Automating SAP system copy operations with Libelle SystemCopy

NetApp solutions for SAP

NetApp
August 18, 2025

Table of Contents

Automating SAP system copy operations with Libelle SystemCopy

TR-4929: Automating SAP system copy operations with Libelle SystemCopy

NetApp solutions for optimizing SAP lifecycle management are integrated into SAP AnyDBs and SAP HANA databases. In addition, NetApp integrates into SAP lifecycle management tools, combining efficient application-integrated data protection with the flexible provisioning of SAP test systems.

Authors:

Holger Zecha, Tobias Brandl, NetApp
Franz Diegruber, Libelle

In today's dynamic business environment, companies must provide ongoing innovation and react quickly to changing markets. Under these competitive circumstances, companies that implement greater flexibility in their work processes can adapt to market demands more effectively.

Changing market demands also affect a company's SAP environments such that they require regular integrations, changes, and updates. IT departments must implement these changes with fewer resources and over shorter time periods. Minimizing risk when deploying those changes requires thorough testing and training which require additional SAP systems with actual data from production.

Traditional SAP lifecycle-management approaches to provision these systems are primarily based on manual processes. These manual processes are often error-prone and time-consuming, delaying innovation and the response to business requirements.

NetApp solutions for optimizing SAP lifecycle management are integrated into SAP AnyDBs and SAP HANA databases. In addition, NetApp integrates into SAP lifecycle management tools, combining efficient application-integrated data protection with the flexible provisioning of SAP test systems.

While these NetApp solutions solve the issue of efficiently managing enormous amounts of data even for the largest databases, full end-to-end SAP system- copy and refresh operations have to include pre- and post-copy activities to completely change the identity of the source SAP system to the target system. SAP describes the required activities in their SAP homogenous system copy guide. To further reduce the number of manual processes and to improve the quality and stability of a SAP system copy process, our partner Libelle has developed the Libelle SystemCopy (LSC) tool. We have jointly worked with Libelle to integrate the NetApp solutions for SAP system copies into LSC to provide full end-to-end automated system copies in record time.

Application-integrated Snapshot copy operation

The ability to create application-consistent NetApp Snapshot copies on the storage layer is the foundation for the system copy and system clone operations described in this document. Storage-based Snapshot copies are created with the NetApp SnapCenter Plug-In for SAP HANA or SAP Any DBs on native NetApp ONTAP systems or by using the Microsoft Azure Application Consistent Snapshot tool (AzAcSnap) and interfaces provided by the SAP HANA and Oracle database running in Microsoft Azure. When using SAP HANA, SnapCenter and AzAcSnap register Snapshot copies in the SAP HANA backup catalog so that the backups can be used for restore and recovery as well as for cloning operations.

Off-site backup and/or disaster recovery data replication

Application-consistent Snapshot copies can be replicated on the storage layer to an off-site backup site or a disaster recovery site controlled by SnapCenter on-premises. Replication is based on block changes and is therefore space and bandwidth efficient. The same technology is available for SAP HANA and Oracle systems running in Azure with Azure NetApp Files by using the Cross Region Replication (CRR) feature to efficiently replicate Azure NetApp Files volumes between Azure regions.

Use any Snapshot copy for SAP system copy or clone operations

NetApp technology and software integration allows you to use any Snapshot copy of a source system for an SAP system copy or clone operation. This Snapshot copy can be either selected from the same storage that is used for the SAP production systems, the storage that is used for off-site backups (such as Azure NetApp Files backup in Azure), or the storage at the disaster recovery site (Azure NetApp Files CRR target volumes). This flexibility allows you to separate development and test systems from production if required and covers other scenarios, such as the testing of disaster recovery at the disaster recovery site.

Automation with integration

There are various scenarios and use cases for the provisioning of SAP test systems, and you might also have different requirements for the level of automation. NetApp software products for SAP integrate into database and lifecycle management products from SAP and other third-party vendors (for example, Libelle) to support different scenarios and levels of automation.

NetApp SnapCenter with the plug-in for SAP HANA and SAP AnyDBs or AzAcSnap on Azure is used to provision the required storage- volume clones based on an application-consistent Snapshot copy and to execute all required host and database operations up to a started SAP database. Depending on the use case, SAP system copy, system clone, system refresh, or additional manual steps such as SAP postprocessing might be required. More details are covered in the next section.

A fully automated, end-to-end provisioning or refresh of SAP test systems can be performed by using Libelle SystemCopy (LSC) automation. The integration of SnapCenter or AzAcSnap into LSC is described in more detail in this document.

Libelle SystemCopy

Libelle SystemCopy is a framework-based software solution to create fully automated system and landscape copies. With the proverbial touch of a button, QA and test systems can be updated with fresh production data. Libelle SystemCopy supports all conventional databases and operating systems, provides its own copy mechanisms for all platforms but, at the same time, integrates backup/restore procedures or storage tools such as NetApp Snapshot copies and NetApp FlexClone volumes. The activities that are necessary during a system copy are controlled from outside the SAP ABAP stack. In this way, no transports or other changes are required in the SAP applications. Generally, all steps necessary to successfully complete a system copy procedure can be categorized into four steps:

- **Check phase.** Check the involved system environments.
- **Pre phase.** Prepare the target system for a system copy.
- **Copy phase.** Provide a copy of the actual production database to the target system from the source.
- **Post phase.** All tasks after the copy to complete the homogeneous system copy procedure and to provide an updated target system.

During the copy phase, NetApp Snapshot and FlexClone functionality is used to minimize the time needed to a couple of minutes even for the largest databases.

For the Check, Pre, and Post phases, LSC comes with 450+ preconfigured tasks covering 95% of typical refresh operations. As a result, LSC embraces automation following SAP standards. Due to the software-defined nature of LSC, system refresh processes can be easily adjusted and enhanced to meet the specific needs of customer SAP environments.

Use cases for SAP system refresh and cloning

There are multiple scenarios in which data from a source system must be made available to a target system:

- Regular refresh of quality assurance and test and training systems
- Creating break fix or repair system environments to address logical corruption
- Disaster recovery test scenarios

Although repair systems and disaster recovery test systems are typically provided using SAP system clones (which don't require extensive post-processing operations) for refreshed test and training systems, these post-processing steps must be applied to enable coexistence with the source system. Therefore, this document focuses on SAP system refresh scenarios. More details about the different use cases can be found in the technical report TR-4667: Automating SAP HANA System Copy and Clone Operations with SnapCenter.

The remainder of this document is separated into two parts. The first part describes the integration of NetApp SnapCenter with Libelle SystemCopy for SAP HANA and SAP AnyDBs systems running on NetApp ONTAP systems on-premises. The second part describes the integration of AzAcSnap with LSC for SAP HANA systems running in Microsoft Azure with Azure NetApp Files provided. Although the underlaying ONTAP technology is identical, Azure NetApp Files provides different interfaces and tool integration (for example, AzAcSnap) compared to native ONTAP installation.

SAP HANA system refresh with LSC and SnapCenter

This section describes how to integrate LSC with NetApp SnapCenter. The integration between LSC and SnapCenter supports all SAP- supported databases. Nevertheless, we must differentiate between SAP AnyDBs and SAP HANA because SAP HANA provides a central communication host that is not available for SAP AnyDBs.

The default SnapCenter agent and database plug-in installation for SAP AnyDBs is a local installation from the SnapCenter agent in addition to the corresponding database plug-in for the database server.

In this section, the integration between LSC and SnapCenter is described using an SAP HANA database as an example. As previously stated for SAP HANA, there are two different options for the installation of the SnapCenter agent and SAP HANA database plug-in:

- **A standard SnapCenter agent and SAP HANA Plug-in installation.** In a standard installation, the SnapCenter agent and the SAP HANA Plug-in are locally installed on the SAP HANA database server.
- **A SnapCenter installation with a central communication host.** A central communication host is installed with the SnapCenter agent, the SAP HANA Plug-in, and the HANA database client that handles all database-related operations needed to back up and restore an SAP HANA database for several SAP HANA systems in the landscape. Therefore, a central communication host does not need to have a complete SAP HANA database system installed.

For more details regarding these different SnapCenter agents and SAP HANA database plug-in installation options, see the technical report TR-4614: SAP HANA backup and recovery with SnapCenter.

The following sections highlight the differences between integrating LSC with SnapCenter using either the

standard installation or the central communication host. Notably, all configuration steps that are not highlighted are the same regardless of the installation option and the database used.

To perform an automated Snapshot copy-based backup from the source database and create a clone for the new target database, the described integration between LSC and SnapCenter uses the configuration options and scripts described in TR-4667: Automating SAP HANA System Copy and Clone Operations with SnapCenter.

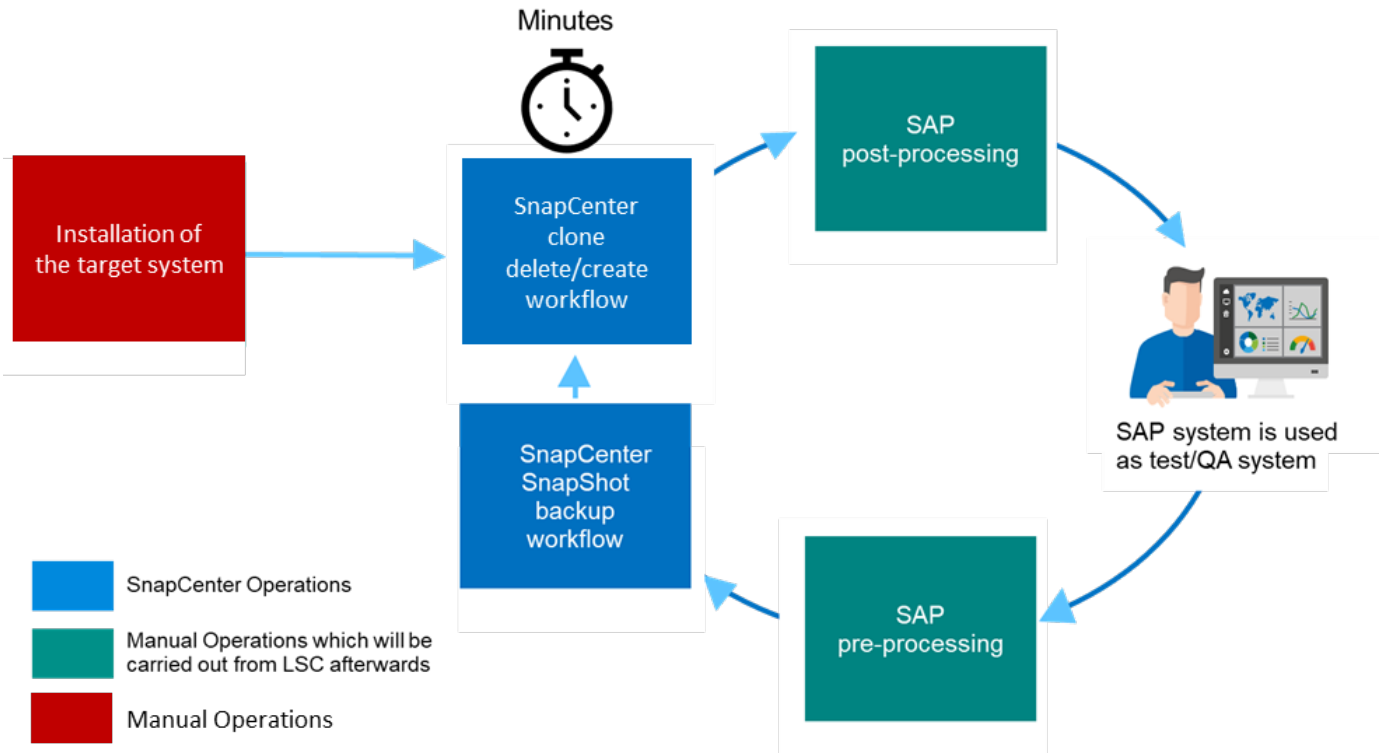
Overview

The following figure shows a typical high-level workflow for an SAP system refresh lifecycle with SnapCenter without LSC:

1. A one-time, initial installation and preparation of the target system.
2. Manual preprocessing (exporting licenses, users, printers, and so on).
3. If necessary, the deletion of an already existing clone on the target system.
4. The cloning of an existing Snapshot copy of the source system to the target system performed by SnapCenter.
5. Manual SAP post-processing operations (importing licenses, users, printers, disabling batch jobs, and so on).
6. The system can then be used as test or QA system.
7. When a new system refresh is requested, the workflow restarts at step 2.

SAP customers know that the manual steps colored in green in the figure below are time consuming and error prone. When using LSC and SnapCenter integration, these manual steps are carried out with LSC in a reliable and repeatable manner with all necessary logs needed for internal and external audits.

The following figure provides an overview of the general SnapCenter-based SAP system refresh procedure.



Prerequisites and limitations

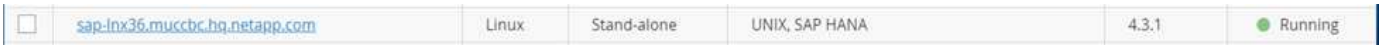
The following prerequisites must be fulfilled:

- SnapCenter must be installed. The source and target system must be configured in SnapCenter, either in a standard installation or by using a central communication host. Snapshot copies can be created on the source system.
- The storage backend must be configured properly in SnapCenter, as shown in the image below.

The next two images cover the standard installation in which the SnapCenter agent and the SAP HANA Plug-in are installed locally on each database server.

The SnapCenter agent and the appropriate database plug-in must be installed on the source database.

