* Created by [Kishore Kumar (ext.)](https://atc.bmwgroup.net/confluence/display/~qxz0crd), last modified on [08 Apr 2020](https://atc.bmwgroup.net/confluence/pages/diffpagesbyversion.action?pageId=520964241&selectedPageVersions=1&selectedPageVersions=2)

| Linux LVM Problems and Solutions |
| --- |

LVM: read failed after 0 of 4096

**Problem**  
  
**NOTE!** This only applies to "Logical Volume" errors, **NOT** block devices such as /dev/xvdb /dev/sda5 etc. If uncertain, then please **contact technical expert**!

When running LVM commands such as pvs, vgs, lvs:

lpmodthk05:~ # pvs

/dev/dgpmodthkapps/lv01: read failed after 0 of 4096 at 2718286413824: Input/output error

/dev/dgpmodthkapps/lv01: read failed after 0 of 4096 at 2718286471168: Input/output error

PV VG Fmt Attr PSize PFree

/dev/xvdb dgpmodthkapps lvm2 a- 506.32g 0

/dev/xvdc dgpmodthkapps lvm2 a- 506.32g 0

..

lpmodthk05:~ # vgs

/dev/dgpmodthkapps/lv01: read failed after 0 of 4096 at 2718286413824: Input/output error

/dev/dgpmodthkapps/lv01: read failed after 0 of 4096 at 2718286471168: Input/output error

VG #PV #LV #SN Attr VSize VFree

dgpmodthkapps 5 1 0 wz--n- 2.47t 0

dgswap 1 1 0 wz--n- 14.06g 13.06g

Here, the LV /dev/dgpmodthkapps/lv01 has problems - most probably due to LUNs being removed without a clean shutdown or mount, or detaching disks from logical volumes.

**Solution**  
  
The steps involved in repairing:  
**1)** Find the affected VG: run vgscan  
**2)** Find the LV's attached to the VG  
**3)** Deactivate the LVs (lvchange -an LV)

lpmodthk05:~ # lvchange -an /dev/dgpmodthkapps/lv01

/dev/dgpmodthkapps/lv01: read failed after 0 of 4096 at 2718286413824: Input/output error

/dev/dgpmodthkapps/lv01: read failed after 0 of 4096 at 2718286471168: Input/output error

**4)** Deactivate the VG (vgchange -an VG)

lpmodthk05:~ # vgchange -an dgpmodthkapps

0 logical volume(s) in volume group "dgpmodthkapps" now active

**5)** Activate the VG (vgchange -ay VG)

lpmodthk05:~ # vgchange -ay dgpmodthkapps

1 logical volume(s) in volume group "dgpmodthkapps" now active

**6)** Scan for LVs

lpmodthk05:~ # lvscan

ACTIVE '/dev/dgswap/lvswap01' [1.00 GiB] inherit

ACTIVE '/dev/dgpmodthkapps/lv01' [2.47 TiB] inherit

**7)** Activate the LVs (lvchange -ay LV)

lpmodthk05:~ # lvchange -ay /dev/dgpmodthkapps/lv01

**8)** A quick check shows all is now OK. **NOTE:** if this **fails**, then please **contact a technical expert**

lpmodthk05:~ # pvs

PV VG Fmt Attr PSize PFree

/dev/xvdb dgpmodthkapps lvm2 a- 506.32g 0

/dev/xvdc dgpmodthkapps lvm2 a- 506.32g 0

..

lpmodthk05:~ # vgs

VG #PV #LV #SN Attr VSize VFree

dgpmodthkapps 5 1 0 wz--n- 2.47t 0

dgswap 1 1 0 wz--n- 14.06g 13.06g

lpmodthk05:~ # lvs

LV VG Attr LSize Origin Snap% Move Log Copy% Convert

lv01 dgpmodthkapps -wi-ao 2.47t

lvswap01 dgswap -wi-ao 1.00g

**9)** Now you can go ahead and re-mount the volume.

vgdisplay command hangs

**Solution**

$ vgssan -v --- /sbin/vgscan ...?

xfs\_buf\_find: Block out of range

If xfs\_growfs fails with "Killed" and messages like

\_xfs\_buf\_find: Block out of range:

BUG: unable to handle kernel NULL pointer dereference "

Upgrade the kernel:

> On 29.11.2013 12:54, Poeschl Andreas, FG-940DG wrote:

> Der 3.0.93er Kernel hat aber ein großes Problem: XFS-Filesysteme lassen sich nicht erweitern.

