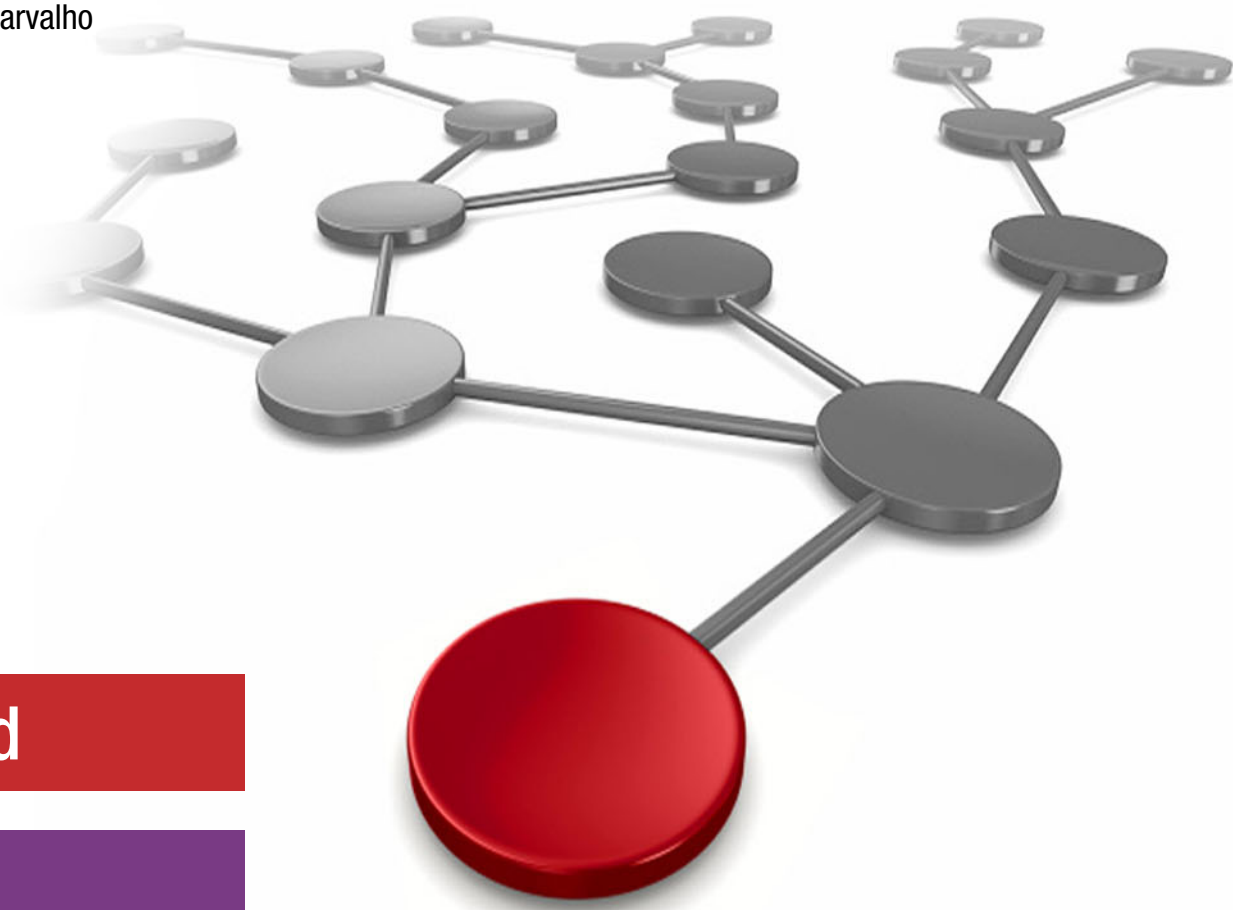


IBM DS8000 Transparent Cloud Tiering

DS8000 Release 9.3

Alexander Warmuth

Nielson "Nino" de Carvalho



 Cloud

Storage



IBM Redbooks

**IBM DS8000 Transparent Cloud Tiering: DS8000
Release 9.3**

August 2023

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

Sixth Edition (August 2023)

This edition applies to IBM DS8900F storage systems with IBM DS8000 Licensed Machine Code (LMC) 7.9.30 (bundle version 89.30.xx.x), referred to as Release 9.3.

© Copyright International Business Machines Corporation 2017, 2022, 2023. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	vii
Trademarks	viii
Preface	ix
Authors	x
Now you can become a published author, too!	x
Comments welcome	x
Stay connected to IBM Redbooks	xi
Summary of changes	xiii
August 2023	xiii
Part 1. Introducing Transparent Cloud Tiering and cloud	1
Chapter 1. Storage tiering history and the value of Transparent Cloud Tiering	3
1.1 Storage tiers	4
1.2 Hardware layers overview	4
1.3 Software layers	5
1.4 DFSMSshm behavior without TCT	7
1.5 Introducing Transparent Cloud Tiering	8
Chapter 2. Cloud overview	11
2.1 What defines a cloud in the context of TCT	12
2.2 Compute cloud versus a storage cloud	13
2.3 Types of storage	13
2.4 Cloud storage delivery models	14
2.5 Object storage hierarchy	16
2.5.1 Storage cloud hierarchy	16
2.5.2 Metadata	18
Chapter 3. Transparent Cloud Tiering	19
3.1 Transparent Cloud Tiering overview	20
3.2 Transparent Cloud Tiering data flow	21
3.2.1 DS8000 cloud connection	21
3.2.2 DFSMS cloud connection	22
3.3 Storing and retrieving data by using DFSMS	24
3.4 Transparent Cloud Tiering and disaster recovery	26
3.5 Transparent Cloud Tiering and DS8000 Copy Services	27
3.5.1 IBM FlashCopy	27
3.5.2 Metro Mirror	28
3.5.3 Metro Mirror with HyperSwap	29
3.5.4 Global Mirror	30
3.5.5 Multi-site data replication	30
3.6 DS8900F multi-cloud support	30
3.7 Transparent Cloud Tiering encryption	31
3.8 Transparent Cloud Tiering secure data transfer with TS7700	32
3.9 Transparent Cloud Tiering compression with TS7700	32
3.10 Selecting data to store in a cloud	33
Part 2. Setting up Transparent Cloud Tiering	35

Chapter 4. Requirements	37
4.1 Ethernet connections on DS8000	38
4.2 z/OS level	40
4.3 IBM zSystems host system hardware requirements	41
4.3.1 The CP Assist for Cryptographic Function feature	41
4.3.2 IBM Crypto Express feature	41
4.4 DS8000 release level	41
4.5 TS7700 release level and features	42
4.6 Authentication information	43
4.6.1 Endpoint	43
4.6.2 Cloud credentials	43
4.6.3 Certificates (if using SSL/TLS)	44
4.7 SSL/TLS considerations	44
Chapter 5. Connectivity and network setup for Transparent Cloud Tiering	45
5.1 Networking communication overview	46
5.2 DS8000 to object storage connection	46
5.2.1 Connecting to a TS7700	47
5.2.2 Connecting to a local cloud object store	47
5.2.3 Connecting to a public cloud object storage service	48
Chapter 6. Configuring the IBM DS8000 for Transparent Cloud Tiering	49
6.1 Configuring the IBM DS8000 for TCT	50
6.1.1 Ethernet configuration	50
6.1.2 Cloud configuration	52
6.2 Maintaining the cloud server configuration	59
6.2.1 Listing a cloud server configuration	59
6.2.2 Updating or removing a cloud server configuration	60
6.3 Managing multiple cloud connections	60
6.4 Preparing the DS8000 as cloud proxy	61
6.4.1 Configuring the DS8000 for the REST API proxy function	62
Chapter 7. Configuring Data Facility Storage Management Subsystem for Transparent Cloud Tiering	65
7.1 DFSMS connections to cloud	66
7.2 Adding digital certificates to IBM Resource Access Control Facility	66
7.2.1 Uploading the certificate files to the z/OS host	66
7.2.2 Importing the certificates to RACF	68
7.3 Controlling access to the DFSMS cloud features	69
7.3.1 Controlling access to DFSMSdss	69
7.3.2 Controlling access to DFSMSHsm	69
7.4 Creating a DFSMS cloud network connection by using ISMF	70
7.4.1 Defining a cloud network connection for Amazon S3, IBM Cloud Object Storage, or TS7700 cloud targets	71
7.4.2 Defining a cloud network connection for a Swift cloud object storage target	73
7.4.3 Using NaviQuest to manage DFSMS network connections	74
7.4.4 Activating the Storage Management Subsystem configuration	75
7.5 Enabling TCT compression with a TS7700	77
7.6 Using TS7700 object policies	78
Chapter 8. Managing cloud credentials	79
8.1 Managing credentials by using Cloud Data Access	80
8.1.1 Preparing the Cloud Data Access configuration	81
8.1.2 Configuring the CSFKEYS general resource class	81

8.1.3	Entering and storing the cloud provider credentials	81
8.1.4	Configuring CDA to run without Crypto Express adapters	83
8.1.5	Deleting the cloud provider credentials	84
8.1.6	Troubleshooting	85
8.1.7	Using the CDA credentials	86
8.2	Legacy method to manage cloud passwords for HSM	86
Part 3.	Operation and usage	89
Chapter 9.	DFSMSHsm	91
9.1	Cloud use overview	92
9.2	Cloud container management	92
9.3	Object management	93
9.3.1	Migration	94
9.3.2	Recall	94
9.4	Fast subsequent migration	94
9.5	Migration update and considerations	95
9.5.1	Command-driven migration	95
9.5.2	Automatic migration	97
9.5.3	CPU utilization considerations	98
9.6	Recall considerations	98
9.7	LIST command updates	98
9.8	Auditing	100
9.9	REPORT command	101
Chapter 10.	Using automatic migration	103
10.1	SMS support for automatic migration	104
10.1.1	Management Class updates	104
10.2	Storage Group affinity enhancements	106
10.3	Space management functions	107
Chapter 11.	Operational integration and reporting considerations	109
11.1	Pre-implementation reporting	110
11.2	Operational monitoring	110
11.2.1	Monitoring cloud setting changes	110
11.2.2	Monitoring migration activities	111
11.2.3	Monitoring reconnections	111
11.2.4	Other messages to consider	111
11.3	Operational reporting	112
11.3.1	Building reports	112
11.3.2	DCOLLECT reports	113
Chapter 12.	DFSMSHsm full volume dump	115
12.1	DFSMSHsm full volume dump for Transparent Cloud Tiering overview	116
12.1.1	DFSMSHsm full volume dump use cases	116
12.1.2	The DUMPCLASS attribute and how it works	116
12.1.3	Containers	117
12.1.4	Objects	118
12.1.5	Automatic dump	118
12.1.6	Expiring dump copies	119
12.1.7	Deleting empty dump containers	119
12.2	Setting up a DFSMSHsm full volume dump to cloud object storage	120
12.2.1	Defining a DFSMS cloud network	120
12.2.2	Defining DUMPCLASS	120

12.2.3	Defining a copy pool	121
12.3	Dumping to cloud object storage.	121
12.3.1	Command-driven dump.	121
12.3.2	Automatic dump	122
12.4	Restoring from cloud object storage	122
12.4.1	RECOVER	122
12.5	Additional operational commands	123
12.5.1	DDELETE	123
12.5.2	LIST	124
Chapter 13. Data Facility Storage Management Subsystem full volume dump		125
13.1	FVD for TCT overview.	126
13.1.1	FVD use cases and how it works	126
13.2	Creating an FVD to cloud object storage	128
13.3	Restoring an FVD from cloud object storage	129
13.4	Managing FVD backups	130
13.4.1	Use case example	131
Appendix A. DFSMSHsm enhancements for Transparent Cloud Tiering		133
	Large number of data sets	134
	Performance improvements	136
	Asynchronous object deletion.	136
	Additional performance improvements	136
	Other considerations	137
Appendix B. Exporting the IBM DS8000 certificate chain		139
	Certificate export with Google Chrome or Microsoft Edge	140
	Certificate export with Firefox.	142
Appendix C. Replacing communication certificates in the IBM DS8900F		145
	Installing a CA-signed certificate by using the Storage Manager GUI	146
	Navigating to the Communication Certificates window	146
	Creating a certificate signing request	147
	Importing the signed certificate	148
	Installing a CA-signed certificate by using the service web interface.	149
	Navigating to the service WUI.	149
	Creating a CSR	151
	Importing the signed certificate	152
	Creating a self-signed certificate on the DS8000.	153
Abbreviations and acronyms		155
Related publications		157
	IBM Redbooks	157
	Online resources	157
	Help from IBM	157

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

AIX®	GDPS®	Power®
Db2®	Guardium®	RACF®
DS8000®	HyperSwap®	Redbooks®
Easy Tier®	IBM®	Redbooks (logo)  ®
FICON®	IBM Cloud®	z/OS®
FlashCopy®	IBM Security®	z/VM®

The following terms are trademarks of other companies:

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Preface

This IBM® Redbooks® publication gives a broad understanding of storage clouds and the initial functions that were introduced for mainframe data to be transferred to cloud storage.

IBM Data Facility Storage Management Subsystem (DFSMS) and the IBM DS8000® added functions to provide elements of serverless data movement, and for IBM z/OS® to communicate with a storage cloud. The function is known as *Transparent Cloud Tiering* (TCT) and is composed of the following key elements:

- ▶ A gateway in the DS8000, which allows the movement of data to and from Object Storage by using a network connection.
- ▶ DFSMSHsm enhancements to support Migrate and Recall functions to and from the Object Storage. Other commands were enhanced to monitor and report on the new functions.
- ▶ DFSMSHsm uses the Web Enablement toolkit for z/OS to create and access the metadata for specific clouds, containers, and objects.
- ▶ DFSMSdss enhancements to provide some basic backup and restore functions to and from the cloud. The IBM TS7700 can also be set up to act as though it is cloud storage from the DS8000 perspective.

This IBM Redbooks publication is divided into the following parts:

- ▶ Part 1, “Introducing Transparent Cloud Tiering and cloud” on page 1 provides you with an introduction to clouds. It provides basic knowledge and terminology.
- ▶ Part 2, “Setting up Transparent Cloud Tiering” on page 35 shows you how we set up the TCT in a controlled laboratory and how the new functions work. We provide points to consider to help you set up your storage cloud, including network connectivity, and integrate it into your operational environment.
- ▶ Part 3, “Operation and usage” on page 89 shows you how we used the new functions to communicate with the cloud and to send data to it and retrieve data from it.

This edition applies to DS8900F Release 9.3 and covers more recent features of TCT such as multi-cloud connections, along with extra advice for high availability (HA) cloud connectivity and DFSMSHsm improvements.

Authors

This book was produced by a team of specialists from around the world:

Alexander Warmuth is a Consulting IT Specialist at the IBM European Storage Competence Center. Working in technical sales support, he designs and promotes new and complex storage solutions, drives the introduction of new products and provides advice to customers, IBM Business Partners, and sales. His main areas of expertise are high-end storage solutions and business resilience for IBM zSystems and Linux. He joined IBM in 1993. Alexander holds a diploma in Electrical Engineering from the University of Erlangen, Germany.

Nielson “Nino” de Carvalho is a Level 2 certified IT specialist at IBM Lab Services in South Africa. He has over 10 years experience of IBM mainframe computing as a customer and with IBM. Nino has extensive technical experience implementing and supporting customers on a broad range of IBM products. His areas of expertise include IBM zSystems and LinuxONE hardware, z/OS and IBM z/VM®, and high-end disk, and tape solutions, including IBM FICON® connectivity.

Thanks to the previous authors of this publication:

Bertrand DufRASne, Monica Falcone, Robert Gensler, and Bjoern Wesselbaum.

We also thank John Thompson, Tabor Powelson, Robert2 Gensler, Yufen Davis and Bjoern Wesselbaum for their efforts to review this publication.

Now you can become a published author, too!

Here’s an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an IBM Redbooks residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- ▶ Use the online **Contact us** review Redbooks form found at:
ibm.com/redbooks
- ▶ Send your comments in an email to:
redbooks@us.ibm.com

- ▶ Mail your comments to:
IBM Corporation, IBM Redbooks
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- ▶ Find us on LinkedIn:
<http://www.linkedin.com/groups?home=&gid=2130806>
- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ Stay current on recent Redbooks publications with RSS Feeds:
<http://www.redbooks.ibm.com/rss.html>

Summary of changes

This section describes the technical changes that were made in this edition of the book. This edition might also include minor corrections and editorial changes that are not identified below.

Summary of Changes for SG24-8381-05
for IBM DS8000 and Transparent Cloud Tiering (DS8000 Release 9.1)
as created or updated on August 11, 2023.

August 2023

This revision reflects the addition, deletion, or modification of new and changed information.

Changed information

► Sixth Edition

- Updates and corrections regarding networking and HyperSwap.

► Fifth Edition:

- Support for HSM full volume dump and restore.
- Support for DFSMS dataset level dump and restore.

► Fourth Edition:

- Networking considerations.
- Support for DFSMS full volume dump and restore.
- Cloud credentials handling with CDACREDS.
- TS7700 Advanced Object Store for DS8000.
- DS8000 multi-cloud support.
- Support for Guardium Key Lifecycle Manager (GKLM) Containerized Edition as the key manager for TCT encryption.
- HSM tuning tips.
- Updated language used in document to clarify support and usage with the S3 protocol.



Part 1

Introducing Transparent Cloud Tiering and cloud

In this part, we introduce topics that are related to a storage cloud. The aim is to provide a basic understanding of storage clouds. If you are new to clouds, it is important you read this section.

This part includes the following chapters:

- ▶ Chapter 1, “Storage tiering history and the value of Transparent Cloud Tiering” on page 3
- ▶ Chapter 2, “Cloud overview” on page 11
- ▶ Chapter 3, “Transparent Cloud Tiering” on page 19



Storage tiering history and the value of Transparent Cloud Tiering

This chapter introduces storage tiering and Transparent Cloud Tiering (TCT) with the IBM DS8000 and its potential benefits for data storage needs.

Traditional management of data across tiers is available across several devices at your site. We included this topic to review what might already be in place in your environment and why you might want to use cloud object storage as a tiering option based on TCT.

This chapter includes the following topics:

- ▶ 1.1, “Storage tiers” on page 4
- ▶ 1.2, “Hardware layers overview” on page 4
- ▶ 1.3, “Software layers” on page 5
- ▶ 1.4, “DFSMSHsm behavior without TCT” on page 7
- ▶ 1.5, “Introducing Transparent Cloud Tiering” on page 8

1.1 Storage tiers

Systems have a finite amount of resources that can be used to store data, whether online or with auxiliary storage. The use of different storage media and categorizing the data within each layer can be an efficient way to manage storage resources. This management allows critical data to be available in high-performance devices, and other data on lower-cost devices.

There are two solutions available for storage tiering: Hardware and software implementation. Each technique can be used alone, or combined, to provide improved data management, and storage efficiency.

The next two sections introduce the concepts behind hardware and software tiering.

1.2 Hardware layers overview

The hardware layers consist of the following storage areas:

- ▶ Custom Flash technologies
- ▶ Solid-state drive (SSD)
- ▶ Enterprise
- ▶ Nearline
- ▶ Tape

Storage media that is used in mainframe systems has changed dramatically, from drives that can store a few megabytes, to current drives that can store terabytes, or flash technologies that provide increased performance gains over spinning disk drives.

Former DS8000 systems might be configured with a mix of flash cards, SSD, Enterprise, and Nearline media and managed by IBM Easy Tier®. This approach enables the system hardware to constantly monitor data extents (or track group, for small extents). Recent models, such as some of the DS8880 and the DS8900F devices, come as all flash configurations only, but can still benefit from Easy Tier to take advantage of different flash tiers and rebalance extents within a rank.

Another layer that is available under storage hardware management includes offline media. Offline media includes any media that cannot be directly accessed by a computer, unless it is made available to the system.

Virtual and physical tapes are considered offline media, and can also be part of hardware storage layers. Although the direct access storage device's hardware cannot directly migrate data to tapes, you can use software tiers to accomplish this migration. Also, Virtual Tape Libraries use disk storage to emulate tapes, which are offloaded to physical tapes when the cache is full. This process is known as pre-migration.

When the requested data is stored in physical tapes, they are mounted, and the virtual tape content is loaded into the Virtual Tape Library and made available for z/OS read. The process of recovering data from physical tapes to cache is also known as recall.

1.3 Software layers

Unlike hardware tiers, there are two types of storage devices available from a z/OS perspective: disk and tape. It is necessary to have a storage management product, such as DFSMSHsm, to enable the use of software tiers. DFSMSHsm can manage data by migrating, recalling, backing up, and recovering data sets as required.

DFSMSHsm can manage storage tiers by using the SMS Management Class construct to identify data sets that can be moved to other tiers based on time since last access or creation. The following tiers are available on DFSMSHsm:

- ▶ Primary volumes (L0)

These SMS or non-SMS direct access storage device volumes store online data, and can be directly accessed by TSO users, Jobs, or applications. These volumes are managed by DFSMSHsm, but are not owned by it.

- ▶ Migration Level 1 (ML1) volumes

These non-SMS direct access storage device volumes contain data that is migrated from L0 volumes based on Management Class attributes. The data in these volumes is owned by DFSMSHsm and cannot be directly read by users or applications. If a read/write operation is required, the data set is recalled to L0 volumes before they can be read. To use a volume as ML1, an ADDVOL command must be included on DFSMSHsm parmlib or dynamically added. If dynamically added, this configuration is reset during DFSMSHsm restart.

- ▶ Migration Level 2 (ML2) volumes

This second level of DFSMSHsm migration often is designated for large data sets or long retention periods. These volumes are a set of non-SMS tapes (physical or virtual) or low-performance direct access storage device volumes that are owned by DFSMSHsm. The data in these volumes cannot be directly read by users or applications unless they are recalled to L0 volumes first.

Class transition

A class transition is a change in the object's management class or storage class. Class transition was introduced in z/OS V2R1, which enables DFSMSHsm to also manage and migrate data set laterally within L0 volumes. By implementing the use of class transition, you can create pools with different performance levels and move your data between these volumes as they age.

Newly created data sets might require high-performance levels. These performance requirements can decrease as the data ages, but the data still must be accessed. In this case, migrating the data to ML1 volumes is not an option because this data is still required by applications. However, leaving this data on high-performance direct access storage device for an extended period reduces the ROI on the high-performance direct access storage device and might deny access for other data with high-performance needs.

Using class transition provides a balanced approach to managing the change in performance needs of data by allowing DFSMSHsm to migrate data sets between different Storage Groups.

DFSMSHsm uses Management Class attributes to define the following migration policies:

- ▶ Time since creation
- ▶ Time since last use
- ▶ Periodic transition

Class transitioning can be started through Primary, Interval, or on-demand migrations. It can also be started by using a user-issued command. An example of this command would be:

```
HSEND MIGRATE DSN(/) TRANSITION
```

After it is started, it references Management Class policies to select the data sets for transition. If it is eligible, DFSMSHsm starts Automatic Class Selection (ACS) routines to assign a new Storage Class, Management Class, or Storage Group. If the Storage Group changes, DFSMSHsm attempts to move the data set to the new pool.

Figure 1-1 shows the sample implementation of class transition. The data is first allocated in high-performance direct access storage device, and then it transitions to cost-efficient devices as the data ages and the data's performance requirement drops. Later, you can migrate the data to even more cost-efficient levels (ML1/ML2).

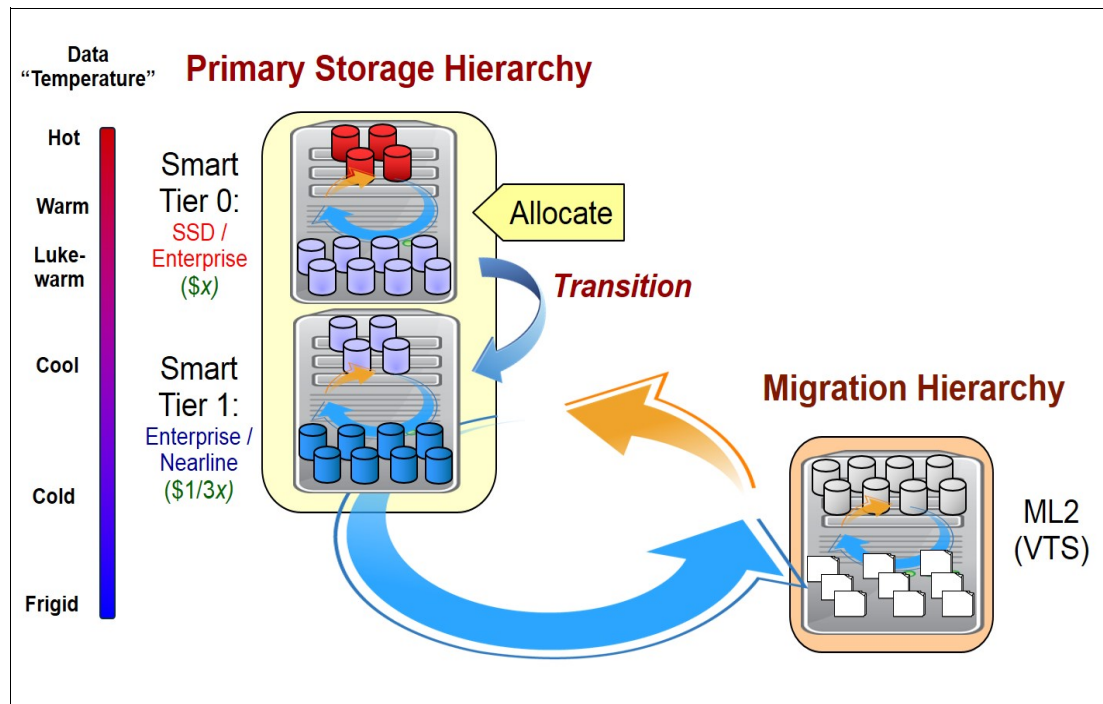


Figure 1-1 Smart tiers within the primary hierarchy

The tiering capability adds depth to a storage management strategy. If your data storage consists of a “flat” structure, such as being held in a single large direct access storage device pool, your options for application quality are limited. A multi-tiered structure (as shown in Figure 1-1) provides opportunities to enrich your business applications through the following qualities of service:

- ▶ Higher availability levels
- ▶ Performance improvements through data positioning
- ▶ High-quality data management through organized constructs and tier migration

1.4 DFSMShsm behavior without TCT

DFSMShsm is the z/OS component responsible for performing the Information Lifecycle Management task. As such, it is responsible for moving data between the different storage tiers, including both Online and Offline media types, like direct access storage device and Tape respectively, based on predefined policies and data access availability needs.

Figure 1-2 shows how the data flows between these distinct technologies, through DFSMShsm, and the challenges related to lifecycle management.

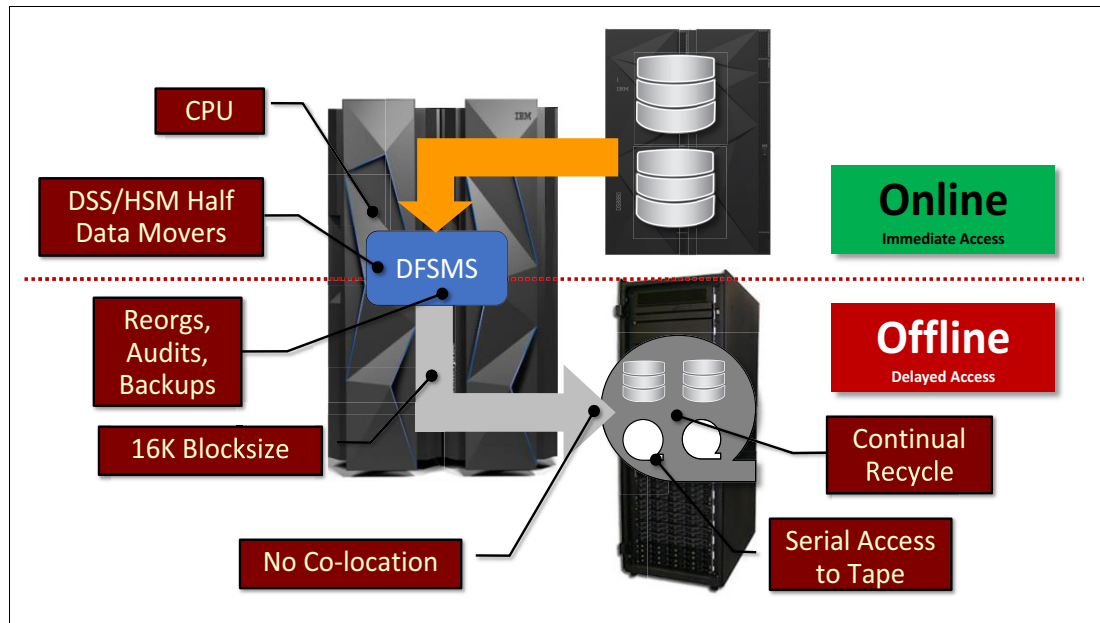


Figure 1-2 DFSMShsm data movement between direct access storage device and tape technologies

To move the data between the tiers, Data Facility Storage Management Subsystem (DFSMS) uses Hierarchical Storage Manager (HSM) and Data Storage Services (DSS) Data Movers to read the data from the source storage and write the data to the target storage, through IBM FICON connectivity. Online media access format is different than Offline media access format.

During the processing of the data movement from disk to tape, DSS reads the data from the source media, passes it to HSM, which converts the data into 16 KB blocks, and writes the data to tape. This movement of data from disk to tape flows through the mainframe, and consumes extra central processing unit (CPU) cycles.

As you can imagine to process all these operations, the system uses IBM zSystems CPU cycles that could be used for other important workloads, such as business applications.

Other challenges of migrating data to tape include the lack of collocation for the data sets, meaning data with different retention will be placed in the same tape, which will also create the need of a recycle process to increase tape usage efficiency. The serial access to tape also prevents the recall of multiple data sets that are stored on the same tape. Otherwise, the system tries to run those data sets concurrently.

1.5 Introducing Transparent Cloud Tiering

TCT for IBM DS8000 was introduced to help customers use storage and IBM zSystems resources more efficiently. With its integration with z/OS through DFSMShsm, TCT allows clients to reduce CPU utilization by eliminating constraints that are tied to the original tape methodologies.

This improvement is accomplished by enabling direct data movement from IBM DS8000 to cloud object storage, without the need for the data to go through the host. DFSMS communicates with DS8000 through a Representational State Transfer (REST) API interface and issues commands for the DS8000 to move the data directly to and from the cloud, as shown in Figure 1-3.

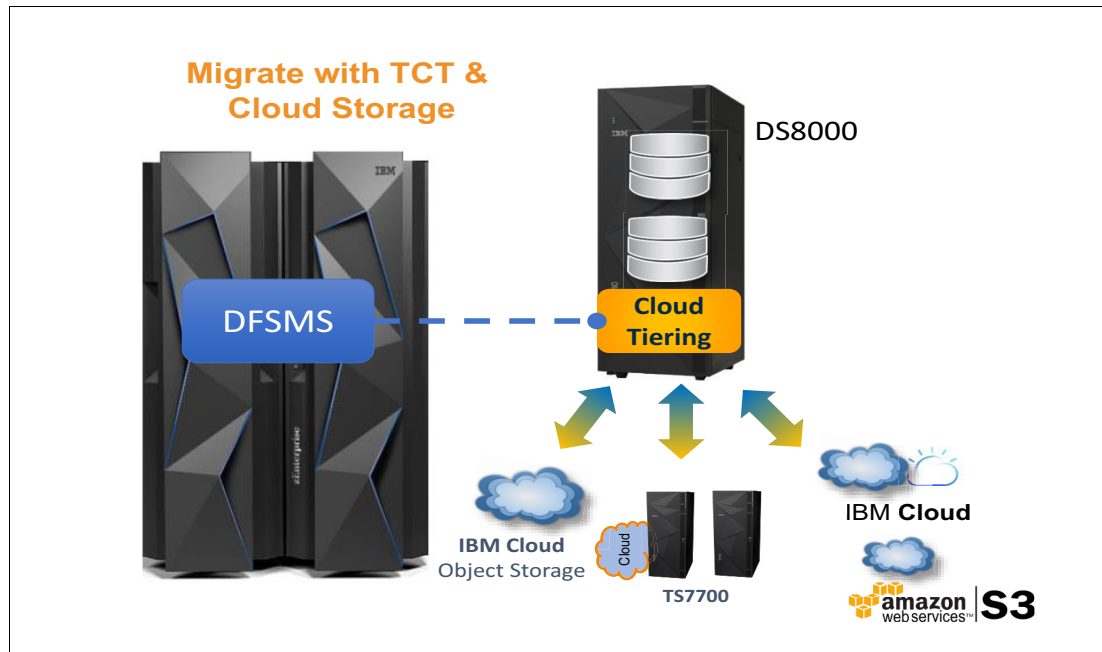


Figure 1-3 TCT data movement layout

In this way, TCT offloads all the actual data movement processing-related workload from z/OS. Significant CPU savings result, as compared with the traditional data movement methods, especially when considering large data sets.

The CPU savings that are expected are achieved by reducing or eliminating CPU processing for the following tasks:

- ▶ Tape recycle
- ▶ Dual (DSS and HSM) data movement
- ▶ Moving data through CPU
- ▶ Reblocking data to 16 K blocks

This approach also gives organizations flexibility to choose the most appropriate Offline media option, depending on cost, performance, and availability requirements.

In addition to HSM, TCT also supports DFSMSdss data set level and full volume dump (FVD) and restore to cloud object storage.

With DS8900F Release 9.1 and later, TCT also supports encryption for data that is stored in storage cloud. TCT supports secure data transfer over Transport Layer Security (TLS) and optionally compression of data that is transferred to the TS7700 used as object storage.

With DS8900F R 9.2, TCT multi-cloud support was introduced. You can now define up to eight cloud storage targets. Each of them can be its own type and can be used for its own purpose, application, or type of data.



Cloud overview

In this chapter, we introduce you to cloud concepts and explain how IBM Transparent Cloud Tiering (TCT) enables cloud integration with IBM DS8000 systems that run IBM z/OS environments. You get a basic understanding of what a cloud is in the context of TCT, through a short description of cloud storage versus cloud computing.

This chapter describes the following topics:

- ▶ 2.1, “What defines a cloud in the context of TCT” on page 12
- ▶ 2.2, “Compute cloud versus a storage cloud” on page 13
- ▶ 2.3, “Types of storage” on page 13
- ▶ 2.4, “Cloud storage delivery models” on page 14
- ▶ 2.5, “Object storage hierarchy” on page 16

2.1 What defines a cloud in the context of TCT

The cloud is a combination of several different solutions, components, and services. It consists of different layers.

Figure 2-1 provides a comprehensive diagram that shows the different layers and where TCT integrates with the cloud:

- ▶ Application Layer: Applications can be hosted and run, and can also take advantage of pre-coded software APIs that you can integrate to create new applications.
- ▶ Infrastructure Layer: Entire systems can be hosted. An Infrastructure Layer can be composed of a mix of interconnected cloud and traditional infrastructures.
- ▶ The Cloud Layer is composed of three main classes of components:
 - Storage Layer
 - Network Layer
 - Compute Layer
- ▶ Within the Storage Layer, we have three Storage types:
 - Block Storage
 - File Storage
 - Object Storage

TCT uses Object Storage architecture for storing data sets. We cover this architecture in more detail in this chapter.

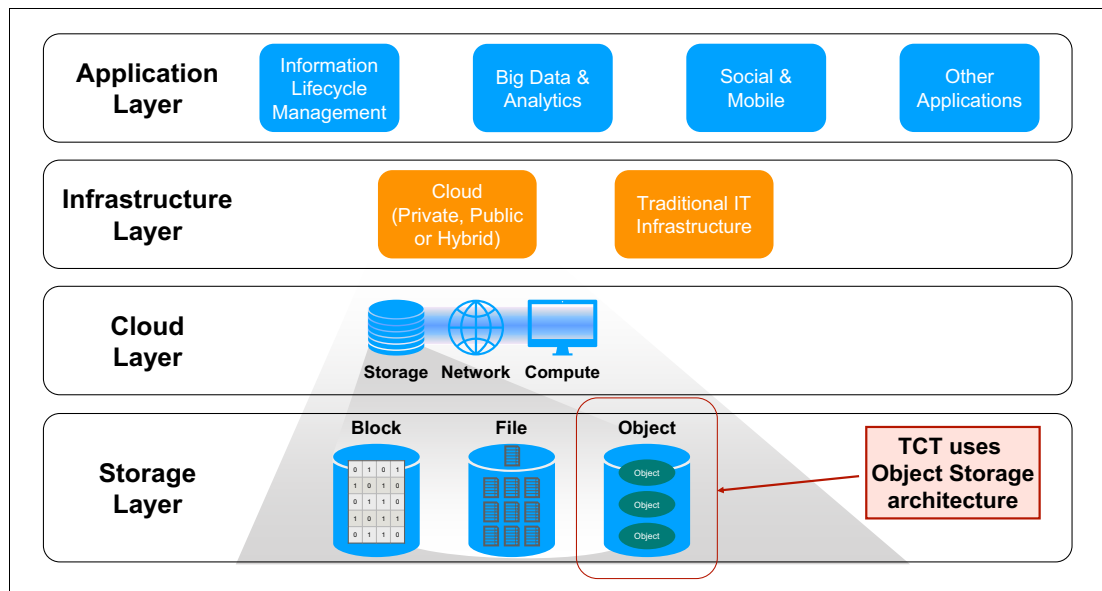


Figure 2-1 TCT in the context of the cloud

2.2 Compute cloud versus a storage cloud

In short, the compute cloud provides the necessary components to run the applications for processing resources, and is where you can deploy and run your chosen software, including operating systems and applications.

The storage cloud holds the data and caters for operations like backing up and archiving data. As mentioned earlier in this chapter, TCT uses Object Storage architecture to store the data it manages in the cloud.

2.3 Types of storage

Before you use storage clouds, you must understand what a gateway is, its function, and how the data is managed between the mainframe and the cloud.

The following types of architectures (see Figure 2-2) can be used for storing data, where each type has its advantages:

- ▶ **Block and File:** This architecture is used on mainframes and other operating systems to store data. It has the advantages of being faster, IOPS-centric, flash-optimized, and allows various approaches.
- ▶ **Object Storage:** This architecture provides larger storage environments, or cool or cold data, which can scale to petabytes of data while being cloud-compatible.

	SAN (Storage area)	NAS (Network-attached storage)	OBS (Object-based storage)
Type	Block-based. Think hard drive.	File-based. Think home shares.	Object based.
Access Protocols	Fibre Channel, iSCSI	CIFS, NFS	HTTP API, no standard
Capacity	GBs to TBs per LUN, 100's TB per system	GBs to TBs Scale out to PBs	TBs to 100's of PBs
Used By	Single server or small cluster	Groups of users or large clusters of servers	Application backends, repositories
Use Cases	Databases, email, virtualization	Users, web farms, virtualization, backup, render

Figure 2-2 Types of storage

Having a storage cloud that uses object storage has several benefits. A storage cloud significantly reduces the complexity of storage systems by simplifying data scaling within a single namespace. Also, the Representational State Transfer (REST) protocol is used for communication between the server and the client. The use of high-density, low-cost commodity hardware turns storage clouds into a scalable, cost-efficient storage option.

The TCT function of IBM DS8000 provides a method to convert the block to Object Storage without more hardware on the LAN.

On storage clouds, the data is managed as objects, unlike other architectures that manage data as a block of storage, as is done on mainframes.

For this reason, the communication between mainframe systems and storage cloud is done by an application responsible for converting cloud storage APIs, such as SOAP or REST, to block-based protocols, such as iSCSI or Fibre Channel, when necessary.

Therefore, the storage cloud can be considered an auxiliary storage option for mainframe systems to be used by applications, such as these:

- ▶ DFSMSHsm to migrate and recall data sets
- ▶ DFSMSdss to store data that is generated by using the DUMP command

2.4 Cloud storage delivery models

Cloud delivery models refer to how a cloud solution is used by an organization, where the data is located, and who operates the cloud solution. There are multiple delivery models that can deliver the capabilities that are needed in a cloud solution.

The cloud delivery models are as follows:

- ▶ Public cloud
- ▶ Private cloud
- ▶ Hybrid cloud

These delivery models can be integrated with traditional IT systems and other clouds. They are divided into two categories:

- ▶ *On-premises*: Consists of a private cloud infrastructure at your organization's location.
- ▶ *Off-premises*: Consists of a cloud infrastructure that is hosted in a cloud service provider's location.

Public cloud

A public cloud is a solution in which the cloud infrastructure is available to the general public or a large industry group over the internet. The infrastructure is not owned by the user, but by an organization that provides cloud services. Services can be provided at no cost, as a subscription, or as a pay-as-you-go model.

There is another delivery model option that is known as community cloud, or multi-tenant cloud, which typically consists of a public cloud that is shared among multiple organizations, to lower costs. For ease of understanding, this book treats this delivery model as part of the public cloud category.

Private cloud

A private cloud is a solution in which the infrastructure is provisioned for the exclusive use of a single organization. The organization often acts as a cloud service provider to internal business units that obtain all of the benefits of a cloud without having to provision their own infrastructure. By consolidating and centralizing services into a cloud, the organization benefits from centralized service management and economies of scale.

A private cloud provides an organization with some advantages over a public cloud. The organization gains greater control over the resources that make up the cloud. In addition, private clouds are ideal when the type of work that is being done is not practical for a public cloud because of network latency, security, or regulatory concerns.

A private cloud can be owned, managed, and operated by the organization, a third party, or a combination of the two. The private cloud infrastructure is provisioned on the organization's premises, but it can also be hosted in a data center that is owned by a third party.

