- Overview

 | Updated today at 4:20 PM by BhratPatel | Tags: cluster, ha, mp, multiplatforms, overview, sa, samp

Product Documentation

System Automation for Multiplatforms V4.1.0

IBM Support pages

Support Resources for SA MP - Think of it as the home page for your initial support needs. This includes links to all Technote FAQ and troubleshooting guides.

Requests For Enhancements (RFE) community

The Requests For Enhancements (RFE) community page displays the top 25 watched and voted requests.

System Automation for Multiplatforms on RFE community

Comments

There are no comments.

System Automation for Multiplatforms

 | Updated July 2, 2018 by ColetteFinneran | Tags: automation, availability, cluster, ha, high, mp, sa, samp, tsa, tsamp

This wiki provides information about **IBM® Tivoli® System Automation for Multiplatforms** including but not limited to best practices, product features, and new tools.

- Roadmap (Updated April 2018)
- Client Facing Presentation - Overview PPT
- Overview Documentation and Forum links for System Automation for Multiplatforms
- Policies, Utilities, and Media Gallery: How to run System Automation for Multiplatforms in production
- Integration Scenarios: How to integrate System Automation for Multiplatforms with other IBM Tivoli products

Latest Announcement Letters

- April 22, 2016 - IBM Tivoli System Automation for Multiplatforms V4.1.0.3 delivers new operating system support for AIX 7.2 on Power and includes an integrated SAP HANA System Replication high availability and automation solution on Power
- February 18, 2014 - IBM Tivoli System Automation for Multiplatforms V4.1.0.0 delivers an integrated SAP Central Services high availability and automation solution

Latest Blog updates

- High availability clustering with SA MP for TSM 7.1 (and higher)
- High availability clustering with SA MP for TWS
- WHAT DO WE WANT -> Quorum. WHEN DO WE WANT IT -> Now !
- Everything is a "Resource" in the TSAMP and RSCT world
- "Stuck online" - How do I stop the madness ?
- What is needed to troubleshoot an unexpected reboot in a clustered environment ?
- TSA Blog Series: High Availability Concepts - Do I need a TieBreaker?
- TSA Blog Series: High Availability Concepts - What is Quorum ?
- Besides logs, what else would TSAMP Support need to help you?
- How important is diagnostic data for TSAMP Support?
- First place to look for help for TSAMP
- SAMP Shared Storage Setup on Linux and VMware ESX
- SAMP Introduction & Overview
- Big Business, built on the bedrock of Tivoli System Automation

If you want to stay up to date with the latest blog articles, you can subscribe to our System Automation blog Data Center Automation.

Comments

There are no comments.

- Policies

 | Updated today at 4:23 PM by BhratPatel | Tags: policies, samp, sample

SAP high availability policy feature

The following policy-based automated recovery capabilities are provided by the IBM Tivoli System Automation for Multiplatforms optional SAP HA policy feature

- SAP Netweaver
- Oracle database
- NFS server

Pre-canned high availability policies

A set of so called "pre-canned" high availability policies for typical scenarios can be downloaded for free from the IBM integrated service management library (former known as OPAL).
Currently this library contains pre-canned HA policies for following applications:

- Apache
- IBM Tivoli Change and Configuration Management Database 1.1.1
- IBM Tivoli Change and Configuration Management Database 7.1
- DP for my SAP 5.3
- HTTP WebServer
- IBM Tivoli Directory Server
- inetd
- MaxDB SAP 7.5
- NFS Server
- Oracle
- Samba
- Sendmail
- Syslog daemon
- IBM Tivoli Application Dependency Discovery Manager 5.1 + 7.1IT
- TEC 3.8
- TSM
- TWS 8.3
- WAS 6.0
- WebSphere MQ 7
- TSAM

SAMP Pre-canned policies on Integrated Service Management Library

Note: You will not find individual policies here - you have to download and install the package including all currently available policies for your platform. After you extracted the package you'll find the individual policies.

Comments

There are no comments.

SAP high availability policy

 | Updated June 27, 2016 by KonstantinKonson | Tags: automation, availability, ha, high, policy, sap, solution, system, tivoli, tsa

See attachment

Comments

There are no comments.

High Availability solution for SAP with TSA

by Konstantin Konson, Andreas Schauberer,
Enrico Joedecke, George McMullen

Abstract

This document describes the procedure to setup SAP, and implement the high availability solution by means of Tivoli System Automation for Multiplatforms out of the box policy. Implementation of the setup including Central Services and Replication Server are discussed. Standard tests are suggested.

Auditorium

System and SAP admins who is interested in high availability solution for SAP environments.

Content

Overview

The purpose of this document is to record the steps that are required to setup SAP and System Automation for Multiplatforms (TSA MP) as an HA solution for SAP. It described how to setup SAP (A)SCS components as a core of high availability solution. It can be used for various SAP installations like ABAP, Java, Dual Stack, with or without SAP Application Servers.

The high availability solution for SAP uses TSA MP to automate all SAP components. TSA MP detects failed components and restarts or initiates a failover. This setup will also help to reduce the operational complexity of an SAP environment and to avoid operator errors resulting from this complexity.

ABAP setup is discussed in this document, as an example. HA policy for other setups can be similar implemented.

Setup planning

Single Point of Failure (SPOF)

In a distributed or standard SAP installation the SAP Central Services, the database server, and the NFS server are single points of failures (SPOFs).

To minimize the impact of SPOF services outages, it is necessary to setup redundancy. Run one or more standby servers to which each of the SPOF services can be failed over and restarted independently. Each SPOF service must be associated with its own virtual host name, which is started where the service runs. Clients reconnect to the same host name independently of where the SPOF service runs.

The following SAP components are available for a distributed SAP system.

(A)SCS node

The (A)SCS node consists of the stand-alone components Enqueue Server (ES) and Message Server (MS) that operate as SAP Central Services instance. Depending on the SAP solution the (A)SCS node contains the ABAP, Java, or both components. An SAP Instance Agent is running for each instance.

ERS node

The ERS node replicates the ES table entries with purpose to recover them in case of ES failure.

> **Note:** *(A)SCS and ERS components are crucial for an HA solution. They are required for TSA HA policy.*

Primary Application Server node

The Primary AS node consists of the Primary Application Server (PAS) instance that is running the SAP Services Dialog, Update, Batch, Gateway, and Spool. An Instance Agent accompanies the Primary Application Server.

Additional Application Server node

Additional AS nodes are optional. They host the Additional Application Server (AAS) instances, which were called Dialog Instance (DI) in releases prior then SAP kernel 7.1. You can have one or more Additional Application Servers. Again an Instance Agent accompanies each Additional Application Server.

Host Agents

One SAP Host Agent runs on each node that hosts SAP-provided components.

Web Dispatcher and SAProuter node

The optional Web Dispatcher and SAProuter nodes run the SAP Web Dispatcher (WD) and SAProuter, which are used as proxies to access the other SAP instances. An Instance Agent is running for the Web Dispatcher.

> **Note:** *Application Servers, Web Dispatcher, SAProuter are the optional components for an HA solution.*

NFS node

The NFS node runs the NFS server. It can also be a NAS device, which exports the NFS file systems.

> **Note:** *It is highly recommended to make NFS highly available outside of the SAP TSA HA cluster.*

Database node

The database node holds the database instance. The database product can be IBM DB2 or another SAP supported database, e.g. Oracle or HANA.

Database can optionally by included in the HA policy.

SAP clients

The SAP clients connect directly to the Application Servers or to an optional Web Dispatcher.

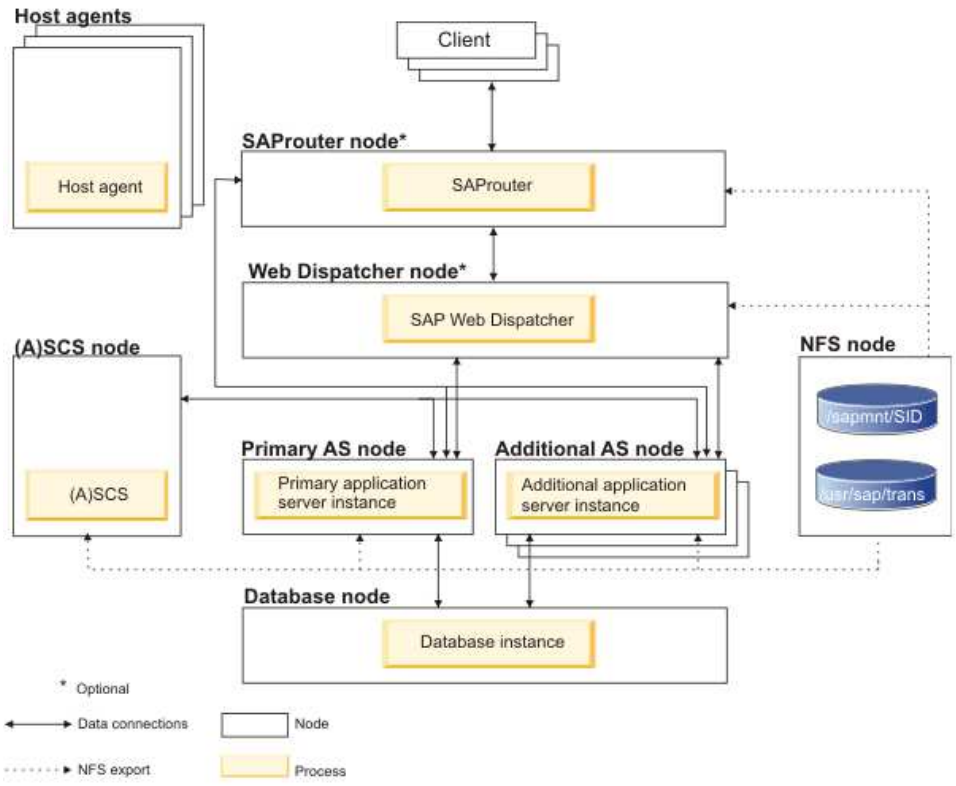

Figure 1. **Components of a distributed SAP system**

Example of a two nodes setup

The minimum hardware setup consists of a two-node System Automation for Multiplatforms domain. The two nodes are either two physical machines or two LPARs running on different physical machines. The systems must be connected via network and need to access the database and SAP data. Data can be provided by a SAN attached disk subsystem, which is connected to each node using fiber channel (FC).