> Dies trifft uns insbesondere in VMs hart. Dafür gibt es nun den 3.0.101er Kernel, der

> auch dieses Problem behebt ...

XFS NOT full but "No space left on device"

**PROBLEM:**after a filesystem extension the cusomer reports that the message "No space left on device" is received when trying to write to the filesystem

lpspluind3b:/global/cold-index # df -hP .

Filesystem Size Used Avail Use% Mounted on

/dev/mapper/dgspluind3ci-lv01 30T 22T 8.5T 72% /global/cold-index

lpspluind3b:/global/cold-index # touch bla

touch: cannot touch `bla': No space left on device

**SOLUTION:** modify /etc/fstab: change options from default to inode64, and remount the filesystem

lpspluind3b:~ # egrep cold-index /etc/fstab

/dev/dgspluind3ci/lv01 /global/cold-index xfs inode64 0 0

lpspluind3b:~ # umount /global/cold-index

lpspluind3b:~ # mount /global/cold-index

lpspluind3b:~ # touch /global/cold-index/bla

lpspluind3b:~ # rm /global/cold-index/bla

lpspluind3b:~ #

**REASON:** see [No space left on device after xfs\_growfs](http://xfs.org/index.php/XFS_FAQ#Q:_Why_do_I_receive_No_space_left_on_device_after_xfs_growfs.3F)

vgdisplay shows LVM Header problem

The error pattern is as follows:

lp34xen05# vgdisplay -v dgstblnet

Using volume group(s) on command line

Finding volume group "dgstblnet"

Wiping cache of LVM-capable devices

Couldn't find device with uuid 'KtKVFt-0m0W-QLaU-HtbT-UmrN-kFNb-cfqCmz'.

Couldn't find all physical volumes for volume group dgstblnet.

PV Name /dev/md66

PV UUID KtKVFt-0m0W-QLaU-HtbT-UmrN-kFNb-cfqCmz

PV Status allocatable

Total PE / Free PE 28811 / 0

**Attention only with aproval of a senior**

The solution (on XEN farms where VM is running or at least one VG running/active). The UUID is taken form the config file you restore from, here /etc/lvm/backup/dgstblnet:

lp34xen05# pvcreate --restorefile /etc/lvm/backup/dgstblnet --uuid 'KtKVFt-0m0W-QLaU-HtbT-UmrN-kFNb-cfqCmz' /dev/md66

Couldn't find device with uuid 'KtKVFt-0m0W-QLaU-HtbT-UmrN-kFNb-cfqCmz'.

Failed to read existing physical volume '/dev/md66'

Physical volume "/dev/md66" successfully created

It is also possible to run that inside a VM (/etc/lvm/backup/dgXXX also exists there), but there the device is named /dev/xvdY - keep that in mind!  
  
Running the above command again should show no error:

lp34xen05# vgdisplay -v dgstblnet

Using volume group(s) on command line

Finding volume group "dgstblnet"

Wiping cache of LVM-capable devices

PV Name /dev/md66

PV UUID KtKVFt-0m0W-QLaU-HtbT-UmrN-kFNb-cfqCmz

PV Status allocatable

Total PE / Free PE 28811 / 0

Volume group vanished/destroyed

We'll see here how a volume group is restored again in case the LVM metadata has been destroyed.

Starting point

For this little tutorial I have created a vg named test\_vg, with a lv named test\_lv, mounted to /test with some files in it.

root@lt1001:/etc/lvm/backup #pvs

PV VG Fmt Attr PSize PFree

/dev/dm-2 test\_vg lvm2 a- 14.07g 0

root@lt1001:/etc/lvm/backup #export PS1='#'

