



SoftNAS Deployment Guide for SAP HANA

Summary

The goal is to provide the SoftNAS best practices for setting up SAP HANA.

Top 5 reasons to choose Buurst SoftNAS

1. Superior performance for processing data
2. Maximum scale to process as much data as possible
3. High reliability
4. Low cost
5. Integration into existing infrastructure

SoftNAS Best practices for SAP HANA

SAP HANA is an in-memory database optimized for data-driven, real-time decision-making and actions and supports all workloads with the broadest advanced analytics on multi-model data. Shared storage allows customers greater flexibility and the ability to take advantage of existing storage capacity they may have in their enterprise arrays. SoftNAS enables customers to integrate the SAP HANA solution into their current data center operations, including data protection, monitoring, and data management. SoftNAS helps to improve the time to value for a SAP HANA implementation and reduce risk and costs. Scalability of SAP HANA is defined by the number of production HANA worker hosts connected to enterprise storage arrays and still meets the key SAP performance metrics. Because SoftNAS can provide more capacity than required for HANA, scalability depends on several factors, including:

- Array cache size
- Array performance
- Array bandwidth, throughput, and latency
- HANA host connectivity to the array
- Storage configuration for the HANA persistence

SAP HANA uses external disk storage to maintain a copy of the data in memory to prevent data loss due to a power failure and enable hosts to failover and have the standby SAP HANA host take over-processing.

SAP HANA Persistent Data Storage

Persistent storage media is required for ongoing save operations for data and redo log files. To protect data from the risk of memory failure, SAP HANA persists in-memory data to storage media and flushes all changed data from memory to the data volumes. This operation takes place based on savepoints, which occur by default every 5 minutes. Storage media types include hard disk storage and non-volatile RAM for the MAIN fragment of column store tables. For more information about the MAIN and DELTA fragments of tables, refer to Delta Merge.

Data and Log Volumes

- SAP HANA persists in-memory data by using savepoints. Each SAP HANA service has its separate savepoints. During a savepoint operation, the SAP HANA database flushes all changed data from memory to the data volumes. The data belonging to a savepoint represents the disk data's consistent state and remains until the next savepoint operation. Redo log entries are written to the log volumes for all changes to persistent data. In the event of a database restart (for example, after a crash), the last completed savepoint is retrieved from the data volumes. The redo log entries are written to the log volumes since the previous save point.
- The frequency at which savepoints are defined can be configured in the persistence section of the global.ini file (every 5 minutes by default). Savepoints are also triggered automatically by several other operations such as data backup and database shutdown and restart. You can start a savepoint manually by executing the following statement ALTER SYSTEM SAVEPOINT.
- You must always ensure that there is enough space on the disk to save data and logs. Otherwise, a disk-full event will occur, and the database will stop working.
- During the installation process, the following default directories are created as the storage locations for data and log volumes:
 - /usr/sap//SYS/global/hdb/data
 - /usr/sap//SYS/global/hdb/log
- These directories contain a separate sub-directory, or storage partition, for each host in the system. These are named mnt00001, mnt00002, mnt00003, and so on, by default. Each host storage partition contains a sub-directory for every database service that persists data. These sub-directories represent the actual volumes. They are named hdb00001, hdb00002, hdb00003, and so on by default. The volume names of tenant databases have a suffix to represent the database. For example, the index server volume for the first tenant database is hdb00002.00002, for the second database hdb00002.00003, and so on.
 - **nameserver:** Only the nameserver service in the system database on the active master host persists data. Slave nameserver hosts communicate with the master but do not persist data.

- **index server:** The index server service in tenant databases on all hosts except standby hosts persists data. This data volume can be partitioned to improve performance when reading and writing.
- **xsengine:** The xsengine service persists data on any host on which it is running.
- Every SAP HANA node requires access to a file system mounted under the local /hana/shared/ mount point to install the SAP HANA binaries and configuration files, traces, and logs. An SAP HANA scale-out cluster requires a single shared file system, which must be mounted on every node.
- To calculate the /hana/shared/ file system's size, use the formulas provided in the SAP paper SAP HANA Storage Requirements.
- Single node (scale-up):
 - $\text{Sizeinstallation}(\text{single-node}) = \text{MIN}(1 \times \text{RAM}; 1 \text{ TB})$
- Multinode (scale-out):
 - $\text{Sizeinstallation}(\text{scale-out}) = 1 \times \text{RAM_of_worker}$ per 4 worker nodes

Partitioning Data Volumes

- Data volumes on the index server can be partitioned so that read and write operations can run in parallel with increased data throughput.
- Each SAP HANA service has one data volume, which stores its data in one single file by default. Administrators have the option to partition the data volume for the index server so that data files can be segmented and distributed over several stripes. Partitioning is a logical distribution of data pages over different partitions, and striping is a physical segmentation of files; using partitioning and striping, data can be read or written in parallel threads.
- If you create a new partition on the index server, it is added simultaneously to all indexservers in the topology. New partitions become active after the next savepoint on each indexserver; this is shown in the partition STATE value, which changes from ACTIVATING to ACTIVE. By default, all data volumes have a single partition with the numeric ID zero. A numeric partition ID is assigned automatically to new partitions by the HANA persistency layer. If the partition numbering is for any reason inconsistent across all indexservers then any attempt to add new partitions will fail.
- Partition details are saved in the indexserver.ini configuration file:
- `basepath_datavolume[]=`
- The *path* value is the default HANA data volume basepath; this cannot be changed, but more flexibility will be introduced in future releases.
- Details of the SQL statements to manage data volume partitions are given in the SAP HANA SQL and System Views Reference, the two main statements to add and drop partitions are shown here. If you execute the command to add a partition on the indexserver, it will add a new partition to all indexservers in the topology:
 - ALTER SYSTEM ALTER DATAVOLUME ADD PARTITION
- You can drop specific data volume partitions by specifying the partition ID number:
 - ALTER SYSTEM ALTER DATAVOLUME DROP PARTITION