Figure 2 shows an example of a two-node System Automation for Multiplatforms domain. It shows all the main SAP instances and the corresponding failover groups of a SAP ABAP system.

Each machine or LPAR must be capable to run all instances. These are the main SAP instances, which must be made highly available by System Automation for Multiplatforms. High availability for application servers is achieved by having at least two application server instances (PAS and AAS) as fixed resources.

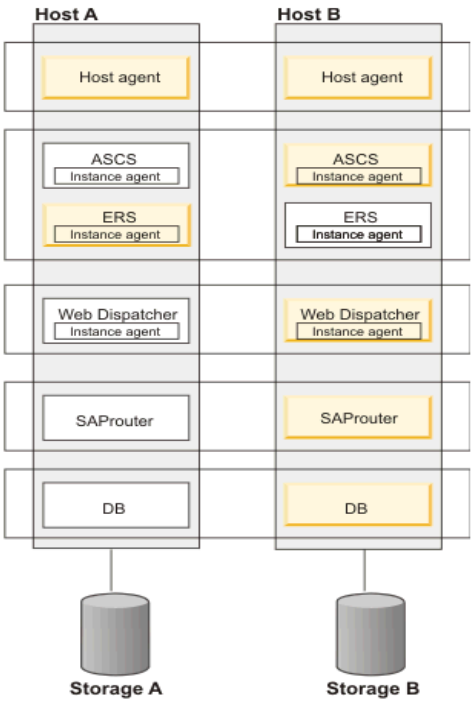

Figure 2. **Example of a two-node setup**

Note: *HA policy for more than two nodes setup is supported.*

Note: *You can manage more than one SAP systems in scope of one TSA cluster*

SAP Central Services high availability setup options

System Automation for Multiplatforms supports three different SAP Central Services high availability installation setups:

- ABAP (ASCS) high availability installation
- Java™ (SCS) high availability installation
- Double-stack (ASCS and SCS) high availability installation

Select the high availability installation that matches your SAP installation.

Location of SAP instance directories

The SAP instance directories `/usr/sap/<sapsid>/<instancename>` must be located on a local file system.

NFS or other distributed file systems are not allowed or accepted.

SAP Central Services and database in the same high availability cluster

If you choose to install the database on the same System Automation for Multiplatforms cluster as SAP, you can run them on the same or on different systems.

When using the policy wizard to define your SAP high availability policy, it is recommended to select a "StartAfter" relationship between the SAP Application Servers and the database server. This relationship will let the database server start before the SAP Application Servers are started. This helps to avoid the problem of having an Application Server started without a database running, which would require a restart of the Application Server. This topic describes a new installation of an ASCS, SCS, or Double Stack high availability SAP system. SAP NetWeaver 7.0 or higher with a kernel version of at least 7.20 is required. This description is based on a two node cluster architecture with a primary and a failover node.

Note: *If DB2 HA policy is implemented in the SAP TSA cluster, do not define StartAfter relationship between AS and DB directly. The additional shadow construct has to be included. Use SAP policy Wizard to make it possible.*

Note: *If the DB2 database runs in the same cluster together with the SAP software, the DB2-bundled System Automation for Multiplatforms license cannot be used, because it allows the automation of DB2 only. So you must ensure that you have a full System Automation for Multiplatforms license for each cluster node.*

Note: *You can use other databases for SAP, e.g. Oracle or HANA. The SAP HA policy is not impacted by the type of database.*

SAP Central Services and database on different high availability clusters

You can have the SAP installation and the database in different System Automation for Multiplatforms high availability clusters.

Advantages:

- The setup and maintenance of the database high availability cluster is independent from the SAP high availability cluster.
- Separate non-root user authorizations for cluster commands against DB2 and SAP resources can be better applied when using separate System Automation for Multiplatforms clusters.

Disadvantages:

- When SAP and DB2 are in separate TSA clusters the *StartAfter* relationship between the SAP cluster and the DB2 cluster require an additional step. A *StartAfter* relationship can be created by adding a *shadow resource* that does a test connection to the DB. Then SAP can start, after the *shadow* indicates the DB is online.

Installing a new high availability SAP system

You can perform a new installation of an ASCS, SCS, or Double Stack high availability SAP system.

SAP NetWeaver 7.0 or higher with a kernel version of at least 7.20 is required. This description is based on a two node cluster architecture with a primary and a failover node. The SAP system on additional nodes can be installed similarly as on failover node.

> **Note:** *Use unique instance numbers for every instance you install on a single host for one SAPSID. The SAP installation does not work if you did not use unique instance numbers. Also the ERS must have a unique instance number.*
>
> *Also the Instance numbers of distinguish SAP systems installed on the clustered OS images, must be unique.*

The SAP installation tool *sapinst* is also called Software Provisioning Manager in the SAP documentation. Make sure that following points are done, before you start *sapinst*.

- Create a separate directory for every installation task to store installation logs and traces. Switch into this directory before you start *sapinst*.
- Ensure to configure the automounter on all nodes to connect to the NFS server and automatically mount the default SAP directory */sapmnt* before you start *sapinst*.
- Register permanent entries for the virtual host names in the DNS server.
- Ensure that the network interfaces that you want to use have the same name on each system. For each virtual IP address defined in the high availability policy, an equivalency of network interfaces is created. Only network interfaces with the same name on each node can be part of each equivalency.
- Temporarily define and activate all required virtual IP addresses on the physical host where you want to start the installation before you start *sapinst*. Be sure to remove the virtual IP addresses after the installation and the initial manual test is completed. Incorrect behavior of the SAP high availability

> **Note:** *You will need a virtual IP address for each resource group that can be moved independently. Usually it is one virtual IP for ASCS group and another one for ERS group. These virtual IPs will be modelled as TSA MP resources.*

solution occurs if you leave the virtual IP addresses permanently defined.

Initial installation on primary node

Use the *sapinst* command to execute the following tasks for the installation option SAP Systems > High-Availability System. For some installation tasks it is required to start *sapinst* with a virtual host name.

1. Activate all virtual IP addresses corresponding to the virtual host names before starting the installations.
2. Central Services Instance for ABAP (ASCS) or Java (SCS):

   ```
   sapinst SAPINST_USE_HOSTNAME=<virtual (A)SCS host name>
   ```

3. Enqueue Replication Server instances (ERS) for ASCS or SCS

   ```
   sapinst SAPINST_USE_HOSTNAME=<virtual ERS host name>
   ```

4. Database instance

   ```
   sapinst SAPINST_USE_HOSTNAME=<virtual DB host name>
   ```

5. Primary Application Server instance

   ```
   sapinst
   ```

6. Remove all virtual IP addresses that have been activated before.

Initial installation on failover node

To install SAP on the failover node, perform the following steps:

1. Activate all virtual IP addresses corresponding to the virtual host names before starting the installations.
2. Use the *sapinst* command to execute the following tasks with the installation option System Copy – Target System - High-Availability System.

   Central Services Instance for ABAP (ASCS) or Java (SCS):

   ```
   sapinst SAPINST_USE_HOSTNAME=<virtual (A)SCS host name>
   ```

   Enqueue Replication Server instances (ERS) for ASCS or SCS

   ```
   sapinst SAPINST_USE_HOSTNAME=<virtual ERS host name>
   ```

Database instance

```
sapinst SAPINST_USE_HOSTNAME=<virtual DB host name>
```

3. Use the *sapinst* command to execute the following installation tasks using the installation option SAP-System > High-Availability System:

   Additional Application Server instance (old name: Dialog Instance)

   ```
   sapinst
   ```

4. Remove all virtual IP addresses that have been activated before.

Configuring SAP profiles

Configure the SAP profiles to comply with the high availability solution provided by System Automation for Multiplatforms.

- Disable `autostart` of all SAP instances in all their profiles by commenting the line `Autostart = 1`.

- To share the enqueue backup file within the Linux or AIX cluster, store the file in the NFS-mounted `/sapmnt/<SID>/global` directory. It can be accessed from all nodes in the cluster where the Enqueue Server can start. Add the following parameter to the (A)SCS profile for sharing the enqueue backup files between nodes:

  ```
  enque/backup_file = $(DIR_GLOBAL)/ENQBCK(A)SCS
  ```

- It is required to disable the SAP restart capability for the Enqueue Server and the Enqueue Replication Server in the appropriate profiles. Otherwise, the automatic restart of SAP by using the command *startsapsrv* mismatches with System Automation for Multiplatforms start function and causes problems with the automation. Set the SAP profile parameter for EN and ERS to `Start_Program_<NR>` in the EN and ERS profiles in the `/sapmnt/<SID>/profile` directory.

  For all servers other than EN and ERS, set the SAP profile parameters `Restart_Program_<NR>`.

  If `Start_Program` is defined, then
  - Initial start is started by `startsapsrv` framework.
  - Recovery start is started by System Automation for Multiplatforms. Resources can start either in place or resources can fail over.