Hybrid cloud

As the name implies, a hybrid cloud is a combination of various cloud types (public, private, and community). Each cloud in the hybrid mix remains a unique entity, but is bound to the mix by technology that enables data and application portability.

The hybrid approach allows a business to use the scalability and cost-effectiveness of a public cloud without making available applications and data beyond the corporate intranet. A well-constructed hybrid cloud can service secure, mission-critical processes, such as receiving customer payments (a private cloud service) and secondary processes, such as employee payroll processing (a public cloud service).

IBM Cloud Object Storage and TCT

IBM Cloud® Object Storage offers all the delivery model options that were described previously. Figure 2-3 provides a summary of each option, with its capabilities.

Object Storage Capability	IBM Cloud Object Storage
Multi-tenant off-premises object storage services <small>Low cost shared public cloud storage options. Table stakes for cloud providers</small>	✓
Single-tenant off-premises object storage services <small>For workloads requiring dedicated, predictable performance and stringent security</small>	✓
On-premises object storage systems <small>Private deployment or appliance at customer location. Best flexibility, security, control</small>	✓
Hybrid object storage deployments <small>Flexibility and elasticity combining on-premises systems with off-premises services</small>	✓
Support for multiple APIs and open standards <small>REST API support for Amazon S3, OpenStack Swift, and IBM Cloud Object Storage Simple Object API</small>	✓

Figure 2-3 IBM Cloud Object Storage capabilities

The TCT solution provides an integration of the IBM DS8000 storage system when running in z/OS environments with a IBM Cloud Object Storage infrastructure, which can be any of the options that are described above.

For detailed information about the IBM Cloud Object Storage service offering, see *Cloud Object Storage as a Service: IBM Cloud Object Storage from Theory to Practice - For developers, IT architects and IT specialists*, SG24-8385 or go to the IBM Cloud Object Storage website at this link:

<http://www.ibm.com/cloud/object-storage>

For other IBM Cloud Storage solutions, see *IBM Private, Public, and Hybrid Cloud Storage Solutions*, REDP-4873 or go to the IBM Cloud website at the link:

<http://www.ibm.com/cloud/solutions>

2.5 Object storage hierarchy

Data that is written to a cloud by using TCT is stored as objects and organized into a hierarchy. The hierarchy consists of accounts, containers, and objects. An account can feature one or more containers and a container can include zero or more objects.

2.5.1 Storage cloud hierarchy

The storage cloud hierarchy consists of the following entities:

- ▶ Account
- ▶ Containers
- ▶ Objects

Each entity plays a specific role on data store, list, and retrieval by providing a namespace, the space for storage, or the objects. There also are different types of objects, data, and metadata.

A sample cloud hierarchy structure is shown in Figure 2-4. Each storage cloud component is described next.

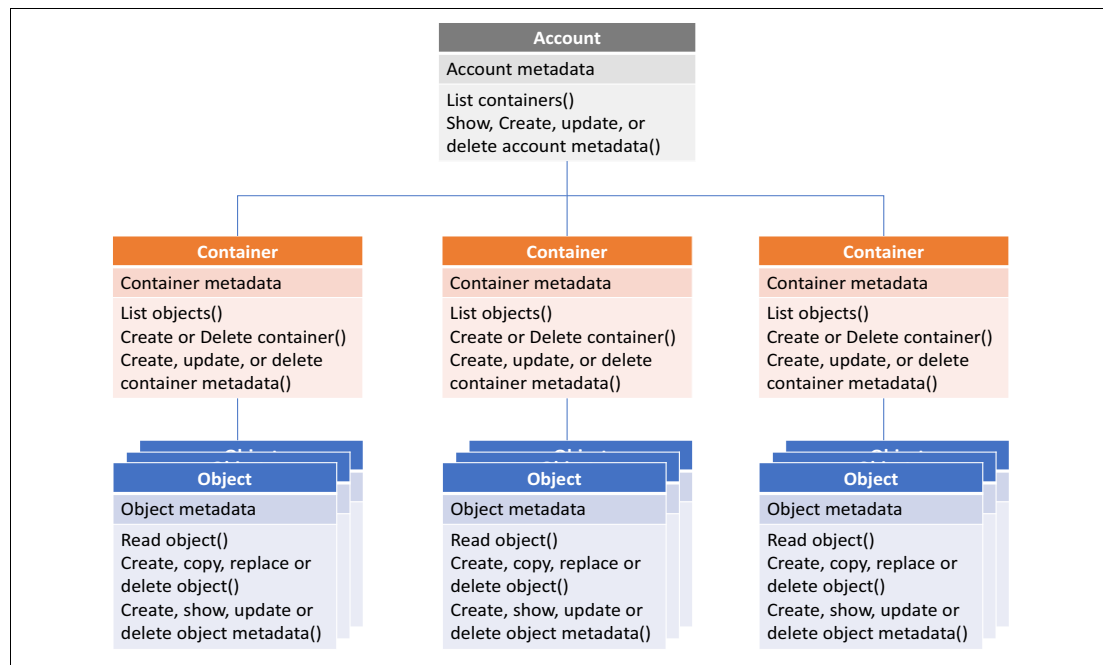


Figure 2-4 Cloud hierarchy

Account

An account is the top level of the hierarchy and is created by the service provider, but owned by the consumer. Accounts can also be referred to as projects or tenants and provide a namespace for the containers. An account has an owner that is associated with it, and the owner of the account has full access to all the containers and objects within the account.

The following operations can be done from an account:

- ▶ List containers
- ▶ Create, update, or delete account metadata
- ▶ Show account metadata

Containers

Containers (also called buckets or vaults) are similar to folders in Windows or UNIX, and provide a way to organize objects. Depending on your cloud solution, you can, for example, set up container-to-container synchronization, quotas, object versioning, and other policies on a container level. One main limitation is that containers cannot be nested. Therefore, you cannot create a container within another container. Container names can be up to 256 characters.

Note: Containers in an object storage context have nothing to do with application containers in a containerized virtualization environment, such as Docker, Kubernetes, or Red Hat OpenShift.

Access to objects within a container are protected by using read and write Access Control Lists (ACLs). There is no security mechanism to protect an individual object within a container. After a user is granted access to a container, that user can access all of the objects within that container.

The following operations are supported for containers:

- ▶ List objects
- ▶ Create container
- ▶ Delete container
- ▶ Create, update, or delete container metadata
- ▶ Show container metadata

Objects

Objects contain the actual user data that is stored in object storage. An object can contain any type of data and be of any size. Object storage by definition doesn't distinguish between types of data. Object storage creates and uses metadata to reference objects in its uniform data store. Metadata can also be used to specify certain object attributes.

The following operations are supported for objects:

- ▶ Read object
- ▶ Create or replace object
- ▶ Copy object
- ▶ Delete object
- ▶ Show object metadata
- ▶ Create, update, or delete object metadata

The objects can have a defined, individual expiration date. The expiration dates can be set when an object is stored and modified by updating the object metadata. When the expiration date is reached, the object and its metadata are automatically deleted.

However, the expiration does not update information about the z/OS host; therefore, DFSMSshm and DFSMSdss do not use this feature. Instead, DFSMSshm handles the expiration of objects.

User-created backups (backups that are created outside of DFSMSshm to the cloud) must be managed by the user. Therefore, a user must go out to a cloud and manually delete backups that are no longer valid. At the time of writing, this process is not recommended.

Note: Older OpenStack Swift implementations can have an object size limit of 5 GB. TCT splits up data sets that are larger. The limitation is transparent to the TCT user. For other object storage types, the restriction does not apply.

2.5.2 Metadata

In addition to the objects, metadata is recorded for account, container, and object information. Metadata consists of data that contains information about the stored data. Some metadata information might include data creation and expiration date, size, owner, last access, and other pertinent information.

The difference between data and metadata is shown in Figure 2-5.

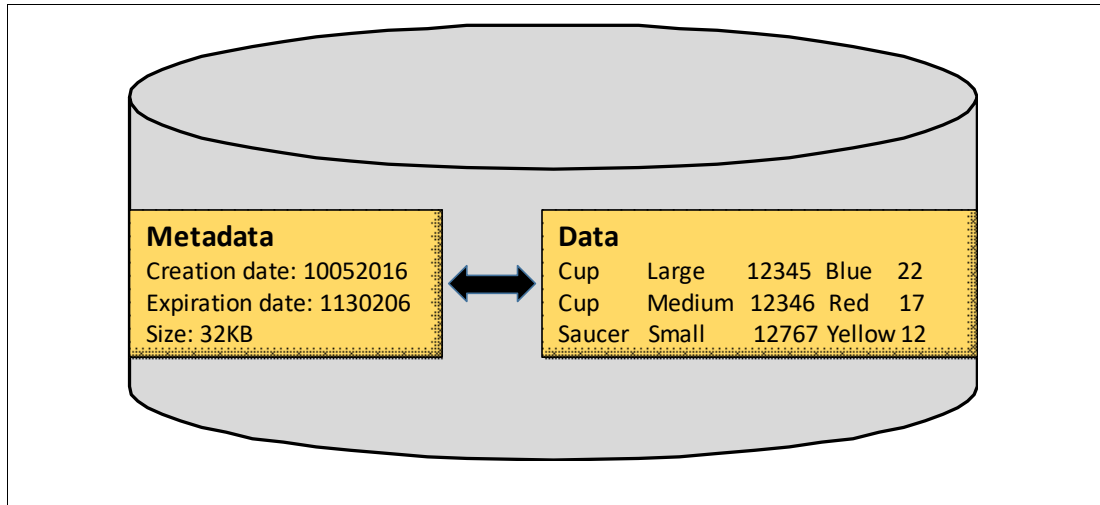


Figure 2-5 Data and metadata differences



Transparent Cloud Tiering

In this chapter, we describe how Transparent Cloud Tiering (TCT) extends the tiering process a step further, by adding cloud object storage as another tier.

This chapter includes the following topics:

- ▶ 3.1, “Transparent Cloud Tiering overview” on page 20
- ▶ 3.2, “Transparent Cloud Tiering data flow” on page 21
- ▶ 3.3, “Storing and retrieving data by using DFSMS” on page 24
- ▶ 3.4, “Transparent Cloud Tiering and disaster recovery” on page 26
- ▶ 3.5, “Transparent Cloud Tiering and DS8000 Copy Services” on page 27
- ▶ 3.6, “DS8900F multi-cloud support” on page 30
- ▶ 3.8, “Transparent Cloud Tiering secure data transfer with TS7700” on page 32
- ▶ 3.9, “Transparent Cloud Tiering compression with TS7700” on page 32
- ▶ 3.10, “Selecting data to store in a cloud” on page 33

3.1 Transparent Cloud Tiering overview

DS8000 TCT provides the framework that enables z/OS applications to move data to cloud object storage with minimum host I/O workload. The host application initiates the data movement, but the actual data transfer is performed by the DS8000. In this section, we provide a high-level overview of how TCT works.

Note: At the time of writing, there are three use cases for TCT:

- ▶ The *Hierarchical Storage Manager* (HSM) component of z/OS Data Facility Storage Management Subsystem (DFSMS) (DFSMSHsm or HSM) can migrate and recall data sets to object storage.
- ▶ You can use DFSMSHsm full volume dump (FVD) to dump to and restore from cloud object storage.
- ▶ You can use a DFSMSdss data set or FVD to back up and restore data to and from cloud object storage.

HSM can migrate data to cloud object storage instead of its traditional Maintenance Levels 1 or 2. You can call HSM manually, or define rules for automatic migration. You can also use HSM to dump full volumes to cloud object storage instead of tape devices. When HSM migrates or dumps data to cloud storage, it creates and runs a dump job for the DFSMS data mover service (DFSMSdss). For DFSMSdss FVD and restore, you can run the DFSMSdss commands manually or in a batch job. In both cases, DFSMSdss then initiates the move of the data:

1. It creates metadata objects that describe the data set and are needed to rebuild it in a recall.
2. It sends these metadata objects to the cloud storage.
3. It sends instructions to the DS8000 to compose and send the extent objects that contain the actual customer data to cloud storage. The DS8000 creates separate extent objects for each volume that the data is stored on.

Note: Originally, the DS8000 created one extent object for each volume:

- ▶ One extent object for each volume that a data set is allocated on for HSM migrations.
- ▶ One extent object for each volume that is backed up with DFSMSdss FVD.

With recent code releases and TS7700 as object storage target, the extent data is split into objects of 2 GiB to improve parallelism and error recovery.

HSM maintains a record in its Control Data Set (CDS), describing if a data set is migrated to cloud storage. When recalling a data set, it uses this record to compose a restore job definition for DFSMSdss, which initiates the data movement back from cloud storage to active DS8000 volumes:

1. It reads the metadata objects that describe the data set.
2. It prepares the restoration of the data by selecting a volume and allocating space.
3. It sends instructions to the DS8000 to retrieve and store the extent objects that contain the actual customer data.

For more information about the DFSMSdss operations and metadata objects, see 3.3, “Storing and retrieving data by using DFSMS” on page 24.

DS8000 TCT supports several cloud storage target types:

- ▶ Swift: the Open Stack cloud object storage implementation, public or on-premises.
- ▶ IBM Cloud Object Storage in the public IBM cloud or as on-premises cloud object storage solution.
- ▶ Amazon Web Services Simple Storage Service (AWS S3) in the Amazon public cloud.
- ▶ IBM Virtual Tape Server TS7700 as cloud object storage.

To create metadata objects, DFSMS must communicate with the cloud object storage solution. Depending on the target type, this process happens in one of two ways:

- ▶ For the Swift target type, DFSMS communicates directly to the cloud storage.
- ▶ For all other target types, the DS8000 Hardware Management Console (HMC) acts as a cloud proxy. DFSMS sends commands and objects to the HMC. The HMC passes them on to the cloud storage by using the appropriate protocol for the target type.

3.2 Transparent Cloud Tiering data flow

Traditional data movement during archive or backup operations is performed over the FICON infrastructure only. The host application reads data from the direct access storage device controller and writes it to a tape controller, and vice versa. With TCT, there are several data flows and connections between the host (z/OS DFSMS), the storage controller (DS8000) and the cloud target.

3.2.1 DS8000 cloud connection

The DS8000 connects to the cloud object storage targets through TCP/IP by using Ethernet ports in each of the two internal servers. You can either use available ports that are available onboard in each server, or purchase and connect a separate pair of more powerful Ethernet controllers. Connect the ports that you intend to use for TCT to the networks that extend to the cloud targets. Both DS8000 internal servers must be able to access all defined cloud targets.

The DS8000 uses these connections to send and retrieve the extent objects that contain the actual customer data. In cloud proxy mode (see “Other cloud target types” on page 23), it also transfers cloud requests and objects on behalf of DFSMS and HSM.

When storing or retrieving data from the cloud, the system uses the Ethernet ports of the internal server that owns the logical subsystem (LSS) that is associated with the request. This approach can lead to unbalanced migrations and recalls if most of the data is on a specific LSS.

Defining storage groups with volumes from LSSs that are equally distributed over both internal servers can reduce the risk of having an unbalanced link usage.

The instructions that cause the DS8000 to initiate the transfer of an object to or from cloud storage come from DFSMS and are transmitted over the FICON connection between z/OS and the DS8000.

3.2.2 DFSMS cloud connection

DFSMS must be able to communicate with the cloud object storage to store and retrieve the metadata objects that it needs to identify, describe, and reconstruct the migrated data sets. For this situation, you define a Network Connection for each cloud object storage target in the DFSMS management application (Interactive Storage Management Facility (ISMF)). HSM also uses this connection for maintenance purposes, such as removing objects that are not used anymore, or for reporting and auditing.

The communication between the mainframe and the cloud is a Representational State Transfer (REST) interface. REST is a lightweight, scalable protocol that uses the HTTP standard. The z/OS Web Enablement Toolkit (WETK) provides the necessary support for secure HTTPS communication between endpoints by using the Secure Sockets Layer (SSL) protocol or the Transport Layer Security (TLS) protocol.

To access the cloud object storage, cloud credentials like a user ID and password are required by DFSMSdss. The user IDs (or equivalent credentials) are included in the DFSMS Network Connection constructs. The matching password must be provided to DFSMSdss for each TCT operation.

There is a common method to store and manage cloud storage credentials securely for DFSMSdss and HSM. It uses the DFSMS *Cloud Data Access* (CDA) framework, which again relies on the *IBM Integrated Cryptographic Service Facility* (ICSF). For HSM, there is a second method to manage credentials itself. We call this method the *legacy* method. We describe both ways in Chapter 8, “Managing cloud credentials” on page 79.

Note: Any users with access to the user ID and password to the cloud have full access to the data from z/OS or other systems perspective. Ensure that only authorized and required personnel can access this information.

Swift cloud type

For Swift cloud object storage targets, DFSMS sends and receives metadata directly to and from the cloud storage by using the z/OS WETK. It uses FICON in-band communication to instruct the DS8000 to move the actual extent objects with the customer data to and from cloud storage. The DS8000 sends and receives these objects through a TCP/IP connection to the cloud storage solution. The communication flow for Swift type cloud object storage is illustrated in Figure 3-1 on page 23.

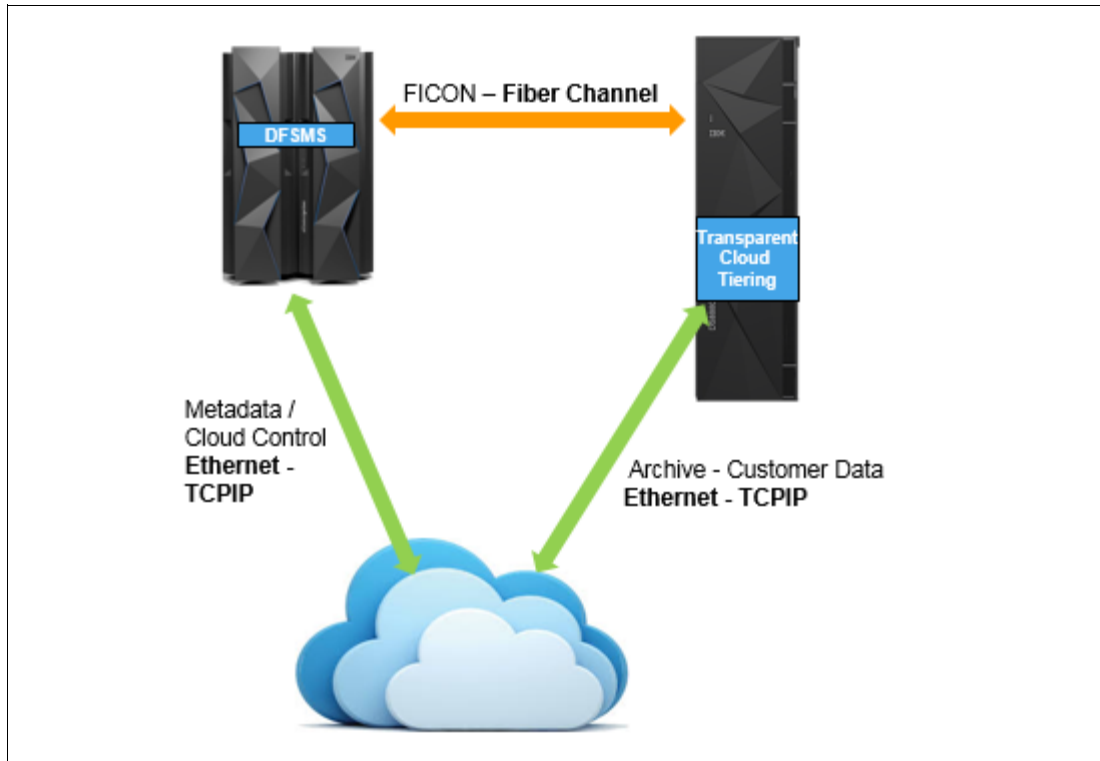


Figure 3-1 Cloud communication with the Swift API

For the DFSMS cloud connection definition, you need information about your cloud storage environment, including the endpoint URL and credentials, such as username and password.

Other cloud target types

The integration between z/OS DFSMS and the other cloud target types (Amazon S3, IBM Cloud Object Storage, and TS7700) is different from the one used for Swift. DFSMS does not communicate with the cloud storage directly.

DFSMS uses the DS8000 as cloud proxy instead, as shown in Figure 3-2. The DFSMS cloud connection definition points to the DS8000 HMC, and uses DS8000 credentials (username and password). It sends cloud commands and metadata objects to the HMC. The HMC passes them on to the DS8000 itself, which is connected to the cloud storage.

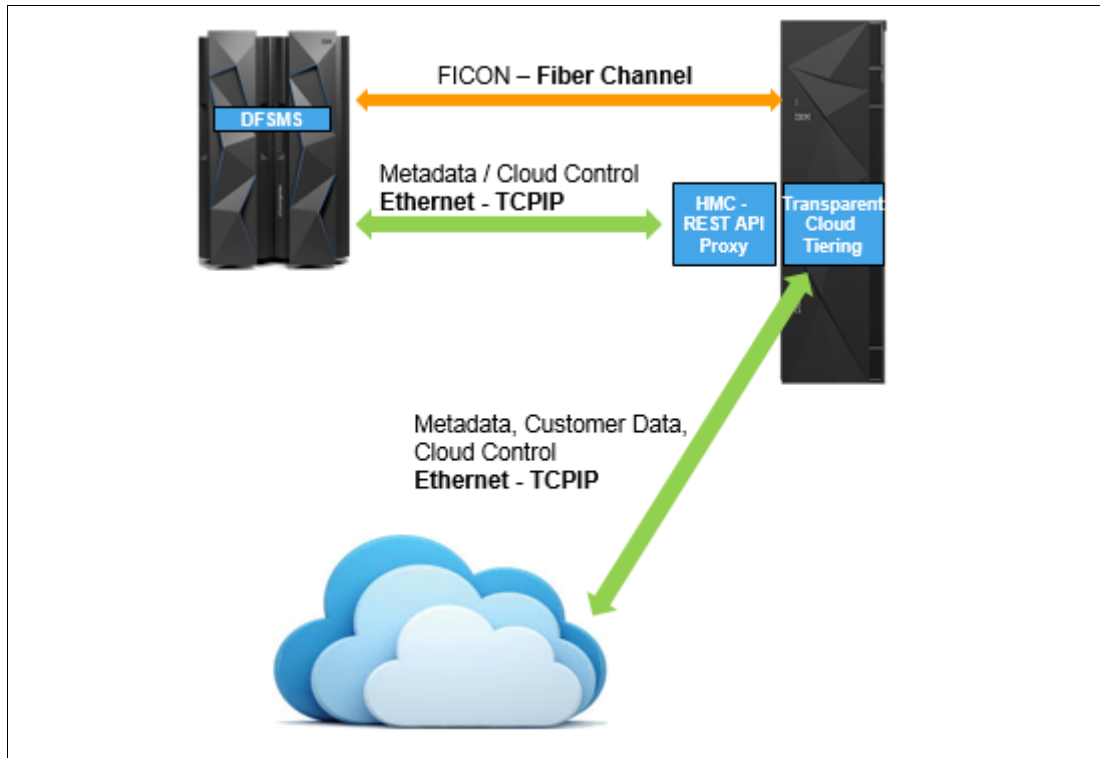


Figure 3-2 Cloud communication with Amazon S3 and IBM Cloud Object Storage APIs

3.3 Storing and retrieving data by using DFSMS

From a z/OS perspective, cloud object storage is an auxiliary storage option, but unlike tapes, it does not provide a block-level I/O interface. Instead, it provides only a simple HTTP get-and-put interface that works at the object level.

DFSMSdss is the data mover that stores and retrieves data to and from object storage. The number of objects for each data set or volume can vary, based on the following factors:

- ▶ The number of volumes that a data set is on. For each volume that the data set is stored on, an extent object and an extent metadata object are created.
- ▶ When Virtual Storage Access Method (VSAM) data sets are migrated to cloud, each component has its own object, meaning a key-sequenced data set (KSDS) has at least one object for the data component and another for the index. The same concept is applied to alternative indexes.

Also, several metadata objects are created to store information about the data set, the dumped volume, and the application invoking TCT.

To maintain some structure in the potentially large number of objects, HSM uses containers. By default, a new container is created every 92 days. The container name reflects the creation date and the HSMplex name. Within the container, HSM addresses the data sets with automatically created unique object prefixes. For more information, see 9.2, “Cloud container management” on page 92.

Note: If you use DFSMSdss directly to dump and restore data, you must maintain container and object prefix naming yourself.

Table 3-1 lists some objects that are created as part of the DFSMSdss data set dump process.

Table 3-1 Created objects

Object name	Description
objectprefix/HDR	Metadata object that contains ADRTAPB prefix.
objectprefix/DTPDSNLnnnnnnn	n = list sequence in hexadecimal. Metadata object that contains a list of data set names that are successfully dumped. Note: This object differs from dump processing that uses OUTDD where the list consists of possibly dumped data sets. For cloud processing, this list includes data sets that were successfully dumped.
objectprefix/dsname/DTPDSHDR	Metadata object that contains data set dumped. If necessary, this object also contains DTCDFATT and DTDSAIR.
objectprefix/dsname/DTPVOLDnn/desc/META	Metadata object that contains attributes of the data set dumped: <ul style="list-style-type: none"> ▶ desc = descriptor ▶ NVSM = NONVSAM ▶ DATA = VSAM Data Component ▶ INDX = VSAM Index Component ▶ nn = volume sequence in decimal, 'nn' is determined from DTDNVOL field inDTDSHDR
objectprefix/dsname/DTPSPHDR	Metadata object that contains Sphere information. If necessary, this object also contains DTSAXS, DTSINFO, and DTSPATHD. Present if DTDSPER area in DTDSHDR is ON.
objectprefix/dsname/DTPVOLDnn/desc/EXTENTS	Data object. This object contains the data that is found within the extents for the source data set on a per volume basis: <ul style="list-style-type: none"> ▶ desc = descriptor ▶ NVSM = NONVSAM ▶ DATA = VSAM Data Component ▶ INDX = VSAM Index Component
objectprefix/dsname/APPMETA	Application metadata object that is provided by application in EIOPTION31 and provided to application in EIOPTION32.

After DFSMSdss creates the metadata objects, DS8000 TCT creates and stores the extent (data) objects. In a data set dump, a data object consists of the extents of the data set that is on the source volume. This process is repeated for every source volume where parts of the data set are stored. For a DFSMS FVD, all allocated extents of a volume are stored in one extent object.

After all volumes for a data set are processed (where Data Storage Services (DSS) successfully stored all the necessary metadata and data objects), DSS creates an extra application metadata object. DFSMSdss supports one application metadata object for each data set that is backed up.

There are some considerations that you must account for when planning to use TCT:

- ▶ Because data movement is offloaded to the DS8000, a data set cannot be manipulated as it is dumped or restored. For example, DFSMSdss cannot do validation processing for indexed VSAM data sets, compress a partitioned data set (PDS) on RESTORE, or reblock data sets, as would be possible during traditional dump and restore operations.
- ▶ TCT cannot move data that is already migrated or dumped to (virtual) tape or an HSM maintenance level (Migration Level 1 (ML1) or Migration Level 2 (ML2)). If you want to relocate such data to cloud object storage, you must read it by using traditional methods, and then backup or migrate with TCT.
- ▶ Data that was migrated or dumped to cloud with TCT can be restored only to volumes on a TCT capable DS8000. If you have a multi-vendor direct access storage device environment, or run a mix of DS8000 generations, the volume allocation for a restore operation must use only TCT-capable volumes.

Note: If you have more than one DS8000 attached to your system, make sure that all of them have access to migrate and recall data from the cloud.

3.4 Transparent Cloud Tiering and disaster recovery

Having a working disaster recovery (DR) solution is vital to maintaining the highest levels of system availability. These solutions can range from the simplest volume dump to tape and tape movement management, to high availability (HA) multi-target Peer to Peer Remote Copy (PPRC) and IBM HyperSwap® solutions. The use of TCT, and storing data in cloud storage can affect your DR strategy. You must review and adapt your DR plans, documentation, procedures, and tools.

Some of the steps required to recover your migrated data after a disaster include, but are not limited to:

- ▶ Network connectivity
Make sure that your DR has network access to the cloud environment, which might include configuring proxy, firewall, and other network settings to secure your connection.
- ▶ Cloud configuration
Your DR DS8000 must be configured with the information necessary to access the cloud storage, including certificates to allow SSL connections. You also might need to set up your z/OS to connect to the cloud environment, depending on your configuration.

- ▶ User ID administration

You might also need to create the user ID and password on your DR DS8000 if you use Amazon S3 or IBM Cloud Object Storage clouds. Update your z/OS to connect to the new DS8000, and define the user ID in your storage.

- ▶ Bandwidth

Keep in mind that during a disaster, a large amount of data set recalls might be requested, such as migrated image copies, and other data sets used only for DR purposes. If these data sets are stored in the cloud, make sure to have enough bandwidth available in your recovery site to avoid recovery delays that are related to network issues.

3.5 Transparent Cloud Tiering and DS8000 Copy Services

Many DR solutions are based on DS8000 Copy Services for data replication. TCT supports all DS8000 data replication technologies (Copy Services), except z/OS Global Mirror, also known as Extended Remote Copy (XRC). In the following sections, we describe the way TCT and the various Copy Services Solution interact.

3.5.1 IBM FlashCopy

As shown in Figure 3-3, you can use TCT to migrate data from IBM FlashCopy® source and target volumes. It does not matter whether a FlashCopy is issued with or without background copy or whether the background copy is still ongoing.

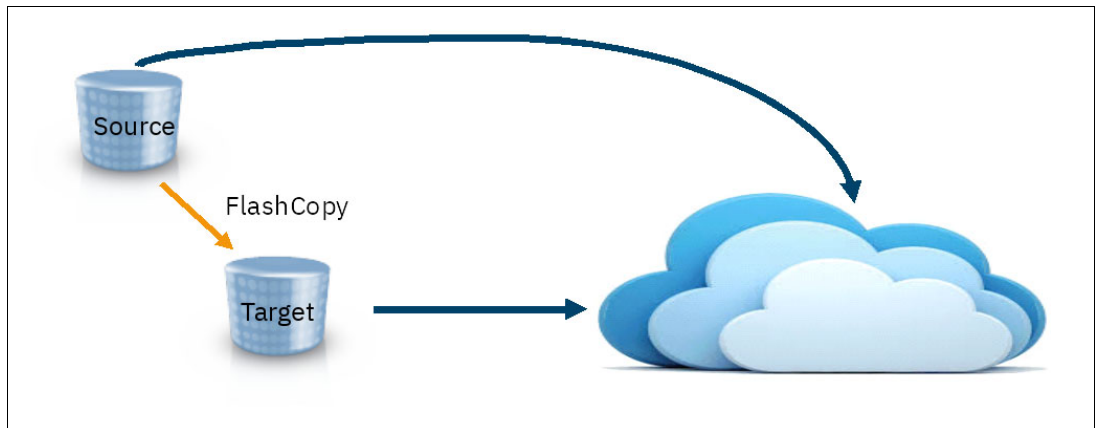


Figure 3-3 TCT migration with FlashCopy

A potential use case for TCT migration from a FlashCopy target volume is the migration of an IBM Db2® image copy that was created with FlashCopy.

Recalling data with TCT currently works only from FlashCopy source volumes, as shown in Figure 3-4. A TCT recall is treated like regular host write I/O, and any data that is overwritten by the recall operation is backed up to the FlashCopy target to preserve the point in time data of the FlashCopy.



Figure 3-4 TCT recall with FlashCopy

3.5.2 Metro Mirror

With TCT, you can migrate and recall data from volumes that are in Metro Mirror primaries, as shown in Figure 3-5.

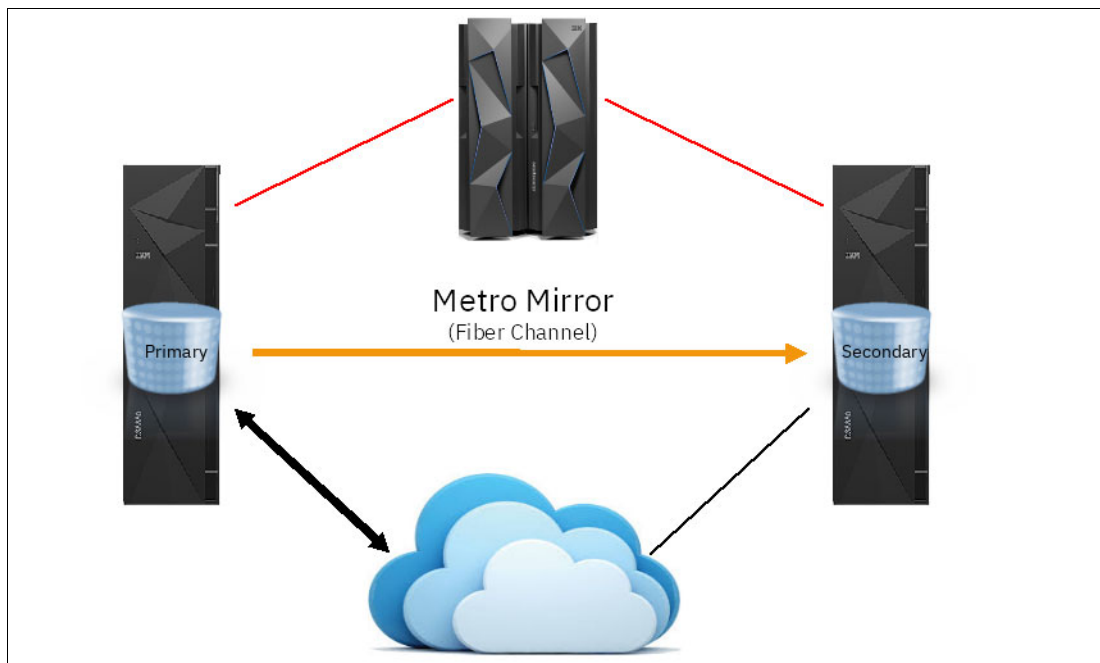


Figure 3-5 TCT with Metro Mirror

As with FlashCopy, TCT recalls are treated the same way as regular host I/O operations, and Metro Mirror replicates recalled data to the secondary volumes. Make sure that your secondary DS8000 is connected to the same cloud object storage targets as the primary. This way, you can continue to migrate and recall after a recovery to the secondary.

Note: If you use TCT with S3, IBM COS, or TS7700 as cloud target type, you might need to change the DFSMS cloud definition after a recovery to the secondary DS8000. You can define only one DS8000 HMC as the cloud proxy, and the one you use might be unavailable after a failure in the primary site.

3.5.3 Metro Mirror with HyperSwap

HyperSwap is a z/OS high availability function. It switches I/O operations seamlessly from the Metro Mirror primary volumes to the secondaries. HyperSwaps can be triggered manually or automatically, for example in case of a primary volume I/O error. HyperSwap needs one of the following management interfaces:

- ▶ IBM Globally Dispersed Parallel Sysplex (GDPS)
- ▶ IBM Copy Services Manager (CSM)

Note: If you want to learn more about HyperSwap for z/OS, you can refer to the following IBM Redbooks publications:

- ▶ *IBM GDPS: An Introduction to Concepts and Capabilities*, SG246374
- ▶ *Best Practices for DS8000 and z/OS HyperSwap with Copy Services Manager*, SG248431

DS8000 TCT operations can get into a conflict with HyperSwap situations. A HyperSwap can occur, while a TCT operation is ongoing, or vice versa. Since HyperSwap changes the volumes that are accessed for I/O, it will impact running TCT operations.

With one exception, the TCT operation will be interrupted by the HyperSwap and fail when a HyperSwap and TCT operation coincide:

- ▶ In case TCT was writing to cloud object storage, the complete operation fails and must be re-issued by the user.
- ▶ In case TCT was reading from cloud object storage, the operation is automatically re-driven by DFSMS.
- ▶ When a planned HyperSwap is initiated in a GDPS managed HyperSwap configuration, GDPS will wait for a maximum of 10 seconds to allow the completion of an ongoing TCT operation. If a TCT operation persists beyond these 10 seconds, the planned HyperSwap will fail.

3.5.4 Global Mirror

As with Metro Mirror, TCT also supports migrate and recall operations to and from Global Mirror primary volumes, as illustrated in Figure 3-6.

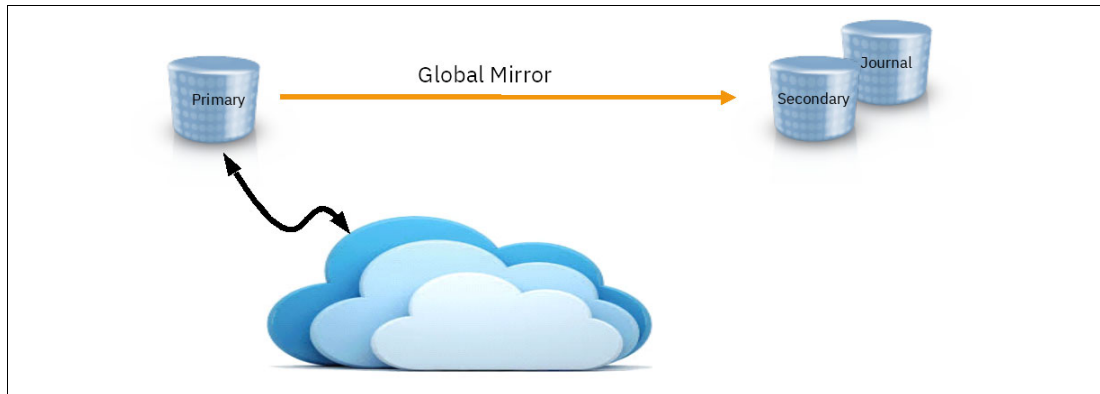


Figure 3-6 TCT and Global Mirror

TCT recalls are treated the same way as regular host I/O operations, and Global Mirror replicates recalled data to the remote site. If you must continue TCT migrate and recall operations after a recovery to the remote site, your remote DS8000 and host systems must be connected to the same cloud storage as the primary site.

Note: z/OS Global Mirror, also known as XRC, is not supported by TCT.

3.5.5 Multi-site data replication

TCT allows migrate and recall operations in all supported combinations of DS8000 Copy Services:

- ▶ Cascaded Metro Global Mirror
- ▶ Multi-Target Metro Mirror - Metro Mirror
- ▶ Multi-Target Metro Mirror - Global Mirror
- ▶ 4-site Metro Mirror - Global Mirror combinations with remote Global Copy

All implications regarding TCT operations after recovery that are described for the 2-site configurations are valid for multi-site, too.

3.6 DS8900F multi-cloud support

Before Release 9.2, the DS8000 supported only a single cloud connection. With the introduction of multi-cloud support, you can now define up to eight cloud connections to a DS8000, which provides more flexibility and opens the TCT solution for a range of new use cases, such as the following ones:

- ▶ Use the TS7700 to define multiple object storage locations and replication options by using TS7700 *Object Policies*.
- ▶ Multiple TS7700 GRIDs: Connect to multiple GRIDs, for example, to separate production from test GRID environments.
- ▶ Public versus private cloud for different types of data: For example, keeping confidential data onsite while other data can move to a public cloud.

- ▶ Performance differentiation: Keep more active backups or frequently recalled data onsite on the TS7700, and move unreferenced, colder data to public clouds.
- ▶ Application separation: Separate HSM and DSS workloads, or maintain individual credentials for applications.
- ▶ Managed service providers: Provide clients with backup and archive solutions that are tailored to each client that is served by a single DS8000.
- ▶ Test environments: Take advantage of having multiple clouds for testing various types of clouds without reconfiguration.

Note: The DS8000 cloud configuration process is different for single and multi-cloud definitions. For more information, see 6.1, “Configuring the IBM DS8000 for TCT” on page 50.

3.7 Transparent Cloud Tiering encryption

TCT object storage can be located outside of your own data center, maybe even in a public cloud outside of your country or on another continent. To protect the migrated data from unauthorized access, even in potentially insecure storage locations, TCT provides encryption capability.

When TCT encryption is enabled, data is encrypted by the DS8000 internal servers when it is about to be transferred over the network. TCT uses IBM Power hardware accelerated 256-bit AES encryption at full line speed, without impact on I/O performance. The data remains encrypted in the cloud storage. When recalled, the data is decrypted when it is received back in the DS8000.

If the data set is already encrypted by data set level encryption, DFSMS informs the DS8000, and TCT encryption will not encrypt again.

Restriction: TCT encryption as described here is not supported by TS7700. However, encryption for data in flight data is possible by using the TS7700 secure data transfer feature (see 3.8, “Transparent Cloud Tiering secure data transfer with TS7700” on page 32).

TCT encryption relies on external key servers for key management: It uses the industry standard Key Management Interoperability Protocol (KMIP). TCT encryption does not require a specific license and can be used with or independently from data at rest encryption.

In HA and DR scenarios that use Metro Mirror or Global Mirror, all DS8000 systems must be connected to the same cloud object storage, and all must be configured for TCT encryption. Every DS8000 must be added to the TCT encryption group of the key manager. This way any DS8000 in the DR configuration can decrypt the data in the cloud, even if it was encrypted by another one.

At the time of writing, the following key management solutions are supported for TCT encryption:

- ▶ IBM Security® Guardium® Key Lifecycle Manager (GKLM) 4.1 or later in multi-master or master-clone configurations:
 - IBM Security Guardium Key Lifecycle Manager Traditional Edition
 - IBM Security Guardium Key Lifecycle Manager Container Edition

Earlier versions of GKLM and its predecessor products might be supported, but they have different requirements.

- ▶ Gemalto SafeNet KeySecure
- ▶ Thales Vormetric Data Security Manager
- ▶ Thales CipherTrust Manager

See *IBM DS8000 Encryption for Data at Rest, Transparent Cloud Tiering, and Endpoint Security (DS8000 Release 9.2)*, REDP-4500 for more details about TCT encryption and implementation instructions..