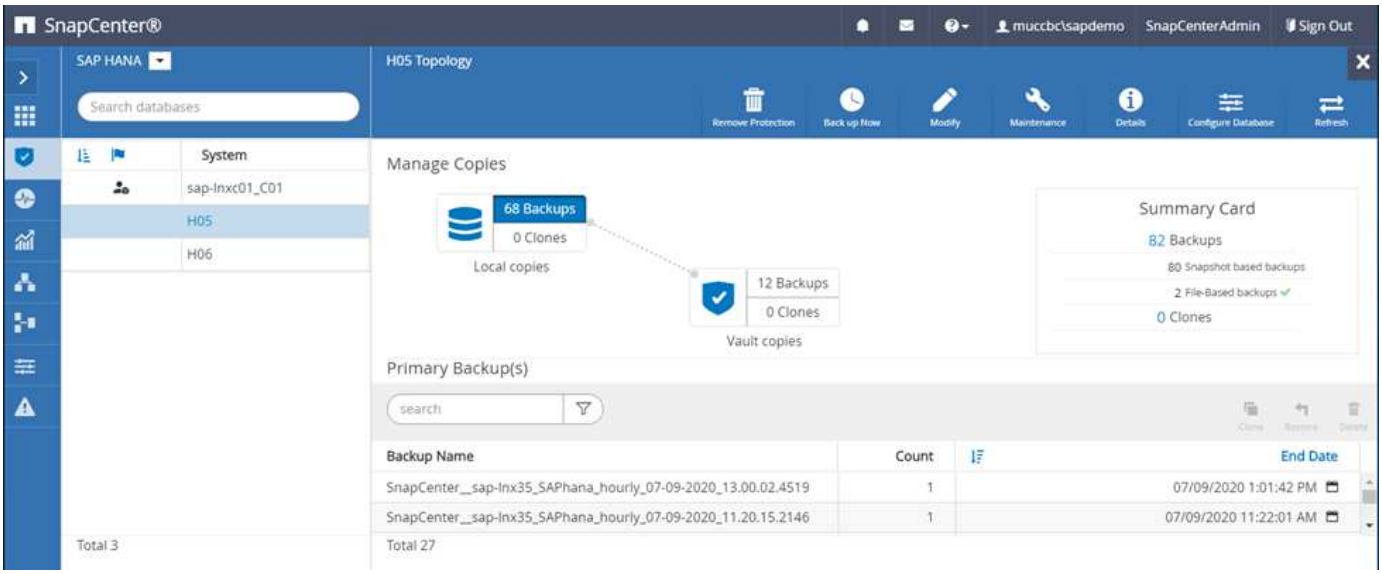
The SnapCenter agent and the appropriate database plug-in must be installed on the target database.



The following image portrays central communication-host deployment in which the SnapCenter agent, the SAP HANA Plug-in, and the SAP HANA database client are installed on a centralized server (such as the SnapCenter Server) to manage several SAP HANA systems in the landscape.

The SnapCenter agent, the SAP HANA database plug-in, and the HANA database client must be installed on the central communication host.

The backup for the source database must be configured properly in SnapCenter so that the Snapshot copy can be successfully created.



The LSC master and the LSC worker must be installed in the SAP environment. In this deployment, we also installed the LSC master on the SnapCenter Server and the LSC worker on the target SAP database server, which should be refreshed. More details are described in the following section "Lab setup."

Documentation resources:

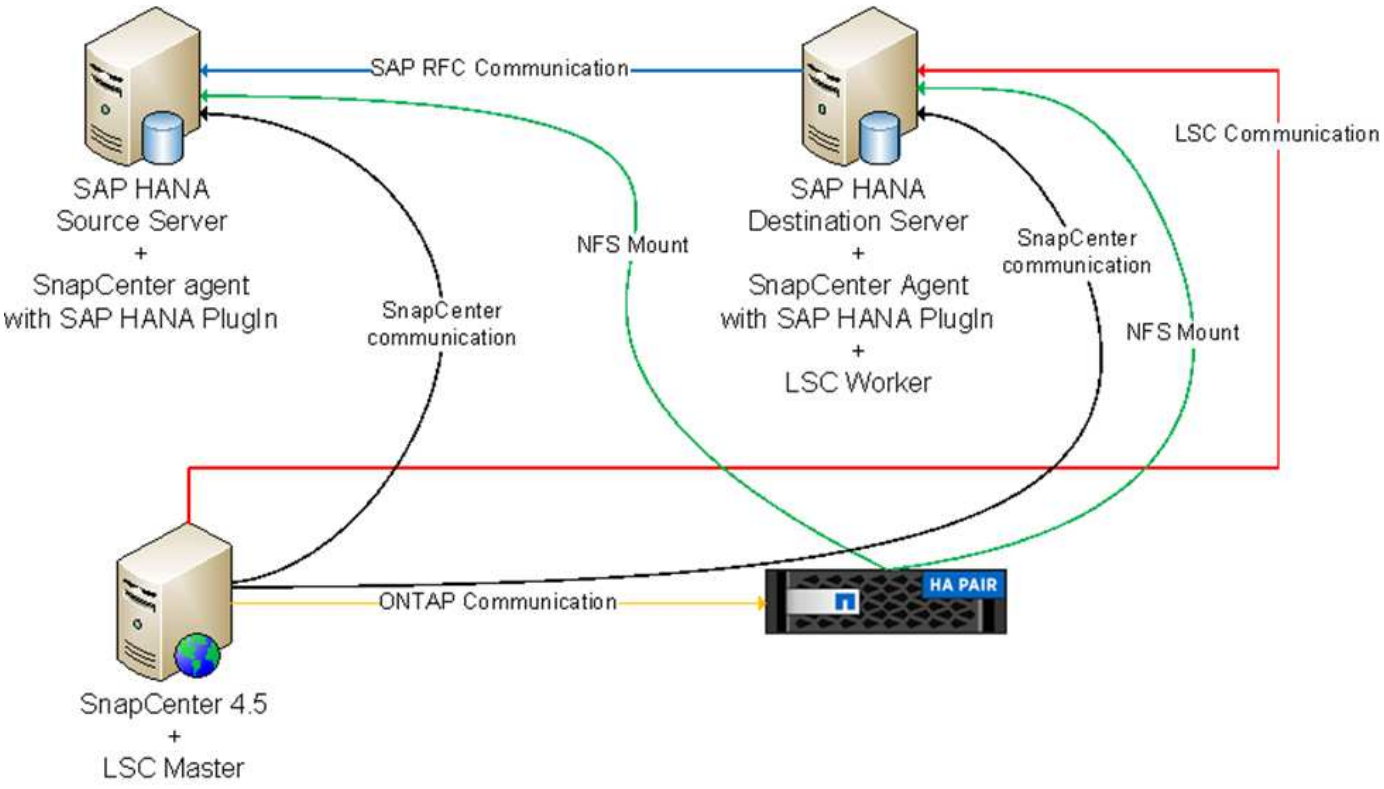
- SnapCenter Documentation Center
- TR-4700: SnapCenter Plug-In for Oracle Database
- TR-4614: SAP HANA Backup and Recovery with SnapCenter
- TR-4667: Automating SAP HANA System Copy and Clone Operations with SnapCenter
- TR-4769 -SnapCenter Best Practices and Sizing Guidelines
- SnapCenter 4.6 Cmdlet Reference Guide

Lab setup

This section describes an example architecture that was set up in a demo data center. The setup was divided into a standard installation and an installation using a central communication host.

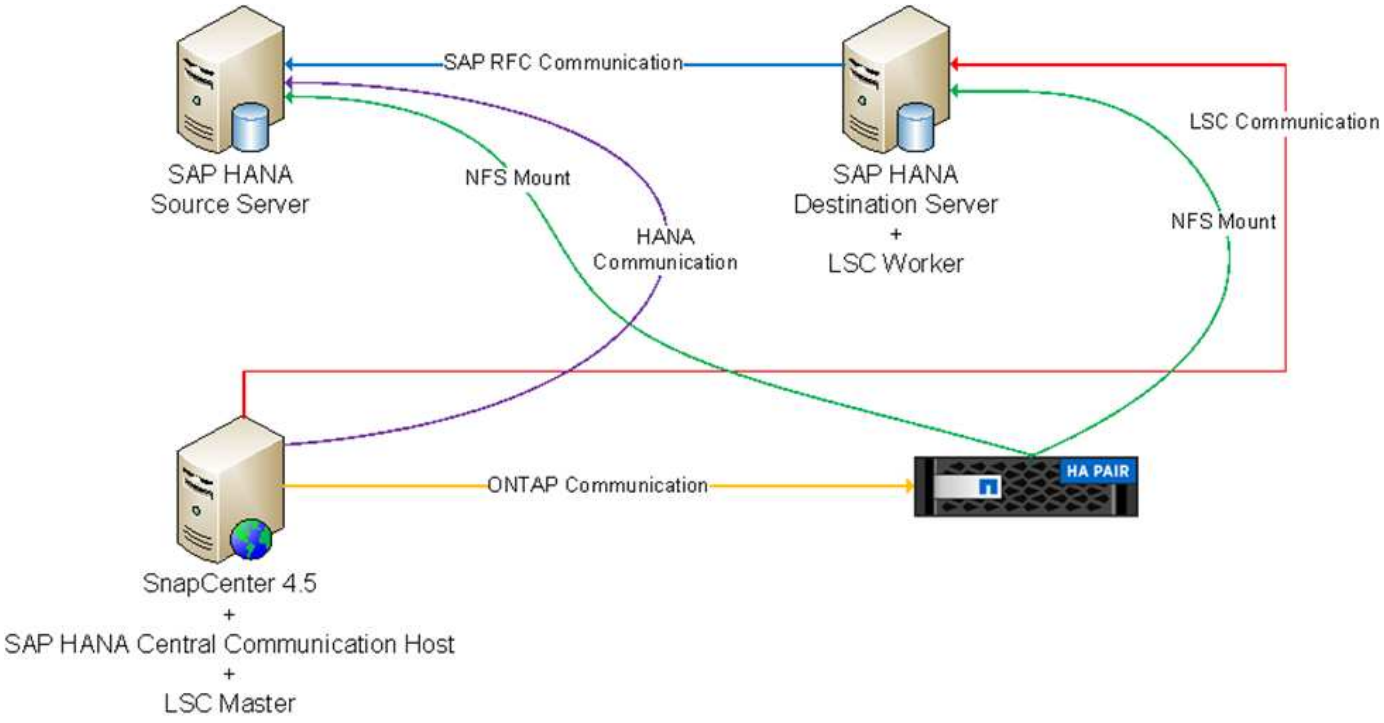
Standard installation

The following figure shows a standard installation in which the SnapCenter agent together with the database plug-in was installed locally on the source and the target database server. In the lab setup, we installed the SAP HANA Plug-in. In addition, the LSC worker was also installed on the target server. For simplification and to reduce the number of virtual servers, we installed the LSC master on the SnapCenter Server. The communication between the different components is illustrated in the following figure.



Central communication host

The following figure shows the setup using a central communication host. In this configuration, the SnapCenter agent together with the SAP HANA Plug-in and the HANA database client was installed on a dedicated server. In this setup, we used the SnapCenter Server to install the central communication host. In addition, the LSC worker was again installed on the target server. For simplification and to reduce the number of virtual servers, we decided to also install the LSC master on the SnapCenter Server. The communication between the different components is illustrated in the figure below.



Initial one-time preparation steps for Libelle SystemCopy

There are three main components of an LSC installation:

- **LSC master.** As the name suggests, this is the master component that controls the automatic workflow of a Libelle-based system copy. In the demo environment, the LSC master was installed on the SnapCenter Server.
- **LSC worker.** An LSC worker is the part of the Libelle software that typically runs on the target SAP system and executes the scripts required for the automated system copy. In the demo environment, the LSC worker was installed on the target SAP HANA application server.
- **LSC satellite.** An LSC satellite is a part of the Libelle software that runs on a third-party system on which further scripts must be executed. The LSC master can also fulfill the role of an LSC satellite system at the same time.

We first defined all the involved systems inside LSC, as shown in the following image:

- **172.30.15.35.** The IP address of the SAP source system and the SAP HANA source system.
- **172.30.15.3.** The IP address of the LSC master and the LSC satellite system for this configuration. Because we installed the LSC master on the SnapCenter Server, the SnapCenter 4.x PowerShell Cmdlets are already available on this Windows host because they were installed during the SnapCenter Server installation. So, we decided to enable the LSC satellite role for this system and execute all SnapCenter PowerShell Cmdlets on this host. If you use a different system, make sure you install the SnapCenter PowerShell Cmdlets on this host according to the SnapCenter documentation.
- **172.30.15.36.** The IP address of the SAP destination system, the SAP HANA destination system, and the LSC worker.

Instead of IP addresses, host names, or fully qualified domain names can also be used.

The following image shows the LSC configuration of the master, worker, satellite, SAP source, SAP target, source database, and target database.

System Identifier	Worker	Source SAP	Source Database	Target SAP	Target Database	Satellite System
172.30.15.35		☑	☑	☐	☐	☐
172.30.15.3	172.30.15.3:9000	☐	☐	☐	☐	☑
172.30.15.36	172.30.15.36:9000	☐	☐	☑	☑	☐

For the main integration, we must again separate the configuration steps into the standard installation and the installation using a central communication host.

Standard installation

This section describes the configuration steps needed when using a standard installation where the SnapCenter agent and the necessary database plug-in are installed on the source and target systems. When using a standard installation, all tasks needed to mount the clone volume and to restore and recover the target system are carried out from the SnapCenter agent that is running on the target database system on the server itself. This allows access to all the clone-related details that are available through environmental variables from the SnapCenter agent. Therefore, you only need to create one additional task in the LSC copy phase. This task carries out the Snapshot copy process on the source database system and the clone and restore and recovery process on the target database system. All SnapCenter related tasks are triggered by using a PowerShell script that is entered in the LSC task `NTAP_SYSTEM_CLONE`.

The following image shows LSC task configuration in the copy phase.