If `Restart_Program` is defined, then
- Initial start is started by `startsapsrv` framework
- Recovery start is started by `startsapsrv` framework. Resources can start in place.

```
SAP profile /sapmnt/<SID>/profile/<SID>_SCS01
#------------------------------------------------------------------------
# Start SAP enqueue server
#------------------------------------------------------------------------
_EN = en.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
Execute_04 = local rm -f $(_EN)
Execute_05 = local ln -s -f $(DIR_EXECUTABLE)/enserver$(FT_EXE) $(_EN)
Start_Program_01 = local $(_EN) pf=$(_PF)
#------------------------------------------------------------------------
# Start SAP message server
#------------------------------------------------------------------------
_MS = ms.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
Execute_02 = local rm -f $(_MS)
Execute_03 = local ln -s -f $(DIR_EXECUTABLE)/msg_server$(FT_EXE) $(_MS)
Restart_Program_00 = local $(_MS) pf=$(_PF)

SAP profile /sapmnt/<SID>/profile/<SID>_ERS02
#------------------------------------------------------------------------
# Start enqueue replication server
#------------------------------------------------------------------------
_ER = er.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
Execute_03 = local rm -f $(_ER)
Execute_04 = local ln -s -f $(DIR_EXECUTABLE)/enrepserver$(FT_EXE) $(_ER)
Start_Program_00 = local $(_ER) pf=$(_PFL) NR=$(SCSID)
```

Verifying the initial installation

The following steps verify the correct setup of the ERS replication.

The commands listed with each verification step assume an ASCS setup. If you verify a Java setup replace the following instance names

- ASCS with SCS
- DVEBMGS with J
- D with J

The syntax of the `ifconfig` commands shown in the samples below applies to the AIX operating system. For Linux, replace the AIX commands with the following Linux commands:

Add IP alias

- **AIX:** `ifconfig <interface_name> <IP_alias> netmask <IP_netmask> alias up`
- **Linux:** `ifconfig <interface_name>:<unique_number> <IP_alias> netmask <IP_netmask> up`

Delete IP alias

- **AIX:** `ifconfig <interface_name> <IP_alias> delete`

- **Linux:** `ip addr del <IP_alias> dev <interface_name>`

Prerequisites:

- All instances are stopped.
- Execute all steps as `<sid>adm` user.

For the verification the SAP system is started and a manual failover of the ASCS and ERS is executed twice. First a failover of the ASCS instance from the first node to the second node and then a failover of the ASCS instance from the second node back to the first node.

Verification step 1: *Initial start - ASCS on first node and ERS on second node*

1. Start the ERS instance on the second node
   ```
   # ifconfig <interface_name> <ERS_IP_alias> netmask <IP_netmask> alias up
   # startsap r3 ERS<ID>
   ```

   Start ASCS and the Primary Application Server instance on the first node:
   ```
   # ifconfig <interface_name> <ASCS_IP_alias> netmask <IP_netmask> alias up
   # startsap r3 ASCS<ID>
   # startsap r3 DVEBMGS<ID>
   ```

2. Start the Additional Application Server instance on the second node:
   ```
   # startsap r3 D<ID>
   ```

3. Check replication status for each ERS instance on the second node using *ensmon* utility:
   ```
   # ensmon pf=/usr/sap/<SID>/ERS<ID>/profile/<SID>_ERS<ID>_<node2>
   ```

   Select task **Get replication information**. The output looks like this

   ```
   ...Replication is enabled in server, replication server is connected.
   Replication is active...
   ```

4. Verify successful start of all Application Servers
   - Logon to the Primary Application Server for ABAP using the SAP graphical user interface.
   - Logon to the Additional Application Server for ABAP using the SAP graphical user interface.

Verification step 2: *Change replication direction - (A)SCS on second node and ERS on first node*

1. Stop Additional Application Server instance and ERS on the second node
   ```
   # stopsap r3 D<ID>
   # stopsap r3 ERS<ID>
   # ifconfig <interface_name> <ERS_IP_alias> delete
   ```

2. Stop ASCS and Primary Application Servers instances on the first node:
   ```
   # stopsap r3 DVEBMGS<ID>
   ```

```
# stopsap r3 ASCS<ID>
# ifconfig <interface_name> <ASCS_IP_alias> delete
```

3. Start ASCS IP instances on the second node:
```
# ifconfig <interface_name> <ASCS_IP_alias> netmask <IP_netmask> alias up
# startsap r3 ASCS<ID>
```

4. Start ERS instances on the first node:
```
ifconfig <interface_name> <ERS_IP_alias> netmask <IP_netmask> alias up
startsap r3 ERS<ID>
```

5. Check replication status for each ERS instance on the first node using *ensmon* utility:
```
# ensmon pf=/usr/sap/<SID>/ERS<ID>/profile/<SID>_ERS<ID>_<node1>
```
Select task **Get replication information**. The output looks like this

...Replication is enabled in server, replication server is connected. Replication is active...

6. Start the Primary Application Server instance on the first node:
```
startsap r3 DVEBMGS<ID>
```

7. Start the Additional Application Server instance on the second node:
```
startsap r3 D<ID>
```

8. Verify successful start of all Application Servers
- Logon to the Primary Application Server for ABAP using the SAP graphical user interface.
- Logon to the Additional Application Server for ABAP using the SAP graphical user interface.

Verification step 3: *Change replication direction - (A)SCS on first node and ERS on second node*

1. Stop Primary Application Server instance and ERS on the first node
```
# stopsap r3 DVEBMGS<ID>
# stopsap r3 ERS<ID>
# ifconfig <interface_name> <ERS_IP_alias> delete
```

2. Stop ASCS and Additional Application Servers instances on the second node:
```
# stopsap r3 D<ID>
# stopsap r3 ASCS<ID>
# ifconfig <interface_name> <ASCS_IP_alias> delete
```

3. Start ASCS IP instances on the first node:
```
# ifconfig <interface_name> <ASCS_IP_alias> netmask <IP_netmask> alias up
```

```
# startsap r3 ASCS<ID>
```

4. Start ERS instances on the second node:
   ```
   ifconfig <interface_name> <ERS_IP_alias> netmask <IP_netmask> alias up
   startsap r3 ERS<ID>
   ```

5. Check replication status for each ERS instance on the first node using *ensmon* utility:
   ```
   # ensmon pf=/usr/sap/<SID>/ERS<ID>/profile/<SID>_ERS<ID>_<node2>
   ```
   Select task **Get replication information**. The output looks like this

   ...Replication is enabled in server, replication server is connected. Replication is active...

6. Start the Primary Application Server instance on the first node:
   ```
   startsap r3 DVEBMGS<ID>
   ```

7. Start the Additional Application Server instance on the second node:
   ```
   startsap r3 D<ID>
   ```

8. Verify successful start of all Application Servers
   - Logon to the Primary Application Server for ABAP using the SAP graphical user interface.
   - Logon to the Additional Application Server for ABAP using the SAP graphical user interface.

After successful verification everything has to be stopped. All virtual IP addresses have to be deactivated.

Now, after initial verification is successfully completed, we can begin with TSA MP installation and implementation of HA policy.

Installing and setting up TSA for Multiplatforms

Make sure, that no iTCO service is active on OS images:

```
lsmod | grep iTCO
```

If active, then disable it:

```
# vi /etc/modprobe.d/blacklist.conf
```

Add:

```
blacklist iTCO_wdt
blacklist iTCO_vendor_support
```

Locating the Software

The customer must download TSA MP from passport advantage to register the product.
Access the Passport Advantage Online website at http://www.ibm.com/software/passportadvantage
Select Passport Advantage Online > Customer sign in Enter your IBM ID and password to log in
Search on "Tivoli System Automation for Multiplatforms" and download version 4.1.0.0

- Tivoli System Automation for Multiplatforms = D03LMLL
- Tivoli System Automation for Multiplatforms SAP HA = D0G6MLL (for the SAP HA solution)

Fixpacks can be downloaded at: http://www.ibm.com/support/fixcentral/

We recommend to take a recent Fixpack (currently 4.1.0.3).
After untar the installation package (see below), the license file for TSA MP is located in ../SAM4100MPLinux/license/sam41.lic. You can optionally copy it into the same location in Fixpack 4.1.0.3 and install 4.1.0.3. So you can skip the installation of 4.1.0.0. Anyway, after TSAMP is installed the license can be applied by running

```
samlicm -i sam41.lic
```

To check the license run

```
samlicm -s sam41.lic
```

Install and configure TSA MP 4.1 on both nodes.

Copy SA_MP_4.1_Linux.tar to /home/tsa and run

```
# tar xfv SA_MP_4.1_Linux.tar
```

As root change to /home/tsa/SAM4100MPLinux

Run

./prereqSAM

If it results in error messages

```
prereqSAM: Error: Prerequisite checking for the ITSAMP installation failed: RHEL
6.5 x86_64
prereqSAM: One or more required packages are not installed: compat-libstdc++-33
(x86_64)
prereqSAM: For details, refer to the 'Error:' entries in the log file:
/tmp/prereqSAM.1.log
```

Then install the library (specifying the lib name)

```
# yum list compat-libstdc++-33
# yum install compat-libstdc++-33
# yum install libstdc++.i686

Total download size: 300 k
Installed size: 908 k
Is this ok [y/N]: y

      Downloading Packages:

      ...
```

If ./prereqSAM is met, then run

./installSAM

Make sure that installation has successfully completed.

Upgrade TSA MP

If necessarily, upgrade TSA MP to the next FixPack.

Download the Fixpack from http://www.ibm.com/support/fixcentral/. Then untar it (tar xfv) and install in the same manner as described above (./installSAM).

Migration to the new TSA MP level

> **Note:** *This step is only necessarily when you have already running TSA MP domain, and then upgrade TSA MP level.*

1. Make sure that the domain is started and that all nodes in the domain are online.
2. Issue the lsrpdomain command to display the version of RSCT that is active in the peer domain, and the mixed version status:

```
# lsrpdomain
Name         OpState RSCTActiveVersion MixedVersions TSPort GSPort
SAP_LOP_test Online  3.1.5.3          Yes            12347  12348
```

3. If the RSCT peer domain is running in mixed version mode (`MixedVersions = Yes`) and all nodes are upgraded to the new release, update the active RSCT version by running the `RSCT CompleteMigration` action on one of the nodes.

