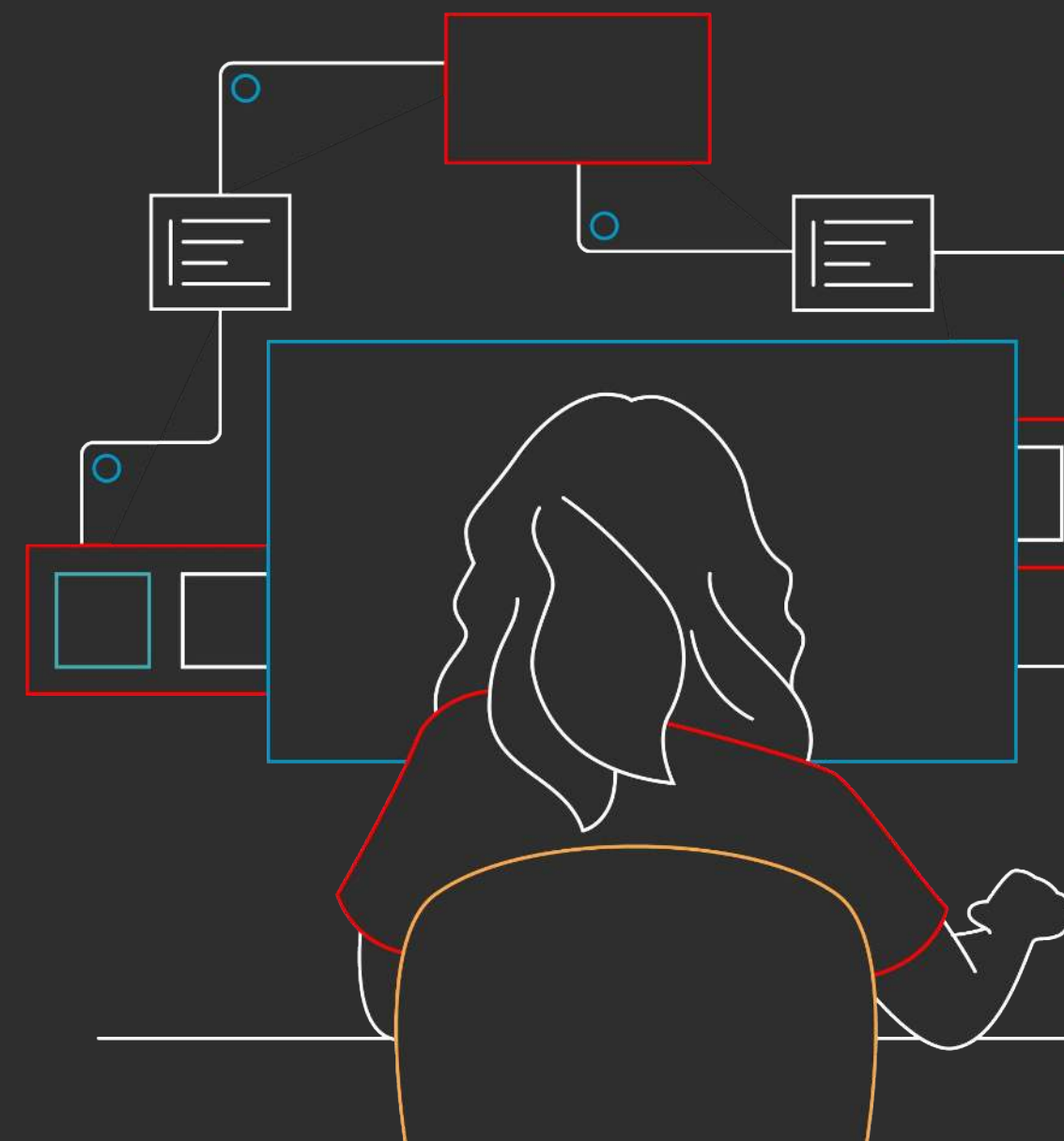


Operation, Update, and **Monitoring**

Updating SAP HANA Operating System and
the cluster



General Overview of Update Procedures

Rolling updates

- remove one node from the fully formed active cluster
 - move all services and stop the daemons
 - remove node from cluster
- update the OS software
- rejoin the node into the cluster and wait until the cluster is fully functional
- repeat these steps with each remaining node

