the backups. The *Avamar for Windows Server User Guide* provides full backup instructions.
3. Back up all system and user databases by using the Avamar Plug-in for SQL Server.

   If you use AlwaysOn availability groups, then you must perform a separate backup of the system databases. During the backup, select the server name of the physical node as the client to back up, and then select the system databases. SQL Server does not support system databases in an availability group.

   NOTE: If you perform the full Windows server backup and the database backups at different times, then perform the full Windows server backup first.

Performing disaster recovery

Steps

1. Restore the server from the Windows client backup as described in the *Avamar for Windows Server User Guide*.
2. Complete the recommended Microsoft SQL Server disaster recovery procedure by using the steps in the “Recovering from a Disaster” article on the MSDN website.
3. Use the SQL Server plug-in to restore the system databases as described in Restoring system databases on page 78.
4. Use the SQL Server plug-in to restore the user databases as described in Restoring to the original location on page 69.
5. When you are sure that the restore of the environment is correct and complete, perform a full backup of the newly restored environment.

A

SQL Plug-in Options

Topics:

- How to set plug-in options
- SQL plug-in backup options
- Backup options
- SQL plug-in restore options
- SQL TLR plug-in options

How to set plug-in options

Plug-in options enable you to control specific actions for on-demand backups, restores, and scheduled backups. The available plug-in options depend on the operation type and plug-in type.

Specify plug-in options in the AUI for on-demand backup or restore wizards, or when a dataset for a scheduled backup is created. Set plug-in options with the graphical user interface (GUI) controls (text boxes, check boxes, radio buttons, and so forth). Type an option and its value in the **Key** and **Value** fields.

NOTE: The Avamar software does not check or validate the information that is typed in the **Show Free Form** section of the **More Options** pane. The values in the **Key** and **Value** fields override settings that are specified with the GUI controls for the options.

SQL plug-in backup options

Backup plug-in options enable you to control backup functionality that is specific to the Avamar Plug-in for SQL Server. You can specify plug-in options for on-demand and scheduled backups.

The following table lists options that are available for the Avamar Plug-in for SQL Server when you perform an on-demand backup or when you configure a dataset for scheduled backups.

An asterisk (*) indicates an advanced option, which only appears when you select **Show Advanced Options** on the **Backup Command Line Options** dialog box.

Table 8. Backup plug-in options

Setting	Description
Backup label	Assigns this descriptive label to the backup.
Backup type	Sets the backup level. One of the following values: ● Full — Backs up the entire database, including all objects, system tables, and data. ● Differential — Backs up any data that changed after the last complete backup. ● Incremental — Backs up only the transaction logs. **NOTE:** Microsoft SQL Server does not allow differential or transaction log (incremental) backups of the master database. You can only create full backups of the master database. If you try to perform a differential or transaction log backup of the master database, a full backup occurs instead.
Copy-only Backup	Instructs the SQL plug-in to perform a copy-only full backup.
Availability group replica for backup	Controls the replica on which the backup occurs in an AlwaysOn environment. One of the following values:

Table 8. Backup plug-in options (continued)

Setting	Description
	● Primary — Specifies that the backup should occur on the primary replica. ● Prefer secondary — Specifies that the backup should occur on a secondary replica except when the primary replica is the only available replica. This is the default value. ● Secondary only — Specifies that the backup should occur on the secondary replica. If no secondary replica is available, then the backup does not occur. ● SQL Server defined — Specifies that the backup should use the backup preference specified in SQL Server settings. You can increase performance of the primary replica when you perform backups on a secondary replica. However, the secondary replica must meet the AlwaysON availability group requirements. In a hybrid IT environment with Microsoft Azure and AlwaysOn availability groups, ensure that the backup occurs on an on-premise replica. Avamar cannot back up databases on a replica on a Microsoft Azure virtual machine.
Enhanced data deduplication	Controls the type of compression for data in a backup by setting the `--compress` option in `avtar`. Enhanced data deduplication typically reduces the amount of client data that the backup process must send to the server, but requires additional client CPU resources. Select one of the following values: ● Default — The `avtar` process evaluates some of the data in the backup to determine whether the data supports compression. If the evaluated data supports compression, then `avtar` compresses all data in the backup. If the evaluated data does not support compression, then `avtar` does not compress any data. ● Disabled — The `avtar` process does not compress data during the backup. ● Enabled — Forces `avtar` to compress data during the backup, even if the data does not support compression. Select this value only when you are certain that data in the backup supports compression. Otherwise, `avtar` unnecessarily uses additional client CPU resources to try to perform compression on data that does not support compression.
Force incremental backup after full backup	If a full backup exists and then another full backup occurs, this option creates a transaction log (incremental) backup. The transaction log backup contains transactions from the end of the first full backup or the most recent transaction log backup, whichever is later, until the time that the forced transaction log backup occurs (after the first backup of the new backup cycle). This ensures that a point-in-time recovery to a point in time between the two full backups can occur, if required. You cannot perform a transaction log backup on databases that use the simple recovery model because those databases do not support transaction log backups. This includes system databases such as the master and msdb databases. Use the **For simple recovery model databases** list to control what Avamar does when you try to perform transaction log backups of databases that use the simple recovery model.
*Force full backup	Performs a full backup in the following scenarios: ● If you try to perform a differential or transaction log (incremental) backup and a full backup does not exist on the server. ● If you try to perform a differential or transaction log backup after you restore a database. ● If you try to perform a differential or transaction log backup after you restore the msdb database. ● If log gaps are detected. Effectively, this option automates taking a full backup when necessary. As a best practice, perform a transaction log backup after the forced full backup.

Table 8. Backup plug-in options (continued)

Setting	Description
	NOTE: If you perform transaction log and differential backups, it is strongly recommended that you leave this option selected (the default setting) for all backups. Otherwise, you might not be able to restore data if there are no full backups on the Avamar server.
Enable consistency check	Instructs the SQL plugin to perform a consistency check of the database prior to performing a backup.
*Truncate database log	Controls database transaction log truncation behavior. One of the following values: ● (Default) Only for incremental backup — The backup process truncates the database transaction log if the backup type is incremental (transaction log). No log truncation occurs if the backup type is full or differential. ● For all backup types — The backup process truncates the database transaction log regardless of the backup type. This setting breaks the chain of log backups. Do not use this setting unless the backup type is full. This value is not supported for SQL Server 2008 R2 and later environments. ● Never — The backup process does not truncate the database transaction log under any circumstances.
*For simple recovery model databases	Specifies what Avamar does when you try to perform incremental (transaction log) backups of databases that use the simple recovery model, which does not support transaction log backups: ● Skip incremental with error — If you select databases with different recovery models for the backup, then the backup does not include the databases with the simple recovery model. The backup completes with exceptions, and an error message is written to the log. If you select only databases with the simple recovery model for the backup, then the backup fails. ● Skip incremental with warning — If you select databases with different recovery models for the backup, then the backup does not include databases with the simple recovery model. The backup completes successfully, and a warning is written to the log for each database that uses the simple recovery model. If you select only databases with the simple recovery model for the backup, then the backup fails. ● Promote incremental to full — A full backup occurs automatically instead of a transaction log backup for databases that use the simple recovery model. Promotion to full is indicated in the Activity Window of the Avamar Administrator with the status **Completed with Promotion**. When only some of the databases in the backup are promoted to full, the status still indicates as **Completed with Promotion**.
*Enable debugging messages	Specifies to write maximum information to log files for debugging purposes. The backup process generates large log files.
*Skip log gap detection	For transaction log backups, controls whether the Avamar Plug-in for SQL Server checks the full transaction log backups that are already saved for a database to verify that there are no "log gaps." This log gap detection ensures that there is a valid chain of SQL Server transaction logs so that point-in-time restore can occur. If a log gap is detected, the backup is promoted to a full backup. Promotion to full is indicated in the Activity Window of the Avamar Administrator with the status **Completed with Promotion**. When only some of the databases in the backup are promoted to full, the status still indicates as **Completed with Promotion**. Log gap detection may require as many as 20 to 30 seconds to complete for each database. Transaction log backups may take an excessive amount of time to complete in environments with a significant number of databases. If the performance issues outweigh the risk of possible log gaps, then you can disable log gap detection by selecting this option to improve performance.

Table 8. Backup plug-in options (continued)

Setting	Description
Store backup on Data Domain system	Stores the backup on a configured Data Domain system instead of on the Avamar server. To store the backup on a Data Domain system, select the checkbox and then select the Data Domain system from the list.
Encryption method to Data Domain system	Specifies the encryption method for data transfer between the client and the Data Domain system during the backup.
Multi-streaming options	
Maximum number of streams	Specifies the maximum number of streams to use for backup and restore. The default is 2, and the maximum value is 10. If you select a value of 1, then Avamar backs up one database at a time with one stream.
Minimum stream size	Specifies the minimum size of each data stream.
Authentication options	
*SQL server address	Connect to this hostname or IP address for the computer that is running SQL Server.
*Authentication method	Specifies whether to use Windows (NT) authentication or SQL Server authentication to connect to SQL Server. If you do not specify an authentication method, then the SQL Server plug-in uses NT authentication and logs in with the NT AUTHORITY\SYSTEM account.
*SQL login ID	If you select SQL server authentication in the **Authentication method** list, connect to the server with this login ID.
*SQL password	If you select SQL server authentication in the **Authentication method** list, type the password for the SQL login account.

Backup options

Backup options for the `avsql` command enable you to control backup behavior. Many of these options are the same as the plug-in options that you specify in Avamar Administrator when you perform an on-demand backup or create a dataset for scheduled backups.

The following `avsql` options are available for the `backup` operation.

Table 9. Backup options for the `avsql` command

Option	Description
`--clusternode=`*HOSTNAME(IP)*	Statically defines the node's IP address in a multi-NIC environment when backups are failing due to a resolution error. *HOSTNAME* is the node's hostname and *IP* is the node's network address. **NOTE:** Include the parentheses around the IP address.
`--consistency-check={true | extendedlogicalchecks | physicalonly}`	Instructs the SQL plugin to perform a consistency check of the database before performing a backup. • `true` enables consistency checks using the default behavior. • `extendedlogicalchecks` performs an extended logical consistency check, which includes checks on indexed view, as well as XML and spatial indexes. • `physicalonly` performs a physical-only consistency check (column-integrity checks are not performed).
`--copy-only={true | false}`	`true` forces the backup to be a copy-only full backup.

Table 9. Backup options for the `avsql` command (continued)

Option	Description
`--ddr-encrypt-strength={high | medium | none}`	Specifies the encryption method for data transfer between the client and the Data Domain system when you store the backup on a Data Domain system. The default value is `high`. This option is an `avtar` option.
`--ddr={true | false}`	Specifies whether to store the backup on a Data Domain system instead of the Avamar server. To store the backup on a Data Domain system, specify `true`. This option is an `avtar` option.
`--ddr-index=`*n*	Specifies the index number (1, 2, 3, and so forth) of the Data Domain system on which to store the backup. The index number is assigned to the Data Domain system when it is added to the Avamar server configuration. This option is an `avtar` option.
`--encrypt=`*value*	Specifies the encryption method for data transfer between the client and the Avamar server during the backup. One of the following values: • `proprietary` • `tcp` • `ssl` • `sslverify` • `tls` • `tls-sa` Use this option with the `--encrypt-strength` option. The *Avamar Product Security Guide* provides details on how to correctly specify the `--encrypt` and `--encrypt-strength` option values together to ensure proper encryption settings. This option is an `avtar` option.
`--encrypt-strength=`*high*	Specifies the encryption strength for data transfer between the client and the Avamar server during the backup. The only allowed value is `high`. Use this option with the `--encrypt` option. The *Avamar Product Security Guide* provides details on how to correctly specify the `--encrypt` and `--encrypt-strength` option values together to ensure proper encryption settings.
`--expires=`*{days | timestamp}*	Specifies backup expiration as a number of days from today (*days*) or an absolute *timestamp*. This option is an `avtar` option.
`--label=`*name*	Assigns this descriptive label to the backup.
`--retention-type=`*type* `--retentiontype=`*type*	Assigns advanced retention to the backup. One of the following values: • `none` — Do not explicitly assign any retention type to this backup. That is, treat the backup as a normal on-demand backup. • `daily` — Explicitly designate this backup as a daily backup. • `weekly` — Explicitly designate this backup as a weekly backup. • `monthly` — Explicitly designate this backup as a monthly backup. • `yearly` — Explicitly designate this backup as a yearly backup. This option is an `avtar` option.
`--after-date=`*date*	To improve the performance of incremental backups, this option specifies the date at which a week-by-week search for a full backup stops and the history of all backups is retrieved. Specify the date in mm/dd/yyyy format.
`--backup-preference= {primary | prefer_secondary | secondary_only | SQL_server_defined}`	Controls the replica on which the backup occurs in an AlwaysOn environment. One of the following values: • `primary` — Specifies that the backup should occur on the primary replica.