3.8 Transparent Cloud Tiering secure data transfer with TS7700

If you use a TS7700 Virtualization Engine as the object storage target, TCT supports encryption of the data that transfers over the IP network. Encryption is applied as the data is sent or retrieved to and from the target. This approach is pure TLS and in-flight encryption only: Data is encrypted as it is sent out to the network, and decrypted when it is received by the target port. You don't need an external key manager, as it is not necessary to be able to decrypt data when it is retrieved back from the object storage target.

If the requirements for TCT secure data transfer are fulfilled, you can enable it without further configuration (see Chapter 4, "Requirements" on page 37 and "Connecting to a TS7700 Virtualization Engine" on page 57).

Note: TCT uses IBM Power® hardware-accelerated functions for encryption and compression. Both functions work with line speed and do not affect performance.

3.9 Transparent Cloud Tiering compression with TS7700

If you use a TS7700 Virtualization Engine as the object storage target, TCT also supports compression of the data that is stored on the object storage. Using compression, you can reduce the transfer time and required space on the object storage target. Data that is already compressed or encrypted with native z/OS methods is not compressed again. DFSMS detects that data is already encrypted or compressed and informs the DS8900F not to do compression for this transfer.

If the requirements for TCT compression are fulfilled, you can turn it on without further configuration (see Chapter 4, "Requirements" on page 37 and 7.5, "Enabling TCT compression with a TS7700" on page 77).

At the time of this writing, no compression is performed by the DS8000 during data migration for any of the other object storage targets. However, your data might already be compressed or encrypted when originally stored on the DS8000. Such data is offloaded to cloud in its original condition: compressed or encrypted. (Compression or encryption by the host is typically done by zEDC or pervasive encryption.)

3.10 Selecting data to store in a cloud

When you decide to implement TCT, you also must plan for who can use this cloud and the type of data that you want to store. Defining correct data to be offloaded to cloud gives you more on-premises storage to allocate to other critical data.

As described in this chapter, the cloud should be considered an auxiliary storage option within a z/OS system, meaning that no data that requires online or immediate access should be moved to the cloud.

If you use DFSMSshm as application to migrate and restore data to and from cloud, container and object management is automated. If you decide to use or allow DFSMSdss direct dump and restore with TCT, you must plan and implement ways for manual container and object management.



Part 2

Setting up Transparent Cloud Tiering

In this part, we show you how to set up a Transparent Cloud Tiering (TCT). It includes the following chapters:

- ▶ Chapter 4, “Requirements” on page 37
- ▶ Chapter 6, “Configuring the IBM DS8000 for Transparent Cloud Tiering” on page 49
- ▶ Chapter 7, “Configuring Data Facility Storage Management Subsystem for Transparent Cloud Tiering” on page 65



Requirements

This chapter describes the requirements for Transparent Cloud Tiering (TCT), including IBM DS8000, network, and IBM z/OS environments.

This chapter includes the following topics:

- ▶ 4.1, “Ethernet connections on DS8000” on page 38
- ▶ 4.2, “z/OS level” on page 40
- ▶ 4.4, “DS8000 release level” on page 41
- ▶ 4.6, “Authentication information” on page 43
- ▶ 4.7, “SSL/TLS considerations” on page 44

4.1 Ethernet connections on DS8000

To implement TCT, you need IP connectivity from each of the DS8000 internal servers to the cloud object storage solution. The DS8000 offers two different ways to connect:

- ▶ Two 1 Gbps Ethernet ports per server that are built-in and available on every DS8000 that meets the requirements for TCT.
- ▶ A pair of Ethernet cards, each providing two 10 Gbps (optical SFP+) and two 1 Gbps (RJ45 copper) Ethernet ports. The cards can be purchased with new machines or as a miscellaneous equipment specification (MES) for existing ones.

The built-in 1 Gbps Ethernet card is in location code P1-C10 or P1-C11 (depending on the model). The upper ports T1 and T2 are used for internal communications, and the bottom ports T3 and T4 are available for TCT, as shown in Figure 4-1 (with an IBM DS8910F central processor complex (CPC) as example). They are empty, and typically covered by a plastic port covering. Remove the plastic covering, and insert the RJ45 cable into the ports that you plan to use.



Figure 4-1 Built-in 1 Gbps Ethernet ports for TCT in a DS8910F CPC

The separately available 10 Gbps Ethernet adapters offer higher performance and bandwidth for TCT data movements. Depending on your DS8000 generation, you use one of two versions:

- ▶ For the previous generation DS8880 and DS8880F, use the following information:
 - Feature Code (FC) 3600: TCT 10 Gb/1 Gb Ethernet pair for 2U controllers (IBM DS8884 and IBM DS8884F). It is plugged into location code P1-C11, as shown in Figure 4-2.



Figure 4-2 Location of the 10 Gbps Ethernet cards in 2U DS8884 servers

- FC 3601: TCT 10 Gb/1 Gb Ethernet pair for 4U controllers (IBM DS8886, IBM DS8888, IBM DS8886F, and IBM DS8888F). It is plugged into location code P1-C11, as shown in Figure 4-3 on page 39.

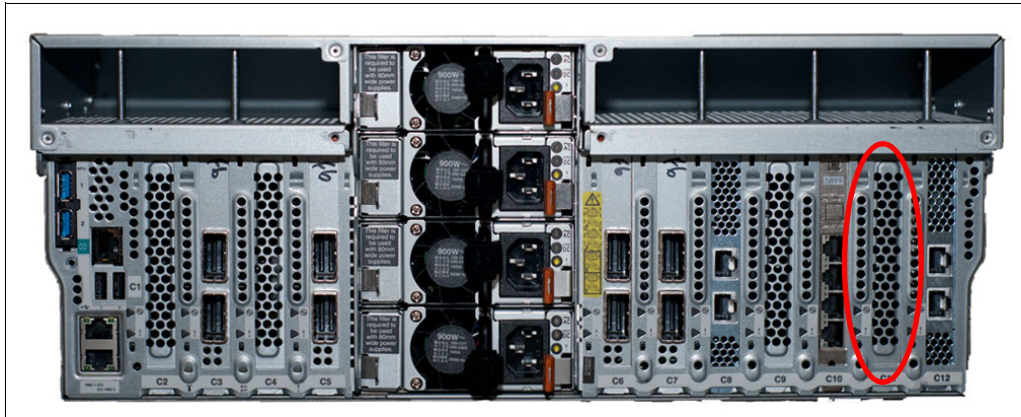


Figure 4-3 Location of the 10 Gbps Ethernet cards in 4U DS8886 servers

They contain two 10 Gbps LR ports (optical SFP+) and two 1 Gbps ports (RJ45 copper). The card is physically in location code P1-C11 or P1-C12 (depending on the model).

- ▶ For the newer DS8900F generation, the cards were changed. They contain two 10 Gbps SR ports (optical SFP+) and 2 x 1 Gbps ports (RJ45 copper) and are partly installed in different locations, as shown in Figure 4-4 and Figure 4-5.
 - FC 3602: TCT 10 Gb/1 Gb Ethernet pair V2 for 2U controllers. It is plugged into location code P1-C4 for model 994 and P1-C11 for model 993.



Figure 4-4 Location of 10 Gbps Ethernet card in 2 U servers (DS8910F model 994)

- FC 3603: TCT 10 Gb/1 Gb Ethernet pair V2 for 4U controllers. It is plugged into location code P1-C10.

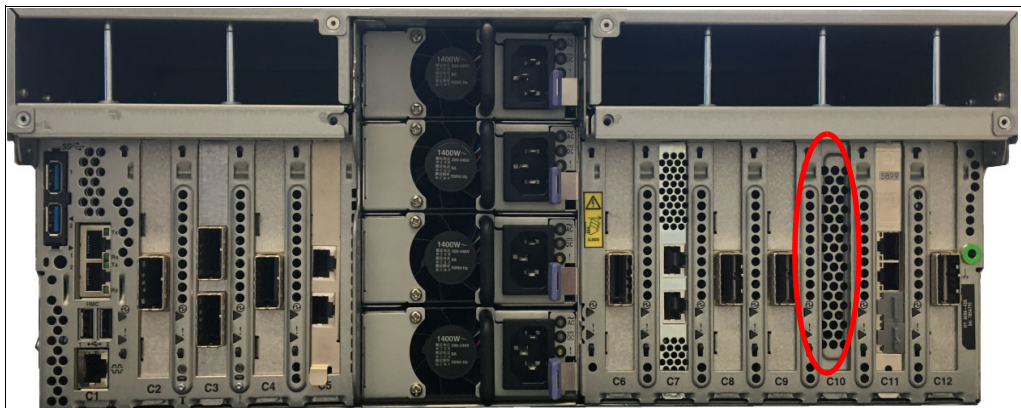


Figure 4-5 Location of 10 Gbps Ethernet card in 4U servers (DS8950F model 996)

All ports on the new cards and the built-in ports can be used for TCT.

With the extra Ethernet cards, you have a total of six Ethernet ports that are available for use with TCT per server. However, port usage and network connectivity have certain limits. For more information, see Chapter 5, “Connectivity and network setup for Transparent Cloud Tiering” on page 45.

Note: To identify and configure the Ethernet ports by using the DS Command-line Interface (DSCLI), you need their port IDs. These IDs depend on the plug location. You can use the DSCLI command `!networkport -1` to determine the port ID against their location code.

4.2 z/OS level

TCT support has been part of the z/OS base operating system since z/OS V2R3. Support for new features is made available for existing z/OS releases with APARs. In this section, we provide a list of the major new feature APARs. Some of these APARs already might be contained in the z/OS release that you are running.

- ▶ [APAR OA55538](#) provides z/OS support for TCT encryption. It enables z/OS so that it can notify the DS8000 if a data set is already encrypted to avoid double encryption.
- ▶ For TCT with a TS7700 as the cloud object target, you might need [APAR OA58225](#).
- ▶ DFSMSdss full volume dump (FVD) and restore is supported on z/OS V2R3 and z/OS V2R4 with [APAR OA57526](#).
- ▶ Support for TCT compression with a TS7700 as a Object Storage target requires [APAR OA59465](#) and its dependent APARS (OA59466, OA59467, OA59468, OA59469, OA59470, and OA59471), which are available for z/OS V2R3 and z/OS V2R4.
- ▶ For TCT secure data transfer with TS7700, there are no extra z/OS requirements.
- ▶ If you plan to use DS8990F multi-cloud connection support, you need z/OS APARs [OA60977](#) and [OA61013](#).
- ▶ Hierarchical Storage Manager (HSM) support for FVD and CDA credential management requires z/OS [APAR OA60278](#).

Note: At the time of writing, the PTFs for APAR OA60278, which support HSM Full Volume Dump and **CDACREDS**, are not available because a problem was discovered. If you intend to use these functions, check for the availability of APAR OA64130, which will fix the issue.

You can use the `IBM.Function.DFSMSCloudStorage` fix category to identify PTFs that are associated with the Data Facility Storage Management Subsystem (DFSMS) TCT support.

4.3 IBM zSystems host system hardware requirements

TCT requires some security and encryption functions on the IBM zSystems host system to handle credentials in a secure fashion.

4.3.1 The CP Assist for Cryptographic Function feature

IBM zSystems FC 3863 *CP Assist for Cryptographic Functions* (CPACF) must be enabled for your IBM zSystems host systems. It allows clear key DES and TDES instructions on all CPs. HSM and DFSMS need CPACF to encrypt and decrypt stored cloud credentials.

4.3.2 IBM Crypto Express feature

DFSMS and HSM use DFSMS *Cloud Data Access* (CDA), an IBM Integrated Cryptographic Service Facility (ICSF) based framework to manage cloud credentials. Installed and configured Crypto Express features allow CDA to wrap and protect the master key that is used to encrypt and decrypt those credentials.

4.4 DS8000 release level

To set up the cloud configuration, your DS8000 must at Release 8.2.3 – Bundle 88.23.19.0 Microcode and DSCLI or later.

To check your current DS8000 microcode level, run the `ver -1` DSCLI command that is shown in Example 4-1.

Example 4-1 Displaying a DS8000 release on the DSCLI

```
dscli> ver -1
DSCLI          7.9.10.223
StorageManager 9.1.2006300253
HMC DSCLI      7.9.10.248
=====Version=====
Storage Image  LMC          Bundle Version
=====
IBM.2107-75LAH81 7.9.10.248 89.10.84.0
```

Note: You also can use the DSCLI commands `1sserver -1` or `1spnode` (available with DS8900F Release 9.1 and later) to determine your current microcode release.

To display the DS8000 release on DSGUI, select **Actions** → **Properties**.

Since the initial release, numerous enhancements were made to the DS8000 TCT functions. Make sure that you have the appropriate code level for the functions that you want to use.

- ▶ DS8880 Release 8.2.3:
 - First release supporting TCT
 - OpenStack Swift API to connect to object storage systems
- ▶ DS8880 Release 8.3:
 - Support for Amazon AWS and IBM Cloud Object Storage through the S3 API
 - Metro Mirror support

- ▶ DS8880 Release 8.3.3:
 - Support for 10 Gbps Ethernet adapters
- ▶ DS8880 Release 8.4:
 - FlashCopy support
- ▶ DS8880 Release 8.5:
 - Global Mirror and Metro Global Mirror support
 - TCT encryption support
- ▶ DS8880 Release 8.5.4 and DS8900F Release 9:
 - A TS7700 as the cloud object storage target
 - An Amazon S3 cloud object storage target
 - Multi-Target Peer to Peer Remote Copy (PPRC) support
- ▶ DS8900F Release 9.1:
 - Compression with a TS7700 as the cloud object storage target
 - Secure data transfer with a TS7700 as the cloud object storage target
 - Support for DFSMSdss FVD and restore (also supported by DS8880 Release 8.5 SP6)
- ▶ DS8900F Release 9.2:
 - Support for up to eight object storage connections
 - Support for Guardium Key Lifecycle Manager (GKLM) Containerized Edition as the key manager for TCT encryption.

4.5 TS7700 release level and features

TS7770 (Models VED) models are supported as cloud object targets in all attachment variations:

- ▶ TS7770T tape-attached
- ▶ TS7770C cloud-attached
- ▶ TS7770 tapeless

All TS7770 Virtual Tape Servers that you want to use as object storage targets must be at least at microcode Release 5.22 and must have the feature *Advanced Object Store for DS8000* FC 5283 enabled. The existing GRID adapters are used for TCT.

Note: Turning on encryption between DS8000 and TS7700 is optional. If you want to turn on encryption then you must install FC5281 on each TS7700 that the DS8000 will directly target.

Note: Some TS7700 machines still have the *DS8000 Object Store* feature (FC 5282) installed. This feature is obsolete and should not be used anymore.

Feature number 5283 (*Advanced Object Store for DS8000*) can be installed only on clusters in a grid where feature number 5282 is *not* installed. Migration services to migrate from the older FC 5282 to the new FC 5283 can be performed with TS7700 R 5.3 or later and are available through a statement of work (SOW) or contract from IBM Lab Services.

4.6 Authentication information

The following account information must be provided by your Cloud Service Provider or Administrator:

- ▶ Endpoint URL with port number
- ▶ Credentials for the used cloud target type
- ▶ Tenant (for Swift)
- ▶ Secure Sockets Layer (SSL) certificates (if using SSL or Transport Layer Security (TLS))

For all cloud target types except Swift, you use the DS8000 Hardware Management Console (HMC) as proxy between z/OS DFSMS and the cloud storage. Therefore, you need connection information for the DS8000 HMC:

- ▶ The HMC IP address or network name.
- ▶ The port number that is used for the cloud proxy connection is 8452.
- ▶ A DS8000 user and password with at least Monitor authority.

4.6.1 Endpoint

The endpoint is the location or URL that the DS8000 (and DFSMS for Swift) uses when accessing and authenticating with the cloud object storage system.

When a swift-keystone authentication method is used, the endpoint must contain the version number of the identity API to use. At the time of writing, only the version 2 API is supported. For example, if the provider endpoint is `https://dallas.ibm.com`, the endpoint should be configured as `https://dallas.ibm.com/v2.0`.

To have access to the endpoint connection, you might also need a port number, which is either already part of your endpoint specification or provided by the cloud storage administrator. The maximum length for the port number is 5 characters, 0 - 65535. You must also ensure that this port is open on the local network firewalls.

4.6.2 Cloud credentials

The cloud administrator provides a set of credentials. Their names and extent differ by cloud target type. You must provide the credentials to the DS8000 when you set up the cloud connection. If you connect to a Swift cloud, you also need these credentials for the DFSMS cloud definition. For more information about the required credentials for the different cloud target types and how to provide them, see Chapter 6, “Configuring the IBM DS8000 for Transparent Cloud Tiering” on page 49 and Chapter 7, “Configuring Data Facility Storage Management Subsystem for Transparent Cloud Tiering” on page 65.

Important: Be careful with the cloud credentials. Anyone with access to them can also access the cloud directly. This access gives the user the power to read, update, or delete the data in the cloud, potentially compromising data integrity or making DFSMShsm unable to recall or restore the data from this cloud account. It is a best practice to have a security administrator who is managing the cloud storage passwords also be the individual who manages the password for DFSMShsm to protect this method of access to the cloud data sets.

For Swift cloud storage environments, an extra abstraction layer is required, which is called Tenant. A Tenant name is the name or project name that identifies your object store environment. This name needs to be something meaningful to your organization's environment, for example, possible Tenant names might be production, development, or test. You can either choose this name when requesting cloud storage access, or it can be predefined by the cloud administrator.

4.6.3 Certificates (if using SSL/TLS)

The first level of encryption-based security provides secure communications between the DS8000 system, DFSMS, and the cloud service provider. The standard protocol, TLS, protects these connections by encrypting authentication data that is transferred between DFSMS, DS8000 systems, and the cloud service provider. Secure communications are mandatory for these connections and require that public certificates are exchanged between the cloud service provider, DFSMS, and the DS8000 systems.

Note: SSL/TLS is used only to encrypt the authentication data between DFSMS, the DS8000, and the cloud object storage. You must configure and enable TCT encryption to encrypt the customer data during transmission and while it is in cloud storage. If you are already using Pervasive Encryption to encrypt data sets on the host, TCT encryption is not required.

For cloud targets that use SSL/TLS to encrypt the authentication path, certificates are required to maintain a chain of trust between DFSMS, the DS8000, and the object store.

If you use self-signed certificates, it is sufficient to provide only them. If you use a CA, you need the CA's root certificate and any intermediate certificates that are required to complete the certificate chain. You provide the certificates wrapped in Privacy-Enhanced Mail (PEM) files that you can import into the DS8000 system and DFSMS. A PEM file can support multiple digital certificates, including a certificate chain.

4.7 SSL/TLS considerations

In Chapter 6, "Configuring the IBM DS8000 for Transparent Cloud Tiering" on page 49 and Chapter 7, "Configuring Data Facility Storage Management Subsystem for Transparent Cloud Tiering" on page 65, we demonstrate how you can configure the DS8000 system and DFSMS to use certificates for secure communications.

DFSMS and IBM DS8000 send account information (usernames and passwords) over an HTTP connection. To ensure that the information is encrypted, as a best practice, establish a secure HTTP connection between the z/OS host, the IBM DS8000 system, and the object storage cloud server.

The supported SSL/TLS versions that are used when making HTTP requests are TLSV12, TLSV11, TLSV1, and SSLV3.

There are two types of authentication:

- ▶ Server authentication: The z/OS host or DS8000 verifies the identity of the object storage cloud server.
- ▶ Mutual authentication: The z/OS host or DS8000 verifies the identity of the object storage cloud server, and the object storage cloud verifies the identity of the z/OS host or DS8000.



Connectivity and network setup for Transparent Cloud Tiering

This chapter describes the general connectivity and network setup for Transparent Cloud Tiering (TCT). It also addresses best practices for a high availability (HA) configuration.

This chapter includes the following topics:

- ▶ 5.1, “Networking communication overview” on page 46
- ▶ 5.2, “DS8000 to object storage connection” on page 46

5.1 Networking communication overview

When planning the network connectivity for the tiered storage, you need to consider two parts for the communication path. Figure 5-1 shows the IP communication paths between the host and cloud object storage.

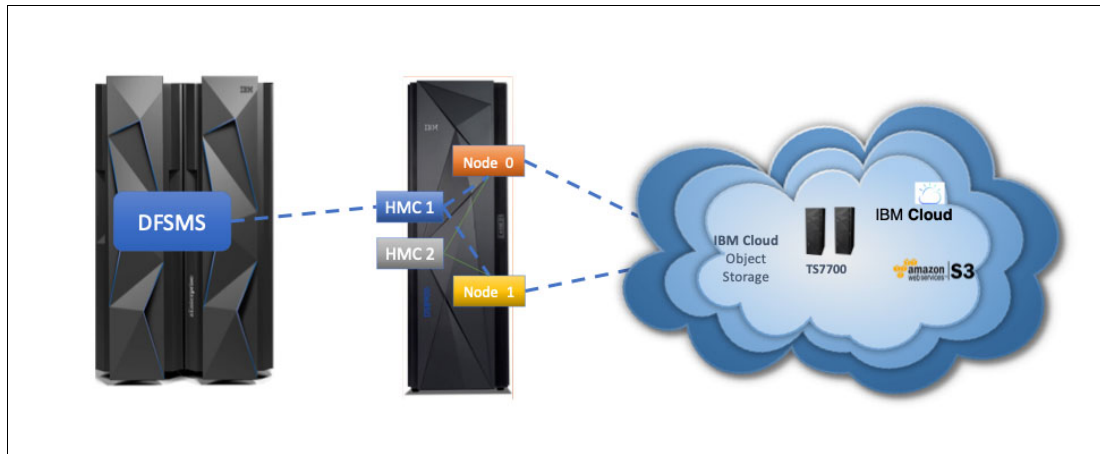


Figure 5-1 IP communication overview

The first part is the communication between DFSMS on the host and the HMC that is acting as cloud proxy (except for SWIFT). This connection is used to exchange the meta-data objects and controls the data that is sent to or received from the cloud.

DFSMS communicates with the HMC through REST API calls. These API calls are issued for a single HMC and you can only configure one HMC connection in DFSMS. Therefore you can come into a situation where the HMC is not available and the API call fails. This situation can occur, for example, when the HMC is updated, rebooted or a service action is performed against the HMC. In this case, TCT operations will fail. You either have to wait until the HMC is operational again or configure another HMC in the DFSMS networking definitions, as described in 7.4, “Creating a DFSMS cloud network connection by using ISMF” on page 70.

The second part is the connection between the DS8000 internal servers and the cloud storage, which is used to transfer the meta data objects and the customer data between the DS8000 and the cloud storage. The configuration of this connection depends on the type of object storage you use and is described in detail in the following sections.

5.2 DS8000 to object storage connection

While connecting to a cloud storage service that is hosted by a cloud provider, you just have to configure the access URI, whereby connecting to a TS7700 or an on-premises cloud storage you might need to take additional considerations to ensure the redundant access.

You can mix and match the following connection methods to prepare a multi-cloud storage configuration.

Note: When connecting the DS8000 to different subnets, the IP gateway definition is for all network ports on the specified processor node. You cannot specify different gateways to individual network ports.

The internal servers communicate only with the cloud endpoint and the DNS service (if defined). Network communication is always initiated by the DS8000 internal servers. Their internal firewall blocks all inbound requests.

5.2.1 Connecting to a TS7700

When using TS7700 Tape Virtualisation Engine as object storage, you have to connect the DS8900F internal servers to the TS7700 grid network.

Each internal server of the DS8000 must be able to communicate with each IP address of the TS7000 that you define in the DS8000 logical configuration (according to “Connecting to a TS7700 Virtualization Engine” on page 57).

If you are using multiple independent grid networks (physically or virtually separated) as shown in Figure 5-2 on page 47, you must connect one network port of each DS8900F internal server to each of the independent networks.

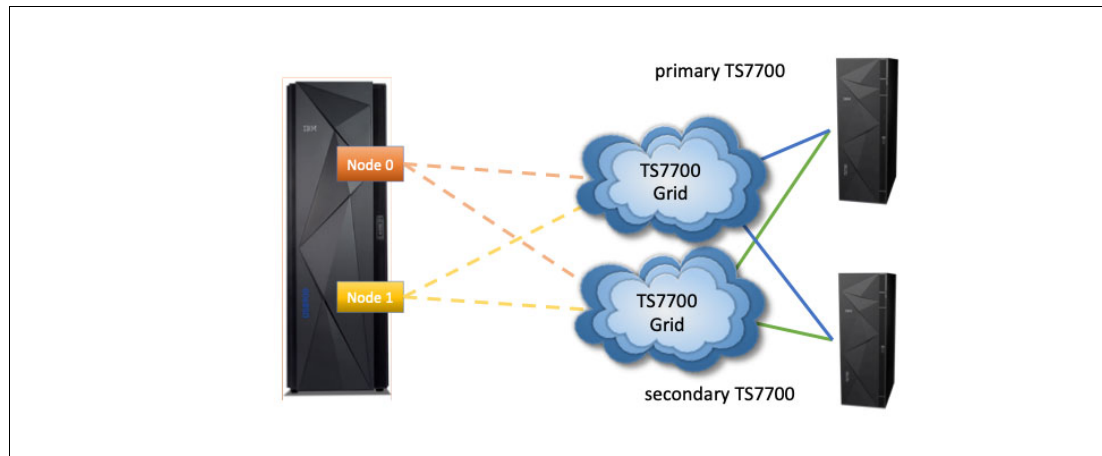


Figure 5-2 TS7700 grid with two physical networks

5.2.2 Connecting to a local cloud object store

When you plan to connect to a privately hosted cloud object store, the access can be either direct or through a load balancer. You can benefit from using a load balancer, as it provides fail-over capabilities and workload distribution to your object store nodes as illustrated in Figure 5-3 on page 48. Follow the cloud object store recommendations to install and configure a load balancer to improve fault tolerance and highly availability.

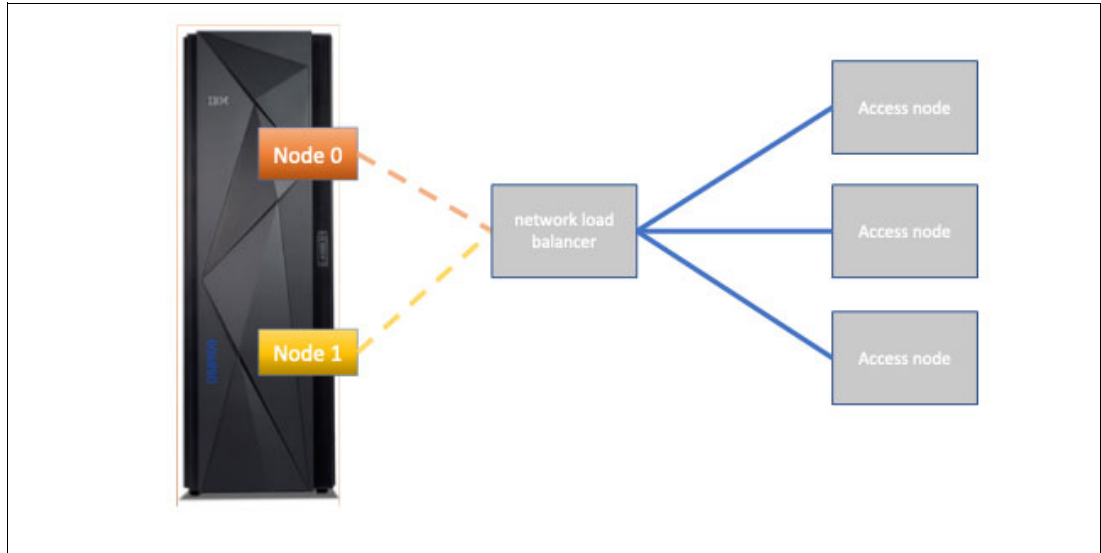


Figure 5-3 Connecting to a local cloud object store through a load balancer

5.2.3 Connecting to a public cloud object storage service

When you connect to a cloud hosted object storage instance, the cloud provider usually provides the URI to access your cloud storage. From a networking perspective, you must ensure that both DS8000 internal servers can access the URI destination. Both internal servers must also be able to access your DNS service to resolve the hostname of your cloud endpoint.

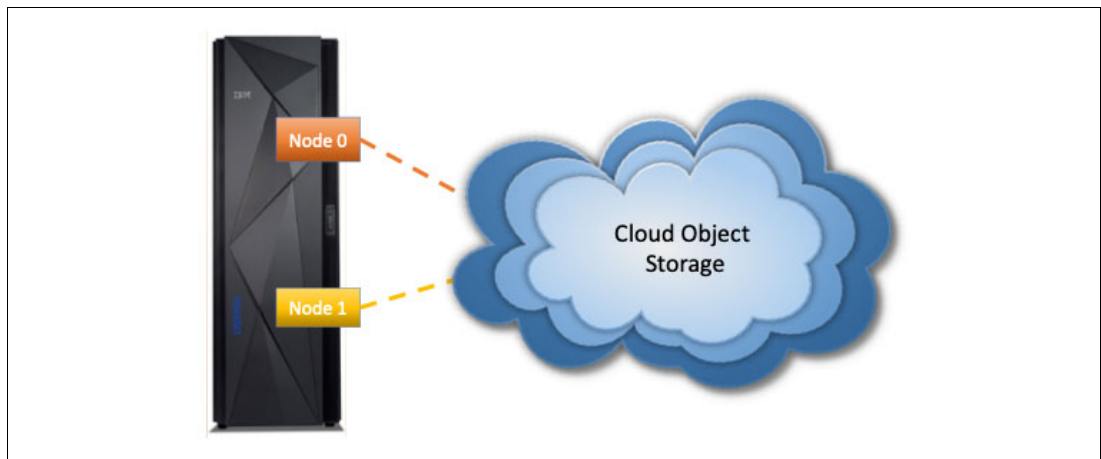


Figure 5-4 Connecting to a cloud object storage at a public or private cloud provider



Configuring the IBM DS8000 for Transparent Cloud Tiering

This chapter describes how to configure the IBM DS8000 to support Transparent Cloud Tiering (TCT).

In the first part, we explain how to set up the network connection. Then, we describe how to set up the cloud connection, providing examples for most supported cloud target types.

Finally, we show how to set up the DS8000 Hardware Management Console (HMC) as cloud proxy for Data Facility Storage Management Subsystem (DFSMS).

This chapter includes the following topics:

- ▶ 6.1, “Configuring the IBM DS8000 for TCT” on page 50
- ▶ 6.2, “Maintaining the cloud server configuration” on page 59
- ▶ 6.4, “Preparing the DS8000 as cloud proxy” on page 61

6.1 Configuring the IBM DS8000 for TCT

To access the cloud services, your DS8000 hardware must be configured to communicate with the cloud by using TCPIP connections. This configuration includes defining the following components:

- ▶ Ethernet port configuration
- ▶ Cloud storage configuration
- ▶ Prepare a DS8000 user ID for the cloud proxy functions (for all cloud target types *except* Swift).

6.1.1 Ethernet configuration

Assuming that you connected the DS8000 to your network according to 4.1, “Ethernet connections on DS8000” on page 38, you can now configure the Ethernet ports that you are using for TCT. You must configure the Ethernet ports enable network connectivity. Use the **lsnetworkport** command from the DS Command-line Interface (DSCLI) to display any current Ethernet configuration. Example 6-1 shows the output of the **lsnetworkport** command where no network ports are configured yet.

Example 6-1 Output from lsnetworkport command with unconfigured network ports

```
dscli> lsnetworkport
```

ID	IP Address	Subnet Mask	Gateway	Primary DNS	Secondary DNS	State
I9831	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	Offline
I9832	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	Offline
I9833	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	Offline
I9834	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	Offline
I98A3	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	Offline
I98A4	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	Offline
I9B31	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	Offline
I9B32	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	Offline
I9B33	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	Offline
I9B34	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	Offline
I9BA3	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	Offline
I9BA4	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	Offline

Example 6-1 lists 12 network ports. The four port IDs that are marked red belong to the 1 Gbps Ethernet ports that are always available in the DS8000 internal servers. The eight port IDs marked blue belong to the TCT network adapter features that can be ordered and installed separately. The port IDs on these adapters are counted from top to bottom. For more information about DS8000 network connectivity for TCT, see 4.1, “Ethernet connections on DS8000” on page 38.

Use the **setnetworkport** command to define the network settings for each port you want to use, as shown in Example 6-2.

Example 6-2 Configuring DS8000 network ports for TCT

```
dscli> setnetworkport -ipaddr 9.155.112.15 -subnet 255.255.240.0 -gateway 9.155.112.1 -primary 9.0.138.50 -secondary 9.0.136.50 I9B32
CMUC00250I setnetworkport: You configured network port I9B32 successfully.
dscli> setnetworkport -ipaddr 9.155.112.16 -subnet 255.255.240.0 -gateway 9.155.112.1 -primary 9.0.138.50 -secondary 9.0.136.50 I9B32
CMUC00250I setnetworkport: You configured network port I9B32 successfully.
```

You must specify at least the **-ipaddr** and **-subnet** parameters, defining the IP address and the subnet mask for a port. If you need routing to access the cloud storage target from the DS8000 TCT network connections, you can specify a gateway by using the **-gateway** parameter. If you need name server resolution to access the object storage endpoint, you can specify up to two DNS servers by using the **-primary** and **-secondary** parameter.

Note: The **-gateway**, **-primary**, and **-secondary** parameters set the default gateway and DNS servers for all TCT network connections of a DS8000 internal server. You can have only one gateway and one set of DNS servers. Therefore, routing into different networks is not possible. All cloud targets must either be accessible through the local subnet or a single gateway.

It is possible to route each internal server’s connection through different subnets if each server can access all cloud targets.

After the configuration is complete, use the **lsnetworkport** command again to confirm that your network configuration is correct. Example 6-3 shows a sample network configuration with four configured ports.

Example 6-3 Verifying the network configuration

```
dsccli> lsnetworkport
```

ID	IP Address	Subnet Mask	Gateway	Primary DNS	Secondary DNS	State
I9831	0.0.0.0	0.0.0.0	9.155.112.1	9.0.138.50	9.0.136.50	Offline
I9832	9.155.112.16	255.255.240.0	9.155.112.1	9.0.138.50	9.0.136.50	Online
I9833	192.168.100.51	255.255.255.0	9.155.112.1	9.0.138.50	9.0.136.50	Online
I9834	0.0.0.0	0.0.0.0	9.155.112.1	9.0.138.50	9.0.136.50	Offline
I98A3	0.0.0.0	0.0.0.0	9.155.112.1	9.0.138.50	9.0.136.50	Offline
I98A4	0.0.0.0	0.0.0.0	9.155.112.1	9.0.138.50	9.0.136.50	Offline
I9B31	0.0.0.0	0.0.0.0	9.155.112.1	9.0.138.50	9.0.136.50	Offline
I9B32	9.155.112.15	255.255.240.0	9.155.112.1	9.0.138.50	9.0.136.50	Online
I9B33	192.168.100.52	255.255.255.0	9.155.112.1	9.0.138.50	9.0.136.50	Online
I9B34	0.0.0.0	0.0.0.0	9.155.112.1	9.0.138.50	9.0.136.50	Offline
I9BA3	0.0.0.0	0.0.0.0	9.155.112.1	9.0.138.50	9.0.136.50	Offline
I9BA4	0.0.0.0	0.0.0.0	9.155.112.1	9.0.138.50	9.0.136.50	Offline

The ports marked blue are configured for routing. The defined gateway is in their network segment. They can be used, for example, to connect to a public cloud endpoint.

The ports marked red are in a private network. They are not able to reach the gateway. They can be used to connect, for example, to an on-premises object storage or a TS7700 Grid that is in the same private network segment.

Note: Network ports cannot be created or deleted. The only way to unconfigure a port is to clear the configured IP address by setting it to “0.0.0.0”.

With the Ethernet configuration complete, you can proceed to the cloud configuration process.

Note: You can have up to six Ethernet ports available in each DS8000 internal server. It is a best practice to use only ports of the same type, for example only the 10 Gbps ports, to avoid performance imbalance. For more information about networking requirements, see Chapter 5, “Connectivity and network setup for Transparent Cloud Tiering” on page 45.

6.1.2 Cloud configuration

In this step, you configure the DS8000 for access to the cloud storage. DS8000 cloud connections are managed by using DSCLI commands. There is no DS8000 GUI equivalent available.

Important: Since Release 9.2, the DS8900F supports up to eight independent cloud connections. Since this release, the way these connections are managed has changed:

- ▶ If you define only one connection, everything works as before. All commands (create or delete connection) have an immediate effect.
- ▶ As soon as you define more than one connection, any modification to the cloud connections requires an extra activation step. This step is disruptive to ongoing TCT operations. It removes and re-creates all defined cloud connections. Active TCT data transfers are terminated and must be restarted by the application that initiated them.

For more information and examples, see 6.3, “Managing multiple cloud connections” on page 60.

Use the **mkcloudserver** command to define a cloud object storage to your DS8000 system. Generally you provide the cloud target type, endpoint information, and cloud credentials with the command. However, the usage differs for the different cloud target types.

The following parameters are available for the **mkcloudserver** command:

- ▶ **-type** (required): The cloud object storage target type. TCT supports the following types of Object Storage protocols and authentication mechanisms:
 - **swift**: Unencrypted (HTTP) communication to Swift cloud object storage.
 - **swift-keystone**: Secure Sockets Layer (SSL)/Transport Layer Security (TLS) secured authentication to a Swift keystone service to access Swift cloud object storage.
 - **ibmcos**: IBM Cloud Object Storage either on premises or in the IBM Cloud.
 - **aws-s3**: Amazon Simple Storage Service cloud object storage that uses the S3 API.
 - **ts7700**: IBM Virtual Tape Server TS7700 as cloud object storage.
- ▶ **-endpoint** (required for all cloud target types except ts7700): The Uniform Resource Identifier (URI) to access the cloud object storage. For the **swift-keystone** type, it is the URI of the keystone authentication service.
- ▶ **-tenant** (required for the **swift** and **swift-keystone** types, not used for all others): Specify the Tenant name that is provided by cloud storage administrator.
- ▶ **-username** (required for all cloud target types, not allowed for TS7700): A user identifier for the cloud object storage account. For Amazon S3 type object storage solution, use the access key that is provided by the cloud administrator.
- ▶ **-pw** (required for all cloud target types, not allowed for TS7700): A user credential for the cloud object storage account. For Amazon S3 type object storage solution, use the secret access key that is provided by the cloud administrator.
- ▶ **-noss1** (optional): Allow insecure authentication with the object storage target. This parameter cannot be specified together with **-rootcaloc**, **-intermcaloc**, or **-syscaloc**.

Note: TCT secure data transfer with TS7700 is supported for DS8900F R 9.1 and later. For prior releases, the **-noss1** option is required for TS7700 as object storage target.

- ▶ **-rootcaloc, intermcaloc, syscaloc** (required for **swift-keystone** and all Amazon S3 type targets if IP security is used. Optional for the TS7700 target type): Use these parameters to provide certificates to the DS8000 for secure authentication with the object storage. Specify the location of the certificate (Privacy-Enhanced Mail (PEM)) file that you want to import into the DS8000 system with each parameter. If you use self-signed certificates, only the SysCA option is required. If you use a CA, the root CA and intermediate CA can be provided. These parameters cannot be used if **-noss1** is specified.

Note: If you use TCT secure data transfer with TS7700, you must provide certificates only if you changed the TS7700 secure data transfer certificate. The DS8900F knows and accepts the factory-provided certificates of the TS7700.

- ▶ **-loc** (optional and valid only for **aws-s3** and **ibmcos** type targets):
 - **ibmcos**: Specify the name of your vault template on the IBM Cloud Object Storage service.
 - **aws-s3**: Specify the *AWS Region* as defined by the endpoint of the Amazon S3 service.
- ▶ **-keygrp** (required if DS8000 TCT encryption is used, not valid for TS7700): Specify the key group that you defined for TCT encryption (for more information, see 3.6, “DS8900F multi-cloud support” on page 30).
- ▶ **-primary7700IPs** (required and valid only for TS7700): Specify IP addresses of the primary TS7700 that you use as TCT cloud storage. You can specify up to four IP addresses (only IPv4 is supported). Separate them with a comma.
- ▶ **-secondary7700IPs** (optional and valid only for TS7700): Specify IP addresses of the secondary TS7700 that you use as TCT cloud storage. You can specify up to four IP addresses (only IPv4 is supported). Separate them with a comma.
- ▶ **cloud_name** (required): Specify a name for your cloud connection. Use the same name as in your DFSMS cloud definition (for more information, see 7.4, “Creating a DFSMS cloud network connection by using ISMF” on page 70). This parameter is a positional one. Place it at the end of the command without a keyword.

When you run the **mkcloudserver** command, the ability of the DS8000 and the object store to communicate is verified. Running the command also verifies that the data path is accessible and encryption certificates are valid. We provide examples for most supported cloud target types in the following sections.

Note: Anybody with access to the cloud credentials you use to connect a DS8000 to cloud storage has full access to all objects TCT stores in the cloud, including the capability to update, move, or delete data. Make sure that you treat these credentials with the appropriate care.

Connecting to on-premises IBM Cloud Object Storage

In this section, we explain how to connect your DS8000 to an on-premises IBM Cloud Object Storage System. Before connecting the DS8000, you must prepare the IBM Cloud Object Storage. For more information about this topic, see [Configuring transparent cloud tiering](#).

During the configuration, you create an IBM Cloud Object Storage user and a Vault Provisioning Profile. To set up the DS8000 cloud connection, you need the Access Key and the Secret Access Key for this user and the provisioning code for the profile.