General Overview of Update Procedures

Entire Cluster Update At Once

- Stop and shutdown the whole cluster
- update the OS software on all nodes
- restart the cluster as whole

Rolling Update Commands Overview

<code>pcs resource disable <i>resource</i></code>	Stop a running resource (group) entirely and to prevent the cluster from starting it again
<code>pcs resource move <i>resource</i></code>	Move resource to another node and add constraints to prevent to run on the originating node
<code>pcs node standby <i>clusternode</i></code>	Place the chosen node into standby mode
<code>pcs cluster stop <i>clusternode</i></code>	Stop the cluster software on the chosen node
<code>yum -y update</code>	Update the cluster (Optional other software)
<code>pcs cluster disable <i>clusternode</i> reboot</code>	In case a reboot is required after the update make sure to disable the cluster service to avoid auto-rejoin
<code>pcs cluster enable <i>clusternode</i></code>	re-enable the cluster software
<code>pcs cluster start <i>clusternode</i></code>	Rejoin the updated node into the cluster
<code>pcs status</code>	Check that everything appears as it should
<code>pcs node unstandby <i>clusternode</i></code>	re-activate node for service
<code>pcs resource move <i>resource</i></code>	Optional: move resource back to this node
<code>pcs resource enable <i>resource</i></code>	re-start previously stopped resource

Entire Cluster Update Commands Overview

<code>pcs resource disable <i>resource</i></code>	Stop all resources (groups) in proper order (or use <code>--all</code>)
<code>pcs node standby --all</code>	Place all nodes into standby mode
<code>pcs cluster stop --all</code>	Stop the cluster software on all nodes
<code>yum -y update</code>	Update the cluster (Optional other software)
<code><i>sidadm</i>> HDB stop</code>	gracefully shutdown HANA DBs
<code>pcs cluster disable --all reboot</code>	In case a reboot is required after the update make sure to disable the cluster service to avoid auto-rejoin
<code>pcs cluster enable --all</code>	re-enable the cluster software
<code>pcs cluster start --all</code>	Rejoin the updated nodes into the cluster
<code>pcs status</code>	Check that everything appears as it should
<code>pcs node unstandby --all</code>	re-activate all nodes for service
<code>pcs resource enable <i>resource</i></code>	re-start all stopped resources in proper order (or use <code>--all</code>)

Updates for SAP HANA DB or for Applications

Entire Cluster Update At Once

- Check for SAP application support
- Put the cluster in to maintenance mode
pcs property set maintenance-mode=true
Alternatively: Stop the cluster completely
pcs cluster stop --all
pcs cluster disable --all
- perform the update
- Release the maintenance mode:
pcs property set maintenance-mode=false
Alternatively: Re-Start the cluster completely
pcs cluster enable --all
pcs cluster start --all
-



SAP Operations Ansible Collection

Separate subscription required for certified Ansible Collection on Red Hat Automation Hub

https://console.redhat.com/ansible/automation-hub/repo/published/sap/sap_operations/

- ▶ Customer feedback driven
- ▶ Native Ansible modules (not commands) for SAP related operations
- ▶ “Lego blocks” for SAP customer automation use cases
- ▶ Main focus on SAP HANA and SAP NetWeaver ABAP automation
- ▶ Fast development cycle and quick pace of customer feedback implementation
- ▶ Test automation and CI/CD
- ▶ Working with partners and customers to find most valuable SAP Ansible automation use cases
- ▶ Partners and customers are actively contributing to SAP Operations Ansible Collection and creating content for their own use cases
- ▶ Community of partners and customers is governing and leading content development for SAP Operations Ansible collection
- ▶ Documentation for all content is available using ansible-doc command

