Frank Feeney
Writing Samples

Configuring client-side encryption for transparent cloud tiering

Client-side encryption for transparent cloud tiering ensures that data is encrypted before it is transferred to cloud storage. The data remains encrypted in cloud storage and is decrypted after it is transferred back to the DS8000 storage system.

Transparent cloud tiering supports the Key Management Interoperability Protocol (KMIP) only.

Setting up client-side encryption for transparent cloud tiering

To set up client-side encryption for transparent cloud tiering, you must configure a key manager and create an encryption key group for transparent cloud tiering. Complete the following steps:

1. Use the **mkkeygrp** command to create an encryption key group that is configured for transparent cloud tiering.
2. Use the **mkkeymgr** command to configure a key server for transparent cloud tiering and assign the encryption key group to the key server.
3. Use the key group as a parameter of the **mkcloudserver** command to define a cloud storage server connection that uses client-side encryption for transparent cloud tiering.

Note: To add a key group that is configured for transparent cloud tiering to an existing key server, use the **managekeymgr** command.

mkkeygrp

To create a key group that is configured for transparent cloud tiering, use the **mkkeygrp** command with the parameter **-type tct** and a key group value that is a whole number in the range 2 - 16. In the following example, key group 3 is configured for transparent cloud tiering.

```
mkkeygrp -keyprotocol kmip -type tct -name demo 3
```

For more information, see mkkeygrp.

mkkeymgr

To configure a key server for transparent cloud tiering and assign a key group, use the **mkkeymgr** command. Use the parameters **-keygrp** (to specify a key group that is configured for transparent cloud tiering) and **-type tct**. In the following example, a key server is configured for transparent cloud tiering with key group 3.

```
mkkeymgr -serverport 1234 -cert /home/hscroot/certs/cloud.crt -type tct -keygrp 3
-addr host.keymanager.com -keyprotocol kmip 4
```

For more information, see mkkeymgr.

mkcloudserver

To define a cloud storage server connection that uses client-side encryption for transparent cloud tiering, use the **mkcloudserver** command with the **-keygrp** parameter. In the following example, a key group value of 3 is assigned to the cloud server.

```
mkcloudserver -type ibmcos -username cosusername -pw cospasswd -endpoint
https://1.11.111.111 -loc ztct -keygrp 3 coschar
```

For more information, see mkcloudserver.

managekeymgr

To add a key group that is configured for transparent cloud tiering to an existing key server, use the **managekeymgr** command with the **-action addgrp**, **-keygrp**, and **-type tct** parameters. You must also provide the key server ID. In the following example, key group 3 is assigned to a key server with an ID value of 4.

```
managekeymgr -action addgrp -type tct -keygrp 3 4
```

For more information, see managekeymgr.

Creating fixed block volumes for open systems hosts

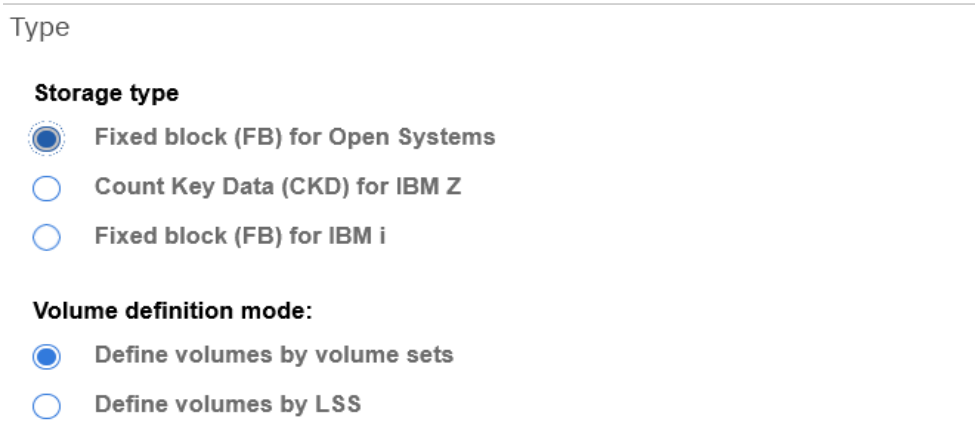
Use the *Create Volumes* window to create fixed block (FB) volumes for open systems hosts. You can configure thin-provisioning, specify the allocation method, and select LSS ranges.