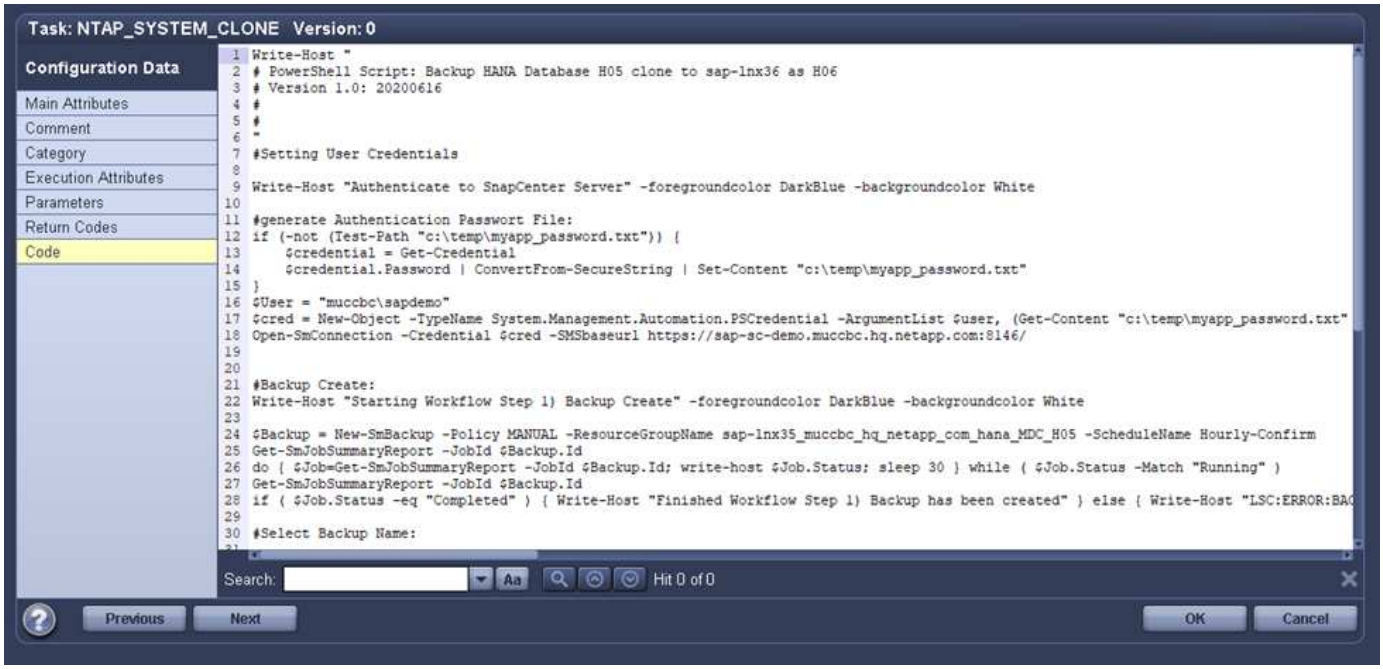
	copy	Copy Phase		phase
	copy 1	NTAP_SYSTEM_CLONE	NetApp SnapShot and Clone	psh
	copy 2	NTAP_SYSTEM_CLONE_CP	NetApp SnapShot and Clone	psh
	copy 3	NTAP_MNT_RECOVER_CP	Mount Volume and Recover HANA Database	cmd
	copy 4	LPDBBCKP	Backup Source DB in Filesystem	lsh
	copy 5	LPDBCPYFLS	Copy DB Backup Files From Source to Target System	lsh
	copy 6	LTDBRESTORE	Restore DB Files	lsh
	copy 7	LTDBRESTORE_TENANT	Restore DB Files for Tenant Database	lsh
	post	Post Phase		phase

The following image highlights the configuration of the `NTAP_SYSTEM_CLONE` process. Because you are executing a PowerShell script, this Windows PowerShell script is executed on the satellite system. In this instance, this is the SnapCenter Server with the installed LSC master that also acts as a satellite system.

Because LSC must be made aware of whether the Snapshot copy, cloning, and recovery operation has been successful, you must define at least two return code types. One code is for a successful execution of the script, and the other code is for a failed execution of the script, as shown in the following image.

- `LSC:OK` must be written from the script to standard out if the execution was successful.
- `LSC:ERROR` must be written from the script to standard out if the execution has failed.

The following image shows part of the PowerShell script that must run to execute a Snapshot-based backup on the source database system and a clone on the target database system. The script is not intended to be complete. Rather, the script shows how integration between LSC and SnapCenter can look and how easy it is to set it up.



Because the script is executed on the LSC master (which is also a satellite system), the LSC master on the SnapCenter Server must be run as a Windows user that has appropriate permissions to execute backup and cloning operations in SnapCenter. To verify whether the user has appropriate permission, the user should be able execute a Snapshot copy and a clone in the SnapCenter UI.

There is no need to run the LSC master and the LSC satellite on the SnapCenter Server itself. The LSC master and the LSC satellite can run on any Windows machine. The prerequisite for running the PowerShell script on the LSC satellite is that the SnapCenter PowerShell cmdlets have been installed on the Windows Server.

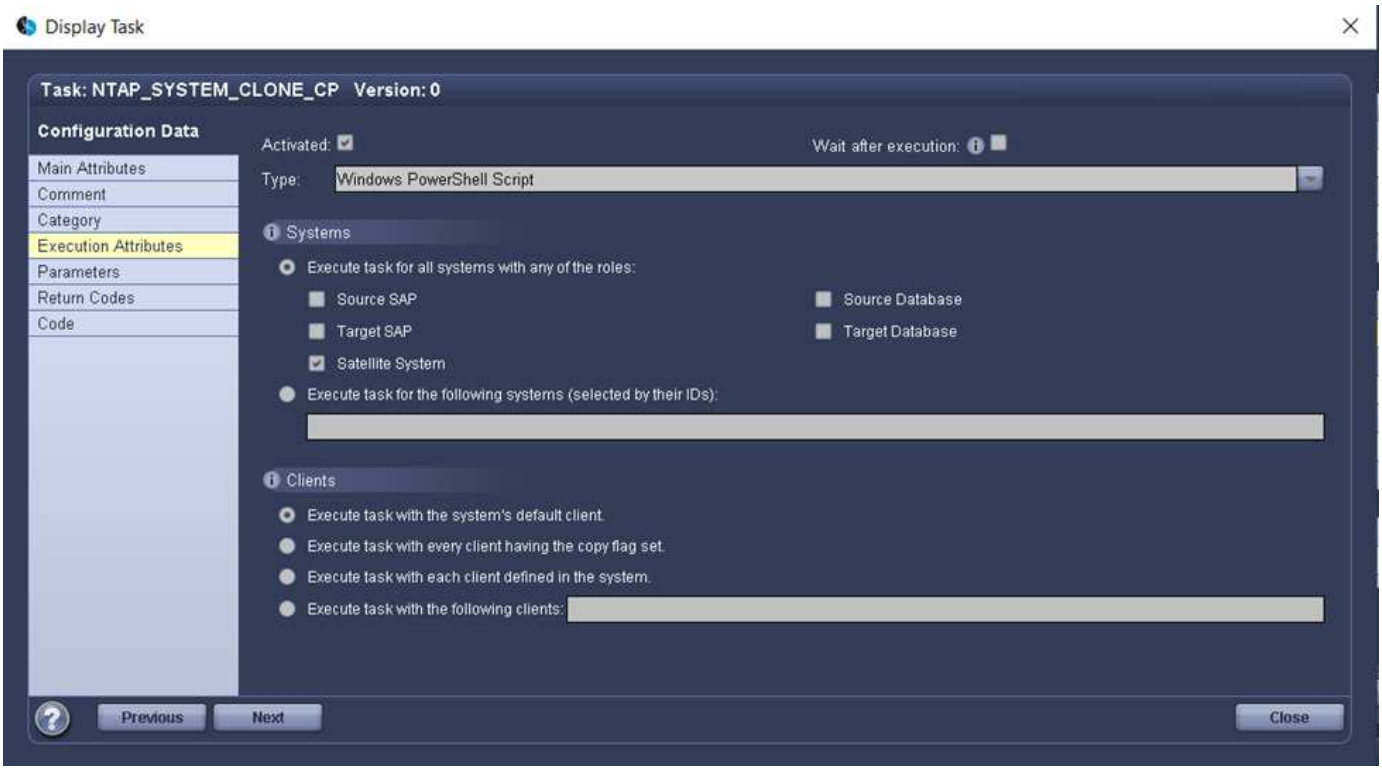
Central communication host

For integration between LSC and SnapCenter using a central communication host, the only adjustments that must be made are performed in the copy phase. The Snapshot copy and the clone are created using the SnapCenter agent on the central communication host. Therefore, all details about the newly created volumes are only available on the central communication host and not on the target database server. However, these details are needed on the target database server to mount the clone volume and to carry out the recovery. This is the reason why two additional tasks are needed in the copy phase. One task is executed on the central communication host and one task is executed on the target database server. These two tasks are shown in the image below.

- **NTAP_SYSTEM_CLONE_CP.** This task creates the Snapshot copy and the clone using a PowerShell script that executes the necessary SnapCenter functions on the central communication host. This task therefore runs on the LSC satellite, which in our instance is the LSC master that runs on Windows. This script collects all details about the clone and the newly created volumes and hands it over to the second task `NTAP_MNT_RECOVER_CP`, which runs on the LSC worker that runs on the target database server.
- **NTAP_MNT_RECOVER_CP.** This task stops the target SAP system and the SAP HANA database, unmounts the old volumes, and then mounts the newly created storage clone volumes based on the parameters that were passed through from the previous task `NTAP_SYSTEM_CLONE_CP`. The target SAP HANA database is then restored and recovered.

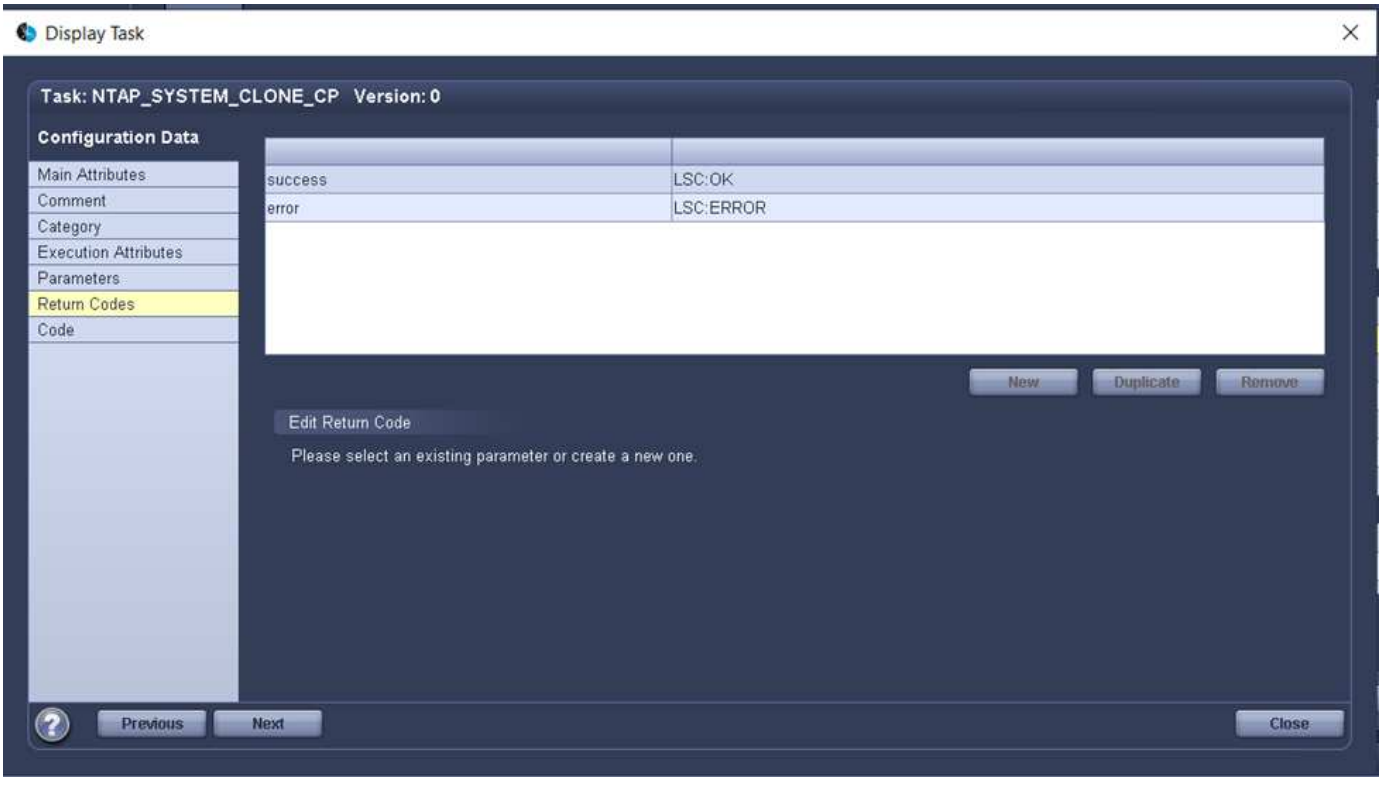
	copy	Copy Phase		phase
	copy 1	*NTAP_SYSTEM_CLONE*	*NetApp SnapShot and Clone*	*psh*
	copy 2	NTAP_SYSTEM_CLONE_CP	NetApp SnapShot and Clone	psh
	copy 3	NTAP_MNT_RECOVER_CP	Mount Volume and Recover HANA Database	cmd
	copy 4	*LPDBBCKP*	*Backup Source DB in Filesystem*	*lsh*
	copy 5	*LPDBCPYFLS*	*Copy DB Backup Files From Source to Target System*	*lsh*
	copy 6	*LTDBRESTORE*	*Restore DB Files*	*lsh*
	copy 7	*LTDBRESTORE_TENANT*	*Restore DB Files for Tenant Database*	*lsh*
	post	Post Phase		phase

The following image highlights the configuration of the task `NTAP_SYSTEM_CLONE_CP`. This is the Windows PowerShell script that is executed on the satellite system. In this instance, the satellite system is the SnapCenter Server with the installed LSC master.