Table 9. Backup options for the `avsql` command (continued)

Option	Description
	• `prefer_secondary` — Specifies that the backup should occur on a secondary replica except when the primary replica is the only available replica. This is the default value. • `secondary_only` — Specifies that the backup should occur on the secondary replica. If no secondary replica is available, then the backup does not occur. • `SQL_server_defined` — Specifies that the backup should use the backup preference specified in SQL Server settings. You can increase performance of the primary replica when you perform backups on a secondary replica. However, the secondary replica must meet the requirements in AlwaysOn availability group requirements on page 34. In a hybrid IT environment with Microsoft Azure and AlwaysOn availability groups, ensure that the backup occurs on an on-premise replica. Avamar cannot back up databases on a replica on a Microsoft Azure virtual machine.
`--brtype={full | differential | incremental}`	Specifies the type of backup to perform: • `full` — Backs up the entire database, including all objects, system tables, and data. • `differential` — Backs up any data that has changed since the last complete backup. • `incremental` — Backs up only the transaction logs. The default value is `full`. **NOTE:** Microsoft SQL Server does not allow differential or transaction log (incremental) backups of the master database. You can only create full backups of the master database. If you try to perform a differential or transaction log backup of the master database, a full backup is created instead.
`--exclude=`*data*	Specifies an instance or database to exclude from the backup. When specifying the local instance, use `(local)`. For example: • `--exclude=(local)` • `--exclude=(local)/DB1`
`--forcefull`	Performs a full backup in the following scenarios: • If you try to perform a differential or transaction log (incremental) backup and a full backup does not exist on the server. • If you try to perform a differential or transaction log backup after you restore a database. • If you try to perform a differential or transaction log backup after you restore the msdb database. • If log gaps are detected. Effectively, this option automates taking a full backup when necessary. **NOTE:** If you perform transaction log and differential backups, we strongly recommend that you leave the default setting of enabled for the `--forcefull` option for all backups. Otherwise, you might not be able to restore data if no existing full backup is present on the Avamar server.
`--forceincremental= {true | false}`	If a full backup exists and another full backup occurs, this option creates a transaction log (incremental) backup that contains transactions which occur from the end of the first full backup or the most recent transaction

Table 9. Backup options for the `avsql` command (continued)

Option	Description
	log backup, whichever is later. This action proceeds until the time that the forced transaction log backup occurs (after the first backup of the new backup cycle). This step ensures that a point-in-time recovery to a point in time between the two full backups can occur, if required. Three backup sessions occur when a forced transaction log backup occurs: ● The first session contains transactions that occur from the end of the first full backup until the start time of the forced transaction log backup. ● The second session contains transactions that occur from the end of the second full backup until the time that the forced transaction log backup occurs. ● The third session contains only the full backup. To restore and recover the database to a point after the forced transaction log backup, select the last backup from the last backup session that has the most recent full backup, during the restore process. **NOTE:** You cannot perform a transaction log backup on databases that use the simple recovery model because those databases do not support transaction log backups. This includes system databases such as the master and msdb databases. As a result, set `--forceincremental=false` when you back up a database that uses the simple recovery model.
`--include=`*data*	Specifies an instance or database to include in the backup that otherwise would have been excluded based on the `--exclude` option. When specifying the local instance, use `(local)`. For example: ● `--include=(local)` ● `--include=(local)/DB1`
`--max-streams=`*n*	Specifies the maximum number of streams to use for backup and restore. The default is 1, and the maximum value is 10.
`--min-snapup-size=`*MB*	Specifies the minimum size of each data stream in MB. The default value is 200.
`--prefix=`*value*	Specifies a prefix for the log filename.
`--skip-log-gap-detection`	For transaction log backups, controls whether the Avamar Plug-in for SQL Server checks the full and transaction log backups that are already saved for a database to verify that there are no "log gaps." This log gap detection ensures that there is a valid chain of SQL Server transaction logs so that point-in-time restore can occur. If a log gap is detected, the backup is promoted to a full backup. Promotion to full is indicated in the Activity Window of the Avamar Administrator with the status **Completed with Promotion**. When only some of the databases in the backup are promoted to full, the status still indicates as **Completed with Promotion**. Log gap detection may require as many as 20 to 30 seconds to complete for each database. Transaction log backups may take an excessive amount of time to complete in environments with a significant number of databases. If the performance issues outweigh the risk of possible log gaps, then you can disable log gap detection by specifying this option to improve performance.
`--skip_or_prom_sdm={skip_with_error | skip_with_warning | promote_to_full}`	Specifies what Avamar does when you try to perform incremental (transaction log) backups of databases that use the simple recovery model, which does not support transaction log backups:

Table 9. Backup options for the `avsql` command (continued)

Option	Description
	• `skip_with_error` — If you select databases with different recovery models for the backup, then the backup does not include the databases with the simple recovery model. The backup completes with exceptions, and an error message is written to the log. If you select only databases with the simple recovery model for the backup, then the backup fails. • `skip_with_warning` — If you select databases with different recovery models for the backup, then the backup does not include databases with the simple recovery model. The backup completes successfully, and a warning is written to the log for each database that uses the simple recovery model. If you select only databases with the simple recovery model for the backup, then the backup fails. • `promote_to_full` — A full backup occurs automatically instead of a transaction log backup for databases that use the simple recovery model. Promotion to full is indicated in the Activity Window of the Avamar Administrator with the status **Completed with Promotion**. When only some of the databases in the backup are promoted to full, the status will still be indicated as **Completed with Promotion**. The default value is `skip_with_error`.
`--truncatelog_ts= {default | enable | disable}`	Specifies database transaction log truncation behavior. One of the following values: • `default` — The backup process truncates the database transaction log if the backup type is set to incremental (transaction log). No log truncation occurs if the backup type is full or differential. • `enable` — The backup process truncates the database transaction log regardless of the backup type. This setting breaks the chain of log backups and should not be used unless the backup type is set to full. This value is not supported for SQL Server 2008 R2 and later environments. • `disable` — The backup process does not truncate the database transaction log under any circumstances.

SQL plug-in restore options

Restore plug-in options enable you to control restore functionality that is specific to the Avamar Plug-in for SQL Server. Set restore options manually from the **Restore Command Line Options** dialog box.

The following table lists options for restore operations with the Avamar Plug-in for SQL Server.

An asterisk (*) indicates an advanced option, which only appears when you select **Show Advanced Options** on the **Restore Command Line Options** dialog box.

Table 10. Restore plug-in options

Settings	Description
Encryption method from Data Domain system	Specifies the encryption for data transfer between the Data Domain system and the client during the restore when the backup is stored on a Data Domain system.
Restore options	
Normal restore	Performs a normal restore of databases.
Use SQL REPLACE option	Adds an SQL `WITH REPLACE` clause statement to the restore Transact-SQL command. The statement specifies that SQL Server should create any necessary database and related files even if another database or file exists with the same name.

Table 10. Restore plug-in options (continued)

Settings	Description
	NOTE: This option overrides a SQL Server safety check to prevent you from accidentally overwriting a different database or file. The *Microsoft Transact-SQL Reference Manual* provides more information in the `RESTORE` command section. You may need to use this option if a previous database restore exited with the following SQL Server error in the Avamar SQL restore log: `One or more devices or files already exist. Reissue the statement using the WITH REPLACE option to overwrite these files and devices.` You should also use this option when restoring system databases.
Tail-log backup	Backs up the tail of the transaction log to capture the log records that have not been backed up. The restore process uses the tail-log backup after the database restore to recover the transactions that were not in the backup. Tail-log backup is available only for databases that use the full and bulk-logged recovery models. Perform a tail-log backup during a direct restore without the SQL `REPLACE` option. Clear the checkbox when you perform a redirected restore to a different SQL Server instance.
Restore only on primary replica	Restores the database on only the primary replica in an AlwaysOn availability group. When you select this option, you must manually perform database synchronization on the secondary replicas. Restoring to an AlwaysOn availability group on page 83 provides details on restoring to an AlwaysOn availability group.
VerifyOnly restore	Performs a verify-only restore, which verifies the backup but does not restore it.
*Recovery operation	Specifies the recovery operation that occurs after the restore: • `RECOVERY` — The database is fully recovered and online. This is the default setting. • `NORECOVERY` — The database remains in a restoring state. This enables you to perform additional manual restore tasks, such as applying additional SQL log files. • `STANDBY` — The database is in standby (read-only) mode. This option creates a file with recovery changes. You can use the file to revert the recovery changes, if required. The size of the file depends on the volume of undo actions from uncommitted transactions. Specify the path to this file in the **Standby file location** box. `STANDBY` enables you to bring up a database for read-only access between transaction log restores. Use this setting with either warm standby server situations or special recovery situations in which it is useful to inspect the database between log restores. Do not use `STANDBY` when a database upgrade is necessary. You may need to perform a database upgrade when restoring backup sets from an earlier version of SQL Server.
*Standby file location	When you select `STANDBY` from the **Recovery operation** list, use this setting to specify the path to the standby file with recovery changes. The path must be in the following format: *`drive:\folder\subfolder`* where *drive* is the drive letter on the destination client, and *folder\subfolder* is the path on the drive letter in which to create the file.

Table 10. Restore plug-in options (continued)

Settings	Description
	If a file with recovery changes exists in the location, then SQL Server overwrites it. If you do not specify a path, then Avamar creates the file in the `C:\Program Files\avs\var` folder, where *C:\Program Files\avs* is the Avamar installation folder.
*Enable debugging messages	Specifies to write maximum information to log files for debugging purposes. The restore process generates very large log files.
System Databases options	
Restore system databases	Causes the plug-in to create an expanded list of databases to restore that includes system databases and user databases. By default, the plug-in only shows user databases.
Manage SQL services automatically during restore	When you restore the master database, this option automatically stops the SQL Server instance, including dependent services such as the SQL Server agent service and the Analysis Service. This step then restarts the instance in single-user mode before the restore. After the restore, the instance automatically restarts. When you restore the msdb database, this option automatically stops the SQL Server agent service, and then restarts it when the restore is complete. When you select both system and user databases for restore, you must select this option to ensure that all system databases restore in the proper order and with the necessary service stops and restarts.
Redirected Restore options	
*New database name	To give a database a different name when you restore it to either a different instance on the same server or to a different server, type the new name here. Only the name of the restored database is different. The source database in the backup remains the same.
*Alternate database location	If you are relocating the database, type the full path of the new database file (`*.mdf`) location on the client, such as `C:\temp`. You also can specify the file locations by clicking **Set Destination** on the **Restore Options** dialog box. If you specify the file locations in the **Restore Options** dialog box, then you do not need to specify them as plug-in options on the **Restore Command Line Options** dialog box. If you are restoring a database with multiple filegroups or files and you want to restore the files to different locations, then you must use the redirected restore options on the **Restore Options** dialog box.
*Alternate log location	Specifies whether the database log file (`*.ldf`) restores to the same location as the database or to a different location.
*Path to alternate log location	If you are restoring the database log file to a different location than the database, type the full path of the new database log file location on the client, such as `C:\temp\logs`.
Authentication options	
*SQL server address	Connect to this hostname or IP address for the computer that is running SQL Server.
*Authentication method	Specifies whether to use Windows (NT) authentication or SQL Server authentication to connect to SQL Server. If you do not specify an authentication method, then the SQL Server plug-in uses NT authentication and logs in with the NT AUTHORITY\SYSTEM account. Security requirements on page 33 provides more information.

Table 10. Restore plug-in options (continued)

Settings	Description
*SQL login ID	If you select SQL server authentication in the **Authentication method** list, connect to the server with this login ID.
*SQL password	If you select SQL server authentication in the **Authentication method** list, type the password for the SQL login account.
Point-in-Time Recovery options	
*Point-in-time recovery mode	Specifies whether to recover the database to a specific date and time or a named mark in the transaction log. Available only if the database uses the full recovery model.
*Point-in-time or mark name string	If you select **Point-in-time** or **Mark name** from the **Point-in-time recovery mode** list, then this option specifies the point in time or named mark to which to recover: ● Point-in-time — To recover to a specific point in time, specify the date and time in yyyy-mm-ddThh:mm:ss format. For example, 2014-05-25T14:15:45 is May 25, 2014 at 2:15:45 p.m. ● Mark name — If named marks are inserted in the transaction log for recovery purposes, then you can recover to a specific mark. Specify the mark name.
*Mark recovery point	Specifies whether to include the mark in the recovery: ● Before mark — Specifies that the log record immediately before the mark is the recovery point. In other words, the recovery rolls forward to the mark and excludes the marked transaction. ● At mark — Specifies that the marked transaction is the recovery point. In other words, the recovery rolls forward to the mark and includes the marked transaction.
*Mark is after date/time	If named marks are not unique in the transaction log, then you can specify a date and time to locate the mark to which it recovers. The recovery process stops at the first mark with the specified name, exactly at or after the specified date and time. Specify the date and time in yyyy-mm-ddThh:mm:ss format. For example, 2014-05-25T14:15:45 is May 25, 2014 at 2:15:45 p.m.

SQL TLR plug-in options

You can set plug-in options in Avamar Administrator for browse and restore operations for table level recovery with the SQL TLR plug-in.

SQL TLR plug-in browse options

Plug-in options are available when you browse backups to perform a table level recovery with the Avamar Plug-in for SQL TLR.

Table 11. Restore options for the SQL TLR plug-in

Option	Description
Drive letter or mount path	Enter the drive letter that will be used to mount the SQL backup data. This option is not case sensitive. If the letter entered here has a drive already mounted on it and that drive is not controlled by the Avamar Plug-in for SQL TLR, the mount operation will fail. If the the Avamar Plug-in for SQL TLR has already mounted a drive on this letter, the existing drive will be unmounted and the new one will be mounted. Entering a forward slash (`/`) or backslash (`\`) will mount the drive using the first available drive letter, beginning with `Z:`.

Table 11. Restore options for the SQL TLR plug-in (continued)

Option	Description
Amount of time to leave AvFS mounted	Select when to automatically dismount the drive that is being mounted for the SQL backup data: • **2 hours** • **8 hours** • **16 hours** • **24 hours** The drive dismounts even if the drive is in use.

SQL TLR plug-in restore options

Plug-in options are available when you perform table level recovery with the Avamar Plug-in for SQL TLR.

Table 12. Restore options for the SQL TLR plug-in

Option	Description
Amount of time to leave AvFS mounted	Select when to automatically dismount the temporary file system from the SQL server that you are using for TLR: • **2 hours** • **8 hours** • **16 hours** • **24 hours** The drive dismounts even if the drive is in use.
Advanced Options	
Enable debugging messages	Select the checkbox to write maximum information to the log files for debugging. Use caution when you select this option. The restore process creates large log files.