In Example 6-4, we show the command to configure the DS8000 cloud connection for the simple case without SSL and encryption.

Example 6-4 DS8000 cloud connection to IBM Cloud Object Storage without SSL and encryption

```
dscli> mkcloudserver -type ibmcos -username zbg...DId -pw jMn...AcA -noss1 -endpoint
http://9.155.115.167 -loc ztct IBMCOS
```

```
CMUC00560W mkcloudserver: Use of the -noss1 flag allows user credentials such as username and
password to be transmitted on the network insecurely. Are
you sure that you want to continue? [y/n]: y
```

```
CMUC00505I mkcloudserver: The entered Cloud server IBMCOS was created successfully on node 0.
CMUC00505I mkcloudserver: The entered Cloud server IBMCOS was created successfully on node 1.
```

We used the following command options:

- ▶ **type**: The cloud storage target type `ibmcos`
- ▶ **endpoint**: The IP address of one of the IBM Cloud Object Storage accessor nodes
- ▶ **username**: The Access Key for the IBM Cloud Object Storage user
- ▶ **pw**: The Secret Access Key for the IBM Cloud Object Storage user
- ▶ **noss1**: Connect without authorization security
- ▶ **loc**: The provisioning code for the IBM Cloud Object Storage Vault Provisioning Profile
- ▶ The last parameter is a required positional parameter without keyword. Specify the name of your DS8000 cloud connection here.

Note: In our example, we specified the IP address of one accessor node of the IBM Cloud Object Storage, which is a single point of failure. For a production configuration, you would either provide the IP address of a load balancer that has access to several accessor nodes, or the virtual address of an accessor node pool.

In Example 6-5, we show another IBM Cloud Object Storage connection, this time with SSL and encryption support.

Example 6-5 DS8000 cloud connection to IBM Cloud Object Storage with SSL and encryption

```
dscli> mkcloudserver -type ibmcos -username zbg...DId -pw jM...AcA -endpoint http://9.155.115.167 -syscaloc
ibmcos_sle_certificates.pem -loc ztct -keygrp 2 ibmcos
```

```
CMUC00505I mkcloudserver: The entered cloud ibmcos was created successfully on node 0.
CMUC00505I mkcloudserver: The entered cloud ibmcos was created successfully on node 1.
```

The meaning of the parameters for type, username, password, endpoint, and cloud connection name are the same as in Example 6-4 on page 54. We still provide a URL starting with `http:` for the endpoint, although we specified that SSL is being used. The systems will switch to `https` communication automatically after successful SSL negotiation.

We use the following extra parameters:

- ▶ **syscaloc**: Specifies the file containing the IBM Cloud Object Storage System certificate. You can download it from the IBM Cloud Object Storage Manager.
- ▶ **keygrp**: Specify the key group used for TCT encryption.

Note: Before you can use TCT encryption, you must configure your IBM Security Key Lifecycle Manager servers and DS8000 encryption settings (key server and key group) according to *IBM DS8000 Encryption for Data at Rest, Transparent Cloud Tiering, and Endpoint Security (DS8000 Release 9.2)*, REDP-4500.

Connecting to the IBM Cloud Object Storage service in the public IBM Cloud

In this example, we connect the DS8000 to the IBM Cloud Object Storage service in the public IBM Cloud.

After you signed up to the IBM Cloud and defined the IBM Cloud Object Storage instance, you can create credentials for your service. The credentials are provided in JSON format like that shown in Example 6-6.

Example 6-6 Credential that is provided by IBM Cloud for IBM Cloud Object Storage

```
{
  "apikey": "LB1Iki88rfqXJSy1oXF1An8i0WLu_lzigjn0xTpAYG73",
  "cos_hmac_keys": {
    "access_key_id": "f9ae19f116de4a37a9e9f0e7d80348e3",
    "secret_access_key": "97f858f693dfcbf4236b16b1c6f512295fcb99d034b138ad"
  },
  "endpoints": "https://control.cloud-object-storage.cloud.ibm.com/v2/endpoints",
  "iam_apikey_description": "Auto-generated for key
f9ae19f1-16de-4a37-a9e9-f0e7d80348e3",
  ...
}
```

Use the `access_key_id` and `secret_access_key` as username and password in the `mkcloudserver` command.

The endpoints definition does not directly provide an endpoint. It refers to a web page with all endpoints for the IBM Cloud Object Storage services. We show a section of this page in Figure 6-1.

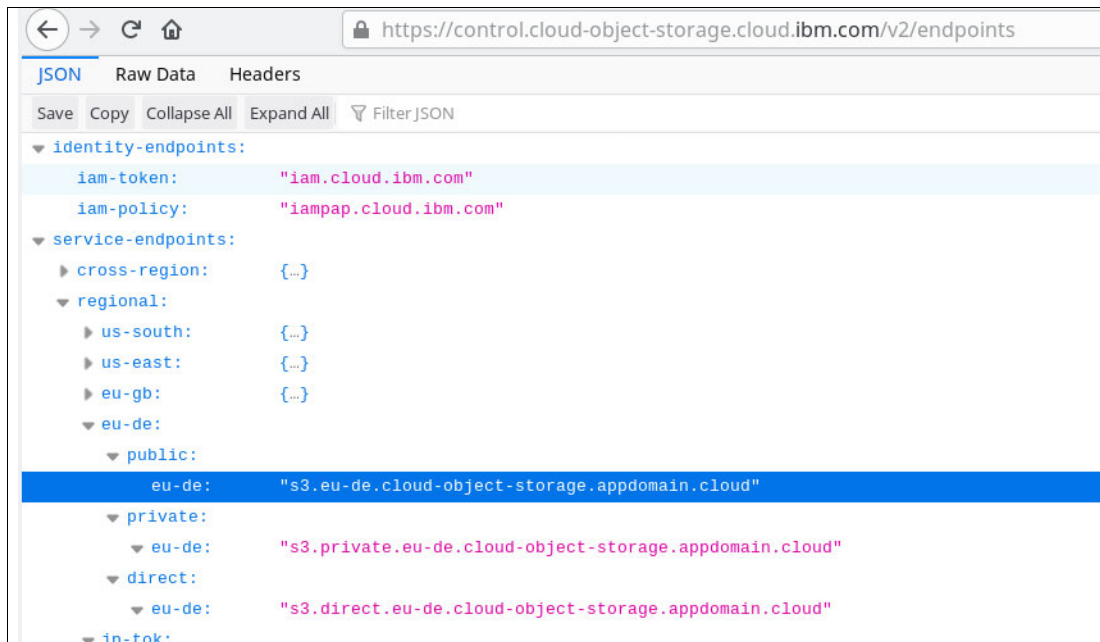


Figure 6-1 IBM Cloud Object Storage service endpoints

Find the endpoint that matches the regional settings of the IBM Cloud Object Storage service you defined and use it for the endpoint parameter of the `mkcloudserver` command, as shown in Example 6-7.

Example 6-7 DS8000 cloud connection to the public IBM Cloud without SSL and encryption

```
dscli> mkcloudserver -type ibmcos -username 648...630 -pw f7b...e6f -noss1 -endpoint
http://s3.eu-de.cloud-object-storage.appdomain.cloud pubcos
CMUC00560W mkcloudserver: Use of the -noss1 flag allows user credentials such as username and
password to be transmitted on the network insecurely. Are you sure that you want to continue?
[y/n]: y
CMUC00505I mkcloudserver: The entered cloud pubcos was created successfully on node 0.
CMUC00505I mkcloudserver: The entered cloud pubcos was created successfully on node 1.
```

Note: If you want to use SSL for secure authentication with the IBM cloud, you can download the endpoint security certificate (for example, by using the `openssl` command). Then, provide the certificate file in the `sysca1oc` parameter of the `mkcloudserver` command.

Connecting to minio compatible object storage

In Example 6-8 on page 56, we show a command that creates a connection to our own on-premises IBM Cloud Object Storage based on the open source project minio (minio.io).

Example 6-8 DS8000 cloud connection to a minio S3 target without security and encryption

```
mkcloudserver -type s3 -username Y0B...8SX -pw 7Sv...Iex -noss1 -endpoint
http://9.155.49.146:9000 minioz
CMUC00560W mkcloudserver: Use of the -noss1 flag allows user credentials such as username and
password to be transmitted on the network insecurely. Are you sure that
```



```
you want to continue? [y/n]: y
CMUC00505I mkcloudserver: The entered cloud minioz was created successfully on node 0.
CMUC00505I mkcloudserver: The entered cloud minioz was created successfully on node 1.
```

The parameters that we must specify follow the same rules as in “Connecting to on-premises IBM Cloud Object Storage” on page 54, except for the `-loc` parameter. The S3 cloud target type does not support a location specification.

Connecting to a TS7700 Virtualization Engine

Before you can connect a DS8000 to a TS7700 Virtualization Engine (VE), the VE must be prepared:

- ▶ A license for the DS8000 Object Store feature (Feature Code (FC) 5283) is required.
- ▶ For TCT secure data transfer with TS7700, the TS7700 Secure Data Transfer feature (FC 5281) is also required.
- ▶ The client activates the feature.
- ▶ An IBM service representative configures the TS7700:
 - Configures the system to handle object data and creates an object partition.
 - Defines the DS8000 systems that can store objects on this machine, by providing the IP addresses and serial numbers.
- ▶ The client can now set the size of the Object Partition as needed.

If the TS7700 is a stand-alone system, there is an extra step to connect the TS7700 grid links to the network.

Additional information: For more information, see *IBM TS7700 Series DS8000 Object Store User's Guide Version 2.0*, REDP-5583. This publication is not updated yet to cover the changes that were introduced with TS7700 release 5.22 and FC 5283.

The connection to a TS7700 Virtualization Engine is different from the other cloud types because we need no cloud credentials. The DS8000 systems that are allowed to access the TS7700 as cloud storage are defined in the TS7700 by their IP addresses and serial numbers.

In Example 6-10 on page 58, we show the DSCLI command to connect a DS8000 to a single TS7700.

Example 6-9 DS8000 cloud connection to an IBM TS7700 Virtualization Engine

```
dscli> mkcloudserver -type TS7700 -primary7700IPs 192.168.100.1,192.168.100.2 TS7700
CMUC00505I mkcloudserver: The entered cloud TS7700 was created successfully on node 0.
CMUC00505I mkcloudserver: The entered cloud TS7700 was created successfully on node 1.
```

The following parameters are required:

- ▶ **type**: Specify the cloud target type, TS7700.
- ▶ **primaryTS7700IPs**: A comma-separated list of IP addresses of the Grid links of the TS7700 that you want to use for TCT data transfer.
- ▶ The last parameter again is a required positional parameter without keyword. Specify the name of your DS8000 cloud connection here.

Note: With DS8900F R 9.1 and later, TCT supports compression with TS7700 as object storage. Compression is enabled or disabled in z/OS. You enable TCT compression in z/OS, as described in 7.5, “Enabling TCT compression with a TS7700” on page 77. You do not need to configure anything in the DS8900F or TS7700 to use TCT compression.

TCT secure data transfer with TS7700 is supported for DS8900F R 9.1 and later. For prior releases, the **-noss1** option is required, as shown in Example 6-10.

Example 6-10 DS8000 cloud connection to an IBM TS7700 Virtualization Engine without secure data transfer

```
dscli> mkcloudserver -type TS7700 -primary7700IPs 192.168.100.1,192.168.100.2 -noss1 TS7700
CMUC00560W mkcloudserver: Use of the -noss1 flag allows user credentials such as username and
password to be transmitted on the network insecurely. Are you sure that you want to continue?
[Y/N]: y
CMUC00505I mkcloudserver: The entered cloud TS7700 was created successfully on node 0.
CMUC00505I mkcloudserver: The entered cloud TS7700 was created successfully on node 1.
```

You can optionally specify a second TS7700 by using the **secondary7700IPs** parameter.

Connecting two TS7700 machines provides redundancy and improves performance. Object data transfers are balanced across the defined TS7700 systems.

With the new feature *DS8000 Object Store Grid Policy Management (FC 5283)*, object replication is handled by the TS7700 engine and defined through TS7700 *Object Policies* and selected through different DS8000 and DFSMS cloud connections.

Note: You can specify up to four IP addresses for each TS7700. Only one link is used at a time. The others are for redundancy. Only IPv4 is supported.

Connecting to on-premises Swift cloud object storage

Example 6-11 shows a sample **mkcloudserver** command to configure a Swift Keystone authenticated cloud object storage as cloud server.

Example 6-11 Configuring a Swift cloud server

```
dscli> mkcloudserver -type swift-keystone -tenant tenant -username username
-pw password -endpoint http://9.155.117.45:5000/v2.0/ -rootcaloc
/home/ssl_cacert.pem -intermcaloc /home/user/ssl_cacert.pem -syscaloc
/home/user/ssl_cert.pem ibmcloud
```

For the swift and swift-keystone cloud target types, the cloud credentials are provided according to the following list:

- ▶ **tenant**: The Tenant name that you were given (or specified yourself) when you signed up for the Swift cloud service. The tenant is sometimes also referred to as a Project.
- ▶ **username**: User that can access to objects of your tenant (or Project).

- ▶ **pw**: The password for this user.
- ▶ **rootcaloc**, **intermcaloc**, or **syscaloc**: For the swift-keystone type that uses SSL/TLS to encrypt the authentication path, certificates are required to maintain a chain of trust between the DS8000 and the object store. If you use self-signed certificates, only the SysCA option is required. If you use a CA, the root CA and intermediate CA can be provided in the **mkcloudserver** command. These items point to a PEM file type that you can import into the DS8000 system.
- ▶ The last parameter again is a required positional parameter without keyword. Specify the name of your DS8000 cloud connection here.

6.2 Maintaining the cloud server configuration

The following sections show how to list, update, or remove a cloud server configuration.

6.2.1 Listing a cloud server configuration

You can also list the cloud configuration on your hardware. Use the **lsccloudserver** command to list cloud information. Sensitive information, such as user ID and password, is not displayed in the output. Example 6-12 displays a sample output from the **lsccloudserver** command.

Example 6-12 Listing cloud information

```

dscli> lsccloudserver
Date/Time: November 30, 2017 2:24:31 PM MST IBM DSCLI Version: 7.8.31.118 DS: -
name      node type      tenant  endpoint
ibmcloud  0 swift-keystone test    https://ibmcloud.ibm.com:5000/v2.0/
ibmcloud  1 swift-keystone test    https://ibmcloud.ibm.com:5000/v2.0/

```

You can use the **lsccloudserver -l** command to display more detail about the cloud connection. Example 6-13 shows the output for a connection to a TS7700 with secure data transfer enabled (**noss1 = false**).

Example 6-13 Listing detailed cloud information

```

dscli> lsccloudserver -l
name  node type  tenant user endpoint IP address          Location noss1 keygrp
-----
TS7700  0 TS7700 -    -    -    192.168.100.1,192.168.100.2 -    false    -
TS7700  1 TS7700 -    -    -    192.168.100.1,192.168.100.2 -    false    -

```

6.2.2 Updating or removing a cloud server configuration

If you need to update any cloud settings, the existing configuration must first be deleted. Then, the new configuration can be defined. Use the **rmcloudserver** command to remove the existing cloud configuration. When the command is issued, a prompt message displays requesting a confirmation if the cloud server configuration is to be removed. If you want to continue with the removal, enter *y*, as shown in the Example 6-14.

Example 6-14 Removing a cloud server configuration

```
dscli> rmcloudserver ibmcloud
Are you sure you want to delete cloud server ibmCloud? [y/n]:y
The cloud server ibmcloud successfully deleted.
```

After deleting the old configuration, you can add a configuration by using the **mkcloudserver** command.

6.3 Managing multiple cloud connections

Since Release 9.2, a DS8900F supports up to eight independent cloud connections. Since this release, the way these connections are managed changed:

- ▶ If you define only one connection, everything works as before. All commands (create or delete connection) have an immediate effect.
- ▶ Starting with the second connection, any modification to the cloud connections requires an extra activation step.

If you define only one cloud target, or with the first definition, the **mkcloudserver** command becomes effective immediately. The command verifies the ability of the DS8000 and the object store to communicate and perform all necessary object storage operations, and whether the encryption certificates are valid. A connection removal in a single cloud setup also is effective immediately.

Starting with the second cloud connection definition, the **mkcloudserver** command accepts the configuration parameters and creates the internal configuration structures. Validation and activation of the connection happen only after you issue the activation command. Removing a cloud connection in a multi-cloud configuration also requires the activation step. To activate a modification, use the **managecloudserver** command with the **-action applypdgconfig**.

Note: If you remove all connections except the last one from a multi-cloud configuration, the configuration turns into a single-cloud configuration.

Example 6-15 on page 61 shows the commands that add and activate a second cloud connection to an existing one.

Example 6-15 Defining and activating a second cloud definition

```
dscli> mkcloudserver -type TS7700 -primary7700IPs 192.168.100.1,192.168.100.2 TS7700
CMUC00602W mkcloudserver: The cloud TS7700 was successfully created on node 0. To activate the cloud, run
managecloudserver -action applypndgconfig command.
CMUC00602W mkcloudserver: The cloud TS7700 was successfully created on node 1. To activate the cloud, run
managecloudserver -action applypndgconfig command.
```

```
dscli> lsccloudserver
```

name	node	type	endpoint	IP address	Location	keygrp	State
ibmcos	0	ibmcos	http://9.155.115.167	-	ztct	-	- Active
ibmcos	1	ibmcos	http://9.155.115.167	-	ztct	-	- Active
ts7700	0	TS7700	-	192.168.100.1,192.168.100.2	-	-	- Activation Pending
ts7700	1	TS7700	-	192.168.100.1,192.168.100.2	-	-	- Activation Pending

```
dscli> managecloudserver -action applypndgconfig
CMUC00607W managecloudserver: This is a disruptive operation that affects all the cloud configurations. Are
you sure you want to apply pending configuration changes? [Y/N]: y
CMUC00601I managecloudserver: The cloud ts7700 was successfully updated to Active state on node 0.
CMUC00601I managecloudserver: The cloud ts7700 was successfully updated to Active state on node 1.
CMUC00601I managecloudserver: The cloud ibmcos was successfully updated to Active state on node 0.
CMUC00601I managecloudserver: The cloud ibmcos was successfully updated to Active state on node 1.
```

The first command (**mkcloudserver**) defines the cloud connection, but it does not activate the connection yet. (The message at command completion indicates this status.) With the second command (**lsccloudserver**), we show how the DS8000 displays the intermediate state of the cloud connections. Finally, with the third command (**managecloudserver**), we activate the newly defined connection. You can add more than one cloud connection and activate them all with a single **managecloudserver** command.

Note: A **managecloudserver -applypndgconfig** command is disruptive to ongoing TCT operations. It removes and re-creates all defined cloud connections. Active TCT data transfers are terminated.

If you remove or modify one or more cloud connections in a multi-cloud configuration, the activation step with the **managecloudserver** command also is required.

Note: A major use case for multiple DS8000 cloud connections is the selection of TS7700 object policies. Using the TS7700 user interface, you can define policies for stored objects that specify where in the TS7700 grid objects are stored and how they are replicated.

To call such a policy, you must define a DS8000 cloud connection to the TS7700 grid and name it like the policy that you want to use. If you have more than one object policy, you define a DS8000 cloud connection for each of them.

6.4 Preparing the DS8000 as cloud proxy

With the TCT feature, the DS8000 acts a cloud proxy for the mainframe. This capability enables support for Amazon S3, IBM Cloud Object Storage, and TS7700 cloud interfaces for TCT. This way, z/OS and DFSMS do not have to connect to the cloud object storage directly, but use the DS8000 to relay the cloud requests. We describe required preparations for this function in the following sections.

6.4.1 Configuring the DS8000 for the REST API proxy function

This section describes the following topics:

- ▶ Creating a proxy user
- ▶ Exporting the security certificate

Creating a proxy user

You need a DS8000 user ID that is used by DFSMS to authenticate on the DS8000 system by using the Representational State Transfer (REST) API interface. The REST API proxy service is automatically enabled when the DS8000 Code level is upgraded to R8.3 or later. No other tasks are required to enable the communication other than configuring the network and the user ID that is used by DFSMS. Example 6-16 shows how to create a local user ID in the DS8000 by using DSCLI.

Example 6-16 Creating a DS8000 user ID for DFSMSHsm to connect

```
dsccli> mkuser -pw REDxxxx -group monitor itsouser
```

```
Date/Time: December 13, 2017 8:56:41 AM MST IBM DSCLI Version: 7.8.31.118 DS: -
```

```
CMUC00133I mkuser: User itsouser successfully created.
```

When a user ID is created on the DS8000, the initial password that is used in the **mkuser** command is temporary and expired. You must change to the final password that will later be used by DFSMS, by logging on as that user and issuing a **chuser** command. Alternatively, this can also be done by your DS8000 security administrator. You can also use the DS8000 GUI to create the needed user ID and change the password.

After the user ID is created and the password is changed, you will be able to connect the DFSMS to the DS8000 by using the steps described in Chapter 7, “Configuring Data Facility Storage Management Subsystem for Transparent Cloud Tiering” on page 65.

Attention: The user ID that is created for the DFSMS to connect to the DS8000 follows the security rules that are defined in the Authentication Policy of the DS8000. Therefore, depending on your policy rules, this password can expire after a certain number of days. To avoid connectivity issues with TCT, change the password of this user ID both in the DS8000 and the DFSMSHsm before the expiration date, or modify the expiration date policy to never expire.

Optionally, you can use LDAP to create the user ID that will be used by DFSMSHsm to connect to the DS8000 REST API Proxy interface. The step-by-step instructions to configure LDAP on the DS8000 system can be found in *LDAP Authentication for IBM DS8000 Systems: Updated for DS8000 Release 9.1*, REDP-5460.

Exporting the security certificate

Section 4.6.3, “Certificates (if using SSL/TLS)” on page 44 describes how to secure the communication between the DS8000 system, DFSMS, and the cloud service provider. You can use encryption-based security with certificates that are exchanged during the authentication process, which can be External CA (signed by a third-party CA) or self-signed certificates.

Also, if you use Amazon S3 or IBM Cloud Object Storage cloud types, DS8000 uses a REST API Proxy interface to communicate with DFSMS. To do so, DFSMS connects to the DS8000 by using its HTTPS interface, so the certificate used by the DS8000 must be added to the IBM Resource Access Control Facility (RACF) for the DFSMSHsm authentication to be successful.

By default, each DS8000 has factory-created communication certificates that are installed. They can be used to secure the communication between DFSMS and the DS8000 cloud proxy function.


To transfer the communication certificate to your z/OS host, first download it to your local workstation, for example, by using the `openssl` tool, as shown in Example 6-17. Any other tool that can download, extract, and dump certificates to PEM files can also be used.

Example 6-17 Downloading the DS8000 HMC self-signed certificate by using openssl

```
openssl x509 -in <(openssl s_client -connect <DS8000-HMC-IP>:8452 -prexit  
2>/dev/null) -text -out certificate.pem
```

The procedure to upload the certificate from your workstation to the z/OS host is demonstrated in 7.2.1, “Uploading the certificate files to the z/OS host” on page 66.

If you must replace the DS8000 certificates, for example, because your organization requires the use of certificates that are signed by a designated CA, see Appendix C, “Replacing communication certificates in the IBM DS8900F” on page 145.



Configuring Data Facility Storage Management Subsystem for Transparent Cloud Tiering

This chapter describes how to configure z/OS Data Facility Storage Management Subsystem (DFSMS) and Hierarchical Storage Manager (HSM) to be able to use Transparent Cloud Tiering (TCT). We use the Interactive Storage Management Facility (ISMF) interface to define the cloud network connection to DFSMS.

This chapter includes the following topics:

- ▶ 7.1, “DFSMS connections to cloud” on page 66
- ▶ 7.2, “Adding digital certificates to IBM Resource Access Control Facility” on page 66
- ▶ 7.3, “Controlling access to the DFSMS cloud features” on page 69
- ▶ 7.4, “Creating a DFSMS cloud network connection by using ISMF” on page 70
- ▶ 7.5, “Enabling TCT compression with a TS7700” on page 77
- ▶ 7.6, “Using TS7700 object policies” on page 78

7.1 DFSMS connections to cloud

As discussed in Chapter 3, “Transparent Cloud Tiering” on page 19, DFSMS and HSM need a connection to the cloud object storage to store and retrieve metadata objects, and to manage the migrated data (list and delete objects).

Depending on the cloud target type, you set up the DFSMS cloud network connection in one of two ways:

- ▶ For cloud target types that use the S3 API (AWS, IBM Cloud Object Storage, IBM Cloud Object Storage services, TS7700), the DS8000 acts as cloud proxy. It relays the cloud requests from DFSMS to the actual cloud storage. You connect DFSMS to the DS8000 Hardware Management Console (HMC). DFSMS posts cloud requests through the DS8000 Representational State Transfer (REST) API. The HMC passes the requests on to one of the DS8000 internal servers, which then performs it on the cloud target.
- ▶ For target types that use the SWIFT API (Swift and Swift Keystone), you define the cloud object storage directly to DFSMS. The DS8000 HMC is not required as proxy.

7.2 Adding digital certificates to IBM Resource Access Control Facility

It is a best practice to set up a secure connection from z/OS (DFSMS) to cloud object storage. If you set up a DS8000 as a cloud proxy, a secure connection is mandatory. DFSMS uses the z/OS Web Enablement Toolkit (WETK) to communicate with cloud storage. Therefore, you use z/OS methods to set up a secure communication. The example commands that we provide in this section use the IBM Resource Access Control Facility (RACF). If you have another security solution in place, the commands are different, but you must use equivalent functions.

You set up secure communication in two steps:

1. Get the required security certificates from your cloud storage provider and upload them to z/OS.
2. Import the certificate or certificate chain into RACF.

Note: If you use a DS8000 HMC as cloud proxy, you have to ensure that the HMC certificates can be validated by the host. The default certificates that are installed during manufacturing are self signed. If you want to install CA signed certificates on your DS8000 HMCs, refer to Appendix C, “Replacing communication certificates in the IBM DS8900F” on page 145.

7.2.1 Uploading the certificate files to the z/OS host

To prepare to send the certificate files to the z/OS system, complete the following steps:

1. Determine which of the following certificate types that you will use:
 - Certificate authority
 - Self-signed site certificate

Which method to use might depend on your organization’s security policies and the security design of your object storage solution.

2. Request the certificate files from your organization's certificate administrator or export them directly from the cloud storage (or DS8000).

Note: If you connect DFSMS to a DS8000 HMC as cloud proxy and use the DS8000 default factory-provided certificate authority chain, you can follow the examples that are described in Appendix B, "Exporting the IBM DS8000 certificate chain" on page 139 to download the necessary certificates.

All certificate file formats that are supported by RACF can be used. For a complete list, see the documentation of the RACF **RACDCERT ADD** command at [RACDCERT ADD \(Add certificate\)](#).

Often, certificates are provided as *Privacy-Enhanced Mail* (PEM) encoded X.509 certificate files. If you use this format, ensure that all the certificates that are required for your certificate chain are available in separate PEM files.

After the certificate files are available, upload them to z/OS. The allocated z/OS data sets must meet the following conditions:

- ▶ Data sets containing certificates must have RECFM=VB.
- ▶ They must be cataloged.
- ▶ They cannot be a partitioned data set (PDS) or a PDS member.

Example 7-1 shows how you can upload a certificate file to a z/OS host by using an FTP client.

Example 7-1 Uploading certificates to the z/OS system by using FTP

```
ITS0User-MBP:ssl_certs itsouser$ ftp my.zoshost.com
Connected to my.zoshost.com.
220-FTP1 IBM FTP CS V2R3 at my.zoshost.com, 17:47:54 on 2017-08-06.
220 Connection will close if idle for more than 5 minutes.
Name (my.zoshost.com:workstation_user): ibmuser
331 Send password .
Password:
230 IBMUSER is logged on. Working directory is "IBMUSER.".
Remote system type is MVS.
ftp> site RECFM=VB
200 SITE command was accepted
ftp> put ds8kcaroot.pem DS8KROOT.PEM
local: ds8kcaroot.pem remote: DS8KROOT.PEM
229 Entering Extended Passive Mode (|||1037|)
125 Storing data set IBMUSER.DS8KROOT.PEM
100% |*****| 1434 606.22 KiB/s --:-- ETA
250 Transfer completed successfully.
1434 bytes sent in 00:00 (11.65 KiB/s)
ftp>
```

Note: If you use plain (unencrypted) FTP to upload certificate files, make sure that the transfer type of the FTP client is set to ASCII.

Inspect the content of the data sets you uploaded to make sure that they were transferred correctly.

7.2.2 Importing the certificates to RACF

After sending the certificate files to the z/OS system, you must add them to RACF. You accomplish this task by using the **RACDCERT** RACF command. To run this command, there are a few other security requirements:

- ▶ The **RACDCERT** command must be authorized under the AUTHCMD list in the IKJTS0xx parmlib member.
- ▶ The user ID that issues the **RACDCERT** command also must be authorized on RACF to do so.

You can check the details of the RACF authorization requirements for the **RACDCERT** command at [Controlling the use of the RACDCERT command](#).

Adding external CA certificates to RACF

Example 7-2 shows the syntax of the **RACDCERT** command that you use to add a certificate authority certificate that is contained in a data set to RACF.

Example 7-2 Adding a CA certificate in a data set to RACF

```
RACDCERT CERTAUTH ADD(<certificate data set>) WITHLABEL('<certificate label>')  
TRUST
```

Your certificate chain contains a CA root certificate and, optionally, one or more intermediate certificates. Make sure to import all certificates that are required to fulfill the certificate chain. After adding the certificates, you must refresh the RACF DIGTCERT class for the new configuration to take effect, as shown in Example 7-3.

Example 7-3 Refreshing the DIGTCERT class

```
SETROPTS RACLIST (DIGTCERT) REFRESH
```

Example 7-4 shows the sequence of commands to import the DS8000 factory-provided certificate authority root certificate into RACF.

Example 7-4 Adding root CA and intermediate CA certificates on RACF

```
RACDCERT CERTAUTH ADD('IBMUSER.DS8KROOT.PEM') WITHLABEL('DS8000 LAH80 CA root') TRUST  
SETROPTS RACLIST (DIGTCERT) REFRESH
```

Use the name of data set that contains the uploaded certificate in the **ADD** option and chose the label in the **WITHLABEL** option according to your organization's naming conventions. Make sure that you enclose the name and the label in single quotation marks. In our example, we add the **TRUST** option because we import a self-signed certificate. You do not need the **TRUST** option if your z/OS system can verify the certificate authority that signed the certificate that you imported.

Adding a self-signed site certificate to RACF

To add a self-signed site certificate data set to RACF, use the **SITE** option of the **RACDCERT** RACF command, as shown in Example 7-5.

Example 7-5 Adding a self-signed site certificate on RACF

```
RACDCERT SITE ADD('IBMUSER.SITECERT.PEM') WITHLABEL('self signed site cert') TRUST  
SETROPTS RACLIST (DIGTCERT) REFRESH
```

For an explanation of the other options, see “Adding external CA certificates to RACF”.

Verifying certificates

You can verify the correct import of your certificates by using the **RACDCERT LIST** command, as shown in Example 7-6.

Example 7-6 Listing a certificate

```
RACDCERT LIST(LABEL('DS8000 LAH80 CA root')) CERTAUTH
```

The label that you provide in the **LABEL** option must exactly match the label of the imported certificate. Other options of the **RACDCERT LIST** command are available to filter for specific certificates. For more information, see [RACDCERT LIST \(List certificate\)](#).

7.3 Controlling access to the DFSMS cloud features

There are RACF facility class profiles that are available to protect and control which users are allowed to use the **DUMP** and **RESTORE** commands, along with **CLOUD**, **CONTAINER**, or **OBJECTPREFIX** keywords. Only users with **READ** access to these profiles can use these commands.

Note: If the profiles are not defined, any user who knows the cloud credentials can use DFSMSdss to store data and retrieve data from a cloud.

7.3.1 Controlling access to DFSMSdss

To prohibit direct use of DFSMSdss, you can define System Authorization Facility (SAF) resources to control access to the **CLOUD** keyword for the DFSMSdss **DUMP** and **RESTORE** commands. Typically, the following FACILITY class profiles are defined with a universal access of **NONE**:

- ▶ STGADMIN.ADR.DUMP.CLOUD applies to a logical dump.
- ▶ STGADMIN.ADR.RESTORE.CLOUD applies to a logical restore.

Example 7-7 shows sample commands that can be used to define these FACILITY class profiles on RACF.

Example 7-7 Defining SAF resources to control access to the CLOUD keyword

```
RDEFINE FACILITY STGADMIN.ADR.DUMP.CLOUD UACC(NONE)  
RDEFINE FACILITY STGADMIN.ADR.RESTORE.CLOUD UACC(NONE)  
SETROPTS RACLIST(FACILITY) REFRESH
```

7.3.2 Controlling access to DFSMSShsm

To control the access to DFSMSShsm, you must set up the following tasks:

- ▶ Defining DFSMSShsm to z/OS UNIX System Services
- ▶ Defining SAF resources to control access to the cloud

Defining DFSMSHsm to z/OS UNIX System Services

Define DFSMSHsm to z/OS UNIX System Services as a super user. Also, the DFSMSHsm RACF user ID must have a default RACF group that has an OMVS segment with a group ID (GID). This user ID must also have an OMVS segment with the following parameters:

```
UID(0) HOME('/')
```

Defining SAF resources to control access to the cloud

The commands in Example 7-8 define SAF resources that control access to the **CLOUD** keyword on the **HMIGRATE** user command in DFSMSHsm, and grant READ access to the STGADMIN.ARC.ENDUSER.HMIGRATE.CLOUD FACILITY class profile.

Example 7-8 Granting READ access to DFSMSHsm to the migrate task

```
RDEFINE FACILITY STGADMIN.ARC.ENDUSER.HMIGRATE.CLOUD UACC(NONE)
PERMIT STGADMIN.ARC.ENDUSER.HMIGRATE.CLOUD CLASS(FACILITY) ID(HSMUSER) -
ACCESS(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

7.4 Creating a DFSMS cloud network connection by using ISMF

You can define cloud object storage targets for DFSMS in a panel of the ISMF. The new network connection construct allows you to define the parameters that are necessary to connect to the cloud targets. The Network Connection option in the ISMF menu is accessible only when you have access to the administrator mode on ISMF panels. For more information, see [Defining network connections](#).

Example 7-9 shows the location of the Network Connection option on the main ISMF menu. Depending on your terminal configuration, you might need to scroll down to see it.

Example 7-9 Network Connection option on the ISMF panel

```
ISMF PRIMARY OPTION MENU - z/OS DFSMS V2 R5
Selection or Command ==>

More: -
2 Volume - Perform Functions Against Volumes
3 Management Class - Specify Data Set Backup and Migration Criteria
4 Data Class - Specify Data Set Allocation Parameters
5 Storage Class - Specify Data Set Performance and Availability
6 Storage Group - Specify Volume Names and Free Space Thresholds
7 Automatic Class Selection - Specify ACS Routines and Test Criteria
8 Control Data Set - Specify System Names and Default Criteria
9 Aggregate Group - Specify Data Set Recovery Parameters
10 Library Management - Specify Library and Drive Configurations
11 Enhanced ACS Management - Perform Enhanced Test/Configuration Management
C Data Collection - Process Data Collection Function
G Report Generation - Create Storage Management Reports
L List - Perform Functions Against Saved ISMF Lists
P Copy Pool - Specify Pool Storage Groups for Copies
R Removable Media Manager - Perform Functions Against Removable Media
S Network Connection - Specify Network Connection Attributes
```

Select the S option (Network Connection) to open the Network Connection Application Selection panel, which is shown in Example 7-10.

It gives you the options to list, display, define, or alter network connection definitions. To define a new cloud target, we provide the name (IBMCOS) and select option 3.

Example 7-10 Defining a cloud network connection

```
                                NETWORK CONNECTION APPLICATION SELECTION

To perform Network Connection Operations, Specify:
CDS Name . . . . . 'SMS.DFSMS.SCDS'
                                (1 to 44 character data set name or 'Active' )
Network Connection Name IBMCOS                                (Name of Network
                                Connection for Cloud. For Network Connection
                                List, fully or partially specified or * for all)

Select one of the following options:
3 1. List    - Generate a list of Network Connections
    2. Display - Display a Network Connection
    3. Define - Define a Network Connection
    4. Alter   - Alter a Network Connection

If List Option is chosen,
    Enter "/" to select option      Respecify View Criteria
                                    Respecify Sort Criteria

Command ==>
```

Note: You must use the same network connection definition name in DFSMS as for the DS8000 cloud network connection that you created according to Chapter 6, “Configuring the IBM DS8000 for Transparent Cloud Tiering” on page 49.

7.4.1 Defining a cloud network connection for Amazon S3, IBM Cloud Object Storage, or TS7700 cloud targets

For all cloud target types that use a different API than SWIFT, DFSMS uses a DS8000 as proxy to access the cloud object storage. Therefore, you define the DS8000 HMC IP address and credentials as cloud definition. DFSMS and the DS8000 cloud proxy function use the Swift API for communication. All object requests that originate from DFSMS are routed through the DS8000 to the “real” cloud target.

Example 7-11 and Example 7-12 on page 72 show the entries that you must make for a DS8000 cloud proxy definition.

Example 7-11 First network connection definition panel for a DS8000 cloud proxy definition

```
                                NETWORK CONNECTION DEFINE                                Page 1 of 2

SCDS Name . . . . . : SMS.DFSMS.SCDS
Network Connection Name : IBMCOS

To DEFINE Network Connection, Specify:

Description   IBM Redbooks DEMO IBMCOS CLOUD
```

Provider . . **SWIFT** (SWIFT, SWIFT-KEYSTONE or TAPE-OBJECT)

Identity . . **cloudproxy**

Command ==>

Specify the following fields in the first network connection definition panel:

- ▶ Provider: SWIFT (specify TAPE-OBJECT when connecting to a TS7700 Object Store).
- ▶ Identity (credentials). The name of a user that is defined to the DS8000 HMC that you want to use for the proxy operations.

Example 7-12 Second cloud definition panel for a DS8000 cloud proxy definition

NETWORK CONNECTION DEFINE

Page 2 of 2

SCDS Name : **SMS.DFSMS.SCDS**

Network Connection Name : **IBMCOS**

To DEFINE Network Connection, Specify:

Endpoint **https://x.xxx.xxx.xxx**

Port Number . . **8452** (0 to 65535)

SSL Version . . **TLSV12** (TLSV12, TLSV11, TLSV1, SSLV3 or blank)

SSL Key ***AUTH*/***

Command ==>

In the second network connection definition panel, enter the remaining parameters:

- ▶ Endpoint: the Uniform Resource Identifier (URI) of the DS8000 HMC that acts as the proxy server. HMC connections require HTTPS communication.
- ▶ Port: The remote port number to which to connect instead of the default HTTP or HTTPS port. At the time of writing, the only port that is supported by the DS8000 is 8452.
- ▶ Secure Sockets Layer (SSL) version: The lowest SSL version acceptable to use when making HTTP requests. The maximum length is 8 characters, and valid values are TLSV12, TLSV11, TLSV1, SSLV3, or blank.
- ▶ SSL key: The name of the keystore to be used (required when SSL version is not blank). The value can be a SAF key ring name, in the form of user ID or key ring, or a PKCS #11 token in the form of **TOKEN*/token_name*. If you plan to use CA certificates, you must specify **AUTH*/** (as in our example). If you plan to use a site certificate, you must specify **SITE*/**.

For more information about setting up a DS8000 as a cloud object proxy, see [z/OS Version 2 Release 3 DFSMSdfp Storage Administration](#).

7.4.2 Defining a cloud network connection for a Swift cloud object storage target

With cloud object storage solutions that use the Swift API, DFSMS communicates directly. Again, the cloud definition process consists of two panels. The first panel is shown in Example 7-13.

Example 7-13 First cloud definition panel for Swift

```
                                NETWORK CONNECTION DEFINE                                Page 1 of 2

SCDS Name . . . . . : SMS.DFSMS.SCDS
Network Connection Name : ITSOSWIFT

To DEFINE Network Connection, Specify:

Description   IBM Redbooks SWFIT CLOUD

Provider    . . SWIFT-KEYSTONE   (SWIFT, SWIFT-KEYSTONE or TAPE-OBJECT)

Identity    . . test:tester

Command ==>
```

The following fields are available for definition in the first network connection definition panel:

- ▶ **Description:** A brief description of the cloud you are defining. You can include some information about the service provider, service expiration date, or availability. Up to 120 characters can be used in description.
- ▶ **Provider:** Specifies the type of cloud provider. At the time of writing, only SWIFT and SWIFT-KEYSTONE options are available.
- ▶ **Identity:** Specifies the credentials that are used when authenticating with the cloud. For SWIFT cloud targets, you usually have a user ID and a tenant ID. Specify both, separated by a colon.

Move to the second definition panel by using the **DOWN** command. The second panel is shown in Example 7-14.

Example 7-14 Second cloud definition panel for SWIFT

```
                                NETWORK CONNECTION DEFINE                                Page 2 of 2

SCDS Name . . . . . : SMS.DFSMS.SCDS
Network Connection Name : ITSOSWIFT

To DEFINE Network Connection, Specify:

Endpoint . . . . https://swift.demo.ibm.com/auth/v2

Port Number . . 5000      (0 to 65535)
SSL Version . . TLSV12    (TLSV12, TLSV11, TLSV1, SSLV3 or blank)
```

SSL Key *AUTH*/*

Command ===>

The following fields are available for definition in the second network connection definition panel:

- ▶ Endpoint: Identifies the URI that is used when authenticating with the cloud. For Swift cloud targets, the SWIFT authentication version number must be added to the URL.
- ▶ Port Number: Specifies the remote port number to which to connect. Possible values are 0 - 65535.
- ▶ SSL Version: Defines the lowest acceptable SSL version that is used when connecting to the cloud.
- ▶ SSL Key: The name of the keystore to be used (required when SSL version is not blank. The value can be a SAF key ring name in the form of user ID or key ring, or a PKCS #11 token in the form of *TOKEN*/*token_name*. If you use CA certificates, you must specify *AUTH*/*. If you use a self-signed certificate, you must specify *SITE*/*.

7.4.3 Using NaviQuest to manage DFSMS network connections

NaviQuest is an alternative to ISMF. It allows you to manage DFSMS in a scripted way. With z/OS V2R4, you can also use NaviQuest to list, display, define, and alter cloud network connections. We show a sample NaviQuest job definition in Example 7-15.

Example 7-15 NaviQuest step to define a cloud network connection

```
//DEFINE EXEC ACBJBAOB, TABL2=IBMUSER.TEST.ISPTABL
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
  PROFILE PREFIX(IBMUSER)
  ISPSTART CMD(ACBQBAC3) NEWAPPL(DGTS) BDISPMAX(999999)
//SYSIN DD *
DEFINE
SCDS('SMS.DFSMS.SCDS')
NCNAME(IBMOS)
PROVIDER(SWIFT)
IDENTITY(cloudproxy)
ENDPOINT(https://x.xxx.xxx.xxx)
PORTNUM(8452)
SSLVER(TLSV12)
SSLKEY(*AUTH*/*)
UPDHLVLSCDS(Y)
DESCR(IBM Redbooks DEMO IBMCOS CLOUD)
```

For more information, see the NaviQuest sections of *z/OS DFSMSdfp Storage Administration*, found at:

<https://www.ibm.com/docs/en/zos/2.4.0?topic=administration-using-naviquest>

7.4.4 Activating the Storage Management Subsystem configuration

After you complete and save the cloud network configuration, you must activate the Source Control Data Set (SCDS) that contains the new definition.

Note: Make sure you activate the correct Control Data Set (CDS). CDS activation is a system-wide operation and impacts the way DFSMS works.

Activating the new configuration does not automatically connect z/OS to the cloud. Each application that is trying to access the cloud is required to provide the password to store and retrieve data. The DS8000 must also be configured to access the cloud before the connection can be established:

1. To activate the SCDS, go to main ISMF menu and select option 8 Control Data Set, as shown in Example 7-16.

Example 7-16 Selecting the Control Data Set option

```
ISMF PRIMARY OPTION MENU - z/OS DFSMS V2 R2

0 ISMF Profile - Specify ISMF User Profile
1 Data Set - Perform Functions Against Data Sets
2 Volume - Perform Functions Against Volumes
3 Management Class - Specify Data Set Backup and Migration Criteria
4 Data Class - Specify Data Set Allocation Parameters
5 Storage Class - Specify Data Set Performance and Availability
6 Storage Group - Specify Volume Names and Free Space Thresholds
7 Automatic Class Selection - Specify ACS Routines and Test Criteria
8 Control Data Set - Specify System Names and Default Criteria
9 Aggregate Group - Specify Data Set Recovery Parameters
10 Library Management - Specify Library and Drive Configurations
11 Enhanced ACS Management - Perform Enhanced Test/Configuration Management
C Data Collection - Process Data Collection Function
G Report Generation - Create Storage Management Reports
L List - Perform Functions Against Saved ISMF Lists
Selection or Command ==>
```

- The CDS Application Selection panel opens. You should validate your SCDS by using option 4 Validate the SCDS before you make it the active CDS, as shown in Example 7-17.

Example 7-17 Validating the CDS

CDS APPLICATION SELECTION

To Perform Control Data Set Operations, Specify:

CDS Name . . 'SMS.DFSMS.SCDS'
(1 to 44 Character Data Set Name or 'Active')

Select one of the following Options:

1. Display - Display the Base Configuration
2. Define - Define the Base Configuration
3. Alter - Alter the Base Configuration
- 4. Validate** - Validate the SCDS
5. Activate - Activate the CDS
6. Cache Display - Display CF Cache Structure Names for all CF Cache Sets
7. Cache Update - Define/Alter/Delete CF Cache Sets
8. Lock Display - Display CF Lock Structure Names for all CF Lock Sets
9. Lock Update - Define/Alter/Delete CF Lock Sets

If CACHE Display is chosen, Enter CF Cache Set Name . . *

If LOCK Display is chosen, Enter CF Lock Set Name . . . *

Command ==>

- After validating the CDS, select option 5. Activate the CDS to activate the configuration.

Example 7-18 Activating the CDS

CDS APPLICATION SELECTION

To Perform Control Data Set Operations, Specify:

CDS Name . . 'SMS.DFSMS.SCDS'
(1 to 44 Character Data Set Name or 'Active')

Select one of the following Options:

1. Display - Display the Base Configuration
2. Define - Define the Base Configuration
3. Alter - Alter the Base Configuration
4. Validate - Validate the SCDS
- 5. Activate** - Activate the CDS
6. Cache Display - Display CF Cache Structure Names for all CF Cache Sets
7. Cache Update - Define/Alter/Delete CF Cache Sets
8. Lock Display - Display CF Lock Structure Names for all CF Lock Sets
9. Lock Update - Define/Alter/Delete CF Lock Sets

If CACHE Display is chosen, Enter CF Cache Set Name . . *

If LOCK Display is chosen, Enter CF Lock Set Name . . . *

Command ==>

- Place a forward-slash in the Confirm Activate Request panel.

An alternative way of activating the CDS is by using the **SETSMS SCDS(dsname)** command.

DFSMSHsm also needs permission to list the key rings to which it has access. It can be allowed by granting READ access to the user ID that is used by the DFSMSHsm started task to the IRR.DIGTCERT.LIST and IRR.DIGTCERT.LISTRING profiles, as shown in the Example 7-19.

Example 7-19 Setting access to DSHSM procedure on RACF

```
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID(DFHSM) ACCESS(READ)
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(DFHSM) ACCESS(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

7.5 Enabling TCT compression with a TS7700

TCT supports compression of data that is moved to the cloud when you use a TS7700 as cloud object target. TCT compression is not enabled by default. You can control it by configuring the z/OS Device Manager.