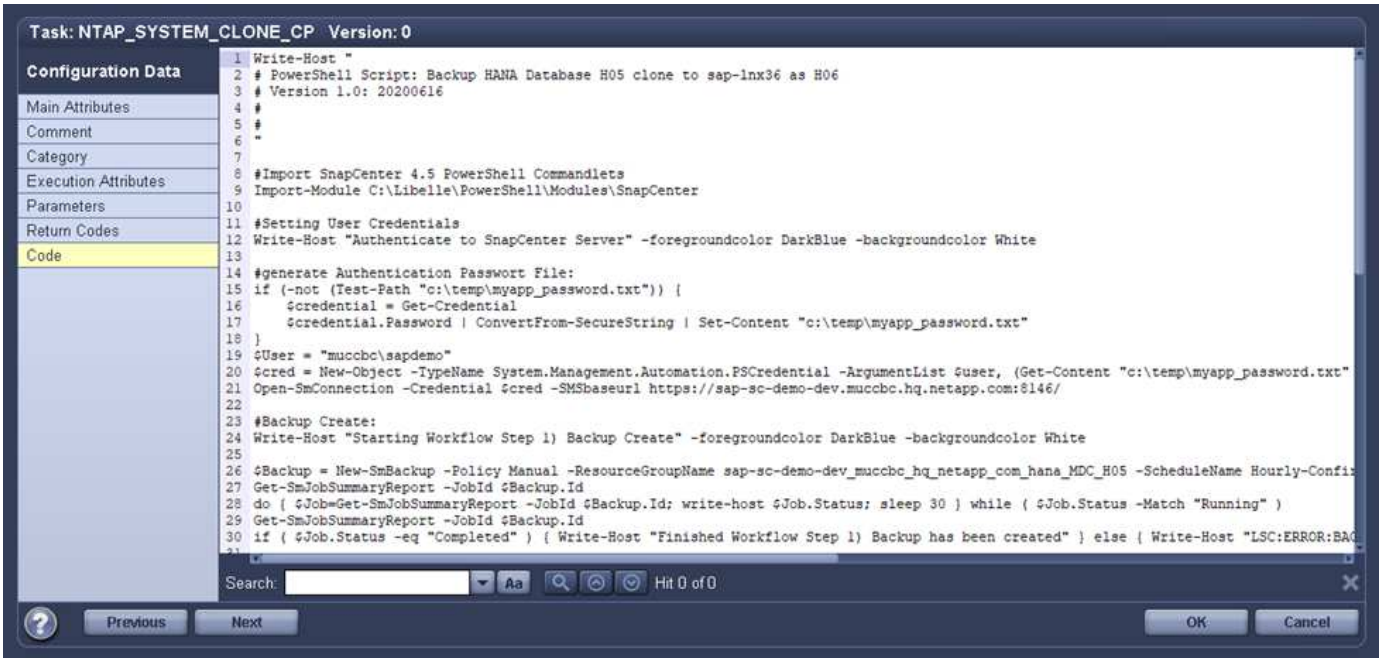


Because LSC must be aware of whether the Snapshot copy and cloning operation was successful, you must define at least two return code types: one return code for a successful execution of the script and the other for a failed execution of the script, as shown in the image below.

- `LSC:OK` must be written from the script to standard out if the execution was successful.
- `LSC:ERROR` must be written from the script to standard out if the execution failed.



The following image shows part of the PowerShell script that must run to execute a Snapshot copy and a clone using the SnapCenter agent on the central communication host. The script is not meant to be complete. Rather, the script is used to show how integration between LSC and SnapCenter can look and how easy it is to set it up.

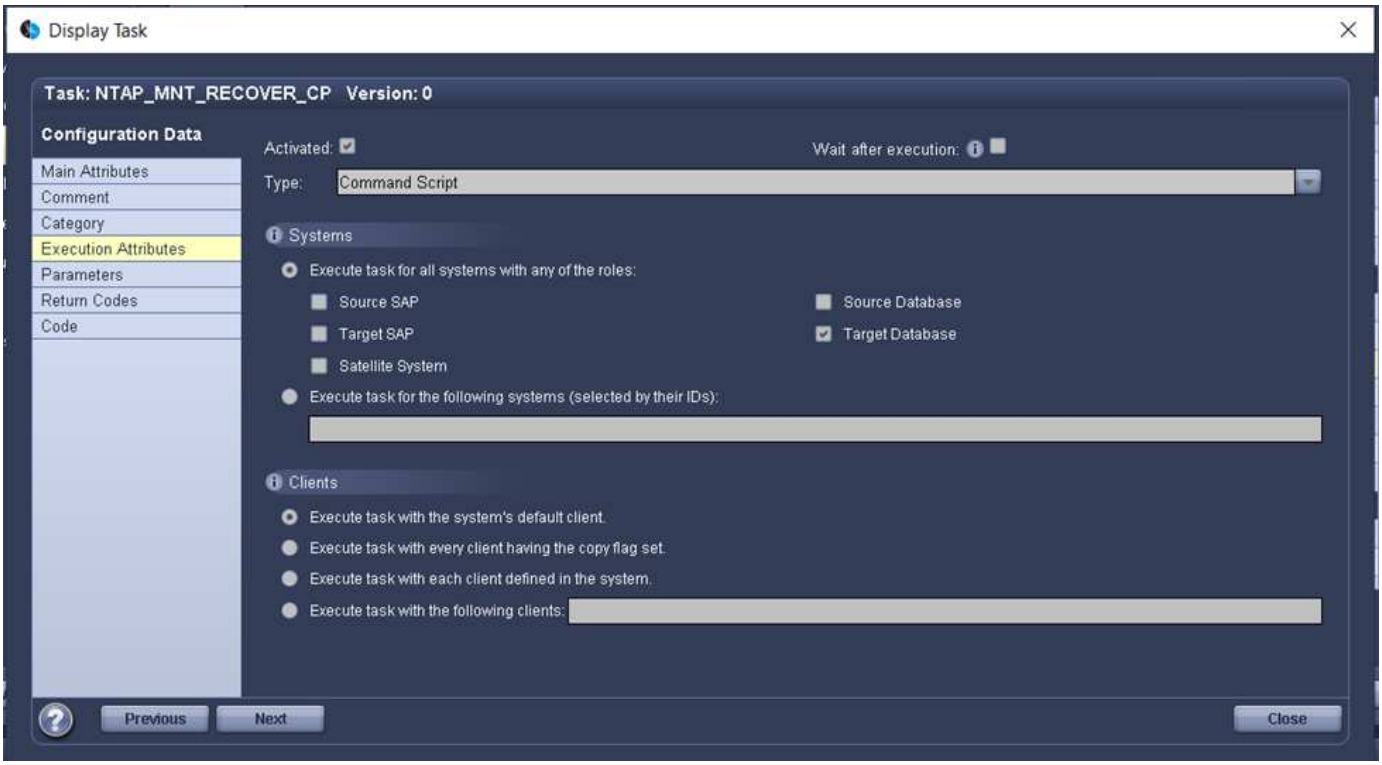


As previously mentioned, you must hand over the name of the clone volume to the next task `NTAP_MNT_RECOVER_CP` to mount the clone volume on the target server. The name of the clone volume, also known as the junction path, is stored in the variable `$JunctionPath`. The handover to a subsequent LSC task is achieved through a custom LSC variable.

```
echo $JunctionPath > $_task(current, custompath1)_$
```

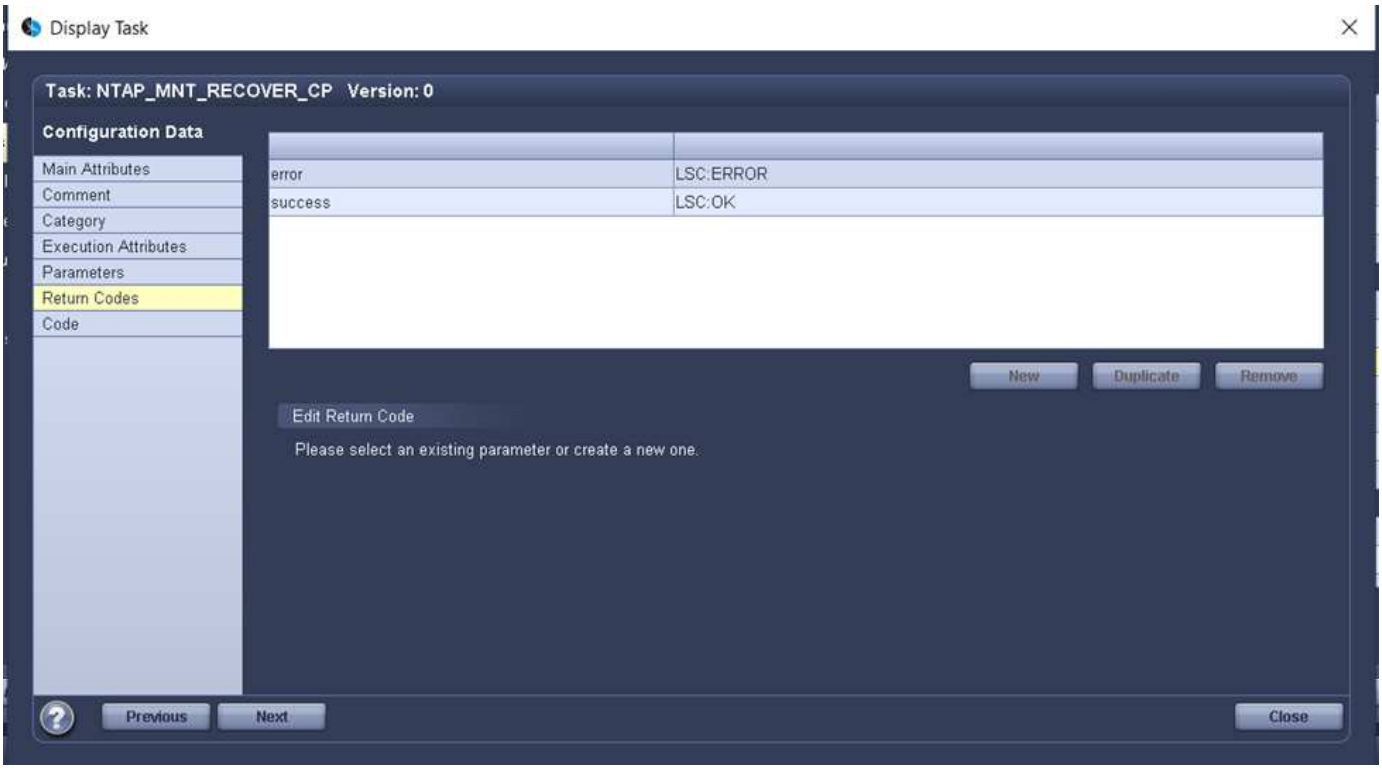
Because the script is executed on the LSC master (which is also a satellite system), the LSC master on the SnapCenter Server must run as a Windows user that has appropriate permissions to execute the backup and cloning operations in SnapCenter. To verify whether it has the appropriate permissions, the user should be able execute a Snapshot copy and clone in the SnapCenter GUI.

The following figure highlights the configuration of the task `NTAP_MNT_RECOVER_CP`. Because we want to execute a Linux Shell script, this is a command script executed on the target database system.



Because LSC must be made aware of mounting the clone volumes and whether restoring and recovering the target database was successful, we must define at least two return code types. One code is for a successful execution of the script, and one is for a failed execution of the script, as is shown in the following figure.

- `LSC:OK` must be written from the script to standard out if the execution was successful.
- `LSC:ERROR` must be written from the script to standard out if the execution failed.



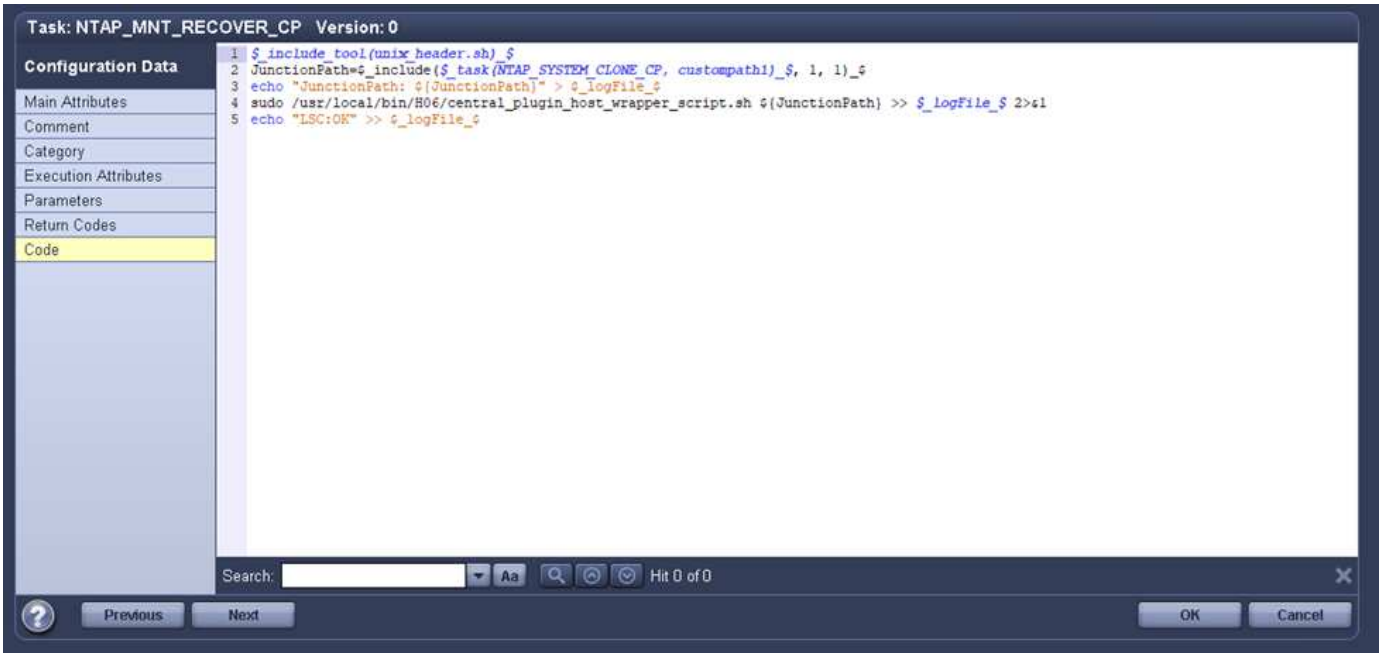
The following figure shows part of the Linux Shell script used to stop the target database, unmount the old

volume, mount the clone volume, and restore and recover the target database. In the previous task, the junction path was written into an LSC variable. The following command reads this LSC variable and stores the value in the `$JunctionPath` variable of the Linux Shell script.

```
JunctionPath=$_include($_task(NTAP_SYSTEM_CLONE_CP, custompath1)_$, 1,
1)_$
```