The *Create Volumes* window calculates the maximum number of volumes that are available and the maximum volume size allowed, based on the usable capacity in pools that you select. The storage system automatically balances the volumes between the pools.

The maximum capacity for FB volumes is 16 TiB.

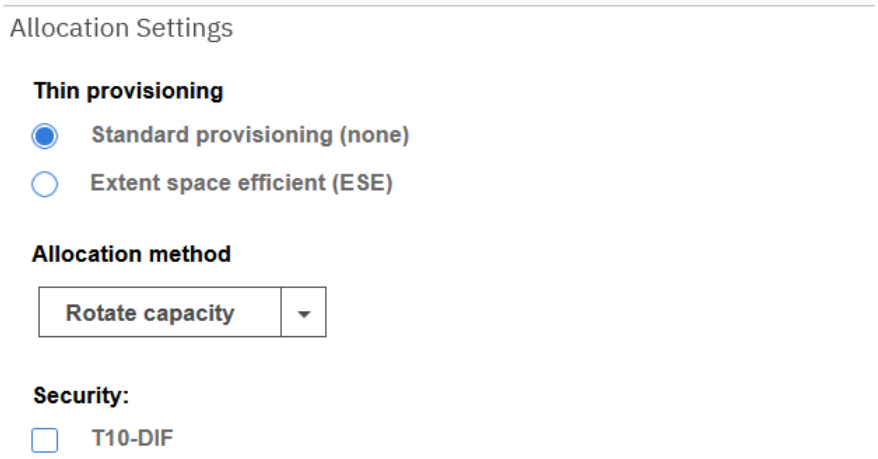
Procedure

1. Open the *Create Volumes* window by clicking **Create Volumes** on the **Volumes** tab.
2. On the **Type** page, select **Fixed block (FB) for Open Systems** as the **Storage Type**. To define an LSS range for the volumes, select the **Define volumes by LSS** as the **Volume definition mode**.



Type page

3. On the **Allocation Settings** page, determine how to provision and allocate the volumes:



Allocation Settings page

Thin provisioning

Select one of the following options:

Standard provisioning

The storage system fully allocates volumes with extents when it creates the volumes.

Extent space efficient (ESE)

Extent space efficient (ESE) volumes allow you to define logical volume sizes that are larger than the usable capacity installed on the storage system. ESE allocates volume capacity based on host-write actions.

Allocation Method

Select one of the following options:

Rotate capacity

The storage system rotates the allocation of extents across arrays to provide optimal performance. The storage system maintains a random sequence of arrays and allocates extents to the next array in the sequence.

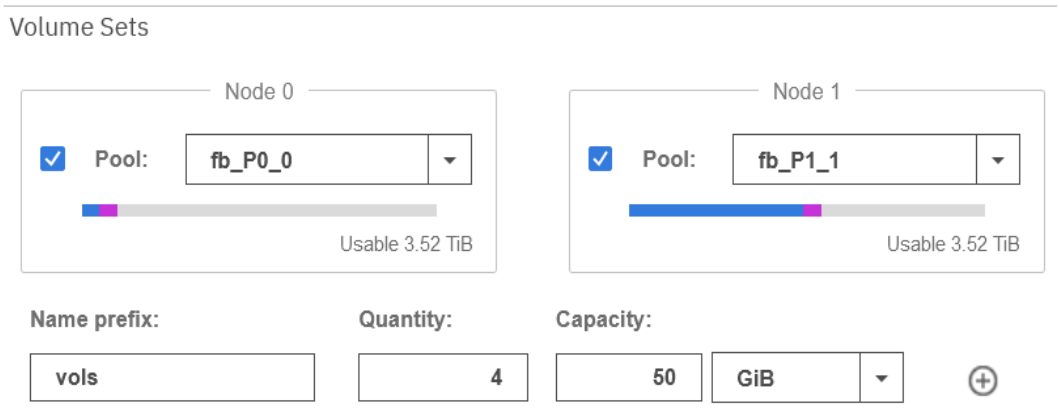
Rotate volumes

Volume extents can be allocated sequentially. In this case, all extents are taken from the same array until there are enough extents for the requested volume size or the array is full. If more than one volume is created in one operation, the allocation for each volume starts in another array. Use this allocation method if you prefer to manage performance manually.

