Upgrade from RHEL 7 to RHEL8

RHEL 7 to RHEL 8

Scope

The scope of this document is to upgrade OS from RHEL 7 to RHEL 8 on both physical and virtual servers. Databases and VCS yet to be tested.

Prerequisites / Checks

- □ Generally, RHEL 7 is not very different from RHEL 8 in terms of volume sizes. Ensure there's no OS filesystem with a high usage. Perform cleanup and/or increase if needed.
- □ A backup should be available (split mirror, VM snapshot, Rear or LVM snapshots).
- □ If you are upgrading a clone of the RH7 VM (original VM renamed to "hostname-preuplift", clone named "hostname") make sure you retired avamar of the old container first and onboard again the new container. See Rh67 instruction for details.
- □ Execute CRQ script. (/root/CRQ/collect_info.ksh) before any other change. Grab any information that can be useful.
- □ Check SAT6 connectivity status.
- □ If it's a VM, the virtual SCSI adapter should not be LSI, but Paravirtualized. Support from Cloud & VMWare will be needed - please ensure you have it. Steps described below.
- □ Again, it it's a VM, the NICs should be of type vmxnet3, not e1000. Steps described below
- □ A preventive reboot should be performed before any other change (except the backup method)
- □ There should be no NICs named ethX. Rename them to netX
- □ Access to Satellite 6, to change an assigned content view.

Use always copy/paste direct to the shell for all commands below and DO NOT copy/paste into a local file/script to execute as that would lead to issues!!!!

General preparation in front. Ensure the prerequisites / checks are met, especially a valid backup!

split OS disk mirror - in case of **physical** server!!

Split mirror

```
#Root mirror split (for HP G8 and above)
    SSACLI=`ls -1 /usr/sbin/ssacli /usr/sbin/hpssacli /usr/sbin/hpacucli 2>/dev/null | tail -1`
    #check status of array:
    $SSACLI ctrl all show status
    #check for volume
    $SSACLI ctrl all show config

    # if all ok, then split
    sync
    echo 3 > /proc/sys/vm/drop_caches
    #get Slot and Array from "check for volume" point above
    $SSACLI controller slot=0 array A splitmirror action=splitwithbackup
    $SSACLI ctrl all show config

# remove rhel6 rpms
# check for hp rpms and have a look that only HP rpms will be included in the removal:
rpm -qa |grep ^hp -i|grep -v HPOv
# in case only HPE rpms were displayed above, remove hp rpms:
yum remove $(rpm -qa |grep ^hp -i|grep -v HPOv)
```

Preparation work - VMs only

```
# Backup the current root password:
grep ^root: /etc/shadow > /root/root.shadow

# Set the password temporarily to Vodafone123
usermod -p '$1$s8WUQ6kW$9BMztZnKlaIbPIL478o8z0' root

# Execute CRQ script
/root/CRQ/collect_info.ksh

# If it's a VM, check if there are LSI adapters:

lsmod | egrep "mptspi|mptbase"
# if it shows any of those drivers, do this and after the next shutdown below, follow the steps along with the
Cloud & VMware team

echo 'add_drivers+="vmw_pvscsi "'  > /etc/dracut.conf.d/pvscsi.conf
dracut -f


# For VMs, also check if there's any NIC of type e1000:

for f in `ip addr show | grep -v " lo" | grep  mtu | cut -f 2 -d:`; do ethtool -i $f | grep driver; done  |
grep e1000
# if there's any output, also follow the steps below along with the Cloud & VMware team


# So, now, if there's any SCSI LSI adapter and/or e1000 NICs:

systemctl poweroff

# Change the SCSI adapter type on the vCenter to Paravirtualized  -> Cloud & VMware
# Change the NIC types from e1000 to vmxnet, keeping the MAC addesses -> Cloud & VMware
# Power on the VM and re-check using the lsmod and for cycle above.
```

Setup repositories and update server to the latest packages. Also install leapp and run the assessment:

Updating to latest

```
# Setup repositories, set the Release [ currently it's 7.9 ]:

subscription-manager release --set=7.9
subscription-manager refresh
yum clean all
rm -rf /var/cache/yum
subscription-manager repos --disable="*" --enable=rhel-7-server-extras-rpms --enable=rhel-7-server-optional-
rpms --enable=rhel-7-server-rpms --enable=rhel-7-server-satellite-tools-6.8-rpms --enable=rhel-7-server-
supplementary-rpms --enable=Vodafone_vf-custom_vf-custom-rhel7
# Update everything to the latest package versions:
yum -y update

# Comment out all NFS entries within the /etc/fstab
sed -i '/\snfs\s/s/^/#Upg78-/g' /etc/fstab ; grep nfs /etc/fstab

# If updates were applied, reboot:
systemctl reboot

# While it reboots, you can change the content view assigned to the host on Satellite 6 to cv-vf-redhat-rhel7-
to-rhel8 ( not covered here ).
```

```
# remove all kernels expect the latest currently loaded kernel
yum -y install yum-utils; package-cleanup -y --oldkernels --count=1

# Refresh subscription and repository data on the server, set the release:
subscription-manager refresh
subscription-manager release --set=7.9
yum clean all
rm -rf /var/cache/yum/
yum repolist all

# Disable any additional repository ( especially EPEL ).  Ensure only the base, extras, optional, supplementary
and latest satellite tools repositories, if they're not already enabled:
subscription-manager repos --disable="*" --enable=rhel-7-server-extras-rpms --enable=rhel-7-server-rpms

# Install leapp
yum -y install leapp-repository leapp-repository-deps leapp python2-leapp

# Remove the rescue image from grub2 due to possible space constraints:
rm -f /boot/*rescue*
grubby --remove-kernel=/boot`grep rescue /boot/grub2/grub.cfg | grep linux16 | awk {' print $2 '}`
# Confirm ( should be no output ):
grep rescue /boot/grub2/grub.cfg | grep linux16 | awk {' print $2 '}
# Redo the config
grub2-mkconfig -o /boot/grub2/grub.cfg

# remove the btrfs module, if loaded
rmmod btrfs pata_acpi floppy

# Remove any existing yum versionlock ( if versionlock is not installed, ignore the error ), check ( and ensure
) locale is set to en_US.UTF-8. Take note of previous versionlocks.
yum versionlock clear
cp /etc/locale.conf /etc/locale.conf_pre_leapp ; echo 'LANG="en_US.UTF-8"' > /etc/locale.conf
cd /etc/leapp/files/ && curl https://`subscription-manager config|  grep baseurl | cut -f3 -d/`/pub/vodafone
/software/RHEL8/upgrade_files/leap-data-latest.tar.gz | tar -xz && cd -

# Fix common leapp inhibitors:
sed -i 's/^MACs/#MACs/g' /etc/ssh/sshd_config

# check /var, it should have at least min. 2G free space left
# if less available, then extend by:
lvextend -r -L +2G /dev/vg0/varvol

# Try to pre-fill the answerfile:
echo "[remove_pam_pkcs11_module_check]" >> /var/log/leapp/answerfile
echo "confirm = True" >> /var/log/leapp/answerfile
echo "[authselect_check]" >> /var/log/leapp/answerfile
echo "authselect select sssd with-faillock with-fingerprint with-mkhomedir --force = None" >> /var/log/leapp
/answerfile
echo "confirm = True" >> /var/log/leapp/answerfile

# Run the pre-upgrade assessment:
leapp preupgrade

# If needed, answer 2 common requirements from leapp:
leapp answer --section authselect_check.confirm=True
leapp answer --section remove_pam_pkcs11_module_check.confirm=True

# Look at the output and see if there are any inhibitors. Check for further details on the log files: /var/log
/leapp/leapp-report.txt  and /var/log/leapp/leapp-preupgrade.log
# Fix any inhibitors if they exist, and re-run, if so, until the output is clean:
leapp preupgrade

# NOTE: If it fails with an error "unable to use yum successfully", and *ONLY* in that case, perform this
workaround:
vi /usr/share/leapp-repository/repositories/system_upgrade/el7toel8/libraries/rhsm.py
# Change line 162    from "cmd = ['yum', 'clean', 'all']"   to  "cmd = ['yum', 'repolist', 'all']"
yum clean all
# Retry leapp preupgrade
leapp preupgrade
```

Once all issues are fixed, it's time to start the actual upgrade

Upgrade

```
leapp upgrade

#If you encounter dependency issues, please double check if the conflicting packages are installed from EPEL or other external repository and eventually remove them (e.g. openssl11 from RH7 EPEL)

# It should run for some time, a lot of output involving rpms should appear. In the end, if it went well, a message requesting a reboot should appear.
# Important: grab the console to see the output.
systemctl reboot


# Check the output on the console. A lot of output involving rpms and configurations should appear. The server should reboot once or twice, until the message that it's finished comes up on the console. The server should be reachable
```