# pvs

PV VG Fmt Attr PSize PFree

/dev/dm-2 test\_vg lvm2 a- 14.07g 0

# vgs

VG #PV #LV #SN Attr VSize VFree

test\_vg 1 1 0 wz--n- 14.07g 0

# lvs

LV VG Attr LSize Origin Snap% Move Log Copy% Convert

test\_lv test\_vg -wi-ao 14.07g

# df -hP /test; ll /test

Filesystem Size Used Avail Use% Mounted on

/dev/mapper/test\_vg-test\_lv 15G 1000M 14G 7% /test

total 0

drwxr-xr-x 8 root root 83 May 31 14:55 var

Destroy metadata

As I do currently not have a destroyed vg, we'll go ahead and do it on our own, just the metadata which makes up the fist 1MB; if you go beyond that point, you also destroy the XFS header and then a restore is most likely quicker:

# dd if=/dev/zero of=/dev/dm-2 bs=1K count=1024

1024+0 records in

1024+0 records out

1048576 bytes (1.0 MB) copied, 0.0547759 s, 19.1 MB/s

# pvs

# vgs

No volume groups found

# lvs

No volume groups found

As now there isn't any metadata any more, all information about the pv, vg and lv is apparently gone. The next point is the one where you'll normally start from:

Get the PV back

In order to recover metadata we need the configuration stored in /etc/lvm/backup/"volumegroup" which is updated every time one changes the lv, vg or pv.  
  
So let's try the first step and get the physical volume back. We need the UUID the physical volume had been created with and write it back to the disk:

# grep -B1 id /etc/lvm/backup/test\_vg

test\_vg {

id = "le26B4-4t20-3k5w-wVoq-L693-wEgV-vuwltn"

--

pv0 {

id = "2iQKIt-kMKH-pEi4-fph2-TE7d-n6fl-TCCGLk"

--

test\_lv {

id = "MR2dbo-oucF-dbUK-dpDR-Zp3s-kIWb-cUGmEu"

Here we need the id for pv0 which we write back to the disk:

# pvcreate --restorefile /etc/lvm/backup/test\_vg -u "2iQKIt-kMKH-pEi4-fph2-TE7d-n6fl-TCCGLk" /dev/dm-2

Couldn't find device with uuid 2iQKIt-kMKH-pEi4-fph2-TE7d-n6fl-TCCGLk.

Can't open /dev/dm-2 exclusively. Mounted filesystem?

OK, interesting, I'm pretty sure that there is nothing mounted, but let's see and try it forcibly again:

# mount |grep test

# pvcreate --restorefile /etc/lvm/backup/test\_vg -u "2iQKIt-kMKH-pEi4-fph2-TE7d-n6fl-TCCGLk" /dev/dm-2 -ff

Couldn't find device with uuid 2iQKIt-kMKH-pEi4-fph2-TE7d-n6fl-TCCGLk.

Can't open /dev/dm-2 exclusively. Mounted filesystem?

Still no success (this is on physical hosts). The problem is, that device mapper has still its hand on the VG and we have to remove that first:

# dmsetup ls

[...]

360060e8016535e000001535e0000002a (253, 17)

360060e8016535c000001535c0000002a (253, 6)

test\_vg-test\_lv (253, 31) <<<<<

360060e8016535e000001535e00000029 (253, 7)

360060e8016535c000001535c00000029 (253, 14)

[...]

# dmsetup remove test\_vg-test\_lv

# dmsetup ls

=> Now device should be gone!

Once done, we should be able to recover:

# pvcreate --restorefile /etc/lvm/backup/test\_vg -u "2iQKIt-kMKH-pEi4-fph2-TE7d-n6fl-TCCGLk" /dev/dm-2 -ff

Couldn't find device with uuid 2iQKIt-kMKH-pEi4-fph2-TE7d-n6fl-TCCGLk.

Physical volume "/dev/dm-2" successfully created

# pvs

PV VG Fmt Attr PSize PFree

/dev/dm-2 lvm2 a- 14.07g 14.07g

OK, so now we have or physical volume back.

Volume group and logical volume

Once we have back the physical volume, let's see if we get the rest by restoring the volume group configuration:

# vgcfgrestore -f /etc/lvm/backup/test\_vg test\_vg

Restored volume group test\_vg

# vgs

VG #PV #LV #SN Attr VSize VFree

test\_vg 1 1 0 wz--n- 14.07g 0

# lvs

LV VG Attr LSize Origin Snap% Move Log Copy% Convert

test\_lv test\_vg -wi--- 14.07g

As you can see, the logical volume is not active so far which means that we can not mount it, so let's do that first:

# vgchange -ay test\_vg

1 logical volume(s) in volume group "test\_vg" now active

# lvs

LV VG Attr LSize Origin Snap% Move Log Copy% Convert

test\_lv test\_vg -wi-a- 14.07g

Now we should be able to mount the file system again.