The LSC worker on the target system runs as `<sidaadm>`, but mount commands must be run as the root user. This is why you must create the `central_plugin_host_wrapper_script.sh`. The script `central_plugin_host_wrapper_script.sh` is called from the task `NTAP_MNT_RECOVERY_CP` using the `sudo` command. Using the `sudo` command, the script runs with UID 0 and we are able to carry out all subsequent steps, such as unmounting the old volumes, mounting the clone volumes, and restoring and recovering the target database. To enable script execution using `sudo`, the following line must be added in `/etc/sudoers`:

```
hn6adm ALL=(root)
NOPASSWD:/usr/local/bin/H06/central_plugin_host_wrapper_script.sh
```

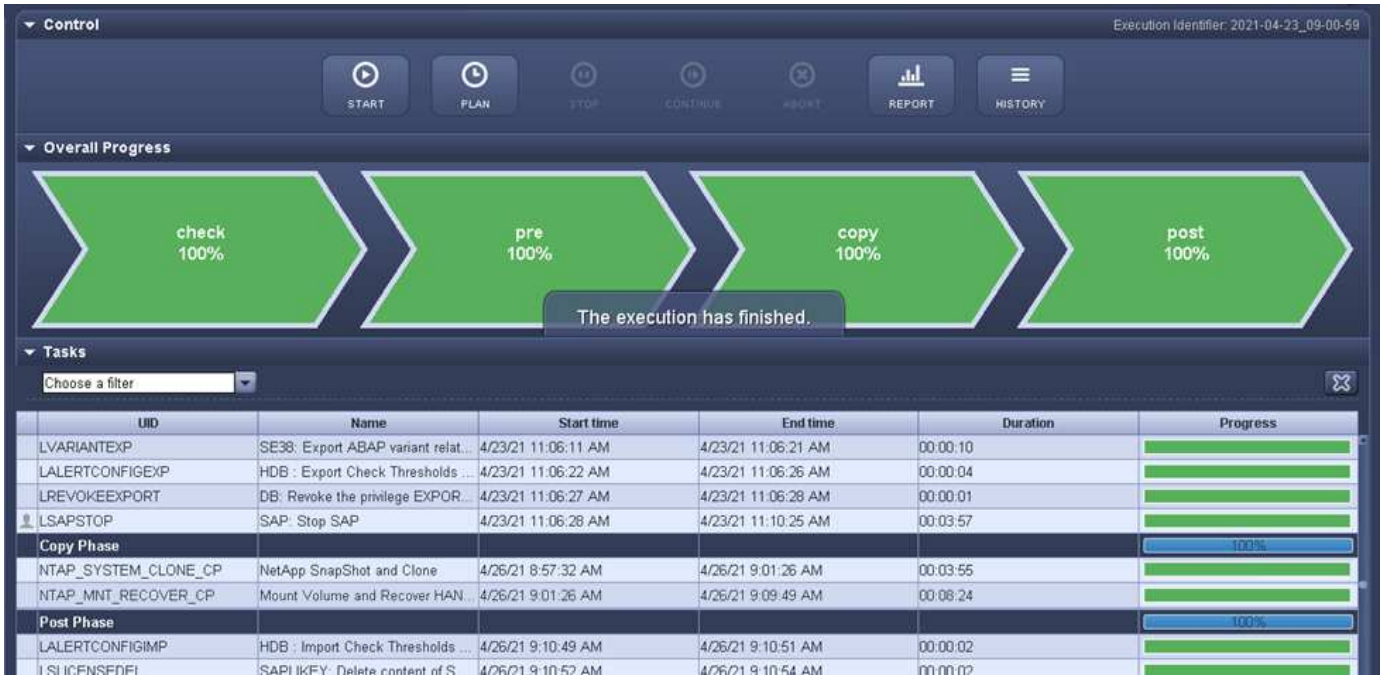


SAP HANA system refresh operation

Now that all necessary integration tasks between LSC and NetApp SnapCenter have been carried out, starting a fully automated SAP system refresh is a one-click task.

The following figure shows the task `NTAP``SYSTEM``CLONE` in a standard installation. As you can see, creating a Snapshot copy and a clone, mounting the clone volume on the target database server, and restoring and recovering the target database took approximately 14 minutes. Remarkably, with Snapshot and NetApp FlexClone technology, the duration of this task remains nearly the same, independent of the size of the source database.

The following figure shows the two tasks `NTAP_SYSTEM_CLONE_CP` and `NTAP_MNT_RECOVERY_CP` when using a central communication host. As you can see, creating a Snapshot copy, a clone, mounting the clone volume on the target database server, and restoring and recovering the target database took approximately 12 minutes. This is more or less the same time needed to carry out these steps when using a standard installation. Again, Snapshot and NetApp FlexClone technology enables the consistent, rapid completion of these tasks, independent of the size of the source database.



SAP HANA system refresh with LSC, AzAcSnap, and Azure NetApp Files

Using Azure NetApp Files for SAP HANA, Oracle, and DB2 on Azure provides customers with the advanced data management and data protection features of NetApp ONTAP with

the native Microsoft Azure NetApp Files service. AzAcSnap is the foundation for very fast SAP system refresh operations to create application-consistent NetApp Snapshot copies of SAP HANA and Oracle systems (DB2 is not currently supported by AzAcSnap).

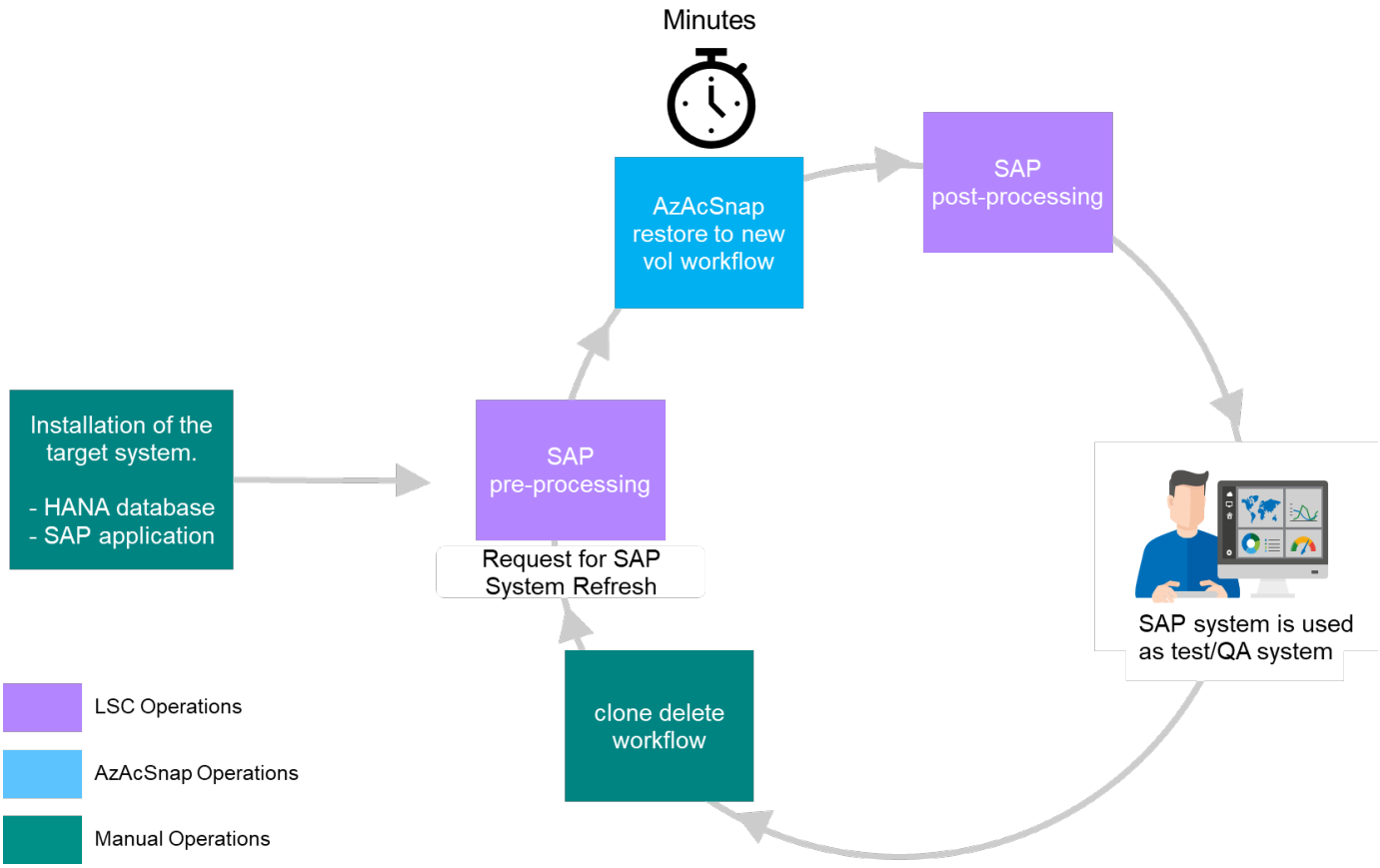
Snapshot copy backups, which are created either on-demand or on a regular basis as part of the backup strategy, can then be efficiently cloned to new volumes and used to quickly refresh target systems. AzAcSnap provides the workflows necessary to create backups and clone them to new volumes, while Libelle SystemCopy performs the pre- and post-processing steps necessary for a full end-to-end system refresh.

In this chapter, we describe an automated SAP system refresh using AzAcSnap and Libelle SystemCopy using SAP HANA as the underlying database. Because AzAcSnap is also available for Oracle, the same procedure can also be implemented using AzAcSnap for Oracle. Other databases might be supported by AzAcSnap in the future, which would then enable system copy operations for those databases with LSC and AzAcSnap.

The following figure shows a typical high-level workflow of an SAP system refresh lifecycle with AzAcSnap and LSC:

- A one-time, initial installation and preparation of the target system.
- SAP preprocessing operations performed by LSC.
- Restoring (or cloning) an existing Snapshot copy of the source system to the target system performed by AzAcSnap.
- SAP post-processing operations performed by LSC.

The system can then be used as a test or QA system. When a new system refresh is requested, the workflow restarts with step 2. Any remaining cloned volumes must be deleted manually.

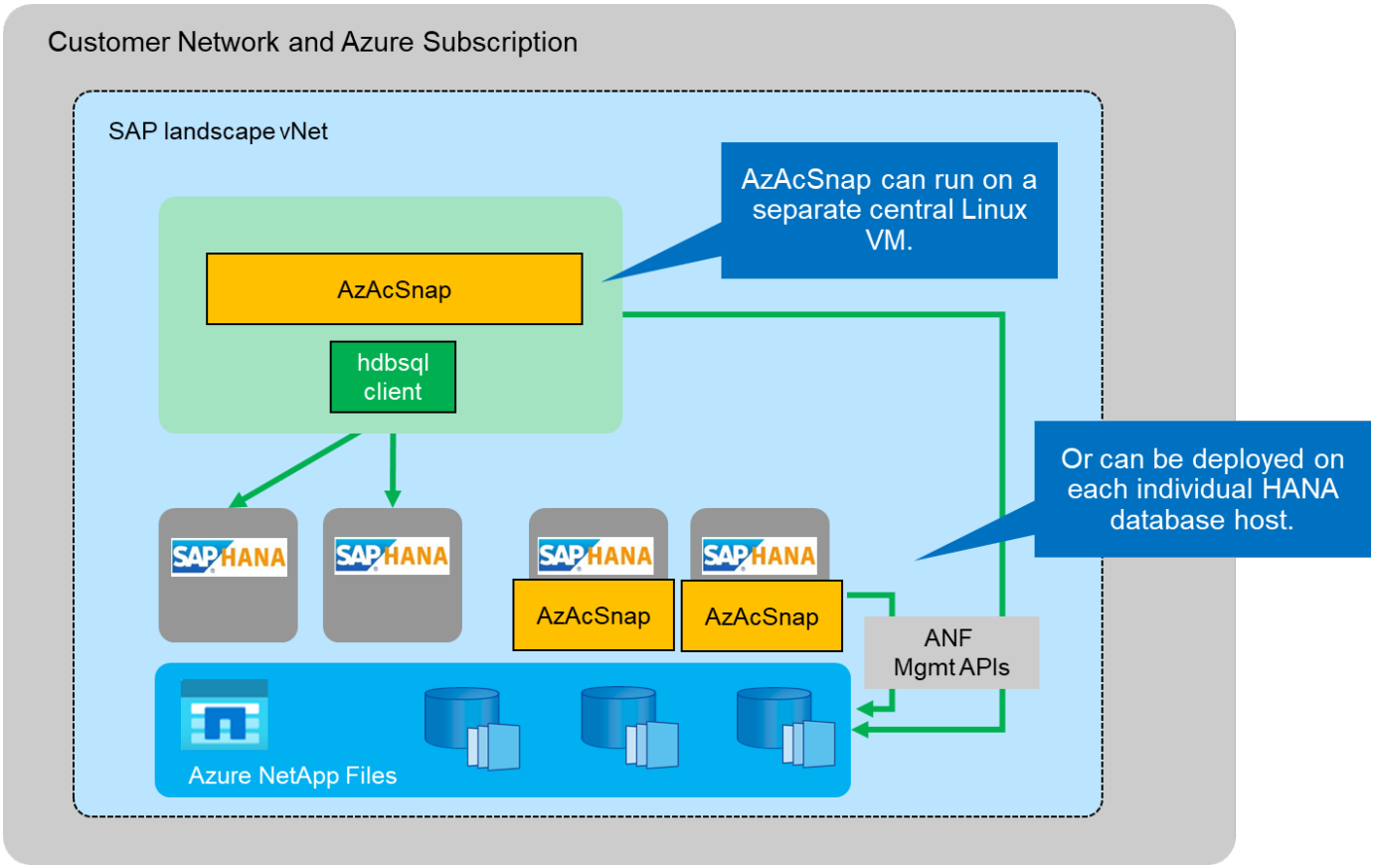


Prerequisites and limitations

The following prerequisites must be fulfilled.

