Host Auto-Failover is a built-in, fully automated high availability solution for recovering from the failure of a SAP HANA host. This paper explains how this mechanism works in detail and describes the important interfaces an administrator has to pay attention to.
# Table of Contents

Legal Disclaimer.......................................................................................................................... 3
Change History ............................................................................................................................ 3
What’s new.................................................................................................................................. 4
Introduction.................................................................................................................................. 5
Heartbeat ...................................................................................................................................... 6
Fencing ....................................................................................................................................... 6
Configuration and Monitoring ..................................................................................................... 7
Host Failure Detection .................................................................................................................. 9
Failover Target Host Selection and Failover Execution ............................................................... 11
  Slave Host Failover to a Standby Host ....................................................................................... 11
  Master Host Failover to a Standby Host .................................................................................... 12
  Master Host Failover without Available Standby Hosts ............................................................ 12
  Master Host Failover with Standby Host but all Master Candidates being in Use (So Called Double
  Failover) .................................................................................................................................. 13
Target host selection ................................................................................................................... 14
FAQ ............................................................................................................................................. 14
  Quorum ................................................................................................................................... 14
  Split Brain ............................................................................................................................... 15
  hdbnsutil ................................................................................................................................. 15
Failover Duration ....................................................................................................................... 16
Host Start Order / Landscape Restart ......................................................................................... 16
Failback ..................................................................................................................................... 16
Master Nameserver Candidates ................................................................................................. 16
Failover Groups ......................................................................................................................... 16
Master Failover without Standby Hosts ...................................................................................... 17
Host Auto-Failover vs. External Cluster Manager ....................................................................... 17
Application Configuration .......................................................................................................... 17
Application Error Handling ....................................................................................................... 17
User Exits ................................................................................................................................... 17
References ................................................................................................................................. 17
Legal Disclaimer

THIS DOCUMENT IS PROVIDED FOR INFORMATION PURPOSES ONLY AND DOES NOT MODIFY THE TERMS OF ANY AGREEMENT. THE CONTENT OF THIS DOCUMENT IS SUBJECT TO CHANGE AND NO THIRD PARTY MAY LAY LEGAL CLAIM TO THE CONTENT OF THIS DOCUMENT. IT IS CLASSIFIED AS "CUSTOMER" AND MAY ONLY BE SHARED WITH A THIRD PARTY IN VIEW OF AN ALREADY EXISTING OR FUTURE BUSINESS CONNECTION WITH SAP. IF THERE IS NO SUCH BUSINESS CONNECTION IN PLACE OR INTENDED AND YOU HAVE RECEIVED THIS DOCUMENT, WE STRONGLY REQUEST THAT YOU KEEP THE CONTENTS CONFIDENTIAL AND DELETE AND DESTROY ANY ELECTRONIC OR PAPER COPIES OF THIS DOCUMENT. THIS DOCUMENT SHALL NOT BE FORWARDED TO ANY OTHER PARTY THAN THE ORIGINALLY PROJECTED ADDRESSEE.

This document outlines our general product direction and should not be relied on in making a purchase decision. This document is not subject to your license agreement or any other agreement with SAP. SAP has no obligation to pursue any course of business outlined in this presentation or to develop or release any functionality mentioned in this document. This document and SAP’s strategy and possible future developments are subject to change and may be changed by SAP at any time for any reason without notice. This document is provided without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP assumes no responsibility for errors or omissions in this document and shall have no liability for damages of any kind that may result from the use of these materials, except if such damages were caused by SAP intentionally or grossly negligent.

© Copyright 2015 SAP SE. All rights reserved.

Change History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>February 2015</td>
<td>Initial Release</td>
</tr>
<tr>
<td>1.1</td>
<td>December 2015</td>
<td>• New introduction section “What’s new”</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Section for target host selection</td>
</tr>
</tbody>
</table>
What’s new

SPS 09

- Additional storage based heartbeat for master nameserver on HANA shared storage (SYS/global/hdb/nameserver.lck)
- new columns/values in SYS.M_LANDSCAPE_HOST_CONFIGURATION for HANA options DT, streaming and ETS/AFA
- manual failover for DT

SPS 10

- Faster master host failure detection (nameserver + daemon + master lock file checks are now done in parallel)
- SAN support & automatic failover for DT (see SAP HANA Fiber Channel Storage Connector Admin Guide)

SPS 11

- Target host selection by host roles (see section Target host selection)
- SAN support for ETS/AFA (see SAP HANA Fiber Channel Storage Connector Admin Guide)
Introduction

The SAP HANA High Availability White Paper [1] describes the different SAP HANA High Availability (HA) solutions offered to support rapid recovery from faults and disasters. Each of these solutions represents a tradeoff, and yields different requirements in terms of storage and network architecture.