   ```
   runact -c IBM.PeerDomain CompleteMigration Options=0
   ```

   To verify that the active RSCT version is updated, enter the lsrpdomain command again:

   ```
   # lsrpdomain
   Name         OpState RSCTActiveVersion MixedVersions TSPort GSPort
   SAP_LOP_test Online  3.1.5.12          No             12347  12348
   ```

4. Run the *samctrl -m* command to activate the new features and to complete the migration:

   ```
   # lssamctrl
   # samctrl –m
   # lssamctrl
   ```

Apply SAP HA solution license

Apply the SAP HA solution license, you downloaded with part D0G6MLL, on both nodes:

```
# samlicm -i sam41SAP.lic
```

You will see output like this

```
# samlicm –s
Product: IBM Tivoli System Automation for Multiplatforms 4.1.0.0
Creation date: Fri Aug 16 00:00:01 2013
Expiration date: Thu Dec 31 00:00:01 2037

Product Annotation: SA for MP - SAP HA policy
Creation date: Fri Dec  6 00:00:01 2013
Expiration date: Thu Dec 31 00:00:01 2037
```

Setting CT_MANAGEMENT_SCOPE variable

Setting the environment variable CT_MANAGEMENT_SCOPE to value 2, you set the scope for TSA (RSCT) commands to PeerDomain, or domain consisting of multiple nodes. This value must be valid in all root and non-root sessions. Therefore it is recommended to put it persistent into e.g. /etc/profile.d :

Create on both nodes

```
/etc/profile.d/tsamp.sh
```

with content

```
export CT_MANAGEMENT_SCOPE=2
```

Configure netmon.cf

If you are running a single-node or two-node cluster, more configuration is required to detect network interface failures.

The cluster software periodically tries to contact each network interface in the cluster. If the attempt to contact an interface fails on one node of a two node cluster, the corresponding interface on the other node is also flagged as offline. It is flagged as offline, because it does not receive a response from its peer.

To avoid this behavior, the cluster software must be configured to contact a network instance outside of the cluster. You may use the default gateway of the sub-net the interface is in.

On each node, create the following file:

```
/var/ct/cfg/netmon.cf
```

Each line of this file contains the system name or IP address of the external network instance. IP addresses can be specified in dotted decimal format.

Example of a `netmon.cf` file:

```
#This is default gateway for all interfaces in the subnet 192.168.1.0
192.168.1.1
```

If Virtual I/O (VIO) is involved, the test becomes unreliable since it not possible to distinguish whether inbound traffic comes from the VIO server or client. The LPAR is not able to distinguish a virtual adapter from a real adapter. To address this problem, the `netmon` library supports up to 32 targets for each local network adapter. If you can ping any of these targets, the local adapter is considered to be up. The targets can be specified in the `netmon.cf` file with the `!REQD` keyword.

```
!REQD <owner><target>
```

or alternatively

```
!IBQPORTONLY !ALL
```

Create and Start the Domain

1. Make sure the hostnames are registered in DNS or properly defined in /etc/hosts
2. Make sure all TSA MP cluster nodes are pingable by each other using short names and dedicated IPs.
3. Run the *preprpnode* command on all nodes in the cluster. This allows all the nodes to exchange the keys and communicate with each other.

```
# preprpnode node1 node2
```

where `node1,node2` are hostnames that is output from command *hostname.*

4. Create a TSA MP domain (aka TSA cluster)

```
# mkrpdomain domain_name node1 node2
```

We recommend to choose the TSA *domain_name* reflecting the SAP system name, hosted by this domain.

```
# lsrpdomain
Name         OpState RSCTActiveVersion MixedVersions TSPort GSPort
SAP_LOP_test Offline  3.1.5.12          No            12347  12348
```

5. Start the TSA MP domain (aka TSA cluster)

```
# startrpdomain domain_name

# lsrpdomain
Name         OpState RSCTActiveVersion MixedVersions TSPort GSPort
SAP_LOP_test Online  3.1.5.12          No            12347  12348

# lsrpnode
Name   OpState RSCTVersion
p6sa62 Online  3.1.5.12
p6sa61 Online  3.1.5.12
```

Configure and activate Tie Breaker

Configure a tiebreaker for cluster environments with an even number of nodes. Tiebreaker provides a decision to grant or deny an Operational Quorum to one of nodes in case of a cluster split brain situation. In this document the network tiebreaker is discussed. Alternatives are the disk, the NFS and the operator-based tie breakers. For more Information refer to TSA Knowledge Center: http://www.ibm.com/support/knowledgecenter/SSRM2X_4.1.0/com.ibm.samp.doc_4.1/sampugsettingup_tiebreaker.html?lang=en

The Network Tiebreaker uses an external IP (network instance) to resolve a tie situation.

To define the Network Tiebreaker run as root

```
# mkrsrc IBM.TieBreaker Type="EXEC" Name="networktb"
DeviceInfo="'PATHNAME=/usr/sbin/rsct/bin/samtb_net Address="xx.xx.xx.xx" Log=1'"
PostReserveWaitTime=30
```

IPAddress is the IP of the external instance which is used for resolving of split situation.

The network default gateway can be used as a tie breaker. To find the gateway run

```
netstat –ar
```

To activate the Network Tiebreaker run as root

```
chrsrc –c IBM.PeerNode OpQuorumTieBreaker=networktb
```

Note: *Please, keep in mind, the IP for network tiebreaker should be chosen the way, it cannot be reached out simultaneously from both sub-clusters (or nodes) in case of cluster split brain. Otherwise, both sub-clusters would get an Operational Quorum, that may damage the critical resources.*

Implementing the SAP Policy

ABAP Central Services (ASCS) high availability policy

> **Note:** *HA policy for Java or Dual SAP setup can be implemented in similar manner. It is also true for additional SAP components like WebDispatcher or SAProuter.*

The ABAP Central Services (ASCS) high availability policy consists of equivalencies, resource groups, floating and fixed resources, that are connected to each other with various relationships. Figure below provides an overview of all resources that can be part of a ABAP Central Services (ASCS) high availability policy.

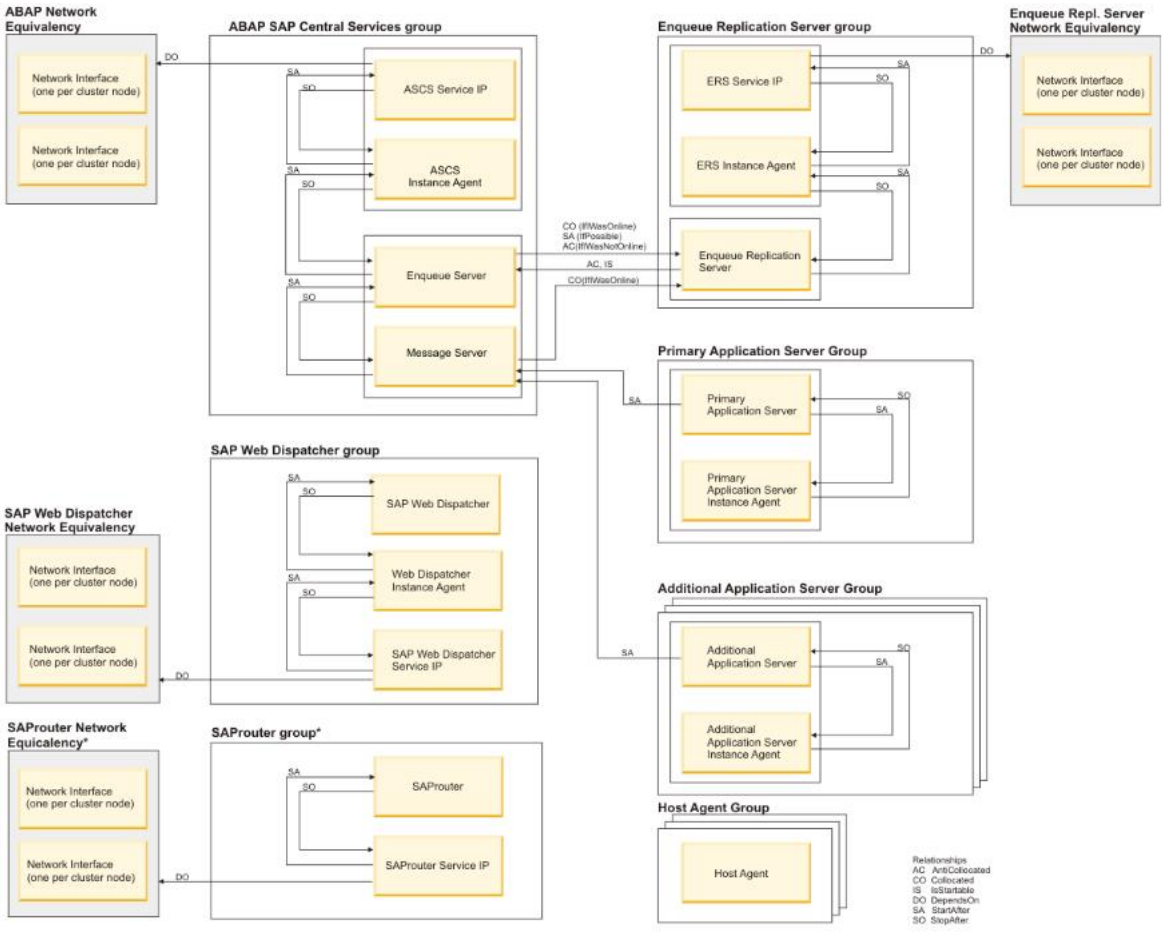

Figure 3. **ASCS high availability policy**

The ABAP SAP Central Services (ASCS) group

The ABAP SAP Central Services (ASCS) group contains four floating resources: a ServiceIP, the Instance Agent resource, and the ABAP Enqueue and Message Servers. All are tied together by StartAfter (SA) and StopAfter (SO) relationships. When the ASCS group is started, the IP resource is started first. Once the IP resource is online, the Instance Agent resource is started next, followed by the Enqueue Server and the Message Server. All resources are contained in collocated groups, so they are always started on the same node. All resources are mandatory group members. No restart is attempted by System Automation for Multiplatforms if one of the resources fails, but a failover of the whole group is triggered instead.

