SAP HANA ™ Fiber Channel Storage Connector Admin Guide

To use block storages together with SAP HANA, appropriate re-mounting and IO fencing mechanisms must be employed. SAP HANA offers a ready to use Storage Connector.

SAP HANA Development Team

V1.8, January 2022

Table of Contents

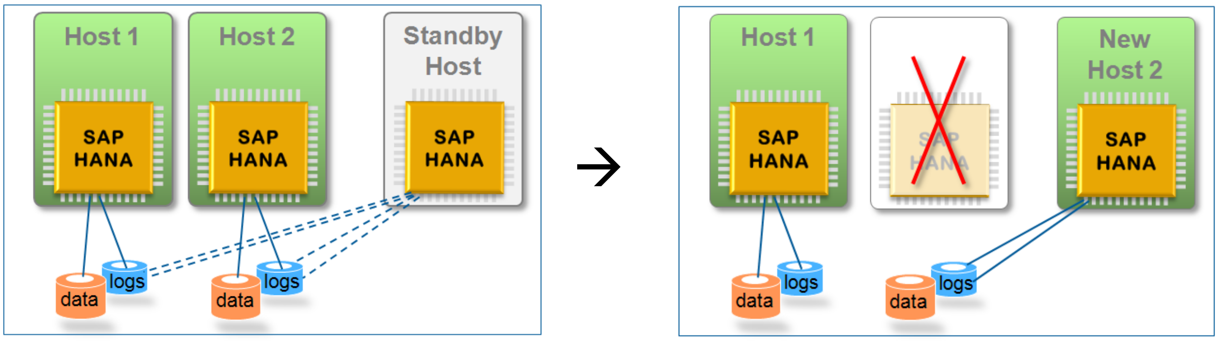
Change history

Version	Date	Description
1.0	April 2013	- Initial version
1.1	August 2013	- Updated “Preparing `global.ini`” section with information about the HANA available installation procedures
1.2	November 2013	- Added information about SAP HANA 1 SPS07’s LVM (logical volume management) support - Updated reservation type discussion - Fixed mountoptions syntax - Added SCSi-3 persistent reservation usage section
1.3	May 2014	- Updated attach/detach description due to code changes - Reworked “Device configuration” section - Minor fixes
1.4	June 2015	- Section for Dynamic Tiering option (new in HANA 1 SPS 10) - Content distinction regarding `/hana/shared` - HANA Lifecycle: Unmount behavior when stopping or killing Hosts - Error message entry: reservation conflict in /var/log/messages - Error message entry: API version mismatch for DT - What if: Mounts remain after shutdown
1.5	December 2015	- Section for AFA option (new in HANA 1 SPS 11) - Using LVM: advantages, naming restriction and a configuration example - Error message entry: Mount/Unmount blocks during system start
1.6	December 2016	- Parallel mount (new in HANA 2 SPS 00) - Error message entry: unable to resolve LVM device ‘dm-XX’ to LUNs - Added explicit LVM explanation in Dynamic Tiering section - Added section for `lvm.conf`
1.7	June 2017	- Restrict `lvm.conf` to distributed systems - Plain LUN requirement also valid for LVM setups
1.8	January 2022	- mpathpersist variants of storage connector implementations

SAP HANA Host Auto-Failover Requirements

For the basic concepts of SAP HANA's Host Auto-Failover, please refer to the *SAP HANA High Availability White Paper [1]*.

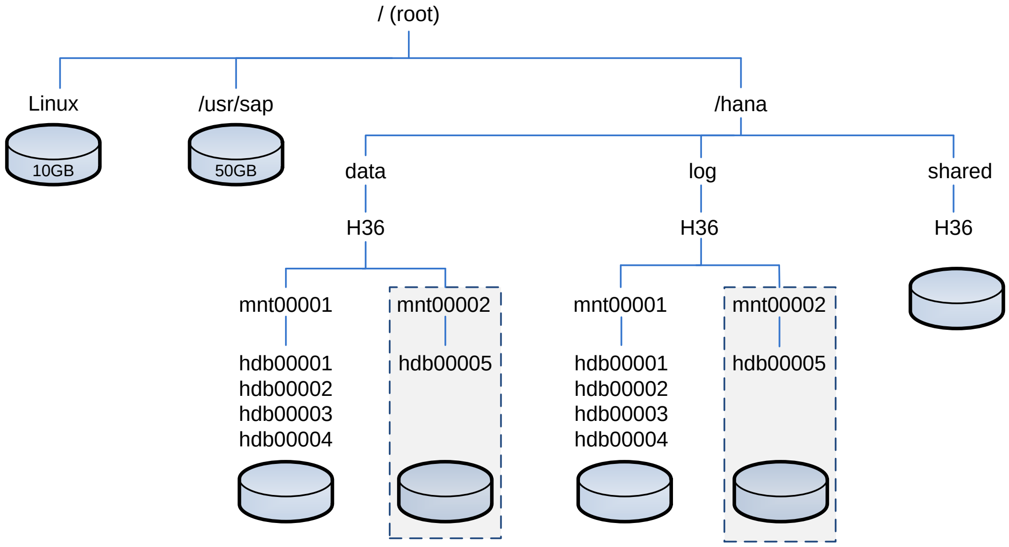
If a SAP HANA host fails in a distributed system, the standby host takes over the persistence of the failing host. In a block storage environment this can only be done by re-mounting the associated LUNs together with proper fencing. This is shown in the image below where the standby host on the left becomes the new host 2 shown on the right, after the failure event.



For every start and failover of a SAP HANA node, a lock of the LUNs is acquired by writing a SCSI-3 Persistent Reservation to the devices and afterwards, the LUNs are mounted to the host.

SAP HANA offers a ready to use Storage Connector Client for setups with native multipathing of Fiber Channel attached devices, which enables Host Auto-Failover on block storages.

