Redbooks
ibm.com/redbooks

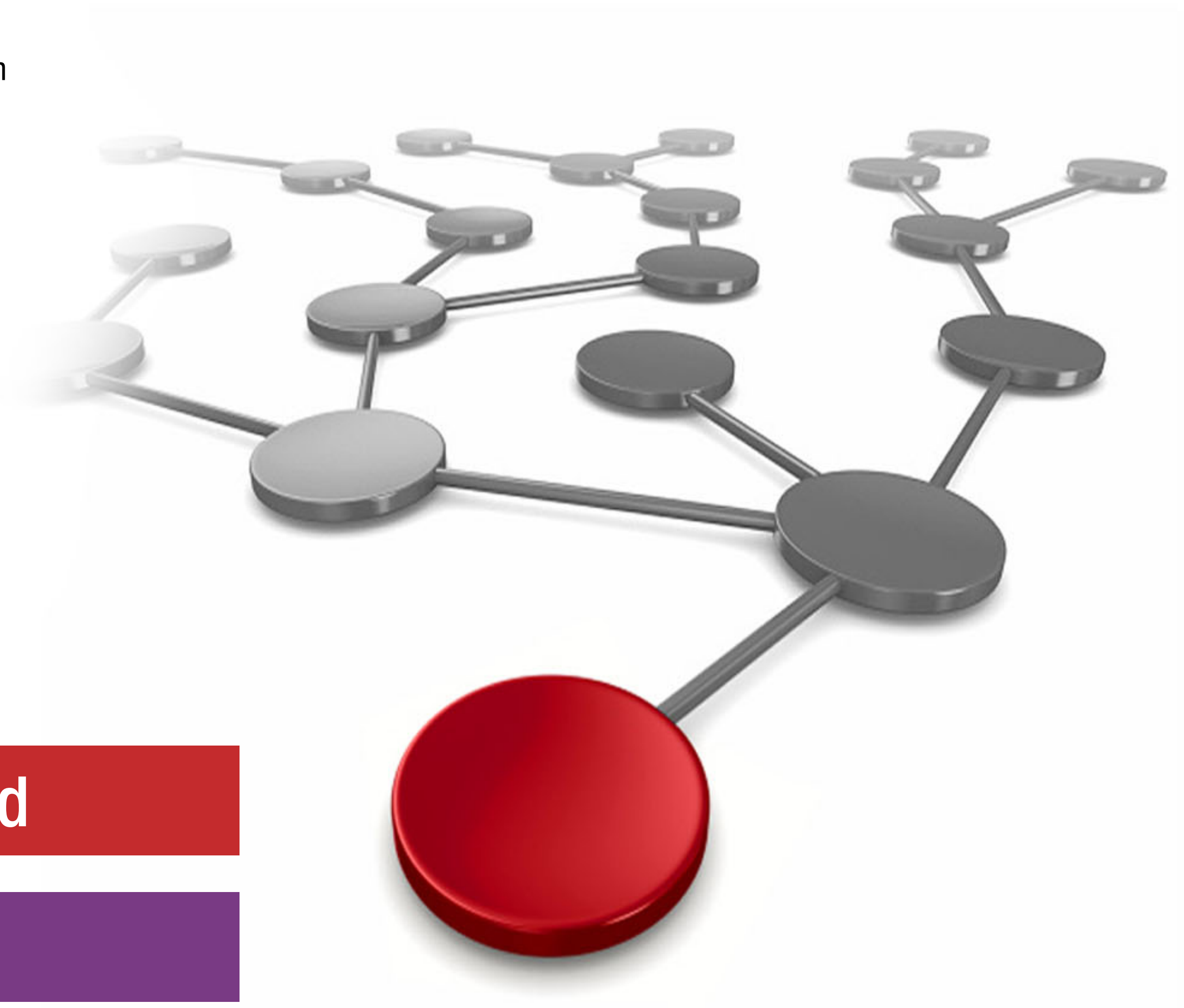
IBM DS8000 and Transparent Cloud Tiering

Alexander Warmuth

Bertrand Dufrasne

Eddie Lin

Andreas Reinhardt



Cloud

Storage

IBM

Redbooks

International Technical Support Organization

IBM DS8000 and Transparent Cloud Tiering

March 2020

SG24-8381-02

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

Third Edition (March 2020)

This edition applies to IBM DS8000 with Licensed Machine Code (LMC) 7.9.0 (bundle version 89.0), referred to as Release 9.0, or later.

Contents

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 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®
DB2®
DS8000®
Easy Tier®
FICON®
FlashCopy®
GDPS®
Global Technology Services®
HyperSwap®
IBM®
IBM Cloud™
IBM Z®
IBM z Systems®
POWER®
RACF®
Redbooks®
Redbooks (logo) ®
System Storage™
z Systems®
z/OS®

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.

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 functionality that was introduced for mainframes to have.

IBM Data Facility Storage Management Subsystem (DFSMS) and the IBM DS8000® added functionality 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* 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 functionality.
- 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 if it were cloud storage from the DS8000 perspective.

This IBM Redbooks publication is divided into the following parts:

- Part 1 provides you with an introduction to clouds. It provides basic knowledge and terminology.
- Part 2 shows you how we set up the Transparent Cloud Tiering in a controlled laboratory and how the new functions work. We provide points to consider to help you set up your storage cloud and integrate it into your operational environment.
- Part 3 shows you how we used the new functionality to communicate with the cloud and to send data to it and retrieve data from it.

Authors

This book was produced by a team of specialists from around the world working at the IBM Redbooks International Technical Support Organization (ITSO), San Jose Center.

Alexander Warmuth is a Consulting IT Specialist in IBM's 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 z Systems® and Linux. He joined IBM in 1993. Alexander holds a diploma in Electrical Engineering from the University of Erlangen, Germany.

Bertrand Dufrasne is an IBM Certified Consulting IT Specialist and Project Leader for IBM System Storage™ disk and flash products at the ITSO, San Jose Center. He has worked at IBM in various IT areas. He has written many IBM Redbooks publications and has developed and taught technical workshops. Before joining the ITSO, he worked for IBM Global Services as an Application Architect. He holds a Master's degree in Electrical Engineering.

Eddie Lin is a Senior Technical Staff Member with IBM Systems Storage located in Tucson, AZ. He is an architect for the IBM DS8000 Enterprise Storage product specializing in developing enablement and solutions in the realm of Cloud Storage and Cloud Computing. Currently Eddie is the lead architect for the DS8000Transparent Cloud Tiering solution. Eddie has over 15 years of experience developing from the original DS8000 to the latest DS8900F line of enterprise storage.

Andreas Reinhardt is an IBM Certified Specialist for high-end disk systems in Kelsterbach, Germany, and has worked in various IT areas at IBM for more than 15 years. Andreas works for IBM Global Technology Services® and started as an IBM System Service Representative (IBM SSR). He is now a member of the DS8000 and FlashSystem PFE teams with a key role in DS8000 fast recovery for DS8000 disk storage systems, FlashSystem, and with the DS8000 encryption implementation team in EMEA.

Thanks to the previous authors of this publication: Jose Gilberto Biondo, Orlando Ariel Fernandez, Robert Gensler, Eddie Lin.

Thanks to the following people for their contributions to this edition:

- Eddie Lin
- Patrick Wolf
- Monica Valeria Falcone

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 ITSO 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:

  ibm.com/redbooks

- Send your comments in an email:

  redbooks@us.ibm.com

- Mail your comments:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for 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

Part 1

Introduction to TCT 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 TCT” on page 3
- Chapter 2, “Cloud Overview” on page 11
- Chapter 3, “Transparent cloud tiering” on page 19

1

Storage Tiering History and the Value of TCT

This chapter, introduces you to storage tiering and to Transparent Cloud Tiering (TCT) and the value it can bring to your data storage needs.

The management of data across tiers is available for you to use across several devices at your site. We included this topic to show you what can be in place and why you might want to use cloud 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 6
- 1.4, “DFSMShsm behavior without TCT” on page 8
- 1.5, “Introducing TCT” on page 9

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 disk (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.

You can configure a DS8000 with a mix of flash cards, SSD, Enterprise, and Nearline volumes with IBM Easy Tier®. This approach enables the system hardware to constantly monitor data extents (or track group, for small extents). Note that recent models, such as some of the DS8880 and the DS8900F, only come as all flash configurations, but can still benefit from Easy Tier.

Easy Tier identifies data temperature based on the number of accesses that are requested within a period, and takes the appropriate actions to maintain efficiency. An extent can become “hot”. In this case, *hot* means that the I/O workload of the extent’s data is higher, compared to other extents in the same extent pool and in the same tier, so it is moved to a higher performance tier. When the data becomes cold, meaning it is being less frequently accessed than other data, it is demoted to slower, cost-efficient volumes.

Figure 1-1 shows a logical volume with hot and cold extents that are allocated on high-performance and cost-efficient hardware.

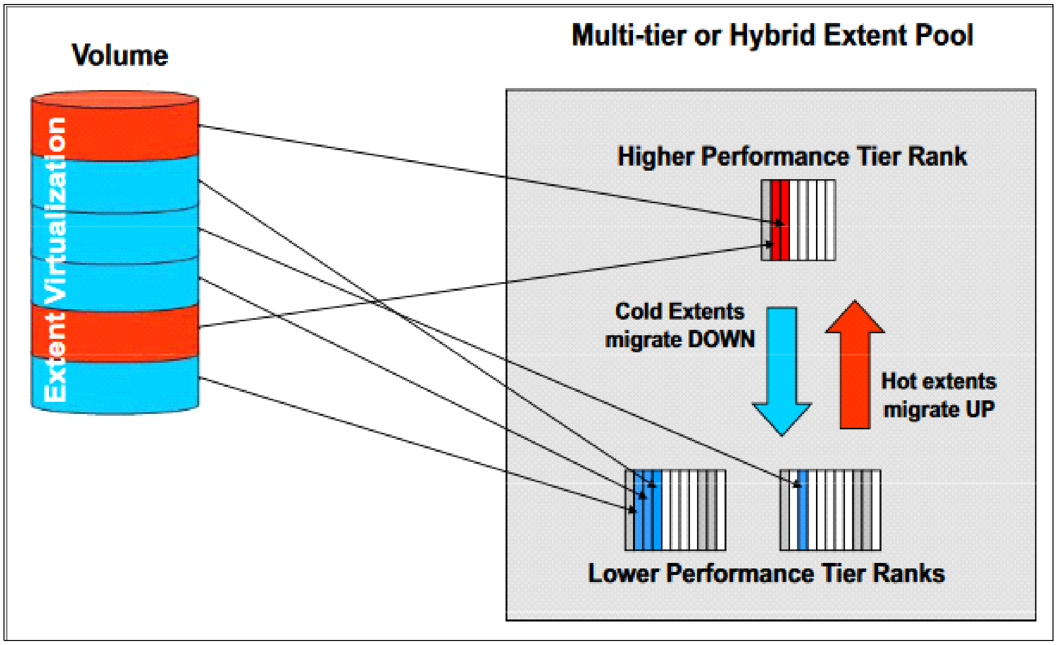

Volume
Extent Virtualization
Multi-tier or Hybrid Extent Pool
Higher Performance Tier Rank
Cold Extents migrate DOWN
Hot extents migrate UP
Lower Performance Tier Ranks

Figure 1-1 Hot and cold extents

IBM Easy Tier monitors data within a single tier and monitors ranks to ensure that one rank is not overloaded while others are idle. This function is called auto-rebalance, or intra-tier rebalancing, and is triggered every 6 hours, if required.

Auto-rebalance migrates extents across ranks within a storage tier to achieve these goals:

- A balanced workload distribution across the ranks and to avoid hotspots.
- Consequently, reduced performance skew within a storage tier and the best available I/O performance from each tier.

The extents are migrated only within the same extent pool. So to take advantage of this feature, at least two ranks must be defined in an extent pool.

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 (DASD) 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

The software layers consist of the following storage areas:

- Level 0
- Migration Level 1
- Migration Level 2

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 DASD 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 DASD 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 DASD 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 DASD for an extended period reduces the ROI on the high-performance DASD 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 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 1-2 shows the sample implementation of class transition. The data is first allocated in high-performance DASD, and then it will transition 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).

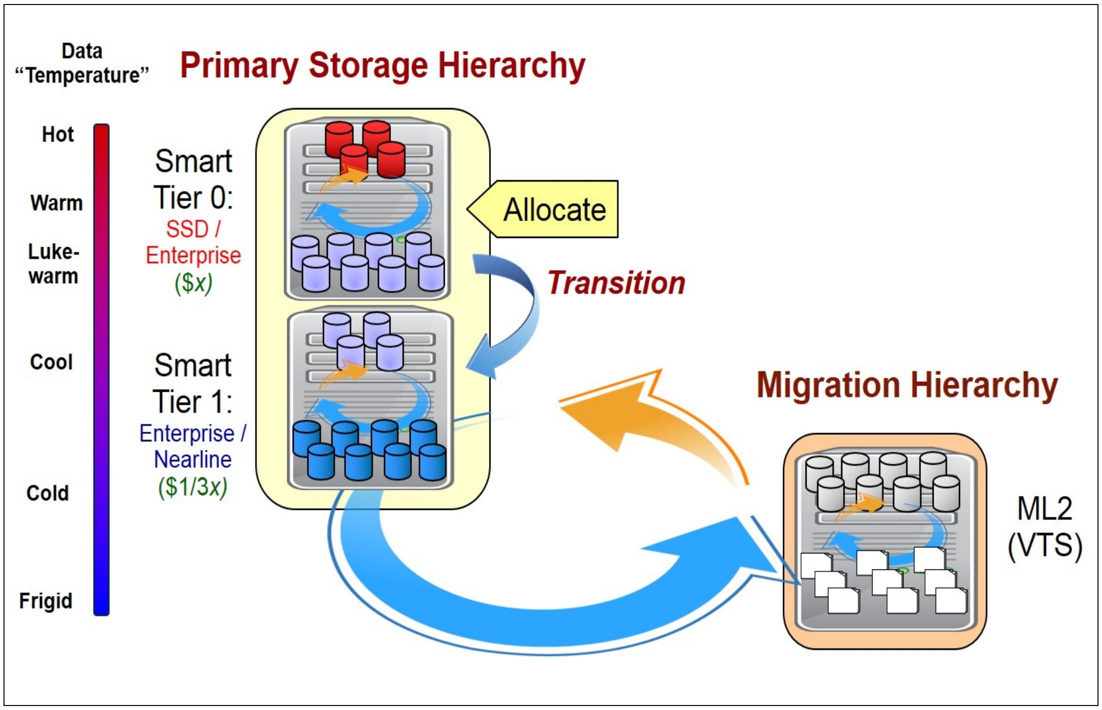


Figure 1-2 Smart tiers within the primary hierarchy

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 DASD pool, your options for application quality are limited. A multi-tiered structure (as shown in Figure 1-2) 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 DASD and Tape respectively, based on pre-defined policies and data access availability needs.

Figure 1-3 shows how the data flows between these distinct technologies, through DFSMShsm, and the challenges related to life cycle management.