Post-upgrade tasks

Post-upgrade checks

```
cat /etc/redhat-release
uname -a
# It should report 8.4 as release, and a RHEL 8.4 kernel
# Red Hat Enterprise Linux release 8.4 (Ootpa)
# 4.18.0-193.14.3.el8_2.x86_64
# On Satellite, change the assigned content view to: ccv-vf-redhat-rhel8 and a matching host group ( not shown here )
# Refresh and clean old repository data
# NOTE: yum has been superseded by dnf


subscription-manager refresh
subscription-manager release --set=8.4
dnf clean all
rm -rf /var/cache/yum/ /var/cache/dnf/


# Enable satellite tools and custom software VF:
subscription-manager repos --disable="*" --enable=Vodafone_vf-custom_vf-custom-rhel8 --enable=satellite-tools-6.8-for-rhel-8-x86_64-rpms --enable=rhel-8-for-x86_64-appstream-eus-rpms --enable=rhel-8-for-x86_64-baseos-eus-rpms --enable=rhel-8-for-x86_64-supplementary-eus-rpms

# Update to the latest packages:
dnf -y update

# update against epel:
subscription-manager repos --enable Vodafone_vf-fedora_vf-epel-rhel8
dnf update -y
subscription-manager repos --disable Vodafone_vf-fedora_vf-epel-rhel8
subscription-manager refresh
dnf clean all
rm -rf /var/cache/dnf

# check if remaining el7 packages which are not related to VF or Red Hat
rpm -qa --qf '%{NAME}-%{VERSION}-%{RELEASE}.%{ARCH} %{VENDOR}\n'|grep -i el7|egrep -v 'Red Hat|patchmgt|HPOv|VFGO-keylogin|BladeLogic|perl-XML-Smart|python2-qpid-proton|hp-ams|HP-CNA-FC-Emulex-Enablement-Kit|sum-|hp-health|hp-snmp-agents|qpid-proton-c|iscsiuio|htop|perl-REST-Client|perl-Object-MultiType|^kmod'

# if other vendor packages are installed, please push them to the customer or team who is managing them to
```

```
upgrade them to latest RHEL 8 rpm version


# Remove leapp packages, old RHEL7 kernel and X packages ( last one optional ):
rpm -e --nodeps $(rpm -qa | grep leapp)
dnf remove "kernel*.el7.*" xorg*

# Remove older el7 packages. Please check manually from the list and confirm if appropriate:
dnf remove "*el7*"

# verify if any el7 rpms are still installed
rpm -qa|grep -i el7

# check for old kernel
ls -l /boot/loader/entries/*el7*
# in case any el7 entry is still there, then remove by (replace "vmlinuz-3.10.0-1160.15.2.el7.x86_64" by the
one which shows up above):
# grubby --remove-kernel=/boot/vmlinuz-3.10.0-1160.15.2.el7.x86_64
# also:
rm -f /boot/*el7*
rm -f /boot/loader/entries/*el7*

#Make sure that the default gateway is being set in /etc/sysconfig/network-scripts/ifcfg-net1 and NOT in /etc
/sysconfig/network. Otherwise you'll end up with default route being set twice or more.

# Uninstall legacy "network-scripts" package. Install NetworkManager package. Enable and start NetworkManager.
dnf -y remove network-scripts
dnf -y install NetworkManager
systemctl enable --now NetworkManager

# Check with nmcli
nmcli conn show

# Reconfigure the kdump volume:
dnf -y install xfsprogs
umount /var/crash
lvrename /dev/vg0/crashvol /dev/vg0/varcrashvol
mkfs -t xfs -f /dev/vg0/varcrashvol
sed -i "s|vg0-crashvol|vg0-varcrashvol|g; s|vg0/crashvol|vg0/varcrashvol|g; s|/var/crash.*ext4|/var
/crash\txfs|g " /etc/fstab
mount /var/crash


# Reconfigure the /tmp filesystem as tmpfs, if needed

grep -q "^tmpfs.* /tmp" /etc/fstab || ( sed -i "s|^/dev/mapper/vg0-tmpvol|#/dev/mapper/vg0-tmpvol|g; s|^/dev/vg0
/tmpvol|#/dev/vg0/tmpvol|g" /etc/fstab ;\
echo "tmpfs   /tmp    tmpfs   size=2048m,nodev,nosuid,noexec  0       0"  >> /etc/fstab )

# Reboot. If possible, check the console for errors:
systemctl reboot

# Also reconfigure /var/tmp as separate filesystem
lvrename /dev/vg0/tmpvol vartmpvol && mkfs -t xfs -f /dev/vg0/vartmpvol && echo "/dev/mapper/vg0-vartmpvol /var
/tmp xfs nodev,nosuid 0 0" >> /etc/fstab
mount /dev/vg0/vartmpvol /mnt
cp -avRi /var/tmp/* /mnt/
umount /mnt
rm -rf /var/tmp/*
mount /var/tmp

# Reboot again ( only if mount failed for some reason )
systemctl reboot

# Reinstall puppet and place back the configuration
dnf -y install puppet
test `grep -c "^server" /etc/puppetlabs/puppet/puppet.conf` = "1" || cat /etc/puppetlabs/puppet/puppet.conf.
rpmsave > /etc/puppetlabs/puppet/puppet.conf

# Remove overwrite for the below puppet SCP from the web GUI. Will be automated later.
vfgosshd > vfgosshd_sshd_config_hostkey
```

```
vfgosshrootkeys > activate_build_key

# Uninstall the BladeLogic_RSCD_Agent package. The agent is no more propperly working otherwise. (Puppet message "Vfgoblade/Service[rscd]/enable" doesn't disappear as example.)
yum -y remove BladeLogic_RSCD_Agent

# Review the outstanding changes and if it's okay you can apply all at once or one by one. The return code must be 0 in the end. (No more changes may be applied.)
/opt/puppetlabs/puppet/bin/puppet agent -t && echo "SUCCESS." || echo "Failed or changes applied. Fix and rerun until you get 'SUCCESS'."

# Create new ssh host key files.
rm -f /etc/ssh/ssh_host_*
systemctl restart sshd

#Hint: If you cannot get rid of an issue with sssd, remove the sssd cache and run puppet again:
rm -f /var/lib/sss/db/*

#Hint: If you can't get rid of an issue with the Vfgocatrust puppet module, remove these two files and try again:
rm -f /etc/pki/ca-trust/source/anchors/ldapca.crt /etc/pki/ca-trust/source/anchors/VodafoneInternalRootCA.crt

# update to 8.4
# NOTE: If upgrading then to RHEL 8.4 or a non-EUS release, switch away from EUS repositories
subscription-manager repos --disable="*" --enable=Vodafone_vf-custom_vf-custom-rhel8 --enable=satellite-tools-6.8-for-rhel-8-x86_64-rpms --enable=rhel-8-for-x86_64-appstream-rpms --enable=rhel-8-for-x86_64-baseos-rpms --enable=rhel-8-for-x86_64-supplementary-rpms
subscription-manager release --set 8.4
subscription-manager refresh;dnf clean all
dnf update -y
subscription-manager repos --enable Vodafone_vf-fedora_vf-epel-rhel8
dnf update -y
subscription-manager repos --disable Vodafone_vf-fedora_vf-epel-rhel8
subscription-manager refresh;dnf clean all;rm -rf /var/cache/dnf

#ATTENTION: At this point please double check that IDM is working properly including sudo rules. You might loose root-access after reboot otherwise. If it's broken, fix it on satellite (enable vfgoidm_active and configure it properly, then re-run puppet)

systemctl reboot
```