- This command drops the identified partition from all indexservers in the topology. The default partition with ID zero cannot be dropped. If you drop a partition, then all data stored in the partition is automatically moved to the remaining partitions, and for this reason, dropping a partition may take time. This operation also removes the partition entry from the configuration file.
- You can see the current data volume configuration from the following two views:
 - M_DATA_VOLUME_STATISTICS: This provides statistics for the data volume partitions on the indexserver, including the number of partitions and size.
 - M_DATA_VOLUME_PARTITION_STATISTICS: This view provides statistics for the individual partitions, identified by PARTITION_ID, and includes the partition STATE value.
- The size of the log directory depends on the RAM size of the node. The SAP HANA Storage Requirements paper provides the following formulas to calculate the minimum size of the log directory:
 - [systems ≤ 512GB] Sizeredolog = 1/2 x RAM
 - [systems > 512GB] Sizeredolog(min) = 512 GB

Persistent Memory

- Persistent memory (non-volatile RAM, also referred to as Storage Class Memory) is supported in SAP HANA as a persistent storage type.
- Persistent memory (or NVRAM) is an emerging class of memory which combines the qualities of both DRAM and Flash storage and bridges the gap between disk storage and main memory. For SAP HANA, its most important characteristics are that it is byte-addressable like DRAM and can be treated by the CPU as RAM, therefore, offering fast read and write performance; its latency characteristics are also very close to DRAM.
- The persistent storage functions as DRAM memory and is used specifically for MAIN data fragments of the column store where approximately 90% of the data of a table is held. In this way, DELTA data can continue to be stored in DRAM (a relatively small quantity of data requiring the fastest possible access) with the MAIN segment of a table in NVRAM. Note that if persistent memory storage is enabled, data is still written to the data volumes but is simply ignored. After a system restart, the main data fragments saved in persistent storage are still available in memory and do not have to be reloaded. This storage type has no impact on backup and recovery or high availability processes.
- As with the data volume persistent storage feature, data is persisted based on savepoints. In non-volatile memory storage, NVM blocks are constructed whenever the in-memory MAIN fragment is created and is kept consistent according to HANA's savepoints.
- The key performance benefit that this will bring is in greatly accelerated start-up times so that data can be quickly reloaded into memory, and any system downtime can be minimized. Other benefits may include higher memory capacities per server.

- Performance differences of this storage type compared to DRAM due to higher latency are mitigated by features such as hardware prefetchers. It is also possible to enable persistent memory storage for selected tables or for specific columns of a table so that a selective implementation of this storage type is possible (see Configuration below).
- The SAP HANA in-memory database requires disk storage for the following purposes:
 - Data:
 - To maintain the persistence of the in-memory data on disk to prevent data loss resulting from a power outage
 - To enable a host auto-failover, where a standby SAP HANA host takes over the in-memory data of a failed worker host in scale-out installations
 - Log: To log information about data changes (redo log)
- Every SAP HANA scale-up node and scale-out (worker) node requires two disk directories on the Isilon storage system—to save the in-memory database on disk (data) and keep a redo log (log). The size of these directories depends on the anticipated total memory requirement of the database and the RAM size of the node. To prepare the disk sizing, see the SAP paper SAP HANA Storage Requirements. Version 2.10 of the paper states that you can calculate the size of the data directory by using the following formula.
 - $\text{Sizedata} = 1.2 \times \text{net disk space for data}$
The net disk space is the anticipated total memory requirement of the database plus 20 percent free space.
- If the database is distributed across multiple nodes in a scale-out cluster, divide the net disk space by the number of SAP HANA worker nodes in the cluster.
 - For example, if the net disk space is 2 TB and the scale-out cluster consists of four worker nodes, then every node must have a 616 GB data volume assigned to it
 $2 \text{ TB} / 4 = 512 \text{ GB} \times 1.2 = 616 \text{ GB}$
- If the net disk space is unknown at the time of storage sizing, use the RAM size of the node plus 20 percent free space to calculate the data file system's capacity.

Network

- The Network File System (NFS) protocol is used when SAP HANA clients mount remote directories on the SoftNAS.
- NFS file systems are mounted via OS mount: in the event of a node failover, the standby host has immediate access to the same shared storage where the data and log volumes are located. This is called a shared everything architecture because all the data and log partitions are mounted on all nodes.

Backup

- If you plan to back up SAP HANA to a directory on SoftNAS, see the information about sizing the backup file system in the SAP HANA Storage Requirements paper. The

capacity depends not only on the data size and the frequency of change operations in the database but also on the number of backup generations kept on disk.

Backint API

- The Backint method is an API-based method — it's generally referred to as an SAP HANA backup using a third-party utility. It enables 3rd party tool vendors to connect their backup agents to the SAP HANA database directly. Backups are transferred via pipe from the SAP HANA database to the 3rd party backup agent, which runs on the SAP HANA database server and sends the backups to the 3rd party backup server.
- 3rd party backup tools can be fully integrated with SAP HANA:
 - Execution of backup and recovery from the SAP HANA cockpit, SAP HANA studio, and via SAP HANA SQL commands
 - The Backint backup can be configured with these options:
 - SAP HANA database stores the in-memory data into persistence storage
 - From the persistence storage, the third-party agent creates backup streams to the third-party backup target.
 - Store the backup on SoftNAS backup Storage

File System Backup

- File System Backups protect complete data to a file. Also, note that a File System Backup will put a load on the network, and it is recommended to use shared backup storage so it can be available to all the nodes of the database.
- You must remember to monitor the file system fill level during the backup.