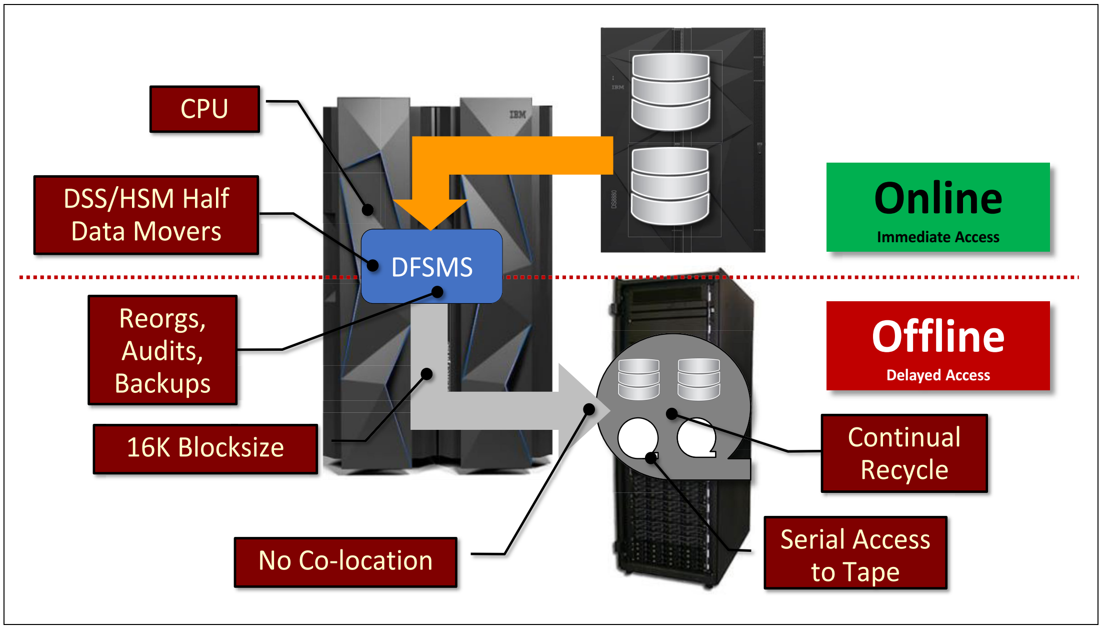


Figure 1-3 DFSMShsm data movement between DASD and Tape technologies

To move the data between the tiers, DFSMS uses HSM and DSS Data Movers to read the data from the source storage and write the data to the target storage, via 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 CPU cycles.

As you can imagine, to process all these operations the system will use IBM Z® 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 colocation 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 TCT

Transparent Cloud Tiering for IBM DS8000 was introduced to help customers to use IBM Z resources more efficiently. With its integration with z/OS through DFSMShsm, it allows clients to reduce CPU utilization by eliminating constraints that are tied to 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 REST API interface and issues commands for the DS8000 to move the data directly to/from the Cloud, as shown in Figure 1-4.

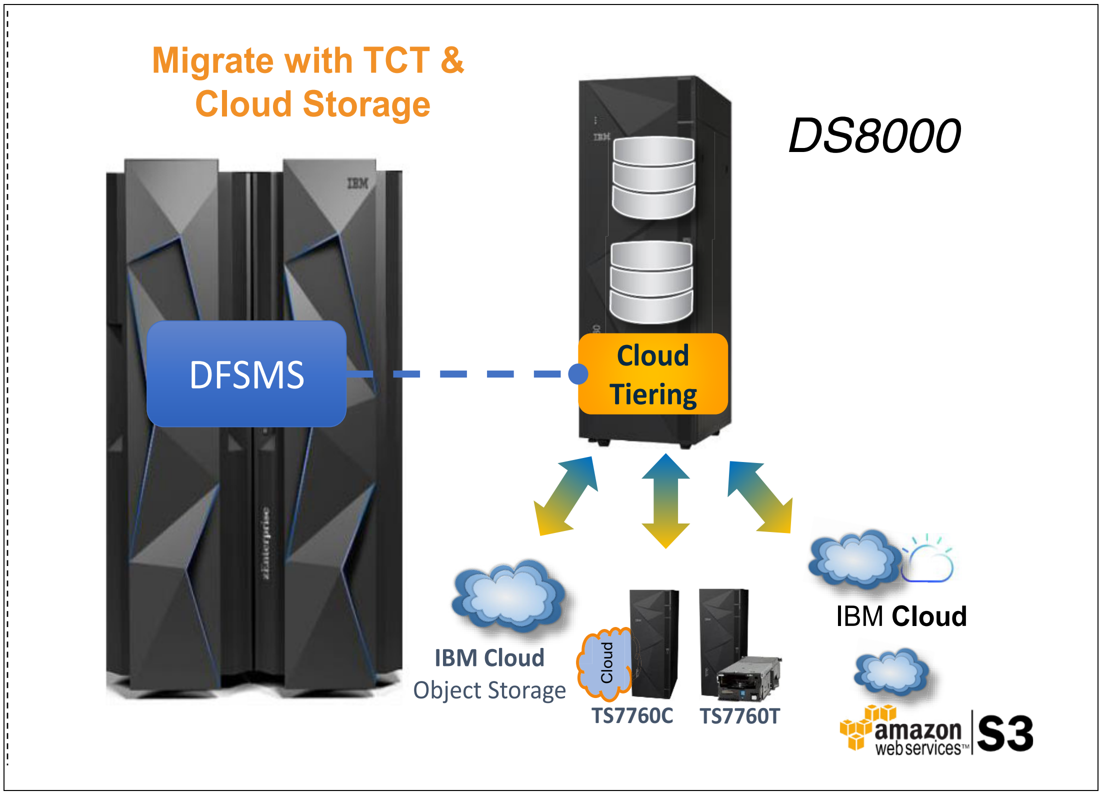


Figure 1-4 TCT data movement layout

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 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 16K blocks

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

2

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 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 cloud in the context of TCT?” on page 12
- 2.2, “Compute cloud versus storage cloud” on page 12
- 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 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 showing the different layers and where TCT integrates with the cloud:

- Application Layer: where 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: where 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 will cover this in more detail in this chapter.

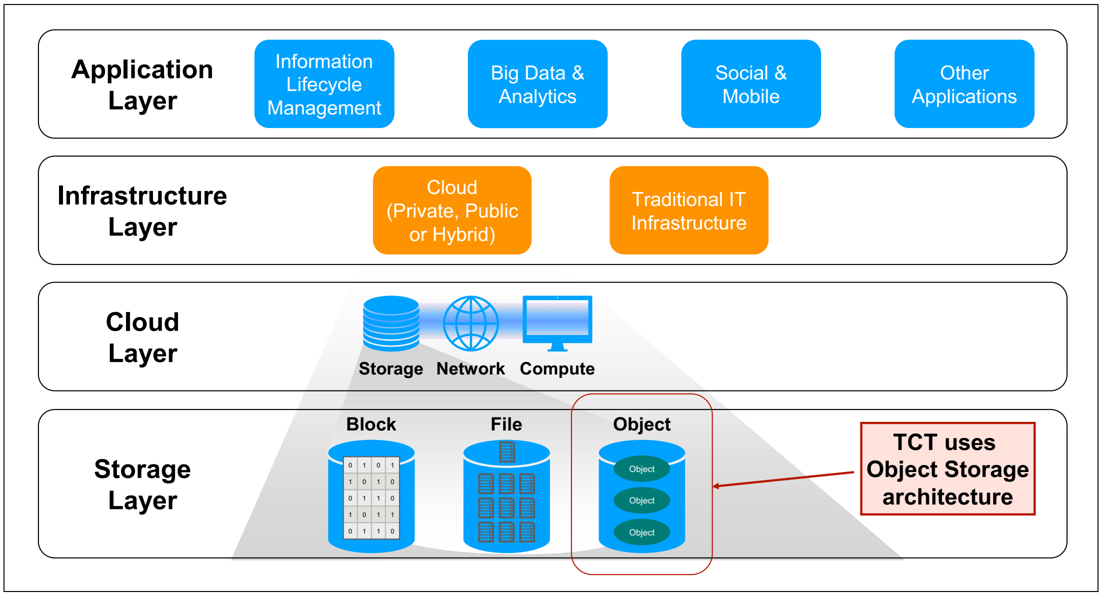


Figure 2-1 TCT in the context of the Cloud

2.2 Compute cloud versus 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.

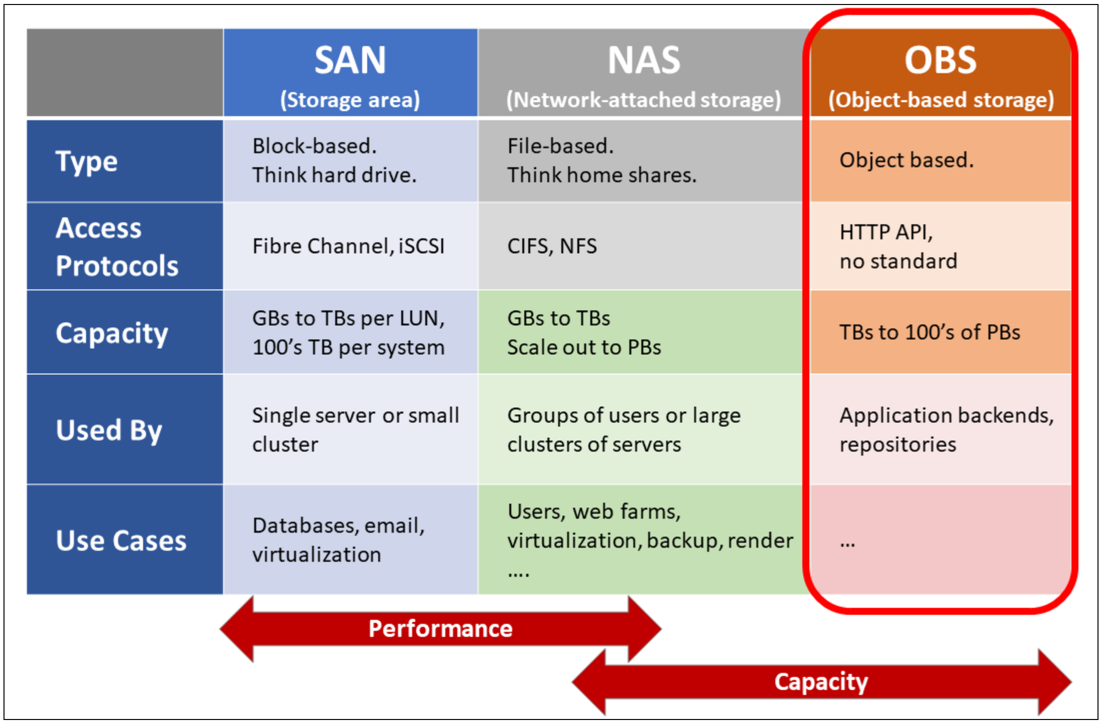
On the other hand, the storage cloud holds the data and caters for operations like backing up and archiving data. As mentioned earlier in this chapter, TCT uses Object Storage architecture to store the data it manages in the cloud.

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/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 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 Transparent Cloud Tiering function of IBM DS8000 provides a method to convert the block to Object Storage without additional 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 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 premise:* Consists of a private cloud infrastructure at your organization’s location.
- *Off premise:* Consists of a cloud infrastructure being 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 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.

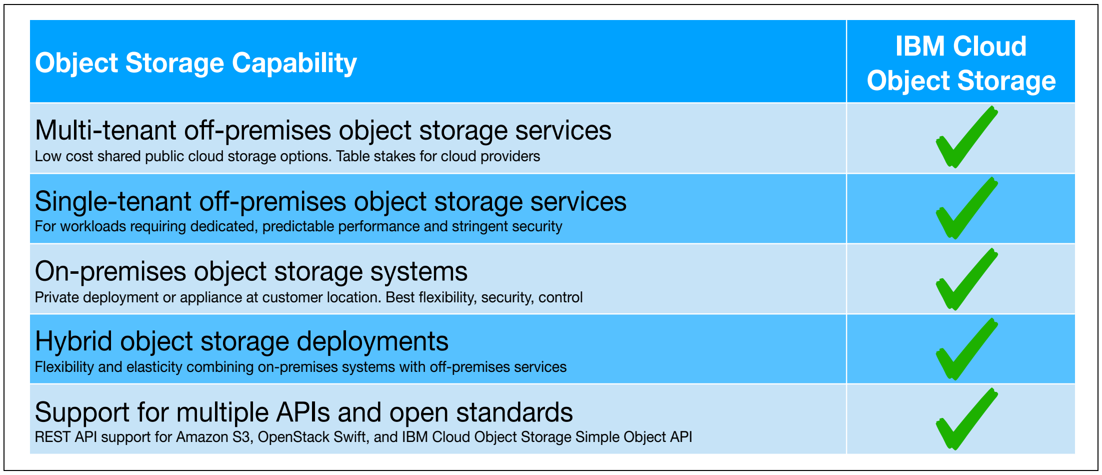
Hybrid Cloud

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

Figure 2-3 IBM Cloud Object Storage capabilities

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

For detailed information about the IBM Cloud Object Storage service offering, see the *Cloud Object Storage as a Service: IBM Cloud Object Storage from Theory to Practice - For developers, IT architects and IT specialists*, SG24-8385 Redbooks publication 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, refer to the *IBM Private, Public, and Hybrid Cloud Storage Solutions*, REDP-4873 Redbooks publication 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 Transparent Cloud Tiering 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.

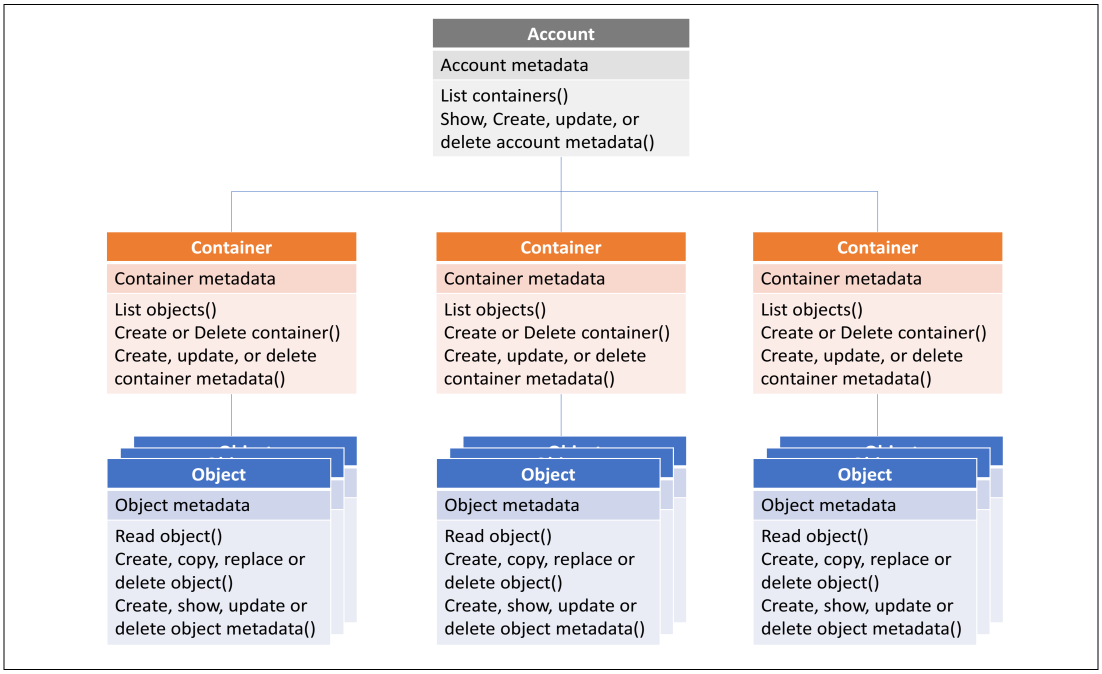

Account
Account metadata
List containers()
Show, Create, update, or delete account metadata()
Container
Container metadata
List objects()
Create or Delete container()
Create, update, or delete container metadata()
Container
Container metadata
List objects()
Create or Delete container()
Create, update, or delete container metadata()
Container
Container metadata
List objects()
Create or Delete container()
Create, update, or delete container metadata()
Object
Object metadata
Read object()
Create, copy, replace or delete object()
Create, show, update or delete object metadata()
Object
Object metadata
Read object()
Create, copy, replace or delete object()
Create, show, update or delete object metadata()
Object
Object metadata
Read object()
Create, copy, replace or delete object()
Create, show, update or delete object metadata()

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 (or buckets) are similar to folders in Windows or UNIX, and provide an area to organize and store objects, container-to-container synchronization, quotas, and object versioning. One main difference is that containers cannot be nested. That is, no support is available for creating a container within another container. Container names can be 256 bytes.