Check and prepare for WAO packages

```
# Verify whether or not any WAO rpm's are installed
rpm -qa | grep ^WAO
# If there is no output and you do not plan to install any WAO packages, skip the remaining part of this codeblock
# Do NOT uninstall any WAO packages. Upgrade them like any other package (Don't care if the version changed) once you enabled the WAO repository via
subscription-manager repos --enable Vodafone_vf-wao_vf-wao-rhel8
yum update -y WAO*
# Only remove WAO-nghttp2 if present
yum erase -y WAO-nghttp2
```

```
# You need to add the host to the "hg-vf-prod/rhel8/wao" host group. Either via web interface on
https://vosat6avr.dc-ratingen.de or by running the below hammer command on vosat6avr
hammer host update --name=<fqdn-hostname> --hostgroup-id 150


# WAO has some puppet modules which will fail if the lvs are not following a certain naming convention and
are not formatted with the xfs file system, eventually we need to rename lvs for /opt/SP and /var/SP

# Attention: This has not been extensivly tested yet.
OPTSPDEVICE=$(awk '/\/opt\/SP/ {print $1}' /etc/fstab | awk 'BEGIN {FS="/"} {print $NF}' | sed 's/vg1-//g')
if [ "$OPTSPDEVICE" != "optspvol" ] && [ "x$OPTSPDEVICE" != "x" ]
then
    lvrename vg1 $OPTSPDEVICE optspvol
    sed -Ei "s#.+vg1.+$OPTSPDEVICE(.+)#/dev/mapper/vg1-optspvol\1#g" /etc/fstab
    umount /opt/SP && mount /opt/SP || echo "unable to umount /opt/SP - reboot required"
else
    echo "$OPTSPDEVICE already matches naming convention"
fi


VARSPDEVICE=$(awk '/\/var\/SP/ {print $1}' /etc/fstab | awk 'BEGIN {FS="/"} {print $NF}' | sed 's/vg1-//g')
if [ "$VARSPDEVICE" != "varspvol" ] && [ "x$VARSPDEVICE" != "x" ]
then
    lvrename vg1 $VARSPDEVICE varspvol
    sed -Ei "s#.+vg1.+$VARSPDEVICE(.+)#/dev/mapper/vg1-varspvol\1#g" /etc/fstab
    umount /var/SP && mount /var/SP || echo "unable to umount /opt/SP - reboot required"
else
    echo "$VARSPDEVICE already matches naming convention"
fi
```