The following figure represents the file system structure of a SAP HANA setup. This guide only covers the storage directories which are found in `/hana/data` respectively `/hana/log`.



Files in `/hana/shared` as HANA binaries or runtime traces are still located on a shared file system like NFS.

About the Storage Connector API

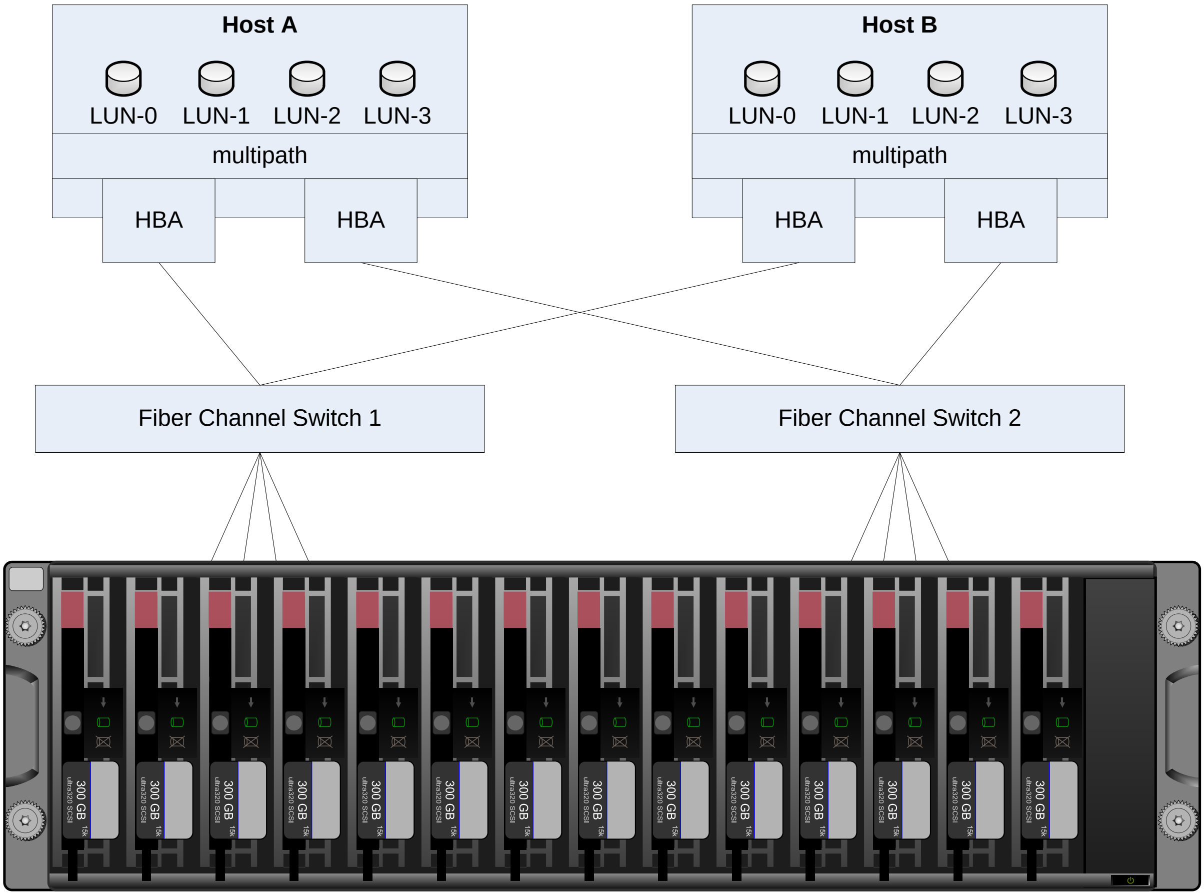
The Fiber Channel Storage Connector is a ready to use implementation of SAP HANA's Storage Connector API. This API provides hooks for database startup and for failing-over nodes. If the Fiber Channel Storage Connector (referred to as `fcClient/fcClientLVM`) does not suffice, for example, because of a shared storage approach or the lack of SCSI-3 persistent reservations, a custom Storage Connector can be implemented. Implementing a custom Storage Connector is outside the scope of this document.

The provided hooks are Python calls from inside the SAP HANA nameserver. A base class defines the interface, which is expected by the name server and, additionally, provides helper functions; for example, for reading the configuration and mounting. A Storage Connector like the `fcClient/fcClientLVM`, implements this interface.

General warning: When working with block storage in a SAP HANA environment, care must be taken at all times. Releasing reservations and incautious mounting of LUNs can lead to severe damage and data loss!

How SCSI-3 Persistent Reservations are used by the Storage Connector

SCSI-3 Persistent reservations are a standard for a wide range of storage subsystems with multiple ways to ensure I/O fencing. Basically, a key must be registered on a device, which is used to make a reservation. A reservation can only be made by the host, which previously registered this key. Finally, only if a device is reserved, the I/O fencing is active. The following figure shows an ordinary – high available – storage connection via Fiber Channel:



There are two hosts A and B, which have two host bus adaptors (HBA) installed each. The first HBA is connected to fiber channel switch 1 and second HBA is connected to fiber channel switch 2. Both switches have four connections to the storage subsystem in place. From the host's perspective this means that each LUN (represented as `/dev/mapper/<wwid>` device) is available through eight paths (represented as `/dev/sd*` devices), which are managed by the multipath daemon within the Linux kernel. The SCSI-3 persistent reservation tool `sg_persist` directly works on the single paths.

SAP HANA's Storage Connector basically uses following three-step approach to ensure fencing (for details see section "Attaching a LUN"). All hosts in a distributed environment use a host-specific key. Each LUN can have more than one key registered, but only one key can be used for a reservation.