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

As of this writing, there is a 5 GB limit to the size of an object due to an Openstack Swift restriction. Objects larger than 5 GB must be broken up and stored by using multiple segment objects. After all of the segment objects are stored, a *manifest* object is created to piece all of the segments together. When a large object is retrieved, the manifest object is supplied and the Object Storage service concatenates all of the segments and returns them to the requester. For most sizes greater than 100 MB, the system does *multi-part uploads*. By creating multiple parts, the system does parallel recall, and greater recall efficiency is achieved.

> **Note:** The segmentation of objects into 5 GB chunks only applies to OpenStack Swift. For Clouds using the S3 API and for the TS7700, the restriction does not apply.

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 also 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, DFSMShsm and DFSMSdss do not use this feature. Instead, DFSMShsm handles the expiration of objects.

User-created backups (backups that are created outside of DFSMShsm 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 present, this process is not recommended.

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.

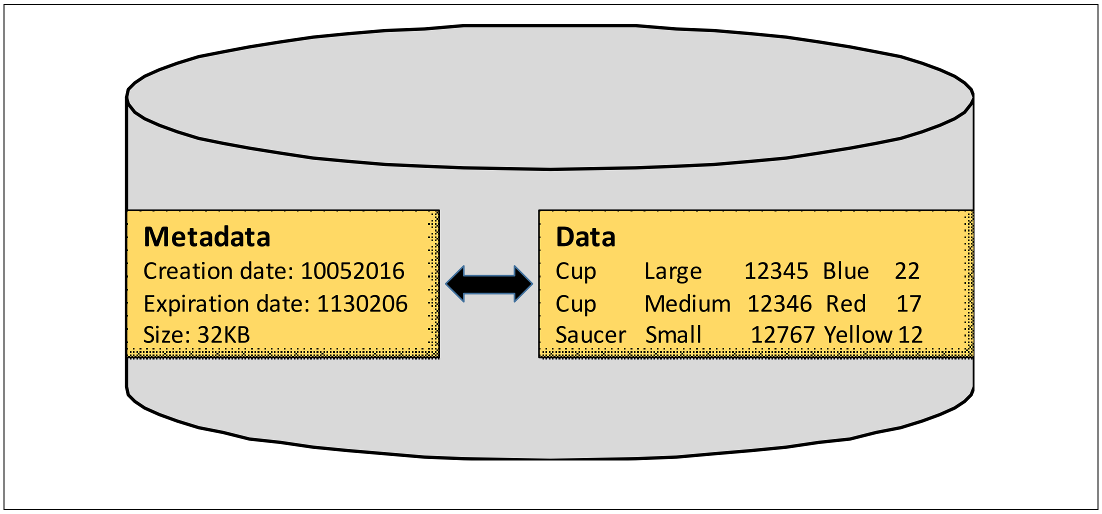


Figure 2-5 Data and metadata differences

3

Transparent cloud tiering

In this chapter, we describe how Transparent Cloud Tiering 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

DS8000 Transparent Cloud Tiering (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:** Currently, the only host application that can use TCT is the *Hierarchical Storage Manager* component of z/OS DFSMS (DFSMShsm or HSM).

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. When it migrates a data set to cloud storage, HSM creates a dump job for the DFSMS data mover service (DFSMSdss). DFSMSdss then initiates the move of the data:

- It creates metadata objects that describe the data set and are needed to rebuild it in case of recall.
- It sends these metadata objects to the cloud storage.
- 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 one extent object for each volume the data set is stored on.

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 in turn 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.

See “Storing and retrieving data using DFSMS” on page 23 for more detail about the DFSMSdss operations and metadata objects.

DS8000 Transparent Cloud Tiering supports several cloud storage target types:

- Swift: the Open Stack cloud object storage implementation, public or on-premise.
- IBM Cloud Object Storage (IBM COS) in the public IBM cloud or as on-premise cloud object storage solution.
- Amazon Web Services Simple Storage Service (AWS S3) in the Amazon public cloud.
- Generic S3 compatible object storage: any S3 compatible cloud object storage solution in a public or on-premise cloud implementation.
- 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 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 HMC acts as a cloud proxy. DFSMS sends commands and objects to the HMC, using the Swift protocol. The HMC passes them on to the cloud storage, using the appropriate protocol for the target type.

3.2 Transparent Cloud Tiering data flow

Traditional HSM data movement during availability and space management operations is performed over the FICON infrastructure. The data is transferred between the DASD controller, the host, and a tape controller. 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 via TCP/IP, using Ethernet ports in each of the two internal servers. You can either use free ports that are available onboard in each server, or purchase and connect a separate pair of more powerful Ethernet controllers. Connect the ports you intend to use for TCT to a network that extends to the cloud target. The DS8000 uses this connection to send and retrieve the extent objects that contain the actual customer data. In cloud proxy mode (see “Other cloud target types” on page 22), 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 server that owns the Logical Subsystem (LSS) associated with the request. This might lead to unbalanced migrations and recalls if the majority of the data is on a specific LSS.

Defining your storage groups with volumes from LSS that belong to both storage facility images can reduce the chances 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. 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 uses 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. Although the password is not included in DFSMS Cloud constructs, it is used by DFSMShsm and DFSMSdss to do store and recovery tasks from the cloud.

DFSMShsm stores an encrypted version of the password in its control data sets (CDSs) for use when migrating and recalling data. For manual DUMP and RESTORE operations, DFSMSdss requires the user ID and password to be included in the job definition.

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.

For more information about security and user ID and password administration for DFSMShsm, see Chapter 8, “Using automatic migration” on page 81.

Swift cloud type

For Swift cloud object storage targets, DFSMS sends and receives metadata directly to and from the cloud storage, using the z/OS Web Enablement Toolkit (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.

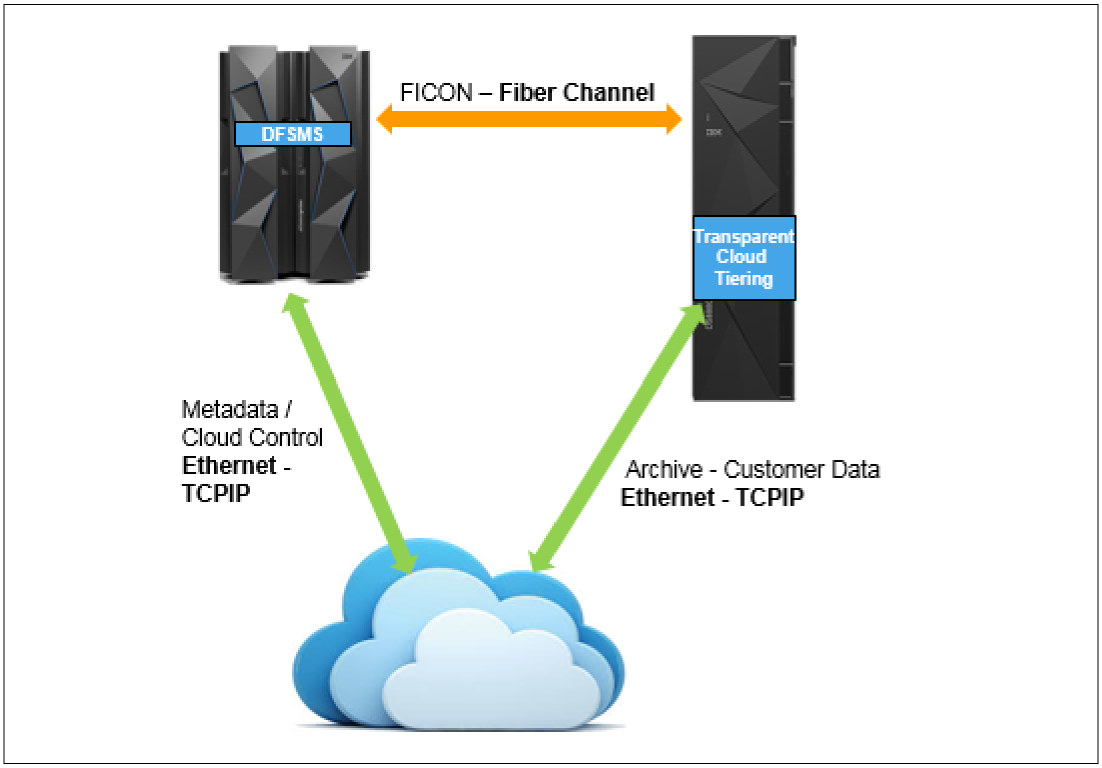

DFSMS
FICON – Fiber Channel
Transparent Cloud Tiering
Metadata / Cloud Control Ethernet - TCPIP
Archive - Customer Data Ethernet - TCPIP

Figure 3-1 Cloud communication with Swift API

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

Other cloud target types

The integration between z/OS DFSMS and the other cloud target types (S3, IBM COS, 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 (user name 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.

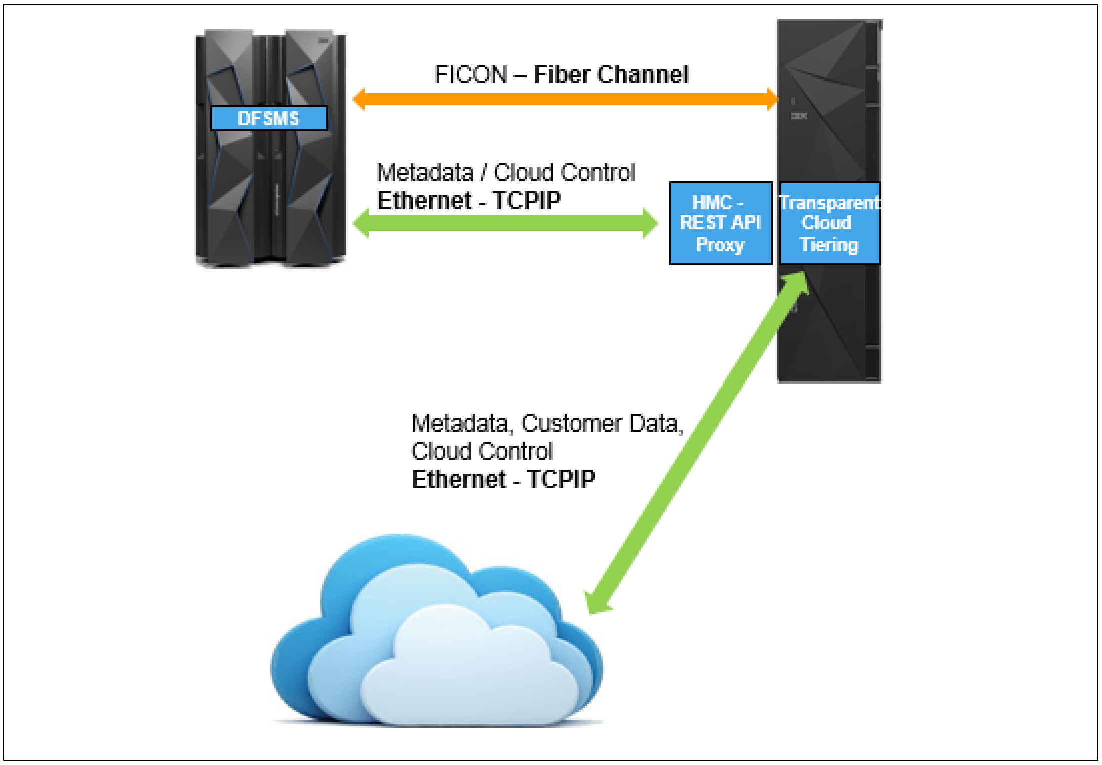


Figure 3-2 Cloud communication with S3 and IBM COS APIs

3.3 Storing and retrieving data using DFSMS

From a z/OS perspective, the storage cloud 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.

HSM uses DFSMSdss as the data mover when data is migrated to or recalled from cloud object storage. The number of objects for each data set can vary, based on the following factors:

- The number of volumes the 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 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 and the application.

Table 3-1 lists some objects that are created as part of the DFSMSdss 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 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: ► 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 DTSAIXS, 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: ► 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 the DSS metadata objects are stored, the extent (data) objects are stored by DS8000 transparent cloud tiering. The data object consists of the extents of the data set that are on the source volume. This process is repeated for every source volume where parts of the data set are stored.

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

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 PDS on RESTORE, nor reblock data sets while it is being dumped.

At the time of this writing, no compression is performed by the DS8000 during data migration. Your data might be compressed or encrypted when it is allocated on the DS8000. Such data is offloaded to cloud in its original condition: compressed or encrypted. (Compression or encryption is typically done by zEDC or pervasive encryption.)

To maintain some structure in the potentially very large number of objects, HSM uses cloud object storage containers. By default, a new container is created every 7 days. The container name reflects the creation date and the HSM plex name. See 7.2, “Cloud container management” on page 70 for details.

If you attempt to create an object prefix that exists within a container, DFSMSdss fails the DUMP to prevent the data from being overwritten.

The following migration considerations can guide clients who are looking towards implementing Transparent Cloud Tiering and configuring the DFSMShsm to use it:

- Already migrated data

  After you configure the DFSMShsm, you might want to move some of your data from other migration tiers, such as disk ML1, or tapes ML2. You need to keep in mind that currently there is no command or automated process to move migrated data directly to the cloud.

  If you want to move data already migrated to the cloud, you need to first recall your data, and then migrate it to the cloud, again.

- Recalling data from the cloud

  As you expand your environment to use cloud services, it is likely that more data will be migrated and recalled from the cloud. When you request a recall for a data set, the ACS routines are called, and the volume selection is done. Only cloud-capable volumes can be selected as target volumes. If no volumes in such condition exist, the recall fails.