This paper discusses the details of the Host Auto-Failover solution.

Host Auto-Failover is a "N+m" host fault recovery solution. One or more hosts are added to a single host or distributed SAP HANA system, and configured to work in standby mode. As long as they are in standby mode the services on these hosts do not contain any data and do not accept requests or queries.

When an active worker host fails, a standby host automatically takes its place. Since the standby host may take over operation from any of the worker hosts, it needs access to all the database volumes. This can be accomplished by a shared networked storage server, by using a distributed file system, or with vendor-specific solutions that use an SAP HANA programmatic interface (the Storage Connector API) to dynamically detach and attach (mount) networked storage (e.g. using block storage via Fiber Channel) upon failover.

Figure 1 shows how Host Auto-Failover logically works:

Figure 1: An active host fails (e.g. Host 2), and the standby host takes over its role by using the persisted data and log files of the failed host (Host 2).

As implied by the name the “Host Auto-Failover” capability of SAP HANA is characterized as follows:

- Failover is performed on the host level. All services of a host are moved to another host. The failure of a single process (service) does not trigger a failover.
- The failover happens automatically as an integral feature of SAP HANA. No external cluster manager is required.
- Data consistency is a key requirement. Data might be corrupted if a failed host (e.g. the original Host 2 in the figure above) would be allowed to restart and write data to disk in parallel to the failover host (New Host 2 in figure above).
To ensure data consistency, at all times it must be guaranteed that a failover does not happen (or at least does not succeed and may not cause corrupt data) if the failed host can potentially still write data. To achieve this, the SAP HANA Host Auto-Failover solution uses a combination of heartbeat and fencing.

**Heartbeat**
The following types of heartbeat are used to check if another host is active as master before starting the current host as master or performing a failover.

- **TCP communication based heartbeats:**
  - Ping from nameserver to nameserver with SAP HANA internal communication protocol
  - Ping from nameserver to hdbdaemon with SAP HANA internal communication protocol

- **Storage based heartbeats:**
The current master nameserver periodically updates heartbeat files located on different storage partitions:
  - Shared storage for the SAP HANA binaries (new in SPS 09)
  - Storage partition 1 for the master node’s data

These types of storage are typically connected with other networks different from the internode network used for service to service communication (e.g. fiber channel for SAN or dedicated Ethernet for NFS) and therefore these heartbeats provide additional value.

**Fencing**
In rare cases the heartbeats cannot detect if another host is alive, for example in split-brain situations where no communication is possible between hosts. I/O fencing ensures that the other side does not access the data or log storage any more.

The SAP HANA Storage Connector API together with a specific Storage Connector allows usage of different types of storage and network architecture to ensure proper I/O fencing:

- **SAN storage**: the SAP HANA Fiber Channel Storage Connector [2] using SCSI-3 persistent reservations (SCSI-3 PGR).
- **NFSv3**: used without file locking, but with a Storage Connector provided by certified storage vendors. This type of Storage Connector implements a STONITH\(^1\) call to reboot a failed host.

\(^1\) Short for “Shoot The Other Node In The Head”. If an NFSv3 client dies (i.e. the SAP HANA server), the file locks are not released on the NFS server side resulting in a deadlock for any host that wants to access these files. Using the nolock mount option will solve the locking problem, but with this option, data is not protected against parallel reading and writing from different hosts. To solve this, STONITH must be implemented.

---

“A heartbeat is a periodic signal generated by hardware or software to indicate normal operation or to synchronize other parts of a system. Usually a heartbeat is sent between machines at a regular interval of the order of seconds. If a heartbeat isn’t received for a time—usually a few heartbeat intervals—the machine that should have sent the heartbeat is assumed to have failed.” (Wikipedia, October 2014)

“Fencing is the process of isolating a node of a computer cluster or protecting shared resources when a node appears to be malfunctioning.” (Wikipedia, October 2014)
• **NFSv4 or cluster file systems like GPFS:** using file locks. A *Storage Connector* is not required here as these file locks reliably prevent false access. However, a STONITH type *Storage Connector* is provided by some storage vendors to speed up failover.