1. Before mounting a LUN, the key is registered on the device (another host might have another key registered)
   `sg_persist --register …`
2. Check if another host holds a reservation:
   `sg_persist --read-reservation …`
3. If…
   a. … no reservation is active:
      `sg_persist --reserve …`
   b. … another host's reservation is active
      `sg_persist --preempt …`

Configuration of the Fiber Channel Storage Connector

This section explains how the Fiber Channel Storage Connector can be set up. SAP HANA always comes with the newest version of the `fcClient/fcClientLVM/fcClientMpath/fcClientLVMMpath`, therefore only a few configuration steps have to be done.

Preparing global.ini

There are two options to configure HANA to use Fiber Channel attached storages:

1. Installing SAP HANA from scratch giving the prepared global.ini as input to the installer. See section Installation for more details. This is available since HANA 1 SPS 06.
2. Pre-mounting LUNs of the first host, installing HANA master server, setting up `SYS/global/hdb/custom/config/global.ini`, adjusting `/etc/sudoers`, restarting HANA and finally adding further hosts. **This way should not be used unless it is necessary due to problems during installation.**

To use the fcClient, the following lines need to be added to `global.ini`:

```
[storage]
ha_provider = hdb_ha.fcClient
partition_1_data__wwid = <wwid1>
partition_1_log__wwid = <wwid2>
partition_2_data__wwid = <wwid3>
partition_2_log__wwid = <wwid4 >
…
```

Explanation:

- The `ha_provider` line tells SAP HANA to call the fcClient on startup and failover. To use SAP's Storage Connector, `hdb_ha.fcClient` or `hdb_ha.fcClientLVM` (HANA 1 SPS 07) can be chosen.
- The lines below follow the schema:
  `partition_<partition number>_<usage type>__<param name> = <value>`
  - `<partition number>`: an integer referring to the HANA partition number
  - `<usage type>`: `data` or `log`
  - `<param name>`: any parameter name for the Storage Connector.
    - The fcClient supports:
      - `wwid`[1] or `alias`
      - `prType`[2]
      - `mountoptions`
    - The fcClientLVM supports:
      - `lvmname`
      - `prType`
      - `mountOptions`
  - Wildcards (*) are allowed
    Note: A double underscore (__) in front of `<param name>` must be used.

[1] "**W**orld **W**ide **ID**entifier": a name identifier that is unique worldwide and that is represented by a 64-bit value that includes the IEEE-assigned organizationally unique identifier (OUI)

[2] Specifies the --prout-type parameter of the sg_persist command

- The `partition_[*]_[data|log]__wwid` lines define the names of the LUNs to be used.
- For user-friendly names, you can use: `partition_[*]_[data|log]__alias = <alias>`
- For the fcClientLVM, `partition_[*]_[data|log]__lvmname = <devicename>` must be used: the device name is the one that is shown under `/dev/mapper`. This is “`<volumegroup name>-<logical volume>`”
- With `partition_*_*__prType = [5|6]` it can be controlled how persistent reservations will influence the accessibility of the devices to other hosts. 6 denies external read and write access, 5 denies write only. The default is 6. A detailed discussion on what type to use, can be found in Reservation Types.
- Additionally, you can specify `mountOptions` for single devices or a series of devices with `partition_[*]_[data|log]__mountoptions = <value>` leading to following command: `mount <value> <device> <path>`
  Example: `partition_*_log__mountOptions = -t xfs` ➔ `mount -t xfs <device> <path>`

Choosing the Reservation Type

The fcClient/fcClientLVM supports two different persistent reservation types:

- Exclusive Access (`--prout-type=6`): blocks any external read or write access on a device
- Write Exclusive (`--prout-type=5`): allows reading from, but not writing to a device

Both have advantages and disadvantages:

	Advantages	**Disadvantages**
Exclusive Access	• *Safety*: No other host can see this device • *Safety*: Administrators will also not be able to manually access the devices during normal operation	• *Potential Issue*: if “extended boot logging” is enabled, a server reboot may cause the machine to hang • *Bug*: system message file is filled with a huge number of messages about reserved conflicts (workaround is in place, but may still occur during manual maintenance) • *Bug*: Multipath daemon can hang in some situations
Write Exclusive	• *Operations*: Reboot problems are not an issue • *Operations*: Message file is not filled with useless messages • *Maintainability*: Administrator’s inspection on another host possible	• *Maintainability*: Read-Accessibility of devices might cause confusion, since the device seem to be available to administrators

The default value is Exclusive Access (`--prout-type=6`). Starting with HANA 1 SPS 06, revision 64, the Write Exclusive reservation support was improved. It is recommended to switch to value 5 if there are any problems as shown above occur with type 6.

When using fcClientLVM, reservation type 5 is mandatory, because the LUNs must be visible to the LVM in order to read metadata from it. The fcClientLVM does not work if the metadata is not accessible.

Additional information is available in SAP Note 1941776.

multipath.conf

The settings in the `/etc/multipath.conf` are mostly independent from the fcClient/fcClientLVM. When using the reservation type 6 (exclusive access), the only requirement is to set the following parameters for each LUN.

```
no_path_retry      0
       features    "0"
```

- `no_path_retry`: If a reservation is active, all other hosts must fail upon sending any IO to this device. If not set, the IO will be queued causing the system to wait until the reservation is released. This would cause SAP HANA not to be able run.
- `features`: On some storage subsystems `no_path_retry` will not change the outputs of `multipath -ll` to the correct value ("features 0"), which might lead to confusion.

When using reservation type 5, it is recommended to raise `no_path_retry` to a value greater than 0 (or "queue") in order to be less prone to errors on storage side.