AzAcSnap installed and configured for the source database

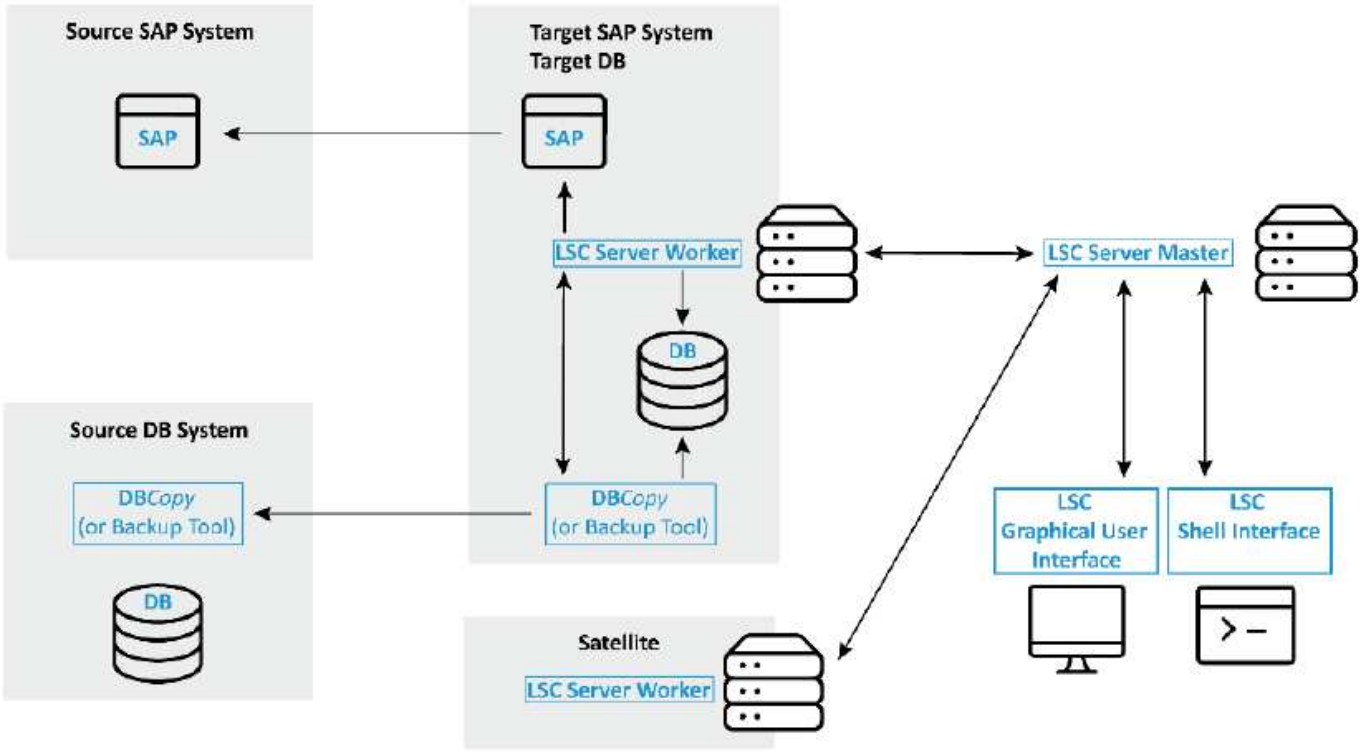
In general, there are two deployments options for AzAcSnap, as is shown in the following picture.



AzAcSnap can be installed and run on a central Linux VM for which all DB configuration files are stored centrally and AzAcSnap has access to all databases (through the hdbsql client) and the configured HANA userstore keys for all these databases. With a decentralized deployment, AzAcSnap is installed individually on each database host where typically only the DB configuration for the local database is stored. Both deployment options are supported for LSC integration. However, we followed a hybrid approach in the lab setup for this document. AzAcSnap was installed on a central NFS share along with all DB configuration files. This central installation share was mounted on all VMs under `/mnt/software/AZACSNAP/snapshot-tool`. The execution of the tool was then performed locally on the DB VMs.

Libelle SystemCopy installed and configured for source and target SAP system

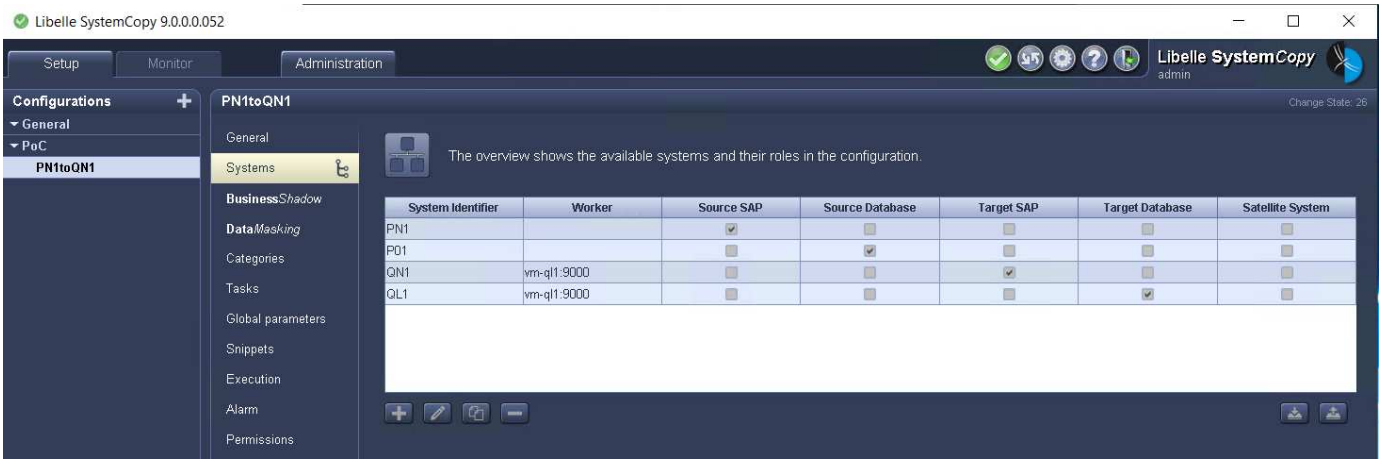
Libelle SystemCopy deployments consist of the following components:



- **LSC Master.** As the name suggests, this is the master component that controls the automatic workflow of a Libelle-based system copy.
- **LSC Worker.** An LSC worker usually runs on the target SAP system and executes the scripts required for the automated system copy.
- **LSC Satellite.** An LSC satellite runs on a third-party system on which further scripts must be executed. The LSC master can also fulfill the role of an LSC satellite system.

The Libelle SystemCopy (LSC) GUI must be installed on a suitable VM. In this lab setup, the LSC GUI was installed on a separate Windows VM, but it can also run on the DB host together with the LSC worker. The LSC worker must be installed at least on the VM of the target DB. Depending on your chosen AzAcSnap deployment option, additional LSC worker installations might be required. You must have an LSC worker installation on the VM where AzAcSnap is executed.

After LSC is installed, the basic configuration for the source and the target database must be performed according to the LSC guidelines. The following images shows the configuration of the lab environment for this document. See the next section for details about the source and the target SAP systems and databases.



You should also configure a suitable standard task list for the SAP systems. For more details about the installation and configuration of LSC, consult the LSC user manual that is part of the LSC installation package.

Known limitations

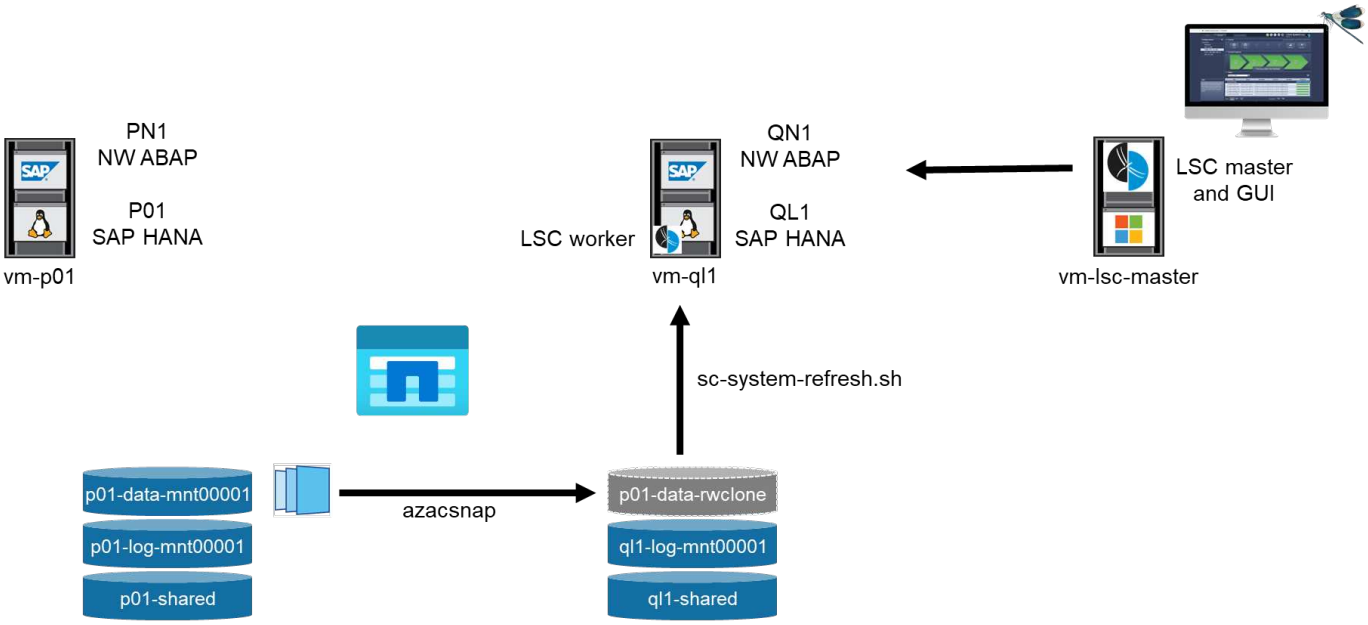
The AzAcSnap and LSC integration described here only works for SAP HANA single-host databases. SAP HANA multiple-host (or scale-out) deployments can also be supported, but such deployments require a few adjustments or enhancements to the LSC custom tasks for the copy phase and the underlaying scripts. Such enhancements are not covered in this document.

SAP system refresh integration always uses the latest successful Snapshot copy of the source system to perform the refresh of the target system. If you would like to use other older Snapshot copies, the corresponding logic in the ZAZACSNAPRESTORE custom task must be adjusted. This process is out of scope for this document.

Lab setup

The lab setup consists of a source SAP system and a target SAP system, both running on SAP HANA single-host databases.

The following picture shows the lab setup.



It contains the following systems, software versions, and Azure NetApp Files volumes:

- **P01.** SAP HANA 2.0 SP5 database. Source database, single host, single user tenant.
- **PN1.** SAP NetWeaver ABAP 7.51. Source SAP system.
- **vm-p01.** SLES 15 SP2 with AzAcSnap installed. Source VM hosting P01 and PN1.
- **QL1.** SAP HANA 2.0 SP5 database. System refresh target database, single host, single-user tenant.
- **QN1.** SAP NetWeaver ABAP 7.51. System refresh target SAP system.
- **vm-ql1.** SLES 15 SP2 with LSC worker installed. Target VM hosting QL1 and QN1.
- LSC master version 9.0.0.0.052.
- **vm- lsc-master.** Windows Server 2016. Hosts LSC master and LSC GUI.

- Azure NetApp Files volumes for data, log, and shared for P01 and QL1 mounted on the dedicated DB hosts.
- Central Azure NetApp Files volume for scripts, AzAcSnap installation, and configuration files mounted on all VMs.

Initial one-time preparation steps

Before the first SAP system refresh can be executed, you must integrate Azure NetApp Files Snapshot copy-and-cloning-based storage operations executed by AzAcSnap. You must also execute an auxiliary script for starting and stopping the database and mounting or unmounting the Azure NetApp Files volumes. All required tasks are performed as custom tasks in LSC as part of the copy phase. The following picture shows the custom tasks in the LSC task list.

Phase	UID	Name	Type
pre 77	LREVOKEEXPORT	DB: Revoke the privilege EXPO...	cmd
pre 78	*LJAVACONFEXP*	*JAVA: Backup java config files...*	*cmd*
pre 79	*LSTOPSLTJOBS*	*LTRC: Stop all replication jobs ...*	*lsh*
pre 80	*LSAPSTOP*	*SAP: Stop SAP*	*intv*
pre 81	LSTOPSAPSYSTEM	Stops all SAP instances (appli...	lsh
copy	**Copy Phase**		**phase**
copy 1	ZSCCOPYSHUTDOWN	Shutdown HANA DB	cmd
copy 2	ZSCCOPYUMOUNT	Unmount data volumes	cmd
copy 3	ZAZACSNAPRESTORE	Restore snapshot backup of so...	cmd
copy 4	ZSCCOPYMOUNT	Mount data volumes	cmd
copy 5	ZSCCOPYRECOVER	Recover target DB based on sn...	cmd
post	**Post Phase**		**phase**
post 1	*LCHNGHDBPWD*	*HDB : Restore the password fo...*	*cmd*
post 2	LHDBLICIMP	HANA DB License Import	lsh
post 3	LALERTCONFIGIMP	HDB : Import Check Threshold...	lsh

All five copy tasks are described here in more detail. In some of these tasks, a sample script `sc-system-refresh.sh` is used to further automate the required SAP HANA database recovery operation and the mount and unmount of the data volumes. The script uses an `LSC: success` message in the system output to indicate a successful execution to LSC. Details about custom tasks and available parameters can be found in the LSC user manual and the LSC developer guide. All tasks in this lab environment are executed on the target DB VM.

The sample script is provided as is and is not supported by NetApp. You can request the script by email to ng-sapcc@netapp.com.

Sc-system-refresh.sh configuration file

As mentioned before, an auxiliary script is used to start and stop the database, to mount and unmount the Azure NetApp Files volumes, and to recover the SAP HANA database from a Snapshot copy. The script `sc-system-refresh.sh` is stored on the central NFS share. The script requires a configuration file for each target database that must be stored in the same folder as the script itself. The configuration file must have the following name: `sc-system-refresh-<target DB SID>.cfg` (for example `sc-system-refresh-QL1.cfg` in this lab environment). The configuration file used here uses a fixed/hard-coded source DB SID. With a few changes, the script and the config file can be enhanced to take the source DB SID as an input parameter.