Note: To see whether your environment has all the requirements for TCT compression, see Chapter 4, “Requirements” on page 37.

Enable TCT compression in z/OS during run time with the **MODIFY DEVMAN** command, as shown in Example 7-20.

Example 7-20 Enabling TCT compression during run time

```
MODIFY DEVMAN,ENABLE(TCTCOMPRESSION)
DM00012I DEVICE MANAGER TCT COMPRESSION ENABLED
```

With the **DISABLE** keyword, you can disable TCT compression again. If you need to check whether TCT compression is enabled or not, use the **DISPLAY DEVSUP** command as in Example 7-21.

Example 7-21 Checking whether TCT compression is enabled

```
DISPLAY DEVSUP
IEA253I DISPLAY DEVSUP Start of Report 761
...
... ENABLE(TCTCOMPRESSION)
...
DISPLAY DEVSUP End of Report
```

Modifying DEVMAN during run time is a non-persistent setting. You would revert to the default after an initial program load (IPL) of the operating system. To enable TCT compression at IPL, you can add a setting to your **DEVSUPxx** member in your **PARMLIB**, as described in Example 7-22.

Example 7-22 Enabling TCT compression at IPL in DEVSUPxx

```
ENABLE(TCTCOMPRESSION)
```

7.6 Using TS7700 object policies

Using the TS7700 as object storage, you can define policies for stored objects that specify where in the TS7700 grid objects are stored and how they are replicated.

Note: You must meet several hardware, microcode, and software requirements to be able to implement and use TS7700 object policies. For more information, see Chapter 4, “Requirements” on page 37.

To migrate or dump data sets according to these policies, you need a DS8000 cloud connection and a DFSMS cloud network connection for each of them. These connections must follow certain rules and naming conventions, as described in the following steps:

1. Enable object storage support in the TS7700 systems, according to *IBM TS7700 Series DS8000 Object Store User's Guide Version 2.0*, REDP-5583.

Note: At the time of writing, this publication is not updated yet to cover the changes that were introduced with TS7700 Release 5.22 and Feature Code (FC) 5283.

2. Create TS7700 object policies to define all the replication and storage variations you plan to use. For more information and detailed instructions regarding this step and step 3, see *IBM TS7700 Release 5.2.2 Guide*, SG24-8464.
3. Create a TS7700 object store for each of the object policies. The names of these object stores are later referred to in the DS8000 cloud definition names and DFSMS cloud network connection names. Avoid special characters and spaces. The case does not matter.
4. Define DS8000 cloud connections according to Chapter 6, “Configuring the IBM DS8000 for Transparent Cloud Tiering” on page 49. You need one cloud connection for each TS7700 object store that you want to use. The name of a DS8000 cloud connection must be the same as that of the TS7700 object store it refers to.
5. Define DFSMS cloud network connections according to 7.4, “Creating a DFSMS cloud network connection by using ISMF” on page 70. You need one DFSMS cloud network connection for each DS8000 cloud connection (and therefore each TS7700 object store) you want to use. The name of a DFSMS cloud network connection must be the same as that of the DS8000 cloud connection (and therefore the TS7700 Object Store) it refers to.

You can now use the DFSMS cloud network connection name in your DFSMSdss or HSM commands or class definitions to specify the rules for object placement and replication in the TS7700 Grid.



Managing cloud credentials

In this chapter, we explain methods to manage the credentials DFSMSdss and Hierarchical Storage Manager (HSM) need to access cloud object storage.

In the Data Facility Storage Management Subsystem (DFSMS) cloud network definition, we specified the endpoint and credentials, but without providing a password. DFSMSdss or HSM need this password whenever they are called to move data to and from cloud object storage with Transparent Cloud Tiering (TCT), or to access the cloud storage for any other reason.

The passwords can be managed jointly for both DFSMSdss and HSM by using the Cloud Data Access (CDA) framework. In this chapter, we explain how to set up CDA and how to enter, store, and manage passwords securely.

For HSM, there is a second method to store cloud storage credentials, which we describe as the *legacy* method in 8.2, “Legacy method to manage cloud passwords for HSM” on page 86.

If you use Swift cloud object storage, you need the full cloud credentials to set up the DFSMS cloud network connection. For all other types of cloud targets, you need a DS8000 user ID and password to set up the connection to the cloud proxy.

Note: In both cases, anybody with access to the cloud credentials has full access to all storage objects in the cloud, including the capability to update, move, or delete data.

It is a best practice to manage the cloud credentials centrally, as described here, and not by using clear text credentials in DFSMSdss job definitions.

This chapter includes the following topics:

- ▶ 8.1, “Managing credentials by using Cloud Data Access” on page 80
- ▶ 8.2, “Legacy method to manage cloud passwords for HSM” on page 86

Note: At the time of writing, the PTFs for APAR OA60278, which support HSM Full Volume Dump and **CDACREDS**, are not available because a problem was discovered. If you intend to use these functions, check for the availability of APAR OA64130, which will fix the issue.

8.1 Managing credentials by using Cloud Data Access

The CDA framework provides a method to store and manage the required cloud credentials. When DFSMSDss or HSM must access the cloud object storage, they can request and retrieve the credentials from CDA. CDA uses the *Integrated Cryptographic Service Facility* (ICSF) to store the credentials securely, encrypted, and protected according to your organization's rules.

The **CDACREDS** keyword in DFSMSDss or HSM commands indicates that they should use CDA APIs to retrieve the password at run time.

Note: It is a best practice to set up and use CDA. Although it is possible to provide the cloud password in clear text with each DFSMSDss job definition, doing so is a highly insecure method.

There are some prerequisites for using CDA:

- ▶ The ICSF server must be configured and running.
- ▶ The user running the DFSMSDss or HSM commands or jobs needs an OMVS segment.
- ▶ Some IBM Resource Access Control Facility (RACF) definitions are required.

CDA uses an IBM zSystems Crypto Express feature that is configured as an IBM Common Cryptographic Architecture (CCA) co-processor to wrap the key that is used to encrypt the cloud credentials with the Crypto Express master key. This approach protects the CDA key from unauthorized access.

Note: If you do not have a Crypto Express feature that is available in your environment, you can store the CDA key locally and unwrapped. This option is *not considered safe*. Use it only for testing purposes.

The CDA framework uses an encrypted data set to store the cloud credentials for a user who is going to run DFSMSDss jobs. You can store the credentials for one or more cloud providers (targets).

A CDA entry contains the z/OS user ID that runs the DFSMSDss or HSM commands or jobs, and the cloud object storage user ID and password. These values are packaged, encrypted, and stored. They can be retrieved by DFSMSDss or HSM when the **CDACREDS** option is used in the job definition or command (see Part 3, "Operation and usage" on page 89).

In the following sections, we explain the CDA setup and configuration with an example for a single user (DSSADMIN) and some defined cloud storage targets.

8.1.1 Preparing the Cloud Data Access configuration

Using the z/OS UNIX Systems Services, complete the following steps:

1. Create a directory that is called `gdk` in the user's home directory of `/u/dssadmin`, and copy two starter files from `/usr/lpp/dfsms/gdk` to the `gdk` directory, as shown in Example 8-1.

Example 8-1 Creating the configuration file

```
mkdir /u/dssadmin/gdk
cp /usr/lpp/dfsms/gdk/samples/gdkconfig.json /u/dssadmin/gdk
cp /usr/lpp/dfsms/gdk/samples/gdkkeyf.json /u/dssadmin/gdk
```

2. Create a `providers` directory and empty files that correspond to the cloud network connection names that are defined in your SMS environment, as shown in Example 8-2.

Example 8-2 Creating cloud network connection provider files

```
mkdir /u/dssadmin/gdk/providers
touch /u/dssadmin/gdk/providers/TS7700LOCAL.json
touch /u/dssadmin/gdk/providers/TS7700SYNC.json
touch /u/dssadmin/gdk/providers/TS7700DEFERRED.json
```

3. Create a separate touch file for each DFSMS cloud network connection for which you want to manage credentials.

8.1.2 Configuring the CSFKEYS general resource class

The CSFKEYS RACF (or equivalent) general resource class must be configured so that CDA can use the encryption services. Complete the following steps:

1. The CSFKEYS general resource class must be active and RACLISTed.
2. The ICSF segment of the CSFKEYS class profile CSF-PROTECTED-KEY-TOKEN (or its generic equivalent) must contain SYMCPACFWRAP (YES).
3. The user who is going to run the DFSMSdss or HSM commands or jobs must have READ access to the CSF-PROTECTED-KEY-TOKEN profile (or its generic equivalent).
4. This user must also have READ access to the new CSFKEYS profile for resources beginning with 'GDK'.
5. The security administrator who enters the cloud provider credentials must have UPDATE access to the new CSFKEYS profile for all resources beginning with GDK.<UserID>, where *UserID* is substituted with the ID of the user who is going to run the DFSMSdss or HSM commands.

8.1.3 Entering and storing the cloud provider credentials

Note: The CDACREDS data entry application uses ISPF panels. Make sure the panel library SYS1.DFQPLIB is part of the ISPLLIB concatenation or that the following members of SYS1.DFQPLIB are added to an ISPLLIB library: GDKAPPOP, GDKAUTHK, GDKAUTHL, GDKAUTHP, GDKMAINP, GDKOBJAC, GDKOBJAL, and GDKOPPOP.

Consider creating a RACF (or equivalent) profile to make sure that only authorized users have access to those members.

To enter and store the cloud provider credentials, complete the following steps:

1. Launch the **GDKAUTHP** exec (found in `SYS1.SAXREXEC`) to start the z/OS Cloud Data Access Authorization Utility by running '`SYS1.SAXREXEC(GDKAUTHP)`', for example, in the ISPF command shell, as shown in Example 8-3.

Example 8-3 TSO Command execution

```
ISPF Command Shell
Enter TSO or Workstation commands below:

===> EX 'SYS1.SAXREXEC(GDKAUTHP)'
```

Place cursor on choice and press enter to Retrieve command

```
=>
=>
```

2. In the z/OS Cloud Data Access Authorization Utility, you see the list of your defined Cloud Providers, as shown in Example 8-4. The *Cloud Provider* names that are listed correspond to the touch files that you created earlier.

Example 8-4 Selecting the cloud provider for credential definition

```
z/OS Cloud Data Access Authorization Utility
Option ===> 0
  L Display Resource Authorization List
  0 Open Credential Entry Panel
```

```
Select Cloud Provider
  2 1.TS7700LOCAL
    2.TS7700SYNC
    3.TS7700DEFERRED
```

```
Encryption Parameters
  Provider . . .
  UserID . . . . DSSADMIN
  Resource . . . /
```

Choose Cloud Provider, User ID, and optional Resource. Enter "0" on the Option to enter the Key and Secret Key.

Note: If you use the z/OS Cloud Data Access Authorization Utility for the first time, no cloud provider and UserID is displayed. You must enter the user ID for which you want to store the credentials in the *UserID* field and press Enter. Then, the cloud providers that are specified in the touch files are displayed.

3. Type 0 on the Option line and select the Cloud Provider by specifying its number in the Cloud Provider data entry field, as shown in Example 8-4. Then, press Enter to open the credentials entry panel.

4. Enter the credential in the Key and Secret Key fields, as shown in Example 8-5.

Example 8-5 Entering the cloud credentials

```
z/OS Cloud Data Access Authorization Utility
Option ==> S
  S Save Resource Authorization          C Clear Secret Key Field
                                         (for hidden input)

Encryption Parameters
  Provider . . . TS7700SYNC
  KeyLabel . . . GDK.IBMUSER.TS7700SYNC
  Keystore . . . /u/dssadmin/gdk/gdkkeyf.json
  Resource . . . /

Authorization Parameters
  Key . . . . . username
  Secret Key . . *****
```

Enter the Key and Secret Key used to access the specified Cloud Provider.

Type **S** on the Option line, and press enter. CDA encrypts the username and password with a key label that is created and stored in ICSF. If this task is successful, you see the message `Key saved/updated...` in the upper right.

Repeat steps 3 on page 82 and 4 with each Cloud Provider entry for which you want to store the credentials.

Note: The keywords `Key` and `Secret Key` that are used here are based on the Amazon S3 protocol and can be misleading. If you use the DS8000 as a cloud proxy, you must enter your DS8000 credentials:

- ▶ For `Key` enter your DS8000 username
- ▶ For `Secret Key` your DS8000 password

If you are using a SWIFT cloud storage target, consider the SWIFT. For more information, see 8.2, “Legacy method to manage cloud passwords for HSM” on page 86.

8.1.4 Configuring CDA to run without Crypto Express adapters

By default, CDA requires a Crypto Express feature that is installed and configured on your system. If you do not have Crypto Express and still want to use CDA to store cloud credentials, you can use the `allow-no-CEX` option to avoid the requirement.

Note: Without a Crypto Express card, the encryption key that is used to encrypt the cloud credentials cannot be wrapped by the master key in the Crypto Express card before being stored in the ICSF Cryptographic Key Data Set (CKDS). When no Crypto Express Card is installed and configured, the encryption key for the Cloud Credentials is stored in the clear in the ICSF CKDS. This approach is considered *insecure*. Use this option only for test purposes.

To override the requirement for a Crypto Express feature, add an option to the configuration file of the DFSMS user's CDA framework. The file name and location is <user-home>/gdk/gdkconfig.json. In our example, it is /u/dssadmin/gdk/gdkconfig.json. The option that you use consists of a key-value pair, which is shown in bold in Example 8-6.

Example 8-6 Contents of gdkconfig.json

```
{
  "allow-no-CEX": true,
  "log-level": "ERROR",
  "web-toolkit-logging": false,
  "translation": true
}
```

If the **"allow-no-CEX": true** option is used and a Crypto Express card is installed later, CDA continues to use the clear key that is stored in the CKDS to decrypt the cloud credentials. To secure the CDA key, identify the cloud credentials that were stored without the Crypto Express card and resave them.

To find the unsecured entries, examine the gdkkeyf.json files for each user. Look for provider entries that contain the key-value pair "NoCEX": "true". A sample gdkkeyf.json file with this option set is shown in Example 8-7.

Example 8-7 Contents of gdkkeyf.json with the "NoCEX" option

```
{
  "Credentials": [
    {
      "MVSUserID": "dssadmin",
      "cloud provider": {
        "TS7700SYNC": [
          {
            "keylabelID": "A00000",
            "name": "/",
            "key": "key",
            "secretkey": "secretkey",
            "NoCEX": "true"
          }
        ]
      }
    }
  ]
}
```

A subsequent call to save the credentials uses the Crypto Express feature regardless of the gdkconfig.json settings. You do not have to change the gdkkeyf.json file.

8.1.5 Deleting the cloud provider credentials

To delete the cloud provider credentials, complete the following steps:

1. Run the **GDKAUTHP** exec to launch the z/OS Cloud Data Access Authorization Utility, as described in 8.1.1, "Preparing the Cloud Data Access configuration" on page 81.
2. Under the Select Cloud Provider heading, enter the number of the cloud provider for which you want to process the credentials and press Enter. This action populates the Encryption Parameters section of the panel.

3. Enter the option L and press Enter to open the credentials panel for the selected provider.
4. Enter a “/” next to the key to be removed, as shown in Example 8-8, and press Enter.

Example 8-8 Selecting the specific provider

```

/u/dssadmin/gdk/gdkkeyf.json.                               Row 1 to 1 of 1
Command ==>                                               Scroll ==> CSR

Command - Enter "/" to select action
Provider

-----
/ /                                                         TS7700SYNC
***** Bottom of data *****

```

5. A panel opens and shows a list of actions for the selected provider. At the time of writing, the only option is 1. Delete. Type a 1 in the entry field, as shown in Example 8-9, and press Enter to confirm the deletion.

Example 8-9 Deleting credentials

```

*-----*
|                                     |
|           Keystore List Action Enter required field |
| Resource: /                               |
| Keystore Action                             |
| 1 1. Delete                                |
|                                             |
| Select a choice and press ENTER to process data set action. |
|-----*

```

8.1.6 Troubleshooting

If you encounter errors when implementing the CDA credentials, check and confirm the following items:

- ▶ The library SYS1.DFQPLIB is in your ISPLLIB concatenation. If not, you get a Panel not found message when invoking panel GDKAUTHP:
ISPP100 Panel 'GDKAUTHP' error
- ▶ The ICSF Cryptographic Key Data Set (CKDS) must be in one of the two available variable length record formats.
- ▶ The ICSF segment of the CSFKEYS class profile CSF-PROTECTED-KEY-TOKEN (or its generic equivalent) must contain SYMPACFWRAP (YES) or you receive the following error message:
Encipher result: rc:8, rsn:0BFB

For more information, see [Enabling use of encrypted keys in callable services that exploit CPACF](#).

8.1.7 Using the CDA credentials

DFSMSdss and HSM can use the defined CDA credentials for tasks that require access to cloud object storage.

DFSMSdss

Specify the keyword **CDACREDS** in any DFSMSdss command or job that uses TCT. For examples, see Part 3, “Operation and usage” on page 89.

HSM

HSM is used to store the cloud object storage password itself in its Control Data Set (CDS). With CDA support for HSM, you do not need to specify the password directly to HSM anymore. Instead, you can instruct HSM to use the CDA framework for an object storage target. Use the HSM **SETSYS** command, as shown in Example 8-10.

Example 8-10 Instructing HSM to use the CDA framework for credentials

```
HSEND SETSYS CLOUD(NAME(TS7700SYNC) CDACREDS)
```

Specify the DFSMS cloud network connection name in the **NAME** option. The **CDACREDS** option tells HSM to get the credentials for this connection from CDA. You must issue a separate **SETSYS** command for each cloud network connection that you want to use with HSM.

8.2 Legacy method to manage cloud passwords for HSM

At the time of writing, there still is an alternative method to manage cloud credentials for HSM. You can store the cloud credential (cloud provider password, secret key, or the DS8000 password if the DS8000 Hardware Management Console (HMC) acts as cloud proxy) in the HSM CDS by using the HSM **SETSYS** command. This method has been available since the introduction of TCT, and we call it the *legacy* method. It has certain limitations.

Note: With this method to manage cloud credentials for HMS, you are limited to a maximum of five cloud network definitions for which you can store passwords. With the CDA method, you can manage credentials for up to 255 cloud network connections.

HSM can receive the cloud password and store it encrypted in its CDSs. To configure DFSMSshm to use cloud storage, use the HSM command **SETSYS CLOUD**. With this command, you can perform the following actions:

- ▶ Connect HSM to a new cloud definition.
- ▶ Refresh cloud credentials.
- ▶ Delete a cloud definition from DFSMSshm CDSs.

The **SETSYS CLOUD** command that is shown in Example 8-11 defines the cloud to the DFSMSshm, attempts to connect, and if it is successful, stores the cloud-related information into the Migration Control data set (MCD).

Example 8-11 Defining the cloud to DFSMSshm

```
HSEND SETSYS CLOUD(NAME(IBMCOS) CLOUDCREDENTIALS)
```

When this command is issued, a WTOR prompts you to supply the cloud password. The message identifier that is related to the WTOR is ARC1585A. Example 8-12 shows the WTOR waiting for a reply on the system log.

Note: DFSMSHsm activity is quiesced until the WTOR receives a reply.

Example 8-12 WTOR that is generated by the SETSYS CLOUD command

```
ARC0300I IBMUSER ISSUED===>SETSYS CLOUD(NAME(IBM COS) CCREDS)
*0029 ARC1585A ENTER PASSWORD FOR CLOUD IBM COS
R 29 SUPPRESSED
IEE600I REPLY TO 0029 IS;SUPPRESSED
ARC0100I SETSYS COMMAND COMPLETED
```

The cloud password can contain uppercase and lowercase characters. As a best practice, reply to this WTOR from the SDSF System Command Extension, as shown in Example 8-13. Replying from the SDSF SYSLOG forces a reply to uppercase, which can result in a failed authentication.

Example 8-13 Case-sensitive password

```
System Command Extension

===> REPLY 29, 'PaSsw0rd'
```

Note: You get the System Command Extension from SDSF by typing a “/” (slash) character in the CLI and pressing Enter. Do not forget the quotation marks around your password. Otherwise, it is converted to uppercase.

During the configuration, the connection to the cloud is tested. If it cannot be established, an error message is returned to the user with the information that is related to the connection error, as shown in Example 8-14.

Example 8-14 Failure to connect to the cloud message

```
ARC1581I UNEXPECTED HTTP STATUS 401 DURING A GET FOR URI
ARC1581I (CONT.) https://swift.demo.ibm.com/auth/v2 ERRTEXT HTTP/1.1
ARC1581I (CONT.) 401 Unauthorized
ARC0100I SETSYS COMMAND COMPLETED
***
```

This message indicates that some of the authentication information that is entered is wrong (it also might be expired).

You can use the **SETSYS CLOUD** command to refresh the cloud settings, change the password, or remove the cloud from DFSMSHsm control records. The following sample **SETSYS CLOUD** commands are supported:

- ▶ **SETSYS CLOUD(NAME(*xxxxx*) REMOVE):** Use this command to remove the cloud “*xxxxx*” from DFSMSHsm CDSs.
- ▶ **SETSYS CLOUD(NAME(*xxxxx*) REFRESH):** Use this command to refresh the cloud “*xxxxx*” credentials to a DFSMSHsm CDS, including the password that was used to connect to the cloud.
- ▶ **SETSYS CLOUD(NAME(*xxxxx*) CLOUDCREDENTIALS):** Use this command to create a WTOR requesting the cloud password. Use this option to update changed cloud credentials.

You can have up to five cloud definitions that are stored in a DFSMSHsm CDS. If you want to set up a new cloud definition and five cloud definitions already are configured, you must delete a definition before creating one. To identify the cloud connections that are configured to DFSMSHsm, you can use the command that is shown in Example 8-15 and search for your cloud connection names. The first name starts on byte x'45C'.

Example 8-15 Displaying cloud connections

```
HSEND FIXCDS S MHCR DISPLAY
...
+0440 00000000 00000000 00000000 00000000 00000000 00000000 00000000 E2F3D7D9
*
      IBMR*
+0460 D6E7E840 40404040 40404040 40404040 40404040 40404040 40400007 80000000
*EDBOOKS
      *
+0480 B70D8B65 BDBAF129 841BDF9F ABOACD01 4AD6B572 436A844E 9CBCE8C6 E8B3BDF5
*
      1      0      YFY 5*
+04A0 954F1E0E 266F6578 0EFDEF51 94584767 492957AF B0B93A62 1937EBF7 744CE463
*
      7 U *
+04C0 B0EDCDD7 93854BB0 0B0109C5 62727A38 00000000 00000000 00000000 00000000
* P . E *
+04E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
...

```

If you try to set the cloud credentials for HSM without defining and activating the cloud network connections in DFSMS, you receive an error message, as shown in Example 8-16.

Example 8-16 DFSMSHsm missing constructs messages

```
COMMAND REQUEST 00000074 SENT TO DFSMSHSM
ARC1584I SETSYS CLOUD - NAME IBMREDBOOKS NOT FOUND
ARC0100I SETSYS COMMAND COMPLETED
***

```



Part 3

Operation and usage

In this part, we show you how we set up a cloud and how we used the new functions to communicate with the cloud, send data to it, and retrieve data from it.

This part includes the following chapters:

- ▶ Chapter 9, “DFSMSHsm” on page 91
- ▶ Chapter 10, “Using automatic migration” on page 103
- ▶ Chapter 11, “Operational integration and reporting considerations” on page 109



DFSMSHsm

This chapter describes the changes that were made to DFSMSHsm to support the cloud tier.

This chapter describes the following topics:

- ▶ 9.1, “Cloud use overview” on page 92
- ▶ 9.2, “Cloud container management” on page 92
- ▶ 9.3, “Object management” on page 93
- ▶ 9.4, “Fast subsequent migration” on page 94
- ▶ 9.5, “Migration update and considerations” on page 95
- ▶ 9.6, “Recall considerations” on page 98
- ▶ 9.7, “LIST command updates” on page 98
- ▶ 9.8, “Auditing” on page 100
- ▶ 9.9, “REPORT command” on page 101

9.1 Cloud use overview

DFSMSHsm can use storage clouds by using DFSMSdss as a data mover to migrate and recall data sets from the cloud. The use of a cloud to migrate data sets can reduce your direct access storage device and tape requirements, and provide disaster recovery (DR) capability for migrated data sets.

DFSMSHsm tracks migrated data sets and their related objects in cloud storage in its Control Data Sets (CDSs). By using the Data Facility Storage Management Subsystem (DFSMS) Cloud construct to connect to the cloud (along with CDS information), DFSMSHsm can manage data on cloud storage the same way it manages offline data.

Unlike tape devices, the space that is left by a deleted object is returned to available space and can be used by other objects as they are created. This feature eliminates the need for recycling containers on storage cloud, which reduces the central processing unit (CPU) resources that are related to **RECYCLE** processing and the window that is necessary to run DFSMSHsm tasks. It also increases the availability of migrated data as RECYCLE processing holds Hierarchical Storage Manager (HSM) tapes during the recycle operation.

Also, the option of having an off-premise cloud increases the options that are available for DR. If your cloud is unaffected after a disaster, you can recover your production systems in an alternative location (including DFSMSHsm CDSs) and define DFSMS Cloud construct to connect to your cloud. This ability gives you access to all data sets that are migrated to the cloud and have their related CDS records.

After a recovery situation, run an **AUDIT** against CDS and cloud to report and correct any mismatches between them.

9.2 Cloud container management

DFSMSHsm can create cloud containers to store the objects. By default, the container name adheres to the syntax that is shown in Example 9-1.

Example 9-1 Default DFSMSHsm container name

```
SYSZARC.<HSMpl exname>.MIG.yyyyddd.<hash>
```

The **hash** value is a random hexadecimal string that ensures that there never are two containers with the same name.

By default, HSM creates a new container every 92 days. You can change the default value from 92 days to a shorter or longer period. Example 9-2 shows the **PATCH** command to change the default value to 10 days.

Example 9-2 PATCH command to change the container creation frequency

```
PATCH .MCVT.+50F X'0A'
```

Note: The default values for the period after which HSM creates a container was changed from 7 days to 92 days with APAR OA60278. You might still see the old value in your environment. As a best practice, change it to the new default or install the APAR.

With a shorter period, your **LIST** and **AUDIT** processing might become quicker, but many containers might impact the overall HSM performance. Be careful when considering to change it to a shorter period.

Just as DFSMSHsm can automatically create containers when required, it can also delete empty and no longer used containers that are owned by DFSMSHsm. You can configure DFSMSHsm to perform empty container deletion as part of the Secondary Space Management tasks by using the new **EMPTYCONTAINERDELETION(x)** keyword on DFSMSHsm **SETSYS MAXSSMTASKS** command.

Example 9-3 shows a sample **SETSYS** command to allow one Cloud Storage containers processing task. This value is the default value. To prevent DFSMSHsm from deleting empty containers, set the **EMPTYCONTAINERDELETION** to 0.

Example 9-3 Setting container deletion task

```
SETSYS MAXSSMTASKS(EMPTYCONTAINERDELETION(1))
```

If you disable automatic deletion of containers by DFSMSHsm, you must create your own process to manage empty containers.

9.3 Object management

DFSMSHsm can automatically create and delete objects from cloud storage by using DFSMSdss as the data mover. For each data set, DFSMSdss is started to migrate or recall the data by using Transparent Cloud Tiering (TCT).

Storage objects that are created by DFSMSHsm follow a new data set naming convention, which is similar to the naming convention that is used to Migration Level 1 (ML1) migrate data sets. The object naming convention is shown in Example 9-4.

Example 9-4 DFSMSHsm object naming convention

```
INSTPFX.HMIG.TCCCCHH.USER1.USER2.?YDDD
```

The naming convention consists of the following parts:

- ▶ **INSTPFX** is an installation-defined prefix.
- ▶ **TCCCCHH** is a form of how HSM expresses the time, where **CCCC** is the number of hundredths of seconds since the beginning of the hour and compressed into four alphanumeric digits. **HH** is the hour. When there is a conflict, **T** can change to be from **U - S** (starting from **T** and wrapping around).
- ▶ **USER1.USER2** are the first two qualifiers of the data set name that is being migrated.
- ▶ **?YDDD** is the Julian Date where **A - F** is for decade; for example, 2000 - 2060.

How DFSMSdss handles the data depends on the request that is performed by DFSMSHsm (a migration or recall process). These processes are described next.

9.3.1 Migration

When a migration process is started, DFSMSHsm calls DFSMSdss to perform the data movement. HSM is responsible for passing to Data Storage Services (DSS) the data set name, along with the cloud constructs, including cloud name, account, container, and object prefix. DFSMSdss then communicates with the DS8000 passing information that is related to the tracks that should be moved to the cloud, along with cloud-related information.

The metadata is stored in the cloud directly by the host for Swift clouds, or DS8000 for S3 and IBM Cloud Object Storage clouds. DFSMSdss returns control to DFSMSHsm after all data is moved to the cloud or after any failures during the process.

During the **DUMP** process, any data sets that are larger than 5 GB are broken up in 5 GB segments. **VALIDATE** processing is skipped for Virtual Storage Access Method (VSAM) data sets.

9.3.2 Recall

During a recall request, DFSMSHsm sends to DFSMSdss the data set name to be restored, along with the cloud attributes. DFSMSdss issues a request to the DS8000 for the objects that should be retrieved from the cloud. Metadata is retrieved by the host for Swift clouds, and by DS8000 for S3 and IBM Cloud Object Storage clouds.

At retrieval time, object segments (for data sets larger than 5 GB) are grouped, and data set extents are reduced when possible. During this phase, no REBLOCKing function is performed.

The storage objects can be deleted or retained during the recall process, if your HSMplex is configured to support fast subsequent migration.

9.4 Fast subsequent migration

When data sets are migrated to Migration Level 2 (ML2), they are stored on tapes until the data set expires or is recalled. If a recall occurs, the data set is not physically deleted from the tape, but the CDS records are marked as invalid. With Fast Subsequent Migration, the recalled data set can be reconnected to the tape, which eliminates the need to rewrite the tape data.

Use of the storage cloud also allows you to reconnect recalled data sets to the cloud objects, which prevents a new migration, and thus reduces the network traffic to the cloud.

A new **SETSYS** command option (see Example 9-5) is available to include in your DFSMSHsm parmlib to allow reconnection.

Example 9-5 Setting up fast subsequent migration

```
SETSYS CLOUDMIGRATION(RECONNECT(ALL))
```

9.5 Migration update and considerations

Data can be migrated to the cloud either by command, or during the automatic space management. The next topics will explain in more detail how you can manage your data for automatic and manual selection for migration.

9.5.1 Command-driven migration

A **CLOUD** parameter is available from the **MIGRATE** or **HMIGRATE** commands to target data sets to cloud. Example 9-6 shows a sample **HSEND MIGRATE** command with the **CLOUD** keyword.

Example 9-6 HSEND MIGRATE command cloud option

```
HSEND MIGRATE DSN(youdsname) CLOUD(yourcloud)
```

Note: The **CLOUD** parameter is mutually exclusive with **MIGRATIONLEVEL1**, **MIGRATIONLEVEL2**, and **CONVERT** parameters. Also, **COMPACT**, **COMPACTPERCENT**, **COMPACT(ALL)**, **CONVERSION(REBLOCKTOANY)**, and **CONCURRENT SETSYS** values are not used when migrating to the cloud.

To migrate a data set to the cloud, it must be SMS-managed. The types of data sets that can be migrated to the cloud, along with migration and recall restrictions, are listed in Table 9-1.

Table 9-1 Data set migration to cloud eligibility and considerations

Data set type	Can it be migrated to the cloud?	Comments
Non-SMS	N	Only SMS-managed data sets can be migrated to the cloud at the time of this writing.
Sequential	Y	
Extended Format	Y	
Extended format multi-volume	N	VSAM restrictions for HURBA=HARBA (used = allocated), and Multi-layer VSAM (volume count > stripe count) cannot be migrated.
Multi-extents sequential/partitioned	Y	Extent reduction is performed at recall time if possible.
Multi-volume sequential/partitioned	Y	
Multi-stripe sequential	Y	If SMS cannot provide enough volumes to keep the stripe count, the recall fails.
VSAM	Y	VALIDATE is not performed during migration.
Multi-extent VSAM	Y	VALIDATE is not performed during migration.
Multi-volume VSAM	Y	VALIDATE is not performed during migration.
VSAM with IBM AIX® and PATHs	Y	VALIDATE is not performed during migration.

Data set type	Can it be migrated to the cloud?	Comments
Data sets in volumes with simplex, two-site Metro Mirror, FlashCopy, Global Mirror, Metro Global Mirror (with or without HyperSwap).	Y	As of this writing, only volumes with XRC and Multi-Target Peer to Peer Remote Copy (PPRC) are not supported.
Data sets spanning more than 26 volumes	Y	An object cannot be restored to more than 26 volumes.
Multi-volume data sets spanning multiple DS8000s	Y	Cannot be restored in volumes spanning DS8000 systems.

The **HSEND MIGRATE** command that is issued from option ISPF panels to migrate a multi-extent sequential data set is shown in Example 9-7. There is no need to supply account, container, object prefix, or cloud credentials because they are handled by DFSMSHsm.

Example 9-7 HSEND MIGRATE issued from ISPF panel

```

Menu Options View Utilities Compilers Help

DSLIST - Data Sets Matching TCT.DEMO                               Row 1 of 1
Command ==>                                                       Scroll ==> PAGE

Command - Enter "/" to select action                               Message                               Volume
-----
HSEND MIGRATE DSN(/) cloud(IBMCOS)                               CLDD2F+
***** End of Data Set list *****

```

After the migration process is complete, the catalog entry is updated to reflect the new location of the data set. In ISPF, to differentiate data sets that are migrated to cloud from data sets on ML1 or ML2, a new **MIGRATC** volume serial number is used for cloud-migrated data sets, as shown in Example 9-8.

Example 9-8 MIGRATC volume entry

```

Command - Enter "/" to select action                               Message                               Volume
-----
TCT.DEMO.DTA1.CP3000.LOG                                         CLDC28
TCT.DEMO.DTA2.CP3000.LOG                                         CLDD24
TCT.DEMO.DTA3.CACH.REPORT                                        CLDD2B
TCT.DEMO.DTA4.XCSV                                              CLDD29
TCT.DEMO.DTA5.JCL                                               CLDC2A
TCT.DEMO.DTA6.TRS                                               MIGRATC

```

A new device type of x'00018000' is used to identify data sets that are migrated to the cloud. The ICF catalog entry is shown in Example 9-9 on page 97. Regardless of the migration tier, migrated data sets are always cataloged by using volser MIGRAT.

Example 9-9 Device type for data set migrated to the cloud

```

NONVSAM ----- TCT.DEMO.DTA6.TRS
IN-CAT --- CATALOG.MVSICF1.VCEBCU1
HISTORY
  DATASET-OWNER----- (NULL)      CREATION-----2018.318
  RELEASE-----2        EXPIRATION-----0000.000
  ACCOUNT-INFO----- (NULL)
SMSDATA
  STORAGECLASS ----TCTTEST      MANAGEMENTCLASS---NOBACK
  DATACLASS ----- (NULL)      LBACKUP ---XXXX.XXX.XXXX
VOLUMES
VOLSER-----MIGRAT DEVTYPE-----X'00018000' FSEQN-----0
ASSOCIATIONS----- (NULL)
ATTRIBUTES
***

```

The information that is necessary to recall the data set from the cloud is stored in HSM CDSs. You can issue a **FIXCDS DISPLAY** command to view the created CDS records. A sample **FIXCDS** command format to display a Migration Control Data Set (MCD) record is shown in Example 9-10.