B

Command Line Interface

Topics:

- Overview of the SQL Server plug-in CLI
- Command reference
- Specifying command line options
- Password encoding
- Performing command line operations in a failover cluster
- Performing command line operations in an AlwaysOn environment
- CLI examples
- avsqltlr command reference
- Monitoring CLI activities

Overview of the SQL Server plug-in CLI

The Avamar Plug-in for SQL Server enables you to use a command line interface (CLI) with the `avsql` command instead of Avamar Administrator to back up and restore SQL Server data. You can use the CLI from all operating systems that the SQL Server plug-in supports. Additionally, the Avamar Plug-in for SQL TLR includes a command line interface, `avsqltlr`, that allows you to extend the amount of time that a database backup mounted for table level recovery which will remain mounted.

When you use the CLI to perform a backup or restore, specify the command options for the SQL Server plug-in binary at the command prompt. The plug-in interacts with the `avtar` process to write backup data to or read backup data from the Avamar server or a Data Domain system.

The `avsql` binary is available in the `C:\Program Files\avs\bin` folder, where *C:\Program Files\avs* is the Avamar client installation folder.

To use the CLI, open a command prompt and change directory to the `bin` folder. Then type the command at the command prompt.

Command reference

The `avsql` command enables you to browse for SQL Server plug-in data to back up or restore, perform the backup, or perform the restore.

Synopsis

```
avsql --operation={browse | backup | restore} [options]
```

Operations

The operation that you specify with the `avsql` command controls the task that `avsql` performs. Available operations include `browse`, `backup`, and `restore`.

Supply one of the operations in the following table for `avsql` by using the `--operation` option.

Table 13. Operations for the `avsql` command

Operation	Description
`browse`	Returns a list of data on the client that is available for backup. You can browse all data on the client, a single instance, or a single database. You must specify only a single browse target on the command line. If you specify more than one instance or database, then `avsql` displays output for only the first target. Example browse commands on page 124 provides the correct syntax for different `browse` operation commands. The command results appear as standard output in the command window. The following information appears for each entry: ● Name ● Date ● Size ● Type ● User The data sorts alphabetically by name.
`backup`	Performs an on-demand backup of the specified data. Include options with the `backup` operation to specify the data to back up, the Avamar server for the backup, Avamar login credentials, a label for the backup, and other options to control backup behavior. Many of these plug-in options are the same options that you specify when you perform an on-demand backup or configure a scheduled backup in Avamar Administrator. The **Enhanced data deduplication** plug-in option appears in the **Backup Command Line Options** dialog box but is not available as an option when you perform a backup with the CLI.
`restore`	Restores the specified data. Include options with the `restore` operation to specify the data to restore, the Avamar server for the restore, Avamar login credentials, the target location for the restore, and other options to control restore behavior. Many of these plug-in options are the same options that you specify when you perform a restore in Avamar Administrator. The **Manage SQL services automatically during restore** option is the only plug-in option that appears in the **Restore Command Line Options** dialog box but is not available when you perform a restore with the CLI.

Options

The options that you specify with the `avsql` command control browse, backup, and restore behavior. Some options are available for multiple operations, such as the account options for connecting to the Avamar server. Other options apply only to backups or restores.

Common options

Common options for the `avsql` command are general options that are available for multiple operations.

The following common options are available for the `avsql` command.

Table 14. Common avsql options

Option	Description
--backup-type=*type*	Internal use only.
--exclude-pluginid-list=*string* --excludepluginids=*string*	Internal use only. This option is an avtar option.
--pluginid-list=*string* --pluginids=*string*	Internal use only. This option is an avtar option.
--version	Displays the build version of the Avamar Plug-in for SQL Server.
--sqltimeoutsecs=*seconds*	The number of seconds after which the connection to SQL Server times out. If not specified, then the timeout occurs after 100 seconds.

Account options

Account options for the avsql command enable you to specify credentials to connect to the Avamar server for backups and restores.

The following account options are available for the avsql command.

Table 15. Account options for the avsql command

Option	Description
--account=*domain/client* --acnt=*domain/client* --path=*domain/client*	Required for all backup and restore operations. Specifies the client to back up or restore from using the following format: *domain/client* where *domain* is the Avamar domain to which the client belongs and *client* is the name of the client computer or the Avamar cluster client. This option is an avtar option.
--hostname-sql=*client*	Specifies the name of the client to browse, back up, or restore from. Required for browse, backup, and restore in an AlwaysOn availability group or a failover cluster. Specify the Avamar cluster client as the client.
--id=*user@domain/client*	Required for all backup and restore operations. Specifies the Avamar username for authentication. **NOTE:** You can specify the values for the --id and --[avtar]account options simultaneously by specifying --id=*user@domain/client*.
--password=*password* --pswd=*password* --ap=*password*	Required for all backup and restore operations. Specifies the password for the Avamar account.
--server=*Avamar_server* --hfsaddr=*Avamar_server*	Required for all backup and restore operations. Specifies the hostname or IP address of the Avamar server. This option is an avtar option.

Table 15. Account options for the `avsql` command (continued)

Option	Description
`--ntorsqlauth= {ntauthentication | sqlauthentication}`	Specifies whether to use Windows (NT) authentication or SQL Server authentication to connect to SQL Server. One of the following values: ● `ntauthentication` ● `sqlauthentication` If you do not specify an authentication method, then the SQL Server plug-in uses NT authentication and logs in with the NT AUTHORITY\SYSTEM account. "Security requirements" on page 39 provides more information.
`--pluginport=`*port*	Specifies the port number on which to connect to the agent. If not specified, then the default value of 28002 is used.
`--sqllogin=`*name*	If you set `--ntorsqlauth=sqlauthentication`, then this option specifies the login ID for the SQL login account.
`--sqlpassword=`*password*	If you set `--ntorsqlauth=sqlauthentication`, then this option specifies the password for the SQL login account.
`--sqlserver=`*server*	Hostname or IP address of the computer with the SQL Server installation that you want to browse, back up, or restore from. This option is required when you use the CLI to browse, back up, or restore data in the following environments: ● In the local instance when the server is in a failover cluster. Specify the virtual server name. ● In an AlwaysOn availability group. Specify the availability group listener. In environments where SQL Server does not use the default port, you can specify the port as part of the `--sqlserver` option by using the format `--sqlserver=`*server,port*, where *server* is the name of the instance in a stand-alone environment, the virtual server name in a failover cluster, or the availability group listener in an AlwaysOn environment; and *port* is the port number.

Logging options

Logging options for the `avsql` command enable you to specify the path and file name for the `avsql` log file, and to control how much information the plug-in writes to the log file.

The following logging options are available for the `avsql` command.

Table 16. Logging options for the `avsql` command

Option	Description
`--informationals=`*n*	Sets the information level for status messages, where *n* is a number such as 0, 1, 2, and so on. This option is an `avtar` option.
`--log=`*file* `--logfile=`*file*	Specifies the full path and file name of the avsql plug-in log file. **NOTE:** The Activity Monitor in Avamar Administrator displays only `avtar` logs for backups with the command line interface. To create a plug-in log for CLI activities, you must specify the `--log` or `--logfile` option on the command line.

Table 16. Logging options for the `avsql` command (continued)

Option	Description
`--noinformationals`	Disables all status messages. This option is an `avtar` option.
`--nostdout`	Disables output to STDOUT. However, output still goes to the log file if you use the `--log` or `--logfile=`*file* options. This option is an `avtar` option.
`--nowarnings`	Disables warning messages. This option is an `avtar` option.
`--quiet`	Suppresses all debugging messages. This option is an `avtar` option.
`--verbose` `--v`	Enables all messages, including status and warning messages. This option is an `avtar` option.

Browse options

In addition to the following browse option for the `avsql` command, you may want to use the `--sqlserver` or `--hostname-sql` account options during certain browse operations.

The following browse option is available for the `avsql` command.

Table 17. Browse option for the `avsql` command

Option	Description
`--sql-version-detection=`*value*	Determiners whether or not event messages will be included. ● If this option is not used with the `--operation=browse option`, no event messages will be observed and detection will be faster. ● If *value* is set to `reg`, no event messages will be observed and detection will be faster. ● If *value* is set to `wmi`, SQL detection will be slower and the event messages will be observed.
`--usesmoversion` = *SMOVERSION*	Specifies the version of the SQL Server Management Objects (SMO) that should be used when the node has multiple SQL versions installed. The respective SMO version will then determine the SQL server that is being used. Allowed values are: ● For SQL server 2008, *SMOVERSION* = **8** ● For SQL server 2012, *SMOVERSION* = **11** ● For SQL server 2014, *SMOVERSION* = **12** ● For SQL server 2016, *SMOVERSION* = **13**

Backup options

Backup options for the `avsql` command enable you to control backup behavior. Many of these options are the same as the plug-in options that you specify in Avamar Administrator when you perform an on-demand backup or create a dataset for scheduled backups.

The following `avsql` options are available for the `backup` operation.

Table 18. Backup options for the `avsql` command

Option	Description
`--clusternode=`*HOSTNAME(IP)*	Statically defines the node's IP address in a multi-NIC environment when backups are failing due to a resolution error. *HOSTNAME* is the node's hostname and *IP* is the node's network address. **NOTE:** Include the parentheses around the IP address.
`--consistency-check={true | extendedlogicalchecks | physicalonly}`	Instructs the SQL plugin to perform a consistency check of the database before performing a backup. • `true` enables consistency checks using the default behavior. • `extendedlogicalchecks` performs an extended logical consistency check, which includes checks on indexed view, as well as XML and spatial indexes. • `physicalonly` performs a physical-only consistency check (column-integrity checks are not performed).
`--copy-only={true | false}`	`true` forces the backup to be a copy-only full backup.
`--ddr-encrypt-strength={high | medium | none}`	Specifies the encryption method for data transfer between the client and the Data Domain system when you store the backup on a Data Domain system. The default value is `high`. This option is an `avtar` option.
`--ddr={true | false}`	Specifies whether to store the backup on a Data Domain system instead of the Avamar server. To store the backup on a Data Domain system, specify `true`. This option is an `avtar` option.
`--ddr-index=`*n*	Specifies the index number (1, 2, 3, and so forth) of the Data Domain system on which to store the backup. The index number is assigned to the Data Domain system when it is added to the Avamar server configuration. This option is an `avtar` option.
`--encrypt=`*value*	Specifies the encryption method for data transfer between the client and the Avamar server during the backup. One of the following values: • `proprietary` • `tcp` • `ssl` • `sslverify` • `tls` • `tls-sa` Use this option with the `--encrypt-strength` option. The *Avamar Product Security Guide* provides details on how to correctly specify the `--encrypt` and `--encrypt-strength` option values together to ensure proper encryption settings. This option is an `avtar` option.
`--encrypt-strength=`*high*	Specifies the encryption strength for data transfer between the client and the Avamar server during the backup. The only allowed value is `high`. Use this option with the `--encrypt` option. The *Avamar Product Security Guide* provides details on how to correctly specify the `--encrypt` and `--encrypt-strength` option values together to ensure proper encryption settings.
`--expires=`*{days | timestamp}*	Specifies backup expiration as a number of days from today (*days*) or an absolute *timestamp*. This option is an `avtar` option.
`--label=`*name*	Assigns this descriptive label to the backup.
`--retention-type=`*type* `--retentiontype=`*type*	Assigns advanced retention to the backup. One of the following values:

Table 18. Backup options for the `avsql` command (continued)

Option	Description
	• `none` — Do not explicitly assign any retention type to this backup. That is, treat the backup as a normal on-demand backup. • `daily` — Explicitly designate this backup as a daily backup. • `weekly` — Explicitly designate this backup as a weekly backup. • `monthly` — Explicitly designate this backup as a monthly backup. • `yearly` — Explicitly designate this backup as a yearly backup. This option is an `avtar` option.
`--after-date=`*date*	To improve the performance of incremental backups, this option specifies the date at which a week-by-week search for a full backup stops and the history of all backups is retrieved. Specify the date in mm/dd/yyyy format.
`--backup-preference= {primary | prefer_secondary | secondary_only | SQL_server_defined}`	Controls the replica on which the backup occurs in an AlwaysOn environment. One of the following values: • `primary` — Specifies that the backup should occur on the primary replica. • `prefer_secondary` — Specifies that the backup should occur on a secondary replica except when the primary replica is the only available replica. This is the default value. • `secondary_only` — Specifies that the backup should occur on the secondary replica. If no secondary replica is available, then the backup does not occur. • `SQL_server_defined` — Specifies that the backup should use the backup preference specified in SQL Server settings. You can increase performance of the primary replica when you perform backups on a secondary replica. However, the secondary replica must meet the requirements in AlwaysOn availability group requirements on page 34. In a hybrid IT environment with Microsoft Azure and AlwaysOn availability groups, ensure that the backup occurs on an on-premise replica. Avamar cannot back up databases on a replica on a Microsoft Azure virtual machine.
`--brtype={full | differential | incremental}`	Specifies the type of backup to perform: • `full` — Backs up the entire database, including all objects, system tables, and data. • `differential` — Backs up any data that has changed since the last complete backup. • `incremental` — Backs up only the transaction logs. The default value is `full`. **NOTE:** Microsoft SQL Server does not allow differential or transaction log (incremental) backups of the master database. You can only create full backups of the master database. If you try to perform a differential or transaction log backup of the master database, a full backup is created instead.
`--exclude=`*data*	Specifies an instance or database to exclude from the backup. When specifying the local instance, use `(local)`. For example: • `--exclude=(local)` • `--exclude=(local)/DB1`
`--forcefull`	Performs a full backup in the following scenarios: • If you try to perform a differential or transaction log (incremental) backup and a full backup does not exist on the server.