  > **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 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 multi-target PPRC and IBM HyperSwap® solutions. The use of transparent cloud tiering, and storing data in cloud storage might affect your disaster recovery plan, and needs to be carried out during implementation.

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 disaster recovery has network access to the cloud environment. This might include configuring proxy, firewall, and other network settings to secure your connection.

- Cloud configuration

  Your disaster recovery DS8000 must be configured with the information necessary to access the cloud storage, including certificates to allow SSL connections. You might also have 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 userid and password on your disaster recovery DS8000 if you use S3 or IBM COS clouds. Update your z/OS to connect to the new DS8000, and have the userid defined 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 disaster recovery (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 related to network issues.

> **Note:** Theoretically it is also possible to create DFSMS dump and restore jobs manually that move data to and from cloud object storage. This is complex and poor practice because it requires clear text cloud credentials in the job definition

3.5 Transparent cloud tiering and DS8000 Copy Services

Many disaster recovery 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 know 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 has been issued with or without background copy or whether the background copy is still ongoing.

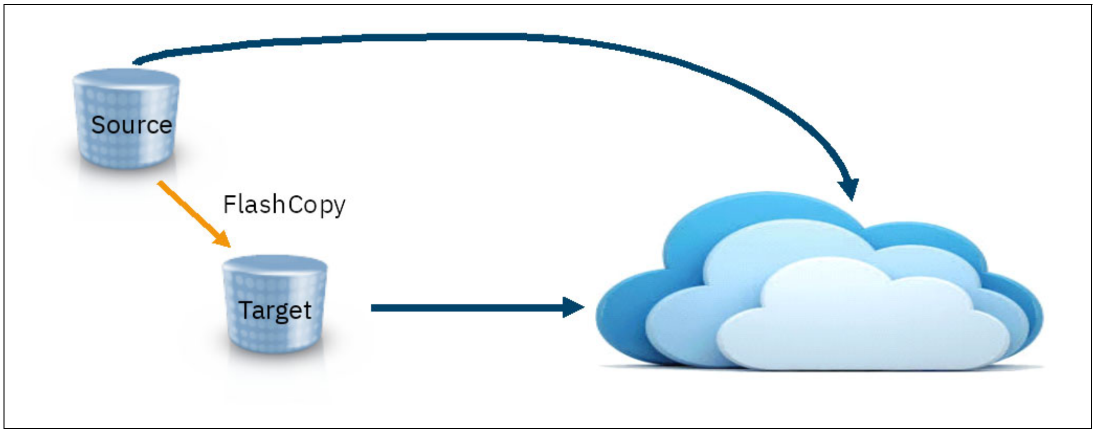


Figure 3-3 TCT migration with FlashCopy

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.

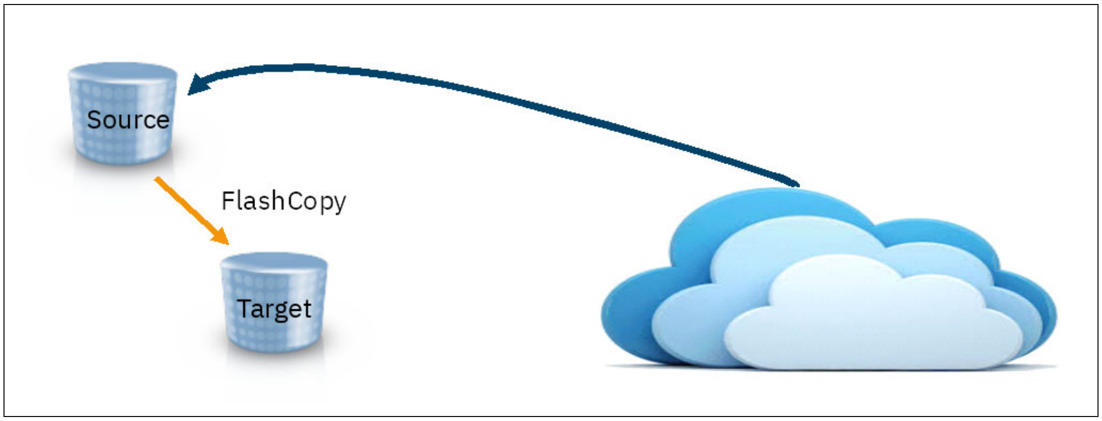


Figure 3-4 TCT recall with FlashCopy

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.

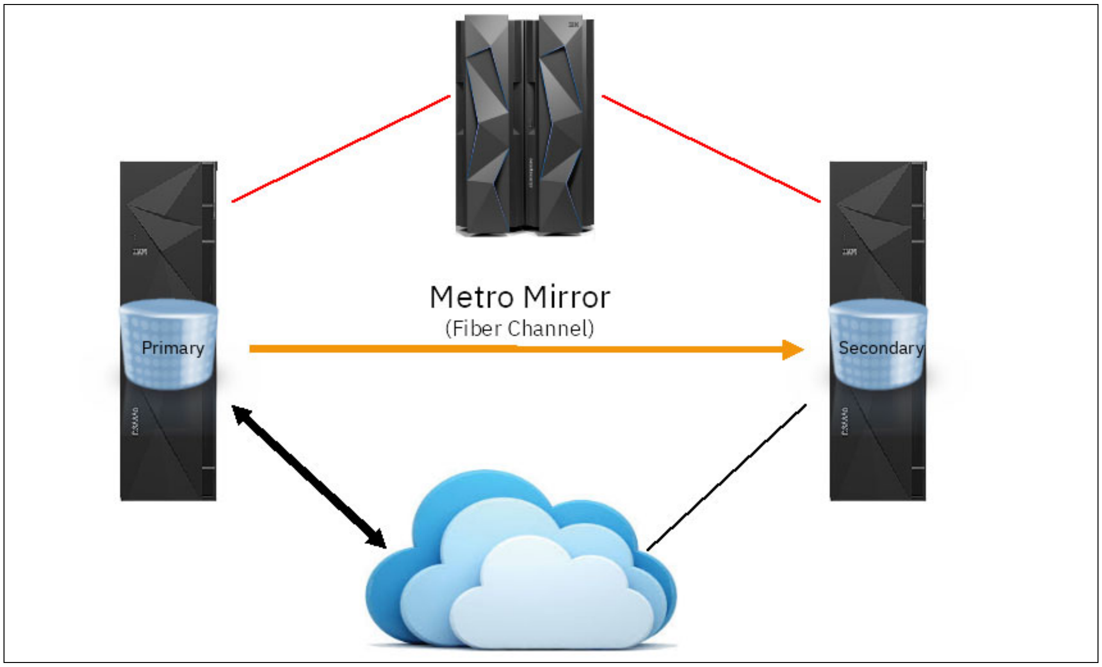


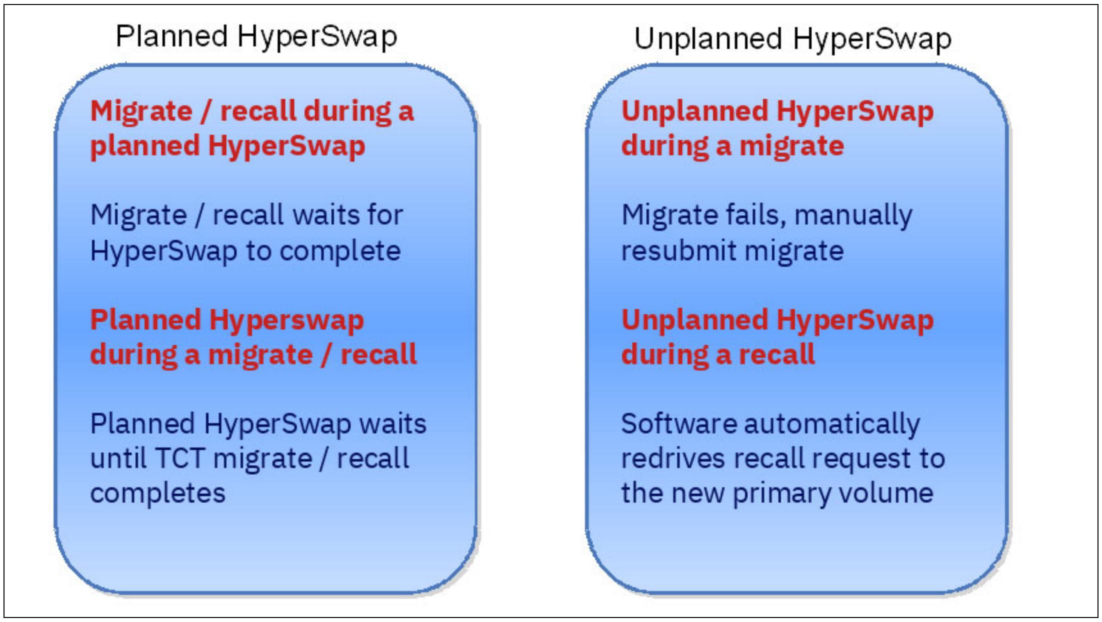
Figure 3-5 TCT with Metro Mirror

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 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 may be unavailable after a failure in the primary site.

3.5.3 Metro Mirror with HyperSwap

DS8000 TCT is aware if your environment is enabled for HyperSwap (either using CSM or IBM GDPS®). A collision of a HyperSwap event and a TCT migrate or recall operation is treated according to Figure 3-6.



Planned HyperSwap

Migrate / recall during a planned HyperSwap

Migrate / recall waits for HyperSwap to complete

Planned Hyperswap during a migrate / recall

Planned HyperSwap waits until TCT migrate / recall completes

Unplanned HyperSwap

Unplanned HyperSwap during a migrate

Migrate fails, manually resubmit migrate

Unplanned HyperSwap during a recall

Software automatically redrives recall request to the new primary volume

Figure 3-6 TCT operations and HyperSwap

3.5.4 Global Mirror

Just as with Metro Mirror, TCT also supports migrate and recall operations to and from Global Mirror primary volumes, as illustrated in Figure 3-7.

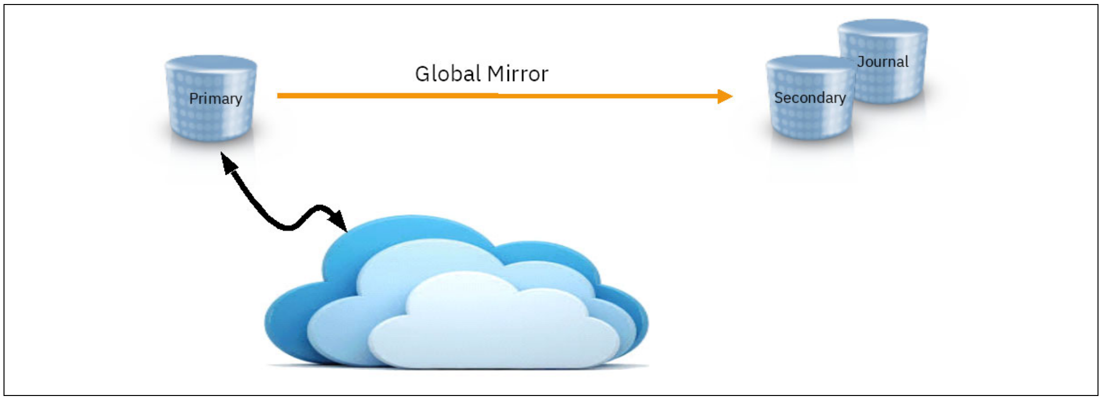


Figure 3-7 TCT and Global Mirror

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 have a requirement to 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 with TCT.

3.5.5 Multisite 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 implication regarding TCT operations after recovery described for the 2-site configurations are valid for multi-site, too.

3.6 Transparent Clout Tiering encryption

Your Transparent Cloud Tiering 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.

Data is encrypted in the DS8000 and therefore already protected while it is transferred over the network. TCT uses IBM POWER® Systems 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, it 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 Transparent Cloud Tiering Encryption will not encrypt again.

> **Restriction:** TCT encryption, as of DS8000 R9.0, is not supported with TS7700.

TCT encryption relies on external key servers for key management: It uses the industry standard Key Management Interoperability Protocol (KMIP). At the time of this writing, IBM SKLM is the only supported key management solution. TCT requires at least two SKLM servers at a minimum version of 3.0.0.2, configured in multi-master mode.

In high availability and disaster recovery scenarios using Metro Mirror or Global Mirror, all DS8000 systems must be connected to the same cloud, and all must be configured for Transparent Cloud Tiering encryption. Every DS8000 must be added to the TCT encryption group in SKLM during setup. This way any DS8000 in the DR configuration can decrypt the data in the cloud, even if it was encrypted by another one.

TCT encryption does not require a specific license and can be used in conjunction or independently from data at rest encryption.

See *IBM DS8000 Encryption for data at rest, Transparent Cloud Tiering, and Endpoint Security (DS8000 Release 9.0)*, REDP-4500 for more details about TCT encryption and implementation instructions.

3.7 Selecting data to store in a cloud

When you decide to implement Transparent Cloud Tiering, 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. Also, only Simplex volumes are eligible to have their data sets moved to the cloud.

Because no Object Storage data is cataloged or automatically deleted (except for DFSMShsm-owned objects), it is suggested that you proceed with caution when deciding which users can dump and restore their data sets from cloud. If any users decide to use the cloud, they must manually do housekeeping and delete the storage objects and containers that are created by them.

The area of DFSMShsm operations is a principal use case for storage cloud, for these reasons:

- DFSMShsm maintains information on containers and objects in its control data sets.
- It can retrieve or expire in its control data sets.
- It can automatically retrieve and expire the objects that are associated with data sets that are migrated to cloud storage.

The latest APARs for auto-migration allow for deleting empty containers.

Part 2

Cloud setup and use

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

This part includes the following chapters:

4

Requirements

In this chapter, we describe the requirements for Transparent Cloud Tiering, including DS8000, network and z/OS environment:

4.1 Ethernet connections on DS8000

In order to implement Transparent Cloud Tiering, 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 MES for existing ones.

The built-in 1 Gbps Ethernet card is located in location code P1-C10 or P1-C11 (depending on the model), with the upper ports T1 and T2 used for internal communications and the bottom ports T3 and T4 available for TCT, as shown in Figure 4-1. 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.