For more information about `multipath.conf` settings, please contact your storage vendor.

sudoers()

Within the fcClient/fcClientLVM script, there is a static method providing the current requirements for the */etc/sudoers* file. The <sidadm> user must be able to issue the appropriate fencing and mounting commands. When using the fcClient/fcClientLVM together with the SAP HANA installer, these settings are done automatically. If an existing system is configured to use block storage, the return value of the method must be read manually and put into the `/etc/sudoers` on **all** hosts for the SAP HANA <sidadm> user.

Example I (HANA 2 SPS00's fcClient):

```
<sidadm> ALL=NOPASSWD: /sbin/multipath, /sbin/multipathd,
/etc/init.d/multipathd, /usr/bin/sg_persist, /bin/mount, /bin/umount,
/bin/kill, /usr/bin/lsof, /usr/bin/systemctl, /usr/sbin/lsof
```

Example II (HANA 2 SPS00's fcClientLVM):

```
<sidadm> ALL=NOPASSWD: /sbin/multipath, /sbin/multipathd,
/etc/init.d/multipathd, /usr/bin/sg_persist, /bin/mount, /bin/umount,
/bin/kill, /usr/bin/lsof, /sbin/vgchange, /sbin/vgscan,
/usr/bin/systemctl, /usr/sbin/lsof
```

Device Configuration

Using plain LUNs

Partitions are not supported, i.e. the whole LUN must be formatted.

Using LVM

The device mapper LVM grants additional flexibility in terms of sizing. Volume groups respectively logical volumes managed by LVM can be resized on demand. By combining several LUNs, more storage space than the upper size limit of a plain LUN (typically 16 Terabyte) can be used for a single partition. Partitions on LUNs managed by LVM are not supported.

To use LVM, each logical volume must be associated with a **unique** set of underlying physical volumes. This is achieved by configuring exactly one logical volume to one volume group. An example for a 2+1 HANA system:

LUNs	Volume Group	Logical Volume	Usable device for fcClientLVM
/dev/mapper/data1_1	hanadata1	vol	/dev/mapper/hanadata1-vol
/dev/mapper/data1_2			
/dev/mapper/data2_1	hanadata2	vol	/dev/mapper/hanadata2-vol
/dev/mapper/data2_2			
/dev/mapper/log1_1	hanalog1	vol	/dev/mapper/hanalog1-vol
/dev/mapper/log1_2			
/dev/mapper/log1_3			
/dev/mapper/log2_1	hanalog2	vol	/dev/mapper/hanalog2-vol
/dev/mapper/log2_2			
/dev/mapper/log2_3			

Neither the volume group nor the logical volume may contain a dash. For better readability and supportability, it is recommended to make the storage partition number a part of the volume group or logical volume name.

To use the fcClientLVM, the following lines need to be added to `global.ini`:

```
[storage]
ha_provider = hdb_ha.fcClientLVM
partition_1_data__lvmname = <lvm_device1>
partition_1_log__lvmname = <lvm_device2>
partition_2_data__lvmname = <lvm_device3>
partition_2_log__lvmname = <lvm_device4>
```

Continuing with the device names above, valid entries for partition 1 are

```
[storage]
…
partition_1_data__lvmname = hanadata1-vol
partition_1_log__lvmname = hanalog1-vol
…
```

Choice of the Filesystem

SAP does neither force to use any specific filesystem nor has requirements for its configuration. Practical experience showed that XFS is used in general.

lvm.conf

The settings in the `/etc/lvm/lvm.conf` are mostly independent from the fcClient/fcClientLVM. But it is required that the optionally available LVM Metadata Daemon (lvmetad) is disabled in distributed SAP HANA systems (scale-out systems with Host Auto-Failover).

Some Linux releases may enable this daemon by default but it lacks the support of clustered (Type 3) locking.

```
use_lvmetad = 0
      locking_type = 3
```

- `use_lvmetad`: Activation state of the LVM Metadata Daemon. If available in the utilized Linux OS it has to be disabled as it doesn't support the Type 3 locking type.
- `locking_type`: Type of locking to use. Type 3 uses built-in clustered locking which is mandatory for HANA in combination with LVM

For more information about `lvm.conf` settings, see the man page in your operating system

Using mpathpersist

Starting with HANA 2 SPS 05 Rev 53 we provide storage connector implementations based on mpathpersist. Those are called fcClientMpath.py and fcClientLVMMpath.py and are similar to the existing implementations fcClient.py and fcClientLVM.py. Instead of using sg_persist to handle registration and reservation of persistent reservations they use the command mpathparsist. Both variants can be used on systems where the command mpathpersist is available.

To use a mpathpersist-based implementation you have to set the parameter [storage] ha_provider of the global.ini to either hdb_ha.fcClientMpath or hdb_ha.fcClientLVMMpath (see also section Preparing global.ini). E.g.:

```
[storage]
ha_provider = hdb_ha.fcClientMpath
```

The other parameters of section storage are configured as previously described.

It is possible to switch from fcClient to fcClientMpath (or fcClientLVM to fcCLientLVMMpath) or vice versa. However, this requires a restart of HANA and the file /etc/sudoers needs to be adjusted accordingly (see section sudoers).

One advantage of this implementation over fcClient.py is that it can handle faulty device paths during a HANA startup. I.e., not all device paths must be running during startup.

Further it allows automatic registration of reservation keys in case new device paths are added to a LUN or become active again. However, for this to work you have to add the reservation key used by SAP HANA for each host to /etc/multipath.conf .

You can run 'python $DIR_INSTANCE/exe/python_support/hdb_ha/hdbmount.py --reservationKey' as sidadm on each host to get the corresponding reservation keys.

