IBM Systems
Technical white paper

August 2021

SAP HANA Agent for IBM VM Recovery Manager

Configuring and monitoring SAP HANA Agent through IBM VM Recovery Manager HA on Linux

Table of contents

Overview

Challenge

Administrators need to manually make sure SAP HANA applications are always up and running to prevent data loss. They also need to maintain replication between such instances for high availability.

Solution

This paper provides detailed explanation about SAP HANA application and its services and helps administrators to achieve high availability of such applications using the IBM VM Recovery Manager solution.

Introduction

SAP HANA is an in-memory database and is widely used for developing and deploying real-time applications. This paper describes the process of installing SAP HANA instances, creating relationship between the instances, monitoring SAP HANA Agent, maintaining replication of SAP HANA Agent across VMs, and achieving high availability of such applications using the IBM® VM Recovery Manager HA for Power Systems solution. It also helps in understanding about the daemons that run inside an SAP HANA instance.

The intended audience for this technical paper includes system administrators or any user of the VM Recovery Manager HA solution.

Prerequisites

As a prerequisite, knowledge about the topics would be helpful:

- SAP HANA application
- VM agent
  KSYS sub system

SAP HANA database

SAP HANA is an in-memory database and is a combination of hardware and software made to process massive real-time data using in-memory computing. It combines row-based and column-based database technology.

SAP HANA system core process list

The following table explains the SAP HANA system core processes.

Process name	Server / Service name	Purpose
Hdbdaemon	HDB Daemon	Hdbdaemon is responsible for starting the below mentioned servers in the right order.
Hdbcompileserver	HDB Compileserver	The compile server performs the compilation of stored procedures and programs (for example, SQLScript procedures). It runs on every host and does not maintain persistent data.
Hdbindexserver	HDB Indexserver	The index server is the crux, and is responsible for managing connection, processing SQL, handling metadata, storing row and column-based data, and so on.
Hdbnameserver	HDB Nameserver	The name server knows which table or partition of tables are located on which Index Server. In short, the Nameserver server is aware of the data location.
hdbpreprocessor	HDB Preprocessor	The preprocessor is responsible for analyzing text data and extracting the information based on the text search capabilities.
hdbwebdispatcher	HDB Web Dispatcher	The web dispatcher server processes inbound HTTP and HTTPS connections to SAP HANA Extended Application Services (XS).
Hdbxsengine	HDB XSEngine	SAP HANA XS is the application server for native SAP HANA-based web applications. It is installed with the SAP HANA system and allows developers to write and run SAP HANA-based applications without the need to run an additional application server. SAP HANA XS is also used to run web-based tools that are included in SAP HANA, for instance for administration, lifecycle management, and development.
Sapstartsrv	SAP start service	The SAP start service is responsible for starting and stopping the other services in the correct order. It also performs other functions, such as monitoring their runtime state. This service triggers the HDB Daemon server that starts the other core services.

Table 1. SAP HANA system core processes

Architecture

Software

- SAP HANA for Linux (RHEL/SUSE)