Table 18. Backup options for the `avsql` command (continued)

Option	Description
	• If you try to perform a differential or transaction log backup after you restore a database. • If you try to perform a differential or transaction log backup after you restore the msdb database. • If log gaps are detected. Effectively, this option automates taking a full backup when necessary. **NOTE:** If you perform transaction log and differential backups, we strongly recommend that you leave the default setting of enabled for the `--forcefull` option for all backups. Otherwise, you might not be able to restore data if no existing full backup is present on the Avamar server.
`--forceincremental= {true | false}`	If a full backup exists and another full backup occurs, this option creates a transaction log (incremental) backup that contains transactions which occur from the end of the first full backup or the most recent transaction log backup, whichever is later. This action proceeds until the time that the forced transaction log backup occurs (after the first backup of the new backup cycle). This step ensures that a point-in-time recovery to a point in time between the two full backups can occur, if required. Three backup sessions occur when a forced transaction log backup occurs: • The first session contains transactions that occur from the end of the first full backup until the start time of the forced transaction log backup. • The second session contains transactions that occur from the end of the second full backup until the time that the forced transaction log backup occurs. • The third session contains only the full backup. To restore and recover the database to a point after the forced transaction log backup, select the last backup from the last backup session that has the most recent full backup, during the restore process. **NOTE:** You cannot perform a transaction log backup on databases that use the simple recovery model because those databases do not support transaction log backups. This includes system databases such as the master and msdb databases. As a result, set `--forceincremental=false` when you back up a database that uses the simple recovery model.
`--include=`*data*	Specifies an instance or database to include in the backup that otherwise would have been excluded based on the `--exclude` option. When specifying the local instance, use `(local)`. For example: • `--include=(local)` • `--include=(local)/DB1`
`--max-streams=`*n*	Specifies the maximum number of streams to use for backup and restore. The default is 1, and the maximum value is 10.
`--min-snapup-size=`*MB*	Specifies the minimum size of each data stream in MB. The default value is 200.
`--prefix=`*value*	Specifies a prefix for the log filename.
`--skip-log-gap-detection`	For transaction log backups, controls whether the Avamar Plug-in for SQL Server checks the full and transaction log backups that are already saved for a database to verify that there are no "log gaps." This log gap detection ensures that there is a valid chain of SQL Server transaction logs so that

Table 18. Backup options for the `avsql` command (continued)

Option	Description
	point-in-time restore can occur. If a log gap is detected, the backup is promoted to a full backup. Promotion to full is indicated in the Activity Window of the Avamar Administrator with the status **Completed with Promotion**. When only some of the databases in the backup are promoted to full, the status still indicates as **Completed with Promotion**. Log gap detection may require as many as 20 to 30 seconds to complete for each database. Transaction log backups may take an excessive amount of time to complete in environments with a significant number of databases. If the performance issues outweigh the risk of possible log gaps, then you can disable log gap detection by specifying this option to improve performance.
`--skip_or_prom_sdm= {skip_with_error | skip_with_warning | promote_to_full}`	Specifies what Avamar does when you try to perform incremental (transaction log) backups of databases that use the simple recovery model, which does not support transaction log backups: • `skip_with_error` — If you select databases with different recovery models for the backup, then the backup does not include the databases with the simple recovery model. The backup completes with exceptions, and an error message is written to the log. If you select only databases with the simple recovery model for the backup, then the backup fails. • `skip_with_warning` — If you select databases with different recovery models for the backup, then the backup does not include databases with the simple recovery model. The backup completes successfully, and a warning is written to the log for each database that uses the simple recovery model. If you select only databases with the simple recovery model for the backup, then the backup fails. • `promote_to_full` — A full backup occurs automatically instead of a transaction log backup for databases that use the simple recovery model. Promotion to full is indicated in the Activity Window of the Avamar Administrator with the status **Completed with Promotion**. When only some of the databases in the backup are promoted to full, the status will still be indicated as **Completed with Promotion**. The default value is `skip_with_error`.
`--truncatelog_ts= {default | enable | disable}`	Specifies database transaction log truncation behavior. One of the following values: • `default` — The backup process truncates the database transaction log if the backup type is set to incremental (transaction log). No log truncation occurs if the backup type is full or differential. • `enable` — The backup process truncates the database transaction log regardless of the backup type. This setting breaks the chain of log backups and should not be used unless the backup type is set to full. This value is not supported for SQL Server 2008 R2 and later environments. • `disable` — The backup process does not truncate the database transaction log under any circumstances.

Restore options

Restore options for the `avsql` command enable you to control restore behavior. Many of these options are the same as the plug-in options that you specify in Avamar Administrator when you perform a restore.

The following `avsql` options are available for the `restore` operation.

Table 19. Restore options for the `avsql` command

Option	Description
`--label=`*name*	Specifies the label of the backup to restore. Use either this option or the `--labelnum` option during the restore.
`--labelnum=`*number*	Specifies the label number of the backup to restore. Use either this option or the `--label` option during the restore.
`--after-date=`*date*	To improve the performance of the search for backups to restore, this option specifies the date at which a week-by-week search for a full backup stops and the history of all backups is retrieved. Specify the date in mm/dd/yyyy format.
`--aftertime=`*datetime*	If you are performing a point-in-time recovery to a named mark and named marks are not unique in the transaction log, then you can use this option to specify a date and time to locate the mark to which to recover. The recovery process stops at the first mark with the specified name, exactly at or after the specified date and time. Specify the date and time in yyyy-mm-ddThh:mm:ss format. For example, 2014-10-25T14:15:45 is October 25, 2014 at 2:15:45 p.m.
`--altlog={samelocat | difflocat}`	Specifies whether the database log file (`*.ldf`) is restored to the same location as the database or to a different location. One of the following values: • `samelocat` • `difflocat` The default value is `samelocat`.
`--app-instance=`*instance*	When restoring to a different instance, this option specifies the target instance for the restore.
`--ddr-encrypt-strength={high | medium | none}`	Specifies the encryption method for data transfer between the Data Domain system and the client when the backup is stored on a Data Domain system. The default value is `high`. This option is an `avtar` option.
`--dblocation=`*path*	If you are relocating the database, this option specifies the full path of the new database file (`*.mdf`) location on the client, where *path* is a value such as `C:\temp`.
`--encrypt=`*value*	Specifies the encryption method for data transfer between the Avamar server and the client during the restore. One of the following values: • `proprietary` • `tcp` • `ssl` • `sslverify` • `tls` • `tls-sa` Use this option with the `--encrypt-strength` option. The *Avamar Product Security Guide* provides details on how to correctly specify the `--encrypt` and `--encrypt-strength` option values together to ensure proper encryption settings. This option is an `avtar` option.
`--encrypt-strength=`*value*	Specifies the encryption strength for data transfer between the Avamar server and the client during the restore. One of the following values:

Table 19. Restore options for the `avsql` command (continued)

Option	Description
	• `cleartext` • `medium` • `high` Use this option with the `--encrypt` option. The *Avamar Product Security Guide* provides details on how to correctly specify the `--encrypt` and `--encrypt-strength` option values together to ensure proper encryption settings.
`--handle-sql-services-automatically` `--handlesqlservicesautomatically`	When restoring system databases, this option stops and restarts SQL services automatically as necessary. When you restore both system and user databases, you must specify this option to ensure that all system databases restore in the proper order and with the necessary service stops and restarts.
`--history-optimization={true | false}`	Enables or disables history search optimization, which you can further refine by using the `--after-date=`*date* option. The default setting is `--history-optimization=true`, which enables history search optimization.
`--loglocation=`*path*	If `--altlog=difflocat`, then this option specifies the path of the new database log file location on the client, where *path* is a value such as `C:\temp\logs`.
`--logtail`	Backs up the tail of the transaction log to capture the log records that are not in a backup. After the database restore, Avamar uses the tail-log backup to recover the transactions that were not in the backup. Tail-log backup is available only for databases that use the full and bulk-logged recovery models. Perform a tail-log backup during a direct restore without the SQL `REPLACE` option. Do not use this option when you restore to a different SQL Server instance.
`--metadata`	Retrieves the SQL metadata file from the backup and displays the metadata as screen output. When you use this option, Avamar does not restore any data.
`--newdbname=`*name*	Restores the database with the specified name (instead of its original name) when you are restoring it to either a different instance on the same server or to a different server.
`--overwrite-default-recoveryplan`	Specify this option to perform individual incremental or differential backups of the SQL server database.
`--pitormark={none | pointintime | markname}`	Whether to recover the database to a specific date and time or a named mark in the transaction log. Available only if the database uses the full recovery model. One of the following values: • `none` — Do not recover to a specific point in time. • `pointintime` — Recover to a specific date and time specified by `--pitormarkstr=`*value*. • `markname` — Recover to the named mark specified by `--pitormarkstr=`*value*. The default value is `none`.
`--pitormarkstr={`*pit* `|` *mark*`}`	If `--pitormark=pointintime` or `markname`, then this option specifies the *pit* (date and time) or named *mark* to recover to. To

Table 19. Restore options for the `avsql` command (continued)

Option	Description
	recover to a date and time, use yyyy-mm-ddThh:mm:ss format. For example, 2014-10-25T14:15:45 is October 25, 2014 at 2:15:45 p.m.
`--prefix=`*value*	Prefix for the name of the logfiles.
`--print-restore-size`	Prints the size required to restore SQL data. No actual restore takes place when this option is used.
`--recovery-primary-only={true | false}`	Specify `true` to restore the database on only the primary replica in an AlwaysOn availability group. When you specify `true`, you must manually perform database synchronization on the secondary replicas. Specify `false` to restore to both primary and secondary replicas.
`--recoveryoperation={RECOVERY | NORECOVERY | STANDBY}`	Specifies the recovery operation that occurs after the restore. One of the following values: • `RECOVERY` — The database is fully recovered and online. This is the default setting. • `NORECOVERY` — The database remains in a restoring state. This enables you to perform additional manual restore tasks, such as applying additional SQL log files. • `STANDBY` — The database is in standby (read-only) mode. This option creates a file with recovery changes. You can use the file to revert the recovery changes, if required. The size of the file depends on the volume of undo actions from uncommitted transactions. Specify the path to this file with the `--standbyfilelocation=`*path* option. `STANDBY` enables you to bring up a database for read-only access between transaction log restores. Use this setting with either warm standby server situations or special recovery situations in which it is useful to inspect the database between log restores. Do not use `STANDBY` when a database upgrade is necessary. You may need to perform a database upgrade when you restore backup sets from an earlier version of SQL Server.
`--recoverypoint={beforemark | atmark}`	When you perform a point-in-time restore to a named mark, this option specifies whether to include the mark in the recovery. One of the following values: • `beforemark` — Specifies that the log record immediately before the mark is the recovery point. In other words, the recovery rolls forward to the mark and excludes the marked transaction. • `atmark` — Specifies that the marked transaction is the recovery point. In other words, the recovery rolls forward to the mark and includes the marked transaction. The default value is `atmark`.
`--redirect=`*string*	Restores database files to the specified location, where string contains the original file name and the destination path and file name: `--redirect=`*original_file= destination_path\new_file* To restore multiple files to multiple different locations, separate the string values with commas:

Table 19. Restore options for the `avsql` command (continued)

Option	Description
	`--redirect=`*original_file1=destination_path1\new_file1*`,`*original_file2=destination_path2\new_file2*`,`*original_file3= destination_path3\new_file3* To restore a file to its original location, specify only the original filename: `--redirect=`*original_file*
`--redirecttofile={true | false}`	Specifies whether to restore database files as operating system files to the specified location.
`--restore-destination={original | multiple | single}`	Specifies the restore destination for the data: • `original` – Restore the data to its original location. This is the default value. • `multiple` – Restore the data to a different location. Use the `--target=`*{path | client\instance}* option to specify the location. • `single` – Restore the data to operating system files. Use the `--target=`*{path | client\instance}* option to specify the location. This option is an `avtar` option.
`--restore-filegroup-list=`*filegroups*	Restores the specified filegroups, using the following format: `--restore-filegroup-list=`*instance/ database/ filegroup,instance/database/ filegroup* where *instance/database/filegroup* are the instance, database, and filegroup to restore, with multiple entries separated by a comma.
`--restoresystem`	Attempts to restore databases that have the system attribute set.
`--standbyfilelocation=`*path*	If `--recoveryoperation=STANDBY`, then this option specifies the path on the destination client to the standby file with recovery changes. Use the following format: *`drive:\folder\subfolder`* where *drive* is the drive letter on the destination client, and *folder\subfolder* is the path on the drive letter in which to create the file. If a file with recovery changes already exists in the specified location, then SQL Server overwrites it. You can use the file to revert the recovery changes, if required.
`--target={`*path* `|` *client\instance*`}`	Specifies the target location for the restored files: • If `--restore-destination=single`, then specify the path on the original client to which to restore the files, such as `--target=C:\sqldata`. • If `--restore-destination=multiple`, then specify the instance on the destination client to which to restore the files. Use the following format: `--target=`*client\instance* where *client* is the name of the destination client and *instance* is the name of the instance on the client.

Table 19. Restore options for the `avsql` command (continued)

Option	Description
`--use-sql-replace-option` `--usesqlreplaceoption`	Adds an SQL `WITH REPLACE` clause statement to the restore Transact-SQL command. This statement specifies that SQL Server should create any necessary database and related files even if another database or file already exists with the same name. **NOTE:** This option overrides a SQL Server safety check to prevent you from accidentally overwriting a different database or file. The *Microsoft Transact-SQL Reference Manual* provides more information in the `RESTORE` command section. You may need to use this option if a previous database restore exited with the following SQL Server error in the Avamar SQL restore log: `One or more devices or files already exist. Reissue the statement using the WITH REPLACE option to overwrite these files and devices.` Use this option when restoring system databases.
`--validate`	Validates the integrity of a backup before performing the restore. This option is an `avtar` option.
`--verifyOnlyrestore={true | false}`	Specifies whether to perform a verify-only restore, which verifies the backup but does not restore it.