More information about SAP HANA and the underlying storage can be found in the *SAP HANA Storage Requirements White Paper* [3].

**Configuration and Monitoring**

The Host Auto-Failover related configuration and status details are shown in the system view `SYS.M_LANDSCAPE_HOST_CONFIGURATION` and *SAP HANA Studio* → *Administration Perspective* → *Landscape* → *Hosts*. With the □□ Button some of the .._CONFIG.. columns can be modified.

![Screenshot of the system view SYS.M_LANDSCAPE_HOST_CONFIGURATION](image)

Figure 2: Screenshot of the system view **SYS.M_LANDSCAPE_HOST_CONFIGURATION**

From left to right, there are the following columns with their names used in the system view / *SAP HANA studio*:

- **HOST / Host:**
  The hostname

- **HOST_ACTIVE / Active:**
  The state of the SAP HANA instance on the host:
  - **YES:** instance is up and running, all services are operational
  - **NO:** instance is shut down, all services are stopped
  - **STARTING:** instance is starting, not all services are started yet, will switch to **YES** soon
  - **STOPPING:** instance is stopping, will switch to **NO** soon
  - **UNKNOWN:** communication to host failed for any reason, state is undetermined

- **HOST_STATUS / Host Status:**
  The failover-relevant state of the host:
  - **OK:** host is at normal operation
  - **IGNORE:** landscape is OK, host is optional standby
  - **INFO:** landscape operational, host has a different role than configured
  - **WARNING:** landscape not operational, but should become operational when host is started/failed over
  - **ERROR:** landscape not operational, host is not available

- **FAILOVER_STATUS / Failover Status:**
  Information about the current failover activities:
  - **WAITING:** failure detected, wait time to allow failback or restart of host
  - **FAILOVER TO <host>:** active failover to <host>
- FAILBACK TO <host>: active failover back to worker host

- NAMESERVER_CONFIG_ROLE / Name Server Role (Configured):
The configured role of the nameserver:
  - MASTER1, MASTER2, MASTER3: defines if a nameserver is a master candidate and its priority; only these (up to) three candidates may acquire the master role
  - SLAVE: all other nameservers

- NAMESERVER_ACTUAL_ROLE / Name Server Role (Actual):
The current role of the nameserver:
  - MASTER: the actual master
  - SLAVE: all other nameservers

- INDEXSERVER_CONFIG_ROLE / Index Server Role (Configured):
The defined role of the indexservers:
  - WORKER: active indexserver that accepts and processes requests
  - STANDBY: inactive indexserver that will take over a role when a failover occurs
  - NONE: host not used as worker/standby for indexservers (new in SPS 09)

- INDEXSERVER_ACTUAL_ROLE / Index Server Role (Actual):
The current role of the indexservers:
  - MASTER: the transaction master
  - SLAVE: all active indexservers processing data
  - STANDBY: the passive indexservers
  - NONE: host not used as worker/standby for indexservers (new in SPS 09)

- HOST_CONFIG_ROLES / Host Roles (Configured):
The defined roles of the host (new in SPS 09):
  - WORKER: worker indexservers
  - STANDBY: standby indexservers
  - EXTENDED_STORAGE_WORKER: dynamic tiering worker
  - EXTENDED_STORAGE_STANDBY: dynamic tiering standby
  - ETS_WORKER: accelerator for SAP ASE worker
  - ETS_STANDBY: accelerator for SAP ASE standby
  - STREAMING: smart data streaming

- HOST_ACTUAL_ROLES / Host Roles (Actual):
The current roles of the host (new in SPS 09)
  - Same as configured roles

- FAILOVER_CONFIG_GROUP / Failover Group (Configured):
A group that the host is configured in; groups are user-defined strings, which are given at installation time or can be configured later in the SAP HANA studio.
By default the Failover process prefers a target host in the current group, but can also use hosts in other groups. See FAQ -> Failover Group for details.

- FAILOVER_ACTUAL_GROUP / Failover Group (Actual):
The group the host currently runs in: during failover the original group of the failed host is taken along.