Security

Select **T10-DIF** to use the T10 Data Integrity Feature (DIF/PI), which stores data by using the CRC-16-T10-DIF algorithm.

4. If you selected the **Define volumes by volume sets** option as the **Volume Definition Mode**, use the **Volume Sets** page to define the volume sets:



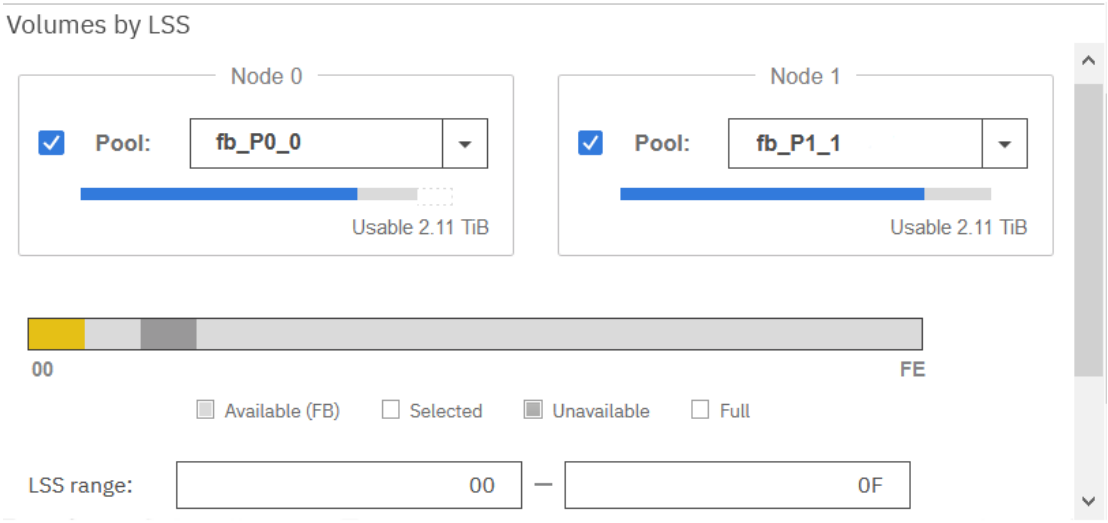
Volume Sets page

a. Select a **Pool** for each node. You can view the capacity details for each pool that you select.

b. Define the volumes. Click the plus or minus icons to add or delete a set of volumes.

- Enter a **Name prefix** to identify the volumes.
- Enter the number of volumes (**Quantity**) to create. The **Volume sets** page informs you if the value exceeds the addresses that are available.
- Specify the **Capacity** of the volumes by entering the size and unit (such as GiB or blocks).

5. If you selected the **Define volumes by LSS** option as the **Volume Definition Mode**, use the **Volumes by LSS** page to define the LSS range and volume sets:



Volumes by LSS page

a. Select the **Pools** for each node. You can choose from the available FB pools in each node.

b. Create **LSSs** to contain the volumes. Enter the hexadecimal values of the **LSS range** to contain the volumes. Up to 255 LSS addresses are available, from 00 - FE.

c. Define **Volumes** in the LSS. You can define a set of multiple volumes within a range of LSS addresses. Click the plus or minus icons to add or delete a set of volumes.

- Enter a **Name prefix** to identify the volumes.
- Enter the number of volumes (**Quantity per LSS**) to include in the LSS. The **Volume sets** page informs you if the number of volumes exceeds the capacity that you specify.
- Specify the **Capacity** of the volumes by entering the size and unit (such as

GiB or blocks).