Help option

The `--help` option displays a list of available operations and options for the `avsql` command. You can view all help output for `avsql`, or only the options for a specific operation.

The following command displays all help output for the `avsql` command:

`avsql --help`

The following command provides a list of options for the backup operation:

`avsql --help --operation=backup`

Specifying command line options

The method that you use to specify options for the `avsql` command depends on the complexity of the command value and the frequency with which you use the options.

About this task

Type options directly on the command line when you perform command line operations on an occasional basis or you are using a limited set of simple options. Use a command file, option file, or environment variables when you frequently perform command line operations or you are using a complex set of options.

The following methods are available for specifying options for the `avsql` command:

- Type the individual options on the command line.
- List the options in the `avsql.cmd` file, which is in the *`C:\Program Files\avs\var`* folder, where *C:\Program Files\avs* is the Avamar client installation folder.

  In a cluster environment, the *`C:\Program Files\avs\var`* folder is typically on a shared network folder or volume. You specify the path for the `var` folder when you configure the cluster client with the Cluster Configuration Tool.

List each option on its own line, as shown in the following example:

```
--debug
--logfile=avamarclient.log
--verbose=5
```

- Create an option file as a text file, and then specify the option file on the command line by using the `--flagfile=path` option, where *path* is the full pathname of the option file.
- Pass options to the CLI as environment variables. For example, if you specify `AvamarServer=12.34.56.78` in the environment variables, then you can specify the environment variable instead of the IP address of the Avamar server in the command, as shown in the following example:

  ```
  avsql --operation=backup --server=%AvamarServer%
  --id=AvamarAdmin --ap=password --brtype=full
  --path=/SQL/SQLServer3 --label=SQL1Full
  --log=C:\Program Files\avs\var\avsql.log " "
  ```

Password encoding

You can use `avtar` to encode passwords that you type on the command line or store in script files, and then use the encoded string with the `--password`, `--ap`, or `--pswd` option.

Steps

1. Type the following command on the command line on the SQL Server client:

   avtar --encodepassword=*password*

   where *password* is the password to encode.

   An encoded string is returned on the command line.
2. Use the encoded string on the `avsql` command line as the password with the `--password`, `--ap`, or `--pswd` option.

Performing command line operations in a failover cluster

About this task

The following steps are required to use the CLI to browse, back up, or restore SQL Server data on shared storage in a failover cluster:

- Run the CLI on the on the active node that is managing the databases to back up.
- Specify the cluster client as the client to back up or restore from.
- Specify the `--hostname-sql` option with a value of the cluster client.
- To browse, back up, or restore the local instance, specify the `--sqlserver` option with a value of the cluster client.

Performing command line operations in an AlwaysOn environment

About this task

The following steps are required to use the CLI to browse, back up, or restore SQL Server data in an AlwaysOn availability group:

- Run the CLI on the cluster node with the primary replica.
- Specify the cluster client for the availability group listener as the client to back up or restore from.

- During a backup, specify the following options:
  - Specify the availability group listener with the `--sqlserver` option.
  - Specify the Avamar cluster client with the `--hostname-sql` option.
  - Specify the availability group listener, the availability group, the instance, and the database in quotation marks using the format *"listener/group\instance\database"*. To back up all databases in an availability group, specify *"listener/group\instance"*.
- When you restore a backup to its original location, specify the following options:
  - Specify the availability group listener in the `--sqlserver` and `--hostname-sql` options.
  - Use the `--recovery-primary-only` option to control whether to restore to only the primary replica or to both the primary and all secondary replicas.
- When you restore a backup to its original location but with a new name, specify the original database name at the end of the command. Use the `--newdbname` option to specify the new database name.

CLI examples

Review the `avsql` command examples for details on how to use options to control browse, backup, and restore behavior.

Example browse commands

These command examples illustrate how to use options to browse different types of data in client backups with the `avsql` command.

Browsing all instances

The following command returns a list of all SQL Server instances on the client:

```
avsql --operation=browse
```

Browsing the local instance

The following command returns a list of all databases in the local instance on the client:

```
avsql --operation=browse (local)
```

Browsing a named instance

The following command returns a list of all databases in the named instance of `Instance1` on the client:

```
avsql --operation=browse Instance1
```

Browsing a database

The following command returns detailed information about the database, including the creation date, size, and owner for the database:

```
avsql --operation=browse (local)/DB1
```

Browsing a failover cluster

The following command returns a list of all databases in the local instance in a failover cluster with the `SQLCluster1` cluster client:

```
avsql --operation=browse --sqlserver=SQLCluster1 --hostname-sql=SQLCluser1
```

Browsing an AlwaysOn availability group

The following command returns a list of all databases in the AlwaysOn availability group with the `SQLGroupListener` availability group listener:

```
avsql --operation=browse --sqlserver=SQLGroupListener --hostname-sql=SQLGroupListener
```

Example backup commands

These command examples illustrate how to use options to perform different types of backups with the `avsql` command.

Backing up all SQL Server data

To back up all SQL Server data (all databases in all instances on the client), specify the domain and client name.

Performing a full backup of all SQL Server data

The following command performs a full backup of all SQL Server data on the `SQLServer1` client in the `SQL` domain, labels the backup `SQL1Full`, and stores the backup on the Avamar server with an IP address of 12.34.56.78:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap=password --brtype=full --path=/SQL/SQLServer1
--log=C:\Program Files\avs\var\avsql.log --label=SQL1Full
```

Performing an incremental (transaction log) backup of all SQL Server data

The following command performs an incremental (transaction log) backup of all SQL Server data on the `SQLServer2` client in the `SQL` domain, labels the backup `SQL2Logs`, and stores the backup on the first Data Domain system that you added to the Avamar configuration:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap=password --brtype=incremental --path=/SQL/SQLServer2
--label=SQL2Logs --log=C:\Program Files\avs\var\avsql.log
--ddr=true --ddr-index=1
```

Backing up the local instance

To back up all databases in the local instance, specify `(local)` as the backup target between the quotation marks at the end of the command.

The following command performs a differential backup of the local instance on the `SQLServer3` client in the `SQL` domain and stores the backup on the Avamar server:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap=password --brtype=differential --path=/SQL/SQLServer3
--log=C:\Program Files\avs\var\avsql.log "(local)"
```

Backing up a named instance

To back up a named instance, specify the client and instance as the backup target between the quotation marks at the end of the command, using the format "*client\instance*".

The following command performs a full backup of the `sql02inst2` instance on the `SQLServer2` client in the `SQL` domain and stores the backup on the second Data Domain system that was added to the Avamar configuration:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap=password --brtype=full --path=/SQL/SQLServer2
--log=C:\Program Files\avs\var\avsql.log
--ddr=true --ddr-index=2 "SQLServer2\sql02inst2"
```

Backing up multiple instances

To back up one or more instances on a client, but not all instances, specify the instances as separate backup targets between quotation marks at the end of the command.

The following command performs an incremental backup of the local instance and the `sql03inst2` instance on the `SQLServer3` client in the `SQL` domain and stores the backup on the Avamar server:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap="password" --brtype=incremental --path="/SQL/SQLServer3"
--log=C:\Program Files\avs\var\avsql.log
"SQLServer3\sql03inst2" "(local)"
```

Backing up a database

To back up a specific database on a client, specify the client, instance, and database as the backup target between the quotation marks at the end of the command, using the format *client\instance/database*, or `(local)`/*database* if the database is in the local instance.

Performing a differential backup of a database in the local instance

The following command performs a differential backup of the `DB1` database in the local instance on the `SQLServer1` client in the `SQL` domain, labels the backup `DB1Diff`, and stores the backup on the Avamar server:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap=password --brtype=differential --path=/SQL/SQLServer1
--log=C:\Program Files\avs\var\avsql.log "(local)/DB1"
--label=DB1Diff
```

Performing a full backup of a database in a named instance

The following command performs a full backup of the `DB3` database in the `sql01inst2` instance on the `SQLServer1` client in the `SQL` domain and stores the backup on the Avamar server:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap=password --brtype=full --path=/SQL/SQLServer1
--log=C:\Program Files\avs\var\avsql.log
"SQLServer1\sql01inst2/DB3"
```

Backing up multiple databases

To back up multiple databases on a client, but not all databases, specify the instances as separate backup targets between quotation marks at the end of the command.

The following command performs a full backup of the `DB3` database in the `sql01inst2` instance and the `DB1` database in the local instance on the `SQLServer1` client in the `SQL` domain, and stores the backup on a Data Domain system:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap=password --brtype=full --path=/SQL/SQLServer1
--log=C:\Program Files\avs\var\avsql.log --ddr=true
--ddr-index=2 "SQLServer1\sql01inst2/DB3" "(local)/DB1"
```

Backing up data in a failover cluster

To back up data in a failover cluster, run the CLI on the active node with the cluster client, and specify the cluster client as the client to back up. If you are backing up the local instance, then you must use the `--sqlserver` option and specify the virtual server for the SQL Server.

Performing a full backup of a database in the local instance in a cluster

The following command performs a full backup of the `DB1` database in the local instance on the `SQLCluster1` cluster client in the `SQL` domain, labels the backup `DB1Full`, and stores the backup on the Avamar server:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap=password --brtype=full --path=/SQL/SQLCluster1
--log=C:\Program Files\avs\var\avsql.log
--sqlserver=SQLCluster1 --hostname-sql=SQLCluster1
"(local)/DB1" --label=DB1Full
```

Performing a transaction log backup of a database in a named instance in a cluster

The following command performs a transaction log backup of the `DB3` database in the `sql01inst2` instance on the `SQLCluster1` cluster client in the `SQL` domain, and stores the backup on the Avamar server:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap=password --brtype=incremental --path=/SQL/SQLCluster1
--log=C:\Program Files\avs\var\avsql.log
--sqlserver=SQLCluster1 --hostname-sql=SQLCluster1
"SQLCluster1\sql01inst2/DB3"
```

Backing up data in an AlwaysOn availability group

To back up data in an AlwaysOn availability group:

- Run the CLI on the primary replica.
- Specify the cluster client for the availability group listener as the client to back up.
- Specify the availability group listener with the `--sqlserver` and `--hostname-sql` options.
- Specify the availability group listener, the availability group, and the database in quotation marks using the format "*listener\group/database*". To back up all databases in an availability group, specify "*listener\group*".

The following command performs a full backup of all databases on the primary replica using the `SQLGroup2` cluster client for the `SQLGroup2` availability group and the `SQLGroupListener` availability group listener. The `SQLGroup2` cluster client is in the `SQL` domain on the Avamar server, and the backup is stored on the second Data Domain system that was added to the Avamar configuration:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap=password --brtype=full --path=/SQL/SQLGroupListener
--log=C:\Program Files\avs\var\avsql.log
--backup-preference=primary --hostname-sql=SQLGroupListener
--sqlserver=SQLGroupListener --ddr=true --ddr-index=2
"SQLGroupListener\SQLGroup2"
```

Backing up to a Data Domain system

To store a backup on a Data Domain system instead of on the Avamar server, use the `--ddr` and `--ddr-index` options.

The following command performs an incremental backup of all SQL Server data on the `SQLServer2` client in the `SQL` domain and stores the backup on the first Data Domain system that you added to the Avamar configuration:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap=MCUser1 --brtype=incremental --path=/SQL/SQLServer2
--log=C:\Program Files\avs\var\avsql.log --ddr=true
--ddr-index=1 " "
```

Excluding data from a backup

To exclude an instance or database from a backup of an entire client, use the `--exclude` option. If you exclude an instance, you can include an individual database from the instance in the backup using the `--include` option.

Excluding the local instance from a backup of all SQL Server data

The following command performs a full backup of all SQL Server data except for the local instance on the `SQLServer1` client in the `SQL` domain, and stores the backup on the Avamar server with an IP address of `12.34.56.78`:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap=password --brtype=full --path=/SQL/SQLServer1
--log=C:\Program Files\avs\var\avsql.log
--exclude=(local) " "
```

Including a database from an excluded instance in a backup of all SQL Server data

The following command performs the same backup as the previous example, but includes the database named `DB3` in the backup:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap=password --brtype=full --path=/SQL/SQLServer1
--log=C:\Program Files\avs\var\avsql.log
--exclude=(local) --include=(local)/DB3 " "
```

Example restore commands

These command examples illustrate how to use options to perform different types of restores with the `avsql` command.

NOTE: After the restore completes successfully, perform a full backup of the database and clear the **Force incremental backup after full backup** checkbox in the plug-in options for the backup. If the checkbox is selected when a full backup occurs after a restore, then the transaction log backup that occurs automatically after the full backup fails.

Restoring to the original location

To restore a backup to its original location, run the CLI on the original client, use the `--path` option to specify the domain and client, and use the `--labelnum` or `--label` option to specify the backup to restore.

Restoring an entire backup to the original location

The following command performs a tail-log backup and restores the backup labeled `223` to its original location of the `SQLServer3` client in the `SQL` domain:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLServer3 --logtail --labelnum=223
--usesqlreplaceoption --log=C:\Program Files\avs\var\avsql.log
```

Restoring the local instance to the original location

The following command performs a tail-log backup and restores the local instance from the backup labeled `Backup6` to its original location on the `SQLServer3` client in the `SQL` domain:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLServer3 --logtail --label=Backup6
(local) --log=C:\Program Files\avs\var\avsql.log
```

Restoring a database to the original location

The following command performs a tail-log backup and restores the `DB3` database from the `Instance2` instance in the backup labeled `842` to its original location on the `SQLServer3` client in the `SQL` domain:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLServer3 --logtail
--usesqlreplaceoption --labelnum=842 Instance2/DB3
--log=C:\Program Files\avs\var\avsql.log
```

Restoring a filegroup or file to the original location

The following command performs a tail-log backup and restores the following items in the backup labeled `842` to the original location on the `SQLServer3` client in the `SQL` domain:

- `FG1` filegroup for the `DB1` database in the local instance
- `File1` file in the `FG2` filegroup for the `DB2` database in the local instance

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLServer3 --logtail
--labelnum=842 --restore-filegroup-list=(local)/DB1/FG1
(local)/DB2/DB2_FG2_File1
--log=C:\Program Files\avs\var\avsql.log
```

Restoring to a new database in the original instance

To restore a backup to its original location but with a new name, run the CLI on the original client, specify the original database name at the end of the command, and use the `--newdbname` option to specify the new database name.