Community Ansible Content for SAP operations (1 of 2)

Galaxy collection: [sap.sap_operations](https://galaxy.ansible.com/redhat-sap/rh_operations/tree/galaxy)

Repository: github.com/redhat-sap/rh_operations/tree/galaxy

Documentation: docs.galaxy.saponrhel.org

Repository:

[community.sap_operations](https://galaxy.ansible.com/redhat-sap/rh_operations/tree/galaxy)

- [os_ansible_user](#)^R
- [os_knownhosts](#)^R
- [sap_rhsm](#)^R
- [sap_facts](#)^M
- [sap_monitor_hana_status](#)^M
- [sap_monitor_nw_status](#)^M
- [sap_monitor_nw_perf](#)^M
- [sap_monitor_nw_response](#)^M
- [sap_control](#)^R
- [sap_fapolicy](#)^R
- [sap_firewall](#)^R
- [os_etchosts](#)^R
- [sap_profile_update](#)^R

R - Ansible Role
M - Ansible Module
P - Ansible plugin

Roles:

[hana_update](#)

Update SAP HANA instance

[pcs_config_checks](#)

Run cluster configuration checks

[cloudconnector](#)

Manage SAP Cloud Connector

[sapjvm](#)

Manage SAP JVM

[prepare](#)

Prepare RHEL managed host

[hana](#)

Manage SAP HANA instance

[cf](#)

Manage cloud foundry CLI

[sap_kernel_update](#)

Update SAP kernel.

[unpack](#)

Unpack *.zip and *.sar files

Filter plugins:

[any](#) - Return true if any element in the iterable is truthy

[all](#) - Return true if all elements in the iterable are truthy

[pcs_resources_from_status](#) - Get pcs cluster resources lists

[me_enr_from_alias](#) - Return the ENR from alias.

[pcs_cluster_property_mapping](#) - Get pcs cluster property from pcs_cib_info result

[pcs_resources](#) - Get pcs cluster resources lists

[valid_sid](#) - Verify that SAP system ID (SID) is valid.

[license_content](#) - Get sap license attributes from sap license file

[pcs_resources_from_cib](#) - Get pcs resources list from result of sap.sap_operations.pcs_cib_info module

[mount_path](#) - Return the mount path of the filepath.

[combine_default](#) - Combine two dictionaries, second dictionary will be used as default.

[any](#) - Return true if any element in the iterable is truthy.

[valid_sid](#) - Verify that SAP system ID (SID) is valid.

[me_valid_alias](#) - Return true if string provided is valid software alias.

[all](#) - Return true if all elements in the iterable are truthy.

Test plugins:

[any](#)

Return true if any element in the iterable is truthy.

[valid_sid](#)

Verify that SAP system ID (SID) is valid.

[me_valid_alias](#)

Return true if string provided is valid software alias.

[all](#)

Return true if all elements in the iterable are truthy.

Community Ansible Content for SAP operations (2 of 2)

Galaxy collection: [sap.sap_operations](https://galaxy.ansible.com/redhat-sap/rh_operations)

Repository: github.com/redhat-sap/rh_operations/tree/galaxy

Documentation: docs.galaxy.saponrhel.org

Cloud Foundry plugins (SAP BTP):

[cf_service_instances_info](#)

Fetch information about Cloud Foundry service instances

[cf_service_instance_info](#)

Fetch information about Cloud Foundry service instance

[cf_marketplace_info](#)

Fetches Cloud Foundry marketplace service offerings

[cf_service_plans_info](#)

Fetch information about Cloud Foundry service plans

[cf_service_instance](#)

Manage Cloud Foundry service instances

[cf_spaces_info](#)

Fetch information about Cloud Foundry spaces

Cluster and HA Modules:

[pcs_capabilities_info](#)

Get pacemaker capabilities

[pcs_cib_info](#)

Get pacemaker CIB (configuration information base) information

[ha_check_failoverconfig_info](#)