- STORAGE_PARTITION / Storage Partition:
A unique number for each worker host.
- **REMOVE_STATUS / Removal Status:**
  
  In case a host is removed from the system, information about the re-organization status can be found here.

The output of the view is also available as a Python script at OS command line. The script `landscapeHostConfiguration.py` is located in `$DIR_INSTANCE/exe/python_support`. For detailed information and consumable output of this script, the `--sapcontrol=1` option can be used as shown below. The return code may be consumed by cluster managers (e.g. for SAP HANA system replication) to come to a decision about the system health state, as follows:

- 0 = Fatal, e.g. database offline
- 1 = Error, e.g. a failover did not happen, because there was no standby host available
- 2 = Warning, e.g. currently failovering
- 4 = Ok
- 5 = Ignore, e.g. system has switched roles (failover), but is fully functional

A return code $>4$ indicates normal system operation.

```
> python landscapeHostConfiguration.py
SAPCONTROL-OK: <begin>
```

```
# python landscapeHostConfiguration.py
```

```
python landscapeHostConfiguration.py
SAPCONTROL-OK: <begin>
```

```
SAPCONTROL-OK: <end>
```

```
> echo $?
4
```

When the system is stopped, this script can also be used, but will fill only a subset of the columns.

More information can be found the SAP HANA Admin Guide [4] (in the section Setting up Host Auto-Failover).

**Host Failure Detection**

A host failure is any dysfunctional state of a host that affects the communication between the hosts of a distributed SAP HANA system. To check the functional state of a host, the nameservers regularly
send a ping on the internal network communication layer to nameservers on other hosts. An additional ping to the hdbdaemon process is executed in the case that the remote nameserver repeatedly did not reply. Only if both services do not reply in time, is the host considered to have failed.

A crash of a single service does not trigger failover, because services are normally restarted by the hdbdaemon. If a service is not able to restart for any reason, it is assumed that it will not be able to start on another host either. Exception: the nameserver aborts itself during startup if the storage connector returns an error. It then instructs the hdbdaemon to shut down the whole database instance on the host including the hdbdaemon itself, which will allow failure detection and failover processing by other hosts.

In detail:

- **Checking slave hosts:**
  - *nameserver communication heartbeat:* The current master nameserver pings all other nameservers every 10 seconds. If a nameserver was active and 5 pings have failed (either immediately or after a 60 second ping timeout) the nameserver is considered *inactive*.
    - by pinging multiple times SAP HANA can recover from short network outages without triggering a failover.
    - see FAQ → *Failover Duration* for typical failure detection times.
  - *hdbdaemon communication heartbeat:* If a slave nameserver was considered *inactive* (or had set itself to *inactive*) the master nameserver pings the slave hdbdaemon process. If the hdbdaemon ping fails (either immediately or after a 60 second ping timeout) the host is considered as *inactive* and a failover is initiated.

- **Checking the master host:**
  - *nameserver communication heartbeat:* nameserver candidates, which are not currently the master ping other candidates with lower priority every 10 seconds. Together with the slave nameserver heartbeat above (current master nameserver pings all other nameservers) normally MASTER1 will ping MASTER2 and MASTER3 and MASTER2 will ping MASTER3. If a master candidate does not receive any ping within 30 seconds, it will ping the master nameserver itself.
  - *hdbdaemon communication heartbeat:* If the ping to the master nameserver fails, the hdbdaemon process on the master host is pinged. If the hdbdaemon does not answer within 1 minute, the current master host is considered as ‘inactive’.
  - *Nameserver storage Heartbeat:* Then the nameserver candidate host checks the heartbeat files for changes for a period of one minute. Those files are updated by the current master nameserver every 10 seconds with the hostname and a random string. A failover begins only if all files do not show any sign of changes for one minute.
Failover Target Host Selection and Failover Execution

When a failure is detected and a replacement host is determined, the actual failover process starts.

Slave Host Failover to a Standby Host

Figure 3: Visualization of a slave host failover to a standby host. On the left the original state of the system is shown. On the right, the second host fails and its role is moved to the fourth host.

