Hewlett Packard
Enterprise

HPE Serviceguard Solutions for Microsoft SQL Server for Linux User Guide

Abstract

This guide provides instructions to integrate and manage Microsoft SQL Server on Linux in the HPE Serviceguard for Linux environment.

Part Number: P04492-001
Published: December 2017
Edition: 1

Notices

Acknowledgments

Contents

Serviceguard Solutions for Microsoft SQL Server on Linux overview

Serviceguard 12.20.00 introduces High Availability (HA) and Disaster Recovery Solution (DRS) for Microsoft SQL Server on Linux (SQL Server). The HPE Serviceguard Solutions for Microsoft SQL Server for Linux (SGMSSQL) manages the lifecycle orchestration of the database, health of the database, Operating System, software, and hardware.

SGMSSQL has inbuilt capability to automatically manage SQL Server deployment models as workloads in Serviceguard. The two SQL Server deployments supported are:

- Always ON Failover Cluster Instance (AOFI)
- Always ON Availability Groups (AOAI)

To use SGMSSQL, you must be familiar with Serviceguard concepts and commands, Linux operating system administration, and Microsoft SQL Server basics. For more information about the deployment modes of SQL Server see, **Microsoft documents**.

Workload for Microsoft SQL Server Always on Failover Instance

AOFI provides high availability that protects the application against any hardware, software, and infrastructure failures by failing over the SQL Server instance from one node to another. Serviceguard workload for AOFI requires that the system and user databases are present on LVM shared disk that is connected to all the nodes of the cluster.

SGMSSQL workload for AOFI provides the automation required for:

- Instance level protection for Microsoft SQL Server
- Full database health monitoring
- Automatic failure detection and failover
- Support for failback
- Failure detection of any component in the infrastructure, hardware, and software

Workload for Microsoft SQL Server Always on Availability groups

An availability group supports a fail over environment for a set of user databases, known as availability databases. An availability group has one fully read/write copy of the database which is on the primary replica and multiple secondary replicas. Serviceguard provides High Availability and Disaster Recovery protection for an availability group.

AOAI workloads support a minimum of two synchronous replicas and a maximum of three replicas with at least two synchronous replicas. SGMSSQL workload for AOAI, provides the automation required for:

- Microsoft SQL Server Availability Group (Database) level protection
- Support for Microsoft SQL Server Instance monitoring

- Automatic failure detection and fail over at the Availability Group level
- Support for failback
- Robust mechanism to promote a replica as primary that is most up-to-date in the availability group
- Automatic failure detection of any component in the infrastructure, hardware, and software
- Graceful handling of cyclic node failures
- Support for RPO based failover using `REQUIRED_SYNCHRONIZED_SECONDARIES_TO_COMMIT`

For more information about Microsoft SQL Server see, **Microsoft Documents**.

Prerequisites to install SGMSSQL

- Serviceguard for Linux must be installed on all the nodes in the cluster
- Install the ODBC driver on all the nodes
- Install the SQL Server application on all the nodes
- You must have one of the following Serviceguard licenses installed to work with different SQL Server workloads:
  - Serviceguard for Linux Advanced, Enterprise, or Instant-On license for MS SQL AOFI workloads
  - Serviceguard for Linux Enterprise or Instant-On license for MS SQL AOAI workloads

For more information about installing MS SQL application see, **Microsoft Documents**.

Installing, upgrading, and compatibility

Installing, uninstalling, and upgrading the SGMSSQL RPM

For information about installing, uninstalling, and upgrading, see the *HPE Serviceguard Enterprise Release Notes* available at **http://www.hpe.com/info/linux-serviceguard-docs**.

For information about compatibility of Serviceguard and toolkit versions, see the HPE Serviceguard Toolkit Compatibility Matrix available at **http://www.hpe.com/info/linux-serviceguard-docs**.

SQL Server credentials management

The first step to deploy any SGMSSQL workloads is to set up the credentials file. This file stores the SQL Server login credentials with administration privileges.

Ensure that the following conditions are met:

- The SQL Server username and password is same across all cluster nodes
- This file is available on the cluster nodes hosting the AOFI or AOAI workload

---

NOTE:

It is recommended to configure the workload using easy deployment from Serviceguard Manager. When you deploy it from the Serviceguard Manager, the credentials file is set up automatically on all the cluster nodes.

Creating the SQL Server credentials file

Procedure

Create a login credentials file with the SQL Server user name in the first line and the password in the second line. Refer the information provided to create the file.
Details of the credential file

```
Location of the file :/var/opt/mssql/secrets/
File name: .mssql_credentials (Note : This is a hidden file)
File permission: 600 ( read-write permission for root user )
```

Example steps to create the file

```
sudo touch /var/opt/mssql/secrets/.mssql_credentials
sudo echo 'loginName' >> /var/opt/mssql/secrets/.mssql_credentials
sudo echo 'loginPassword' >> /var/opt/mssql/secrets/.mssql_credentials
sudo chmod 600 /var/opt/mssql/secrets/.mssql_credentials
```

Modifying the password

You can modify the SQL Server password only when the workloads are halted and not running. Updated credentials must be made available to all the nodes where the workloads will run.

Microsoft SQL Server workload deployment

Before you deploy any of the SGMSSQL workloads, ensure that the credential management setup is complete. SGMSSQL uses the credentials to access information from the SQL Server required to monitor AOFI or AOAI deployments. This setup is required for the workload to access the SQL Server.

More information

SQL Server credentials management on page 7

About AlwaysOn Failover Instance solution

SQL Server AOFI provides availability for the entire installation of SQL Server, known as an instance. The workload provides fail over protection for the instance. This entire instance will fail over to the adoptive node if software or hardware fails.

An AOFI workload consists of a single failover package. This package monitors the SQL Server processes and the server database health. During a fail over, the package halts all the instance on the failed node and starts it on the adoptive node.

NOTE:

You cannot create more than one AOFI workload in a cluster.