The following parameters must be adjusted according to the specific environment:

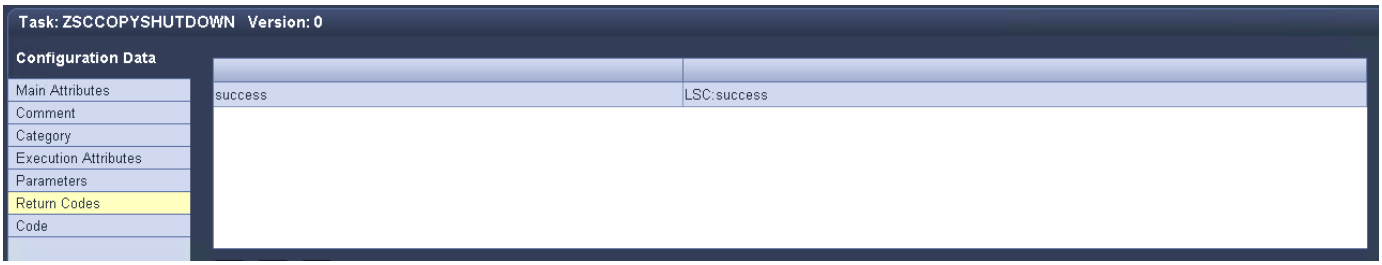
```
# hdbuserstore key, which should be used to connect to the target database
KEY=”QL1SYSTEM”
# single container or MDC
export P01_HANA_DATABASE_TYPE=MULTIPLE_CONTAINERS
# source tenant names { TENANT_SID [, TENANT_SID]* }
export P01_TENANT_DATABASE_NAMES=P01
# cloned vol mount path
export CLONED_VOLUMES_MOUNT_PATH=`tail -2
/mnt/software/AZACSNAP/snapshot_tool/logs/azacsnap-restore-azacsnap-
P01.log | grep -oe “[0-9]*\.[0-9]*\.[0-9]*\.[0-9]*:/.* “`
```

ZSCCOPYSHUTDOWN

This task stops the target SAP HANA database. The Code section of this task contains the following text:

```
$_include_tool(unix_header.sh)_$
sudo /mnt/software/scripts/sc-system-refresh/sc-system-refresh.sh shutdown
$_system(target_db, id)_$ > $_logfile_$
```

The script `sc-system-refresh.sh` takes two parameters, the `shutdown` command and the DB SID, to stop the SAP HANA database using sapcontrol. The system output is redirected to the standard LSC logfile. As mentioned before, an `LSC: success` message is used to indicate successful execution.



ZSCCOPYUMOUNT

This task unmounts the old Azure NetApp Files data volume from the target DB operating system (OS). The code section of this task contains the following text:

```
$_include_tool(unix_header.sh)_$
sudo /mnt/software/scripts/sc-system-refresh/sc-system-refresh.sh umount
$_system(target_db, id)_$ > $_logfile_$
```

The same scripts as in the previous task is used. The two parameters passed are the `umount` command and the DB SID.

ZAZACSNAPRESTORE

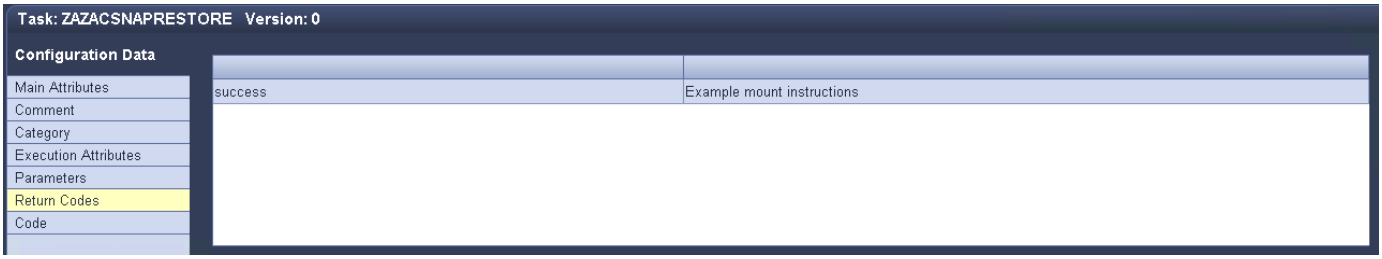
This task runs AzAcSnap to clone the latest successful Snapshot copy of the source database to a new volume for the target database. This operation is equivalent to a redirected restore of backup in traditional backup environments. However, the Snapshot copy and cloning functionality enables you to perform this task within seconds even for the largest databases, whereas, with traditional backups, this task could easily take several hours. The code section of this task contains the following text:

```
$_include_tool(unix_header.sh)_$
sudo /mnt/software/AZACSNAP/snapshot_tool/azacsnap -c restore --restore
snaptovol --hanasid $_system(source_db, id)_$
--configfile=/mnt/software/AZACSNAP/snapshot_tool/azacsnap
-$_system(source_db, id)_$.json > $_logfile_$
```

Full documentation for the AzAcSnap command line options for the `restore` command can be found in the Azure documentation here: Restore using Azure Application Consistent Snapshot tool. The call assumes that the json DB configuration file for the source DB can be found on the central NFS share with the following naming convention: `azacsnap-<source DB SID>. json`, (for example, `azacsnap-P01.json` in this lab environment).

> Because the output of the AzAcSnap command cannot be changed, the default `LSC: success` message cannot be used for this task. Therefore, the string `Example mount instructions` from the AzAcSnap output is used as a successful return code. In the 5.0 GA version of AzAcSnap, this output is only generated if the cloning process was successful.

The following figure shows the AzAcSnap restore to new volume success message.



ZSCCOPYMOUNT

This task mounts the new Azure NetApp Files data volume on the OS of the target DB. The code section of this task contains the following text:

```
$_include_tool(unix_header.sh)_$
sudo /mnt/software/scripts/sc-system-refresh/sc-system-refresh.sh mount
$_system(target_db, id)_$ > $_logfile_$
```

The sc-system-refresh.sh script is used again, passing the `mount` command and the target DB SID.

ZSCCOPYRECOVER

This task performs an SAP HANA database recovery of the system database and the tenant database based on the restored (cloned) Snapshot copy. The recovery option used here is to specific database backup, such as no additional logs, are applied for forward recovery. Therefore, the recovery time is very short (a few minutes at most). The runtime of this operation is determined by the startup of the SAP HANA database that happens automatically after the recovery process. To speed up the startup time, the throughput of the Azure NetApp Files data volume can be increased temporarily if needed as described in this Azure documentation: Dynamically increasing or decreasing volume quota. The code section of this task contains the following text:

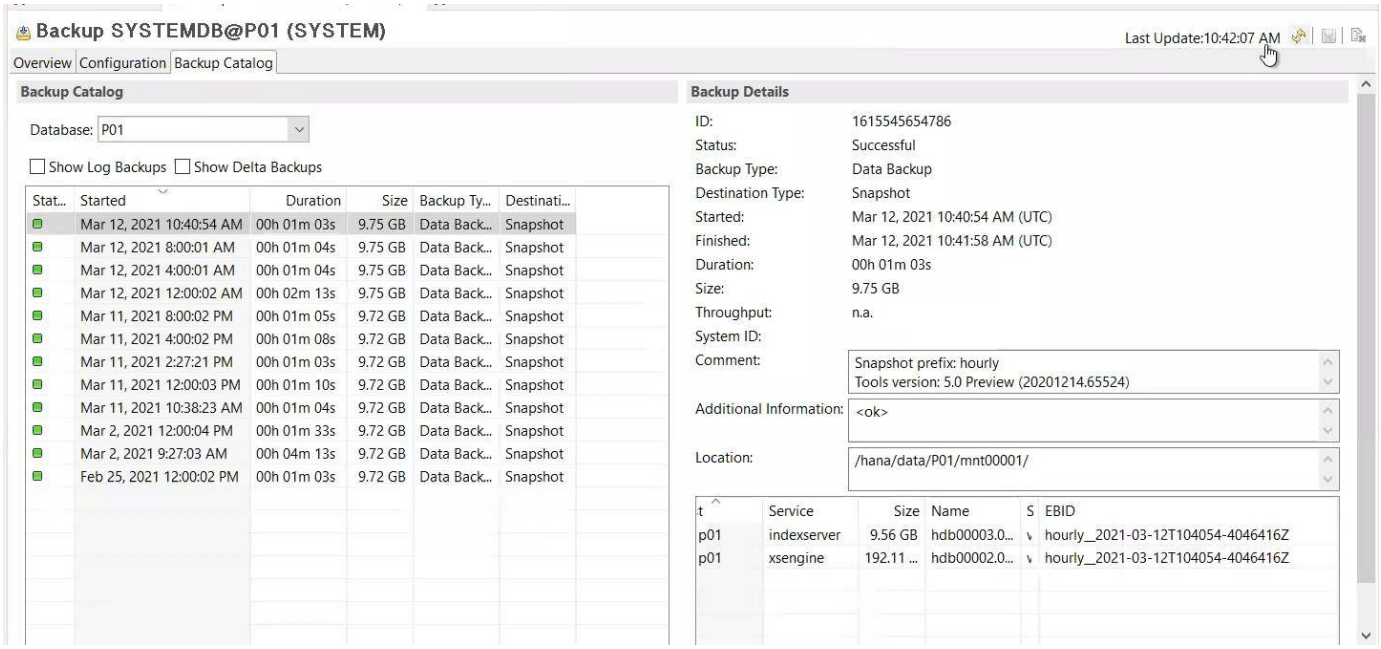
```
$_include_tool(unix_header.sh)_$
sudo /mnt/software/scripts/sc-system-refresh/sc-system-refresh.sh recover
$_system(target_db, id)_$ > $_logfile_$
```

This script is used again with the `recover` command and the target DB SID.

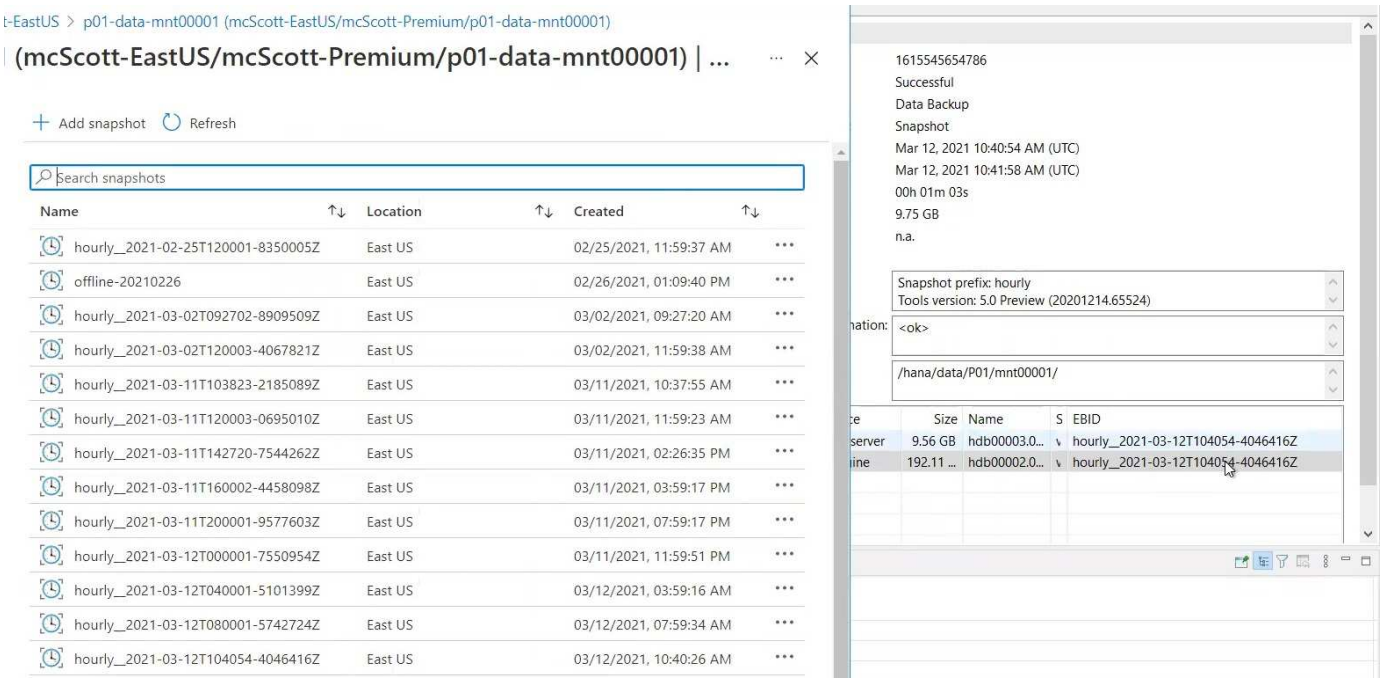
SAP HANA system refresh operation

In this section a sample refresh operation of lab systems shows the main steps of this workflow.

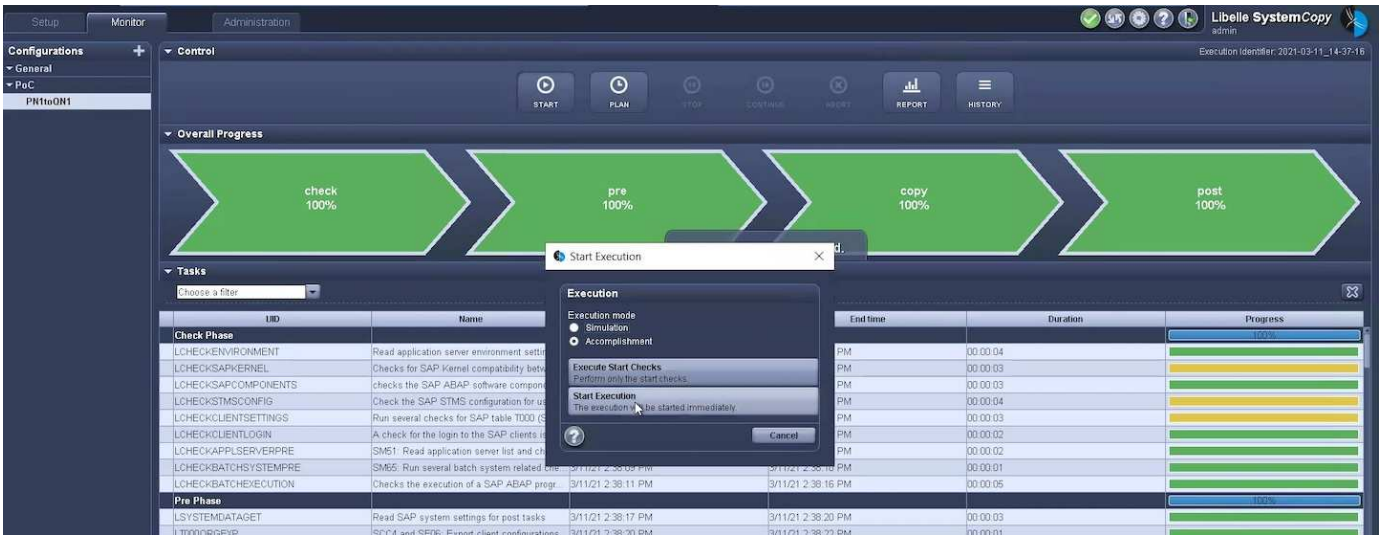
Regular and on-demand Snapshot copies have been created for the P01 source database as listed in the backup catalog.