Bring back file system

As we had some trouble with the volume, it is not unlikely that we do have trouble with the file system, so before mounting, check it:

# xfs\_check -s /dev/mapper/test\_vg-test\_lv

That looks good - in case it does not, try a **xfs\_repair** and then mount it again:

# mount -t xfs /dev/mapper/test\_vg-test\_lv /test

And we're done :-)

Online LUN replacement using LVM Mirror

Sometimes it's required to replace LUNs without downtimes. This howto describes how to do that.  
  
Work Plan:

* add disks see at xen lvm grow
* use one or more luns to mirror existing luns
* detach pvs/ luns & san return

Pro / Cons:  
+ mirroring speed can be controlled  
+ mirroring can be halted or stopped  
+ progress can be monitored  
- requires at least LVM version > v2.02.54 for stablity (according an EMC document)

>> Sample from lt34xendb002, lt34db0008vm - TAS000000380190 / TAS000000412164

# dmesg

[...]

[612704.164007] blkfront: xvdg: flush diskcache: enabled

[612704.167421] xvdg: unknown partition table

[612801.317234] blkfront: xvdh: flush diskcache: enabled

[612801.325368] xvdh: unknown partition table

[613021.129910] blkfront: xvdi: flush diskcache: enabled

[613021.132883] xvdi: unknown partition table

[613203.065475] blkfront: xvdj: flush diskcache: enabled

[613203.068925] xvdj: unknown partition table

[613386.295576] blkfront: xvdk: flush diskcache: enabled

[613386.300956] xvdk: unknown partition table

[613536.703682] BIOS EDD facility v0.16 2004-Jun-25, 0 devices found

[613536.703688] EDD information not available.

root@lt34dbhk0006vm:~# pvcreate /dev/xvdg /dev/xvdh /dev/xvdi /dev/xvdj /dev/xvdk

Physical volume "/dev/xvdg" successfully created

Physical volume "/dev/xvdh" successfully created

Physical volume "/dev/xvdi" successfully created

Physical volume "/dev/xvdj" successfully created

Physical volume "/dev/xvdk" successfully created

root@lt34dbhk0006vm:~# pvs

PV VG Fmt Attr PSize PFree

/dev/xvdb dgtl34dbhk0006vmswap lvm2 a- 14.06g 0

/dev/xvdc dgtl34dbhk0006vmdb lvm2 a- 56.24g 0

/dev/xvdd dgtl34dbhk0006vmdb lvm2 a- 56.24g 0

/dev/xvde dgtl34dbhk0006vmdb lvm2 a- 56.24g 18.71g

/dev/xvdf dgtl34dbhk0006vmbkup lvm2 a- 56.24g 0

/dev/xvdg lvm2 a- 506.32g 506.32g

/dev/xvdh lvm2 a- 506.32g 506.32g

/dev/xvdi lvm2 a- 506.32g 506.32g

/dev/xvdj lvm2 a- 506.32g 506.32g

/dev/xvdk lvm2 a- 506.32g 506.32g

>> Extend the VG:

root@lt34dbhk0006vm:~# vgextend dgtl34dbhk0006vmdb /dev/xvdg

Volume group "dgtl34dbhk0006vmdb" successfully extended

>> View LVs:

root@lt34dbhk0006vm:~# lvs -a -o+devices

LV VG Attr LSize Origin Snap% Move Log Copy% Convert Devices

lvbkup dgtl34dbhk0006vmbkup -wi-ao 56.24g /dev/xvdf(0)

lvdb dgtl34dbhk0006vmdb -wi-ao 150.00g /dev/xvdc(0)

lvdb dgtl34dbhk0006vmdb -wi-ao 150.00g /dev/xvdd(0)

lvdb dgtl34dbhk0006vmdb -wi-ao 150.00g /dev/xvde(0)

lvswap dgtl34dbhk0006vmswap -wi-ao 14.06g /dev/xvdb(0)

>> Mirror LVs:

root@lt34dbhk0006vm:~# lvconvert -b -m1 --corelog dgtl34dbhk0006vmdb/lvdb

Logical volume lvdb converted.