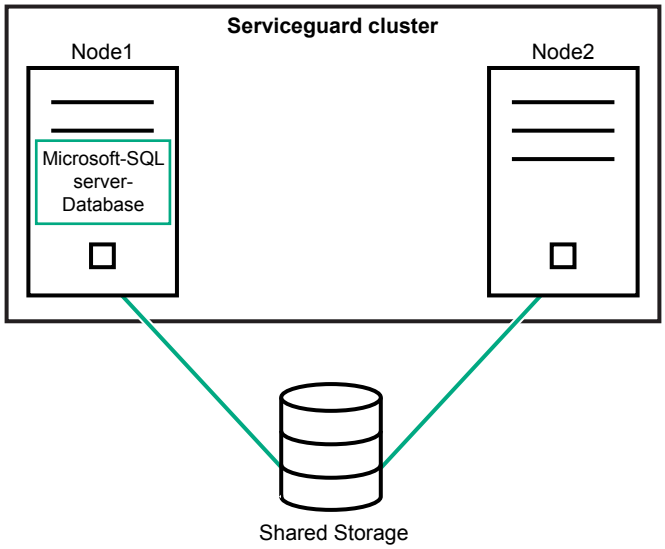


Figure 1: AOFI workload with shared storage

The instance is running on Node 1 and is monitored by Serviceguard fail over package *Microsoft_SQL_Server_Database*. The instance data resides on a shared disk and is accessed from Node 1. When Serviceguard detects that the instance can no longer run on Node 1 due to any failure (DB failure, node failure, site failure, or others), a fail over is initiated. This results in the package fail over to the adaptive node (Node 2).

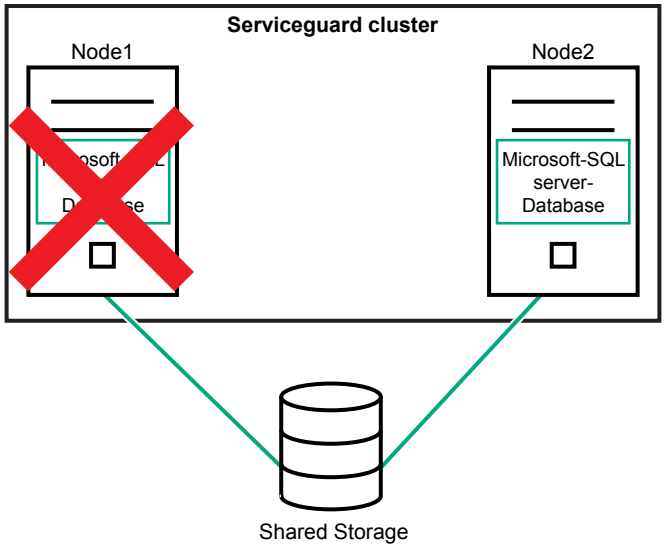


Figure 2: Workload fail over

Now the failover package *Microsoft_SQL_Server_Database* starts and ensures that the instance is running on Node 2. And the instance data on the shared storage is accessed from Node 2.

Requirements to configure AOFI workload

- All the fail over cluster information (FCI) must reside on a shared disk
- The Serviceguard node must be able to access this shared disk

Deploying the AOFI workload

You can deploy the AOFI workload either from the Serviceguard Manager or manually from the CLI. Serviceguard Manager provides for easy deployment of the AOFI workload. It is recommended that you use Serviceguard Manager for the AOFI deployment. For more information about deploying workload from Serviceguard Manager see, the Serviceguard Manager online help.

Complete the steps to manually deploy the workload from CLI.

Procedure

1. Run the `cmmakepkg` command to create the package configuration template for the AOFI workload package.
   `# cmmakepkg -m sgemssql/mssqldbinstance <mssql_aofi_pkg.conf>`

   A modular style package configuration file is created with all the required attributes for the AOFI workload.

NOTE:

The following conditions are applicable for the MSSQL AOFI deployment to support the MetroCluster and Extended distance cluster.

Include the appropriate Metrocluster or Extended Distance Clusters (XDC) modules when creating AOFI workload package as required.

---

2. Provide the values for the following attributes in the <*mssql_aofi_pkg.conf*> package configuration file:

Table 1: Variables in mssql_pkg.conf file

Variable Name	Description
`sgemssql/mssqldbinstance/MSSQL_PORT`	This attribute specifies the Port Number on which Microsoft SQL Server on Linux listens for client connections. This port number will be monitored by SGMSSQL. Default value is 1433. Online modification of this attribute is not supported.
`sgemssql/mssqldbinstance/MONITOR_INTERVAL`	This attribute specifies time interval, in seconds. This is the Microsoft SQL Server monitor service waits between checks to make sure that Microsoft SQL Server is running. Default value is 10 seconds. Online modification of this attribute is supported.
`sgemssql/mssqldbinstance/MSSQL_HOME`	This is the base directory where Microsoft SQL Server on Linux is installed. Default value is `C:\data`. This parameter is added for future use. Online modification of this attribute is not supported.

Table Continued

sgemssql/mssqldbinstance/instance_name	This attribute specifies the name of database that will be managed by this package. Online modification of this attribute is supported. When the workload package is up and running (online modification), you can only add a database to the package, but you cannot remove a database already configured in the package. When a new database is added as a part of online or offline addition, sgemssql/mssqldbinstance/critical attribute also has to be specified for each newly added database.
sgemssql/mssqldbinstance/critical	This attribute specifies the criticality of the corresponding database name which is specified in the sgemssql/mssqldbinstance/instance_name attribute. If the value is set to true, then the database is marked as critical. If the value is set to false, then the database is marked as non-critical. When a database marked as critical fails, then the entire instance will fail over to the adoptive node. If none of the databases are marked as critical, then an instance fail over is initiated only when all the databases fail. Default value is true. Online modification of this attribute is supported.

3. Add the shared storage information on which the instance data and database files resides, to the package.

   For example if the instance data resides on a volume group named vg-mssql, add the following attribute.

   ```
   vg vg-mssql
   ```

4. Add file system details, on which the instance data resides, to the package.

   Example:

   ```
   fs_name                 /dev/vg-mssql/lvol0
   fs_server                       ""
   fs_directory            /mssql-data
   fs_type                 ext4
   fs_mount_opt                    ""
   fs_umount_opt                   ""
   fs_fsck_opt                     ""
   ```

5. Optionally edit the attributes of the pre-defined services.

   NOTE:

   Do not change or edit the service_cmd attribute.