Migrate `/opt/SP` and `/var/SP` to `xfs`
You need to have some temporary storage available, big enough to hold the data stored on `/opt/SP` respectively `/var/SP`. The code example below uses `/var/tmp/` another good candidate with even more space is `/var/crash`. If there is no partition with enough free space (and you haven't enough free space on any volume group) either store the data elsewhere or attach temporary storage.

```
#Migrate /opt/SP to xfs:
mkdir /var/tmp/opt_SP
shopt -s dotglob
cp -av /opt/SP/* /var/tmp/opt_SP/
umount /opt/SP
mkfs.xfs -f /dev/mapper/vg1-optspvol
sed -Ei 's#(^.+optspvol.+)ext4(.+$)#\1xfs\2#g' /etc/fstab
mount /opt/SP
cp -av /var/tmp/opt_SP/* /opt/SP/
rm -rf /var/tmp/opt_SP/

#Migrate /var/SP to xfs:
mkdir /var/tmp/var_SP
shopt -s dotglob
cp -av /var/SP/* /var/tmp/var_SP/
umount /var/SP
mkfs.xfs -f /dev/mapper/vg1-varspvol
sed -Ei 's#(^.+varspvol.+)ext4(.+$)#\1xfs\2#g' /etc/fstab
mount /var/SP
cp -av /var/tmp/var_SP/* /var/SP/
rm -rf /var/tmp/var_SP/
```

```
#Run puppet again:
/opt/puppetlabs/puppet/bin/puppet agent -t && echo "SUCCESS." || echo "Failed or changes applied. Fix and
rerun until you get 'SUCCESS'."
```

Upgrade hardware firmware and install hptools - **SKIP** this point for **virtual** server

# /etc/sudoers

```
## execute only for Gen8 or higher:
dmidecode -t system|grep Product | egrep "Gen"

# drop off the console and close the ssh session to ILO, if FW upgrade for ILO will be applied, because it will
timeout your session.

mount -o remount,exec /tmp
cd /var/tmp
SAT=`grep -w ^hostname /etc/rhsm/rhsm.conf | awk '{print $3}'`
wget https://$SAT/pub/vodafone/software/HPE_SCRIPT/new-install-hp.sh
chmod 744 new-install-hp.sh
./new-install-hp.sh
rm -f /var/tmp/new-install-hp.sh
```

Check, if chrony has a valid sync

```
#check chrony source states
chronyc sources

#if there are no NTP servers configured or others are needed which are different from puppet standards, please
reconfigure them in the satellite GUI

#if all NTP servers are configured properly, but sync is not working due to high difference, sync manually:
chronyc makestep
```