Example 9-10 FIXCDS command to view a CDS record

```
HSEND FIXCDS D dsname DISPLAY
```

The command output for a cloud-migrated data set is shown in Example 9-11. Areas are highlighted to show recognizable eye catchers within the CDS record.

Example 9-11 Output of the FIXCDS command

```

F DFHSMBC, FIXCDS D TCT.DEMO.DTA6.TRS DISPLAY
MCH= 02580000 D53C0FE6 68920609 D53B23A7 A3FBB90A
+0000 6CC3D3D6 E4C48001 00340000 0118318F 00000000 0118319F 00000000 00000000 * N W N *
* CLOUD *
+0020 09522468 0118319F 40006C00 009002F1 00029388 3A3D6A27 000747AD 00020000 * 1 *
+0040 C3D3C4C3 F2F30200 00000000 3030200F 09204263 0118319F 00010000 00000000 *CLDC23 *
+0060 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * *
+0080 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * DFHS*
+00A0 D44BC8D4 C9C74BE3 F1F9F4F6 F0F94BE3 C3E34BC4 C5D4D64B C1F8F3F1 F9404040 *M.HMIG.T194609.TCT.DEMO.A8319 *
+00C0 40404040 40404040 00000000 00000000 00000000 00000000 00000000 * *
+00E0 00000000 00004040 40404040 40404040 40404040 40404040 40404040 * *
+0100 40404040 0007E3C3 E3E3C5E2 E3404040 40404040 40404040 40404040 40404040 * TCTTEST *
+0120 40404040 0006D5D6 C2C1C3D2 40404040 40404040 40404040 40404040 40404040 * NOBACK *
+0140 40404040 00000000 00000000 00880000 00000000 00000000 00000000 00000000 * *
+0160 00000000 0000C000 03E80000 00000000 00000000 03020200 00000000 00000400 * Y *
+0180 22404040 4000C400 00000000 0007E3C3 E3E3C5E2 E3000000 00000000 00000000 * D TCTTEST *
+01A0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * *
+01C0 00000000 00000000 0006C9C2 D4C3D6E2 40404040 40404040 40404040 40404040 * IBCOS *
+01E0 40404040 40404040 E2E8E2E9 C1D9C34B C1D9C3D7 D3C5E7F0 48D4C9C7 48F2F0F1 * SYSZARC.ARCPLEX0.MIG.201*
+0200 F8F3F1F6 40404040 40404040 40404040 40404040 00000000 *8316 *
ARC0197I TYPE D, KEY TCT.DEMO.DTA6.TRS, FIXCDS DISPLAY
ARC0197I (CONT.) SUCCESSFUL

```

The **FIXCDS** command output includes the regular data set information (such as SMS constructs) and the cloud-related information (including the cloud name) and the container where the data set is stored.

9.5.2 Automatic migration

The DFSMSHsm can also migrate data sets to the cloud during the automatic space management, including primary space management, interval migration, or on-demand migration.

To support automatic migration functions, new parameters were included in DFSMSHsm parmlib and SMS management class constructs. These changes are described in more detail in Chapter 10, “Using automatic migration” on page 103.

9.5.3 CPU utilization considerations

Because the data movement to and from the cloud is performed directly by the DS8000, it is expected to have a variation on CPU utilization by DFSMSHsm. You can create reports to estimate possible CPU savings from implementing cloud migration. The pre-implementation reporting is discussed in more detail in Chapter 11, “Operational integration and reporting considerations” on page 109.

9.6 Recall considerations

The RECALL process is automatically triggered when access to the data set is requested or the HSEND RECALL command is issued. There are no changes to the RECALL command.

During the RECALL, the Automatic Class Selection (ACS) routines are called to define the Storage Class, Management Class, and Storage Group for the data set. The data set can also be extent-reduced if possible. Example 9-12 shows the recalled data set from the HSEND MIGRATE command that was issued, where the recalled data set has a single extent.

Example 9-12 Recalled data set with consolidated extent

DSL	LIST - Data Sets Matching TCT.DEMO.DTA6	Row 1 of 1
Command	====>	Scroll ====> CSR
Command	- Enter "/" to select action	Tracks %Used XT

	TCT.DEMO.DTA6.TRS	168840 99 3

When data sets expire, all data and metadata objects that are related to the data set are automatically deleted from the storage cloud and the catalog entry is removed.

9.7 LIST command updates

The LIST command has updates to support the cloud. DFSMSHsm also gives you the options to list the following elements:

- ▶ Cloud
- ▶ Containers
- ▶ Objects

New **CLOUD**, **CONTAINER**, and **PREFIX** parameters can be used within the LIST command to retrieve cloud and container content information. The LIST command can list DFSMSHsm and non-DFSMSHsm owned containers, which give the users the chance to list user-created containers and retrieve object information.

The output from the LIST command can be directed to a terminal or data set. The sample command that is used to list IBMREDBOOKS cloud information is shown in Example 9-13.

Example 9-13 LIST command for a specific cloud

```
HSEND LIST CLOUD(IBM COS)
```

The command output is shown in Example 9-14 on page 99.

Example 9-14 LIST command output

CLOUD NAME	CONTAINER NAME
IBMCOS	SYSZARC.ARCPLEX0.MIG.2018127
IBMCOS	SYSZARC.ARCPLEX0.MIG.2018316
IBMCOS	SYSZARC.ARCPLEX0.MIG.2018120
IBMCOS	SYSZARC.ARCPLEX0.MIG.2018134

The output shows the cloud IBMCOS and lists four containers, such as:

SYSZARC.ARCPLEX0.MIG.2018316

DFSMSHsm and user-created containers data can be displayed by using the **HSEND LIST** command. Add the **CONTAINER(containername)** keyword to your **LIST** command. The **HSEND LIST** command lists the container SYSZARC.ARCPLEX0.MIG.2018316, as shown in Example 9-15.

Example 9-15 Listing a specific container in the IBMCOS cloud

```
HSEND LIST CLOUD(IBMCOS) CONTAINER(SYSZARC.ARCPLEX0.MIG.2018316)
```

The output from the command that is shown in Example 9-15 is shown in Example 9-16.

Example 9-16 LIST the content of a specific container

CLOUD NAME	CONTAINER NAME	PREFIX NAME
IBMCOS	SYSZARC.ARCPLEX0.MIG.2018316	DFHSM.HMIG.T194609.TCT.DEMO.A8319
		DFHSM.HMIG.T474010.TCT.DEMO.A8319

Listing objects is available by using **PREFIX** parameter. When listing objects, the cloud and container names must also be included. The **LIST** command with the **PREFIX** name is shown in Example 9-17. This option brings all of the data and metadata objects that are stored under the specific prefix.

Example 9-17 Listing objects by using the PREFIX keyword

```
HSEND LIST CLOUD(IBMCOS) CONTAINER(SYSZARC.ARCPLEX0.MIG.2018316)
PREFIX(DFHSM.HMIG.T194609.TCT.DEMO.A8319)
```

The output from the **LIST** command with the **CLOUD**, **CONTAINER**, and **PREFIX** command is shown in Example 9-18.

Example 9-18 Listing objects by PREFIX

```
CLOUD INFORMATION FOR DATASET: TCT.DEMO.DTA6.TRS
```

CLOUD NAME	CONTAINER NAME	OBJECT NAME
IBMCOS	SYSZARC.ARCPLEX0.MIG.2018316	DFHSM.HMIG.T194609.TCT.DEMO.A8319/DTPDSNLOO00001
		DFHSM.HMIG.T194609.TCT.DEMO.A8319/HDR
		DFHSM.HMIG.T194609.TCT.DEMO.A8319/TCT.DEMO.DTA6.TRS/APPMETA
		DFHSM.HMIG.T194609.TCT.DEMO.A8319/TCT.DEMO.DTA6.TRS/DTPDSHDR
		DFHSM.HMIG.T194609.TCT.DEMO.A8319/TCT.DEMO.DTA6.TRS/DTPVOLD01/NVSM/EXTENTS
		DFHSM.HMIG.T194609.TCT.DEMO.A8319/TCT.DEMO.DTA6.TRS/DTPVOLD01/NVSM/META
		DFHSM.HMIG.T194609.TCT.DEMO.A8319/TCT.DEMO.DTA6.TRS/DTPVOLD02/NVSM/EXTENTS
		DFHSM.HMIG.T194609.TCT.DEMO.A8319/TCT.DEMO.DTA6.TRS/DTPVOLD02/NVSM/META
		DFHSM.HMIG.T194609.TCT.DEMO.A8319/TCT.DEMO.DTA6.TRS/DTPVOLD03/NVSM/EXTENTS
		DFHSM.HMIG.T194609.TCT.DEMO.A8319/TCT.DEMO.DTA6.TRS/DTPVOLD03/NVSM/META

For more information about each object, see Table 3-1 on page 25.

9.8 Auditing

The tasks to audit DFSMSHsm data are vital to keep CDS records error-free, and identify, report, and correct any discrepancies between the CDS records and the data sets.

Unexpected software or hardware errors during migration and recall processes might leave migrated data sets and CDS records out of sync. Regularly auditing CDS and media reduces the number of orphan or invalid data in the physical media and the CDS.

AUDIT DATASET NAMES, LEVEL, and MCDS commands can perform a cloud-migrated data sets audit. If the CDS record indicates that the data set is stored in cloud storage, DFSMSHsm verifies that objects corresponding to the archive are in the expected cloud and container.

DFSMSHsm lists the objects in the cloud, beginning with the prefix that is stored in the MCD record. If the expected objects are found, it moves onto the next MCD record.

In addition, a new **CLOUD** parameter is available from the **AUDIT MEDIACONTROLS** command. This option allows you to audit a cloud and validate if the objects have corresponding CDS entries. If any inconsistencies are found, the **AUDIT** command reports the error back to the user.

Note: The **AUDIT** command does *not* automatically fix any identified inconsistencies because the orphan objects can be user data that is in the wrong container.

An **HSEND AUDIT** command to audit IBMREDBOOKS cloud is shown in Example 9-19.

Example 9-19 Auditing a specific cloud

```
HSEND AUDIT MEDIACONTROLS(CLOUD(IBM COS))
```

The output from the command that is shown in Example 9-19 is shown in Example 9-20, with one inconsistent entry.

Example 9-20 Example of AUDIT output

```
AUDIT MEDIACONTROLS(CLOUD(IBM COS)) ODS('TCTRBOOK.AUDIT')
```

```
/* ERR 210 CDD IS NOT FOUND FOR PREFIX DUMP.4XTENTS.PDS IN CONTAINER
/* DFHSM.HMIG.T015821.TCT.VSM.A7284
- END OF -      ENHANCED AUDIT - LISTING
-
```

The prefix **DUMP.4XTENTS.PDS** is not in the CDS as expected. The follow-up action is to investigate why it is missing and resolve the issue.

9.9 REPORT command

After you first implement a storage cloud to your z/OS systems, create reports about key metrics for analysis, such as the following ones:

- ▶ Number of data set migrations to the cloud
- ▶ Amount of data transferred
- ▶ Number of successful and failed requests
- ▶ Average times

The **REPORT** command provides all of the information that is necessary to efficiently monitor your data on the cloud.

You can create daily, weekly, or monthly reports and store this information for further analysis. You also can create simple programs to process the data and return reports with data growth, percentage of data sets migrated to the cloud versus standard migration, and usage trends.

A simple **REPORT** command to display migration statistics to the cloud is shown in Example 9-21.

Example 9-21 Report command for daily migration activity

```
HSEND REPORT DAILY FUNCTION(MIGRATION(TOCLLOUD))
```

It is also possible to retrieve recall specific information by issuing the **REPORT** command, as shown in Example 9-22.

Example 9-22 Report command for daily recall activity

```
HSEND REPORT DAILY FUNCTION(RECALL(FROMCLOUD))
```

Migration and recall reports display the following migration-to-the-cloud information:

- ▶ Number of data sets
- ▶ Number of tracks that are read and written
- ▶ Number of bytes read and written
- ▶ Number of system requests
- ▶ Number of user requests
- ▶ Failed requests
- ▶ Average age
- ▶ Average queue time
- ▶ Average wait time
- ▶ Average process time
- ▶ Average total time

A sample output from the **HSEND REPORT DAILY** command is shown in Example 9-23.

Example 9-23 Report output for daily activity

DAILY STATISTICS REPORT FOR 18/11/15

STARTUPS=000, SHUTDOWNS=000, ABENDS=000, WORK ELEMENTS PROCESSED=003023
DATA SET MIGRATIONS BY VOLUME REQUEST= 0000000, DATA SET MIGRATIONS BY
EXTENT REDUCTIONS= 0000000 RECALL MOUNTS AVOIDED= 00000 RECOVER MOUNTS
DATA SET MIGRATIONS BY RECONNECTION = 000000, NUMBER OF TRACKS RECONNE

HSM FUNCTION	NUMBER DATASETS	-----READ----- TRK/BLK	-----WRITTEN----- BYTES	TRK/BLK	BYTES	---	SYS
MIGRATION							
PRIMARY - CLOUD	0000382	00064520	001632042	00064520	001632042	000	

***** Bottom of Data *****

SMF records are also written when data sets are migrated to the cloud. You can use SMF records to create more specific reports that are based on users, high-level qualifiers, and other information. For more information about the SMF records and how to create reports, see Chapter 11, “Operational integration and reporting considerations” on page 109.



Using automatic migration

This chapter describes the automatic migration function and cloud usage.

The automatic migration is performed as part of primary space management, interval, or on-demand migration tasks. The ability to automatically select data that is eligible for cloud migration is vital to maximize central processing unit (CPU) savings that are related to data migration in DFSMSHsm.

This chapter describes the following topics:

- ▶ 10.1, “SMS support for automatic migration” on page 104
- ▶ 10.2, “Storage Group affinity enhancements” on page 106
- ▶ 10.3, “Space management functions” on page 107

10.1 SMS support for automatic migration

To enable the DFSMSHsm automatic migration to cloud, it is necessary to define the policies that define one of the following targets for migration of a data set:

- ▶ The existing Migration Level 2 (ML2) volumes
- ▶ The defined cloud storage that is configured to your systems

The definitions for data set migration are included in the SMS Management Class construct. Therefore, new fields are included to allow storage administrators to define conditions that are required for a data set to be eligible for cloud migration.

10.1.1 Management Class updates

The data sets cannot be automatically migrated from ML2 to cloud the same way that Migration Level 1 (ML1) data sets can be migrated to ML2. So, the management class was updated to create the rules for deciding the migration level tier.

The following fields are now available on Management Class under the Migration Attributes panels:

- ▶ Level 2 Days Non-usage

Specifies a direct migration to cloud storage. Current possible values are 0 and NOLIMIT.

A blank value also indicates NOLIMIT. A value of NOLIMIT indicates that a data set migrates to Level 2 based on the value of Level 1 Days Non-usage.

A value of 0 for this attribute indicates that the data set migrates to the cloud storage specified by the Cloud Name field if the Primary Days Non-usage value is met and the data set still is on Level 0.

The Level 2 Days Non-usage processing takes precedence over the Level 1 Days Non-usage value. The default value is NOLIMIT.

- ▶ Size LTE

The “Size Less than or equal” field shows the low data set size threshold in tracks. It is used to take the action described in the Action LTE field.

- ▶ Action LTE

The value in the Action LTE column shows which action to perform if the data set size is less than or equal to Size LTE. The following values are possible:

- NONE: No action is taken.
- ML1: Target migration level is ML1.
- ML2: Target migration level is ML2 regardless of the values for LEVEL 1 DAYS NON-USAGE.
- MIG: Target migration level is ML1 or ML2 according to the value of LEVEL 1 DAYS NON-USAGE.
- TRANS: Data set transition.
- CLOUD: Target migration level is CLOUD.

If no value is specified, DFSMSHsm performs the migration action the same way it is performed today.

► Size GT

The “Size Greater than” field shows the high data set size threshold in tracks. It is used to take the action described on “Action GT” field.

► Action GT

The value in the Action GT column shows which action to perform if the data set size is greater than Size GT. The following values are possible:

- NONE: No action is taken.
- ML1: Target migration level is ML1.
- ML2: Target migration level is ML2 regardless of the values for LEVEL 1 DAYS NON-USAGE.
- MIG: Target migration level is ML1 or ML2 according to the value of LEVEL 1 DAYS NON-USAGE.
- TRANS: Data set transition.
- CLOUD: Target migration level is CLOUD.

If no value is specified, DFSMSHsm performs the migration action the same way it is performed today.

► Cloud Name

The value in the CLOUD NAME shows the name of a previously defined cloud construct for the data set migration to the cloud during automatic migration (Primary Space Management, Interval migration, and On-Demand migration).

The new fields work in a way that is similar to the existing DFSMSHsm data set migration exit (MD), which controls the migration level for data sets selected by automatic migration processing. You can configure these fields by using these values in the exit, and you can disable the exit.

Table 10-1 Valid installation exit module names and their meanings

Module Name	Abbreviation	Meaning
ARCMDEXT	MD	Data set migration exit

Note: If you have the DFSMSHsm MD exit on, it overrides the values that are used in the management class construct.

EXIT0N is an optional SETSYS command parameter specifying active installation exits in the DFSMSHsm primary address space. For modname, substitute the module name of the installation exit that you want to be active.

For more information about exiting, see the following website:

<https://www.ibm.com/docs/en/zos/2.4.0?topic=command-exiton-specifying-active-installation-exits>

10.2 Storage Group affinity enhancements

The use of cloud storage by DFSMSHsm can affect the length of your space management windows, as your migration to the cloud can take longer than ML1/ML2 migrations depending on your network usage, latency, and cloud performance.

For some large systems, with several storage groups and DFSMSHsm images running, storage administrators might decide to spread the workload between these Hierarchical Storage Manager (HSM) images. There are enhancements to the storage group affinity to allow the selection of storage groups management for specific HSM hosts.

To enable the storage group affinity, you can use the new **SETSYS** command to identify the storage groups that are managed by an HSM host. Example 10-1 shows a sample SETSYS command to create the affinity between storage group BATCH1 and the HSMIMG2 DFSMSHsm image.

Example 10-1 Sample STORAGEGROUPAFFINITY command

```
F HSMIMG2,SETSYS STORAGEGROUPAFFINITY(BATCH1)
```

By setting up the storage group affinity, the specific HSM usage manages the defined storage groups. Define this command in your ARCCMD parmlib member to save your settings across initial program loads (IPLs).

The **SETSYS MIGRATIONAUTOCLLOUD** parameter enables the host to run cloud migration, and is an optional parameter that defines the possible target tier during automatic migration. The default value of this parameter is NOCLOUD:

- ▶ **ALL**: Specifying that data set migration to ML1, ML2, transition, and migration to cloud storage are performed.
- ▶ **CLOUDONLY**: Specifying that only data set migration to cloud storage is performed, as shown in Example 10-2.
- ▶ **NOCLOUD**: Specifying that only data set migration to ML1, ML2, and transition are performed.

Example 10-2 Specifying that only data set migration to cloud storage is performed

```
SETSYS MIGRATIONAUTOCLLOUD(CLOUDONLY)
```

10.3 Space management functions

This section describes the space management functions for managing SMS-managed storage, which is updated with cloud information:

- ▶ Automatic primary space management
- ▶ Automatic secondary space management

Automatic primary space management

During automatic primary space management, the following events occur:

- ▶ All DFSMSHsm hosts delete temporary data sets and expired data sets from the DFSMSHsm-managed volumes that they are processing. This operation is done under the control of the management class that is associated with each data set on the volume or the expiration date that is contained in the data set's volume table of contents (VTOC) entry.
- ▶ Under control of the data set management classes, all DFSMSHsm hosts release unused allocated space in physical sequential, partitioned, and extended format Virtual Storage Access Method (VSAM) data sets.
- ▶ During data set and volume processing, fast subsequent migration reconnects eligible data sets to the ML2 tape or cloud from which they were most recently recalled.
- ▶ Under the extent reduction function, all DFSMSHsm hosts also reduce the number of extents of physical sequential, partitioned, and direct-access data sets that exceed a specified number of extents. During the process of extent reduction, they also release any unused space in the data sets and compress partitioned data sets.
- ▶ Data sets that are eligible for a class transition are moved. If a data set is eligible for both a class transition and migration, the data set is migrated only, which prevents the data set from being moved from one level 0 volume to another for a transition, only to be later migrated to a migration volume.
- ▶ Data sets that are eligible for migration are migrated.

Primary space management continues until the SMS-managed volumes have the specified amount of available space. However, if deletion of expired data sets, fast subsequent migration, and space reduction of the remaining data sets achieves the specified volume available space low threshold, no actual data sets are moved.

If data sets are migrated during primary space management, they are migrated in compacted form if possible. Data sets that are expected to be smaller than 110 KB (where 1 KB is 1024 bytes) after they are compacted are candidates for small data set packing (SDSP) if enabled.

Automatic secondary space management

Automatic secondary space management schedules **TAPECOPY** commands for migration tape copy needed (TCN) records, deletes expired data sets from the migration volumes, deletes obsolete MCDs, VSRs, and DSRs during migration cleanup, and moves data sets (under control of the management class) from ML1 to ML2 volumes.

On the first day of the secondary space management cycle, DFSMSHsm examines the containers in the clouds that are defined to DFSMSHsm, looking for containers that do not have objects. Any empty containers are deleted. If the end of the Secondary Space Management window is reached and not all containers are examined, the processing continues in the next Secondary Space Management window.

Others space management considerations

When DFSMShsm processes an SMS-managed volume for space management, it records the time at which the volume was processed for cloud or non-cloud migrations. If another DFSMShsm host attempts to process the same volume for space management to cloud or non-cloud migrations, it checks the time at which the volume was last processed. If the volume was processed within the last 14 hours, DFSMShsm does not process the volume again for that type of target. A volume can be processed by one host for non-cloud migrations, and later processed by another host for cloud migrations.

Under Tasks for automatic space management, Specifying the DFSMShsm hosts that process each storage group, see 10.2, “Storage Group affinity enhancements” on page 106.



Operational integration and reporting considerations

This chapter reviews operational integration considerations.

The Storage cloud setup is the first stage in of moving your data into the cloud. Use of a strong operational framework including set of instructions, housekeeping jobs, and security considerations are encouraged to ensure that you can take the best from your cloud implementation.

We encourage you to consider the suggestions that are described in this chapter and plan and implement your own operations and automation procedures that are based on your system requirements.

This chapter includes the following topics:

- ▶ 11.1, “Pre-implementation reporting” on page 110
- ▶ 11.2, “Operational monitoring” on page 110
- ▶ 11.3, “Operational reporting” on page 112

11.1 Pre-implementation reporting

Before implementing automatic migration to the cloud, a storage administrator might want to report on possible central processing unit (CPU) savings that are related to offloading the migration task to the DS8000.

IBM released a package to assist storage administrators to collect and report data regarding CPU consumption that is related to migration, recall, and recycle tasks. You can download the package from the following link:

<ftp://public.dhe.ibm.com/eserver/zseries/zos/DFSMS/HSM/zTCT/>

You can use this package to extract and parse your SMF data. Then, you can generate reports and graphics to analyze the CPU savings that can be achieved with cloud migration.

11.2 Operational monitoring

After you finish configuring and activating your cloud settings, the cloud is ready for use. You then perform several tests, and they all complete with success, meaning that the hardware and software configurations are correct, and the network access to the cloud is functional.

Now, it is time to ensure that this access remains functional, and that any errors are tracked by automation systems.

11.2.1 Monitoring cloud setting changes

The first automation process that can be set up is to identify any cloud setting changes within DFSMSHsm. Whenever a change to DFSMSHsm storage cloud settings fails, a new ARC1581I message is issued with a description of the error that was encountered.

You might add message handling to our automation system to be notified whenever this message is issued and provide a timely reaction to the error. The error message code that is a result of an internal server error is shown in Example 11-1.

Example 11-1 ARC1581I message

```
Display Filter View Print Options Search Help
-----
SDSF SYSLOG      33.101 3090 3090 10/10/2016 0W          4,414  COLUMNS 52- 131
COMMAND INPUT ===>                                SCROLL ===> CSR
0010 ARC1581I UNEXPECTED HTTP STATUS 500 DURING A POST FOR 423
0010 ARC1581I (CONT.) URI
0010 ARC1581I (CONT.) https://IBMREDBOOKS.tuc.stglabs.ibm.com/v2.
0010 ARC1581I (CONT.) 0/tokens/ ERRTXT HTTP/1.1 500 Internal Server Error
```

11.2.2 Monitoring migration activities

You can track data set migration failures by using the ARC0279I message, which is issued when a data set migration fails. Set up automation to report on these failures based on this message.

An alternative approach is to create a REXX program to read the DFSMSHsm log or system log to periodically search for ARC0279I messages. A sample migration error that is the result of a non-existing cloud definition is shown in Example 11-2.

Example 11-2 ARC0279I message

```
Display Filter View Print Options Search Help
-----
SDSF OUTPUT DISPLAY DFHSM   STC00131  DSID    2 LINE 1,361  COLUMNS 02- 81
COMMAND INPUT ===>                               SCROLL ===> CSR
17.00.38 STC00131  ARC0279I MIGRATION REJECTED - CLOUD NAME VOIDCLOUD NOT 770
              770          ARC0279I (CONT.) FOUND
```

Use the information from the REXX-generated reports to track users that might be trying to migrate invalid data sets, or specifying wrong cloud information. The information helps to identify what information can be included in training for those users that are new to the cloud.

11.2.3 Monitoring reconnections

If you implemented periodic checks of the Hierarchical Storage Manager (HSM) SETSYS configurations, include the cloud configuration information. The new ARC0444I message identifies if cloud-recalled data sets can be reconnected to cloud objects. A sample output from the **HSEND QUERY SETSYS** command to display the cloud-reconnect setting is shown in Example 11-3.

Example 11-3 ARC0444I reconnection message

```
Display Filter View Print Options Search Help
-----
SDSF OUTPUT DISPLAY DFHSM   STC00131  DSID    2 LINE 120   COLUMNS 02- 81
COMMAND INPUT ===>                               SCROLL ===> CSR
              273          ARC0410I (CONT.) PERCENTAGE=020%, TAPEMAXRECALLTASKS=01, ML2
              273          ARC0410I (CONT.) NOT ASSOCIATED GOAL=010, RECONNECT(NONE)
06.17.32 STC00131  ARC0444I CLOUDMIGRATION RECONNECT(ALL)
06.17.32 STC00131  ARC0411I TAPESECURITY=PASSWORD, DEFERMOUNT
```

11.2.4 Other messages to consider

Other situations that can be monitored include the **DUMP** and **RESTORE** processes that are performed by DFSMSdss. Several new messages identify and describe errors during **DUMP** or **RESTORE** processing. Consider investigating the following messages to determine whether they can help build a robust operational cloud:

- ▶ ADR600E: DFSMSdss did not process the data set because of the condition code detected.
- ▶ ADR601E: DFSMSdss invokes the **ANTRQST** macro for an **MCLIST**, **STORE**, or **RETRIEVE** request and ANTRQST fails with the listed hex return code, reason code, and return information.

- ▶ ADR602E: DFSMSDss found that a backup exists with the same object prefix in the specified container.
- ▶ ADR604E: A failure occurred while trying to store an object that is related to the dump process or a data set. All related objects stored that use the object-pre fix-name are not usable because of a previous error that was encountered.
- ▶ ADR606E: A failure occurred while performing the identified z/OS Client Web Enablement Toolkit service.
- ▶ ADR607E: A failure occurred while performing the identified request.
- ▶ ADR609E: I/O errors were encountered while the indicated type of dump meta-record was being read during logical data set RESTORE processing.
- ▶ ADR610E: DFSMSDss detected an unexpected internal error during processing of an HTTP/HTTPS request.
- ▶ ADR612E: DFSMSDss encountered an error obtaining a SYSZADRO enqueue; the resource might be in use.
- ▶ ADR705E: A nonexistent storage class, management class, or cloud was specified in the STORCLAS, MGMTCLAS, or CLOUD keyword.

Other monitor options can also be implemented in your systems to monitor and control how clouds are used.

11.3 Operational reporting

There are different options for reporting on DFSMSHsm cloud usage. Whether by using the **HSEND REPORT** command or SMF records, plan to have a reporting and archiving job to analyze and retain storage cloud usage.

11.3.1 Building reports

The JCL and REXX that are included in this topic are intended to show you how to extract cloud migration and recall activity from a daily report, and append the data in a CSV format to output data sets. This file might then be downloaded and imported into a spreadsheet for further analysis.

The JCL that is used to run the report by running the REXX **RXMEMBER** procedure is shown in Example 11-4.

Example 11-4 JCL to run the REXX procedure

```
//JOBLIST1 JOB (XXXX), 'RUN RPT', NOTIFY=&SYSUID, MSGLEVEL=(1,1),
// MSGCLASS=W
//STEP1 EXEC PGM=IKJEFT01, REGION=8M
//SYSTSPRT DD SYSOUT=A
//HSMREPT DD DSN=YOUR.INPUT.DATA, DISP=SHR
//CLOUDRPT DD DSN=YOUR.OUTPUT.REPORT, DISP=(NEW,CATLG),
// LRECL=80, RECFM=FB, SPACE=(TRK,(1,1)), DSORG=PS
//SYSTSIN DD *
EX 'YOUREXX.DATASET(RXMEMBER)'
```

The REXX source code is shown in Example 11-5 on page 113. You can use this code as a base to develop your own specific reports.

Example 11-5 REXX source code

```
/* REXX */
"EXECIO * DISKR HSMREPT (STEM HSMREPT. FINIS)"
/* NUMBER OF MIGRATION TO CLOUD LINES */
MIG=0
/* NUMBER OF RECALLS TO CLOUD LINES */
REC=0
DO Z=1 TO HSMREPT.0
  /* GET REPORT DATE */
  IF LASTPOS('DAILY STATISTICS REPORT FOR',HSMREPT.Z) > 0 THEN
    PARSE VAR HSMREPT.Z 'DAILY STATISTICS REPORT FOR' REPDATE .
  IF LASTPOS('PRIMARY - CLOUD',HSMREPT.Z) > 0 THEN DO
    PARSE VAR HSMREPT.Z . . . NDS RTRK RBYT WTRK WBYT SYSR USRR FAIL ,
      AGE QTIME WTIME PTIME TTIME
    MIG = MIG + 1
    OUTMIG.MIG = REPDATE', 'NDS', 'RTRK', 'RBYT', 'WTRK', 'WBYT', 'SYSR',
      ||', 'USRR', 'FAIL', 'AGE', 'QTIME', 'WTIME', 'PTIME', 'TTIME
  END
  IF LASTPOS('CLOUD - PRIMARY',HSMREPT.Z) > 0 THEN DO
    PARSE VAR HSMREPT.Z . . . NDS RTRK RBYT WTRK WBYT SYSR USRR FAIL ,
      AGE QTIME WTIME PTIME TTIME
    REC = REC + 1
    OUTREC.REC = REPDATE', 'NDS', 'RTRK', 'RBYT', 'WTRK', 'WBYT', 'SYSR',
      ||', 'USRR', 'FAIL', 'AGE', 'QTIME', 'WTIME', 'PTIME', 'TTIME
  END
END
"EXECIO * DISKW CLOUDRPT (STEM HSMREPT. FINIS)"
```

Other reports can be created by using SMF records. Some suggestions of reports that can be generated include filtering migration and recall by data set high-level qualifiers, users, management class, data set size, and others. We suggest that you create at least one report that consolidates data sets by high-level qualifiers so that you can identify the applications that are making most use of cloud resources.

11.3.2 DCOLLECT reports

Along with the changes in DFSMSHsm Control Data Sets (CDSs) and SMS constructs to enable the use of cloud storage, the DCOLLECT was also updated to reflect the extra information available.

In the DCOLLECT record type 'MC', cloud-related fields are also displayed, including the cloud names the management class relates to, and actions to take based on data set size during migration.

The 'M' records are updated to include the cloud name length, cloud name, container name, and number of objects stored.

A sample usage for this extra information includes using the DCOLLECT to gather information about the containers that are created and owned by DFSMSHsm in the cloud, and the number of objects stored. This information might be specially valuable for large cloud environments, where list commands can take an extended amount of time to complete.



DFSMShsm full volume dump

DFSMShsm (Hierarchical Storage Manager (HSM)) is enhanced to support full volume dump (FVD) to, and restore from, cloud object storage. These full volume backups can be used to repair or recover a production environment that is corrupted either by a system failure, a human error, or compromised either by a cyberattack or internal fraud.

This chapter discusses answers to the following questions:

- ▶ Which use cases are appropriate for HSM FVDs?
- ▶ Which preparation steps do you need to consider?
- ▶ How to implement HSM FVD to cloud object storage?
- ▶ How you can manage your backups in the cloud?
- ▶ How to restore a dump from the cloud target?

Note: At the time of writing, data set level restore from an FVD copy in cloud object storage is not supported. If you must restore individual data sets, you can restore the volume, use **ICKDSF** to condition the volume, bring it online, and then copy the data sets by using physical data set **COPY**.

The data flow and the way HSM FVDs are stored in the cloud are described in Chapter 3, “Transparent Cloud Tiering” on page 19, where you also can find the supported cloud targets.

For more information about how to configure those cloud targets, including the IBM TS7700, see Chapter 6, “Configuring the IBM DS8000 for Transparent Cloud Tiering” on page 49.

This chapter includes the following topics:

- ▶ 12.1, “DFSMShsm full volume dump for Transparent Cloud Tiering overview” on page 116
- ▶ 12.2, “Setting up a DFSMSHsm full volume dump to cloud object storage” on page 120
- ▶ 12.3, “Dumping to cloud object storage” on page 121
- ▶ 12.4, “Restoring from cloud object storage” on page 122
- ▶ 12.5, “Additional operational commands” on page 123

Note: At the time of writing, the PTFs for APAR OA60278, which support HSM Full Volume Dump and **CDACREDS**, are not available because a problem was discovered. If you intend to use these functions, check for the availability of APAR OA64130, which will fix the issue.

12.1 DFSMSHsm full volume dump for Transparent Cloud Tiering overview

To support the HSM FVD enhancement, the **DEFINE DUMPCLASS** command parameter was updated with a new option that is called **CLOUD (cloud_network_connection_name)** to indicate that the FVD should be directed to the specified cloud object storage. The **cloud_network_connection_name** parameter is the name of a defined SMS network connection construct.

When designated with **CLOUD** dump classes, HSM FVD commands and auto-dump functions invoke DFSMSdss to store FVD copies in cloud object storage. When the dump copy is in cloud object storage, HSM full volume recover commands invoke DFSMSdss to restore full volumes from cloud object storage.

Note: At the time of writing, HSM FVD is not enabled by default. You enable it by using the HSM patch `PATCH .ARCCVT.+5D7 BITS(1.....)`. Add this patch to the HSM startup parmlib member `ARCCMDxx` to make it permanent.

With the patch, you can control the timing of activation. The patch is not intended to be an enable and disable switch. After you use the patch, do not turn off the patch without doing the following actions:

- ▶ Remove all the dump copies in the cloud from the HSM inventory.
- ▶ Alter all the dump classes to go to a target tape instead of the cloud.
- ▶ Remove the cloud definition from HSM by using the `setsys cloud(name(...))` command.

12.1.1 DFSMSHsm full volume dump use cases

You can implement HSM FVD to cover the following use cases:

- ▶ Use the IBM Db2 **BACKUP SYSTEM** utility to take volume-level copies of data and logs for a non-data sharing Db2 subsystem or a Db2 data sharing group and off load them to th cloud.
- ▶ Create point-in-time (PiT) volume copies by using HSM FVD.

Note: if you plan to use the Db2 **BACKUP SYSTEM** utility, all the Db2 data sets must be on Data Facility Storage Management Subsystem (DFSMS) managed volumes.

12.1.2 The DUMPCLASS attribute and how it works

HSM manages volume dumps by using dump classes. A *dump class* is a HSM-named set of characteristics that describe how volume dumps are managed.

Dump class names can be specified in the SMS storage groups, copy pools, and on the **ADDVOL PRIMARY** command for non-SMS managed volumes, which indicates which dump classes the auto-dump function should use and how many dump copies are made when the dump occurs.

The **DUMPCLASS(class)** parameter can be specified on the FVD command when processing SMS or non-SMS volumes, as shown in Example 12-1 on page 117 and Example 12-2 on page 117.

Example 12-1 DUMPCLASS BACKVOL command example

```
HSEND BACKVOL SG(SGRP1) DUMP(DUMPCLASS(TCTDMPC))
```

Example 12-2 DUMPCLASS RECOVER command example

```
HSEND RECOVER * TOVOLUME(PRIM01) UNIT(3390) FROMDUMP(DUMPLCLASS(TCTDMPC))
```

12.1.3 Containers

By default, a new container is created at 92-day intervals when volumes are dumped to cloud object storage. You can change the default value for DMP containers to an alternative value. Example 12-3 shows the **PATCH** command to change the default from 92 to 122 days.

Example 12-3 Changing the default container creation period for DMP containers

```
PATCH .ARCCVT.+598 X'7A'
```

The container naming rules are like the ones for HSM migration (see 9.2, “Cloud container management” on page 92). The dump copy container naming syntax is shown in Example 12-4.

Example 12-4 Default DFSMShsm dump container syntax

```
SYSZARC.HSMplexname.DMP.yyyyddd.hash
```

The container names consist of the following parts:

- ▶ SYSZARC is constant.
- ▶ HSMplexname is the defined HSMplex name.
- ▶ DMP is constant for FVD.
- ▶ yyyy is the four-digit year of the dump date stamp.
- ▶ ddd is the Julian day of the dump date stamp.
- ▶ hash is a random container name suffix.

Example 12-5 shows an example dump container name.

Example 12-5 DFSMShsm dump container name

```
SYSZARC.ARCPLEX0.DMP.2022093.6959FB19
```

Note: HSM specifies the container name in the **CONTAINER** keyword on the DFSMSdss **DUMP** and **RESTORE** commands.

12.1.4 Objects

For FVD to cloud object storage support, the volume metadata and extent tracks are stored in separate objects within the same cloud and organized under a pseudo-directory structure. The HSM base object name for a cloud dump copy follows a similar naming convention as the one for tape dump data sets. The dump data set name is passed to DFSMSdss in the **OBJPFX** keyword of the volume **DUMP** and **RESTORE** commands. Currently, a dump data set on tape is named by using the syntax that is shown in Example 12-6.

Example 12-6 Syntax of the object naming convention

```
prefix.DMP.dclass.Vvolser.Dyyddd.Tssmmhh
```

The object naming convention consists of the following parts:

- ▶ `prefix` is the prefix that is specified with the **SETSYS BACKUPPREFIX** command.
- ▶ `DMP` is constant.
- ▶ `dclass` is the dump class name.
- ▶ `V` is constant.
- ▶ `volser` is the source volume serial number.
- ▶ `D` is constant.
- ▶ `yy` is the last 2 digits of the year of the dump timestamp.
- ▶ `ddd` is the Julian day of the dump timestamp.
- ▶ `T` is constant.
- ▶ `ssmmhh` is the time in seconds, minutes, and hours.

Example 12-7 Object name example

```
DFHSM.DMP.TCTDMPC.VCLA024.D22174.T120618
```

12.1.5 Automatic dump

SMS storage group and copy pools can be dumped by using the auto-dump function. If `Auto Dump = Y` is specified in the storage group or copy pool construct, DFSMSHsm uses the dump classes that are specified for the corresponding storage group or copy pool auto-dump processing.

Non-SMS managed volumes can be dumped with the auto-dump function when the volumes are identified to the DFSMSHsm hosts by specifying the **AUTODUMP**(`dclass_name1`, `dclass_name2`, ..., `dclass_name5`) parameter on the **ADDVOL** command.

When a dump class is defined with **CLOUD**(`cloud_network_connection_name`), the associated storage groups, copy pool, or HSM-managed volumes are dumped to specific cloud object storage by auto-dump processing.

During dump processing, HSM allows a single dump class to be used if it designates **CLOUD**. Therefore, the auto-dump and dump class specifications should avoid having mixed **CLOUD** or **TAPE** or multiple **CLOUD** dump classes that run on the same day in the cycle for a storage group, copy pool, or volume. If a volume is dumped to cloud object storage, only that **CLOUD** dump class should be eligible to run for the storage group, copy pool, or volume. If multiple dump classes are eligible to be dumped and at least one of them targets **CLOUD**, HSM does the following tasks:

- ▶ **CLOUD** or **TAPE** dump classes:
 - Dump to all the eligible **TAPE** dump classes.
 - Skip **CLOUD** dump classes and issue a message. Storage groups or volumes were not dumped to **CLOUD** due to multiple eligible dump classes.

Note: The storage administrator can issue the **BACKVOL DUMP(DUMPCLASS(dc1ass_name))** command to dump to the missed dump classes. A single **CLOUD** dump class can be specified on each command.

- ▶ Multiple **CLOUD** dump classes and no **TAPE** dump classes:
 - Dump to the first **CLOUD** dump class.
 - Fail the rest of the dump classes and issue a message.

Note: The storage administrator can issue the **BACKVOL(DCLASS(dc1ass_name))** command to dump to the missed dump classes.

12.1.6 Expiring dump copies

Deletion of expired dump copies occurs in phase 1 of auto-dump processing on the primary host. In tape dumps, the status of the dump containing the deleted copies is changed from expired to available if they remain under HSM control. The cloud dump copies also can be deleted by using the new **DDELETE** and the existing **FRDELETE** commands.