The following command performs a tail-log backup and then restores the `DB2` database in the local instance from the backup labeled `DB2Full` to its original location of the `SQLServer2` client in the `SQL` domain, but with a new name of `DB6`:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLServer2 --logtail --newdbname=DB6
--log=C:\Program Files\avs\var\avsql.log --label=DB2Full
(local)/DB2
```

Restoring to a different instance

To restore a backup to a different instance, run the CLI on the destination client, use the `--path` option to specify the original domain and client, use the `--target` option to specify the new client and instance, and set `--restore-destination=multiple`.

Restoring a database to a different instance on the same client

The following command restores the `DB3` database from `Instance2` in the backup labeled `239` to an instance named `Instance3` on the `SQLServer1` client:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLServer1
--target=SQLServer1\Instance3 --restore-destination=multiple
--labelnum=239 Instance2/DB3
--log=C:\Program Files\avs\var\avsql.log
```

Restoring multiple databases to a different instance on a different client

The following command restores the `DB3` database from `Instance2` and `DB1` from `Instance1` on the `SQLServer1` client in the `SQL` domain in the backup labeled `SQL1Backup8` to `Instance3` on the `SQLServer2` client:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLServer1
```

```
--target=SQLServer2\Instance3 --label=SQL1Backup8 Instance2/DB3
Instance1/DB1 --log=C:\Program Files\avs\var\avsql.log
```

You must run the previous command from a command line on the SQLServer2 (destination) client.

Restoring files to a different location on a different client

The following command restores the `File1` and `File2` files in `C:\SQLData` from the `(local)/DB3` database on the `SQLServer1` client to `C:\SQL` on the `SQLServer2` client in the `Instance2/DB8` database:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLServer1
--target=SQLServer2\Instance2 --newdbname=DB8
--redirect="C:\SQLData\File1"="C:\SQL\File1",
"C:\SQLData\File2"="C:\SQL\File2" --labelnum=163 (local)/DB3
--log=C:\Program Files\avs\var\avsql.log
```

Restoring to a file

To restore a SQL Server database from an Avamar backup to operating system files, use the `--redirecttofile=true` option and specify the target path for the files by using the `--target` option.

Restoring a database to a file on the original client

The following command restores the `DB4` database in the local instance from the backup with a label number of `423` to the `temp` directory on the `C:\` drive of the original location of the `SQLServer1` client in the `SQL` domain:

```
avsql --operation=restore --id=AvamarAdmin --ap=password
--server=12.34.56.78 --logtail=false -path=/SQL/SQLServer1
--labelnum=423 --redirecttofile=true --target=C:\temp
(local)/DB4 --log=C:\Program Files\avs\var\avsql.log
```

You must run the previous command from a command line on the original client.

Restoring a database to a file on a different client

The following command restores the `DB2` database in `Instance3` from the backup labeled `Backup22` to the `temp` directory on the `C:\` drive of the `WinServer3` client in the `ITServers` domain, when `WinServer3` is not the original client on which the backup was performed:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --label=Backup22 --target=C:\temp
--redirecttofile=true --path=/ITServers/WinServer3
Instance3/DB2 --log=C:\Program Files\avs\var\avsql.log
```

You must run the previous command from a command line on the WinServer3 (destination) client.

Restoring system databases

To restore SQL Server system databases from an Avamar backup, run the CLI on the original client and use the following options:

- `--handle-sql-services-automatically`
- `--restoresystem`
- `--use-sql-replace-option`

In addition, do not use the `--logtail` option.

The following command restores the backup labeled `395` to its original location of the `SQLServer3` client in the `SQL` domain, including the system databases in the backup:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLServer3 --restoresystem
--handle-sql-services-automatically --logtail=false
--use-sql-replace-option --labelnum=395
--log=C:\Program Files\avs\var\avsql.log
```

Restoring to an AlwaysOn availability group

To restore data to an AlwaysOn availability group, run the CLI on the primary replica. The options you use in the command depend on whether you are restoring to the original availability group, a new database in the original availability group, or a different availability group. The steps to restore to a different instance or to operating system files are the same whether you restore from a backup of databases on a stand-alone server, in a cluster, or in an availability group.

Restoring to the original availability group

To restore a backup to its original location in an availability group:

- Use the `--path` option to specify the Avamar domain and the cluster client for the availability group listener.
- Use the `--labelnum` or `--label` option to specify the backup to restore.
- Specify the availability group listener in the `--sqlserver` and `--hostname-sql` options.
- Use the `--recovery-primary-only` option to control whether to restore to only the primary replica or to both the primary and all secondary replicas.

Restoring to the original location on all replicas

The following command performs a tail-log backup and restores the `DB3` database in the backup labeled `842` to its original location on all replicas in the `SQLGroup3` availability group client in the `SQL` domain:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLGroup3
--sqlserver=SQLGroupListener --hostname-sql=SQLGroupListener
--logtail --usesqlreplaceoption
--log=C:\Program Files\avs\var\avsql.log
--recovery-primary-only=false --labelnum=842
SQLGroupListener\SQLGroup3/DB3
```

Restoring to the original location on only the primary replica

The following command performs the same restore as the previous command, but restores to only the primary replica:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLGroup3
--sqlserver=SQLGroupListener --hostname-sql=SQLGroupListener
--logtail --usesqlreplaceoption
--log=C:\Program Files\avs\var\avsql.log
--recovery-primary-only=true --labelnum=842
SQLGroupListener\SQLGroup3/DB3
```

When you restore the databases on only the primary replica, the corresponding database on the secondary replicas is in a restoring state. To restore the databases on the secondary replicas as part of the availability group, manually prepare and restore the databases, and join them to the availability group on the secondary replica.

Restoring to a new database in the original availability group

To restore a backup to its original location but with a new name, specify the original database name at the end of the command, and use the `--newdbname` option to specify the new database name.

The following command performs a tail-log backup and then restores the `DB2` database in the `SQLGroup2` availability group from the backup labeled `DB2Full` to its original location, but with a new name of `DB6`:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLGroup2
--sqlserver=SQLGroupListener --hostname-sql=SQLGroup2Listener
```

```
--logtail --newdbname=DB6
--log=C:\Program Files\avs\var\avsql.log --label=DB2Full
SQLGroupListener\SQLGroup2/DB2
```

Restoring to a different availability group

The following command restores the `DB3` database from the `SQLGroup2` availability group in the backup labeled `239` to an availability group named `SQLGroup3`:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLGroup3
--sqlserver=SQLGroupListener --hostname-sql=SQLGroupListener
--target=SQLGroup3Listener\SQLGroup3
--restore-destination=multiple --labelnum=239
SQLGroupListener\SQLGroup2/DB3
--log=C:\Program Files\avs\var\avsql.log
```

Run the previous command on the primary replica of the target cluster with the `SQLGroup3` availability group.

Point-in-time restore

To restore to a specific point in time, set `--pitormark=pointintime` and specify the date and time with the `--pitormarkstr` option.

To restore to a named mark, set `--pitormark=markname`, specify the named mark using the `--pitormarkstr` option, and specify whether to include the mark in the recovery using the `--recoverypoint` option. If named marks are not unique in the transaction log, then use the `--aftertime` option to specify a date and time to locate the mark.

Restoring to a specific date and time

The following command restores the `DB3` database in the local instance from the backup that is labeled `FullBackup2` to its original location and a point in time of December 15, 2014 at 2:15:45 p.m.:

```
avsql --operation=restore --id=AvamarAdmin --ap=password
--server=12.34.56.78 --path=/SQL/SQLServer1
--log=C:\Program Files\avs\var\avsql.log
--pitormark=pointintime --pitormarkstr=2014-12-15T14:15:45
--label=FullBackup2 (local)/DB3
```

Restoring to a named mark

The following command restores the `DB3` backup in `Instance2` from the backup that is labeled `452` to its original location and a named mark of `M2`, and the recovery includes the transaction at the named mark:

```
avsql --operation=restore --id=AvamarAdmin --ap=password
--server=12.34.56.78 --path=/SQL/SQLServer1
--log=C:\Program Files\avs\var\avsql.log --pitormark=markname
--pitormarkstr=M2 --recoverypoint=atmark --labelnum=452
Instance2/DB3
```

Restore an individual incremental or differential backup

The following command performs an individual restore of SampleDb.

```
avsql --operation=restore --server=12.34.56.78
--id=AvamarAdmin --ap=password --path=/SQL/SQLServer1 --overwrite-default-recoveryplan
--hostname-sql=SQLServer1 --labelnum=9 "SQLServer1\sql01inst2/DB3"
--recoveryoperation=standby --standbyfilelocation="C:\Restordb" --usesqlreplaceoption
--restore-destination=original
```

Discover disk space size requirements for a restore operation

The following command discovers disk space size requirements for DB2 and DB3.

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin --ap=password --path=/SQL/
SQLServer1 --print-restore-size --hostname-sql=SQLServer1 --labelnum=28 "(local)/DB2"
"(local)/DB3"
```

Viewing backup metadata

To view the metadata for a backup as screen output, use the `--metadata` option with the `restore` operation.

The following command displays metadata for the backup with a label number of `22`:

```
avsql --operation=restore --id=AvamarAdmin --ap=password
--server=12.34.56.78 --path=/SQL/SQLServer1 --metadata
--labelnum=22
```

avsqltlr command reference

The `avsqltlr` command enables you to extend the amount of time that a database backup mounted for table level recovery will remain mounted. This operation is only supported when a database backups has already been mounted and the operation will fail if the drive is unmounted.

avsqltlr synopsis

`avsqltlr --extendmnttime=`*time*

avsqltlr options

One option is supported by the `avsqltlr` command, to extend the amount of time that a database backup mounted for table level recovery will remain mounted.

The following option is available for the `avsqltlr` command.

Table 20. `avsqltlr` option

Option	Description
`-- extendmnttime=`*time*	Extends the amount of time that a database backup mounted for table level recovery will remain mounted. Allowed values are: ● "*n* hours" ● "*n* minutes" ● "*n* seconds" The maximum period that period that the database mount can be extended is 24 hours.

Monitoring CLI activities

The status of backups and restores that you perform by using the SQL Server plug-in CLI appears on the **Session Monitor** tab in the **Server** window in Avamar Administrator.

About this task

CLI activities do not appear in the **Activity Monitor** tab in the **Activity** window. The **Activity Monitor** displays only backups and restores that you perform by using Avamar Administrator.

C

Troubleshooting

Topics:

- Backup problems and solutions
- Restore problems and solutions
- Upgrade problems and solutions

Backup problems and solutions

The following topics provide details on how to troubleshoot problems that may occur during backups with the Avamar Plug-in for SQL Server.

Log truncation error appears during a full backup

The following error may appear if you try to perform a full backup with log truncation enabled (by selecting **For all backup types** from the **Truncate database log** list in the SQL Server plug-in options):

```
avsql Error <15754>: SQL TRUNCATE LOG is not supported in SQL Server 2008 or higher,
therefore skipping log truncation
```

To truncate logsr, perform an incremental (transaction log) backup after a full backup automatically by selecting the Force incremental backup after full backup checkbox and leaving the default setting for the **Truncate database log** list.

When the backup process successfully truncates the log file, `LOG_BACKUP` appears in the header of the database.

Error appears for a forced incremental backup after a restore

The following error appears in the Avamar client log file if you perform a full backup with a forced incremental backup after you restore a database:

```
avsql Error <15762>: Skipping incremental backup after full for dbname database. Error:
Last restore was performed after the last full backup.
```

The full backup completes successfully, but Avamar does not perform the incremental backup.

A forced incremental backup contains transactions that occur from the end of the first full backup or the most recent transaction log backup, whichever is later. This action proceeds until the forced incremental backup occurs (after the first backup of the new backup cycle). This step ensures that a point-in-time recovery to a point in time between the two full backups can occur, if required.

If a restore occurs between the two full backups, then the log chain breaks. Avamar cannot create a transaction log backup (forced incremental backup) for the transactions between the backups.

Avamar session security requires prior registration of all clients

Avamar session security requires that all clients be registered with the server. For SQL Always On cluster and SQL virtual cluster, all nodes must be registered with the server.

SQL Auto Close feature not supported

If the SQL Auto Close feature is being used, databases may not be backed up and no error message will be displayed. Event ID 17137, with the message

```
Starting up database database_name
```

will be generated repeatedly in the **Event Monitor** of the Avamar Administrator. This indicates that SQL Auto Close setting is enabled for these databases, which closes the database when the last connection closes. The solution to this problem is to disable the SQL Auto Close feature on the SQL server.

SQL cluster backups fail with no errors in AlwaysOn environment

In case of SQL backup with big databases, consistency check might take more time for which backup from secondary nodes may fail with timeout. To resolve this issue, we should use the following flags in `avagent.cmd` file.