The ABAP Enqueue Replication Server group

The Enqueue Replication Server group contains three mandatory floating resources: a ServiceIP, the Instance Agent resource and the ABAP Enqueue Replication Server (ERS) itself. All are tied together by StartAfter (SA) and StopAfter (SO) relationships. When the ABAP Enqueue Replication Server group is started, the ServiceIP is started first, followed by Instance Agent and the ABAP enqueue replication server.

Using the wizard to configure and activate the SAP Central Services high availability policy

Each SAP Central Services high availability policy consists of a policy template that is tailored by using the `sampolicy` wizard. To configure the template, run the following command:

```
sampolicy –w templateFileName
```

where `templateFileName` is `/usr/sbin/rsct/sapolicies/sap/sap_ABAP_v41.tmpl.xml`
`or /usr/sbin/rsct/sapolicies/sap/sap_JAVA_v41.tmpl.xml` for Java
`or /usr/sbin/rsct/sapolicies/sap/sap_DoubleStack_v41.tmpl.xml` for Dual Stack

The results of the initial template modification will be stored to

```
/etc/opt/IBM/tsamp/sam/policyPool/sap_ABAP_v41.tmpl.xml
```

The next time you start the wizard, use the file that is stored in the policy pool. If you want to start the wizard for the second time, enter the following command:

```
sampolicy -w /etc/opt/IBM/tsamp/sam/policyPool/sap_ABAP_v41.tmpl.xml
```

The initial SAP wizard launch will look like this:

```
System Automation for Multiplatforms Policy Setup Wizard

Policy: SAP ABAP Central Services (ASCS) - Enqueue Replication Server (ERS) HA
policy

Overall parameter status: Missing
------------------------------------------------------------------------------

                            Parameter overview
  # Parameter                                                        Value
------------------------------------------------------------------------------
  1 Enter the name of your SA MP domain.                           Missing
  2 Select the IP version used in the SAP environment.             OK
  3 Specify the existing SAP system ID (SID).                      Missing
  4 Specify your SAP instance owner user name.                     Missing
  5 Enter your desired prefix for all ABAP resources.              OK
```

Figure 4. **Initial launch of SAP HA policy Wizard**

Fill out the required parameters. Use function harvest (4) of the Wizard to get the values automatically suggested.

After all parameters are defined, you can activate the SAP HA policy. Alternatively, you can update the already existing policy using option 2 (Update). Then only difference between already existing and new defined policy will be created.

```
------------------------------------------------------------------------------
System Automation for Multiplatforms Policy Setup Wizard

Policy: SAP ABAP Central Services (ASCS) - Enqueue Replication Server (ERS) HA
policy

Overall parameter status: OK
------------------------------------------------------------------------------

Do you want to activate the policy now?

  1  Yes, activate as new policy
  2  Yes, activate by updating currently active policy
  3  No, save modifications and exit
  4  No, return to parameter overview
```

Figure 5. **Activating or updating SAP HA policy**

Verifying SAP HA solution

Start and stop your SAP Central Services high availability solution to verify if your installation run successfully. Verify the failover scenarios for planned and unplanned outages.

Starting and stopping the SAP Central Services high availability solution

You can start your entire SAP system by issuing the command:

```
chrg -o online -s "Name like '%'"
```

Enter the following command to display your sample SAP ABAP high availability policy:

```
# samcc -V
```

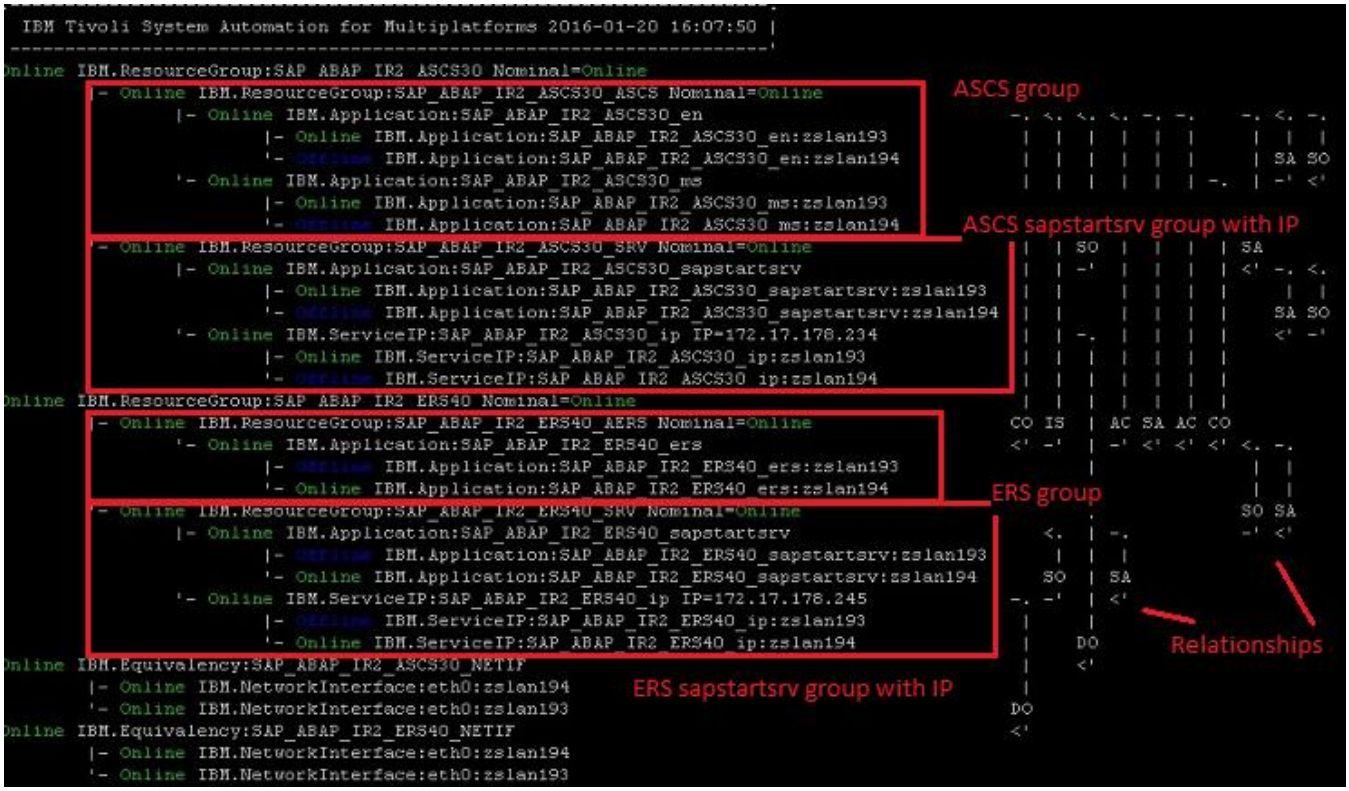

Figure 6. **SAP HA policy as TSA resources**

To stop everything run command

```
chrg -o offline -s "Name like '%'"
```

Note: *Do not start / stop the groups one by one using* ***chrg*** *command, since the policy is designed to bring up and shut down the entire SAP environment. However, you can put the stop requests on the particular groups or resources.*

Note: *It is highly recommended to save the created policy as an xml file. This allow later to activate or update it using sampolicy –a [-u] command, or port it to other TSA cluster. Run command*

```
Create directory /etc/opt/IBM/tsamp/sam/policyPool/savedPolicies
# sampolicy -s /etc/opt/IBM/tsamp/sam/policyPool/savedPolicies/AX0.xml
```

Policy understanding

The Enqueue Server and Enqueue Replication Server are the core of the HA policy. To better understand the policy behavior in various situations, you need to learn the interaction between both servers.

A set of six relationships between the Enqueue and Message Servers and the Enqueue Replication Server provides the most important rules for the high availability of the SAP Central Services.

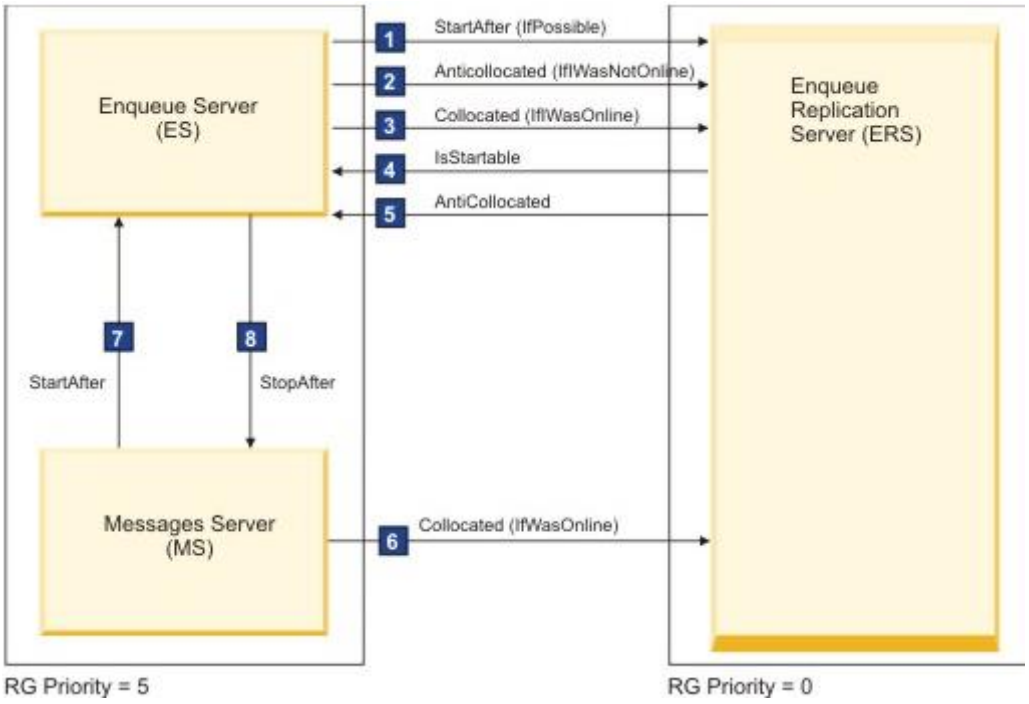

Figure 7. **Relationships between the Enqueue and Message Servers and the Enqueue Replication Server**

ES and MS always start collocated on the same node because of their common group constraint.