>> Monitor LVs:

root@lt34dbhk0006vm:~# lvs -a -o+devices

LV VG Attr LSize Origin Snap% Move Log Copy% Convert Devices

lvbkup dgtl34dbhk0006vmbkup -wi-ao 56.24g /dev/xvdf(0)

lvdb dgtl34dbhk0006vmdb mwi-ao 150.00g 1.07 lvdb\_mimage\_0(0),lvdb\_mimage\_1(0 )

[lvdb\_mimage\_0] dgtl34dbhk0006vmdb Iwi-ao 150.00g /dev/xvdc(0)

[lvdb\_mimage\_0] dgtl34dbhk0006vmdb Iwi-ao 150.00g /dev/xvdd(0)

[lvdb\_mimage\_0] dgtl34dbhk0006vmdb Iwi-ao 150.00g /dev/xvde(0)

[lvdb\_mimage\_1] dgtl34dbhk0006vmdb Iwi-ao 150.00g /dev/xvdg(0)

lvswap dgtl34dbhk0006vmswap -wi-ao 14.06g

/dev/xvdb(0)

root@lt34dbhk0006vm:~# lvs -a -o+devices

LV VG Attr LSize Origin Snap% Move Log Copy% Convert Devices

lvbkup dgtl34dbhk0006vmbkup -wi-ao 56.24g /dev/xvdf(0)

lvdb dgtl34dbhk0006vmdb mwi-ao 150.00g 5.58 lvdb\_mimage\_0(0),lvdb\_mimage\_1(0)

[lvdb\_mimage\_0] dgtl34dbhk0006vmdb Iwi-ao 150.00g /dev/xvdc(0)

[lvdb\_mimage\_0] dgtl34dbhk0006vmdb Iwi-ao 150.00g /dev/xvdd(0)

[lvdb\_mimage\_0] dgtl34dbhk0006vmdb Iwi-ao 150.00g /dev/xvde(0)

[lvdb\_mimage\_1] dgtl34dbhk0006vmdb Iwi-ao 150.00g /dev/xvdg(0)

lvswap dgtl34dbhk0006vmswap -wi-ao 14.06g /dev/xvdb(0)

>> When the mirror is at 100%, we sync amd detach old LUNs/storage:

root@lt34dbhk0006vm:~# lvs -a -o+devices

LV VG Attr LSize Origin Snap% Move Log Copy% Convert Devices

lvbkup dgtl34dbhk0006vmbkup -wi-ao 56.24g /dev/xvdf(0)

lvdb dgtl34dbhk0006vmdb mwi-ao 150.00g 100.00

lvdb\_mimage\_0(0),lvdb\_mimage\_1(0)

[lvdb\_mimage\_0] dgtl34dbhk0006vmdb iwi-ao 150.00g /dev/xvdc(0)

[lvdb\_mimage\_0] dgtl34dbhk0006vmdb iwi-ao 150.00g /dev/xvdd(0)

[lvdb\_mimage\_0] dgtl34dbhk0006vmdb iwi-ao 150.00g /dev/xvde(0)

[lvdb\_mimage\_1] dgtl34dbhk0006vmdb iwi-ao 150.00g /dev/xvdg(0)

lvswap dgtl34dbhk0006vmswap -wi-ao 14.06g /dev/xvdb(0)

root@lt34dbhk0006vm:~# sync

>> Remove old devices, break the mirror:

root@lt34dbhk0006vm:~# lvconvert -m0 dgtl34dbhk0006vmdb/lvdb /dev/xvdc /dev/xvdd /dev/xvde

Logical volume lvdb converted.