Hardware

- 8 core CPU
- 80 GB memory
- 200 GB hard disk space

Architecture overview: VM Recovery Manager solution and SAP HANA application

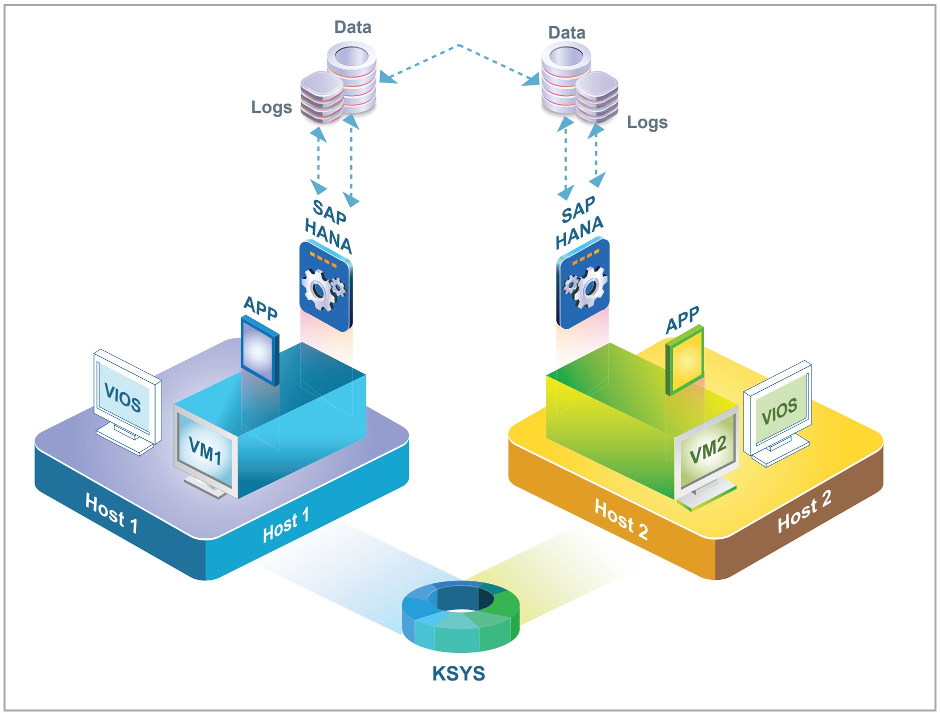

Figure 1. SAP HANA application installed on VMs with KSYS and VIOS

The following list explains the components in Figure 1.

- **VM**: Virtual machines (VMs), also known as logical partitions (LPARs).
- **HA**: High availability, to provide continuous processing for all important applications.
- **KSYS**: The controlling system, which is a controlling software for the HA operation.
- **VIOS**: A special LPAR that hosts I/O resources to provide advanced virtualization capabilities across other client logical partitions (VMs).
- **SAP HANA:** SAP HANA is a high-performance in-memory database that speeds data-driven, real-time decisions and actions.
- **VM Agent**: VM Monitor daemon, which is used to monitor the virtual machine and to monitor applications on the virtual machines.

Configure SAP HANA for system replication

This section explains how to set up the SAP HANA application for system replication.

Prerequisites

Before setting up the SAP HANA application for system replication, make sure that the following prerequisites are fulfilled:

- User should have more than one SAP HANA host, SAP system ID, and the instance number (IP address for the local host in /etc/host must be same for all the hosts involved in this process).
- All hosts targeted for system replication must have all the other hosts defined in their respective /etc/hosts file.

Configuration

Perform the following steps using the command line interface (CLI) to configure SAP HANA for system replication:

1. Configure the primary host.

   a. On the system that needs to be configured as the primary host, set the path to the *exe* directory of SAP HANA installation
   `PATH="$PATH:/usr/sap/<SID>/HDB02/exe"`

   b. Create a DB user (ignore if you want to use the system DB user created during installation), where `slehasync` is the username and `L1nuxlab` is the password for that user.

   ```
   hdbsql -u system -i 01 'CREATE USER slehasync
   PASSWORD L1nuxlab'
   hdbsql -u system -i 01 'GRANT DATA ADMIN TO
   slehasync'
   hdbsql -u system -i 01 'ALTER USER slehasync
   DISABLE PASSWORD LIFETIME'
   ```

   c. Create a user secure key, where `slehaloc` is the newly created key for accessing the DB. Username is `slehasync`, password is `L1nuxlab`, and the instance number is `30115` **–** `01`. This information provided during installation must be run/created on all hosts participating in system replication.

   ```
   hdbuserstore SET slehaloc localhost:30115
   slehasync L1nuxlab
   ```

d. Verify if the newly created user can access the DB with the newly created key, where `slehaloc` is the key and **-U** tells to use the key for credentials.

```
hdbsql -U slehaloc "select * from dummy"
```

Note: Exit from the resulting window by pressing **q**.

e. Create a backup of the primary host DB (mandatory) in a file named backup_file, using the key `slehaloc`.

```
hdbsql -U  slehaloc  "backup_file
('SC0_INITIAL')"
```