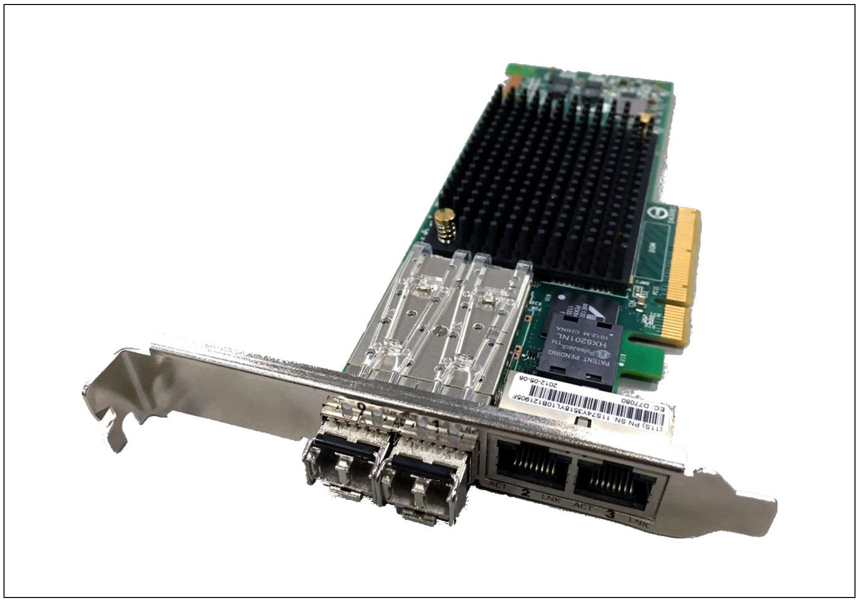


Figure 4-1 Built in 1 Gbps Ethernet ports for TCT

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 3600 – Transparent cloud tiering 10 Gb/1 Gb ethernet pair for 2U controllers (DS8884 and DS8884F). It is plugged into location code P1-C11, as shown in Figure 4-2 on page 35.

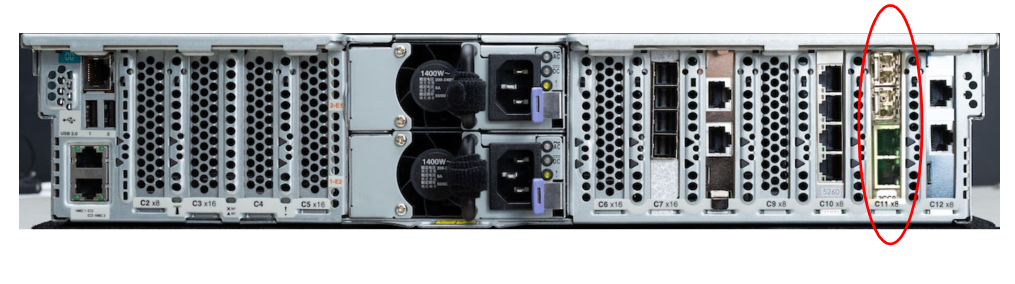


Figure 4-2 Location of the 10 Gbps Ethernet cards in 2U DS8884 servers

– Feature code 3601 – Transparent cloud tiering 10 Gb/1 Gb ethernet pair for 4U controllers (DS8886, DS8888, DS8886F and DS8888F). It is plugged into location code P1-C11, as shown in Figure 4-3.

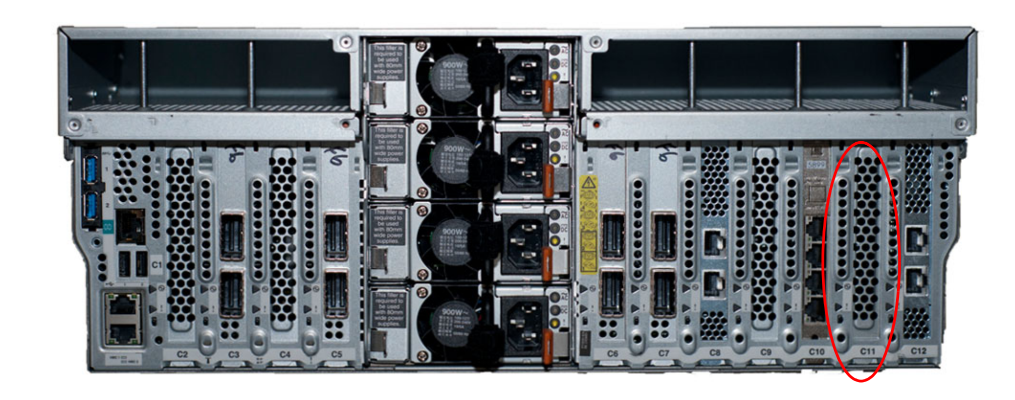


Figure 4-3 Location of the 10 Gbps Ethernet cards in 4U DS8886 servers

They contain 2 x 10 Gbps LR ports (optical SFP+) and 2 x 1 Gbps ports (RJ45 Copper). The card is physically located in location code P1-C11 or P1-C12 (depending on the model),

- For the newer DS8900F generation, the cards were slightly changed. They contain 2 x 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 on page 36.

  – Feature code 3602 - Transparent cloud tiering 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.

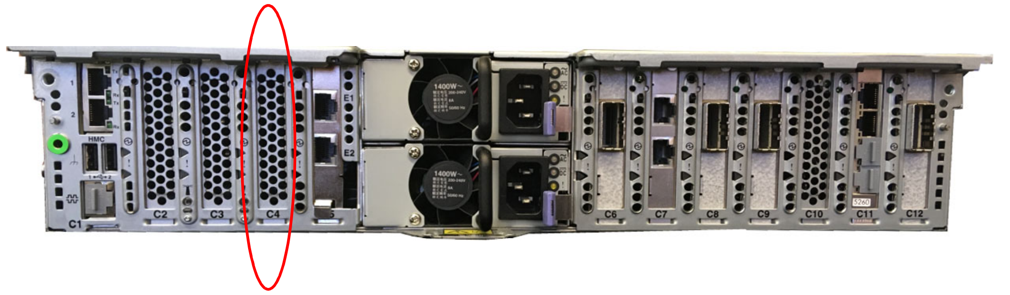


Figure 4-4 Location of 10 Gbps Ethernet card in 2 U serves (DS8910F models 994

– Feature code 3603 - Transparent cloud tiering 10 Gb/1 Gb ethernet pair V2 for 4U controllers. It is plugged into location code P1-C10.

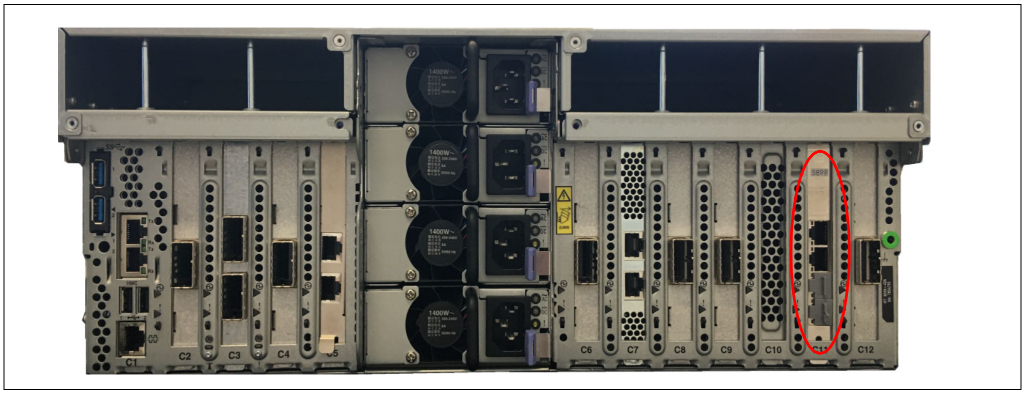


Figure 4-5 Location of 10 Gbps Ethernet card in 4U serves (DS8950F model 996)

The adapters are installed in addition. All ports on the new cards and the built-in ports can be used for TCT.

With the additional Ethernet cards, you have a total of six Ethernet ports available for use with TCT per server. However, you can only connect each DS8000 to one cloud object storage, and only one port per server at a time is used for data transfer. All others that are configured and can reach the cloud storage are there for redundancy.

The following configurations are considered good practices:

- Connect both 10 Gbps Ethernet ports for redundancy.
- Do not intermix 10 Gbps and 1 Gbps ports.

Note: To configure the Ethernet ports using the DSCLI, you need their port IDs. These IDs depend on the plug location. You can use the DSCLI command `lsnetworkport -l` to determine the port ID against their location code.

4.2 z/OS Level

To set up cloud configuration you must have z/OS V2R1 with PTF/APAR OA51622, z/OS V2R2 with PTF/APAR OA50667, PTF/APAR OA52901, PTF/APAR OA56048 or higher.

The details of these APARs can be found at the following web pages:

http://www.ibm.com/support/docview.wss?uid=isg1OA51622

http://www.ibm.com/support/docview.wss?uid=isg1OA50667

http://www.ibm.com/support/docview.wss?uid=isg1OA52901

http://www.ibm.com/support/docview.wss?uid=isg1OA56048

DFSMShsm automatic migration support is available on z/OS V2R2 and above with PTF for APAR OA52913. Coexistence support is also available on z/OS V2R1 with PTF for APAR OA52913.

http://www.ibm.com/support/docview.wss?uid=isg1OA52913

APARs OA55538 provides z/OS support for TCT encryption. They enable z/OS to notify the DS8000 if a data set is already encrypted to avoid double encryption.

https://www.ibm.com/support/docview.wss?uid=isg1OA55538

For TCT with TS7700 as cloud object target, you need APAR OA58225:

https://www.ibm.com/support/docview.wss?uid=isg1OA58225

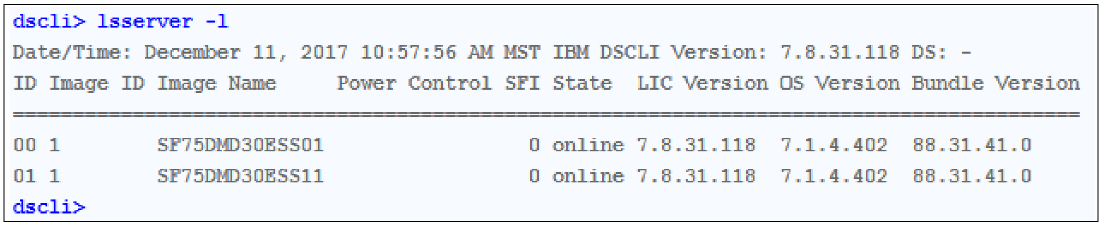
Use the IBM.Function.DFSMSCloudStorage fix category to identify PTFs associated with the DFSMS TCT support.

https://www.ibm.com/systems/z/os/zos/features/smpe/fix-category.html

4.3 DS8000 Release Level

To set up the cloud configuration, your DS8000 must have at least release level 8.2.3 – Bundle 88.23.19.0 Microcode and DSCLI.

To check your current DS8000 microcode level, issue the **lsserver -l** DSCLI command as shown in the Figure 4-6.



```
dscli> lsserver -l
Date/Time: December 11, 2017 10:57:56 AM MST IBM DSCLI Version: 7.8.31.118 DS: -
ID Image ID Image Name     Power Control SFI State  LIC Version OS Version Bundle Version
=========================================================================================
00 1        SF75DMD30ESS01                 0 online 7.8.31.118  7.1.4.402  88.31.41.0
01 1        SF75DMD30ESS11                 0 online 7.8.31.118  7.1.4.402  88.31.41.0
dscli>
```

Figure 4-6 Display DS8000 Release on DSCLI

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

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

- **DS8880 Release 8.2.3**
  - First release supporting Transparent Cloud Tiering
  - OpenStack Swift API to connect to object storage systems
- **DS8880 Release 8.3**
  - Support for Amazon AWS and IBM Cloud Object Storage via 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/Metro Global Mirror support
  - Transparent Cloud Tiering encryption support

- **DS8880 Release 8.5.4 and DS8900F Release 9**
  - TS7700 as cloud object storage target
  - Generic S3 cloud object storage target
  - Multi Target PPRC support

4.4 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)
- SSL Certificates (if using SSL/TLS)

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

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

4.4.1 Endpoint

The endpoint is the location or URL that the DS8000 (and DFSMS for Swift) use 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. As of the writing of this book, 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 may 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 five characters, ranging from 0 to 65535. You must also ensure that this port is open on the local network firewalls.

4.4.2 Cloud credentials

The cloud administrator provides a set of credentials. Their names and extent differ by cloud target type. You have to 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. See Chapter 5, “Configuring the DS8000 for TCT” on page 41 and Chapter 6, “Configuring DFSMS for TCT” on page 53 for more details about the required credentials for the different cloud target types and how to provide them.

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 good 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 additional abstraction layer is required, called *Tenant*. This 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 could be *production*, *development*, *test*, and so on. You can either choose this name when requesting cloud storage access, or it is pre-defined by the cloud administrator.

4.4.3 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, Transport Layer Security (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 only used to encrypt the authentication data between DFSMS, the DS8000, and the cloud object storage. You have to configure and enable TCT encryption to encrypt the customer data during transmission and while it resides 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 these. If you use a Certificate Authority (CA), you need the CA's root certificate and any intermediate certificates 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.

4.5 TLS/SSL considerations

In Chapter 5, “Configuring the DS8000 for TCT” on page 41 and Chapter 6, “Configuring DFSMS for TCT” on page 53 of this book we demonstrate how you can configure the DS8000 system and DFSMS to use certificates for secure communications.

DFSMS and IBM DS8000 will send account information (user names and passwords) over HTTP connection. To ensure that the information is encrypted, we highly recommend establishing a secure HTTP connection between the z/OS host, the IBM DS8000 system, and the object storage cloud server.

The supported SSL/TLS versions to be used when making HTTP requests are: TLSV12, TLSV11, TLSV1, SSLV3.

There are two types of authentication:

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

4.5.1 External CA versus self-signed certificates with DS8000 and DFSMS/RACF

Determining the type of certificate to use for secure communications sessions and the method to generate the certificate is challenging. Self-signed certificates and digital certificates issued by certificate authorities offer advantages and disadvantages.

The Table 4-1 compares the advantages and disadvantages of self-signed and CA-signed certificates.

Table 4-1 External certificate versus self-signed certificate

Type of Certificate	Advantages	Disadvantages
Self-signed certificate	No cost	Requires you to distribute your certificate, minus the private key, to each trading partner in a secure manner
	Easy to generate	Difficult to maintain; anytime the certificate is changed, it must be distributed to all clients
	Self-validated	Not validated by a third-party entity
	Efficient for small number of trading partners	Inefficient for large number of trading partners
CA-signed certificate (External)	Eliminates having to send your certificate to each trading partner	Trading partners must download digital CA-signed certificate used to verify the digital signature of trading partner public keys
	No changes are required on the trading partner's system if you recreate the CA digitally-signed certificate using the same CA	Must be purchased from third-party vendor

For the Swift API, swift-keystone cloud interface is used to encrypt authentication credentials and connect DFSMS and IBM DS8000 to a cloud storage target. The authentication is done using root or system certificates with either Secure Sockets Layer or Transport Layer Security, SSL/TLS. For S3 or IBM Cloud Object Storage, the DS8000 system will have the certificate files to communicate with the cloud storage target and will communicate with DFSMS through a REST API proxy that runs in the HMC of the DS8000. The configuration steps to allow this communication are detailed in the next chapters of this book.

5

Configuring the DS8000 for TCT

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

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 HMC as cloud proxy for DFSMS.

This chapter includes the following topics:

- 5.1, “Configuring the IBM DS8000 for TCT” on page 42
- 5.2, “Maintaining the cloud server configuration” on page 49
- 5.3, “Prepare the DS8000 as cloud proxy” on page 50

5.1 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 the components:

- Ethernet port configuration
- Cloud storage configuration
- Prepare a DS8000 userid for the cloud proxy functionality (for all cloud target types except Swift).