Automatic dump expiration is performed at the dump class level for copy pool volumes. The last dump copy is never deleted during expiration processing or version roll-off. It is deleted only when done so by a command:

- ▶ Use **FRDELETE** to delete the last dump class for a copy pool version.
- ▶ Use the **LIST DUMPVOLUME** command with the **SELECT(EXPIRED)** option to get a list of expired dumps. Then, issue **DELVOL** commands with the **LASTCOPY** option for the associated dump volumes.
- ▶ Use the **LIST PRIMARYOVLUME(srcvo1) ALLDUMPS BCDS** command to get dump copy information. Then, issue the **DDELETE** command with the **LASTCOPY** option to delete a non-copy pool cloud dump copy.

12.1.7 Deleting empty dump containers

Empty DMP container deletion runs at the end of dump cleanup (phase 4). Automatic dump cleanup on a primary host scans for inactive empty DFSMSHsm DMP containers and deletes them. A dump container within the current creation period is considered active and is not deleted.

12.2 Setting up a DFSMSHsm full volume dump to cloud object storage

This section describes the following topics:

- ▶ Defining a DFSMS cloud network
- ▶ Defining DUMPCLASS
- ▶ Defining a copy pool

12.2.1 Defining a DFSMS cloud network

As described in Chapter 3, “Transparent Cloud Tiering” on page 19, DFSMS and HSM need a connection to cloud object storage to store and retrieve metadata objects and manage the dump copies.

For more information about how to define cloud object storage targets for DFSMS in a panel of the Interactive Storage Management Facility (ISMF) interface, see 7.4, “Creating a DFSMS cloud network connection by using ISMF” on page 70.

Note: You may use an existing cloud network because unique containers are created separately from the volume dump objects of your existing migration objects.

12.2.2 Defining DUMPCLASS

The new optional **CLOUD** subparameter was added to the **DEFINE DUMPCLASS** command to specify that the associated volumes should be offloaded to cloud object storage. The **cloud_network_connection_name** parameter is the name of a defined SMS network connection construct. The syntax and an example of the **DEFINE DUMPCLASS** are shown in Example 12-8 and Example 12-9.

Example 12-8 Syntax of the DEFINE DUMPCLASS command

```
DEFINE DUMPCLASS(class CLOUD(cloud_network_connection_name))
```

Example 12-9 DEFINE DUMPCLASS CLOUD example

```
DEFINE DUMPCLASS(TCTDMPC CLOUD(TS7700SYNC))
```

In Example 12-9, dump class TCTDMPC represents the class whose copies are kept in cloud object storage, and TS7700SYNC represents the name of the SMS network connection construct.

Note: If the specified name does not exactly match what is in the SMS definition, **DEFINE DUMPCLASS** fails.

The following parameters are not applicable when the dump copy is in the cloud object storage because these meetings apply to dump tape volumes:

- ▶ **AUTOREUSE** | **NOAUTOREUSE**
- ▶ **DATASETRESTORE** | **NODATASETRESTORE**
- ▶ **ENCRYPTION**
- ▶ **FRRECOV(AFM(NO|YES))**
- ▶ **HWCOMPRESS(NO | YES)**

- ▶ **STACK** | **MAXSTACK**, **MINSTACK**
- ▶ **TAPEEXPIRATIONDATE**
- ▶ **UNIT**(unitname)

For more information about the **DUMPCCLASS** optional parameters, see [DUMPCCLASS: Adding or changing a volume dump class](#).

12.2.3 Defining a copy pool

HSM manages the usage of volume-level fast replication functions, such as FlashCopy and snapshot. These functions provide PIT copy services that can quickly copy data from a source location to a target location. Using a copy pool, you can specify the pool storage groups that you want HSM to process collectively for fast replication.

Each pool storage group in a copy pool contains the name of the associated target copy pool backup storage group. A copy pool backup storage group is a type of SMS storage group that contains eligible target volumes that HSM can select for fast replication backup versions.

For more information about defining a copy pool by using ISMF, see [Defining a copy pool](#).

Note: It is a best practice that you create copy pools for cloud dumps. If you already have copy pools that are defined in your environment, you can use the same characteristics as the existing copy pools.

Be sure to specify the cloud-based dump class as defined in your **DEFINE DUMPCCLASS**.

12.3 Dumping to cloud object storage

Volumes can be dumped to the cloud either by command or during the automatic dump process. The following sections explain how you can manage your volumes for manual and automatic dump.

12.3.1 Command-driven dump

There are no changes to the **BACKVOL** command syntax. The cloud-defined dump classes provide all the necessary information.

Example 12-10 shows a sample of the **BACKVOL VOLUMES** command with a cloud-defined dump class.

Example 12-10 HSEND BACKVOL VOLUMES command example

```
HSEND BACKVOL VOLUMES(CLA024) DUMP(DUMPCCLASS(TCTDMPC))
```

Example 12-11 shows a sample of the **BACKVOL Storage Group** command with a cloud-defined dump class.

Example 12-11 HSEND BACKVOL SG command example

```
HSEND BACKVOL SG(SGTEST) DUMP(DUMPCCLASS(TCTDMPC))
```

Example 12-12 shows a sample of the **BACKVOL PRIMARY** command with a cloud-defined dump class. This command causes HSM to back up every volume that is identified to it as a primary volume by the **ADDVOL** command, and not only those volumes with the **AUTOBACKUP** attribute.

Example 12-12 HSEND BACKVOL PRIMARY command example

```
HSEND BACKVOL PRIMARY DUMP(DUMPCLASS(TCTDMPC))
```

12.3.2 Automatic dump

There are no changes that are needed for an automatic dump to cloud object storage. The Dump class indicates when a cloud object storage should be targeted. Both SMS and non-SMS volumes are supported for automatic dump.

Note: When cloud is targeted, only a single dump class may be specified. Automatic dump dumps to the first class and skips the others. For more information, see 12.1.5, “Automatic dump” on page 118.

12.4 Restoring from cloud object storage

The HSM full volume recover command invokes DFSMSdss to restore (retrieve) full volumes from cloud object storage. Common recover queue (CVQ) supports full volume restore from cloud object storage.

12.4.1 RECOVER

There are no changes to the **RECOVER** command syntax for this enhancement. **RECOVER FROMDUMP** processing is updated to support full volume restore from cloud object storage. Here are examples of ways to restore a full volume from a dump copy.

Example 12-13 shows a **RECOVER FROMDUMP DUMPCLASS** command.

Example 12-13 RECOVER FROMDUMP DUMPCLASS example

```
RECOVER * TOVOLOLUME(VOL001) FROMDUMP(DUMPCLASS(TCTDMPC))
```

Example 12-14 shows a **RECOVER FROMDUMP DATE** command.

Example 12-14 RECOVER FROMDUMP DATE example

```
RECOVER * TOVOLUME(VOL001) FROMDUMP DATE(221714)
```

12.5 Additional operational commands

This section covers additional commands that were added or updated to support HSM FVD to cloud object storage, which includes the new **DDELETE** command to explicitly delete cloud dump copies, and the **LIST** command, which is updated to include cloud information for dump copies.

12.5.1 DDELETE

The **DDELETE** command allows the deletion of dumps that are associated to individual volumes or to entire storage groups.

Note: The **DDELETE** command is intended for occasional cloud dump copy deletion. It is not designed for bulk deletion. Use automatic dump expiration to manage dumps and delete empty dump containers instead.

Do not issue **DDELETE** commands when automatic dump is active.

With **DDELETE STORAGEGROUP**, you specify a storage group with the dump copies that you want to delete, as shown in Example 12-15.

Example 12-15 DDELETE STORAGEGROUP syntax

```
HSEND DDELETE STORAGEGROUP(storage_group_name) DATE(yyyy/mm/dd)
```

With **DDELETE VOLUMES**, you specify the volumes with the dump copies that you want to delete, as shown in Example 12-16.

Example 12-16 DDELETE VOLUMES syntax

```
HSEND DDELETE VOLUMES(volser) DATE(yyyy/mm/dd) TIME(hhmmss)
```

With **DATE** and **TIME**, you specify the particular cloud dump copy to delete:

- ▶ **DATE(yyyy/mm/dd)** alone is used to delete the cloud dump copy of specified **STORAGEGROUP** volumes. The oldest dump that is generated on the specified date is deleted.
- ▶ **DATE(yyyy/mm/dd)** and **TIME(hhmmss)** are used together to delete the cloud dump copy of a specified volume. The dump that is generated on the specified date and time is deleted.

Note: **DATE** and **TIME** are required when the **VOLUMES** parameter is specified. **TIME** cannot be specified when the **STORAGEGROUP** parameter is specified.

LASTCOPY is an optional parameter that must be specified to delete the last valid dump copy of a source volume, as shown in Example 12-17. **LASTCOPY** is ignored if it is not part of the only copy.

Example 12-17 DDELETE LASTCOPY syntax

```
HSEND DDELETE STORAGEGROUP(storage_group_name) DATE(yyyy/mm/dd) LASTCOPY
```

12.5.2 LIST

The **LIST** command is updated to include cloud information for dump copies. You can use **LIST** to display information about the status of your dump copies.

The **LIST COPYPOOL**, **LIST CLOUD**, and **LIST DUMPCLASS** outputs are updated to display cloud information for dump copies in cloud object storage. The syntax of these commands has not changed.

Example 12-18 shows the syntax of the **LIST COPYPOOL** command. CPX is the name of the SMS-defined copy pool.

Example 12-18 LIST COPYPOOL command example

```
HSEND LIST COPYPOOL(CPX)
```

Example 12-19 shows the syntax of the **LIST CLOUD** command. TS7700SYNC is the name of a defined SMS cloud network connection.

Example 12-19 LIST CLOUD command example

```
HSEND LIST CLOUD(TS7700SYNC)
```

Example 12-20 shows the syntax of the **LIST DUMPCLASS** command. TCTDMPC is the name of the defined dump class.

Example 12-20 LIST DUMPCLASS command example

```
HSEND LIST DUMPCLASS(TCTDMPC)
```

For more information about the **LIST** command and its subparameters, see [LIST command: Listing information from the MCDS, BCDS, and OCDS](#).



Data Facility Storage Management Subsystem full volume dump

Instead of using Hierarchical Storage Manager (HSM), you can use DFSMSdss directly to dump and restore full volumes to and from cloud object storage by using Transparent Cloud Tiering (TCT) with the full volume dump (FVD) function. These full volume backups can then be used to repair or recover a production environment that was corrupted by either a system failure, a human error, or compromised by either a cyberattack or internal fraud.

This chapter describes how to implement FVD to cloud object storage, which preparation steps you need to consider, how you can manage (list and delete) FVD backups in the cloud and for which use cases the FVD is appropriate. Finally, it explains how to restore an FVD from the cloud target.

Note: DFSMSdss does not support restoring individual data sets from FVDs. If you need to restore individual data sets, you can restore the volume, use ICKDSF to condition the volume, bring it online and then copy the data sets by using physical data set COPY.

The data flow and the way FVDs are stored in the cloud are the same as described in Chapter 3, “Transparent Cloud Tiering” on page 19, where you can also find the supported cloud targets.

How to configure those cloud targets including the IBM TS7700 is described in Chapter 6, “Configuring the IBM DS8000 for Transparent Cloud Tiering” on page 49.

This chapter includes the following topics:

- ▶ 13.1, “FVD for TCT overview” on page 126
- ▶ 13.2, “Creating an FVD to cloud object storage” on page 128
- ▶ 13.3, “Restoring an FVD from cloud object storage” on page 129
- ▶ 13.4, “Managing FVD backups” on page 130

13.1 FVD for TCT overview

FVD enables you to create not just backups of inactive but also active data to be stored in a cloud target. The FVD data flow is directly between the DS8000 and the target cloud, and does therefore not consume mainframe central processing unit (CPU) cycles.

You can create offline copies of your data, which can be used to protect your business against data loss and data corruption, thus meeting regulatory and compliance guidance. In any malicious event affecting your data, you can perform an FVD restore of your stored data in the cloud target back into your production environment.

The FVD to the cloud is in principle a standard DFSMSdss FVD whose target is an object store. The difference and specifics of the FVD to cloud are:

1. The configuration of the cloud target as described in Chapter 6, “Configuring the IBM DS8000 for Transparent Cloud Tiering” on page 49.
2. The implementation of your cloud credentials handling.
3. The new parameters to be used in the FVD job to target the cloud.
4. The handling and managing of the dumps with new DFSMSdss commands.

Note: When accessing the cloud object storage, Data Facility Storage Management Subsystem (DFSMS) needs the cloud credentials (user ID and password). To avoid having to specify the password in clear text in all job definitions, use the CDACREDS method to store the credentials securely, as described in 8.1, “Managing credentials by using Cloud Data Access” on page 80.

13.1.1 FVD use cases and how it works

You can implement FVD to cover the following use cases:

- ▶ Create point-in-time (PiT) backups of IBM Db2 image copies by using DFSMSdss and FlashCopy.
- ▶ Create PiT volume copies by using DFSMSdss and FlashCopy with Dump Conditioning.
- ▶ Retain versions of Safeguarded Copy backups.

The FVD may be performed against FlashCopy target volumes, or directly from production volumes. Dumping from a FlashCopy target is a best practice because it reduces the amount of time that the production volume is reserved. Using dump conditioning, this volume dump looks as if it was taken directly from the production volumes and not from the intermediate FlashCopy volumes. You can use the same methods to create these FlashCopy copies as you would use for classical dumps to disk or tape.

If you use aTS7700 as object storage target, you can enable compression and save significant space in the TS7700 object storage partition. For more information, see 3.9, “Transparent Cloud Tiering compression with TS7700” on page 32.

The workflow, which performs an FVD from the FlashCopy volumes targeting to the cloud is illustrated in Figure 13-1 on page 127. The workflow is the same when you are running an FVD of Db2 image copies.

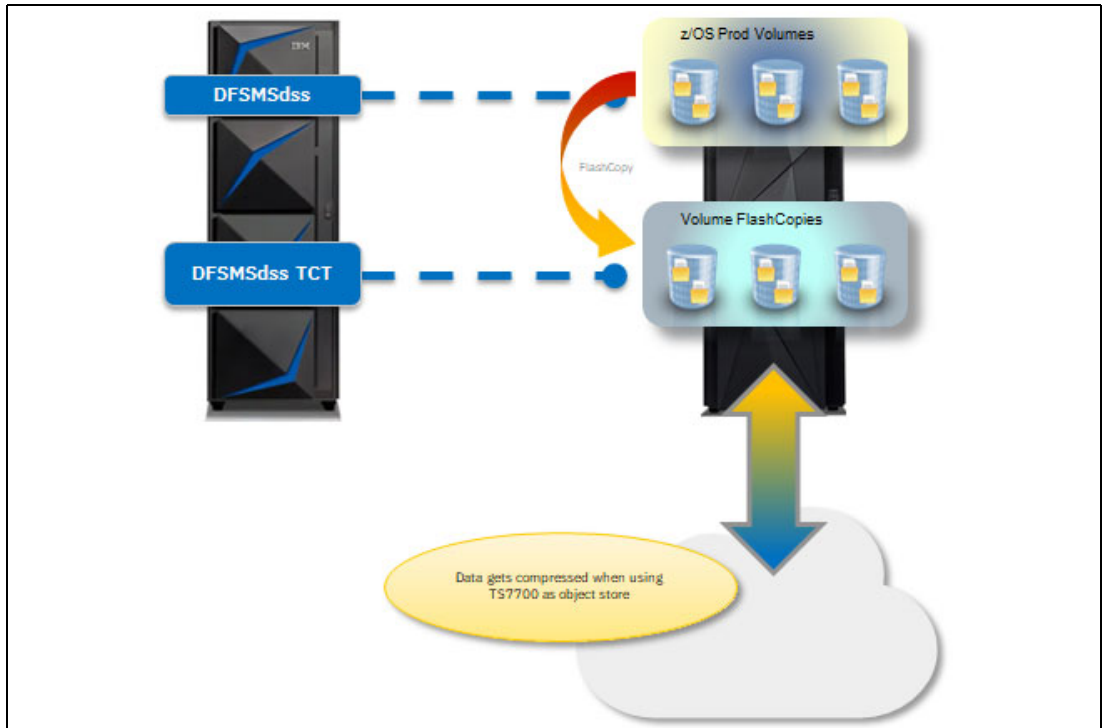


Figure 13-1 Performing an FVD to a cloud object store workflow

Another use case, which performs an FVD from Safeguarded Copy recovery volumes targeting the cloud is illustrated in Figure 13-2. The number of Safeguarded Copy copies that you can keep online is finite, and if you must keep more backups than you can keep with Safeguarded Copy, you can periodically recover a volume and dump it to the cloud.

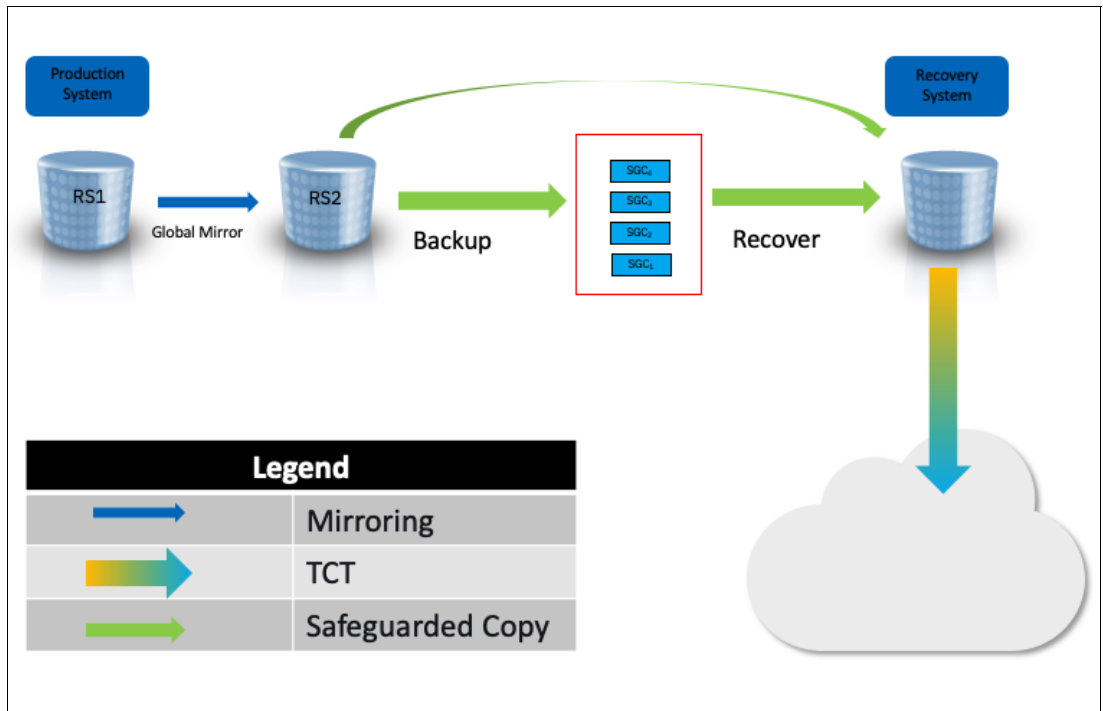


Figure 13-2 FVD from a Safeguarded Copy recovery volume

Whether you are restoring from a backup of a production volume, a FlashCopy target or a Safeguarded Copy recovery volume, there is no need to manually recall the data before the restore can be done. The restore is directly performed to the production volumes. If the data was compressed, it is decompressed before it is written to the production volumes.

The corresponding FVD recovery workflow is depicted in Figure 13-3.

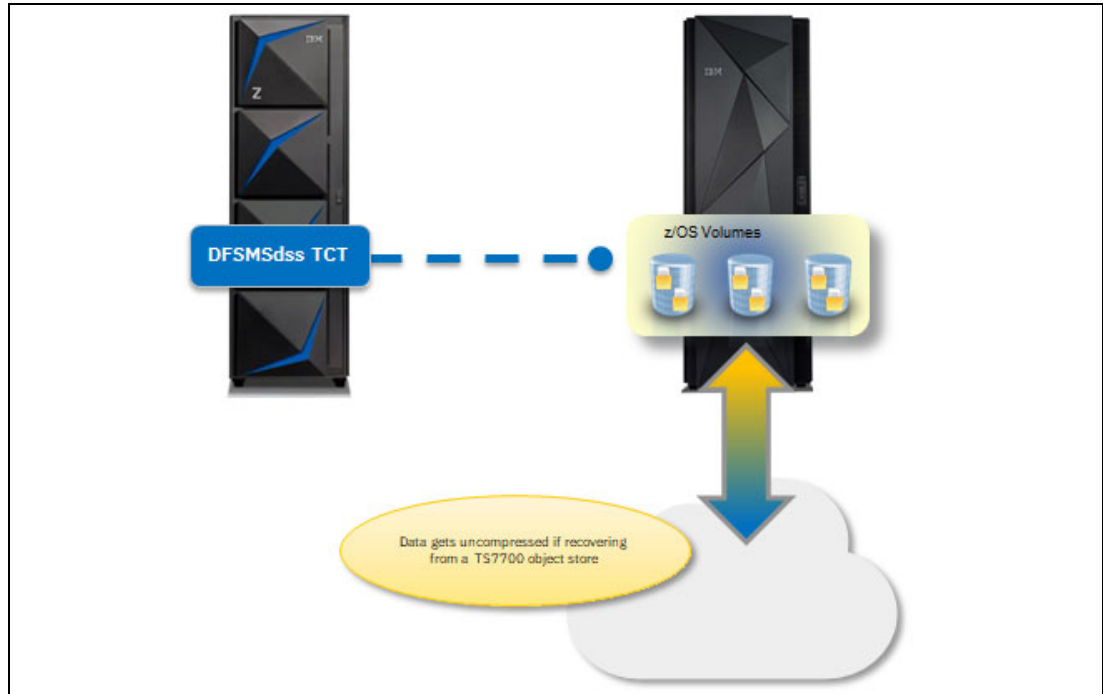


Figure 13-3 FVD restore workflow

13.2 Creating an FVD to cloud object storage

The job that you establish to run an FVD looks like a usual FVD with the following new parameters:

- ▶ **CLOUD(c1oudname)**: Invoke the cloud target that was defined in DFSMS.
- ▶ **CONTAINER(conta inername)**: Targets the specified container that is defined in the cloud object storage. If the container does not yet exist, it is created when running the FVD.
- ▶ **CDACREDS**: Tells the FVD to use the stored CDA credentials when connecting to the object storage.
- ▶ **OBJECTPREFIX('objectprefix')**: This prefix makes the created FVD definite so that it can recognize FVDs that were created for the same data. A best practice approach for the prefix is to use IBM z/OS variables if you plan to perform regular FVDs of the same data. Some examples are:
 - **&DATE/&TIME**
 - **&JOBNAME/&PREFIX**

For more information, see [Coding variables in JCL](#).

The JCL snippet in Example 13-1 on page 129 shows a sample dump job. In this example, the cloud name is **TS7700CLOUD** that targets an IBM TS7700 object partition. The container name is defined as **DSSDUMPS**. We use the predefined CDA credentials to access the cloud. The object prefix is set to the **jobname** of the FVD job followed by the **VOLSER** for which the FVD is performed and the **date** when the FVD was created.

The SYSPRINT in this example is written into a IBM z/OS Generation Data Group (GDG) file, which is a best practice to track the created FVD information.

Example 13-1 Sample job definition to create an FVD

```
//STEP0001 EXEC PGM=ADRSSU
//SYSPRINT DD DISP=(NEW,CATLG),DSN=MY.CLOUD.INVENTORY(+1)...
//DISK1 DD UNIT=SYSDA,DISP=SHR,VOL=SER=VOL001
//DISK2 DD UNIT=SYSDA,DISP=SHR,VOL=SER=VOL002
//DISK3 DD UNIT=SYSDA,DISP=SHR,VOL=SER=VOL003
//SYSIN DD *,SYMBOLS=(EXECSYS)
PARALLEL
DUMP INDDNAME(DISK1) CLOUD(TS7700CLOUD) -
      CONTAINER(DSSDUMPS) CDACREDS -
      OBJECTPREFIX('&JOBNAME./VOL001/&DATE. ')-
      WAIT(0,0) ALLDATA(*) ALLEXCP
DUMP INDDNAME(DISK2) CLOUD(TS7700CLOUD) -
      CONTAINER(DSSDUMPS) CDACREDS -
      OBJECTPREFIX('&JOBNAME./VOL002/&DATE. ')-
      WAIT(0,0) ALLDATA(*) ALLEXCP
DUMP INDDNAME(DISK3) CLOUD(TS7700CLOUD) -
      CONTAINER(DSSDUMPS) CDACREDS -
      OBJECTPREFIX('&JOBNAME./VOL003/&DATE. ')-
      WAIT(0,0) ALLDATA(*) ALLEXCP
```

For more information and a complete list of the options, see DUMP FULL command syntax, found at:

<https://www.ibm.com/docs/en/zos/2.4.0?topic=dfsmsdss-dump-full-command-syntax>

13.3 Restoring an FVD from cloud object storage

A restore of an FVD in the cloud is like a restore from a classical dump on disk or tape, except the additional parameters you need to define the cloud object specifics. In Example 13-2 on page 129, we provide a sample job definition for a restore from the cloud.

Example 13-2 Sample job definition to restore an FVD

```
//STEP0002 EXEC PGM=ADRSSU
//DISK1 DD UNIT=SYSDA,DISP=SHR,VOL=SER=VOL001
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
RESTORE -
      FULL COPYVOLID -
      OUTDDNAME (DISK1) -
      WAIT(0,0) PURGE -
      CLOUD(TS7700CLOUD) -
      CDACREDS -
      CONTAINER(DSSDUMPS) -
      OBJECTPREFIX('DNIGHTLY/VOL001/20201028')
```

The restore is performed directly to the production volumes as shown in Figure 13-3 on page 128. The example JCL will restore the FVD of **vo1ser VOL001** created the **2020-10-28** with the job named **DNIGHTLY**.

For more information about the DFSMSdss restore syntax, see [RESTORE FULL command syntax](#).

13.4 Managing FVD backups

Along with the ability to do FVD to cloud, DFSMSdss provides the capability to manage those dumps. Other than IBM DFSMSshm migrated data, DFSMSdss does not manage dumps automatically. There is no automatic expiry and deletion, and no space management. You must track your dumps yourself and delete the one that you do not need anymore.

For this purpose, DFSMS supports the **CLOUDUTILS LIST** and **CLOUDUTILS DELETE** commands. They manage your FVD backups to list and delete them as required.

- ▶ The **CLOUDUTILS LIST** command shown in Example 13-3 on page 130:
 - Creates a list of all the containers in a cloud, all the dumps within a container or a subset of dumps in a container depending on how you specify the parameters **CONTAINER** and **OBJECTPREFIX**:
 - Specifying **CONTAINER** lists all available FVDs in that container.
 - With the **OBJECTPREFIX** parameter, it shows all the FVDs in the specified container with the defined prefix.
 - With no **CONTAINER** and no **OBJECTPREFIX** specified, the command lists all available containers.
 - With no **OBJECTPREFIX** specified, the command lists all available dumps in a container.
 - **OBJECTPREFIX** supports the '*' wildcard.
 - Access to the **CLOUDUTILS LIST** command requires at least read access to **STGADMIN.ADR.CLOUDUTILS** facility class profile.

Example 13-3 Cloud list creation JCL

```
/*  
//STEP0001 EXEC PGM=ADRDSSU  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
CLOUDUTILS LIST CLOUD(TS7700CLOUD) CDACREDS -  
CONTAINER(DSSDUMPS) -  
OBJECTPREFIX('*')  
//
```

The command that is issued in the example lists all available FVD for the object store **TS7700CLOUD** in the specified container **DSSDUMP**, as no specific prefix is defined but a wildcard.

- ▶ The **CLOUDUTILS DELETE** command is used in Example 13-4 on page 131:
 - Deletes one or more dumps in a container or entire containers.
 - **CLOUDUTILS DELETE** considerations:
 - It is required to specify a **CONTAINER**.
 - **OBJECTPREFIX** supports the '*' wildcard, and it is optional.
 - Use **FORCE** to delete an entire non-empty container.
 - Access to the **CLOUDUTILS DELETE** command requires at least read access to **STGADMIN.ADR.CLOUDUTILS.DELETE** facility class profile.
 - Deleting non-empty containers with **CLOUDUTILS DELETE FORCE** requires at least read access to the **STGADMIN.ADR.CLOUDUTILS.FORCE** facility class profile.

Example 13-4 Deleting FVD backups in the cloud JCL

```

/*
//STEP0001 EXEC PGM=ADRDUSSU
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
CLOUDUTILS DELETE CLOUD(TS7700CLOUD) CDACREDS -
CONTAINER(DSSDUMPS) -
OBJECTPREFIX('DNIGHTLY/VOL001/202001*')
//

```

The JCL deletes all FVDs that are created with the prefix **DNIGHTLY/VOL001/202001*** in the container **DSSDUMPS**, which in this case is a Storage Group on the IBM TS7700 object store partition.

13.4.1 Use case example

Suppose a client creates nightly FVDs of a set of volumes and keeps one week worth of dumps. Within the **SYSZADR.DSSDUMP** container, we have a week worth of backups for **vo1001** and **vo1002**. Figure 13-4 shows an example of a **SYSZADR** content for the container **DSSDUMP**.

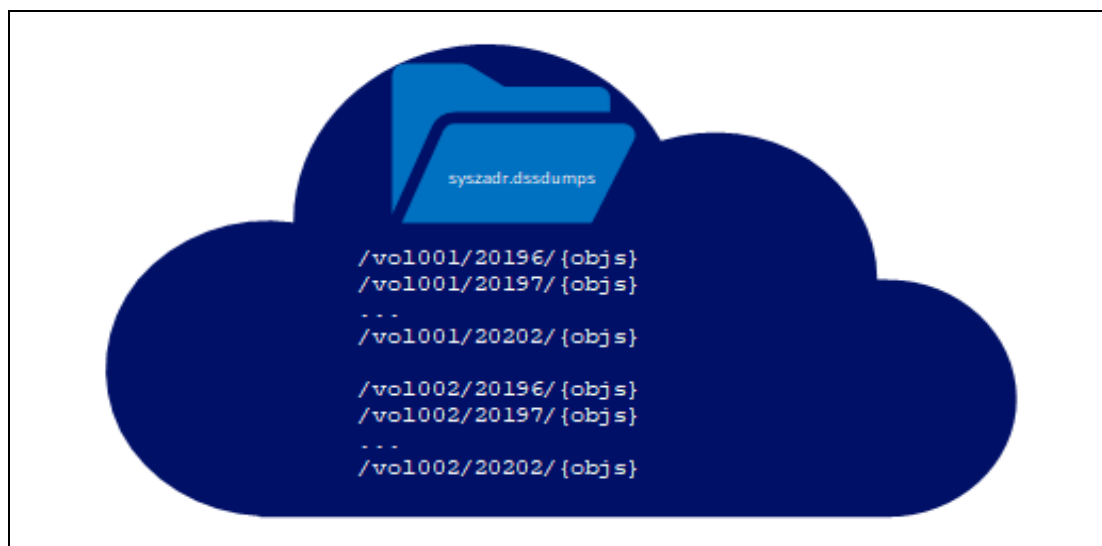


Figure 13-4 Content of SYSZADR.DSSDUMP

We create another set of dumps with the prefix 'vo100x/20203', and if those dumps are successful, we delete all of the previous day's created dumps for 'vo100x/20202'.

For that purpose, we create a new version of the FVD. and if the step is successful, we delete the previous version, as shown in Example 13-5 on page 132.

Example 13-5 Creating a new version of the specific FVD and deleting the previous version

```
//STEP0001 EXEC PGM=ADRSSU
//SYSPRINT DD DISP=(NEW,CATLG),DSN=MY.CLOUD.INVENTORY(+1)
//DISK1 DD UNIT=SYSDA,DISP=SHR,VOL=SER=VOL001
//DISK2 DD UNIT=SYSDA,DISP=SHR,VOL=SER=VOL002
//SYSIN DD *,SYMBOLS=(EXECSYS)
PARALLEL
DUMP INDDNAME(DISK1) CLOUD(TS7700CLOUD) -
CONTAINER(DSSDUMPS) CDACREDS -
OBJECTPREFIX('&JOBNAME./VOL001/20203. ')-
WAIT(0,0) ALLDATA(*) ALLEXCP
DUMP INDDNAME(DISK2) CLOUD(TS7700CLOUD) -
CONTAINER(DSSDUMPS) CDACREDS -
OBJECTPREFIX('&JOBNAME./VOL002/20203. ')-
WAIT(0,0) ALLDATA(*) ALLEXCP
SERIAL

IF MAXCC=0 THEN
DO
CLOUDUTILS DELETE CLOUD(TS7700CLOUD) -
CONTAINER(DSSDUMPS) CDACREDS -
OBJECTPREFIX('&JOBNAME./VOL001/20202')-
CLOUDUTILS DELETE CLOUD(TS7700CLOUD) -
CONTAINER(DSSDUMPS) CDACREDS -
OBJECTPREFIX('&JOBNAME./VOL002/20202')-
END
```

This use case is one specific use case that can be adapted to meet your requirements and it can be improved and automated by implementing a script that prepares the JCL needed. You can cover different cases when to delete previous created FVD, depending on your policy on how long you want to keep FVD backups available



DFSMSHsm enhancements for Transparent Cloud Tiering

In this appendix, we provide methods to improve the DFSMSHsm Transparent Cloud Tiering (TCT) migration to cloud experience. Before implementing any of the changes that are documented in this appendix, ensure that the latest TCT maintenance level is installed. You find the maintenance level by going to [New Function For Transparent Cloud Tiering](#) and searching for the following FIXCAT categories:

- ▶ IBM.Function.DFSMSCloudStorage
- ▶ DFSMSCS/K

Note: Thanks to Jeannie Vangsness and Glenn Wilcock for their contribution to this appendix.

This appendix includes the following topics:

- ▶ “Large number of data sets” on page 134
- ▶ “Performance improvements” on page 136
- ▶ “Other considerations” on page 137

Large number of data sets

With [APAR OA58915](#), you can increase the maximum number of large data sets that can be migrated to the cloud concurrently. The feature is available in base z/OS V2R5. PTFs for OA58915 are available for V2R4 and V2R3.

- ▶ A limited number of large data sets can be migrated to the cloud concurrently.

If this number is exceeded, large data sets are skipped. These large data sets are revisited later when the large data sets in the process complete. When only large data sets remain to process, DFSMSHsm waits a defined interval before the next scan.

The following **PATCH** command can be used to change this interval:

```
PATCH .MGCB.+152 X'nnnn'
```

nnnn is the time in seconds.

The default value is X'001E' (30 seconds).

- ▶ Set the interval for issuing the Disk controller busy, ARC1587I message.

An ARC1587I message indicates that all disk controller cloud data movement threads are busy for a specified number of seconds. The following **PATCH** command can be used to change the interval in which the message is issued:

```
PATCH .MGCB.+164 X'nnnn'
```

nnnn is the time in seconds.

If this value is set to X'7FFF', no interval checking is done, and no ARC1587 messages are issued. The default value is X'7FFF'.

- ▶ Setting the maximum wait time in seconds to get a disk controller for moving data to the cloud.

You can set the maximum wait time that Hierarchical Storage Manager (HSM) waits to get a disk controller cloud data movement thread lock. HSM Space Management for a volume terminates when a disk controller cannot be reserved within this time. When the wait time is exceeded, a ARC0535I message is issued, and volume processing is terminated.

The following **PATCH** command can be used to tune the wait time:

```
PATCH .MGCB.+9A X'nnnn'
```

nnnn is the time in seconds.

If this value is set to X'7FFF', no interval checking is done, and Space Management of a volume is not terminated. The default value is X'7FFF'.

- ▶ Setting the minimum size of a large data set.

A data set is considered “large” when it is larger than this value in the management communication vector table (MCVT). The following **PATCH** command can be used to tune this value:

```
PATCH .MCVT.+5AC X'nnnnnnnn'
```

nnnnnnnn is a data set size in tracks.

The default value is X'000186A0' (100.000) tracks.

- ▶ Setting the minimum number of small data set threads for moving data to the cloud.

There is a limited number of DS8000 threads for data movement. By default, DFSMSHsm reserves one thread per DS8000 internal server to process small data sets to avoid the situation where large data sets processing consumes all available threads and no small data sets are migrated.

If you know that only large data sets will be processed, you can make this reserved thread available for processing large data sets too. You can use the following **PATCH** command to tune the number of threads that are reserved for small data sets:

```
PATCH .MCVT.+3CE X'nnnn'
```

nnnn is the number of threads that is reserved for small data sets.

The default value is X'0001'.

A large data set is one that exceeds the default value of 100,000 tracks or the customer-defined value in the MCVT+5AC. A **PATCH** value of '0' enables DFSMSHsm to use all DS8000 threads for processing large data sets without concern for blocking out small data set processing.

- ▶ Allowing DFSMSHsm to trace threads checking and reserving during migration to cloud storage.

By default, threads checking and reserving traces are not written during migration to cloud storage. To enable this option, run the following **PATCH** command:

```
PATCH .MCVT.+24F BITS(...1....)
```

This processing can be reversed or disabled by running the following **PATCH** command:

```
PATCH .MCVT.+24F BITS(...0....)
```

- ▶ Defining the **MIGRATE STORAGEGROUP** command behavior.

The **MIGRATE STORAGEGROUP** command migrates the eligible data sets on the volumes in one or more storage groups. When the **MIGRATE STORAGEGROUP** command is issued, DFSMSHsm reads the SMS storage group definition and determines the list of volumes that are defined by this storage group. Next, the data sets on these volumes are processed in a similar way to automatic migration processing.

By default, the main differences are as follows:

- Volumes can be processed several times a day.
- In a multi-host environment, the command can go into a wait state for up to 5 minutes if the volume that is required for processing is reserved by another host.

The following **PATCH** command can be applied to provide **MIGRATE STORAGEGROUP** command behavior that is fully consistent with automatic migration:

```
PATCH .MGCB.+113 BITS(..1....)
```

This processing can be reversed or disabled by issuing the following **PATCH** command:

```
PATCH .MGCB.+113 BITS(..0....)
```

- ▶ Allowing DFSMSHsm automatic functions or the **MIGRATE STORAGEGROUP** command to process volumes more than once per day.

If the **MIGRATE STORAGEGROUP** command behavior is fully consistent with automatic migration (the **PATCH** command above is applied) and the **DAYS(0)** or **MOVE** parameter is not specified, the volumes can be processed only once every 14 hours. In this case, the tuning of the **MIGRATE STORAGEGROUP** command processing is the same as automatic primary space management of SMS volumes.

MIGRATE STORAGEGROUP with the **DAYS(0)** or **MOVE** parameter can process the volumes once every 3 hours and 59 minutes. You can change this limit by using the following **PATCH** command:

```
PATCH .MCVT.+5B0 X'nnnnnnnn'
```

nnnnnnnn is the minimum time in seconds between **MIGRATE STORAGEGROUP** commands with the **DAYS(0)** or **MOVE** parameter.

- ▶ Setting the wait Interval before the next Secure Sockets Layer (SSL) check is run during migration to cloud storage when multiple migration subtasks are enabled.

The following **PATCH** command can be used to change the interval in seconds before the next migration subtask starts to migrate the first data set to cloud storage:

```
PATCH .MGCB.+166 X'nnnn'
```

nnnn is the time in seconds.

If this value is set to X'0000', the next migration subtask does not wait before data set processing. The default value is X'0000'.

If for example, you set this value to 60 seconds, Subtask1 starts immediately, Subtask2 starts 60 seconds after the first, Subtask3 starts 120 seconds after the first, Subtask4 starts 180 seconds after the first, and so on.

Performance improvements

This section describes the following topics:

- ▶ Asynchronous object deletion
- ▶ Additional performance improvements

Asynchronous object deletion

With [APAR OA59765](#), DFSMSshsm is enhanced to modify automatic migration functions to delete old migration copies from cloud storage asynchronously. This new function reduces the time that it takes to migrate data sets to the cloud storage.

To enable this option, apply the following **PATCH** command:

```
PATCH .MGCB.+113 BITS(...1....)
```

This processing can be reversed or disabled by issuing the following **PATCH** command:

```
PATCH .MGCB.+113 BITS(...0....)
```

The APAR is available in base z/OS V2R5. PTFs are available for V2R4 and V2R3.

Additional performance improvements

To improve DFSMSshsm performance with TCT, consider the following measures:

- ▶ Enable migration subtasking by running the following **SETSYS** command:
SETSYS MIGRATIONSUBTASKS(YES)
- ▶ With DS8880 8.5 or higher or DS8900, increase the number of concurrent DS8000 cloud movement threads per DS8000 internal server from 6 to 12 with the following **PATCH** command:

```
PATCH .MCVT.+494 X'000C'
```

- ▶ Change the container creation cycle to 92 days if it is still set to 7 in your environment. Run the following **PATCH** command:

```
PATCH .ARCCVT.+50F X'5C'
```

The default was changed from 7 to 92 with APAR OA60278.

- ▶ Set a unique plex name per HSMplex with the following **SETSYS** command:
SETSYS PLEXNAME(plexname)
- ▶ Separate cloud work with the following **SETSYS** command:
SETSYS MIGRATIONAUTOCLLOUD(NOCLOUD | CLOUDONLY)
- ▶ Testing migration to cloud
The **MIGRATE DSN** command is a single threaded command. It is not a proficient way of testing migration to cloud. Instead, use the **MIGRATE STORAGEGROUP DAYS(0)** or **MOVE** parameter so that the volumes are distributed equally across the DS8000 internal servers (central processor complexes (CPCs)).