```
--remote-connection-alive-timeout=<timeout>
```

`avagent.cmd` file with above flag should be added to `var` folder as well as `Cluster client's var directory` on the primary node and all the secondary nodes. The `Cluster client's var` folder for AlwaysOn group is created while configuring wccw.

We should also add the following flag in `avsql.cmd` file in `var` folder as well as `Cluster client's var directory` on all the nodes:

```
--secondary-replica-timeout=<timeout>
```

Restore problems and solutions

The following topics provide details on how to troubleshoot problems that may occur during restores with the Avamar Plug-in for SQL Server.

Restore fails with an incorrect error that a log gap occurred

When you restore from an incremental backup of a database on a secondary replica in an AlwaysOn availability group environment, and you select the option to perform a tail-log backup, the restore sometimes fails with an error that a log gap occurred, even though a log gap does not exist. In addition, you cannot perform a point-in-time restore from the backup.

The issue occurs when system tables on the secondary replica are not yet synchronized with other availability replicas at the exact moment when the incremental backup occurs. The synchronization delay may be due to slow network performance or other environmental performance issues.

To work around this issue:

- Perform a full backup to capture the tail of the log.
- Restore the backup to a file, and then use SQL Server tools to restore the data to SQL Server to the required point in time.

Upgrade problems and solutions

The following topic provides details on how to troubleshoot problems that may occur when you upgrade the Avamar client software.

Upgrade requires a reboot

The system may require a reboot after you upgrade the Avamar client software if a file was locked during the upgrade. This issue typically occurs if a backup is in progress when the upgrade starts. Ensure that no backups are in progress when you start the upgrade.

D

Cluster Configuration Tool CLI

Topics:

- Cluster Configuration Tool CLI Overview
- Plugins
- Operations
- Cluster Configuration Tool CLI examples

Cluster Configuration Tool CLI Overview

The Cluster Configuration Tool CLI enables you to use a command line interface (CLI) with the `av_cluster_config_console` command instead of Windows Cluster Configuration Wizard to configure the cluster environment for Avamar backups.

When you use the CLI to configure the cluster, you specify the command options for the `av_cluster_config_console` binary at command prompt. The tool reads the input arguments and configures the cluster.

The `av_cluster_config_console` binary is available in the `C:\Program files\avs\bin` folder, where `C:\Program files\avs\bin` is the Avamar client installation folder.

To use the CLI, open a command prompt, change the directory to the `bin` folder, and then type the command.

Command reference

The `av_cluster_config_console` command enables you to configure the cluster, remove the existing configuration, and view the existing configuration for the SQL and SQLAlwayson plugins.

Synopsis

```
av_cluster_config_console - -plugin= {SQL | SQLAlwayson} - -operation= {configure |
remove | view} [Options]
```

Plugins

The plugin that you specify with the `av_cluster_config_console` command controls the plugin that is used during the configuration.

Supply one of the plugins specified in the following table by using the `- - plugin` option.

Table 21. Plugin descriptions

Plugin	Description
SQL	Configures the SQL failover cluster environment for Avamar backups. The list of supported operations for the SQL plugin includes the following: ● Configure — To configure a new cluster client on all nodes of a SQL fail over cluster. ● Remove — To remove the existing cluster client on all nodes of a SQL fail over cluster. ● View — To view all existing cluster client configuration of a SQL fail over cluster. ● Configure —To configure a new cluster client on all nodes of a SQL fail over cluster.

Table 21. Plugin descriptions (continued)

Plugin	Description
	To get the list of supported operations for this plugin, use the following command: `av_cluster_config_console --help --plugin="SQL"`
SQLAlwayson	Configures the SQL Alwayson environment for Avamar backups. The list of supported operations for the SQL Alwayson environment includes the following: Configure — To configure a new cluster client on all nodes of a specified Alwayson group. Remove — To remove the existing cluster client on all nodes of a specified Alwayson group. View — To view all existing cluster client configuration of a specified Alwayson group. To get the list of supported operations for this plugin, use the following command: `av_cluster_config_console --help --plugin="SQLAlwayson"`

Operations

The operation and options that you specify with the `av_cluster_config_console` command controls the task that the tool performs.

Supply one of the operations in the following table along with the listed options.

Table 22. Operations for the `av_cluster_config_console` command

Operation	Description
Configure	Configures the new cluster client on all the nodes for the specified plugin.
Remove	Removes the existing cluster client on all the nodes for the specified plugin.
View	View all the existing cluster client on all the nodes for the specified plugin.

SQL plugin configure options

The configure option enables you to control the configure functionality for SQL fail over cluster instance for Avamar.

The following table lists all of the mandatory and optional options available for the SQL plugin and configure operation.

To get the list of options available for the SQL plugin and configure operation, type the following command:

`av_cluster_config_console --help --plugin="SQL" --operation="configure"`

All the options that are prefixed with [Optional] are optional for the specified plugin and operation. The options without the [Optional] prefix are mandatory for the specified plugin and operation.

Table 23. SQL Configure Options

Option	Description
mcsaddr	Avamar server address. You can provide either the FQDN name of the server or the IP address. If the server details are wrong then tool cannot configure the environment.
clusterservice	Name of the SQL fail over instance cluster resource group name. To get the list of available cluster resource group names use the `view` command.

Table 23. SQL Configure Options (continued)

Option	Description
[OPTIONAL]sharedvolume	Shared volume for the cluster client. If the fail over instance has a single shared disk then the tool automatically detects the disk and uses it as a shared volume for the configuration. If multiple shared volumes are present, this field is mandatory.
[OPTIONAL]ipversion	IP version that is used on the nodes. If this option is not specified, then IPV4 is used as the default value. IPV4 is supported for the configurations. Only IPV4 is supported for this release.
[OPTIONAL]mcsport	Avamar port number. This is the port that is used to communicate with the Avamar server. If this option is not specified, then 28001 is used as the default port number.
[OPTIONAL]bringonline	Whether to bring the Avamar cluster client online and register with Avamar server. Valid values are "TRUE" or "FALSE". The default value is "TRUE".
[OPTIONAL]clientdomain	Specifies the Avamar domain that the cluster client is associated with. The default value is /clients.
[OPTIONAL]vardir	Avamar cluster client `var` directory. This directory should only reside in the fail over instance shared disk. If not provided by default, the directory is created under `SharedVolume:\Program Files\Backup Agents for Cluster Groups\ClusterClient\var`.

SQL plugin remove options

The SQL plugin remove option removes the existing SQL fail over Avamar cluster client configuration.

The following table lists all of the optional and mandatory options for the SQL plugin and the remove operation.

To get the list of options available for the SQL plugin and remove operation, type the following command:

```
av_cluster_config_console --help --plugin="SQL" --operation="remove"
```

Table 24. SQL remove option

Option	Description
[OPTIONAL]clusterservice	SQL fail over instance cluster resource name. This argument is optional if only a single SQL fail over instance is configured. The tool automatically gets the name of the resource and removes the configuration. If multiple SQL fail over instances are configured, this argument is mandatory. To get the list of existing SQL fail over instance cluster resource names, type the following command: `av_cluster_config_console.exe --plugin="SQL" --operation="view"`

SQLAlwayson plugin configure options

The configure option for the SQLAlwayson plugin enables you to configure SQLAlwayson groups for Avamar backups.

The following table lists the available optional and mandatory options for the SQLAlwayson plugin and configure operation.

To get the list of options available for the SQLAlwayson plugin and configure operation, type the following command:

```
av_cluster_config_console.exe --plugin="SQLAlwayson" --operation="configure" --help
```

Table 25. SQLAlwayson configure options

Option	Description
mcsaddr	Avamar server address. You can provide either the FQDN name of the server or the IP address.
vardir	Avamar cluster var directory. It can be a shared path or a local path.
aongrpname	Name of the alwayson group to be configured. To get the list of alwayson groups available, use the `view` command. `av_cluster_config_console.exe --plugin="SQLAlwayson" --operation="view"`
[OPTIONAL]ipversion	IP version that is being used on the nodes. If this option is not specified, then IPV4 is used as the default value. IPV4 is supported for the configurations.
[OPTIONAL]mcsport	Avamar port number. This is the port that is used to communicate with the Avamar server. If this option is not specified, then 28001 is used as the default port number.
[OPTIONAL]bringonline	Whether to bring the Avamar cluster client online and register with the Avamar server. Valid values are "TRUE" or "FALSE". If this option is not specified, then "TRUE" is used as the default value.
[OPTIONAL]clientdomain	Specifies the Avamar domain that the cluster client is associated with. If this option is not specified, then "/clients" is used as the default value.
[OPTIONAL]sysdir	Avamar cluster sys directory. This is an optional parameter. If this option is not specified, then sysdir is created under the parent directory of vardir.

SQLAlwayson plugin remove options

The remove option for the SQLAlwayson plugin enables you to remove existing SQL alwayson groups that are configured in the Avamar cluster client environment.

The following table lists the available optional and mandatory options for the SQLAlwayson plugin and remove operation.

To get the list of options available for the SQLAlwayson plugin and remove operation, type the following command:

`av_cluster_config_console.exe --plugin="SQLAlwayson" --operation="remove" --help`

Table 26. SQLAlwayson remove options

Option	Description
[OPTIONAL]aongrpname	Name of the SQL Alwayson group to be removed. This is optional if only a single SQL Alwayson group is configured in the environment. This option is mandatory if multiple SQL Alwayson groups are configured.

Table 26. SQLAlwayson remove options

Option	Description
	`av_cluster_config_console.exe --plugin="SQLAlwayson" --operation="view"`

Cluster Configuration Tool CLI examples

Review the `av_cluster_config_console` command examples for details on how to configure, remove, and view operations for different plugins.

View list of supported plugins and operations

The following command returns a list of supported plugins and lists the commands and their usage.

```
av_cluster_config_console.exe --help
```

View list of supported operations for the SQL plugin

The following command returns the list of supported operations for the SQL plugin.

```
av_cluster_config_console.exe --help --plugin="SQL"
```

View list of available options for configure operation for SQL plugin

The following command returns the list of available options for the configure operation for the SQL plugin.

```
av_cluster_config_console.exe --help --plugin="SQL" --operation="configure
```

View list of available options for remove operation for SQL plugin

The following command returns the list of available options for the remove operation for the SQL plugin.

```
av_cluster_config_console.exe --help --plugin="SQL" --operation="remove"
```

Configure new cluster client on all nodes for a SQL fail over instance for Avamar

The following command configures the new cluster client on all nodes for a SQL fail over instance.

```
av_cluster_config_console.exe --plugin=SQL --operation=configure --
mcsaddr="10.31.196.222" --sysdir="C:\avsys\etc" --clusterservice="SQL Server
(FINSTANCE2)" --sharedvolume="H"
```

Remove existing cluster client on all nodes for a SQL fail over instance for Avamar

The following command removes the existing cluster client on all nodes for a SQL fail over instance.

```
av_cluster_config_console.exe --plugin=SQL --operation=remove --clusterservice="SQL
Server (FINSTANCE2)"
```

View the existing cluster client information for the SQL fail over configuration

The following command displays the existing cluster client configuration for all the available SQL fail over instances.

```
av_cluster_config_console.exe --plugin=SQL --operation=view
```

View list of supported operations for SQLAlwayson plugin

The following command returns the list of supported operations for the SQLAlwayson plugin.

```
av_cluster_config_console.exe --help --plugin=" SQLAlwayson "
```

View list of available options for the configure operation for SQLAlwayson plugin

The following command returns the list of available options for the configure operation for the SQL plugin.

```
av_cluster_config_console.exe --help --plugin=" SQLAlwayson " --operation="configure"
```

View list of available options for the remove operation for SQLAlwayson plugin

The following command returns the list of available options for the remove operation for the SQL plugin.

```
av_cluster_config_console.exe --help --plugin=" SQLAlwayson " --operation="remove"
```

Configure new cluster client on all nodes for SQLAlwayson group Grp1 for Avamar

The following command configures the new cluster client on all nodes for an SQLAlwayson group Grp1.

```
av_cluster_config_console.exe --plugin=sqlalwayson --operation=configure --
mcsaddr="10.31.197.22" --vardir="\\node1\sharevar" --aongrpname="Grp1" --clientdomain="/
AgGroups" --bringonline="false"
```

Remove the existing cluster client on all nodes for an SQLAlwayson fail over instance for Avamar

The following command removes the existing cluster client on all nodes for an existing SQLAlwayson group.

```
av_cluster_config_console.exe --plugin=SQLAlwayson --operation=remove
```

View the existing cluster client information for the SQL Alwayson configuration

The following command displays the existing cluster client configuration for all the available SQL Alwayson groups.