If no other services on the hosts are using SAN storage, then you can add on each host the corresponding reservation key to the default section of the multipath.conf, e.g.:

```
defaults {
    reservation_key   0x123abc
}
```

If other services also access the SAN storage on the hosts then you can add the reservation key to the LUN/multipath section that is used by SAP HANA. E.g.:

```
multipaths {
    multipath {
        wwid    XXXXXXXXXXXXXXXX
        alias       XXXX
        reservation_key   0x123abc
    }
}
```

Each reservation key in the multipath.conf needs the prefix 0x.

After adjusting the file multipath.conf you also have to restart the multipath daemon using:

```
systemctl restart multipathd
```

Configuration of SAP HANA options

This chapter covers tailored configuration properties which are not part of the standard HANA installation on SAN storages.

Dynamic Tiering

HANA 1 SPS 10 introduced the possibility to use LUNs as storage containers for Dynamic Tiering (DT) as depicted in Figure 1. The configuration for that scenario is found in the next section.

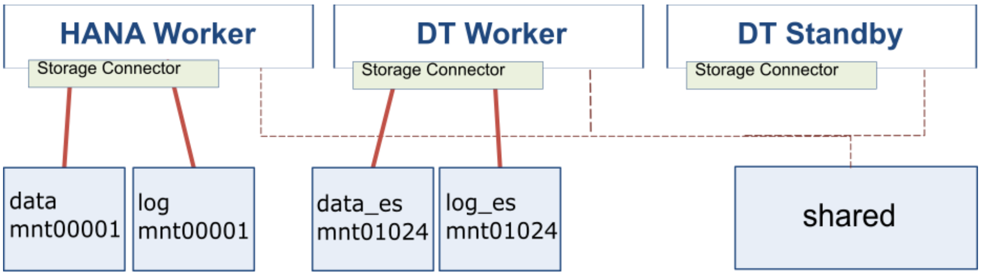


Figure 1: HANA and DT are stored on dedicated LUNs

The DT persistence, which have the usage types `data_es` respectively `log_es`, may be stored on a shared filesystem. Figure 2 shows an example of this configuration which was the only option prior to HANA 1 SPS 10.

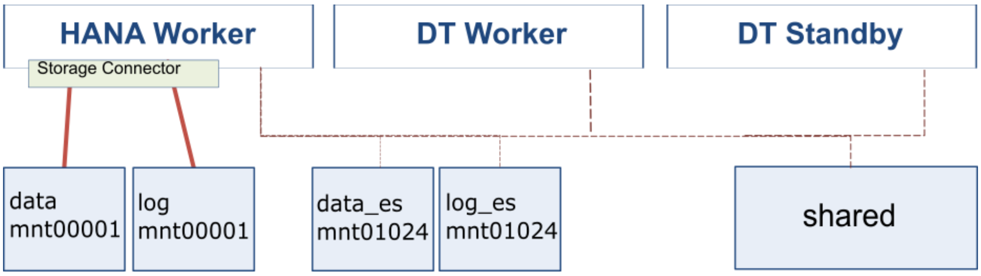


Figure 2: HANA uses SAN while DT resides on the shared filesystem

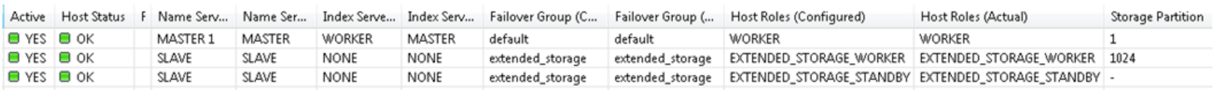
HANA 1 SPS 10 changed the default names of the failover groups to enable automatic failover of the DT hosts. The failover groups are not changed when an upgrade to HANA 1 SPS 10 or higher is performed.

The following table shows the group name depending on the DT host role:

Host role	SPS09 Failover Group	SPS10 Failover Group
EXTENDED_STORAGE_WORKER	EXTENDED_STORAGE_WORKER	EXTENDED_STORAGE
EXTENDED_STORAGE_STANDBY	EXTENDED_STORAGE_STANDBY	

Overwriting the default failover groups to unequal names disables the automatic failover for DT hosts. The installation of DT is described in the *SAP HANA Dynamic Tiering: Installation Guide.*

The next picture shows an example of a system composed of three hosts. One host is a normal HANA worker, the other two hosts only have DT-related host roles:



Active	Host Status	F	Name Serv...	Name Ser...	Index Serve...	Index Serv...	Failover Group (C...	Failover Group (...	Host Roles (Configured)	Host Roles (Actual)	Storage Partition
🟩 YES	🟩 OK		MASTER 1	MASTER	WORKER	MASTER	default	default	WORKER	WORKER	1
🟩 YES	🟩 OK		SLAVE	SLAVE	NONE	NONE	extended_storage	extended_storage	EXTENDED_STORAGE_WORKER	EXTENDED_STORAGE_WORKER	1024
🟩 YES	🟩 OK		SLAVE	SLAVE	NONE	NONE	extended_storage	extended_storage	EXTENDED_STORAGE_STANDBY	EXTENDED_STORAGE_STANDBY	-

Storing HANA and DT on SAN

Storage configuration for DT in global.ini must comply these requirements:

- The partition number for DT is defined to be in the range between 1024 and 2023.
- The usage type is either "`data_es`" or "`log_es`"

For fcClient, the `global.ini` has to be extended with the following settings:

```
[storage]
ha_provider = hdb_ha.fcClient
partition_1024_data_es__wwid = <wwid1>
partition_1024_log_es__wwid = <wwid2>
```

The user-friendly mode to specify aliases instead of `wwids` is supported for DT as well.

If fcClientLVM is used the `global.ini` requires the LVM names:

```
[storage]
ha_provider = hdb_ha.fcClientLVM
partition_1024_data_es__lvmname = < lvm_device1>
partition_1024_log_es__lvmname = < lvm_device2>
```

Storing HANA on SAN and DT on shared filesystem

To enable this option which corresponds to the HANA 1 SPS 09 configuration, the subsequent parameter has to be set in `global.ini`

```
[storage]
…

enable_extended_storage = false
```

In case that the parameter is not specified the default of "`true`" is assumed. Any partition definition in the `global.ini` which is meant for DT are ignored by the Storage Connector API if the parameter is set to "`false`".