Table 2: Microsoft_SQL_Server_Monitor service

Attribute	Value
`service_name`	Microsoft_SQL_Server_Monitor
`service_cmd`	`"$SGCONF/scripts/sgemssql/hamssqldb.sh monitor"`
`service_restart`	none
`service_fail_fast_enabled`	no
`service_halt_timeout`	0
`service_halt_on_maintenance`	no

Table 3: Microsoft_SQL_Server_Health service

Attribute	Value
`service_name`	Microsoft_SQL_Server_Health
`service_cmd`	`"$SGCONF/scripts/sgemssql/hamssqldb.sh health"`
`service_restart`	none
`service_fail_fast_enabled`	no
`service_halt_timeout`	100
`service_halt_on_maintenance`	no

Table 4: Storage_Monitoring_MSSQL_DB service

Attribute	Value
`service_name`	Storage_Monitoring_MSSQL_DB
`service_cmd`	"$SGSBIN/cmresserviced <vg1> <vg2> " **NOTE:** The shared disk details provided in step 3 must be provided in this attribute. For example, if volume group named vg-mssql is provided in step 3, the same volume group name must be provided in the service_cmd of this service as follows: `"$SGSBIN/cmresserviced vg-mssql"`
`service_restar`	none

Table Continued

service_fail_fast_enabled	yes **NOTE:** If the service_fail_fast_enabled attribute is set to yes and if the service fails on the node on which this package is running, then the node will reboot.
service_halt_timeout	0
service_halt_on_maintenance	yes

Table 5: Microsoft_SQL_Server_DB_Monitor service

Attribute	Value
service_name	Microsoft_SQL_Server_DB_Monitor
service_cmd	"$SGCONF/scripts/sgemssql/hamssqldb.sh instance_monitor"
service_restart	none
service_fail_fast_enabled	no
service_halt_timeout	100
service_halt_on_maintenance	yes

Table 6: Microsoft_SQL_Server_Port_Monitor service

Attribute	Value
service_name	Microsoft_SQL_Server_Port_Monitor
service_cmd	"$SGCONF/scripts/sgemssql/hamssqldb.sh port_monitor"
service_restart	none
service_fail_fast_enabled	no
service_halt_timeout	0
service_halt_on_maintenance	yes

6. Verify the validity of the AOFI toolkit package configuration file using the `cmcheckconf` command.
   `cmcheckconf -P <mssql_aofi_pkg.conf>`
7. Run the `cmapplyconf` command to add the package to the Serviceguard environment, if the `cmcheckconf` command does not report any errors.
   `cmapplyconf -P <mssql_aofi_pkg.conf>`

   The failover package is created in the cluster with the name provided in the package configuration file.
8. Run the `cmmodpkg` command to enable the package switching.

This enables the AOFI toolkit package to fail over to an adoptive node in the event of a failure.

```
cmmodpkg -e -n node1 -n node2 <AOFI_pkg_name>
cmmodpkg -e <AOFI_pkg_name>
```

Here <AOFI_pkg_name> is the package name provided in the package configuration file `mssql_aofi_pkg.conf`.

About Always on Availability groups solution

AOAI with Serviceguard provide HA and DR protection for an availability group that comprises of groups of databases. An availability group supports a failover environment for a set of user databases, known as availability databases. AOAI provide database-level protection by sending each transaction of a database to another instance. This instance is known as a replica and contains a copy of that database.

The replicas can be primary or secondary replicas. The primary replica enable the database for read/write access to the clients and also sends the transaction log records for each primary database to every secondary replica.

The secondary replica is the backup copy of each availability group. The secondary replicas are configured as the failover targets for the primary replicas.

AOAI workload

Serviceguard for AOAI workload provides HA and DR protection for an availability group. An AOAI workload consists of three packages.

SQL Server Instance Package (SVR-MNP)

This package is a Serviceguard multinode package and monitors the SQL Server processes and the server instance health. You can create only one SVR-MNP package for the entire cluster.

Always on Availability Instance Package (AOAI-MNP)

This package is a Serviceguard multinode package and it monitors the availability group specified in the package configuration. One AOAI-MNP package is required for each availability group.

Write Intent Package (AOAI-WI)

This package is a Serviceguard failover package configured for an availability group. This package runs on the Serviceguard node which is the current primary replica. This package also signifies which Serviceguard node has Write access to the availability databases that are configured in the availability group.

All the three packages together monitor the MS SQL availability group and provide HA and DR protection for a single availability group. A new workload has to be configured for each availability group.

Rules and restrictions to configure AOAI workloads

- An AOAI workload is supported on two node and three node deployments.
- An AOAI workload is supported for write scale.
- User databases that are part of the availability group (availability databases), must reside on local LVM volumes.

Deploying the AOAI workload

You can deploy the AOAI workload either from the Serviceguard Manager or manually from the CLI. Serviceguard Manager provides for easy deployment of the AOAI workload. It is recommended that you

use Serviceguard Manager for the AOAI deployment. For more information about deploying workload from Serviceguard Manager see, the Serviceguard Manager online help.

To deploy the AOAI workload from the CLI all the three AOAI packages must be configured manually.

Procedure

1. **Configuring the Microsoft SQL Server Instance Package**.
2. **Configuring the Availability Instance Package**.
3. **Configuring the Write Intent Package**.

Configuring the Microsoft SQL Server Instance Package

After installing the Microsoft SQL application on all the nodes, install SGMSSQL and configure the server instance package. This is a multinode package. This package monitors the processes, listening port, and the health of the SQL Server. For more information about installing and configuring MS SQL server see, **Microsoft Documents**.

Procedure

1. Run the `cmmakepkg` command to create the package configuration template for SGMSSQL.
   `# cmmakepkg –m sgemssql/mssqlserver <mssql_srv_pkg.conf>`

   The multinode package configuration file is created with all the required attributes for SGMSSQL package.
2. Enter the values for the following attributes in the package configuration file:

Table 7: Variables in file <mssql_srv_pkg.conf>

Variable Name	Description
`sgemssql/mssqlserver/MSSQL_PORT`	This attribute specifies the Port Number on which Microsoft SQL Server on Linux runs. This port number will be monitored by Serviceguard.
`sgemssql/mssqlserver/MONITOR_INTERVAL`	This attribute specifies time interval, in seconds. This is the Microsoft SQL Server monitor service which will wait between checks to make sure that Microsoft SQL Server is running. Default value is 10 seconds. Online modification of this attribute is supported.

3. Add the volume group and file systems information.

   Sample configuration if the volume group used is `vg_ag1` and has a logical volume `/dev/vg_ag1/lvol0` mounted on `/var/opt/mssql/data/salesdb`:

```
vg                  vg_ag1
fs_name             /dev/vg_ag1/lvol0
fs_server           ""
fs_directory        /var/opt/mssql/data/salesdb
fs_type             ext4
```

```
fs_mount_opt            ""
fs_umount_opt           ""
fs_fsck_opt             ""
```

NOTE:

The databases must reside on the local LVM volumes and having the same volume group name on all the nodes.

4. You can optionally add the volume group and filesystem details of the shared volume where the home directory of SQL Server (`/var/opt/mssql`) resides. Sample configuration if the volume group used is `vg_mssql` and has a logical volume `/dev/vg_mssql/lvol0` mounted on `/var/opt/mssql/`

```
vg                      vg_mssql
fs_name                 /dev/vg_mssql/lvol0
fs_server               ""
fs_directory            /var/opt/mssql/
fs_type                 ext4
fs_mount_opt            ""
fs_umount_opt           ""
fs_fsck_opt             ""
```

   a. When you add the home directory of the SQL Server, you can also optionally add a storage monitoring service.