5.1.1 Ethernet configuration

Assuming you have connected the DS8000 to your network according to 4.1, “Ethernet connections on DS8000” on page 34, you can now configure the Ethernet ports that you are using for TCT. This is required to enable network connectivity. Use the **lsnetworkport** command from the DS8000 command-line interface (DSCLI) to display any current Ethernet configuration. Example 5-1 shows a sample output from the **lsnetworkport** command.

Example 5-1 Output from lsnetworkport command

```
ID      IP address  Subnet Mask  Gateway Primary DNS    Secondary DNS   State
I9813   0.0.0.0     0.0.0.0      0.0.0.0  0.0.0.0       0.0.0.0         Offline
I9814   0.0.0.0     0.0.0.0      0.0.0.0  0.0.0.0       0.0.0.0         Offline
I9B13   0.0.0.0     0.0.0.0      0.0.0.0  0.0.0.0       0.0.0.0         Offline
I9B14   0.0.0.0     0.0.0.0      0.0.0.0  0.0.0.0       0.0.0.0         Offline
```

Note: There is no way to delete a network port at this time. You can only clear the IP address or just leave it.

Ensure that you have the correct IP addresses, subnet mask, and DNS information available while you are configuring your hardware. Then, use the **setnetworkport** command to define the network settings, as shown in Example 5-2.

Example 5-2 Defining the network settings

```
setnetworkport -ipaddr 10.0.1.2 -subnet 255.255.255.0 I9814
setnetworkport -ipaddr 10.0.1.3 -subnet 255.255.255.0 I9B14
```

After the configuration is complete, issue another **lsnetworkport** command to confirm that the network was properly configured. Example 5-3 shows the new network configuration.

Example 5-3 Verifying the network configuration

```
ID      IP address  Subnet Mask   Gateway   Primary DNS    Secondary DNS  State
I9813   0.0.0.0     0.0.0.0       0.0.0.0   9.0.000.10     0.0.0.0        Offline
I9814   10.0.1.2    255.255.255.0 0.0.0.0   9.0.000.10     0.0.0.0        Online
I9B13   0.0.0.0     0.0.0.0       0.0.0.0   9.0.000.10     0.0.0.0        Offline
I9B14   10.0.1.3    255.255.255.0 0.0.0.0   9.0.000.10     0.0.0.0        Online
```

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

Note: Although you can have up to six Ethernet ports available in each DS8000 internal server, only one is used for data transfer at any given time. Configure at least two ports of the same type per server for redundancy. All configured ports in a server must be connected to the same network.

5.1.2 Cloud configuration

In the next step, you configure the DS8000 for access to the cloud storage. 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: SSL/TLS secured authentication to a Swift keystone service to access Swift cloud object storage.
  - ibmcos: IBM Cloud Object Storage (COS) either on premise or in the IBM cloud
  - aws-s3: Amazon Simple Storage Service cloud object storage using the S3 API
  - s3: Generic S3 compatible cloud target
  - ts7700: IBM Virtual Tape Server TS7700 as cloud object storage

- **-endpoint** (required for all cloud target types except ts7700): the 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 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 S3 type object storage solution, use the *access key* 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 S3 type object storage solution, use the *secret access key* provided by the cloud administrator.
- **-rootcaloc, intermcaloc, syscaloc** (required for swift-keystone and all S3 type targets if IP security is to be used): use these parameters to provide certificates to the DS8000 for secure authentication with the object storage. Specify the location of the certificate (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 certificate authority (CA), the root CA and intermediate CA can be provided. These parameters cannot be used if **-nossl** is specified.
- **-nossl** (required for the TS7700 cloud target type, optional for all others): allow insecure authentication with the object storage target. This parameter cannot be specified together with -**rootcaloc**, **-intermcaloc**, or **-syscaloc**.
- **-loc** (optional and valid only for aws-s3 and ibmcos type targets):
  - ibmcos: specify the name of your vault template on the IBMCOS 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 TC encryption (refer to 3.6, “Transparent Clout Tiering encryption” on page 29 for more information).
- **`-primary7700IPs`** (required and valid only for TS7700): specify IP address(es) of the primary TS7700 VTS 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 address(es) of the secondary TS7700 VTS 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 (see **6.4, “Creating a DFSMS cloud connection using ISMF” on page 58** for details). This is a positional parameter. 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 object 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.

Connection to on-premise IBM Cloud Object Storage

In this section we explain how to connect your DS8000 to an on-premise IBM Cloud Object Storage (IBMCOS) system. Before connecting the DS8000, you have to prepare the IBMCOS according to the DS8000 Knowledge Center section *Configuring the IBM Cloud Object Storage System for Transparent Cloud Tiering*. The following link leads you to the DS8000 Release 9.0 version:

https://www.ibm.com/support/knowledgecenter/SSHGBU_9.0.0/com.ibm.storage.ssic.help.doc/f2c_configuring_trans_cloud_tiering.html

During the configuration, you create an IBMCOS 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 5-4, we show the command to configure the DS8000 cloud connection for the simple case without SSL and encryption.

Example 5-4 DS8000 cloud connection to IBMCOS without SSL and encryption

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

CMUC00560W mkcloudserver: Use of the -nossl 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 IBMCOS accessor nodes
- username: the Access Key for the IBMCOS user
- pw: the Secret Access Key for the IBMCOS user
- nossl: connect without authorization security
- loc: the provisioning code for the IBMCOS 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 IBMCOS. This 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 5-5, we show another IBMCOS connection, this time with SSL and encryption support.

Example 5-5 DS8000 cloud connection to IBMCOS 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, user name, password, endpoint, and cloud connection name are the same as in Example 5-4. 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 additional parameters:

- syscaloc: specifies the file containing the IBMCOS system certificate. You can download it from the IBMCOS manager.
- keygrp: specify the key group used for TCT encryption.

> **Note:** Before you can use TCT encryption, you have to configure your SKLM 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 9.0)*, REDP-4500.

Connection to cloud object storage service in the public IBM Cloud

In this example, we connect the DS8000 to the cloud object storage service in the public IBM cloud.

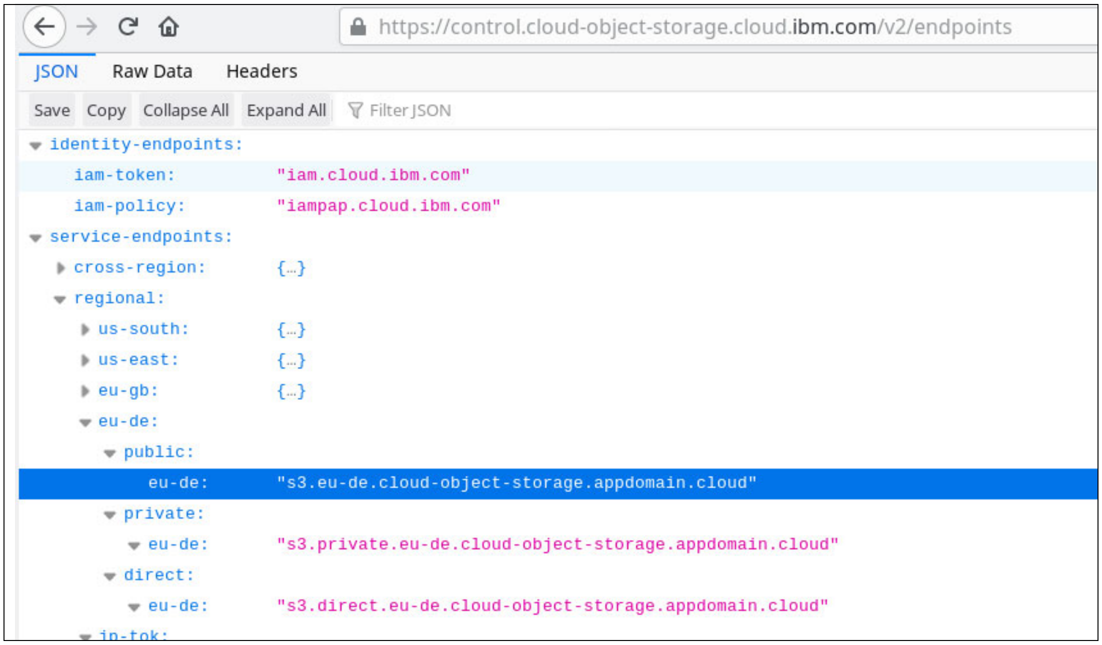
After you signed up to the IBM Cloud and defined the cloud object storage instance, you can create credentials for your service. The credentials are provided in JSON format like that shown in Example 5-6 on page 46.

Example 5-6 Credential provided by IBM Cloud for cloud object storage

```
{
  "apikey": "LB1Iki88rfqXJSy1oXFlAn8iOWLu_lzigjnOxTpAYG73",
  "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 user name 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 5-1.


https://control.cloud-object-storage.cloud.ibm.com/v2/endpoints
JSON
Raw Data
Headers
Save
Copy
Collapse All
Expand All
Filter JSON

```
identity-endpoints:
  iam-token:        "iam.cloud.ibm.com"
  iam-policy:       "iampap.cloud.ibm.com"
service-endpoints:
  cross-region:     {…}
  regional:
    us-south:       {…}
    us-east:        {…}
    eu-gb:          {…}
    eu-de:
      public:
        eu-de:      "s3.eu-de.cloud-object-storage.appdomain.cloud"
      private:
        eu-de:      "s3.private.eu-de.cloud-object-storage.appdomain.cloud"
      direct:
        eu-de:      "s3.direct.eu-de.cloud-object-storage.appdomain.cloud"
    ip-tok:
```

Figure 5-1 IBM Cloud object storage service endpoints

Find the endpoint that matches the regional settings of the cloud object storage service you defined and use it for the endpoint parameter of the **`mkcloudserver`** command, as shown in Example 5-7.

Example 5-7 DS8000 cloud connection to the public IBM Cloud without SSL and encryption

```
dscli> mkcloudserver -type ibmcos -username 648...630 -pw f7b...e6f -nossl -endpoint
http://s3.eu-de.cloud-object-storage.appdomain.cloud pubcos
CMUC00560W mkcloudserver: Use of the -nossl 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, using the **openssl** command). Then provide the certificate file in the **syscaloc** parameter of the **mkcloudserver** command.

Connection to on-premise S3 compatible object storage

The *Generic S3* cloud target type of the DS8000 allows to connect any other cloud storage using the S3 API for access. In Example 5-8, we show a command that creates a connection to our own on-premise cloud object storage based on the open source project minio (minio.io).

Example 5-8 DS8000 cloud connection to a generic S3 target without security and encryption

```
mkcloudserver -type s3 -username YOB...8SX -pw 7Sv...Iex -nossl -endpoint
http://9.155.49.146:9000 minioz
CMUC00560W mkcloudserver: Use of the -nossl 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 we have to specify follow the same rules as in “Connection to on-premise IBM Cloud Object Storage” on page 44, with the exception of the **-loc** parameter. The *Generic S3* cloud target type does not support a location specification.

Connection to a TS7700 Virtual Tape Server

Before you can connect a DS8000 to a TS7700 Virtual Tape Server (VTS), the VTS must be prepared:

- A license for the *DS8000 Object Store* feature (FC 5282) is 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 *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 standalone system, there is an additional 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 1.0*, REDP-5583, or the IBM white paper, *TS7700 Series DS8000 Object Store Users Guide V1.0* which is available on the following website:

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102800

The connection to a TS7700 Virtual Tape Server 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 5-9, we show the DSCLI command to connect a DS8000 to a single TS7700.

Example 5-9 DS8000 cloud connection to an IBM TS7700 Virtual Tape Server

```
dscli> mkcloudserver -type TS7700 -primary7700IPs 192.168.100.1,192.168.100.2 -nossl TS7700
CMUC00560W mkcloudserver: Use of the -nossl 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.
```

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.
- nossl: the `nossl` parameter is required, because there is no authentication.
- The last parameter again is a required positional parameter without keyword. Specify the name of your DS8000 cloud connection here.

You can optionally specify an alternate TS7700, using the `secondary7700IPs` parameter. If you specify a second VTS, the DS8000 will mirror all cloud objects to both of them. Therefore, migrate objects can be retrieved from the TS7700 cloud targets, even if one of the VTSs where unavailable.

If one of the two TS7700s should become unavailable while the DS8000 is transferring data to the pair, the transfer and therefore the migration will complete regardless, with the single copy. The DS8000 will instruct the TS7700 to send the missing objects to the second one as soon as it is back in operation.

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

Connection to on-premise Swift cloud object storage

Example 5-10 shows a sample `mkcloudserver` command to configure a Swift Keystone authenticated cloud object storage as cloud server.

Example 5-10 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, 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 certificate authority (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.

5.2 Maintaining the cloud server configuration

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

5.2.1 Listing a cloud server configuration

You can also list the cloud configuration on your hardware. Use the **`lscloudserver`** command to list cloud information. Sensitive information, such as user ID and password, is not displayed in the output. The Example 5-11 displays a sample output from the **`lscloudserver`** command.

Example 5-11 Listing cloud information

```
dscli> lscloudserver

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/
```

5.2.2 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 5-12.

Example 5-12 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 new configuration using the **`mkcloudserver`** command.

> **Note:** At the time of this writing, only a single cloud can be configured at a specific time. If a new cloud must be configured, the current setting must be deleted before the new cloud definition is used, using the **`rmcloudserver`** command.

5.3 Prepare the DS8000 as cloud proxy

With the TCT feature, the DS8000 acts a cloud proxy for the mainframe. This capability enables support for S3, IBM COS, and TS7700 cloud interfaces for Transparent Cloud Tiering. 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 functionality in the following sections.

5.3.1 Configure a DS8000 user for REST API proxy

You need a DS8000 user ID that will be used by DFSMS to authenticate on the DS8000 system, using the REST API interface. The REST API proxy service is automatically enabled when the DS8000 Code level is upgraded to R8.3 or higher. No other tasks are required to enable the communication other than configuring the network and the user ID that will be used by DFSMS. Example 5-13 shows how to create a local user ID in the DS8000 using DSCLI.

Example 5-13 Creating a DS8000 user ID for DFSMShsm to connect

```
dscli> mkuser -pw REDBOOKS -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 used in the **`mkuser`** command is temporary and expired. You have to 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.

After the user ID is created and the password is changed, you will be able to connect the DFSMS to the DS8000 using the steps described in Chapter 6, “Configuring DFSMS for TCT” on page 53.

Attention: The user ID created for the DFSMS to connect to the DS8000 follows the security rules 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 the *IBM DS8880 Integrated Copy Services Manager and LDAP Client on the HMC*, REDP-5356 publication.

5.3.2 External CA versus self-signed certificates and REST API Proxy

Section 4.4.3, “Certificates (if using SSL/TLS)” on page 39 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 Certificate Authority) or self-signed certificates.

Also, if using S3 or IBMCOS cloud types, DS8000 uses a REST API Proxy interface to communicate with DFSMShsm. To do so, DFSMShsm connects to the DS8000 using its HTTPS interface, so the certificate used by the DS8000 must be added to the IBM RACF® for the DFSMShsm authentication to be successful.

We demonstrate how to configure both the self-signed and External CA certificate options in the DS8000 next. The steps required to configure RACF with the certificates are described in Chapter 6, “Configuring DFSMS for TCT” on page 53.