Re-install Backup rpm packages on **Physical servers** after upgrade

Re-install backup rpm packages on Physical server after RHEL upgrade (Before starting this you need to have backup rpm packages)

```
# Check existing installed backup rpm packages, if following packages are not installed, then skip this step
rpm -qa|grep -i lgto
lgtonode-18.2.0.2-1.x86_64
lgtoclnt-18.2.0.2-1.x86_64

# get the SAME VERSION of backup RPMs on toledo8:
ssh toledo8
ls -la /sls_backup/NETWORKER_INSTALL/182*
# copy SAME VERSION to your server
# e.g.:
# from toledo8 (if no route exists, please copy through jumpserver)
scp /sls_backup/NETWORKER_INSTALL/18204/nw18204_linux_x86_64.tar.gz vgnp04hr:/var/tmp/

# if oracle RAC is active then:
#
# scp /sls_backup/NETWORKER_INSTALL/18204/NMDA/nmda18204_linux_x86_64.tar.gz vgnp04hr:/var/tmp/

# remove all backup packages which were discovered in the first rpm -qa cmd (client, node, etc..)
rpm -e lgtoclnt-18.2.0.2-1.x86_64 lgtonode-18.2.0.2-1.x86_64

# Re-install those packages from backup rpm bundle(Backup rpm bundle need to check with Backup team)
cd /var/tmp/
tar -xf nw18204_linux_x86_64.tar.gz
cd linux_x86_64/
rpm -ivh --relocate /usr=/opt/networker lgtoclnt-18.2.0.4-1.x86_64.rpm lgtonode-18.2.0.4-1.x86_64.rpm

# if oracle RAC is active then:
#
# cd /var/tmp/
# tar -xf nmda18204_linux_x86_64.tar.gz
# rpm -ivh --relocate /usr=/opt/networker lgtonmda-18.2.0.4-1.x86_64.rpm

cd /root
rm -f /var/tmp/nw18204_linux_x86_64.tar.gz /var/tmp/lgtonmda-18.2.0.4-1.x86_64.rpm /var/tmp
/nmda18204_linux_x86_64.tar.gz
rm -rf /var/tmp/linux_x86_64/

# Enable, start networker backup service and check status
systemctl enable networker
systemctl start networker
systemctl status networker

# if oracle RAC is active then execute following nsr posttasks
#
# /opt/networker/sbin/nsroraadmin -r update NSR_NWPATH=/opt/networker/sbin/
# /opt/networker/sbin/nsroraadmin -r list
```

Re-enable NFS mounts

```
# Enable and start some services.
systemctl enable rpcbind.service nfs-client.target nfs-server.service
systemctl start rpcbind.service nfs-client.target nfs-server.service

# mount NFS again
sed -i 's/^#Upg78-//g' /etc/fstab
mount -a

# After upgrade if NAS FS not mounted during boot time add _netdev mount option in fstab file.

# Reboot
systemctl reboot
```

Enable sysstat

Check for failed services

```
systemctl enable --now sysstat.service
systemctl enable --now sysstat-collect.timer
systemctl enable --now sysstat-summary.timer
```

Check for failed services

Check for failed services

```
# It should not be "degraded"
systemctl is-system-running
# If the system is in degraded state, check which services are failing and fix or deactivate if not needed
systemctl --failed
```

Reset the root password to previous password

```
pw="`cat /root/root.shadow | cut -d: -f2`" ; usermod -p "$pw" root
```

Post clean up

```
# clean up boot
dnf remove -y --oldinstallonly --setopt installonly_limit=1 kernel

dnf clean all
rm -rf /var/cache/dnf/*
rm -f /root/root.shadow*
/root/CRQ/collect_info.ksh
```

After customer confirmed (mostly 1 week after activity), that the app is running fine, then the project manager will send out a request to the implementer for:

- FOR VIRTUAL :
  - VMware Team should remove the source VM and relabel the clone to proper naming like source server on VMware level
- FOR PHYSICAL:
  - execute following

    ```
    #Root mirror remirror (for HP G8 and above)
            SSACLI=`ls -1 /usr/sbin/ssacli /usr/sbin/hpssacli /usr/sbin/hpacucli 2>/dev/null | tail -1`
        #check for volume
        $SSACLI ctrl all show config
        #get Slot and Array from "check for volume" point above
        $SSACLI controller slot=0 Array A splitmirror action=remirror
            #check remirror status
        $SSACLI ctrl all show config
    ```