Other considerations

When you migrate data sets to the cloud, consider the following items:

- ▶ Large data sets take a long time to go to the cloud. Be patient.
- ▶ Be sure to know your average throughput going to your cloud provider.
- ▶ If DFSMShsm seems stalled or not progressing, make sure that you set the **ENABLE(AOM496I)** parameter in the **DEVSUPXX** parmlib member set. The **AOM496I** status message goes to the console for TCT operations. It is disabled by default. For more information, see [z/OS MVS Initialization and Tuning Reference](#).
- ▶ When a user cancels a “stalled” HSM task, this action cancels the task in the z/OS host, but it does not cancel the DS8000 thread moving the data set to the cloud.
- ▶ The new patches are documented in “Tuning patches supported by DFSMShsm” in [z/OS 2.5 DFSMShsm Implementation and Customization Guide](#).
- ▶ Always run with the latest maintenance levels, which can be found by going to [New Function For Transparent Cloud Tiering](#) and searching for the following FIXCAT categories:
 - IBM.Function.DFSMSCloudStorage
 - DFSMSCS/K



Exporting the IBM DS8000 certificate chain

You might have to export the DS8000 security certificates so that they are available for import to z/OS and IBM Resource Access Control Facility (RACF) when you set up secure communication between Data Facility Storage Management Subsystem (DFSMS) and the DS8000 Hardware Management Console (HMC) acting as a cloud proxy.

This appendix explains how to export the DS8000 security certificate chain by using two different web browsers. Other methods exist and can be used.

This appendix includes the following topics:

- ▶ “Certificate export with Google Chrome or Microsoft Edge” on page 140
- ▶ “Certificate export with Firefox” on page 142

Certificate export with Google Chrome or Microsoft Edge

If you use Microsoft Edge as your web browser, you can export the required certificates by completing the following steps.

Note: The process to export certificates to files by using the Google Chrome web browser is like the steps that are described here.

1. Open the browser and connect to the DS8000 HMC from which you want to export the certificates.
2. When you see the login window of the DS8000, click the lock icon to the left of the URL field.

Note: In our example, we use self-signed certificates in the DS8000 HMC and the window displays a warning saying “Insecure” instead of the lock icon. Click this warning to continue as described and as shown in Figure B-1. The procedure to export the certificates is the same.

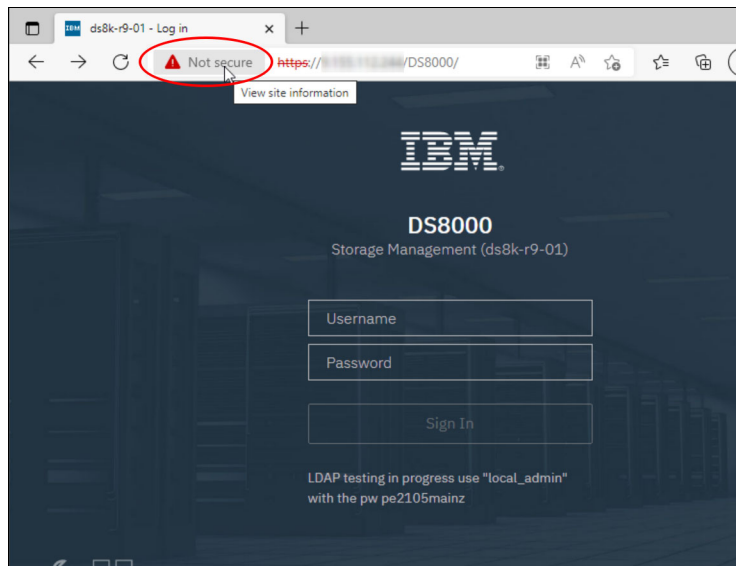


Figure B-1 DS8000 login window by using Microsoft Edge

3. In the website information window, click the security information section, as shown in Figure B-2.

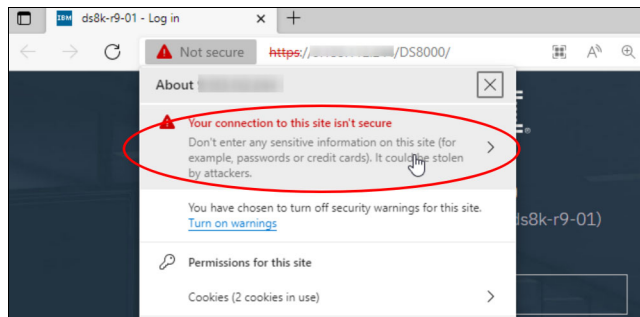


Figure B-2 Microsoft Edge website information window

4. In the Security information window, click the **Show Certificate** icon, as shown in Figure B-3.

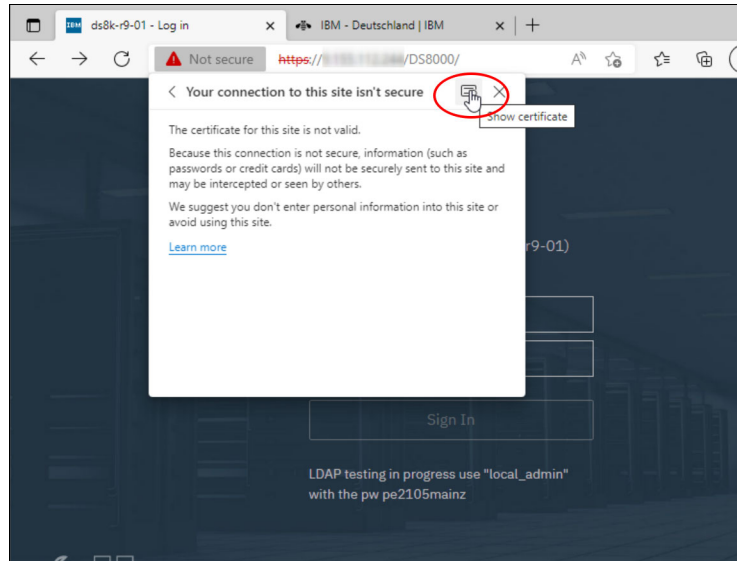


Figure B-3 Security information window in Microsoft Edge

5. In the Certificate Viewer window, select the **Details** tab to see the certificate details, as shown in Figure B-4.

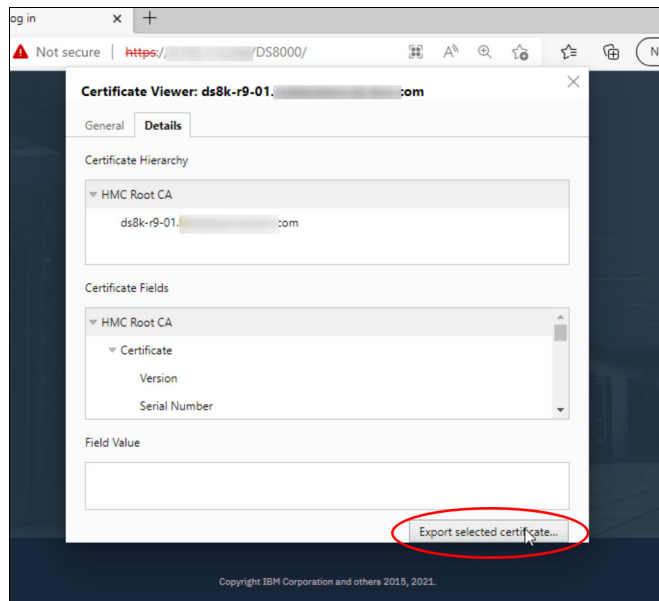


Figure B-4 Certificate details in the Certificate Viewer window in Microsoft Edge

6. Select the certificate that you want to export and click **Export selected certificate** to save the certificate to a file.
7. Repeat step 6 for all certificates that you need.

Certificate export with Firefox

If you use the Firefox web browser, you can export the required certificates by completing the following steps:

1. Open the Firefox browser and connect to the DS8000 HMC from which you want to export the certificates.
2. When the browser connects and you see the login window of the DS8000, click the lock icon to the left of the URL field, as shown in Figure B-5.

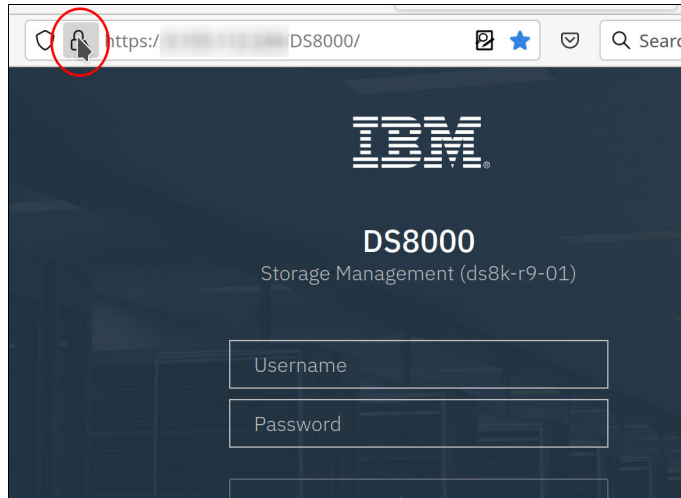


Figure B-5 DS8000 login window by using Firefox

3. In the website information window, click the security information section, as shown in Figure B-6.

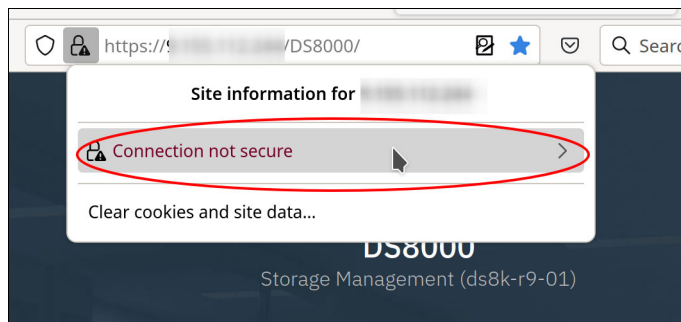


Figure B-6 Firefox website information window

4. In the Security information window, click **More information**, as shown in Figure B-7 on page 143.

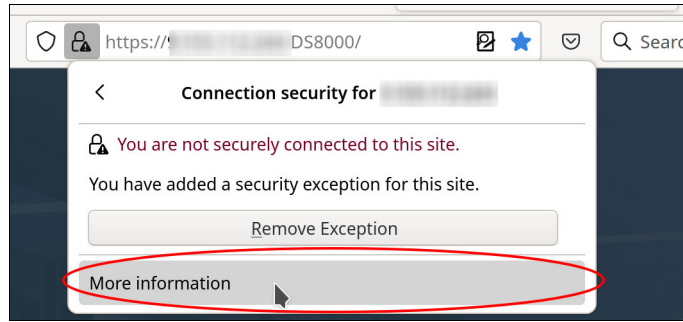


Figure B-7 Firefox security information window

5. The window that opens shows the security details for the website. Click **View Certificates**, as shown in Figure B-8.

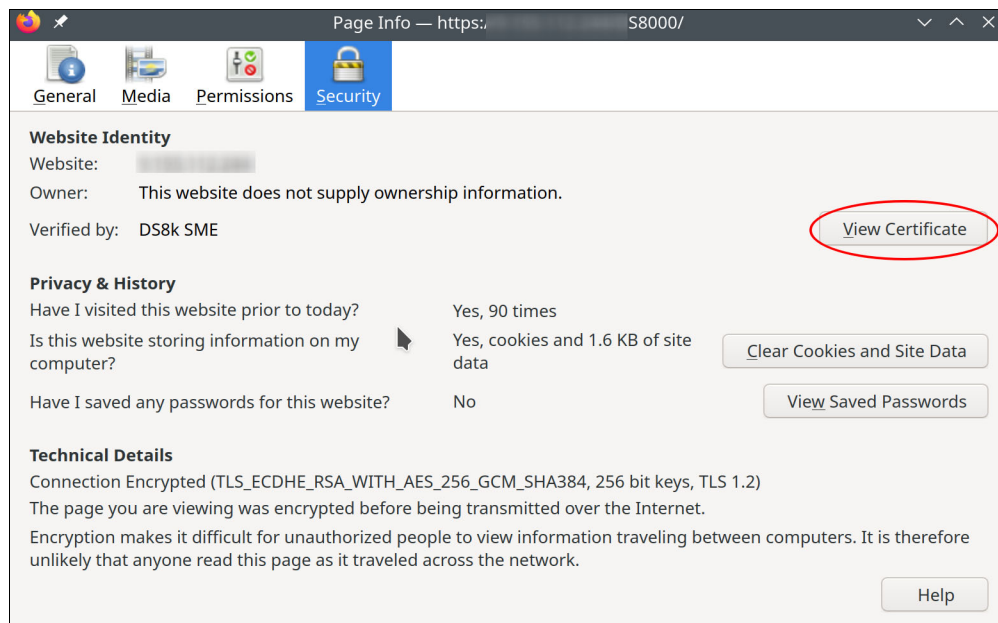


Figure B-8 Firefox website security details

6. Firefox now shows the certification viewer for this website. There is information for each certificate that the website is using in separate tabs. Click the tab for the certificate that you want to export, as shown in Figure B-9.

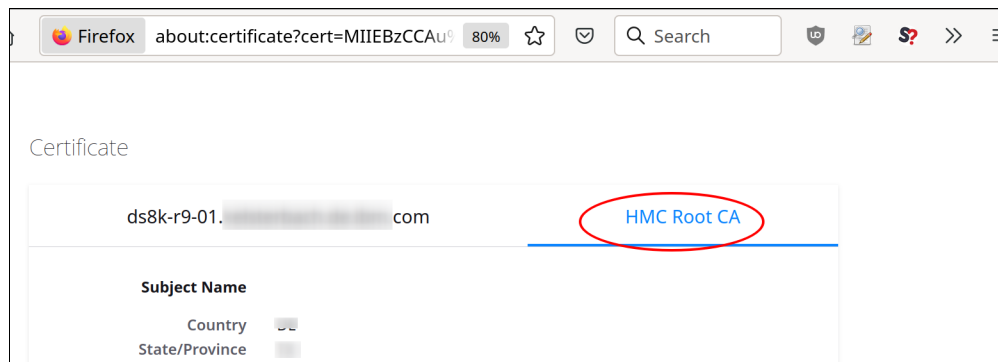


Figure B-9 Firefox certificate viewer

7. Scroll down until you see the Miscellaneous section. It contains a link to export the selected certificate. Because the exported files are in Privacy-Enhanced Mail (PEM) format, you must place each certificate into a separate file. Therefore, use the **PEM(cert)** link, as shown in Figure B-10, and not the one for the entire CA chain, even though it might be available.

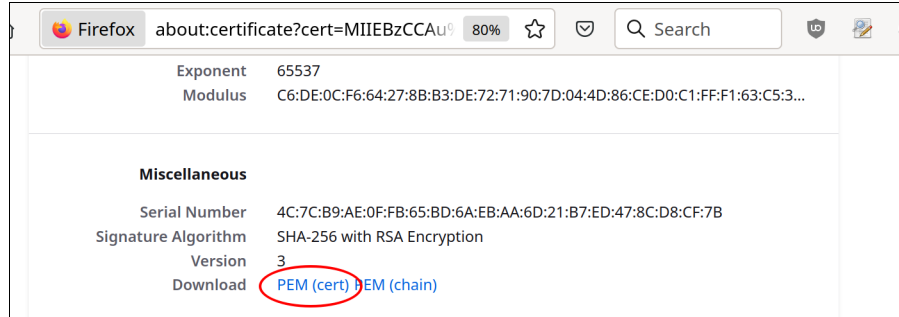


Figure B-10 Exporting a certificate in Firefox

8. Repeat step 7 for all certificates that you need.



Replacing communication certificates in the IBM DS8900F

In this appendix, we describe how you can exchange the default self-signed security certificates with either another self-signed certificate, or a certificate that is signed by a CA of your choice.

We explain how to install CA-signed certificates by using the DS8000 Storage Manager GUI or the DS8000 Service Web User Interface.

Furthermore, we describe how to generate a *Certificate Signing Request* (CSR) and import the certificate that is issued by the CA.

Important: Installing a certificate requires a Hardware Management Console (HMC) restart, which might cause a short interruption or alerts in your management environment, for example, if you are using the *IBM Copy Services Manager*.

Attention: Replacing the existing certificate or CA chain of a DS8000 HMC might impact other components or systems that communicate with this HMC. You might need to update their secure communication setup (truststores and trusted certificates) to continue communication.

This appendix includes the following topics:

- ▶ “Installing a CA-signed certificate by using the Storage Manager GUI” on page 146
- ▶ “Installing a CA-signed certificate by using the service web interface” on page 149
- ▶ “Creating a CSR” on page 151
- ▶ “Importing the signed certificate” on page 152
- ▶ “Creating a self-signed certificate on the DS8000” on page 153

Installing a CA-signed certificate by using the Storage Manager GUI

In this section, we provide a way to install a CA-signed certificate with various pre-set certificate information settings. If you need more advanced certificate settings, see “Installing a CA-signed certificate by using the service web interface” on page 149.

In this section, we explain how to perform the following actions:

- ▶ Navigating to the Communication Certificates window
- ▶ Creating a certificate signing request
- ▶ Importing the signed certificate

Navigating to the Communication Certificates window

You can install a CA-signed certificate by using the Storage Manager GUI while logged in with a user ID as an admin role.

To change the certificate settings, open the Communications Certificate window by selecting **Settings** → **Security** → **Communications Certificate**, as shown in Figure C-1. In this window, you can change the certificate for both HMCs, HMC1, and HMC2.

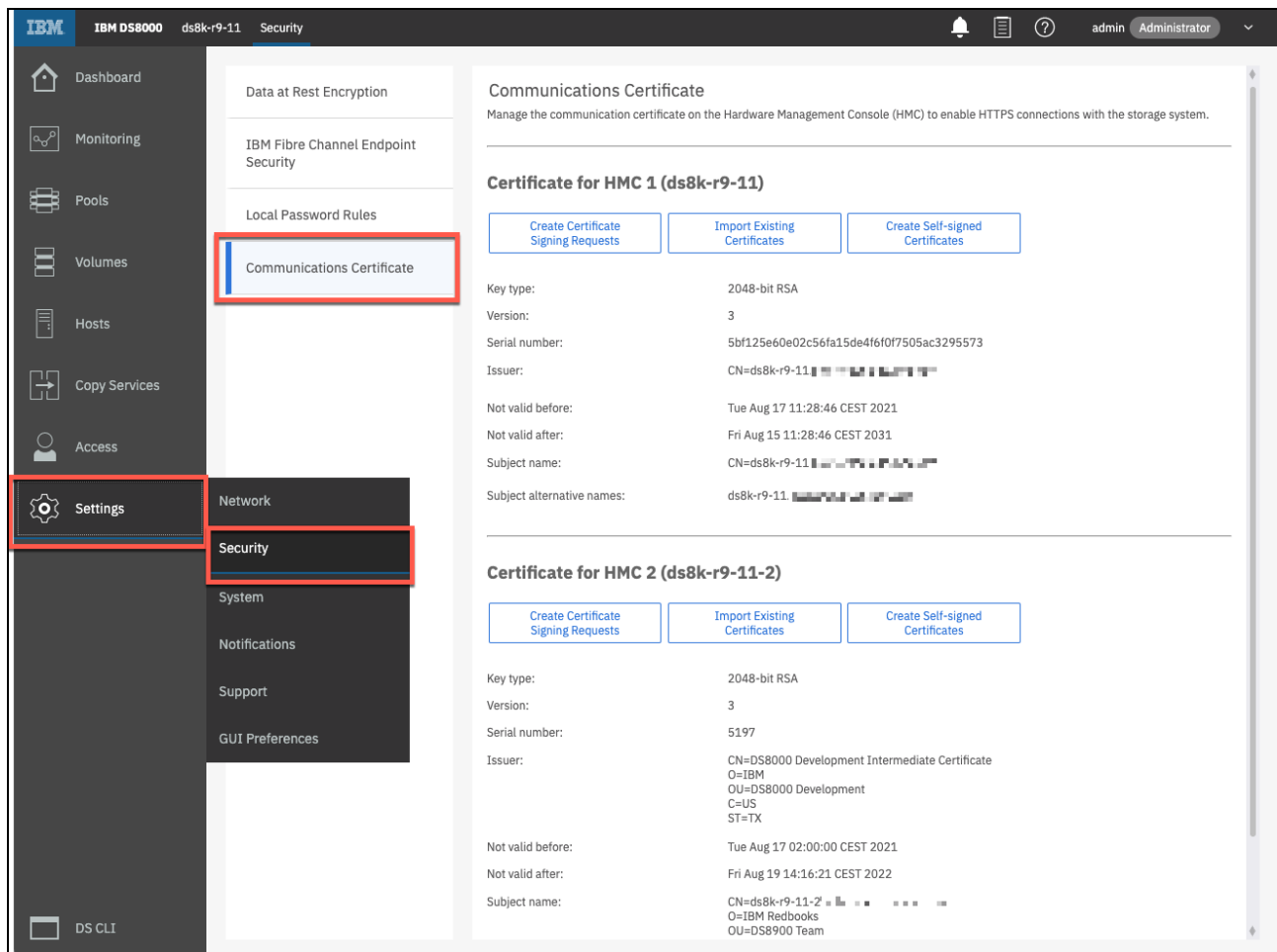


Figure C-1 Communication Certificate window in the DS8900 Storage Manager GUI

Creating a certificate signing request

To create a certificate signing request, complete the following steps:

1. From the Communication Certificate window that is shown in Figure C-1 on page 146, you can create a CSR. When you click **Create Certificate Signing Requests**, the window that is shown in Figure C-2 opens. You must complete the certificate-related information.

The screenshot shows a window titled "Create Certificate Signing Request" with a close button (X) in the top right corner. Below the title bar, there is a small icon and the text "HMC 1 (ds8k-r9-11)". A horizontal line separates the header from the main content area. The main content area contains the following text and form fields:

- Enter information to generate and download a certificate signing request for the Hardware Management Console (HMC 1 ds8k-r9-11).
- Key type: 2048-bit RSA
- HMC DNS hostname:
- Organization:
- Organization unit:
- Country:
- State or province:
- City or locality:
- Number of days until expiration:
- Email address:

At the bottom right of the window, there are two buttons: "Create" and "Cancel".

Figure C-2 DS8900 Storage Manager GUI Certificate Signing Request window

2. After you complete the details for the signing request and click **Create**, a File-Save dialog box opens. You can download the CSR.
3. Send the CSR file to the CA for signing. When placing the signing request with your CA, request that they return the certificate in Base64 (ASCII) format because the DS8900 accepts only this format when importing the signed certificate.

Importing the signed certificate

After the CA returns the signed certificate, you can import it to the HMC by completing the following steps:

1. Go to the Communication Certificates window (as shown in Figure C-1 on page 146) and click **Import Existing Certificates**.
2. A window opens where you specify the certificate file, as shown in Figure C-3. Click **Import**.

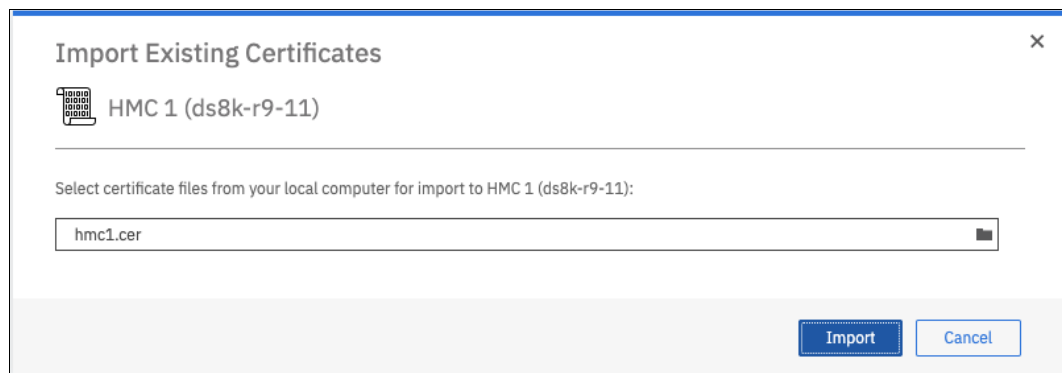


Figure C-3 DS8900 Storage Manager GUI: Import Existing Certificates

3. After you specify the certificate file, another window opens, as shown in Figure C-4. It informs that all connected users will be logged off and that the HMC will restart. Click **Yes**.

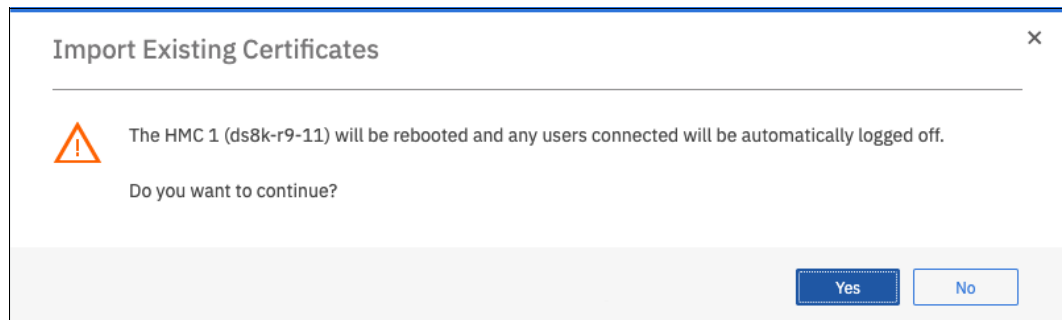


Figure C-4 DS8900 Storage Manager GUI: Confirmation for the HMC restart

The HMC restarts and uses the newly installed certificate during the restart of the communication services for the DS Storage Manager GUI, IBM Copy Services Manager, and the DS8000 Representational State Transfer (REST) API.

Installing a CA-signed certificate by using the service web interface

If you need more advanced certificate settings, you can follow the procedure that is described in this section by using the DS8000 Service Web User Interface (WUI).

Navigating to the service WUI

To go to the service WUI, complete the following steps:

1. On the DS8000 Storage Manager GUI login window, click the wrench symbol in the lower left, as shown in Figure C-5.

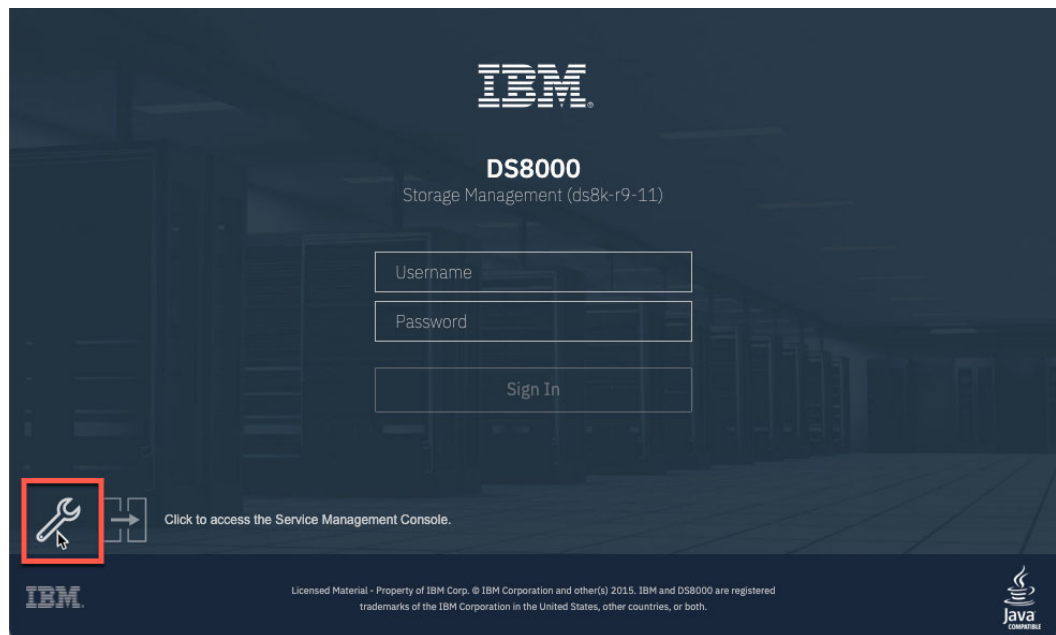


Figure C-5 DS8900 login window

2. You are diverted to home window of the Service interface. Click **Log on and launch the Hardware Management Console web application**. Log in with the user ID customer.

Note: The default password for the user ID customer is cust0mer. You must change this password the first time that you use this ID to log in.

- Once you are authenticated, go the left pane and click **HMC Management**. In the HMC Management menu in the right pane, click **Manage Certificates**, as shown in Figure C-6.

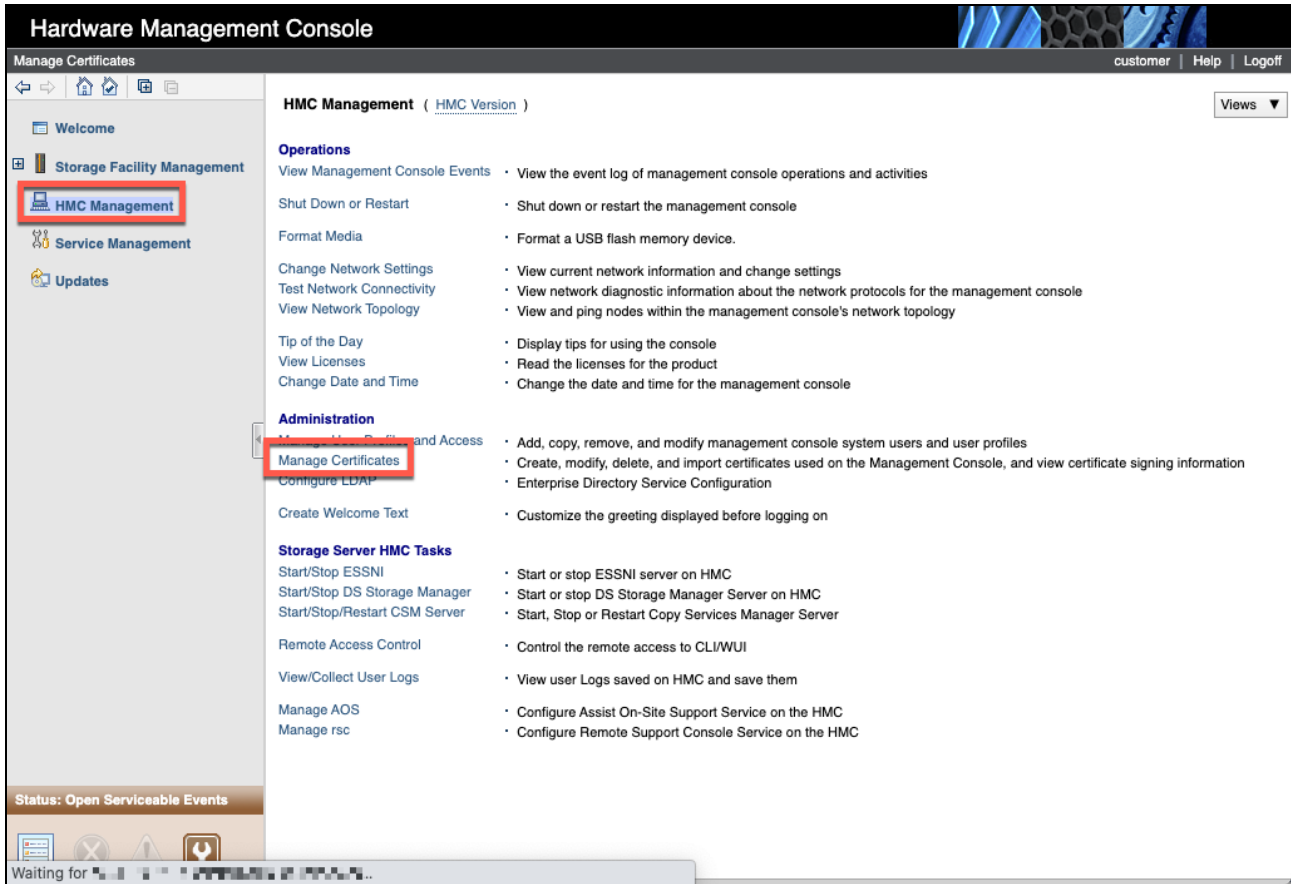


Figure C-6 DS8900 Service WUI: HMC Management window

The Certificates Management window shows the details of the certificate, as shown in Figure C-7.

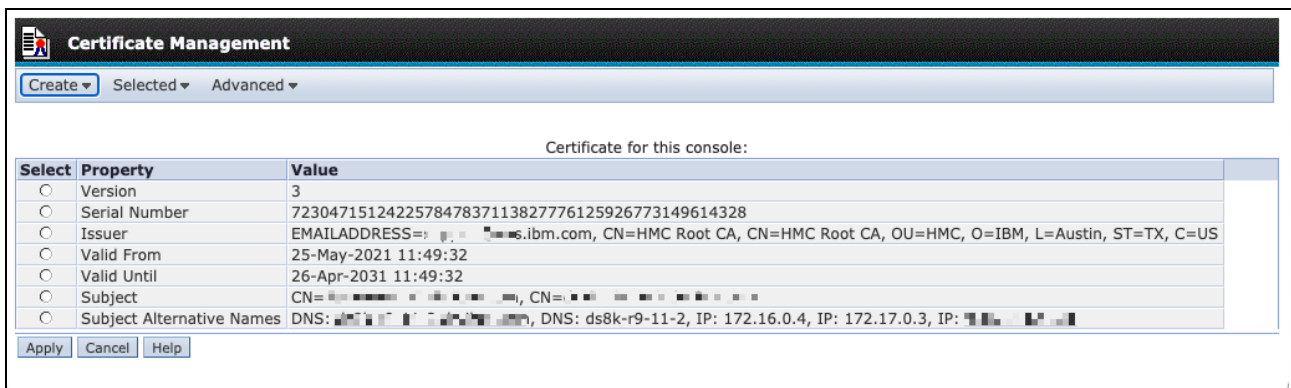


Figure C-7 DS8900 Service WUI Certificate Management

Creating a CSR

If you want to replace the certificate with a certificate that was signed by your internal CA, you must create a CSR.

1. In the Certificate Management window, select **Create** → **New Certificate**. Then, select **Signed by a CA** and click **OK**.
2. Enter the required values into the New Certificate dialog box, as shown in Figure C-8.

New Certificate

Enter the following information for the certificate signing request to be created:

Common Name * ds8k-r9-11-2

Organization (e.g. IBM) IBM Redbooks

Organization unit (e.g. Hardware Development) DS8900 Team

Two letter country or region code (e.g. US) DE - Germany

State or Province (e.g. CA) HE

Locality (e.g. Los Angeles) Frankfurt

Number of days until expiration (e.g. 365) * 3650

E-mail address (e.g. xxxx@ibm.com) xxx@ibm.com

IP Address

172.16.0.4 Add Remove

172.17.0.3

DNS

ds8k-r9-11-2 Add Remove

ds8k-r9-11-2

OK Cancel Help

Figure C-8 Creating a Certificate Signing Request

3. After clicking **OK**, you are asked to store the signing request. Select **The filesystem on the system running the browser**, as shown in Figure C-9.

Question

Will the certificate signing request be saved to removable media on the console or to the file system on the system running the web browser?

ACT05111

Removable media on the console The file system on the system running the browser

Figure C-9 Selecting the destination to save the CSR

4. In the following window, you click the link text **Certificate Signing Request** to save the CSR file to your local workstation. After you download the file, continue by clicking **OK**. Send the generated CSR file to your CA for signing.
5. You are asked whether you want to continue with a newly generated self-signed certificate until you get the signed certificate back from the CA. You can continue with the currently installed certificate.

Importing the signed certificate

After you received the signed certificate back from your CA, complete the following steps:

1. Go to the **Certificate Management** window, as shown in “Navigating to the service WUI” on page 149. From there, select **Advanced** → **Import Server Certificate**.
2. Click **The filesystem on the system running the browser**. In the next dialog box, open the file selection dialog box by clicking **Choose File**, as shown in Figure C-10. Select the signed certificate that you received from your CA.

Note: The certificate files must be in Base64 (ASCII) format. Otherwise, the import fails.

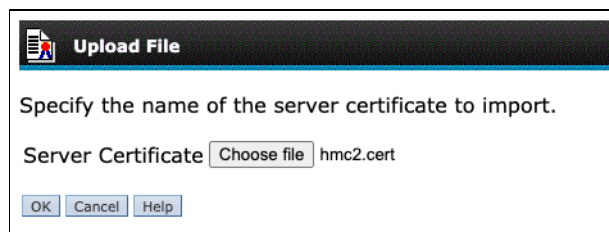


Figure C-10 Uploading the certificate file

3. If you have more files, for example, certificates from any intermediate CA and the root CA, you can import them. Click **Yes** in the dialog box. Otherwise, click **No** and the dialog box is skipped. In the dialog box that is shown in Figure C-11, you can select and upload multiple additional certificates that are required as part of the servers certificate chain.

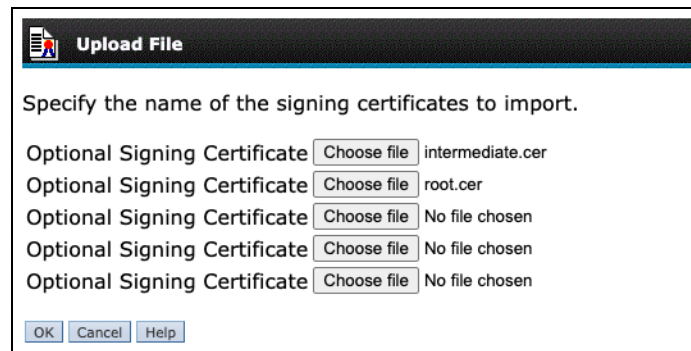


Figure C-11 Uploading more signing certificate files

After you upload all the certificates, the HMC indicates that all certificates were successfully imported. The HMC requires a restart to activate the newly installed certificates.

Creating a self-signed certificate on the DS8000

To create a self-signed certificate on the DS8000, you can use the DS Storage Manager GUI to complete the following steps:

1. Log on to the DS Storage Manager GUI.
2. Select **Settings** → **Security**.
3. Click the **Communications Certificate** tab.
4. Click **Create Self-signed Certificates**.
5. Enter the information that is requested about your organization.
6. Click **Create**. A warning message appears and states that the HMC will restart and any users that are connected will be automatically logged off.
7. Click **Yes** to continue with certificate creation. After creation, the certificate is automatically loaded at the HMC.

After creating a self-signed certificate, the HMC restarts to activate the new certificate.

Abbreviations and acronyms

ACL	Access Control List	MES	miscellaneous equipment specification
ACS	Automatic Class Selection	ML1	Migration Level 1
CCA	IBM Common Cryptographic Architecture	ML2	Migration Level 2
CDA	Cloud Data Access	PDS	partitioned data set
CDS	Control Data Set	PEM	Privacy Enhanced Mail
CKDS	Cryptographic Key Data Set	PiT	point in time
CPACF	CP Assist for Cryptographic Functions	PPRC	Peer to Peer Remote Copy
CPC	central processor complex	RACF	IBM Resource Access Control Facility
CPU	central processing unit	REST	Representational State Transfer
CSR	Certificate Signing Request	SAF	System Authorization Facility
CVQ	common recover queue	SCDS	Source Control Data Set
DFSMS	Data Facility Storage Management Subsystem	SDSP	small data set packing
DR	disaster recovery	SOW	statement of work
DSCLI	DS8000 Command-line Interface	SSD	solid-state disk
DSS	Data Storage Services	SSL	Secure Sockets Layer
FC	Feature Code	TCN	tape copy needed
FVD	full volume dump	TCT	Transparent Cloud Tiering
GDG	Generation Data Group	TLS	Transport Layer Security
GID	group ID	URI	Uniform Resource Identifier
GKLM	Guardium Key Lifecycle Manager	VE	Virtualization Engine
HA	high availability	VSAM	Virtual Storage Access Method
HMC	Hardware Management Console	VTOC	volume table of contents
HSM	Hierarchical Storage Manager	WETK	Web Enablement Toolkit
IBM	International Business Machines Corporation	WUI	Web User Interface
ICSF	IBM Integrated Cryptographic Service Facility		
IPL	initial program load		
ISMF	Interactive Storage Management Facility		
KMIP	Key Management Interoperability Protocol		
KSDS	key-sequenced data set		
LIC	Licensed Internal Code		
LSS	logical subsystem		
MCD	Migration Control Data Set		
MCVT	management communication vector table		
MD	migration exit		

Related publications

The publications that are listed in this section are considered suitable for a more detailed description of the topics that are covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topics in this document. Some publications that are referenced in this list might be available in softcopy only.

- ▶ *IBM DS8900F Architecture and Implementation: Updated for Release 9.3*, SG24-8456
- ▶ *IBM TS7700 Release 5.2.2 Guide*, SG24-8464
- ▶ *IBM TS7700 Series DS8000 Object Store User's Guide Version 2.0*, REDP-5583

You can search for, view, download, or order these documents and other Redbooks, Redpapers, web docs, drafts, and additional materials at the following website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- ▶ IBM DS8900 documentation:
<https://www.ibm.com/docs/en/ds8900>
- ▶ IBM TS7700 documentation:
<https://www.ibm.com/docs/en/ts7700-virtual-tape>
- ▶ IBM z/OS documentation:
<https://www.ibm.com/docs/en/zos>

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Redbooks

IBM DS8000 Transparent Cloud Tiering: DS8000 Release 9.3

SG24-8381-05
ISBN 0738460915



(0.5" spine)
0.475" <-> 0.873"
250 <-> 459 pages

Redbooks

IBM DS8000 Transparent Cloud Tiering: DS8000 Release 9.3

(0.2" spine)
0.17" <-> 0.473"
90 <-> 249 pages



SG24-8381-05

ISBN 0738460915

Printed in U.S.A.

Get connected