Or, use the following command to create a backup of the primary host with SYSTEM as the user.

```
hdbsql -i 01 -u system -p Sap01database -d
SYSTEMDB "BACKUP DATA USING FILE ('/tmp/foo')"

hdbsql -u system -i 02 "BACKUP DATA USING FILE
('SC0_INITIAL')"
```

If there is a newer version of HANA, you need to create a complete backup of SYSTEM DB. To do so, run the following command:

```
hdbsql -u system -i 02 -d SystemDB "BACKUP DATA
USING FILE
('/usr/sap/S02/HDB02/backup/data/SYSTEMDB/SC0_INI
TIAL') COMMENT 'Full SYSTEMDB data backup of Type
FILE'"
```

Note: Without a valid backup you could not bring SAP HANA into a system replication configuration.

f. Define the current host as the primary host and enable system replication on it. In the following example, the primary site is named HANA_SITE_01.

```
hdbnsutil -sr_enable --name=HANA_SITE_01
```

g. Verify the state of replication.

```
hdbnsutil -sr_state
```

The resulting mode is displayed as **primary** along with the site ID and name.

h. Check the status of replication with details including configuration parameters being used, and so on.

```
hdbcons -e hdbindexserver "replication info"
```

2. Configure the secondary host or hosts using the commands provided in the following steps:
   a. Stop the SAP HANA DB instance.
   ```
   sapcontrol -nr 01 -function StopSystem HDB
   ```
   Where, 01 is the instance number.

   b. Verify the SAP HANA DB status.
   ```
   sapcontrol -nr 02 -function GetSystemInstanceList
   sapcontrol -nr 02 -function GetProcessList
   ```

   c. Copy the following directories from the primary host to the secondary host (source and target path is same).
   ```
   /hana/shared/S02/global/security/rsecssfs/data
   /hana/shared/<SID>/global/security/rsecssfs/key
   ```

   d. Register this site as the secondary host (which makes it entitled for replication).
   ```
   hdbnsutil -sr_register --remoteHost=r2r1m1p36 --
   remoteInstance=01 --replicationMode=sync --
   name=HANA_SITE_47
   ```

   Where,
   - r2r1m1p36 is the hostname of the primary host.
   - 01 is the instance number, which is same on secondary as well as the primary hosts.
   - sync is the mode of replication (sync/async/syncmem).
   - HANA_SITE_47 is the name you want to give to this secondary site.

   e. Start the SAP HANA DB instance using the following command:
   ```
   sapcontrol -nr 01 -function StartSystem HDB
   ```

3. Verify that system replication is working.
   a. On the primary host, run the following command to return the status.

   ```
   hdbcons -e hdbindexserver "replication info"
   ```

   The output shows all replication parameters along with the status (including the percentage of replication completed for an ongoing process).

   Note: At this stage, the DB would not be available and therefore, a connection error will be displayed.

Run the following command to find the replication status:

```
hdbsql -U slehaloc 'select distinct
REPLICATION_STATUS from
SYS.M_SERVICE_REPLICATION'
```

Or, run as root user:

```
hdbsql -U slehaloc 'select * from
SYS.M_SERVICE_REPLICATION'
```

b. Trigger a `takeover` event on the secondary host (and before that, stop the HANA DB on the primary host).

```
hdbnsutil -sr_takeover
```

Note: After the takeover event is complete, the original primary host remains as the primary host in addition to the new primary host. So, manually configure the original primary host as the secondary host.

Again, run the command mentioned in step (a) and this time, the result should be available at secondary host (which now is the primary host.)

Integration with VM Recovery Manager HA

This section explains the procedure to integrate the SAP HANA Agent with VM Recovery Manager.

Prerequisites

- Two VMs with required hardware SAP capabilities configured and SAP HANA running by creating a relationship between them.
- A KSYS subsystem managing these two VMs.

Configuring SAP HANA instances on VM node

If you configure the high-availability function at the VM level or the application level, you must set up the VM Agent. IBM VM Recovery Manager HA for Power Systems supports the following SAP HANA configurations:

- **SAP HANA scale up configuration with host-based replication**: Users can create a replication between two SAP HANA nodes and add them to a VM agent. IBM VM Recovery Manager HA for Power Systems manages the SAP HANA nodes and replication between the two nodes, such as takeover of primary node failures.