Initial Start: all nodes are available

ERS starts first (because of [1]), followed by ES and MS in succession. Since ES was not online before the initial start, ES/MS starts on another node than ERS because of [2], relationships [3] and [5] are not applicable in this situation. So the shadow enqueue table is maintained by ERS on another node than then the one on which ES is running.

Initial Start: only one node is available in a two-node cluster

Because of [2] and [5] ES/MS and ERS cannot be started on the same node. The competitive situation is resolved by the priorities that are assigned to the groups of ES/MS and ERS. The group that holds ES and MS has a higher priority than the ERS group, thus it is started on the sole node. The “IfPossibe” condition relaxes relationship [1]. Thus the SAP Central Services can be made available under the adverse conditions.

Failure of ES

When the Enqueue Server fails, it brings down all members of its group too, because ES is a mandatory member of the group. System Automation for Multiplatforms recovers ES on the node where ERS is running because of [3]. So ES can rebuild its enqueue table from the shadow that was maintained by ERS. Relationship [5] does not apply to this situation nor does relationship [2] since ES was Online before. MS and the other group members follow ES to the node where it was restarted.

There is also an optional restart feature for the Enqueue Server in the SAP profile, which is able to recover a failed ES on the same node. This restart feature must be disabled. Otherwise, ES does not start on the node where ERS runs, hence the rebuild of the enqueue table is not possible. For more information, see chapter *Configuring SAP profiles*.

Failure of MS

The Relationship [6] forces the restart of MS on the node where ERS runs, pulling all other group members, including ES to move to the ERS node.

The restart feature for the Message Server in the SAP profile must be disabled to recover a failed MS on the node with ERS, the same way as for ES.

ERS stop or relocation after ES or MS failure recovery

As described in the previous paragraphs, a failed ES is restarted on the node where ERS is running. After ES recovered its enqueue table from the shadow table, ERS stops itself and is restarted by System Automation for Multiplatforms in succession. The restart of ERS takes place anticollocated to ES on another node because of [5]. Arrow number [4] allows only a node where the appropriate ES constituent is not Failed Offline, so ES would be startable on that node. All other relationships do not apply here.

There is also an optional restart feature for the Enqueue Replication Server in the SAP profile, which is able to recover a failed ERS on the same node. This restart feature must be disabled. Otherwise, ERS does not start on another node away from EN. For more information, see chapter *Configuring SAP profiles.*

ERS failure

If ERS fails for any reason in an otherwise up and running SAP system, it is restarted anticollocated to the ES node because of [5]. Arrow number [4] allows only a node where the appropriate ES constituent is not Failed Offline. All other relationships do not apply. Thus, ERS will restart in place in case of usual outage.

As already mentioned before there is also an optional restart feature for the Enqueue Replication Server in the SAP profile, which must be disabled. For more information, see chapter *Configuring SAP profiles.*

The node where ES is running fails

This scenario is a similar to *Failure of ES*, followed by the *ERS stop or relocation after ES or MS failure recovery* scenario. In a two-node cluster, ERS cannot be restarted on another node as enforced by [5] if the failed ES node is not recovered.

The node where ERS is running fails

This situation is similar to *ERS stop or relocation after ES or MS failure recovery*. The restart of ERS is anticollocated on another node because of [5], but in a two-node cluster there is no other node left. ERS cannot be restarted on another node if the failed ERS node is not recovered.

SAP Actions:

After the ES successfully recovers the table from the ERS shadow the ES terminates the ERS so that the ERS can be restarted by TSA on the next available node

Failover scenarios

The scenarios cover both planned outages (normal operation, maintenance) and unplanned outages (failures). Each scenario should be verified for proper operation.

The following scenarios expect the topology, as defined in the sample policy, to be a cluster with two nodes (node1, node2). We have floating groups for the ASCS and the ERS, and fixed groups for one application server on each node. If your policy differs from this one, e.g. you have more than one SAP Instances, or no Application servers are modelled, the standard test scenarios should be similar.

You can use the `samcc -test` command to monitor the reaction of the system to the actions taken. The colored columns on the right hand site indicate the resources OpStatus change during the action.

Tables below list the important scenarios for planned and unplanned outages. The preconditions for executing the scenarios are listed above the Action, Command and Expected result columns. Each scenario is divided into steps, where each steps precondition is the successful completion of the preceding action. The commands to be executed are listed in the Command column. If you have different naming conventions, you have to adapt the commands accordingly. The last column of the tables lists the expected result.

For the command examples replace the ABAP or Java prefix depending on whether you have ABAB or Java setup. Also replace DVEBMSG and D for ABAP application servers with J for Java application servers.

Table 1. Normal Operations and Maintenance scenarios

Scenario	Action	Command	Expected result
Normal operation	Precondition: All groups offline		
	Start an SAP system AX6	`chrg -o online -s "Name like 'AX6_%'"`	• (A)SCS and DVEBMGS/J groups start on node1. • ERS and D/J groups start on node2.
	Stop SAP system AX6	`chrg -o offline -s "Name like 'AX6_%'"`	• (A)SCS, ERS, DVEBMGS/J and D/J groups stop.
	Move (A)SCS AX6 to other node	`rgreq -o move SAP_ABAP_AX6_ASCS10`	• (A)SCS moves to the node with running ERS (node2). • After replication ERS → ES is completed, ERS is shutdown by SAP. TSA is not involved into this step. • ERS is started by TSA on the next available node from ERS NodeNameList (node1).
	Move ERS AX6 to other node	`rgreq -o move SAP_ABAP_AX6_ERS20`	• ERS moves to the node with no running (A)SCS. This scenario is only valid for more

Scenario	Action	Command	Expected result
			than two nodes TSA clusters. In the two nodes cluster the move request against ERS will have no effect (ERS will restart in place).
Maintenance	Precondition: • (A)SCS and DVEBMGS/J groups are online on node1 • ERS and D/J online on node2		
	Move all resources away from node1 in order to apply operating system or hardware maintenance.	`samctrl -u a node1`	• (A)SCS and DVEBMGS/J groups stop • DVEBMGS/J groups have status "Failed Offline". • (A)SCS group starts on node2. • ERS terminates (by SAP). • ERS groups sacrificed.
		Apply maintenance, reboot, etc. Then resume node1 for automation	
		`samctrl -u d node1`	• DVEBMGS/J groups and ERS groups start on node1.
	Stop and restart Enqueue Replication Server in order to apply SAP maintenance (code or profile changes).	`rgreq -o stop SAP_ABAP_AX6_ERS20`	• ERS group stops. • Offline Request appears on the ERS group.
		`rgreq -o cancel SAP_ABAP_AX6_ERS20`	• ERS group starts on node1. • Offline Request disappears on the ERS group.
	Stop and restart Primary Application Server in order to apply SAP maintenance (code or profile changes).	`rgreq -o stop SAP_ABAP_AX6_DVEBMGS02_node1`	• DVEBMGS/J group stops. • Offline Request appears on the AppServer group.
		`rgreq -o cancel SAP_ABAP_AX6_DVEBMGS02_node1`	• DVEBMGS/J group restarts on node1. • Offline Request disappears on the AppServer group.

Table 2. Unplanned outages

Scenario	Simulation action/command	Expected result
Precondition: • (A)SCS and DVEBMGS/J groups online on node1 • ERS and D groups online on node2 To simulate a software failure create a script using the name `killscript`. Add the following content: `kill $1 `ps -ef | grep $2 | grep -v grep | awk '{print $2}'``		
Failure of the (A)SCS Enqueue Server	node1: `killscript -9 en.sapAX6_10_ASCS`	• (A)SCS group stops and restarts on node2. • ERS ends after some seconds (by SAP). • ERS group restarts on node1.
Failure of the Enqueue Replication Server	node1: `killscript -9 er.sapAX6_20_ERS`	• ERS group stops and restarts on node1.
Failure of the Message Server	node2: `killscript -9 ms.sapAX6_10_ASCS`	• (A)SCS MS restarts on node2 if restart is configured in MS profile. • (A)SCS MS restarts on node1 if restart is not configured in MS profile.

Failure of an (A)SCS application server	node1, ASCS:
	`killscript -9 dw.sapAX6_ DVEBMGS30`
	node1, SCS:
	`killscript -2 jc.sapAX6_J40`
		• DVEBMGS/J application servers restart on node1.

Failure of the node where ES is running	node2: `reboot`	• (A)SCS groups are started on node1. • ERS on node1 stops after some seconds and is restarted on node2 as soon as node2 is available in the cluster again.

Enabling the SAP high availability Connector

Configure SAP profiles

After System Automation for Multiplatforms is installed on all cluster nodes, the SAP high availability Connector must be configured in the SAP profiles. It is sufficient to enter the required entries into the default profile.