Table 8: Example Storage_Monitoring service

Attribute	Value
`service_name`	Storage_Monitoring_Service
`service_cmd`	"$SGSBIN/cmresserviced <vg_mssql> " **NOTE:** The LVM volumes where the home directory of the SQL Server resides must be provided in the argument.
`service_restart`	none
`service_fail_fast_enabled`	no **NOTE:** If the`service_fail_fast_enabled` attribute is set to yes and if the service fails on the node on which this package is running, then the node will reboot.
`service_halt_timeout`	0
`service_halt_on_maintenance`	yes

5. Verify the following predefined values of the generic resources:

Table 9: Generic Resource Attributes

generic_resource_name	generic_resource_up_criteria
SQL_Server_Instance_System	<=3
SQL_Server_Instance_Resource	<=3
SQL_Server_Instance_Query_Processing	<=3
SQL_Server_Instance_IO_Subsystem	<=2
SQL_Server_Instance_Events	N/A

6. Verify the validity of the package configuration file using the `cmcheckconf` command.
   ```
   cmcheckconf -k -P mssql_srv_pkg.conf
   ```
7. Run the `cmapplyconf` command to add the package to the Serviceguard environment, if the `cmcheckconf` command does not report any errors.
   ```
   cmapplyconf -k -P mssql_srv_pkg.conf
   ```
8. Run the `cmmodpkg` command to enable the package switching.

   This enables the AOAI toolkit package to failover to the adaptive node when it fails on the primary node.

   ```
   cmmodpkg -e -n node1 -n node2 <mssql_server_pacakage>
   cmmodpkg -e <mssql_server_pacakage>
   ```

   Here `<mssql_server_package>` is the package name provided in the package configuration file `mssql_srv_pkg.conf`.

Configuring the Availability Instance Package

The Availability Instance Package is a Multi-node Package. This package monitors the MSSQL Availability group specified in the package. This package will provide HA and DR for only one availability group. A new package should be configured for each availability group. This package is dependent on the MSSQL Server Instance Package.

Procedure

1. Run the `cmmakepkg` command to create the package configuration template for the MS SQL toolkit.
   ```
   # cmmakepkg -m sgemssql/mssqlaoai <aoai_pkg.conf>
   ```

   The modular style package configuration file is created with all the required attributes for MS SQL toolkit package.
2. Enter the values for the following attributes in the *<aoai_pkg.conf>* package configuration file:

Table 10: Variables in file <*aoai_pkg.conf*>

Variable Name	Description
`sgemssql/mssqlaoai/availability_group`	Availability Group is a container for a set of databases, that fail over together. This attribute specifies the Microsoft SQL for Availability Group name. Maximum length is 128 characters.
`sgemssql/mssqlaoai/aoai_database`	An availability group supports a replicated environment for a discrete set of user databases, known as availability databases. The attribute "aoai_database" are availability database configured for a given Availability Group.
`sgemssql/mssqlaoai/required_synchronized_secondaries_to_commit`	This parameter specifies the number of secondary replicas that needs to replicated synchronously. The value of `required_synchronized_copies_to_commit`, for a 3 node synchronous replica can range from 0 to 2. If availability mode of one of the nodes in 3 node replica is ASYNCHRONOUS, then the value can either be 0 or 1. The online modification of this attribute is supported. You can modify the attribute value provided the value for this is changed outside the Serviceguard context, that is in the SQL server configuration.
`sgemssql/mssqlaoai/nodename`	The attribute `nodename` specifies the name of the node that is part of the Availability Group and also Serviceguard cluster. Online addition of node names is supported. When `nodename` is newly added, availability mode should be specified for the newly added `nodename`.
`sgemssql/mssqlaoai/availability_mode`	This attribute specifies the data replication relationship between the primary and the secondary replica(s). The value can be SYNCHRONOUS or ASYNCHRONOUS.

3. Optionally modify the service name of the pre-configured role monitoring service as required. Update the `service_cmd` argument if you have changed the pre-defined generic resource name.

Table 11: Role Monitoring service

Attribute	Value
`service_name`	Microsoft_SQL_Availability_Replica_Role

Table Continued

service_cmd	"$SGCONF/scripts/sgemssql/hamssqlaoai.sh replica_monitor SQL_Availability_Replica_Role"
service_restart	none
service_fail_fast_enabled	no
service_halt_timeout	65
service_halt_on_maintenance	no

4. Optionally add a storage monitoring service for all the LVM volumes used by this availability group.

Table 12: Example Storage_Monitoring service

Attribute	Value
service_name	Storage_Monitoring_Service
service_cmd	"$SGSBIN/cmresserviced <vg1> <vg2> " **NOTE:** The LVM volumes used by the availability group must be provided in this attribute.
service_restart	none
service_fail_fast_enabled	no **NOTE:** If the service_fail_fast_enabled attribute is set to yes and if the service fails on the node on which this package is running, then the node will reboot.
service_halt_timeout	0
service_halt_on_maintenance	yes

5. Verify the pre-defined Generic Resource Attribute in the package.

Table 13: Generic Resource Attributes

generic_resource_name	generic_resource_up_criteria
SQL_Availability_Replica_Role	<=4

NOTE:

If the generic resource name is changed, then the role monitor service command argument must also be changed in the reflection.

6. Set the dependency on the AOAI multi-node package.

Table 14: Dependency Attributes

Dependency Attributes	Dependency Attribute Value
`dependency_name`	`mssql-svr-mnp-dep`
`dependency_condition`	`mssql_server_package = up`
`dependency_location`	`same_node`

NOTE:

Here, the `mssql-svr-mnp` is the name of the MS SQL server MNP package name.

7. Verify the validity of the package configuration file using the `cmcheckconf` command.
   `cmcheckconf -P aoai_pkg.conf`
8. Run the `cmapplyconf` command to add the package to the Serviceguard environment, if the `cmcheckconf` command does not report any errors.
   `cmapplyconf -P mssql_pkg.conf`
9. Run the `cmmodpkg` command to enable the package switching.

   This enables the AOFI toolkit package to failover to the adaptive node when it fails on the primary node.

   ```
   cmmodpkg -e -n node1 -n node2 aoai_mnp
   cmmodpkg -e aoai_mnp
   ```

   Here `<aoai_mnp>` is the package name provided in the package configuration file `<aoai_pkg.conf>`.