Note: To work with certificate files on the DS8000, you need a user ID with an Administrator role in the DS8000.

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, by completing the following steps:

1. Log on to the DS Storage Manager GUI.
2. Select **Settings** > **Security.**
3. Access the “Communications Certificate” tab.
4. Click **Create Self-signed Certificates**.
5. Enter the information requested regarding your organization.
6. Click **Create.** A warning message is displayed stating that the HMC will be rebooted and any users connected will be automatically logged off.
7. Click **Yes** to continue with certificate creation. After creation, the certificate is automatically loaded at the HMC.

This self-signed certificate must be uploaded to the z/OS host that will connect to the DS8000. First, download the certificate to your workstation using an *openssl* command, as shown in the Example 5-14. (In the example, **“**`DS8000-HMC-IP`**”** is the IP address of the DS8000 HMC that is configured as the REST API Proxy server):

Example 5-14 Downloading the DS8000 HMC self-signed certificate

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

Note: Certain operating systems support the **`openssl`** command natively. Others may require a client to be installed to support the **`openssl`** command.

The procedure to upload the certificate from your workstation to the z/OS host is demonstrated in the section 6.2.1, “Uploading the certificate files to the z/OS host” on page 54 of this book.

Creating and using an External CA certificate on the DS8000

To create an External CA certificate on the DS8000, you may also use the DS Storage Manager GUI to create a Certificate Signing Request (CSR) file that will be signed by the third-party Certificate Authority in the creation of the CA certificate, using the following steps:

1. Log on to the DS Storage Manager GUI.
2. Select **Settings** > **Security**.
3. Access the “Communications Certificate” tab.
4. Click **Create Certificate Signing Requests**.
5. Enter the HMC DNS host name and the information regarding your organization.
6. Click **Create**. The CSR file is created, and you receive a Web-browser download window to specify the destination path to be used to save the file in your computer.
7. Save the file, and send it to the Certificate Authority to generate the certificate.

After the Certificate Authority generates and sends you the final certificate file, you must import it to the DS8000. You can do this through the DS Storage Manager GUI, using the steps below:

1. Log on to the DS Storage Manager GUI.
2. Select **Settings** > **Security**.
3. Access the “Communications Certificate” tab.
4. Click **Import Existing Certificates**. A message window prompts you to select the certificate file from your local computer.
5. Navigate to the folder in your computer where the file is stored, select the certificate file, and click **Import**. A warning message states that the HMC will be rebooted and any users connected will be automatically logged off.
6. Click **Yes** to import and load the certificate at the HMC

6

Configuring DFSMS for TCT

In this chapter, we describe how to configure z/OS DFSMS and HSM to be able to use Transparent Cloud Tiering. We use the *Interactive Storage Management Facility* (ISMF) interface to define the cloud connection to DFSMS.

This chapter includes the following topics:

6.1 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 connection in one of two ways:

- For cloud target types that use the S3 API (AWS. IBMCOS, IBM Cloud object storage services, Generic S3 targets, and TS7700), the DS8000 acts as cloud proxy. It relays the cloud requests from DFSMS / HSM to the actual cloud storage. You connect DFSMS to the DS8000 HMC. DFSMS posts cloud requests using the SWIFT API. The HMC passes the requests on to one of the DS8000 internal servers, which then performs it on the actual 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.

6.2 Adding digital certificates to RACF

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

You set up the secure communication in two steps:

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

6.2.1 Uploading the certificate files to the z/OS host

Take these steps to prepare to send the certificate files to the host:

1. Decide which of the following certificate types you will use:
   - External certificate (from a third-party Certificate Authority)
   - Self-signed certificate
2. Receive this certificate from the certificate administrator. If you use self-signed certificates, you might also be able to download them from the cloud storage (or DS8000) directly.

Now you can upload the certificate files to z/OS. The z/OS data sets must meet the following conditions:

- Data sets containing the certificate must have `RECFM=VB`.
- They must be cataloged.
- They cannot be a PDS or a PDS member.

Example 6-1 shows how you can upload certificate files to a z/OS host using a command line FTP client.

Example 6-1   Uploading certificates to the z/OS host using ftp

```
ITSOUser-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 please.
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 carootcert.pem INTROOT.PEM
local: carootcert.pem remote: INTROOT.PEM
229 Entering Extended Passive Mode (|||1037|)
125 Storing data set IBMUSER.INTROOT.PEM
100% |***********************************|  1434      606.22 KiB/s    --:-- ETA
250 Transfer completed successfully.
1434 bytes sent in 00:00 (11.65 KiB/s)

ftp> put caintermediatecert.pem INTRMED.PEM
local: caintermediatecert.pem remote: INTRMED.PEM
229 Entering Extended Passive Mode (|||1038|)
125 Storing data set IBMUSER.INTRMED.PEM
100% |***********************************|  1838      787.24 KiB/s    --:-- ETA
250 Transfer completed successfully.
1838 bytes sent in 00:00 (14.63 KiB/s)
ftp>
```

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

6.2.2 Adding external CA certificates to RACF

After sending the certificate files to the host, you must add them to RACF. This is done using the **`RACDCERT`** RACF command. For the user to be able to run this command, there are also a few other security requirements to be satisfied:

- The **`RACDCERT`** command needs to be authorized under the `AUTHCMD` list in the **`IKJTSOxx`** parmlib member.
- The user ID that will issue the **`RACDCERT`** command also needs to be authorized on RACF to do so.

You can check the details of the RACF authorization requirements for the RACDCERT command in the IBM Knowledge Center page, at the following link:

https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.3.0/com.ibm.zos.v2r3.icha700/cracd.htm

Example 6-2 shows how you can add the certificate data set to RACF, using the **RACDCERT** command.

Example 6-2 Adding a certificate data set to RACF

```
RACDCERT CERTAUTH ADD(<certificate dataset>) WITHLABEL(’Cloud Certificate') TRUST
```

Also, after adding the certificate file to the RACF, the DIGTCERT class needs to be refreshed for the new configuration to take effect, as shown in Example 6-3.

Example 6-3 Refreshing the DIGTCERT class

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

When you use an external CA, the certificate might be signed by either a root CA or an intermediate CA.

If the certificate was signed by a root CA, you will only need to add the cloud Root CA certificate data set to the RACF, as shown in Example 6-4.

Example 6-4 Adding only the Root CA certificate to RACF

```
RACDCERT CERTAUTH ADD('IBMUSER.INTROOT.PEM') WITHLABEL(’Cloud Root CA') TRUST
SETROPTS RACLIST (DIGTCERT) REFRESH
```

If the certificate was signed by an intermediate CA, you must add both the root CA certificate and the intermediate CA certificate data sets to RACF, as demonstrated in the Example 6-5.

Example 6-5 Adding Root CA and Intermediate CA certificates on RACF

```
RACDCERT CERTAUTH ADD('IBMUSER.INTROOT.PEM') WITHLABEL(’Cloud Root CA') TRUST
RACDCERT CERTAUTH ADD('IBMUSER.INTRMED.PEM') WITHLABEL(’Cloud Intermediate CA')
SETROPTS RACLIST (DIGTCERT) REFRESH
```

6.2.3 Adding self-signed certificates to RACF

To add a self-signed certificate data set to RACF, you must use the `SITE` option of the **RACDCERT** RACF command, as demonstrated in Example 6-6.

Example 6-6 Adding site self signed certificates on RACF

```
RACDCERT SITE ADD('IBMUSER.INTROOT.PEM') WITHLABEL(’Self-Signed-Cert') TRUST
SETROPTS RACLIST (DIGTCERT) REFRESH
```

6.3 Controlling access to the Cloud features

If you use a SWIFT cloud object storage, you need the full cloud credentials to set up the DFSMS cloud 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 good practice to protect the cloud credentials and avoid creating and running DFSMSdss jobs to move data to cloud. In this case, you would have to provide the cloud password in clear text in the job definition (JCL). Anybody with access to your JCL (from SDSF panels or your JCL libraries) could spot the cloud password.

Use DFSMShsm to manage the migration of your data to and from the cloud. DFSMShsm stores an encrypted version of the password in its Control Data Sets (CDSs), which users cannot access.

There are IBM 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 can use DFSMSdss to store data and retrieve data from a cloud if they know the cloud credentials.

6.3.1 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 logical dump
- STGADMIN.ADR.RESTORE.CLOUD applies to logical restore

Example 6-7 shows sample commands that can be used to define these FACILITY class profiles on RACF:

Example 6-7 Define 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
```

6.3.2 Controlling access to DFSMShsm

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

- Enable the CP Assist for Cryptographic Functions (CPACF)
- Define DFSMShsm to z/OS UNIX System Services
- Define SAF resources to control access to the CLOUD

Enable the CP Assist for Cryptographic Function feature

Ensure that the IBM Z feature code 3863 *CP Assist for Cryptographic Functions* (CPACF) is enabled. It enables clear key DES and TDES instructions on all CPs. For more information, see the IBM Redbooks publication, *Getting Started with z/OS Data Set Encryption*, SG24-8410:

http://www.redbooks.ibm.com/abstracts/sg248410.html?Open

HSM needs CPACF to store the encrypted cloud password in its control data sets.

Define 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('/')

Define SAF resources to control access to the cloud

The commands in Example 6-8 define SAF resources that control access to the `CLOUD` keyword on the **`HMIGRATE`** end user command in DFSMShsm, and grant `READ` access to the `STGADMIN.ARC.ENDUSER.HMIGRATE.CLOUD` FACILITY class profile.

Example 6-8 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
```

6.4 Creating a DFSMS cloud connection using ISMF

To allow the definition of cloud object storage targets for DFSMS, a new panel is available in the *Interactive Storage Management Facility* (ISMF). The new cloud definition construct allows you to define the parameters that are necessary to connect to the clouds. The `Cloud` option in the ISMF menu is accessible only when you have access to the administrator mode on ISMF panels. Detailed information is available in *z/OS DFSMSdss Storage Administration*:

https://www.ibm.com/support/knowledgecenter/SSLTBW_2.3.0/com.ibm.zos.v2r3.idas200/cloud.htm

Example 6-9 shows the new Cloud option that is available from the main ISMF menu panel. Depending on your terminal configuration, it might be necessary to scroll down to see it.

Example 6-9 New Cloud option on ISMF panel

```
                    ISMF PRIMARY OPTION MENU - z/OS DFSMS V2 R2
 Selection or Command ===>
                                                                      More:   -
 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  Cloud                     - Specify Cloud Attributes
```

Select the **S** option (Cloud) to open the **Cloud Application Selection** panel (see Example 6-9 on page 58). It gives you the options to list, display, define, or alter cloud definitions. To define a new cloud target, we provide the name (`IBMCOS`) and select option 3, as shown in Example 6-10.

Example 6-10 Defining the IBMREDBOOKS cloud

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

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

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

> **Note:** You must use the same cloud definition name as for the DS8000 cloud connection, that you created according to Chapter 5, “Configuring the DS8000 for TCT” on page 41.

6.4.1 Defining a cloud connection for S3, IBMCOS, or TS7700 cloud targets

For all cloud target types using 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 originating from DFSMS are routed through the DS8000 to the “real” cloud target.

Example 6-11 and Example 6-12 on page 60 show the entries that you have to make for a DS8000 cloud proxy definition.

Example 6-11 First cloud definition panel for DS8000 cloud proxy definition

```
CLOUD DEFINE                   Page 1 of 2

 SCDS Name  . . : SMS.DFSMS.SCDS
 Cloud Name . . : IBMCOS

 To DEFINE Cloud, Specify:

 Description   IBM REDBOOK DEMO IBMCOS CLOUD

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

 Identity  . . cloudproxy

Command ===>
```

Specify the following fields in the first cloud definition panel:

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

Example 6-12 Second cloud definition panel for DS8000 cloud proxy definition

```
CLOUD DEFINE                    Page 2 of 2

 SCDS Name  . . : TCTRBOOK.SCDS
 Cloud Name . . : IBMCOS

 To DEFINE Cloud, Specify:

 Endpoint . . . . https://x.xxx.xxx.xxxx


 Port Number  . . 8452 (0 to 65535)
 SSL Version  . . TLSV12    (TLSV12, TLSV11, TLSV1, SSLV3 or blank)
 SSL Key  . . . . *SITE*/*

Command ===>
```

In the second cloud 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. Currently, the only port supported by the DS8000 is 8452.
- SSL version: the lowest SSL version acceptable to use when making HTTP requests. Maximum length: eight characters, valid values: TLSV12, TLSV11, TLSV1, SSLV3, blank.
- SSL key: the name of the key store to be used (required when SSL version is not blank. The value can be one of the following: a SAF keyring name, in the form of userid/keyring, or a PKCS #11 token in the form of `*TOKEN*/`*`token_name`*. If you plan to use CA certificates, you must specify `*AUTH*/*`. If you plan to use a self-signed certificate, you must specify `*SITE*/*`, as we do in this example.

See the DFSMSdfp Storage Administration manual for more details about setting up a DS8000 as a cloud object proxy.

6.4.2 Defining a cloud connection for a SWIFT cloud object storage target

With cloud object storage solutions using the SWIFT API, DFSMS communicates directly. Again, the cloud definition process consists of two panels. The first panel is shown in Example 6-13.

Example 6-13 First cloud definition panel for SWIFT

```
                                  CLOUD DEFINE                Page 1 of 2

 SCDS Name  . . : SMS.DFSMS.SCDS
 Cloud Name . . : ITSOSWIFT

 To DEFINE Cloud, Specify:
```

```
 Description   IBM REDBOOK SWIFT CLOUD


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

 Identity  . . test:tester


 Command ===>
```

The following fields are available for definition in the first cloud 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 this 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 6-14.

Example 6-14 Second cloud definition panel for SWIFT

```
                                   CLOUD DEFINE                   Page 2 of 2

 SCDS Name  . . : SMS.DFSMS.SCDS
 Cloud Name . . : ITSOSWIFT

 To DEFINE Cloud, 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 cloud definition panel:

- Endpoint: Identifies the Uniform Resource Identifier (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 key store to be used (required when SSL version is not blank. The value can be one of the following: a SAF keyring name, in the form of userid/keyring, or a PKCS #11 token in the form of `*TOKEN*/`*`token_name`*. If you will use CA certificates, you must specify `*AUTH*/*`. If you will use a self-signed certificate, you must specify `*SITE*/*`.

6.4.3 Activating the Storage Management Subsystem configuration

After you completed and saved the cloud configuration (SWIFT or cloud proxy), you must activate the SCDS that contains the cloud definition.

> **Note:** Make sure you activate the correct 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 6-15.

Example 6-15 Selecting the Control Data Set option

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

                                                                More:     +
 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 ===>
```

2. The CDS Application Selection panel is displayed, you should validate your Source Control Data Set (SCDS) by using option **4 Validate the SCDS** before you make it the active CDS, as shown in Example 6-16.

Example 6-16 Validate 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:
   4  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 ===>
```

3. After validating the CDS, select option **5. Activate the CDS** to activate the new configuration.

Example 6-17 Activate 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:
   5  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 ===>
```

4. 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.