Enable the SAP high availability connector. Otherwise, all start or stop commands of all SAP tools, for example `sapcontrol` or SAP MC, are reversed by System Automation for Multiplatforms.

Depending on whether you have an AIX or Linux platform, add the following entries to the default profile of your SAP system. Replace `<SAPSID>` with the SAPSID of your SAP System:

AIX

```
#-----------------------------------------------------------------------
# SAP high availability connector
#-----------------------------------------------------------------------
service/halib = /usr/sap/<SAPSID>/SYS/exe/uc/rs6000_64/saphascriptco.o
service/halib_cluster_connector = /usr/sbin/rsct/sapolicies/sap/bin/sap_tsamp_cluster_connector
```

Linux

```
Replace <your platform> with the appropriate directory name:
#-----------------------------------------------------------------------
# SAP high availability connector
#-----------------------------------------------------------------------
service/halib = /usr/sap/<SAPSID>/SYS/exe/uc/<your platform>/saphascriptco.so
service/halib_cluster_connector = /usr/sbin/rsct/sapolicies/sap/bin/sap_tsamp_cluster_connector
```

Setting up non-root user Id for the command line interface

> **Note:** *It is required to setup non-root user security for the <SID>adm user. The sapstartsrv process calls the HA connector by using the <SID>adm user.*

> **Note:** *This procedure is valid even if you just want to set up a non-root user Id for sa_admin role.*

You must create a role for a System Automation for Multiplatforms admin with general settings that allow non-root users to manage all resource classes from any node that is defined within the cluster. Use the following

procedure to create the `sa_admin` role. Note that root authority is required. This example shows the commands for a Linux environment.

The commands 1-3 may not be required, when you want to assign the TSA admin role to already existing SAP user ID.

1. Create the user ID that is authorized to manage System Automation for Multiplatforms on all nodes:

   ```
   # /usr/sbin/useradd ax6adm
   ```

2. Create a group for the user ID on all nodes:

   ```
   # /usr/sbin/groupadd sapgroup
   ```

3. Add the user ID to the group on all nodes

   ```
   # /usr/sbin/usermod –G sapgroup ax6adm
   ```

   Make sure environment variable CT_MANAGEMET_SCOPE=2 for all TSA users on all nodes.

4. Change the group ownership of the file `/var/ct/IBM.RecoveryRM.log`.

   By default, the file is owned by the user group `root`:

   ```
   -rw-r--r--  1 root root   204 Oct 4 22:00 /var/ct/IBM.RecoveryRM.log
   ```

   Change the group ownership to `sapgroup`

   ```
   /bin/chgrp sapgroup /var/ct/IBM.RecoveryRM.log
   ```

   Change the file permission to `664`

   ```
   # /bin/chmod 664 /var/ct/IBM.RecoveryRM.log
   -rw-rw-r-- 1 root sapgroup 204 Oct 4 22:00 /var/ct/IBM.RecoveryRM.log
   ```

   If the file `/var/ct/IBM.RecoveryRM.log` does not exist after the initial installation of System Automation for Multiplatforms, you can create a dummy file by running the `/usr/bin/touch` command:

   ```
   # /usr/bin/touch /var/ct/IBM.RecoveryRM.log
   ```

5. Modify the file `/var/ct/cfg/ctsec_map.global` on all nodes.

   You must add the following entries for the user ID ax6adm to the RSCT global authorization identity mapping file (`/var/ct/cfg/ctsec_map.global`) on every node in the cluster. Add the new entries above the entry for the user `clusteruser`:

   ```
   unix:ax6adm@<cluster>=sa_admin
   unix:ax6adm@<any_cluster>=sa_admin
   unix:ax6adm@<iw>=sa_admin
   ..
   unix:*@*=clusteruser
   ```

The file is used to map a local user ID on a node to a global user ID within the System Automation for Multiplatforms domain. In the example, the local user ID `ax6adm` is mapped to the global user ID `sa_admin`.

You can authorize more local user IDs for System Automation for Multiplatforms by adding lines to this global map file (on all nodes), and mapping them to the wanted role operator or administrator.

If the file `/var/ct/cfg/ctsec_map.global` does not exist on a node, copy the default file `/usr/sbin/rsct/cfg/ctsec_map.global` to the directory `/var/ct/cfg` and add the new entries to the file `/var/ct/cfg/ctsec_map.global`. Do not remove any entries from the file `/var/ct/cfg/ctsec_map.global` that exist in the default file you copied. The `/var/ct/cfg/ctsec_map.global` files on all nodes within the cluster must be identical. Always add new IDs for non-root users above the entries for the user `clusteruser`.

6. Modify the file `/var/ct/cfg/ctrmc.acls` on all nodes. You must add the following entries for the global user ID `sa_admin` to the file `/var/ct/cfg/ctrmc.acls` on every node in the cluster and comment the line that starts with `LOCALHOST`, for example:

```
# The following stanza contains default ACL entries.
# These entries are appended
# to each ACL defined for a resource class and
# are examined after any entries
# explicitly defined for a resource class
# by the stanzas in this file,
# including the OTHER stanza.

   DEFAULT
    root@LOCALHOST    *    rw
    # LOCALHOST       *     r  // comment this line out!
    none:root   *  rw          // give root access to all
    none:sa_admin   *  rw            // append this row for sa_admin
```

When you completed the required modifications, run the following command on every node in the cluster to activate the changes:

```
# /usr/bin/refresh -s ctrmc
```

7. Extra changes that are required to use *sampolicy* and **samadapter* commands:

```
# /bin/chgrp -R sagroup /opt/IBM/tsamp/sam/cfg
# /bin/chmod g+ws /opt/IBM/tsamp/sam/cfg
# /bin/chmod g+w /opt/IBM/tsamp/sam/cfg/*

# /bin/chgrp -R sagroup /var/ibm/tivoli/common/eez/logs
# /bin/chmod g+ws /var/ibm/tivoli/common/eez/logs
# /bin/chmod g+w /var/ibm/tivoli/common/eez/logs/*
```

```
# /bin/chgrp -R sagroup /etc/opt/IBM/tsamp/sam/cfg
# /bin/chmod g+ws /etc/opt/IBM/tsamp/sam/cfg
# /bin/chmod g+w /etc/opt/IBM/tsamp/sam/cfg/*
```

When you completed the steps successfully, the local user ax6adm can run operational tasks of System Automation for Multiplatforms, such as issuing start and stop requests against resources, and run administrative tasks, such as defining and modifying policies.

Troubleshooting

ERS does not failover

> **Note:** *For all operations: make sure the environment variable CT_MANAGEMENT_SCOPE=2 for all users of TSA. You can set it permanently, e.g. in /etc/profile.d/tsamp.sh:*
>
> `export CT_MANAGEMENT_SCOPE=2`

In the Enqueue Server failover scenario, the ES moves to the ERS node in order to recover its tables. After the table replication ERS → ES is completed, ERS should be shutdown by SAP. If this does not happen, and ERS remains online on the initial node, then:

- Check definitions in ES and ERS profiles (see chapter *Configure SAP profiles*).
  In particular, make sure, Start_Program_00 is configured properly:
  `Start_Program_00 = local $(_ER) pf=$(_PFL) NR=$(SCSID)`
- Check replication is running after initial start and after failover

  Check replication status for the ERS instance on the second node using *ensmon* utility:
  `# ensmon pf=/usr/sap/<SID>/ERS<ID>/profile/<SID>_ERS<ID>_<node2>`

  Select task **Get replication information**. The output looks like this

  ```
  ...Replication is enabled in server, replication server is connected.
  Replication is active...
  ```

- SAP high availability failover scenarios may result in not shutting down the ERS by SAP, after ES has failed over to the ERS server. Consequently, TSA is not able to start the ERS on the alternative server.

  Check the consistency of global and local ERS profiles: as opposed to the standard architecture of any SAP instance, the ERS instance directory on the LINUX servers has a local profile directory. Below are the global and local profile directories location:

  global

  /sapmnt/<SID>/profile

  specific to ERS instance:

  /usr/sap/<SID>/ERS03/profile

Make sure, that the content of the global and local ERS profiles is consistent, in particular the parameter Autostart has to be out commented:

```
#Autostart=1
```