Configuring the Write Intent Package

The Write Intent Package is a failover package. This package starts or runs only on the node where the role of the availability instance or group is primary. This package should be configured for each availability group.

Procedure

1. Run the `cmmakepkg` command to create the package configuration template for the MS SQL toolkit.
   `# cmmakepkg –m sgemssql/mssqlwriteintent write-intent-pkg.conf`

   The modular style package configuration file is created with all the required attributes for MS SQL toolkit package.
2. Set the Generic Resource Attributes in the package.

Table 15: Generic Resource Attribute

generic_resource_name	generic_resource_up_criteria
SQL_Availability_Replica_Role	==2

3. Set the dependency on the AOAI multi-node package.

Table 16: Dependency Attributes

Dependency Attributes	Dependency Attribute Value
`dependency_name`	`aoai-mnp-dep`
`dependency_condition`	`aoai_mnp = up`
`dependency_location`	`same_node`

NOTE:

Here, the `aoai-mnp` is the name of the Availability Instance Package name.

4. Verify the validity of the package configuration file using the `cmcheckconf` command.
   `cmcheckconf -P write-intent-pkg.conf`
5. Run the `cmapplyconf` command to add the package to the Serviceguard environment, if the `cmcheckconf` command does not report any errors.
   `cmapplyconf -P write-intent-pkg.conf`
6. Run the `cmmodpkg` command to enable the package switching.

   This enables the AOFI toolkit package to failover to the adaptive node when it fails on the primary node.

   ```
   cmmodpkg -e -n node1 -n node2 <aoai_write_pkg>
   cmmodpkg -e <aoai_write_pkg>
   ```

   Here `<write-intent-pkg.conf>` is the package name provided in the package configuration file `<aoai_pkg.conf>`.

Different AOAI workload deployments

You can deploy different types of AOAI workloads to provide HA and DR protection for the availability groups. This section provides more information on the types of AOAI workload deployments and how the HA and DR protection is provided to the workloads. SGMSSQL supports AOAI deployments on two or three nodes.

AOAI two node deployments

In a two node deployment, one of the nodes is configured as the primary replica and the other node is configured as the secondary replica.

In this example two node deployment, replica on Node 1 is primary and the replica on Node 2 is configured as the secondary synchronous commit replica. In the event of failure of the primary replica on Node 1, SGMSSQL promotes the secondary sync replica on Node 2, as the primary if it meets all the eligibility criteria.

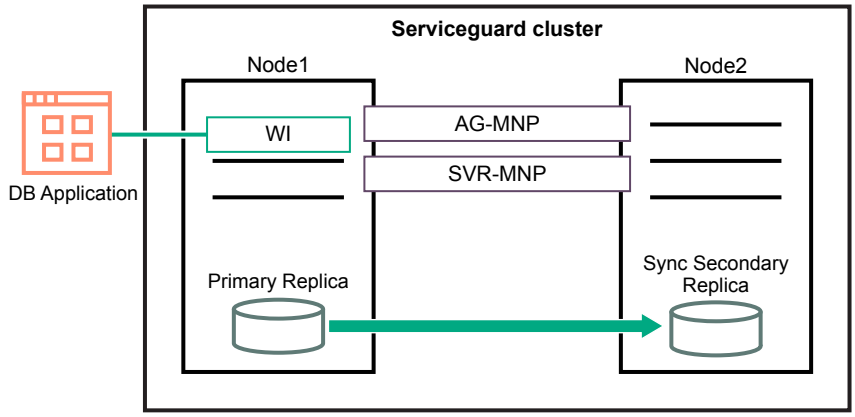


Figure 3: Two node synchronous deployment

AOAI three node deployments

In an AOAI three node deployment, two nodes are configured as synchronous commit replica and the third node can either be synchronous or asynchronous commit replica. Typically all the synchronous replicas are present in the same site.

In a three node synchronous deployment, all the three replicas are configured as synchronous commit. The Write Intent Package (AOAI-WI) runs on a node which hosts the primary replica. In the event of a failure of the primary replica, one of the eligible secondary replicas is promoted as the primary replica by SGMSSQL.

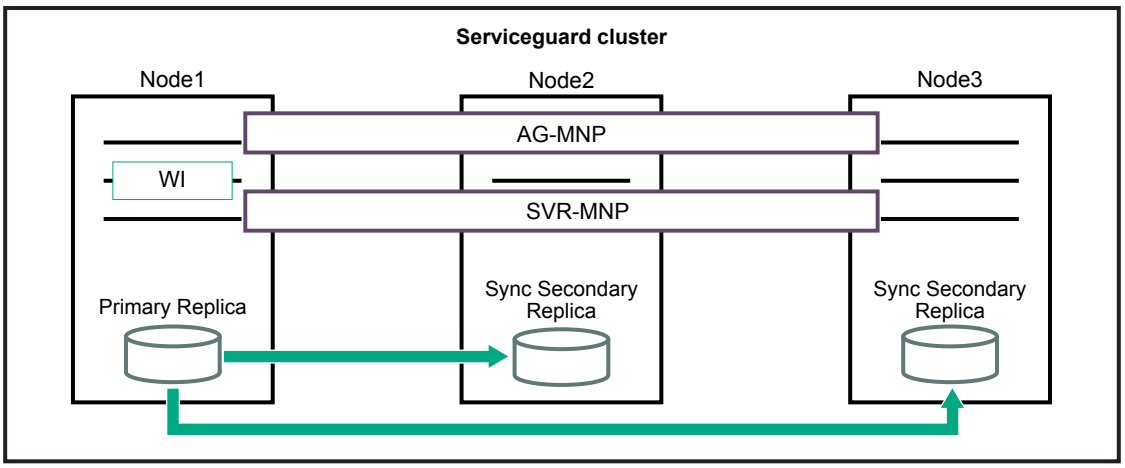


Figure 4: Three node synchronous deployment

In a two node synchronous and a third node asynchronous replica, the asynchronous commit replica might be available at a geographically distant site (Site B).