The DFSMShsm also needs permission to list the keyrings it has access to. 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 6-18:

Example 6-18 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
```

6.5 Configuration of DFSMShsm for TCT

The final step before you can start migrating data to cloud object storage is the configuration of DFSMShsm. HSM is the application that creates and submits DFSMSdss dump and restore jobs to move data to and from cloud storage. In the DFSMS cloud definition, we specified the endpoint and credentials, but without providing a password. DFSMSdss requests the cloud password each time it is called to move a data set or access the cloud storage for any other reason.

HSM has the ability to receive the cloud password and store it encrypted in its Control Data Sets. To configure DFSMShsm to use a cloud storage, you must use the HSM command **`SETSYS CLOUD`**. The command allows you to perform the following actions:

- Connect HSM to a new cloud definition
- Refresh cloud credentials
- Delete a cloud definition from DFSMShsm CDSs

The **`SETSYS CLOUD`** command that is shown in Example 6-19 defines the cloud to the DFSMShsm, attempts to connect, and if it's successful, stores the cloud related information into the MCDS.

Example 6-19 Defining the cloud to DFSMShsm

```
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 6-20 shows the `WTOR` waiting for a reply on the system log.

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

Example 6-20 WTOR generated by SETSYS CLOUD command

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

The cloud password can contain upper and lower case characters. Therefore, we recommend that you reply to this `WTOR` from the SDSF System Command Extension, as shown in Example 6-21. Replying from the `SDSF SYSLOG` forces a reply to upper case, which can result in a failed authentication.

Example 6-21 Case-sensitive password

```
                      System Command Extension

===> REPLY 29,’PaSswOrd’
```

Note: You get the System Command Extension from SDSF by typing a "/" (slash) character in the command line and pressing **`Enter`**. Do not forget the quotes around your password. Otherwise it will be converted to upper case.

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 related to the connection error, as shown in Example 6-22.

Example 6-22 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 entered is wrong (it might also 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 control data sets.
- **SETSYS CLOUD(NAME(xxxxx) REFRESH)**: Use this command to refresh the cloud “xxxxx” credentials to DFSMShsm CDS, including the password that was used to connect to the cloud.
- **SETSYS CLOUD(NAME(xxxxx) PASSWORD)**: Use this command to create a `WTOR` requesting the cloud password. After the new password is suppressed, it is encrypted and stored on DFSMShsm CDS.

It is possible to have up to seven cloud definitions stored on DFSMShsm CDS. If it is needed to set up a new one after having seven cloud definitions already configured, you must first delete one definition before creating the new one. To identify the clouds currently configured to DFSMShsm, you can issue the command shown in Example 6-23, and search for configured clouds. The first cloud name starts on byte `x'45C'`.

Example 6-23 Displaying Cloud information

```
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 AB0ACD01 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 password for HSM without having defined and activated the cloud structures in DFSMS yet, you will receive an error message like that shown in Example 6-24.

Example 6-24   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 functionality to communicate with the cloud, send data to it, and retrieve data from it.

This part includes the following chapters:

- Chapter 7, “DFSMShsm” on page 69
- Chapter 8, “Using automatic migration” on page 81
- Chapter 9, “Operational integration and reporting considerations” on page 87

7

DFSMShsm

In this chapter, we describe the changes that made to DFSMShsm to support the cloud tier.

This chapter describes the following topics:

7.1 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 DASD and tape requirements, and provide disaster recovery 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 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 free 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 CPU resources 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 HSM tapes during the recycle operation.

Also, the option of having an off-premise cloud increases the options that are available for disaster recovery. 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, we suggest that you run an **AUDIT** against CDS and Cloud to report and correct any mismatches between them.

7.2 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 7-1.

Example 7-1 Default DFSMShsm container name

```
SYSZARC.<HSMplexname>.MIG.yyyyddd
```

By default, a container is created every 7 days. Creating a container every few days allows you to better manage the objects within a container, list migrated data, and **AUDIT** the cloud.

If you decide an alternative value is required for the frequency of container creation that is based on listing performance considerations, you can change the default value from 7 days to shorter or larger amounts. Example 7-2 shows the **PATCH** command to change the default value from 7 to 10 days.

Example 7-2 PATCH to change container creation frequency

```
PATCH .MCVT.+50F X'09'
```

Just as DFSMShsm can automatically create containers when required, it can also delete empty and no longer used containers 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 7-3 shows a sample **SETSYS** command to allow one Cloud Storage containers processing task. This is the default value. To prevent DFSMShsm from deleting empty containers, set the EMPTYCONTAINERDELETION to 0.

Example 7-3 Setting container deletion task

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

If you disable automatic deletion of containers by DFSMShsm, you will need to create your own process to manage empty containers.

7.3 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.

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 ML1 migrate data sets. The object naming convention is shown in Example 7-4.

Example 7-4 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 is 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.

7.3.1 Migration

When a migration process is started, DFSMShsm calls DFSMSdss to perform the data movement. HSM is responsible for passing to 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 (COS) 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 VSAM data sets.

7.3.2 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 COS 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.

7.4 Fast Subsequent Migration

When data sets are migrated to 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 7-5) is available to include in your DFSMShsm parmlib to allow the reconnect.

Example 7-5 Set up fast subsequent migration

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

7.5 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.

7.5.1 Command-driven migration

A **CLOUD** parameter is available from the **MIGRATE** or **HMIGRATE** commands to target data sets to cloud. Example 7-6 shows a sample **HSEND MIGRATE** command with the **CLOUD** keyword.

Example 7-6 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 7-1.

Table 7-1 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 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 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 DS8000s.

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

Example 7-7 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 7-8.

Example 7-8 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 7-9. Regardless of the migration tier, migrated data sets are always cataloged by using volser `MIGRAT`.

Example 7-9 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 Dataset (MCD) record is shown in Example 7-10.

Example 7-10 FIXCFDS command to view a CDS record

```
HSEND FIXCDS D dsname DISPLAY
```

The command output for a cloud-migrated data set is shown in Figure 7-11. Areas are highlighted to show recognizable eye catchers within the CDS record.

Example 7-11 Output of FIXCFDS command

```
F DFHSMBC,FIXCDS D TCT.DEMO.DTA6.TRS DISPLAY
MCH=   02580000 D53C0FE6 68920609 D53B23A7 A3FBB90A                              *    N  W    N               *
+0000  6CC3D3D6 E4C48001 00340000 0118318F 00000000 0118319F 00000000 00000000   * 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 00000000   *                            *
+0080  00000000 00000000 00000000 00000000 00000000 00000000 0000000F C4C6C8E2   *                        DFHS*
+00A0  D44BC8D4 C9C74BE3 F1F9F4F6 F0F94BE3 C3E34BC4 C5D4D64B C1F8F3F1 F9404040   *M.HMIG.T194609.TCT.DEMO.A8319   *
+00C0  40404040 40404040 00000000 00000000 00000000 00000000 00000000 00000000   *                            *
+00E0  00000000 00004040 40404040 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   *          IBMCOS                *
+01E0  40404040 40404040 E2E8E2E9 C1D9C34B C1D9C3D7 D3C5E7F0 4BD4C9C7 4BF2F0F1   *        SYSZARC.ARCPLEXO.MIG.201*
+0200  F8F3F1F6 40404040 40404040 40404040 40404040 0000000A                     *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.

7.5.2 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 8, “Using automatic migration” on page 81.

7.5.3 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 9, “Operational integration and reporting considerations” on page 87

7.6 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 7-12 shows the recalled data set from the **HSEND MIGRATE** command issued, where the recalled data set has a single extent.

Example 7-12 Recalled data set with consolidated extent

```
DSLIST - 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.

7.7 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 gives 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 7-13.

Example 7-13 LIST command for a specific cloud

```
HSEND LIST CLOUD(IBMCOS)
```

The command output is shown in Example 7-14.

Example 7-14 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 4 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 7-15.

Example 7-15 List 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 7-15 is shown in Example 7-16.

Example 7-16 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 7-17. This option brings all of the data and metadata objects that are stored under the specific prefix.

Example 7-17 LIST objects by using the PREFIX keyword

```
HSEND LIST CLOUD(IBMCOS) CONTAINER(SYSZARC.ARCPLEXO.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 7-18.

Example 7-18 LIST objects by PREFIX

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

CLOUD NAME                     CONTAINER NAME                            OBJECT NAME

IBMCOS                        SYSZARC.ARCPLEXO.MIG.2018316             DFHSM.HMIG.T194609.TCT.DEMO.A8319/DTPDSNL000
                                                                          00001
                                                                         DFHSM.HMIG.T194609.TCT.DEMO.A8319/HDR
                                                                         DFHSM.HMIG.T194609.TCT.DEMO.A8319/TCT.DEMO.D
                                                                          TA6.TRS/APPMETA
                                                                         DFHSM.HMIG.T194609.TCT.DEMO.A8319/TCT.DEMO.D
                                                                          TA6.TRS/DTPDSHDR
                                                                         DFHSM.HMIG.T194609.TCT.DEMO.A8319/TCT.DEMO.D
                                                                          TA6.TRS/DTPVOLD01/NVSM/EXTENTS
                                                                         DFHSM.HMIG.T194609.TCT.DEMO.A8319/TCT.DEMO.D
                                                                          TA6.TRS/DTPVOLD01/NVSM/META
                                                                         DFHSM.HMIG.T194609.TCT.DEMO.A8319/TCT.DEMO.D
                                                                          TA6.TRS/DTPVOLD02/NVSM/EXTENTS
                                                                         DFHSM.HMIG.T194609.TCT.DEMO.A8319/TCT.DEMO.D
                                                                          TA6.TRS/DTPVOLD02/NVSM/META
                                                                         DFHSM.HMIG.T194609.TCT.DEMO.A8319/TCT.DEMO.D
                                                                          TA6.TRS/DTPVOLD03/NVSM/EXTENTS
                                                                         DFHSM.HMIG.T194609.TCT.DEMO.A8319/TCT.DEMO.D
                                                                            TA6.TRS/DTPVOLD03/NVSM/META
```

For more information about each object, see Table 3-1 on page 24.

7.8 Audit

The tasks to audit DFSMShsm data are vital to keep CDS records error free, 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 DATASETNAMES, **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 7-19.

Example 7-19 Audit a specific cloud

```
HSEND AUDIT MEDIACONTROLS(CLOUD(IBMCOS))
```

The output from the command that is shown in Example 7-19 is shown in Example 7-20, with one inconsistent entry.

Example 7-20 Example of Audit output

```
AUDIT MEDIACONTROLS(CLOUD(IBMCOS)) 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.

7.9 REPORT command

After you first implement a storage cloud to your z/OS systems, it is suggested that you create reports about key metrics for analysis, such as these:

- Number of data set migrations to the cloud,
- Amount of data transferred,
- Number of successful/failed requests, and
- 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 7-21.

Example 7-21 Report command for daily migration activity

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

It is also possible to retrieve recall specific information by issuing the **REPORT** command, as shown in Example 7-22.

Example 7-22 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 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 7-23.

Example 7-23 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

                     NUMBER  ------READ--------  -----WRITTEN------  ---
    HSM FUNCTION    DATASETS TRK/BLK    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 9, “Operational integration and reporting considerations” on page 87.

8

Using automatic migration

In this chapter, the automatic migration function and cloud usage are described.

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 CPU savings related to data migration in DFSMShsm.

This chapter describes the following topics:

8.1 SMS support for automatic migration

To enable the DFSMShsm automatic migration to cloud, it is necessary to define the policies to be used to define if a data set will be migrated to existing ML2 volumes, or the defined cloud storage configured to your systems.

The definitions about data set migration are included in the SMS Management Class construct, and 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.

8.1.1 Management Class updates

The data sets cannot be automatically migrated from ML2 to cloud the same way that 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 will migrate 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 will be migrated to the cloud storage specified by the Cloud Name field as long as the Primary Days Non-usage value is met and the data set still resides 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 will be 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 will perform 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 will be 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 will perform 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 very 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 using these values in the exit, and you can disable the exit.

Table 8-1 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 will override the values used in the management class construct.

`EXITON` 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.

More exit information is available in the following link:

`https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.3.0/com.ibm.zos.v2r3.arcf000/exit.htm`

8.2 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 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 will be managed by a given HSM host. Example 8-1 shows a sample SETSYS command to create the affinity between storage group BATCH1 and the HSMIMG2 DFSMShsm image.

Example 8-1 Sample STORAGEGROUPAFFINITY command

```
F HSMIMG2,SETSYS STORAGEGROUPAFFINITY(BATCH1)
```

By setting up the storage group affinity, the specific HSM usage will manage the defined storage groups. Define this command in your `ARCCMD` parmlib member to save your settings across Initial Program Load (IPLs).

The **`SETSYS MIGRATIONAUTOCLOUD`** parameter enables the host to execute Cloud migration, and is an optional parameter that defines the possible target tier during automatic migration. The default value of this parameter is `ALL`:

- 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 8-2.
- NOCLOUD: Specifying that only data set migration to ML1, ML2, and transition are performed.

Example 8-2 Specifying that only data set migration to Cloud storage is performed

```
SETSYS MIGRATIONAUTOCLOUD(CLOUDONLY)
```

8.3 Space management functions

This section describes the space management functions for managing SMS-managed storage, 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 associated with each data set on the volume or the expiration date 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 have exceeded 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 transitioned. If a data set is eligible for both a class transition and migration, the data set will only be migrated. This 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 free space. However, if deletion of expired data sets, fast subsequent migration, and space reduction of the remaining data sets achieves the specified volume free space low threshold, no actual data sets are moved.

If data sets are to be migrated during primary space management, they will be migrated in compacted form if possible. Data sets 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 has been reached and not all containers have been 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 8.2, “Storage Group affinity enhancements” on page 83.

9

Operational integration and reporting considerations

In this chapter, we review operational integration considerations.

The Storage cloud setup is the first stage in the process 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:

9.1 Pre-implementation reporting

Before implementing automatic migration to the cloud, a storage administrator might want to report on possible Central Processing Unit (CPU) savings 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 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.

9.2 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 as long as possible, and that any errors are tracked by automation systems.

9.2.1 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 9-1.

Example 9-1 ARC1581I message

```
Display  Filter  View  Print  Options  Search  Help
 -------------------------------------------------------------------------------
 SDSF SYSLOG     33.101 3090 3090 10/10/2016 OW        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/ ERRTEXT HTTP/1.1 500 Internal Server Error
```

9.2.2 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 9-2.

Example 9-2 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.

9.2.3 Monitoring reconnections

If you implemented periodic checks of the 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 cloud-reconnect setting is shown in Example 9-3.

Example 9-3 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
```

9.2.4 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.

9.3 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.

9.3.1 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 9-4.

Example 9-4 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 9-5. You can use this code as a base to develop your own specific reports.

Example 9-5 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.

9.3.2 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 created and owned by DFSMShsm in the cloud, and the number of objects stored. This might be specially valuable for large cloud environments, where list commands can take an extended amount of time to complete.

Redbooks

IBM DS8000 and Transparent Cloud Tiering

(0.2”spine)
0.17”<->0.473”
90<->249 pages

SG24-8381-02

ISBN 073845852x

Printed in U.S.A.