In detail:

- **Target host selection:**
  - See section Target host selection for details
- The master nameserver calls the stonith() method of all installed HA/DR provider hooks (in [4] refer to the section “Implementing a HA/DR Provider”) and the Storage Connector stonith() method. Typically the stonith() method is only implemented in NFSv3 related storage connectors and reboots the failed host.
  - If STONITH fails: failover is aborted, all hosts remain in their old roles
- Swap actual services, host roles, storage partition number, volume IDs of all services between both hosts in the topology and inform all other hosts
- The master nameserver (which selected a replacement host), calls the nameserver on the target host to perform the failover
- The host that was promoted to a new role, will call the Storage Connector’s attach() method to acquire the correct storage partition (if applicable) and call the failover() method of all installed HA/DR provider hooks
  - If this fails, the host will stop; if there are still standby hosts available, another failover will be triggered; this host will be set to ERROR
- Reconfigure running standby services to load their newly assigned volume
  - If this fails, this is like a service failure and will not initiate a further failover
- Reconfigure hdbdaemon to start/stop services that should run on only one of the two hosts
  - If this fails, this is like a service failure and will not initiated a further failover
**Note:** The master nameserver is the only entity in the whole system that is able to make a failover target host selection. Since the master has mechanisms to avoid split brain situations, there is conceptually no split brain situation for slave hosts possible (more details in the Split Brain section).

If a slave loses its connection to the master nameserver it waits and will be notified by the new master. If a slave cannot connect to a master during startup, it will terminate itself.

### Master Host Failover to a Standby Host

In detail:

- The nameserver master candidate with the highest priority (= smallest number in configured nameserver role) detects the failure condition and initiates the failover.
- If a nameserver candidate is available which is currently a standby host, the failover is forwarded to this host. This avoids a so called **Double Failover** (see second example below).
- The failover includes the same steps as in the **Slave Host Failover** scenario above.
- The nameserver reloads its persistence from disk.

**Figure 4:** Visualization of a master host failover to a standby host. On the left the original state of the system is shown. On the right, the first host fails and its role is moved to the fourth host.

### Master Host Failover without Available Standby Hosts

**Figure 5:** Visualization of a master host failover to a slave host. On the left original state of the system is shown. On the right, the first host fails and its role is moved to the second host. The original role of the second host will not be available until a standby host is added to the system or the failed first host is re-activated.
In detail:

- The nameserver master candidate with the highest priority detected the failure conditions and executes the failover steps itself.
- The new master nameserver calls the stonith() method of all installed HA/DR provider hooks and the Storage Connector stonith() method (if applicable) to reboot the failed host.
  - If STONITH fails, the failover is aborted and the new master will shut down itself.
  - The (possibly) remaining third master candidate will then retry the failover.
  - If this also fails, no master is available throughout the whole landscape – the slave hosts will eventually shut down themselves.
- The new master stops all its services (except hdbdaemon and nameserver).
- The new master will call the Storage Connector’s detach() method for the old storage partition and the attach() method for the storage partition 1 (mnt00001 directory) and call the failover() method of all installed failover hooks:
  - If this fails, failover is aborted, the new master will shut down itself.
  - The (possibly) remaining third master candidate will then retry the failover.
  - If this also fails, no master is available throughout the whole landscape – the slave hosts will shut down themselves.
- The new master nameserver loads its persistence from disk.
- Swap currently existing services, host roles, storage partition number, volume IDs of all services between both hosts in the topology and inform all nameservers.
- The hdbdaemon process is reconfigured, which will start all required services.
- The role of the displaced slave host remains inactive, the system is only partially available.

**Master Host Failover with Standby Host but all Master Candidates being in Use (So Called Double Failover)**

![Diagram showing master host failover to a slave host](image)

Figure 6: Visualization of a master host failover to a slave host. On the left the original state of the system is shown. On the right, the first host fails and its master role is moved to the second host. The original slave role of the second host will be failed over to the standby host.

In detail:
- If no master nameserver candidate with standby as actual indexserver role is available, one of the master candidates currently used as indexserver slave is chosen as new master.
- The failover steps for the master host are similar to the scenario *Master Host Failover without Available Standby Hosts* above.
- The previously assigned slave is marked as failed and enters the failover queue. As a standby host is available, slave failover will start shortly after master failover.
- Both failovers are executed in parallel.

**Target host selection**

This section describes the selection process of the replacement host. Beginning with SPS 11 the actual host roles (*HOST_ACTUAL_ROLES*) are taken into account.