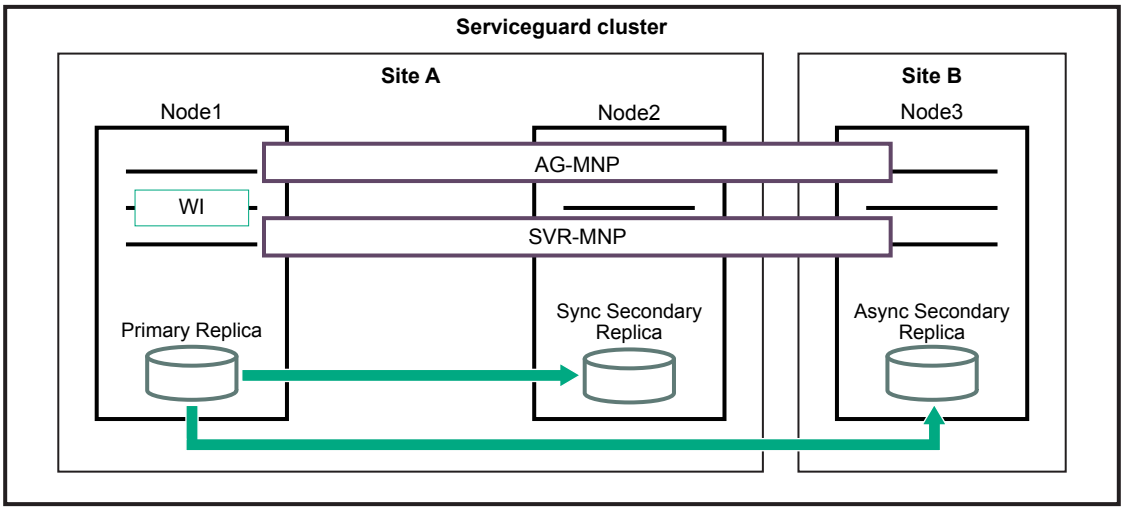


Figure 5: Two node synchronous and a third node asynchronous replica

In the event of a failure of the primary synchronous replica, the secondary synchronous replica will be attempted to be promoted as the primary replica.

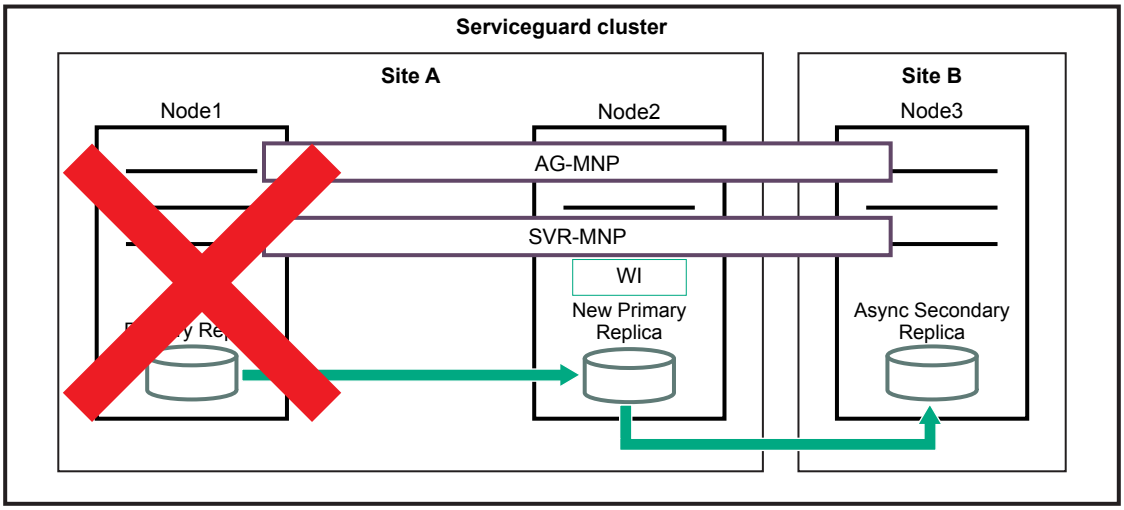


Figure 6: Node two promoted as the primary replica

In case the entire site (Site A) goes down where the synchronous replicas are configured, the asynchronous secondary replica will have to be promoted manually.

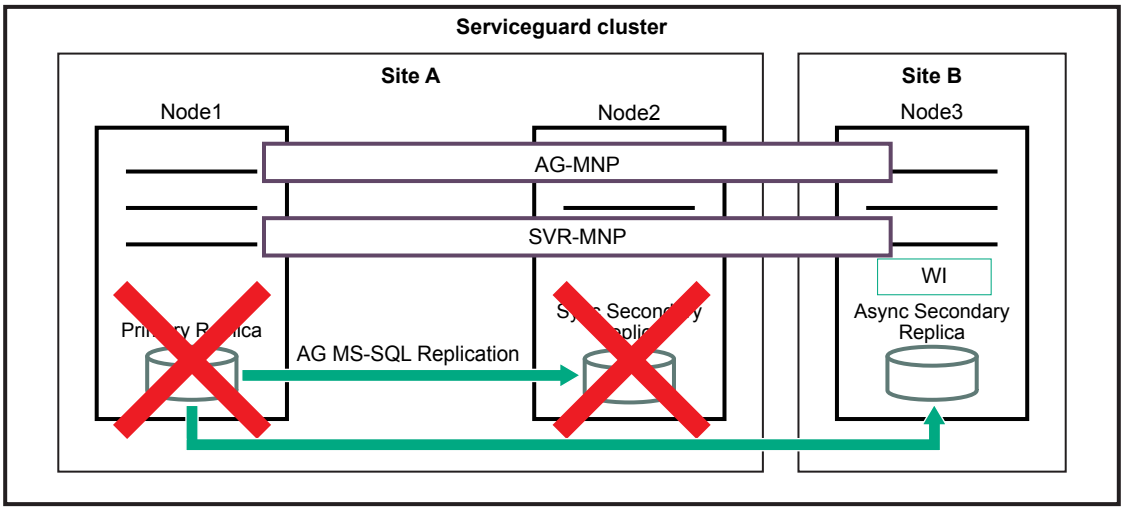


Figure 7: Manual promotion of the asynchronous replica

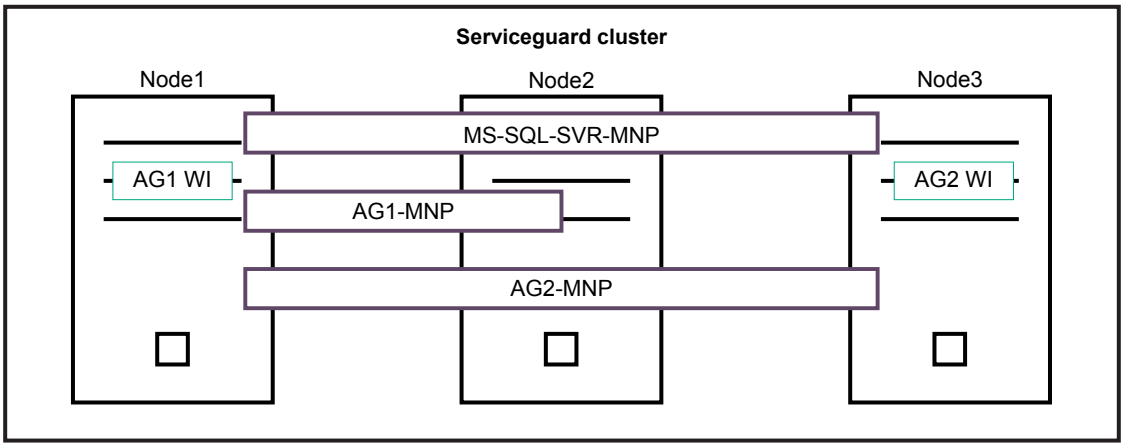
Multiple AOAI deployments in a cluster

You can configure the following three deployments modes on multiple availability groups in the same cluster.