If a manual failover is initiated, the new DT worker requires access to the shared network path.

SAP HANA accelerator for SAP ASE

HANA 1 SPS 11 introduced the possibility to use LUNs as storage containers for SAP HANA accelerator for SAP ASE (AFA/ETS). The configuration for that scenario is found in the next section.

Storing HANA and AFA on SAN

Storage configuration for AFA in global.ini must comply these requirements:

- The partition number for AFA is defined to be in the range between 2048 and 3071.
- The usage type is either “data_ets” or “log_ets”

Therefore the *global.ini* has to be extended with the following settings:

```
[storage]
ha_provider = hdb_ha.fcClient
partition_2048_data_ets__wwid = <wwid1>
partition_2048_log_ets__wwid = <wwid2>
```

The user-friendly mode to specify aliases instead of wwids is supported for AFA as well.

On the contrary, when fcClientLVM is used the global.ini requires the LVM names:

```
[storage]
ha_provider = hdb_ha.fcClientLVM
partition_2048_data_es__lvmname = < lvm_device1>
partition_2048_log_es__lvmname = < lvm_device2>
```

Storing HANA on SAN and AFA on shared filesystem

To enable this option the parameter has to be set in global.ini

```
[storage]
…

enable_ets = false
```

In case that the parameter is not specified the default of “true” is assumed. Any partition definition in the global.ini which is meant for AFA are ignored by the Storage Connector API if the parameter is set to “false”.

If a manual failover is initiated, the new AFA worker requires access to the shared network path.

SAP HANA Lifecycle

This section describes briefly how the Fiber Channel Storage Connector integrates into SAP HANA life cycle management.

Installation

To employ the fcClient/fcClientLVM already during the SAP HANA installation, the `multipath.conf` and `global.ini` must be prepared beforehand (note: `/etc/sudoers` is updated automatically). The standard fcClient/fcClientLVM comes with the installation package and can be activated by using the parameter `––storage_cfg=/some/path` with `/some/path` pointing to the directory, which contains the `global.ini`.

Update

The usage of the fcClient/fcClientLVM Storage Connector does not have any influence on the update process: the newest versions of the scripts are copied to the `hdb_ha` directory, which will be used after the restart of the database. The `/etc/sudoers` file will be changed accordingly on all hosts if necessary.

Adding Hosts

When the fcClient/fcClientLVM is already configured on the master host, the procedure for adding a new host will automatically use the fcClient/fcClientLVM. The newly associated storage partition number must already be present in the `global.ini` and the OS settings must be correct. Here, the `/etc/sudoers` file is updated automatically.

Removing Hosts

Devices will be unmounted and reservation will be cleared when the host is removed.

Stopping HANA instances

Devices are unmounted if possible, but open file handles or timeouts may prevent the unmount operation. See SAP Note 2167727 for details about timeouts in hard or soft shutdowns.

Killing HANA instances

Devices are not unmounted.

Renaming

If the SAP HANA database is renamed, care must be taken when the SID of the system is changed. Since the <sidadm> user is renamed as well, the `/etc/sudoers` file must be adapted manually.

Attaching a LUN

When attaching a LUN, the whole fencing mechanism is employed. The host that takes over the LUNs, registers its host-specific key on the devices and reserves or preempts them. Only after this, the actual mounting is done. The whole procedure includes some cleaning up of mounts and multipath before the actual attach will happen.

In detail, the following major steps (there is a bit more to it, but here only the relevant commands are listed) will be executed in fcClient.
Up to HANA 1 SPS 12 this is done sequentially for both, data and log LUNs, in succession. Starting with HANA 2 SPS 00, the LUNs are mounted simultaneously.

1. `sudo /etc/init.d/multipathd force-reload       # for Persistent Reservation Exclusive Access only`
   This will force the multipathing daemon to show all devices with all paths, even if they have failed, for example, by existing reservations
2. `sudo /sbin/multipathd disablequeueing maps     # for Persistent Reservation Exclusive Access only`
   Issued to ensure that IO queuing is disabled
3. `sudo /sbin/multipath -l <wwid|alias>`
   Retrieval and check of `/dev/mapper/<wwid>` device name, single devices (paths) are extracted
4. `ls /sys/block/<wwid>/slaves`
   Extraction of all single devices associated with the LUN
5. `umount <device/path>`
   Unmounts everything that is mounted to the requested path and unmounts every mount of the device
   a. on error do (usually there are left-over processes blocking the path):
      `lsof | grep <path>`
      Get PIDs of blocking process(es)
   b. `kill -9 <pids>`
      Kill blocking process(es)
6. `umount <other paths>`
   Cleanup: all mntXXXXX paths will be unmounted (upon failover those mounts remain in the system but they are neither readable nor writeable)
7. For all single devices:
   `sudo /usr/bin/sg_persist --out --register --param-sark=<key> <single device>`
   Registers a host-specific key on the device (multiple keys can be registered associated). If the key for this host is already registered, it will be used.
8. For all single devices:
   `sudo /usr/bin/sg_persist -i -k <single device>`
   Check that the key is actually registered.