<table>
<thead>
<tr>
<th>HANA Release</th>
<th></th>
</tr>
</thead>
</table>
| SPS 11            | - If there is a standby host with an exact match of corresponding actual host roles, it will be used  
|                   | - If there is a standby host with one of the roles that corresponds with the failing host, it will be used.  
|                   | - If the failing host has a HANA worker role, any unassigned standby is used |
| SPS 10 and before | - If there is a standby host, it will be used  
|                   | - If there are multiple equivalent options available, the first host will be chosen |

The search steps are restricted to the same failover group, unless `global.ini/[failover]/cross_failover_groups=false` was configured.

If no host is available, no failover will happen and *HOST_STATUS* will show *ERROR*.

**FAQ**

This section contains remarks and notes about the SAP HANA Host Auto-Failover solution.

**Quorum**

In contrast to other HA solutions SAP HANA does not use a quorum consisting of multiple SAP HANA hosts to decide which host can become master at initial startup or master failover.

With *heartbeats* and *fencing* a single host can reliably decide initial startup or master failover.

"A quorum is the minimum number of votes that a distributed transaction has to obtain in order to be allowed to perform an operation in a distributed system. A quorum-based technique is implemented to enforce consistent operation in a distributed system." (Wikipedia, October 2014)
**Split Brain**

In SAP HANA’s master/slave/standby failover solution, there is only one entity in the whole system, which is able to make failover decisions: the master nameserver. A slave or standby host will never execute a failover by itself. Therefore, only the master host must be considered for split brain situations.

SAP HANA would run into a split-brain situation when multiple hosts try to become master nameserver/indexserver and would access the same set of data (persistence) from disk. This would irreparably destroy the data. To overcome this problem, SAP HANA uses I/O fencing to prevent the other host from accessing the storage:

- **SAN storages**: The storage devices are locked by the current active host with SCSI-3 persistent reservations. If another host tries to mount those devices, the old host will automatically lose write permissions and the services will abort themselves.
- **NFSv3 shared storages**: The NFSv3 file lock implementation cannot be used as locks would not be released if an NFSv3 client dies, so a STONITH procedure must be provided by the storage vendor, which will reboot a failed host.
- **NFSv4 shared storages or cluster file systems like GPFS**: The file locking implementation works reliably across hosts. Non availability of a host and thus lock release is handled by the file system. A host that tries to open a persistence that is already open will fail and abort itself.

Communication network and storage network based heartbeats are used to detect activeness of other hosts and prevent unnecessary failover attempts. If the target master host detects that another master is still active, it terminates itself to let the other master continue. Without this, different hosts could try to become master and would fence each other repeatedly.

In split brain situation a quorum is sometimes used to decide, which side should ‘survive’. This makes sense in stateless compute clusters to have the bigger parts of resources remaining active. But in SAP HANA tables are bound to specific storage partitions and service instances. Tables in the other partition would not be accessible and applications typically cannot continue with some tables being inaccessible. Therefore SAP HANA lets the initial master continue.

**hdbnsutil**

Some actions supported by the hdbnsutil executable access the persistence while the system is stopped. To avoid data corruption caused by unexpected active or reviving services, this program also checks for active nameservers with network and storage based heartbeats and uses fencing to set the SCSI-3 persistent reservation.

**SAN storages**: After stopping hdbnsutil (or the nameserver) the SCSI-3 persistent reservations are intentionally not released. This ensures that no other service unintentionally accesses a persistence, e.g. still running services on other hosts after a split brain situation.

---

“A split-brain situation is a term that indicates data or availability inconsistencies originating from the maintenance of two separate data sets with overlap in scope, either because of servers in a network design, or a failure condition based on servers not communicating and synchronizing their data to each other. This last case is also commonly referred to as a network partition.” (Wikipedia, October 2014)
**Failover Duration**

The failover phase can be split into:

1. **Failure Detection**
   Several watchdogs, retries and timeouts are involved. Based on the failure condition the detection time can vary, e.g.
   
   a. **SAP HANA instance terminated or host shutdown**
      
      The checking host will immediately get errors from the OS layer and typically detects the failure in less than 1 minute.
   
   b. **Network Split**
      
      The checking host has to wait until the network times out, so failure detection typically takes 3 to 6 minutes.
      