- Two node synchronous deployments
- Three node synchronous deployments
- Two node synchronous and one asynchronous

The SQL Server Instance Package is a common or unique package for the entire cluster. There must be one server instance package for all the availability groups within the cluster. But each availability group must contain individual availability instance and write intent packages.

For example, the workload for availability group AG1 consists of all the three packages for the AOAI solution, such as the server instance (MS-SQL-SVR-MNP), availability instance (AG1-MNP), and the write intent (AG1 WI) packages. And within the same cluster, if you create another workload for an availability group AG2, then you must add the availability instance package (AG2-MNP) and write intent package (AG2 WI) for AG2 availability group. The server instance package is available for the entire cluster and is common to all the availability groups.



Creating multiple AOAI workload deployments in a cluster

Creating multiple AOAI workload deployments is exactly the same as creating single AOAI workload, except that you must add the LVM storage information to the Microsoft SQL Server Instance Package. Rest of the steps to create the workload remains the same.

You can use Serviceguard Manager to deploy the first AOAI workload. For the subsequent AOAI workload deployments in the same cluster, complete the steps from Serviceguard.

Procedure

1. Identify the LVM storage that is used by the availability databases, to create an AOAI workload for a new availability group
2. Add the LVM storage to the Microsoft SQL Server Instance Package.
   a. Get the package configuration file.
   ```
   # cmgetconf -p Microsoft_SQL_Server mssql_srv_pkg.conf
   ```
   b. Edit the package configuration file and add the volume group and file systems information.

   For example, if the volume group used is `vg_ag1` and has a logical volume `/dev/vg_ag1/lvol0` must be mounted on `/var/opt/mssql/data/salesdb`, then update the package configuration file as follows:

   ```
   vg                      vg_ag1
   fs_name                 /dev/vg_ ag1/lvol0
   fs_server               ""
   fs_directory            /var/opt/mssql/data/salesdb
   fs_type                 ext4
   fs_mount_opt            ""
   fs_umount_opt           ""
   fs_fsck_opt             ""
   ```
   c. Verify the validity of the package configuration file using the cmcheckconf command.
   ```
   # cmcheckconf -k -P mssql_srv_pkg.conf
   ```
   d. Apply the new configuration of Microsoft SQL Server Instance Package to the cluster
   ```
   # cmapplyconf -k -P mssql_srv_pkg.conf
   ```
3. **Create and apply the new Availability Instance Package** for the availability group.
4. **Create and apply the Write Intent Package** for the availability group.
5. Run the `cmrunpkg` command to start the packages.

Recovery Point Objective based failover

In AOAI deployments, SGMSSQL supports Recovery Point Objective (RPO) sensitive failover.

RPO based failover is achieved by

- The SQL Server cluster resource attribute `REQUIRED_SYNCHRONIZED_SECONDARIES_TO_COMMIT`
- SGMSSQL periodically monitoring the availability group for data lag and the synchronization state between the primary and the secondary replicas. The monitoring interval is set to 10 seconds.

The cluster resource attribute `REQUIRED_SYNCHRONIZED_SECONDARIES_TO_COMMIT` defines the minimum number of synchronous secondary replicas required to commit before the primary commits a

transaction. The value of this attribute has to be configured for the availability group before adding it to the AOAI workload. For more information about this attribute see, **Microsoft documents**.

The attribute value configured should be lesser than the number of synchronous commit replicas configured in the availability group. The values can be set to either:

- Greater than or equal to 1
- Equal to 0

REQUIRED_SYNCHRONIZED_SECONDARIES_TO_COMMIT >=1

When the `REQUIRED_SYNCHRONIZED_SECONDARIES_TO_COMMIT` value is set to greater than or equal to 1, then there is no data loss or the RPO is zero. In an event of outage of the primary replica, SGMSSQL promotes the eligible secondary as primary replica automatically based on the SGMSSQL periodical monitoring.

The newly promoted replica will be available for user transactions if the value of the attribute `REQUIRED_SYNCHRONIZED_SECONDARIES_TO_COMMIT` is satisfied, else it will be continue to be available for read-only operations.

For example let us consider the case of a two node synchronous deployment, where the `REQUIRED_SYNCHRONIZED_SECONDARIES_TO_COMMIT` is set to 1. In case of a failure of the primary replica, the secondary synchronous replica will be promoted as primary replica automatically. The newly promoted primary replica will be available only for read operations and will not be available for user transactions until the old primary replica joins the cluster.

Similarly an outage to the secondary replica renders the primary replica not available for user transactions. The primary replica is available only for read operations until the secondary replica joins the cluster.

REQUIRED_SYNCHRONIZED_SECONDARIES_TO_COMMIT ==0

When the `REQUIRED_SYNCHRONIZED_SECONDARIES_TO_COMMIT` value is set to 0 the primary replica fails, SGMSSQL validates for data lag and synchronization state before promoting the secondary synchronous replica as the primary. The new primary replica will be available for user transactions.

In case the surviving secondary replicas are lagging or the synchronization state are not healthy, then SGMSSQL will not promote them as primary replica. You might have to manually failover the surviving secondary replica as primary.

If there is an outage to the secondary replica, the primary continues to run and is available for user transactions.

More information

Manual failover of an availability group on page 27
Recovering the primary replica on page 27
Recovering the secondary replica on page 28

Criteria for automatic failover of a secondary replica as a primary

When the primary replica fails, Serviceguard automatically promotes the secondary replica as the primary. The promotion of a secondary replica as primary during a failure is based on following criteria, automatically monitored by Serviceguard:

- The data in the secondary replica is up to date with the previously known primary replica.
- The synchronization state of the secondary replica is healthy (synchronized).
- The secondary replica is configured as a synchronous commit replica.

- The databases of the availability group are healthy.
- The health status of the availability database that are part of the availability group.