For the refresh operation, the latest backup from March 12th was used. In the backup details section, the external backup ID (EBID) for this backup is listed. This is the Snapshot copy name of the corresponding Snapshot copy backup on the Azure NetApp Files data volume as shown in the following picture.

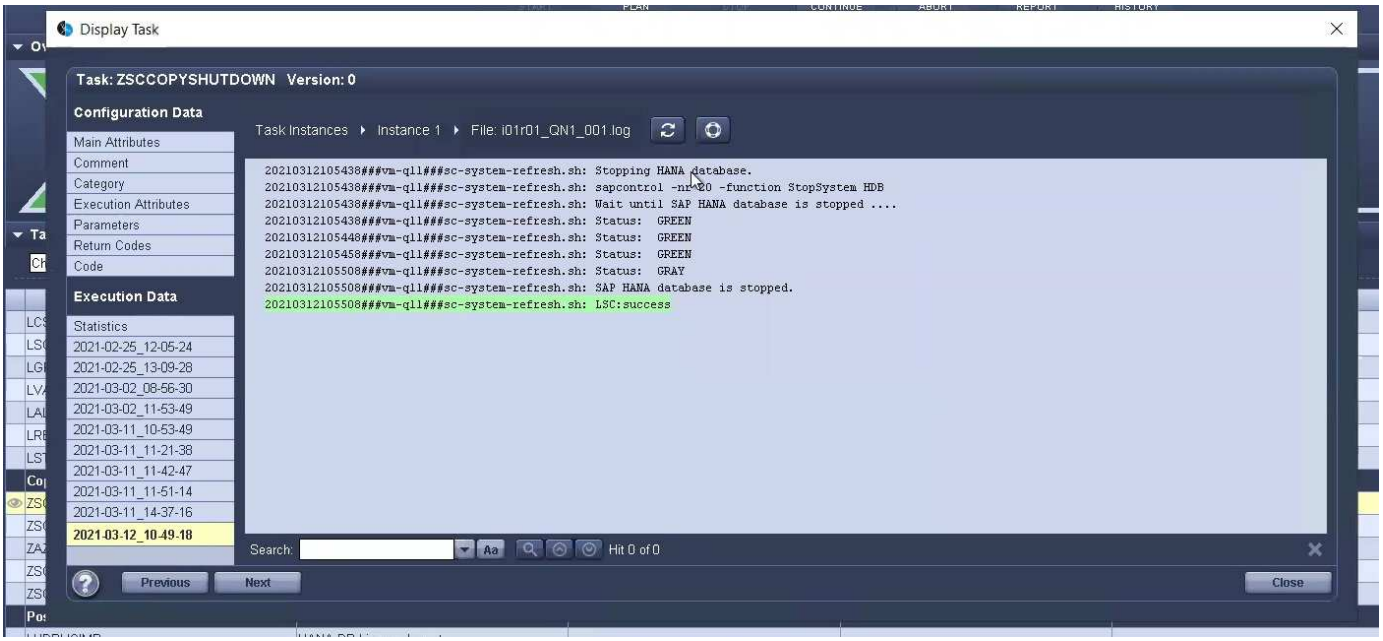


To start the refresh operation, select the correct configuration in the LSC GUI, and then click Start Execution.

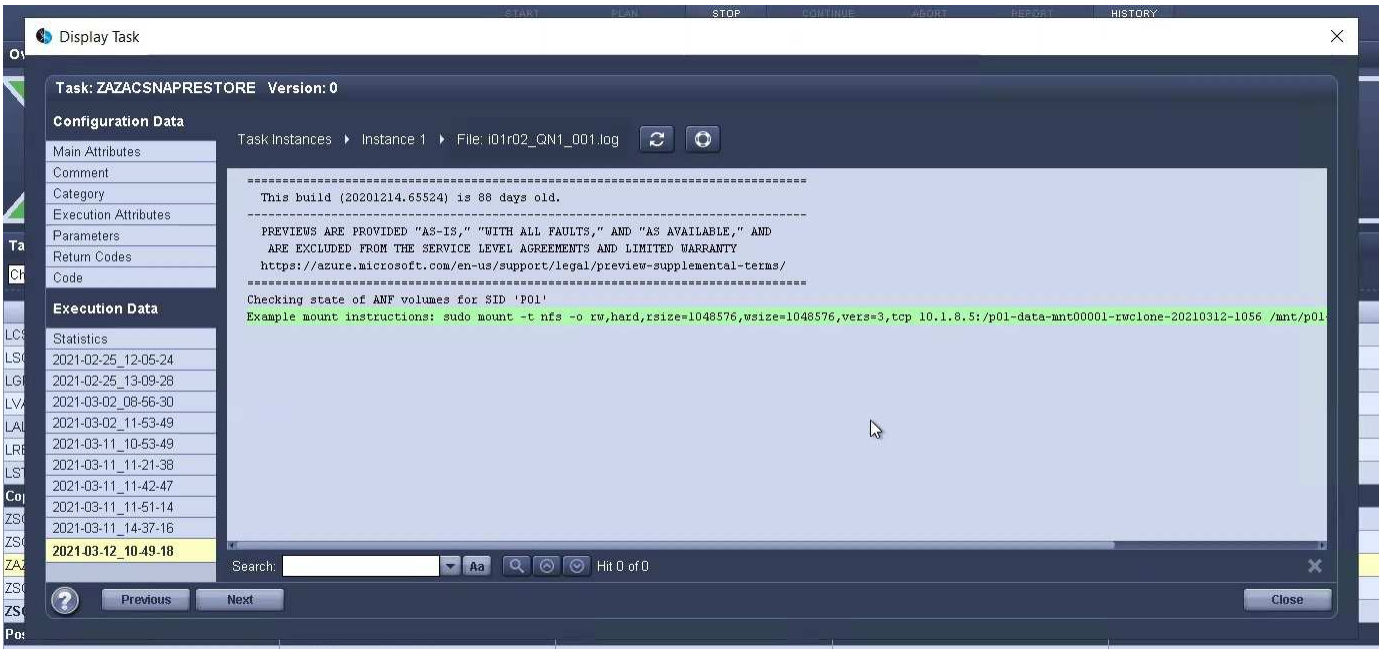


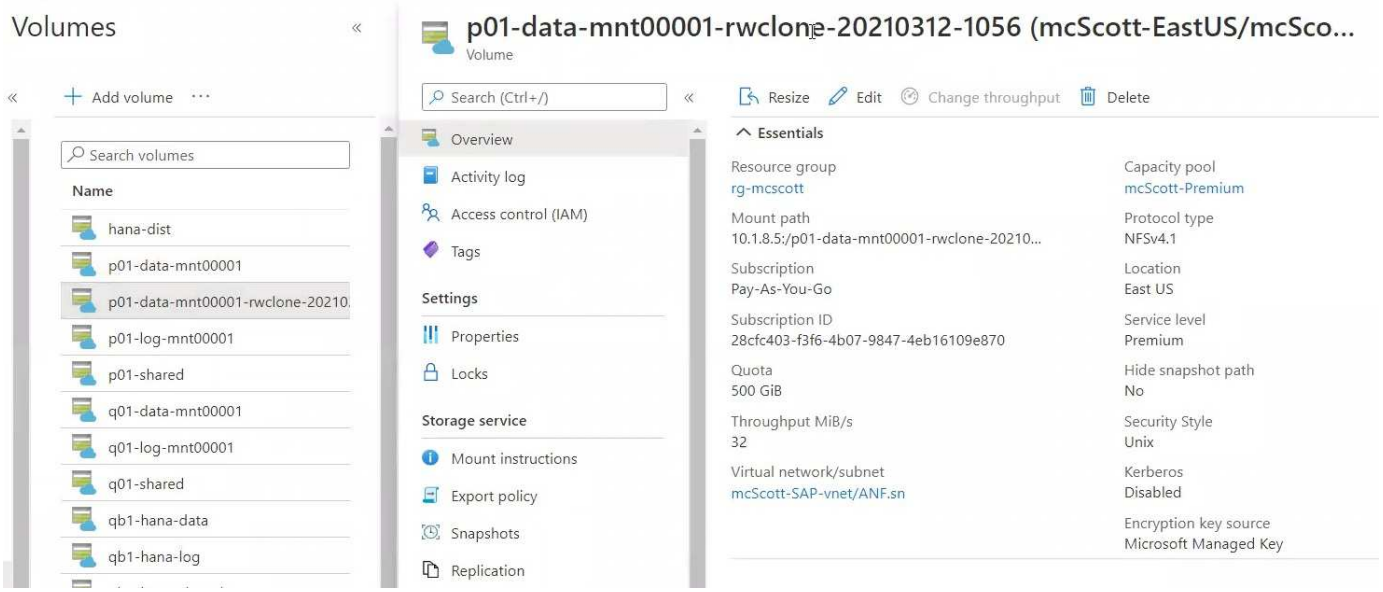
LSC starts to execute the tasks of the Check phase followed by the configured tasks of the Pre phase.

As the last step of the Pre phase, the target SAP system is stopped. In the following Copy phase, the steps described in the previous section are executed. First, the target SAP HANA database is stopped, and the old Azure NetApp Files volume is unmounted from the OS.

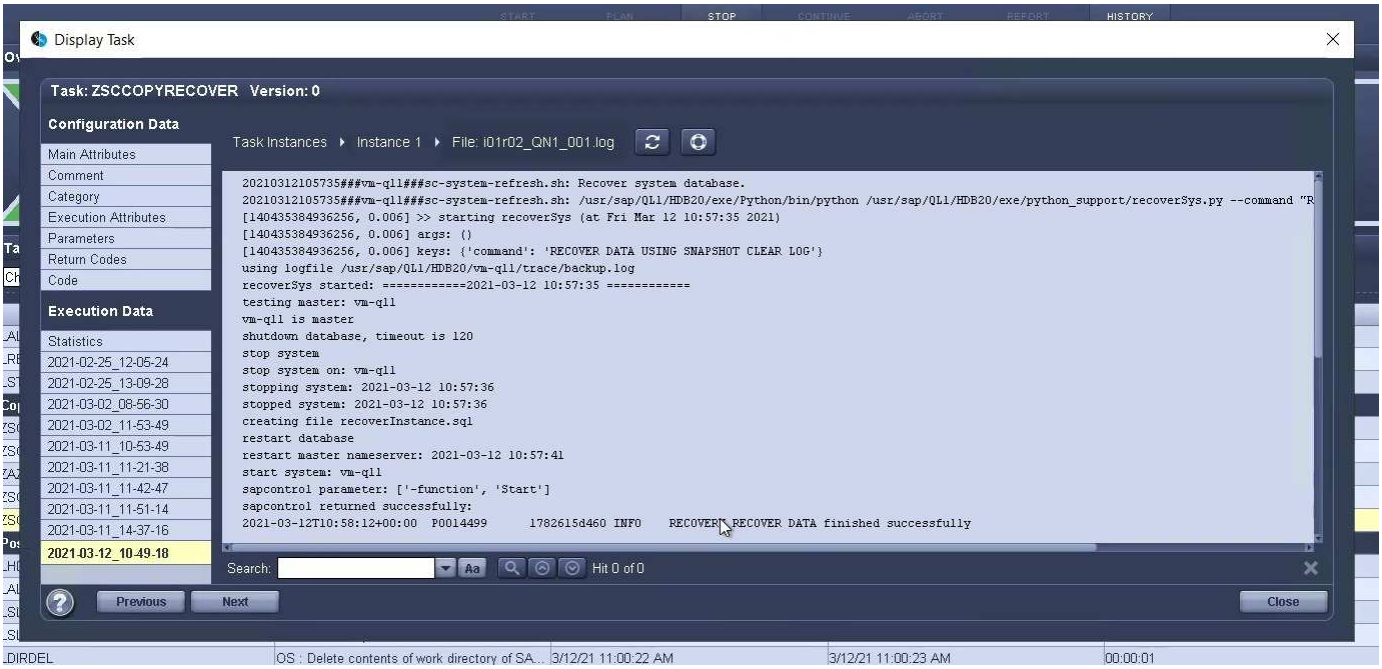


The ZAZACSNAPRESTORE task then creates a new volume as a clone from the existing Snapshot copy of the P01 system. The following two pictures show the logs of the task in the LSC GUI and the cloned Azure NetApp Files volume in the Azure portal.





This new volume is then mounted on the target DB host and the system database and the tenant database are recovered using the containing Snapshot copy. After successful recovery, the SAP HANA database is started automatically. This startup of the SAP HANA database occupies most of the time of the Copy phase. The remaining steps typically finish in a few seconds to a few minutes, regardless of the size of the database. The following picture shows how the system database is recovered using SAP- provided python recovery scripts.



After the Copy phase, LSC continues with all the defined steps of the Post phase. When the System Refresh process finishes completely, the target system is up and running again and fully usable. With this lab system, the total runtime for the SAP system refresh was roughly 25 minutes, of which the Copy phase consumed just under 5 minutes.

Where to find additional information and version history

To learn more about the information that is described in this document, review the following documents and/or websites:

- NetApp Product Documentation

  https://docs.netapp.com

Version history

Version	Date	Document Version History
Version 1.0	April 2022	Initial release.

Copyright information

Trademark information