TSA groups are not stable Online or ServiceIP not online

ServiceIP cannot not be brought online. TSA Groups are not stable online. The other constituent resource indicates PendingOffline instead Offline, like this:

```
.-----------------------------------------------------------------------.
| IBM Tivoli System Automation for Multiplatforms 2016-03-29 15:01:59 |
'-----------------------------------------------------------------------'
Online IBM.ResourceGroup:SAP_ABAP_XH0_ASCS92 Nominal=Online
        |- Online IBM.ResourceGroup:SAP_ABAP_XH0_ASCS92_ASCS Nominal=Online
                |- Online IBM.Application:SAP_ABAP_XH0_ASCS92_en
                        |- Online IBM.Application:SAP_ABAP_XH0_ASCS92_en:tst1500213
                        '- Pending offline IBM.Application:SAP_ABAP_XH0_ASCS92_en:tst1500221
                '- Online IBM.Application:SAP_ABAP_XH0_ASCS92_ms
                        |- Online IBM.Application:SAP_ABAP_XH0_ASCS92_ms:tst1500213
                        '- Pending offline IBM.Application:SAP_ABAP_XH0_ASCS92_ms:tst1500221
        '- Online IBM.ResourceGroup:SAP_ABAP_XH0_ASCS92_SRV Nominal=Online
                |- Online IBM.Application:SAP_ABAP_XH0_ASCS92_sapstartsrv
                        |- Online IBM.Application:SAP_ABAP_XH0_ASCS92_sapstartsrv:tst1500213
                        '- Pending offline IBM.Application:SAP_ABAP_XH0_ASCS92_sapstartsrv:tst1500221
                '- Online IBM.ServiceIP:SAP_ABAP_XH0_ASCS92_ip
                        |- Online IBM.ServiceIP:SAP_ABAP_XH0_ASCS92_ip:tst1500213
                        '- Offline IBM.ServiceIP:SAP_ABAP_XH0_ASCS92_ip:tst1500221
Online IBM.ResourceGroup:SAP_ABAP_XH0_ERS92 Nominal=Online
        |- Online IBM.ResourceGroup:SAP_ABAP_XH0_ERS92_AERS Nominal=Online
                '- Online IBM.Application:SAP_ABAP_XH0_ERS92_ers
                        |- Pending offline IBM.Application:SAP_ABAP_XH0_ERS92_ers:tst1500213
                        '- Online IBM.Application:SAP_ABAP_XH0_ERS92_ers:tst1500221
        '- Online IBM.ResourceGroup:SAP_ABAP_XH0_ERS92_SRV Nominal=Online
                |- Online IBM.Application:SAP_ABAP_XH0_ERS92_sapstartsrv
                        |- Pending offline IBM.Application:SAP_ABAP_XH0_ERS92_sapstartsrv:tst1500213
                        '- Online IBM.Application:SAP_ABAP_XH0_ERS92_sapstartsrv:tst1500221
                '- Online IBM.ServiceIP:SAP_ABAP_XH0_ERS92_ip
                        |- Offline IBM.ServiceIP:SAP_ABAP_XH0_ERS92_ip:tst1500213
                        '- Online IBM.ServiceIP:SAP_ABAP_XH0_ERS92_ip:tst1500221
Online IBM.ResourceGroup:SAP_HOST_AGENT_tst1500213 Nominal=Online
        '- Online IBM.Application:SAP_HOST_AGENT_tst1500213_ha:tst1500213
Online IBM.ResourceGroup:SAP_HOST_AGENT_tst1500221 Nominal=Online
        '- Online IBM.Application:SAP_HOST_AGENT_tst1500221_ha:tst1500221
Online IBM.Equivalency:SAP_ABAP_XH0_ASCS92_NETIF
        |- Online IBM.NetworkInterface:eth0:tst1500213
        '- Online IBM.NetworkInterface:eth0:tst1500221
Online IBM.Equivalency:SAP_ABAP_XH0_ERS92_NETIF
        |- Online IBM.NetworkInterface:eth0:tst1500213
        '- Online IBM.NetworkInterface:eth0:tst1500221
```

Figure 8. **TSA groups are not stable online**

Check that the ServiceIP is defined properly. For example in Linux the ServiceIP has not to be defined in file `/etc/sysconfig/network/ifcfg-eth0`. If it was defined there with purpose of manual failover test, then remove this definition from the file.

> ***Note**: Generally, ensure that the ServiceIP addresses are not enabled within node startup procedures...*

SCS IDs have to be unique

SAP Instance numbers (SCS IDs) for ASCS, ERS, AppServer, etc. have to be unique, for example ASCS20, ERS30, DVEBMGS40, etc. Also the Instance numbers of distinguish SAP systems installed on the clustered OS images, must be unique.

For more information

IBM Tivoli System Automation for Multiplatforms Knowledge Center
http://www.ibm.com/support/knowledgecenter/SSRM2X_4.1.0/com.ibm.samp.doc_4.1/samp_IC_hapol.html?lang=en

About the Authors

Konstantin Konson is an expert in High Availability and Disaster Recovery automated solutions using IBM System Automation and Runbook Automation Product Family. He is working in IBM Research and Development Laboratory Boeblingen (Germany) as an IT consultant for customer solutions.

George McMullen is an expert in High Availability and Disaster Recovery automated solutions using IBM System Automation and Runbook Automation Product Family. He has been assisted over 75 customers with TSA solutions since 2007.

Andreas Schauberer is the IBM System Automation for Multiplatforms Lead Developer working in IBM Research and Development Laboratory Boeblingen (Germany).

Enrico Joedecke is the IBM System Automation for Multiplatforms Change Team Lead working in IBM Research and Development Laboratory Boeblingen (Germany).

IBM Deutschland Research & Development GmbH
Department 1311
Schoenaicher Str. 220
D-71032 Boeblingen
Federal Republic of Germany

- Utilities

1 Like | Updated July 30, 2014 by Andreas.Schauberer | Tags: appcmd, samadmin, samp, sharedstorage, testsuite, tools

Using the tool appcmd to develop and test IBM Tivoli System Automation for Multiplatforms policies

The tool appcmd can be used for testing IBM Tivoli System Automation for Multiplatforms policies as well as IBM Tivoli System Automation Application Manager end-to-end and Agentless Adapter policies. You can use it to instrument IBM.Application resources or IBM.RemoteApplication resources. appcmd simulates these applications, allows to mimic failures and tracks the execution of commands which makes it a useful tool to develop and test policy constructs without need to install real applications or impacting real applications.

appcmd_v4.zip

The zip archive contains the files:

- A Readme.txt
- The appcmd tool, a single Perl script.
- A script appcmd_policy_converter.pl which can be used to convert policies
- A whitepaper that describes how appcmd works and how it can be used for policy testing including a tutorial.

Using the TestSuite Light to develop and test IBM Tivoli System Automation for Multiplatforms policies

You can use the standalone tool TestSuite Light to verify a System Automation for Multiplatforms cluster and to verify your automation policies.

TestSuiteLight_v1.zip

The zip archive contains the files:

- This README.txt
- The TSL.pm tool, a single Perl script that is the entire testsuite
- A whitepaper TSA_TestSuiteLight.pdf that documents the TSL
- A whitepaper TSA_cluster_verification.pdf that documents the TSL
- TSL_TC_TEMPLATE.pm, a testcase template to implement own test cases
- The test case CLUSTER_VERIFICATION.pm which implements the TSA_cluster_verification.pdf checks to semi-automatically verify a domain setup.
- appcmd.zip the appcmd tool which simulates resources. It contains an own documentation.

Using "SAMADMIN" - a smitty-like menu driven administrator interface for SAMP

The samadmin tool provides an utility which acts as new interface for System Automation for Multiplatforms. It is designed to provide common SA MP functions for easy use in text-only environments ("putty window"). Instead of looking up and typing-in command line commands of SAMP it provides menus to do common tasks which easily guide you in the creation of a first SAMP cluster with resources. It can be used to learn the command line interface commands as well.

Currently this tool provides following features:

- Domain Management
- Resource and Group Management
- Equivalency Management
- Relationship Management
- TieBreaker Management
- Cluster Overview

sam.admin-3.2.2.0-11138.zip

New functionality will be added to this tool in the future - **feedback is always welcome**...

Prerequisites:

- Linux (i386, ppc, s390)
- SA MP 3.1 or greater
- ncurses library in 32 Bit (e.g. ncurses-libs-5.7-3.20090208.el6.s390)

Installation:
rpm -i sam.admin-3.2.2.0-11138.i386.rpm

Uninstallation:
rpm -e sam.admin

Start Application:
samadmin

Comments

There are no comments.

SAMP Adapter non-root setup

Like | Updated July 16, 2013 by seddik | Tags: *None*

By default, the Tivoli System Automation for Multiplatforms (SA MP) end-to-end automation adapter is configured to run with a root user.

We have now documented how the adapter can be configured to run with a non-root user, for version 3.2.2 of SA MP, starting with version SA MP 3.2.2.4.

In addition to the documentation, we provide a script to aid in the setup.

The attachement section of this page contains a zip file with the documentation and the setup script.

Comments

1-1 of 1 Previous | Next

AxelHoppe commented on June 21, 2013 Permalink

Cool stuff, Isabell! This beauty can probably avoid some discussions... ;-)

Show 10 | 25 | 50 items per page Previous | Next

- Media Gallery

 | Updated July 31, 2014 by Andreas.Schauberer | Tags: *None*

Videos

- SAP High Availability with IBM Tivoli System Automation for Multiplatforms
- SAP High Availability with IBM Tivoli System Automation for Multiplatforms - Part II

Redbooks and Redpieces

- Practical Migration to Linux on System z - with HA considerations using SA for Multiplatforms
- End-to-end Automation with IBM Tivoli System Automation for Multiplatforms
- IBM Tivoli System Automation for Multiplatforms Terminology
- Websphere Application Server Network Deployment V6: High Availability Solutions - Chapter 10
- IBM Tivoli Storage Manager in a Clustered Environment
- DB2 Integrated Cluster Environment Deployment Guide
- mySAP Business Suite Managed by IBM Tivoli System Automation for Linux
- Using IBM Tivoli System Automation for Linux
- Reliable Scalable Cluster Technology (RSCT)

White Papers and Articles

- SAP on IBM System z (zOS and zLinux)
- High Availability Architectures For Linux on IBM System z
- Education - Getting Started Guide

Comments

There are no comments.

- Integration Scenarios

Like | Updated October 6, 2016 by Konstantin_Konson | Tags: *None*

System Automation for Multiplatforms integrates with other Tivoli applications to provide a comprehensive solution. The integration of Tivoli applications in your environment requires specific configuration tasks to adapt to your existing infrastructure.

Tivoli System Automation for Application Manager

Event consoles: Tivoli Enterprise Console® (TEC) or Tivoli® Netcool/OMNIbus (OMNIbus)

Tivoli Business Service Manager: TBSM delivers the real-time information that you need in order to respond to alerts effectively and in line with business requirements, and optionally to meet service-level agreements (SLAs).

IBM Cloud Orchestrator High Availability and End-to-end Automation Solution

Comments

1-1 of 1 Previous | Next

glotti commented on February 21, 2014 Permalink

Not much :-)

Show 10 | 25 | 50 items per page Previous | Next