```
av_cluster_config_console.exe --plugin=SQLAlwayson --operation=view
```

E

ItemPoint for Microsoft SQL Server Notes

Topics:

- Supported environments for ItemPoint for Microsoft SQL Server
- Limitations for ItemPoint for Microsoft SQL Server

Supported environments for ItemPoint for Microsoft SQL Server

The following table lists the supported environments for ItemPoint for Microsoft SQL Server:

Table 27. Component requirements

Component	Requirement
Operating Systems	The following operating systems are supported: ● Windows Server 2008 ● Windows Server 2008 R2 ● Windows Server 2012 ● Windows Server 2012 R2 ● Windows 7 ● Windows 8 **NOTE:** 64-bit versions, virtual and physical, of the listed operating systems are required. ItemPoint for Microsoft SQL Server can only be run by users with administrative privileges and in administrative mode.
Microsoft SQL Server	Source File Formats: ● Native SQL backup ○ Full, differential, and incremental (transaction log) backups ○ Compressed and uncompressed ● Offline SQL database files (.MDF/.NDF/.LDF) Source Data: ● SQL 2005 ● SQL 2008 ● SQL 2008 R2 ● SQL 2012 ● SQL 2014 ● SQL 2016 Target Servers: ● SQL 2005 ● SQL 2008 ● SQL 2008 R2 ● SQL 2012

Table 27. Component requirements (continued)

Component	Requirement
	• SQL 2014 • SQL 2016 Remote Blob Stores (RBSs): • FILESTREAM Provider • Metalogix StoragePoint Provider (v4.2 1320 or higher) • StorSimple SharePoint Database Optimizer
Microsoft Azure	Source Data: • Local backups of Microsoft Azure • Downloaded copies of database files • Downloaded copies of backup files
Virtual Environments	Virtual operation of tape devices may have restrictions that are imposed by virtual operating systems.
Additional Software	Microsoft .NET Framework: • 4.0 In order for ItemPoint for Microsoft SQL Server to operate fully, you must ensure that the source database files are on EMC storage. If not, attempts to open a source database produces an error message.

Limitations for ItemPoint for Microsoft SQL Server

The following are not supported in ItemPoint for SQL:

- Database encryption and compression
- Backup data encryption
- SQL Server database objects other than tables, rows, and columns (such as system tables, SQL Server database objects other than tables, rows, and columns (such as system tables, stored procedures, triggers, primary and foreign keys, and CLR) are not supported and do not appear in the ItemPoint for Microsoft SQL Server interface.
- Limited support column (data) types:

  NOTE: These column types appear in the Source List as either <UNSUPPORTED> or binary and are copied as binary.

  - CLR UDT (User Defined Type)
  - Cursor
  - Date, DateTimeOffset, DateTime2, Time
  - Geography, Geometry
  - HierarchyId
  - TimeStamp/RowVersion
- Unsupported column (data) types:
  - COLUMN_SET column

    NOTE: While sparse columns defined by COLUMN_SET column appear in the Source List, tables containing this column cannot be copied.
  - Filestream

    NOTE: Tables containing Filestream columns cannot be accessed.
- Unsupported backup types:
  - Transaction Log
  - Files and filegroup

  NOTE: Memory-Optimized tables cannot be accessed.
- Columnstore indexes

  NOTE: Tables with a columnstore index cannot be accessed.

- Restoring data
  - Referential integrity, indexes, and permissions are not restored.
  - When restoring tables with TimeStamp columns, the TimeStamp column is not preserved. The value is the time of the restoration for that row. This is the normal behavior for TimeStamp columns.
  - Restoring tables with identity columns does not preserve the value of the identity column; it follows the rules of the identity column on the target table.

If the source table has three rows with identity values of 1, 3, 5, once restored, the target will have values of 1, 2, 3. (This step is the normal behavior of identity columns.)

Glossary

A

activation
The process of passing the client ID (CID) back to the client, where it is stored in an encrypted file on the client file system.

See also client activation

administrator
Person who normally installs, configures, and maintains software on network computers, and who adds users and defines user privileges.

alias
A SQL Server feature that enables you to configure a name on a computer with SQL Server that points to a server, instance, or database on either the local computer or on a different computer.

AlwaysOn availability group
In a cluster environment with SQL Server 2012, 2014, and 2016, a group of databases stored locally on each node that can fail over from one node to another for high availability.

See also availability group

authentication
The method by which users and applications, such as Avamar client software, connect to SQL Server. Authentication methods include *Windows authentication* and *SQL Server authentication*.

avagent
The name of the *client agent* process.

availability group
In a cluster environment with SQL Server 2012, 2014, or 2016, a group of databases stored locally on each node that can fail over from one node to another for high availability.

See also AlwaysOn availability group

availability group listener
A virtual network name (VNN) to which clients can connect to access a database in an availability replica of an availability group.

availability replica
A SQL Server instance on a cluster node that hosts an *AlwaysOn availability group*.

See also replica

Avamar Administrator
A graphical management console software application that is used to remotely administer an Avamar system from a supported Windows or Linux client computer.

Avamar client
A computer or workstation that runs Avamar software and accesses the Avamar server over a network connection. Avamar client software comprises a *client agent* and one or more *plug-ins*.

Avamar server
The server component of the Avamar client/server system. Avamar server is a fault-tolerant, high-availability system that efficiently stores the backups from all protected clients. It also provides essential processes and services required for data restores, client access, and remote system administration. Avamar server runs as a distributed application across multiple networked storage nodes.

avsql
The application executable and command line interface for the Avamar Plug-in for SQL Server. You can use avsql to perform on-demand backups and restores of SQL Server data.

avtar
The Avamar process that performs backups and restores.

B

backup
A point-in-time copy of client data that can be restored as individual files, selected data, or as an entire backup.

backup policy
In the AUI, a backup policy specifies a dataset, schedule, and retention settings that are applied to a client or a group of clients. A backup policy must contain at least one Avamar client. If the backup policy contains two or more clients, the clients must belong to the same Avamar domain. You can override backup policy settings at the client level.

bare metal recovery (BMR)
The process of restoring data from a backup to recover a computer system from “bare metal,” or without previously installed software or operating system.

browse
The process of viewing data that is available for backup on a client computer or restore from the Avamar server.

bulk-logged recovery model
A type of SQL Server *recovery model* that permits high-performance bulk copy operations and reduces log space usage by using minimal logging for most bulk operations. This recovery model enables you to recover to the end of any backup. *Point-in-time restore* is not supported.

C

client activation
The process of passing the client ID (CID) back to the client, where it is stored in an encrypted file on the client file system.

See also activation

client agent
A platform-specific software process that runs on the client and communicates with the Management Console Server (MCS) and with any plug-ins installed on that client.

client registration
The process of establishing an identity with the Avamar server. When Avamar recognizes the client, it assigns a unique client ID (CID), which it passes back to the client during *client activation*.

See also registration

cluster
Two or more independent network servers, usually with exact copies of the same software and data, that operate and appear to clients as if they are a single unit. A cluster configuration enables work to be shifted from one server to another, providing high availability, which allows application services to continue despite most hardware or software failures.

Cluster Configuration Tool
Avamar configuration wizard to configure the cluster client.

D

database
A collection of data arranged for ease and speed of update, search, and retrieval by computer software.

database mirroring
A high availability strategy where SQL Server maintains two copies of a single database that must reside on different SQL Server instances.

See also mirroring

Data Domain system
Disk-based deduplication appliances and gateways that provide data protection and disaster recovery (DR) in the enterprise environment.

dataset
A policy that defines a set of files, directories, and file systems for each supported platform that are included or excluded in backups across a group of clients. A dataset is a persistent and reusable Avamar policy that can be named and attached to multiple groups.

default instance
The default *instance* on a computer with SQL Server. The name of the default instance is the name of the local computer.

See also local instance

differential backup
A type of backup that includes only data that changed after the last *full backup*. The backup also copies relevant portions of the transaction logs.

disaster recovery
Recovery from any disruptive situation, such as hardware failure or software corruption, in which ordinary data recovery procedures are not sufficient to restore a system and its data to normal day-to-day operations. A disaster recovery can be a *bare metal recovery*.

DNS
Domain Name Server. A dynamic and distributed directory service for assigning domain names to specific IP addresses.

domain
A feature in Avamar Administrator that is used to organize large numbers of clients into named areas of control and management.

F

Failover Cluster Instances (FCIs)
A single *instance* of SQL Server that is installed across Windows Server Failover Clustering (WSFC) nodes. On the network, an FCI appears to be an instance of SQL Server running on a single computer, but the FCI provides failover from one WSFC node to another if the current node becomes unavailable.

filegroup
Named collections of SQL Server files that are used to help with data placement and administrative tasks such as backup and restore operations. The two types of filegroups are the *primary filegroup* and *user-defined filegroups*.

file system
Software interface used to save, retrieve, and manage files on storage media by providing directory structures, data transfer methods, and file association.

full backup
A type of backup that includes the entire database, including all objects, system tables, and data, as well as the transaction log.

full recovery model
A type of SQL Server *recovery model* that logs all database transactions in the transaction log. This type of recovery model enables you to perform a point-in-time restore.

G

group
A level of organization in Avamar Administrator for one or more Avamar clients. All clients in an Avamar group use the same group policies, which include the *dataset*, *schedule*, and *retention policy*.

group policy
In Avamar Administration, a group policy is defined as a *dataset*, *schedule*, and *retention policy* for all clients in an Avamar group.

H

hybrid IT environment
An information technology (IT) environment where an enterprise provides and manages some resources on-premise but uses cloud-based services for other resources.

I

incremental backup
A type of backup that includes only the transaction log. Also called a *transaction log backup*.

instance
A single occurrence of the SQL Server Database Engine. You can install multiple instances on a single computer. The two types of instances are the *local instance*, also called the *default instance*, and *named instances*.

L

LAN
Local Area Network.

local instance
The default *instance* on a computer with SQL Server. The name of the local instance is the name of the local computer.

See also default instance

M

mark
The user-defined name for a starting point of a specific type of transaction in the transaction log. You can perform a *point-in-time restore* for a database to a mark.

See also named mark

master database
The *system database* that records all of the system level information for a SQL Server system.

MCS
Management console server. The server subsystem that provides centralized administration (scheduling, monitoring, and management) for the Avamar server. The MCS also runs the server-side processes used by *Avamar Administrator*.

metadata
Data about the backup, including information about the original database files, the backup types, the date and time of the backup, and other information necessary for restore.

Microsoft Azure
The Microsoft cloud computing platform and infrastructure. Formerly known as Windows Azure.

mirroring
A high availability strategy where SQL Server maintains two copies of a single database that must reside on different SQL Server instances.

See also database mirroring

model database
The *system database* that provides the template for all databases that you create on a SQL Server *instance*.

msdb database
The *system database* for scheduling alerts and jobs, and for recording operators. The msdb database also contains history tables, such as the backup and restore history tables.

multi-streaming
The process of backing up or restoring data using multiple parallel data streams. Multi-streaming enables you to improve backup and restore performance in most environments.

N

named instance
All instances of the database engine other than the *local instance*. You specify the name of the instance when you install the instance.

named mark
The user-defined name for a starting point of a specific type of transaction in the transaction log. You can perform a *point-in-time restore* for a database to a named mark.

See also mark

P

plug-in
Avamar client software that recognizes a particular kind of data resident on that client.

plug-in options
Options that you specify during backup or restore to control backup or restore functionality.

point-in-time restore
The process of restoring a database with the full recovery model to a specific date and time or to a *named mark* in the transaction log.

policy
A set of rules for client backups that can be named and applied to multiple groups. Groups have dataset, schedule, and retention policies.

primary data file
The data file that is the starting point of the database and points to the other files in the database. Every database has one primary data file. The recommended filename extension for primary data files is `.mdf`.

primary filegroup
The *filegroup* that contains the primary data file and any other files that are not specifically assigned to another filegroup. All pages for the system tables are allocated in the primary filegroup.

primary replica
The *availability replica* that handles read/write activity from clients and sends transaction log updates to the secondary replicas.

R

recovery model
A database property that controls how transactions are logged, whether the transaction log requires and allows backups, and the available restore operations. Three recovery models exist: simple, full, and bulk-logged.

recovery operation
The type of recovery to perform when you restore SQL Server data. The recovery operation controls the status of restored databases after the restore. The three recovery models are `RECOVERY`, `NORECOVERY`, and `STANDBY`.

redirected restore
The process of restoring a backup to a different location than the original location where the backup occurred.

registration
The process of establishing an identity with the Avamar server. When Avamar recognizes the client, it assigns a unique client ID (CID), which it passes back to the client during *client activation*.

See also client registration

replica
A SQL Server instance on a cluster node that hosts an *AlwaysOn availability group*.

See also availability replica

Resource database
A read-only *system database* that contains copies of all system objects that ship with SQL Server. The Avamar Plug-in for SQL Server cannot back up the Resource database because SQL Server cannot back up the Resource database.

restore
An operation that retrieves one or more file systems, directories, files, or data objects from a backup and writes the data to a designated location.

retention
The time setting to automatically delete backups on an Avamar server. Retention can be set to permanent for backups that should not be deleted from an Avamar server. Retention is a persistent and reusable Avamar policy that can be named and attached to multiple groups.

S

schedule
The ability to control the frequency and the start and end time each day for backups of clients in a group. A schedule is a persistent and reusable Avamar policy that can be named and attached to multiple groups.

secondary data files
The data files other than the *primary data file* for a database. Some databases may not have any secondary data files, while others have several secondary data files. The recommended filename extension for secondary data files is `.ndf`.

secondary replica
An *availability replica* that contains a copy of the databases for the availability group. The database copies synchronize with the databases on the primary replica through transaction log updates from the primary replica.

simple recovery model
A type of SQL Server *recovery model* that permits high-performance bulk copy operations and reclaims log space to minimize space requirements. This recovery model enables you to recover to the end of any backup. Databases with the simple recovery model do not support transaction log backups or point-in-time restore.

sqlcmd utility
A command line utility for entering *Transact-SQL* statements and scripts.

SQL Server authentication
An *authentication* method by which users and applications, such as Avamar client software, connect to SQL Server using a login account defined in SQL Server.

SQL Server Management Studio (SSMS)
A software application with script editors and graphical tools to configure, manage, and administer SQL Server components.

system databases
Databases that store system-level information, objects, and templates for a SQL Server *instance*. The master, msdb, model, and Resource databases are system databases.

T

tail-log backup
A transaction log backup of log records that have not yet been backed up (the “tail of the log”). A tail-log backup occurs before you restore data from a backup. The backup prevents work loss and keeps the log chain intact.

transaction log
A log file that contains a history of database transactions.

transaction log backup
A type of backup that includes only the transaction log. Also called an *incremental backup*.

Transact-SQL
An extension of Structured Query Language (SQL). You can use Transact-SQL statements to back up and restore SQL Server data.

U

user-defined filegroup
Any *filegroup* that you define in SQL Server. Log files are never part of a filegroup.

V

virtual device interface (VDI)
The Microsoft application programming interface (API) that the Avamar Plug-in for SQL Server uses to back up and restore SQL Server data.

virtual machine (VM)
A computer that is a software implementation of a computer. Virtual machines are used to run different operating systems at the same time on one physical computer. Each operating system runs in its own isolated execution environment.

W

Windows authentication
An *authentication method* by which users and applications, such as Avamar client software, connect to SQL Server using a Windows user account that has privileges in SQL Server.