Manual failover of an availability group

A manual failover of an availability group is an operation where the secondary replica in the availability group is promoted as the primary replica manually. There are two types of manual failovers and employed during the following situations.

Forced manual failover

Forced manual failover is employed when, there is an outage to the primary replica and the automatic failover of the secondary replica as primary is unsuccessful. An automatic failover of secondary is unsuccessful when one or more of the **automatic failover** criteria is not met. The forced manual failover to unsynchronized secondary may lead to data loss. For more information see, "Forced Manual Failover of an Availability Group" in Microsoft documents.

Planned Manual Failover

Planned manual failover is employed during a window of planned maintenance (hardware or software) on a server which hosts the primary replica. For more information about planned manual failover, see "Perform a planned manual failover of an availability group" in the Microsoft documents.

Performing manual failover of an availability group

Procedure

1. Identify the secondary replica that must be promoted as the primary replica through forced manual failover or planned manual failover.
2. To perform manual failovers, run the commands:

   For performing Forced manual failover

   ```
   # sqlcmd -U <user> -P <password> -Q "sp_set_session_context @key =
   N'external_cluster', @value = N'yes', @read_only = 1; ALTER AVAILABILITY
   GROUP [<AG>] FORCE_FAILOVER_ALLOW_DATA_LOSS" -W
   ```

   For performing Planned manual failover

   ```
   # sqlcmd -U <user> -P <password> -Q "sp_set_session_context @key =
   N'external_cluster', @value = N'yes', @read_only = 1; ALTER AVAILABILITY
   GROUP [<AG>] FAILOVER" -W
   ```

NOTE:

Here, the user is the Microsoft SQL Server user name and the password is the password for the user. The AG is the availability group that is considered for manual failover.

Recovering the primary replica

When the primary replica in the availability group fails, Serviceguard ensures promotion of the eligible secondary replica as primary. The recovery of the old primary replica involves resolving the issue that caused the failure and joining it back to the cluster.

Procedure

1. Resolve the issues that caused the failure of the old primary replica.

The issue can be with the node (hardware) or the software running on the node.

2. Join the node if the node is not running.
   `cmrunnode <node_name>`
3. If Microsoft SQL Server Instance package is not running, start the package using the command:
   `cmrunpkg <SVR-MNP>`
4. If the Availability Instance package is not running, start the package using the command:
   `cmrunpkg <AOAI_AG_MNP_PKG>`

Situations while recovering old primary replica

There can be two types of situations while recovering the primary replica which had failed.

1. Recovering when there is a new primary replica for the availability group

   The old primary replica joins the availability group as resolving and is automatically promoted as the secondary replica by Serviceguard.

2. Recovering when there is no primary replica for the availability group

   This is a case where Serviceguard did not find any eligible secondary replicas for automatic failover. The old primary replica joins the availability group as resolving. The monitoring script might promote the old primary replica as primary again if it meets the required eligibility. If not, the eligible replica must be promoted manually, as there is no clear replica which can be promoted as primary. For more information about failure to promote the secondary replica as the primary, see the package log files.

   For more information about manual failover see, **Manual failover of an availability group** on page 27.

Recovering the secondary replica

Procedure

1. Resolve the issues that caused the outage or failure the secondary replica.

   The issue can be with the node (hardware) or the software running on the node.

2. Join the node if the node is not running.
   `cmrunnode <node_name>`
3. If Microsoft SQL Server Instance package is not running, start the package using the command:
   `cmrunpkg <SVR-MNP>`
4. If the Availability Instance package is not running, start the package using the command:
   `cmrunpkg <AOAI_AG_MNP_PKG>`

Administration of the workload packages

This chapter provides the basic information about administering the packages. For more information about the how to manage and administer the packages see, *Managing HPE Serviceguard for Linux*.

Running a package

To start a package, run the following command:

```
# cmrunpkg <package_name>
```

Halting a package

To halt a package, run the following command:

```
# cmhaltpkg <package_name>
```

Deleting a package

To delete a package from the cluster, run the following command:

```
# cmdeleteconf -p <package_name>
```

This command prompts for a confirmation, before removing the package configuration from the cluster binary file, unless you use the `-f` option.

Maintenance mode of a package

A package can be set in the maintenance mode to perform maintenance operations on the components of a package while the package is running. Maintenance operations are performed to modify the networking and other resource components. Serviceguard does not monitor the package for any failures when the toolkit package is in maintenance mode. It can also be used to perform application-specific maintenance without causing the Serviceguard package to be failed.

For more information, see Maintaining a Package: Maintenance Mode available at Managing HPE Serviceguard for Linux.

Online Modification of the Toolkit package

In addition to the Online modification feature supported by Serviceguard, the online modification feature of toolkit allows you to modify some of the toolkit attribute values, while the toolkit package is up and running.

To perform online package modification:

1. While package is up and running, retrieve the current package configuration values using the command `cmgetconf`.

   For example: `#cmgetconf -p` <<Toolkit_Package_Name>> <<Latest_Toolkit_Conf_File>>

2. Backup this latest configuration file using a copy utility. This allows you to refer to the previous configuration values.

   For example: `# cp -p` <<Latest_Toolkit_Conf_File>> <<Backup_Toolkig_Conf_File>>

3. Edit the <<Latest_Toolkit_Conf_File>> file to modify the required attribute values.

4. Use the `cmcheckconf` command to verify the package configuration file.

   For example: `#cmcheckconf -P` <<Latest_Toolkit_Conf_File>>

5. Use the `cmapplyconf` command to add the new changes in the package configuration to the Serviceguard cluster environment.

   For example: `#cmapplyconf -P` <<Latest_Toolkit_Conf_File>>

NOTE:

Information related to package reconfiguration is logged in the package log.

If you modify any other attributes, the following error message appears on the console.

```
ERROR: Online modification of attribute "<Attribute name>" is not supported
```

Toolkit features

Alert mail notification

This feature facilitates you to configure email address to get the alert notification from toolkit package. Alert mails are sent to configured email addresses in case of failure scenarios. It can be a resource failure or altogether database failure that results in package failure.

Cluster verification feature for MS SQL toolkit

Cluster verification is a proactive mechanism to identify cluster inconsistencies that adversely affect toolkit package fail over to an adoptive node.

- Provide commands to perform cluster wide verification of resources.
- Provide commands to perform package verification within the cluster.

Websites

General websites

Hewlett Packard Enterprise Information Library

www.hpe.com/info/EIL

Single Point of Connectivity Knowledge (SPOCK) Storage compatibility matrix

www.hpe.com/storage/spock

Storage white papers and analyst reports

www.hpe.com/storage/whitepapers

For additional websites, see **Support and other resources**.