9. For one of the single devices:
   ```
   sudo /usr/bin/sg_persist -r <single device>
   ```
   Check if another host holds a reservation. If applicable, read the key `<oldKey>`.
   a. If not:
      ```
      sudo /usr/bin/sg_persist --out --reserve --param-rk=<key>--prout-type=<prType> <single device>
      ```
      Activation of the registration for the device
   b. If yes:
      ```
      sudo /usr/bin/sg_persist --out –preempt --param-sark=<oldKey> --param-rk=<key>--prout-type=<prType> <single device>
      ```
      Preempt the reservation from the other host. This is atomic.
10.
    ```
    mount <mountoptions> <device> <path>
    ```
    The LUN is finally mounted to the path
11.
    ```
    sudo /sbin/multipath -F                  # for Persistent Reservation Exclusive Access only
    ```
    Cleanup of the multipath table in order to avoid a massive amount of `/var/log/messages` entries for fenced devices (“reservation conflict”).

For the fcClientLVM, the procedure is as follows:

1.
   ```
   sudo /etc/init.d/multipathd force-reload       # for Persistent Reservation Exclusive Access only
   ```
   This will force the multipathing daemon to show all devices with all paths, even if they have failed, for example, by existing reservations
2.
   ```
   sudo /sbin/multipathd disablequeueing maps     # for Persistent Reservation Exclusive Access only
   ```
   Issued to ensure that IO queuing is disabled
3.
   ```
   sudo /sbin/vgscan
   ```
   Checks for LVM metadata updates
4.
   ```
   ls /sys/block/<lvmname>/slaves
   ```
   Extraction of all LUNs associated with the LVM device
5. For all LUNs:
   ```
   ls /sys/block/<wwid>/slaves
   ```
   Extraction of all single devices associated with all the LUNs
6. *Follow all steps from step 5 to step 9 like shown above for the fcClient*
7.
   ```
   sudo /sbin/vgchange -ay <volumegroup>
   ```
   Activates the volume for the use with HANA
8. *Continue with step 5 shown above for the fcClient*

Detaching a LUN

The `detach()` method unmounts the devices. Reservations are not cleared. Detailed procedure for the fcClient:

1. `umount <device/path>`
   Unmounts everything that is mounted to the requested path and unmounts every mount of the device
   a. on error do (usually there are left-over processes blocking the path):
      `lsof | grep <path>`
      Get PIDs of blocking process(es)
   b. `kill -9 <pids>`
      Kill blocking process(es)
2. `sudo /sbin/multipath -F   # for Persistent Reservation Exclusive Access only`
   Cleanup of the multipath table in order to avoid a massive amount of `/var/log/messages` entries for fenced devices ("reservation conflict").

The fcClientLVM uses this procedure:

1. `umount <device/path>`
   Unmounts everything that is mounted to the requested path and unmounts every mount of the device
   a. on error do (usually there are left-over processes blocking the path):
      `lsof | grep <path>`
      Get PIDs of blocking process(es)
   b. `kill -9 <pids>`
      Kill blocking process(es)
2. `vgchange -an <lvmname>`
   Deactivate the devices
3. `sudo /sbin/multipath -F   # for Persistent Reservation Exclusive Access only`
   Cleanup of the multipath table in order to avoid a massive amount of `/var/log/messages` entries for fenced devices ("reservation conflict").

Custom Extensions

The fcClient script is delivered with an empty sub-class fcClientRefined. If some custom code is needed, this script can be copied to a place on the binary share outside the SAP HANA installation. This script offers several hooks, which will be called during attach and detach.

The global.ini would need to be changed with regard to the change:

```
[storage]
ha_provider = fcClientRefined
ha_provider_path = /hana/shared/myFcClient
```

The base classes can still be taken from the HANA installation leading to a simple file system structure. Example:

```
/hana/shared/myFcClient/fcClientRefined.py
```

If new operating system dependencies arise with the custom refinements, the `sudoers()` method should be overloaded accordingly.

Troubleshooting

This section discusses common errors that might be found in the name server trace files and gives advice how to fix the problems. In addition it gives advice on how to cope with different situations requiring manual intervention.

Error Messages

Table: Possible errors that might occur and their solutions.

Error Message:	**Reason/Solution:**
could not reload multipath topology	Check `/etc/sudoers` for correct entries
could not disable path queuing	Check `/etc/sudoers` for correct entries
no storage with key (partition, usageType) = (<partition number>, <usage type >) configured	Missing entry in `global.ini` for `<partition number>` and `<usage type>` combination
unsupported prout-type '<prType>' for persistent reservation	Check `global.ini:[storage]:partition_*_*__prType`, only values "5" and "6" are allowed
error while reading multipath map for wwid '%s'	`sudo /sbin/multipath -l <wwid>` failed for unknown reason – check this command on OS manually
unable to find available device for writing SCSI-3 Persistent Reservation	Registration or clearing of persistent reservation failed. Check single devices with `sg_persist -r /dev/<single device>` - probably a different key is active on the device
unable to find PR key on device '<wwid>'	Key registration failed, check manually
unable to unmount path \`<path>\` (mounted to \`<device>\`), reboot required	Device is blocked by OS, unable to unmount. Check path with `lsof` and end blocking processes. Sometimes only a reboot of the server helps.
Reservation of Persistent Reservation key failed for device '<device>'	Error on registration or reservation. Check device manually: `sg_persist -r /dev/<single device>`
device to mount not ready	Multipath daemon not fast enough, check if system is under high load or something is wrong the mountoptions parameter
Many warnings in /var/log/messages: hostname01 kernel: [873.926030] sd 2:0:0:36: **reservation conflict** hostname01 sudo: abcadm : TTY=unknown ;PWD=/hanamnt/shared/abc/HDB00/hostname01 ; USER=root ;COMMAND=/usr/bin/sg_persist --out --register --param-sark=285436c3e178c43e /dev/sdbd	To prevent data corruption the SCSI-3 Persistent Reservations are not removed during HANA shutdown. This avoids situations in which short running scripts (e.g. hdbnsutil) change the persistence without any notice of the corresponding idling HANA service. The "reservation conflict" warning can be ignored. It does not affect HANA startup.
wrong API version for mounting ES storages with ha_provider	If this error message is encountered, please verify that the official `client.py` is utilized and not an older copy with API version 1. To resolve the issue remove the file from a custom location specified by *`ha_provider_path`* and verify that the `$DIR_INSTANCE/exe/python_support/hdb_ha/` contains the official revision
Mount/unmounting of LUNs block during system start:	Please ensure that the LUNs used for the HANA system are not listed in `/etc/fstab`