Run sap host agent function

HACheckFailoverConfig

[ha_get_failoverconfig_info](#)

Run sap host agent function HAGetFailoverConfig

[ha_check_config_info](#)

Run sap host agent function HACheckConfig

[pcs_status_info](#)

Get pacemaker status information

[pcs_fence_azure_arm_list_info](#)

Get list of hosts from Azure fence ARM

ABAP, HANA Modules:

[hana_backup](#)

Create SAP HANA database file backup

[abap_transport_info](#)

Fetch information about transport request from SAP ABAP system

[hana_restore](#)

Restore SAP HANA database backup

[abap_transport_create](#)

Create transport request in SAP ABAP system

[service](#)

Manage SAP HANA services

[abap_system_info](#)

Fetch SAP ABAP system information

[hdbuserstore](#)

Manage the HANA user store (HANA command hdbuserstore)

[abap_transport_import](#)

Import transport request

[abap_transports_info](#)

Fetch transport requests from SAP ABAP system based on search criteria

[hdbuserstore_info](#)

Get information from HANA user store (HANA command hdbuserstore)

SAP host Modules:

[saplikey_show_info](#)

Get information about SAP license keys with saplikey program

[rolling_kernel_switch](#)

Manage Rolling kernel switch

[host_info](#)

Collect information about installed SAP instances on the host

[system](#)

Manage SAP system

[saplikey](#)

Manage sap license keys for SAP application instance with saplikey program.

[parameter_info](#)

Parameter information

[service](#)

Manage SAP HANA services

[saplikey_get_info](#)

Get host hardware key information and other useful information with saplikey program.

[system_info](#)

SAP system information

Other modules:

[proc_info](#)

Reads content from files in the /proc directory

[azure_imds](#)

Get Azure instance metadata

[pcs_fence_azure_arm_list_info](#)

Get list of hosts from Azure fence ARM

SAP for me modules:

[me_downloadbasket_info](#)

Fetch information from SAP software center download basket

[me_file_info](#) - File information

[me_downloaditemset_info](#)

Fetch information from SAP software download center

Red Hat Supported Ansible Content for SAP operations

AAP collection: [sap.sap_operations](#)¹

Repository: [redhat-sap/rh_operations](#)

Roles:

- [hana_update](#) - Update SAP HANA instance
- [sap_kernel_update](#) - Update SAP kernel.
- [unpack](#) - Unpack *.zip and *.sar files
- [prepare](#) - Prepare RHEL managed host to run content from sap.sap_operations
- [sapjvm](#) - Manage SAP JVM

Filter plugins:

- [valid_sid](#) - Return true if SAP system ID is valid

Filter plugins:

- [valid_sid](#) - Verify that SAP system ID (SID) is valid

Modules:

- [abap_system_info](#) - Fetch SAP ABAP system information
- [system](#) - Manage SAP system
- [service](#) - Manage SAP HANA services
- [rolling_kernel_switch](#) - Manage Rolling kernel switch
- [host_info](#) - Collect information about installed SAP instances on the host
- [parameter_info](#) - Parameter information
- [system_info](#) - SAP system information
- [hana_backup](#) - Create SAP HANA database file backup
- [hdbuserstore_info](#) - Get information from HANA user store (HANA command hdbuserstore)
- [hdbuserstore](#) - Manage the HANA user store (HANA command hdbuserstore)
- [ha_get_failoverconfig_info](#) - Run sap host agent function HAGetFailoverConfig
- [ha_check_failoverconfig_info](#) - Run sap host agent function HACheckFailoverConfig
- [ha_check_config_info](#) - Run sap host agent function HACheckConfig

¹ - Available via Red Hat Automation Hub with Ansible Automation Platform (AAP) subscription



Thank You !



<https://linkedin.com/company/Red-Hat>



<https://facebook.com/RedHatinc>



<https://youtube.com/user/RedHatVideos>



<https://twitter.com/RedHat>