- **SAP HANA scale up configuration without replication**: Users can install the SAP HANA DB and add it to a VM agent. The VM agent monitors the database status and manages any failures.

Agent scripts

The KSYS VM daemon uses the following agent scripts to start, stop, and monitor the SAP HANA application:

- /usr/sbin/agents/saphana/startsaphana
- /usr/sbin/agents/saphana/stopsaphana
- /usr/sbin/agents/saphana/monitorsaphana

The `ksysvmmgr` command provides a consistent interface for the VM agent to manage the virtual machine) and the applications that are running in the virtual machine. Run the following command to add the SAP HANA VM agent:

```
ksysvmmgr [-s] [-l {0|1|2|3}] add app <NAME> type=SAPHANA
instancename=<VALUE#1>
database=<VALUE#2> configfile=<VALUE#3> [<ATTR#n>=<VALUE#n>]
```

Examples:

- Add the SAP HANA application without replication:
  ```
  ksysvmmgr -s add app sapapp1 type=SAPHANA instancename=S01
  database=HDB01
  ```

- Add the SAP HANA application with replication:
  ```
  ksysvmmgr -s add app sapapp2 type=SAPHANA instancename=S01
  database=HDB01
  configfile=/var/ ksys/config/samples/SAPHANAconfig.xml
  ```

In these examples, **S01** is the SAP HANA system ID, **HDB01** is the database name.

The following table explains each attribute.

Attribute	**Description**
`Type`	While creating a SAP HANA application, the `type` attribute must have the value, SAPHANA.
`instancename`	The `instancename` attribute must be specified with SAP HANA system ID.
`database`	The `database` attribute must be specified with the SAP HANA database name.
`configfile`	The `configfile` attribute specifies the file path of the configuration file, which stores the settings of the application configuration. This attribute is not required for the SAP HANA application without replication configuration.

Table 2. KSYS command attributes

Users must specify the path of the configuration file while adding the SAP HANA application with replication configuration. A sample configuration file, SAPHANAconfig.XML is provided in the /var/ksys/config/samples folder. You can use this sample file by updating the attribute values. If you do not specify the configuration file path or appropriate values in the configuration file, the SAP HANA application will be added without replication configuration.

The SAPHANAconfig.xml file contains the following attributes:

`instanceId`, `replication`, `role`, `localsite`, `remotesite`, `secondarymode`, `virtualip`, `interfacename`, `executabledir`, `timeout`, and `remotenode`.

If the `replication` attribute of the instance is set to `yes`, you must specify values for all mandatory attributes (for example, `replication`, `role`, `localsite`, `remotesite`, `secondarymode`, and `remotenode`). If you do not specify values for all mandatory attributes, the SAP HANA application will not be added to the virtual machine.

Few attributes are optional, and they are: `virtualip`, `interfacename`, `executabledir`, and `timeout`.

The following table describes the attributes present in the SAP HANA configuration file.

Attribute name	Description
`SAPinstance id`	The SAP HANA system ID must be specified while installing the SAP HANA application.
`replication`	It defines whether the SAP HANA application runs in stand-alone or in replication mode. The keyword `yes` indicates that the SAP HANA application is in the replication mode. The keyword no indicates that the SAP HANA application is in stand-alone mode.
`role`	The value of this attribute can be `primary`, `sync`, `syncmem`, or `async`. For a primary node, the value of this attribute is `primary` and for secondary node, the value of this attribute is `sync`, `syncmem`, or `async`.
`localsite`	It is the site name of the current node (the value that is specified while configuring the SAP HANA replication).
`remotesite`	It is the site name of the remote node (the value that is specified while configuring the SAP HANA replication).
`secondarymode`	It is the mode of secondary node. The value of the attribute can be `sync`, `syncmem`, or `async`.
`remotenode`	It is the hostname of the remote node. This name should be the same as the name shown in the output of the

	following SAP command: hdbnsutil - sr_state
`virtualip`	It is the service or the virtual IP address that can be aliased to the interface name and through which the primary or secondary node can be accessed by other applications.
`interfacename`	It is the interface on which the specified virtual or service IP address will be aliased.
`executabledir`	It is the directory path where the shared libraries and executable files of SAP HANA VM Agent are present.
`timeout`	It is the duration of time within which the SAP_HANA command is expected to complete. This value is indicated in seconds. The default value is 120 seconds.