6. On the **Host** page, select the host or cluster that accesses data in the volumes.

Host page

7. On the **Summary** page, review details about the volumes.
8. Click **Create** to create the volumes.

Configuring TS7700 as a target for transparent cloud tiering

An IBM Z host can use an TS7700 storage system to act as an object store for transparent cloud tiering with a DS8000 storage system. Use this function to move data sets directly from the DS8000 to the TS7700.

Compression for TS7700 data

The DS8000 will compress data for the TS7700 object store if the IBM Z host uses HSM/DSS to compress the target data set.

To enable or disable compression, there are no configuration options within the DS8000. The IBM Z host determines whether the data set needs to be compressed.

Encrypting data in flight

The DS8000 will encrypt data in flight from the DS8000 to the TS7700 object store if the **`mkcloudserver`** command does not include the **`-nossl`** parameter. The data will be decrypted by the TS7700.

Encryption of data in flight from the DS8000 to the TS7700 object store provides AES-256 bit encryption and does not use an encryption key server. The TS7700 requires Feature Code 5281 (Secure Data Transfer) installed on the cluster.

Procedure

1. Use the DS command-line interface (DS CLI) or DS8000 Storage Management GUI to create a local user account to support the DS8000 as a proxy for metadata and administrative commands from z/OS. z/OS requires this username and password as part of its configuration.

   DS CLI

   To create a local user account from the DS CLI, use the `mkuser` command:

   ```
   mkuser -pw password -group monitor clouduser
   ```

   DS8000 Storage Management GUI

   To create a local user account from the DS8000 Storage Management GUI:

   a. Click **Access** > **Users** to open the **Users** page.

   b. Click **Add User** to open the *Add User* window.

   c. Enter user name, role, and password information.

   d. Click **Add**.

2. From the DS CLI, use the `mkcloudserver` command to configure and connect the DS8000 to the TS7700. The **`-primary7700IPs`** and **`-secondary7700IPs`** parameters must align with the TS7700 GRID IP addresses that were configured when setting up the

TS7700 to be an object store. The **-secondary7700IPs** value will enable forking of migrations between two TS7700 clusters to enable data redundancy.

To encrypt data in flight from the DS8000 to the TS7700, do not use the **-nossl** parameter with the **mkcloudserver** command. The data will be decrypted by the TS7700.

Note: The **-syscaloc** parameter is required if the default certificate in the TS7700 is replaced by either a user supplied certificate or a certificate signed by a certificate authority (CA). If the TS7700 is using the default supplied certificate, **-syscaloc** is not required.

The following example uses the **mkcloudserver** command to create a connection to a TS7700 with forking for data redundancy. Encryption of data in flight is not enabled.

```
mkcloudserver -nossl -type TS7700 -primary7700IPs 10.11.12.13
-secondary7700IPs 10.11.13.14 CLOUDNAME
```

After you enter the command, the DS CLI will display the following message:

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

The following example uses the **mkcloudserver** command to create a connection to a TS7700 without forking. The command enables encryption of data in flight and specifies a user or CA signed certificate.

```
mkcloudserver -type TS7700 -primary7700IPs 10.11.12.13,10.11.12.14
-syscaloc /home/user/usercert.pem CLOUDNAME
```

After you enter the command, the DS CLI will display the following message:

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

3. Use z/OS SMS to configure DFSMS. The user name, password, and cloud_name values must match the values that you specified in the **mkcloudserver** command.

4. **Optional:** Use the lscloudserver command to query a TS7700 object store. You can also use the command to verify if encryption of data in flight has been enabled.

```
lscloudserver -l
```

After you enter the command, the DS CLI will display the following information:

```
name          node type  endpoint IP address              location  nossl keygrp
=============================================================================
CLOUDNAME     0    TS7700 -       10.11.12.13,10.11.12.14  -         false -
CLOUDNAME     1    TS7700 -       10.11.12.13,10.11.12.14  -         false -
```

Configuring safeguarded virtual capacity

You can configure safeguarded virtual capacity to provision and accommodate safeguarded backups on your storage system.