client.py(00278) : run OS command `sudo umount /hana/data/<SID>/mnt00001` but no return code is seen afterwards in the form of client.py(00298) : => return code: 0	The LUNs need to be managed by HANA exclusively and not by the operating system. Otherwise startup or failover may fail.
unable to resolve LVM device 'dm-XX' to LUNs	Resolving the LVM device to a multipath LUN failed. This corresponds to step 5 of “Attaching a LUN” for the fcClientLVM variant. Execute `ls /sys/block/dm-XX/slaves` and replace the device number by the one in the error message. This should yield one or more devices which have to be in turn part of the output of “`multipath -l`”. If that is not the case, please check the LVM & LUN configuration.

What if?

Table: Typical use-cases that require manual intervention.

<table>
<tr><th>Use Case:</th><th>How to:</th></tr>
<tr><td>I want to do administrative work on a LUN.</td><td>Stop the whole SAP HANA system in order to ensure all reservations are released and no failover mechanism disrupts your planned work.</td></tr>
<tr><td>SAP HANA was not shut down gracefully and I cannot access some LUNs.</td><td>Reservations are still active. Run as root:

Warning: Only remove reservations manually when you know exactly what you are doing – possible data loss.

1. Retrieve the names of all paths of your LUN: multipath -l <wwid>
2. Read reservation key: sg_persist -r /dev/<single device>
3. For all single devices register this key: sg_persist --out --register --param-sark=<key> <single device>
4. For all single devices clear this key: sg_persist --out --clear --param-rk=<key> <single device&gt;</td></tr>
<tr><td>A server does not come up after reboot.</td><td>A LUN is still reserved by another host with an r/w reservation (prType = 6).

1. Stop the whole SAP HANA system
2. Identify which LUNs cause the server to hang
3. Clear reservation on another server as described in: SAP HANA was not shut down gracefully, I cannot access some LUNs.
4. The hanging server should come up
5. Start SAP HANA

To avoid this problem, you can switch to write-only reservation by adding the following line to global.ini in the storage section:

partition_*_*__prType = 5</td></tr>
<tr><td>I see an “**Input/output Error**” when trying to access the persistences on operating system level.</td><td>A LUN is reserved by another host, but the mount is still visible on the previously failed host.
The device can safely be unmounted, but neither read/write (prType=6) access nor write (prType=5) access will not be possible. **Do not make changes to the persistent reservation of the LUN** unless SAP HANA is stopped completely on all hosts.</td></tr>
<tr><td>After a reboot my LVM devices are all gone.</td><td>Please ensure the LVM is started at boot time. The can be done by running:

chkconfig boot.lvm on
chkconfig boot.multipath on</td></tr>
<tr><td>After testing, reservations are messed up completely. How do I remove these?</td><td>This command reads the persistent reservation keys of all multipath devices and deletes them:

for d in `multipath -ll | grep "sd.." -o `; do export KEY=""; export KEY=`sg_persist -i -k /dev/$d | grep 0x | grep -v "reservation key" | cut -d"x" -f2`; for k in $KEY; do sg_persist --register --out --param-sark=$k /dev/$d; sg_persist --out --clear --param-rk=$k /dev/$d; done; sg_persist -i -k /dev/$d; done

Please make sure that HANA is stopped on all hosts.</td></tr>
</table>

After host failure, the former data and log mounts remain in the operating system	If the HANA is not shut down gracefully, the nameserver is not able to call umount on the LUNs. Those mounts will remain in the system until the host takes over a new active role (not standby). Since the *actual* active host on the LUNs holds the SCSI-3 persistent reservations, the remaining mounts on the failed host do not harm. An “Input/output Error” will be thrown when accessing those devices.
Use Case:	**How to:**

I want to mount a device like HANA does	As <sidadm> run
	`hdbnsutil -attachStorage --partition=<partno> --type=[data|log]`
	`hdbnsutil -detachStorage --partition=<partno> --type=[data|log]`

Mounts remain after a host/system shutdown	Please see the section about “Stopping Hosts” for an explanation

Terminology Appendix

Fencing

“Fences out” an entity of a distributed system that is not acting normally. Usually this entity will be killed or all shared resources will be revoked from it.

Host Auto Failover

> The **Master host** coordinates transactions and governs the system topology. There is only one master at a time.
> A **Standby host** is a passive component of the system. It has all services running, but not data volumes assigned waiting for failure of others to take over their role.
> A **Worker host** is an active component accepting and processing requests.

HBA – Host Bus Adapter

An entity of server hardware that connects the host to the storage subsystem.

LUN

Logical Unit Number – an identifier of a storage device

LVM

Logical Volume Management – provides a method of allocating space on mass-storage devices that is more flexible than conventional partitioning schemes. In particular, a volume manager can concatenate, stripe together or otherwise combine partitions into larger virtual ones that administrators can re-size or move, potentially without interrupting system use [source: Wikipedia].

SCSI-3 Persistent Reservations

A built-in mechanism of the SCSI-3 protocol, which is widely supported by most storage subsystems. Based on registered keys, a device can be reserved, i.e., locked.

Split Brain

A situation in a distributed system where more than one host demands the master role for itself, usually because the connection is broken between them.

[1] SAP HANA High Availability White Paper
https://www.sap.com/documents/2016/05/f8e5eeba-737c-0010-82c7-eda71af511fa.html