Table 3. SAP HANA configuration file attributes

Refer to an example SAP HANA configuration file without replication setup:

```
<?xml version="1.0" encoding="UTF-8"?>
<HANAConfig>
<NOTE: PLEASE UNCOMMENT THE REQUIRED ATTRIBUTES AND SET
APPROPRIATE VALUES.
If replication attribute is 'yes', then all below attributes are
mandatory.>
<SAPinstance id="S01">
<replication>no</replication>
<role>primary</role>
<localsite>local.hostname.ibm.com</localsite>
<remotesite>remote.hostname.ibm.com</remotesite>
<secondarymode>sync/syncmem/async</secondarymode>
<virtualip>192.168.2.3</virtualip>
<interfacename>eth2</interfacename>
<remotenode>remotenode_name</remotenode>
<timeout>120</timeout>
<executabledir>/usr/sap/S01/HDB01/exe</executabledir>
</SAPinstance>
</HANAConfig>
```

After adding the SAP HANA Agent, using the `ksysmgr query app` command, users can query the application and validate the attributes.

The next topic explains how to create application dependency at the KSYS node and monitor it.

Creating application dependency at KSYS node

The following section explains the steps to create an application dependency at KSYS node.

Primary-secondary dependency at KSYS
The applications on KSYS node can be configured under the `primary_secondary` dependency type. This creates dependency between applications, which have a hierarchical primary-secondary structure across VMs.

The `ksysmgr` command provides a consistent interface to configure the controller system (KSYS) and to perform VM Recovery Manager HA operations. This command can be run from a terminal or a script.

To establish dependency between the primary application and the secondary application, run the following command:

```
ksysmgr add app_dependency <name> type=<primary_secondary>
app_list=<vm1:app1,vm2:app2>
```

Note: The `app_list` attribute must have only two `vmname:appname` pairs for the `primary_secondary` structure of applications across VMs.

You can verify the dependency between applications across VMs and also delete the dependency between applications.

- To verify a dependency that you have created, run the following command:
  `ksysmgr query app_dependency [name]`

- To delete a dependency that you have created, run the following command:
  `ksysmgr delete app_dependency [name]`

Automatic application takeover

After creating the dependency at KSYS sub system, KSYS learns about the SAP HANA application attributes (such as application status, modes, states, and so on) from both the VMs.

The VM agent is continuously monitoring the applications configured on its node and keeps the KSYS informed about any change, as KSYS will be required to take appropriate actions, whenever required. When a VM agent updates about a primary application failure to KSYS, KSYS sends a command to take over the secondary application running on the other VM agent node.

Example: `action="takeover"`
When the secondary VM receives this command, it invokes commands internally to take over the SAP instance. After successful completion, the VM agent informs KSYS about this status change. Now KSYS is aware that a host has the SAP instance role as *primary* and other application status marked ‘down’. Later

at any point of time when the defective application is brought normal, the VM agent will share this update to KSYS and KSYS will invoke a command to start this SAP instance in the *secondary* role.

Example: `start:secondary`
The process to update the application status is done by VM agent, and the same reflects at KSYS. The VM agent monitors the applications on its node, and KSYS is ready to take actions, when required.

Summary

This white paper enabled users to understand the end-to-end HA solution by VM Recovery Manager and the process involved in configuring and monitoring SAP HANA application, their health, takeover, auto restart applications whenever required.

Related links

- Introduction to High Availability for SAP HANA
- IBM VM Recovery Manager HA for Power Systems

About the author

Manish Laddha is an advisory software engineer in the VM Recovery Manager product team. He has more than 3 years of experience in the IBM Power platform and has knowledge about disaster recovery and high availability. You can reach Manish at mladdha0@in.ibm.com.

IBM Systems
3039 Cornwallis Road
RTP, NC 27709

Produced in the United States of America

References in the publication to IBM products or services do not imply that IBM intends to make them available in all countries in the IBM operates.

♲ Please recycle