      The timeouts could be reduced, but this is not recommended, as it would not allow recovery from short network outages, or may lead to false failover decision in the case of heavy system load, where pings can take longer.

2. **Failover Execution**
   
   The failover time is comparable to the time required for SAP HANA startup, because the services on a standby host are initially started, but run idle. During failover they do the same initialization and persistence load like regular services startup.

**Host Start Order / Landscape Restart**

All hosts can be started concurrently. The master nameserver candidates have different priorities as indicated by the role name MASTER 1, 2, 3. The first master candidate will become the active master. The indexserver roles, host roles and storage partitions are reset, meaning that all configured worker hosts are used as worker again even if the landscape was in a failed over state before shutdown. However, if a host was previously used as worker, its storage partition is kept as is, to avoid inefficient access patterns in clustered file systems, so over time the storage partitions can have different sorting compared to the initial state after installation.

**Failback**

When a failover was performed and the failed host is available again, no automatic failback will happen; the host will start as a standby. A controlled failback can be performed by stopping or restarting the configured standby host which - after a previous failover - actually is worker. Automatic failback will only happen when the complete landscape is restarted.

**Master Nameserver Candidates**

The initial host is master candidate and the first two hosts added to a landscape will become master candidates. When a standby host is added and none of the master candidates is a standby host, the last master candidate is moved to the new standby host. Having a standby host in the master candidate list allows faster master host failover as it avoids the above mentioned double failover.

**Failover Groups**

During installation and with SAP HANA studio a failover group can be configured per host. If a failover target host is available in the same group, it will be preferred over hosts from other groups. This can be used to achieve better 'locality' in large systems, e.g. to use network / storage connection with less latency. When the parameter `nameserver.ini/[failover]/cross_failover_group is`
set to false, failover is restricted to hosts in the same group. This can be used to separate differently sized hardware or separate storages.

**Master Failover without Standby Hosts**
Distributed landscapes without standby hosts may also perform a failover to ensure that the master host is always available. Of course, a slave host (and all tables located there) will be inaccessible after failover. This failover mechanism can be disabled by removing nameserver roles MASTER 2 and MASTER 3 in *SAP HANA-Studio ➔ Administration-Perspective ➔ Landscape ➔ Hosts*. Disabling is required if you use (not recommended) local storage on each host or the landscape is controlled by an external cluster manager.

**Host Auto-Failover vs. External Cluster Manager**
Instead of the built in SAP HANA Host Auto-Failover you could monitor and (re)start virtualized hosts on different hardware with an external cluster manager. With multiple SAP HANA instances this would have the advantage that fewer standby hosts would be needed, but on the other hand all failure detection and fencing logic has to be implemented externally. To avoid unnecessary SAP HANA controlled master failovers the nameserver MASTER 2 and MASTER 3 roles can be removed as described above.

**Application Configuration**
In the connection information for SAP HANA SQL client libraries (e.g. with hdbuserstore) you can configure multiple host names. All master nameserver candidates should be configured there. The master candidates can be found using the following SQL statement:

```sql
select HOST from SYS.M_LANDSCAPE_HOST_CONFIGURATION where NAMESERVER_CONFIG_ROLE like 'MASTER%' order by NAMESERVER_CONFIG_ROLE
```

**Application Error Handling**
Failover is not seamless, errors during a failure phase will be returned to the clients. Neither server nor client libraries have a built-in 'retry' logic. Applications have to be prepared and should try to reconnect.

*Master Host Failure:*
The client typically gets error -11312 (Connection to database server lost; check server and network status [System error: ...])

*Slave Host Failure:*
Basically any error code can happen, because the master connection is still available, but some tables are not accessible anymore and statements can fail at varying steps.

**User Exits**
SAP HANA provides hooks (named HA/DR Provider) with SPS 09 that can call certain function of user-defined Python script for Host Auto-Failover, STONITH, landscape start and stop as well as system replication takeover. All details can be found in [4] section “Implementing a HA/DR Provider”.

**References**
[1] SAP HANA High Availability White Paper
http://scn.sap.com/docs/DOC-60334

[2] Fibre Channel Storage Connector Admin Guide (attachment to SAP Note)
http://service.sap.com/sap/support/notes/1900823

http://scn.sap.com/docs/DOC-60313