Safeguarded virtual capacity is the amount of volume capacity that is configured to store safeguarded backups for a safeguarded source. The capacity that is required depends on the size of the source volume, the number of backups, and the predicted destage rate of the source volume's data.

To configure the amount of virtual capacity that is required for safeguarded backups, you must first determine an estimate for the safeguarded capacity multiplier. The safeguarded capacity multiplier is multiplied with the capacity of the data you want to back up to create the safeguarded virtual capacity.

Creating virtual capacity

To create safeguarded virtual capacity with the safeguarded capacity multiplier, click **Safeguarded** > **Configure Capacity** on the **Volumes** page.

To create safeguarded virtual capacity with the safeguarded capacity multiplier from the DS CLI, use the **`-action mksafeguardedcap`** parameter with the `manageckdvol` command for count key (CKD) data or the `managefbvol` command for fixed block (FB) data.

Expanding virtual capacity

To expand safeguarded virtual capacity with the safeguarded capacity multiplier, click **Safeguarded** > **Expand Capacity** on the **Volumes** page.

The new capacity multiplier value is based on the overall virtual capacity that you require. For example, if the original capacity multiplier value was 12 and your capacity calculation determines that you need to expand capacity by one third, the new capacity multiplier value would be 16.

To expand safeguarded virtual capacity with the safeguarded capacity multiplier from the DS CLI, use the **`-action expandsafeguardedcap`** parameter with the **`manageckdvol`** command for CKD data or the **`managefbvol`** command for FB data.

Determining an estimate for the safeguarded capacity multiplier

1. Decide the number and retention period of the backups that are maintained by your safeguard schedule.

   For example, if your safeguard schedule creates one backup per hour and holds each backup for up to 3 days, the system will need to maintain 72 backups per volume.

2. Determine the destage rate of the data that you want to back up by using the cachetrans metric, which shows a count of destaged tracks.

   The cachetrans metric is available with the `showckdvol` command for CKD data and the

`showfbvol` command for FB data.

To obtain a clear history of the destage rate on your system, query the cachetrans metric periodically, such as every 30 minutes, and gather about a week's worth of destage data. To ensure that the virtual capacity will accommodate your backups, use the number of destaged tracks from a peak period that matches the period of your safeguard schedule.

For example, if your safeguard schedule period is 72-hours, you might find 15,000,000 destaged tracks within a peak period.

3. Convert the destage rate to extents by dividing the destaged tracks by the number of tracks per extent.

   For example, if the destage rate is 15,000,000 tracks and the number of tracks per extent is 256, then 58,594 extents (15,000,000/256), or approximately 890 GiB, would be required for the safeguarded backups.

   The number of tracks per extent depends on both the extent and volume type:

   - For CKD small extents (21 cylinders), the number of tracks per extent is 315.
   - For CKD large extents (1113 cylinders), the number of tracks per extent is 16695.
   - For FB small extents (16 MiB), the number or tracks per extent is 256.
   - For FB large extents (1 GiB), the number or tracks per extent is 16384.

4. Determine the safeguarded capacity multiplier by dividing the number of extents required for safeguarded backups by the number of extents in the volume.

   For example, if the source volume size is 500 GiB (32000 extents), and the number of extents required is 58,594, then the safeguarded capacity multiplier would be 1.83 (58594/32000).

Considerations:

- The calculation might over-estimate the safeguarded capacity multiplier, as it does not account for tracks that are destaged multiple times within one backup time period.
- The calculation tends to be most accurate for configurations where there is a shorter time period between backups.
- The maximum required value of the safeguarded capacity multiplier is the number of backups. If you are using a small number of backups with a high retention period, use the number of backups as the multiplier if the calculated multiplier is a higher value.
- The maximum safeguarded capacity for a volume is 14.6 TiB for CKD data and 16 TiB for FB data. If the multiplier results in a virtual capacity that is higher than the maximum capacity, the retention period might need to be reduced for the workload.