>> View LVs, PVs:

root@lt34dbhk0006vm:~# lvs -a -o+devices

LV VG Attr LSize Origin Snap% Move Log Copy% Convert Devices

lvbkup dgtl34dbhk0006vmbkup -wi-ao 56.24g /dev/xvdf(0)

lvdb dgtl34dbhk0006vmdb -wi-ao 150.00g /dev/xvdg(0)

lvswap dgtl34dbhk0006vmswap -wi-ao 14.06g /dev/xvdb(0)

root@lt34dbhk0006vm:~# pvs

PV VG Fmt Attr PSize PFree

/dev/xvdb dgtl34dbhk0006vmswap lvm2 a- 14.06g 0

/dev/xvdc dgtl34dbhk0006vmdb lvm2 a- 56.24g 56.24g

/dev/xvdd dgtl34dbhk0006vmdb lvm2 a- 56.24g 56.24g

/dev/xvde dgtl34dbhk0006vmdb lvm2 a- 56.24g 56.24g

/dev/xvdf dgtl34dbhk0006vmbkup lvm2 a- 56.24g 0

/dev/xvdg dgtl34dbhk0006vmdb lvm2 a- 506.32g 356.32g

/dev/xvdh lvm2 a- 506.32g 506.32g

/dev/xvdi lvm2 a- 506.32g 506.32g

/dev/xvdj lvm2 a- 506.32g 506.32g

/dev/xvdk lvm2 a- 506.32g 506.32g

>> Remove old PVs from VG and wipe them:

root@lt34dbhk0006vm:~# vgreduce dgtl34dbhk0006vmdb /dev/xvdc /dev/xvdd /dev/xvde

Removed "/dev/xvdc" from volume group "dgtl34dbhk0006vmdb"

Removed "/dev/xvdd" from volume group "dgtl34dbhk0006vmdb"

Removed "/dev/xvde" from volume group "dgtl34dbhk0006vmdb"

root@lt34dbhk0006vm:~# pvremove /dev/xvdc

Labels on physical volume "/dev/xvdc" successfully wiped

root@lt34dbhk0006vm:~# pvremove /dev/xvdd

Labels on physical volume "/dev/xvdd" successfully wiped

root@lt34dbhk0006vm:~# pvremove /dev/xvde

Labels on physical volume "/dev/xvde" successfully wiped

>> On the farmservers, remove and cleanup the old PVs:

root@lt34xendb001:~# vm info lt34dbhk0006vm | egrep '(xvdc|xvdd|xvde)'

Block device xvdc : /dev/md/lt34dbhk0006vm\_2

Block device xvdd : /dev/md/lt34dbhk0006vm\_3

Block device xvde : /dev/md/lt34dbhk0006vm\_4

root@lt34xendb001:~# mdadm --detail /dev/md/lt34dbhk0006vm\_2

/dev/md/lt34dbhk0006vm\_2:

Version : 1.1

Creation Time : Thu Jul 10 13:44:23 2014

Raid Level : raid1

Array Size : 58972604 (56.24 GiB 60.39 GB)

Used Dev Size : 58972604 (56.24 GiB 60.39 GB)

Raid Devices : 2

Total Devices : 2

Persistence : Superblock is persistent

Intent Bitmap : Internal

Update Time : Thu Oct 2 20:06:23 2014

State : active

Active Devices : 2

Working Devices : 2

Failed Devices : 0

Spare Devices : 0

Name : SAN:lt34dbhk0006vm\_2

UUID : b4c0a673:a05cc538:dc0decf1:a4fa60aa

Events : 2

Number Major Minor RaidDevice State

0 253 126 0 active sync /dev/dm-126

1 253 137 1 active sync /dev/dm-137

root@lt34xendb001:~# mdadm --detail /dev/md/lt34dbhk0006vm\_3

/dev/md/lt34dbhk0006vm\_3:

Version : 1.1

Creation Time : Thu Jul 10 13:45:23 2014

Raid Level : raid1

Array Size : 58972604 (56.24 GiB 60.39 GB)

Used Dev Size : 58972604 (56.24 GiB 60.39 GB)

Raid Devices : 2

Total Devices : 2

Persistence : Superblock is persistent

Intent Bitmap : Internal

Update Time : Thu Oct 2 20:06:24 2014

State : active

Active Devices : 2

Working Devices : 2

Failed Devices : 0

Spare Devices : 0

Name : SAN:lt34dbhk0006vm\_3

UUID : f5e182ed:6c1d30f4:dc7e7632:a68fbd90

Events : 4

Number Major Minor RaidDevice State

0 253 123 0 active sync /dev/dm-123

1 253 136 1 active sync /dev/dm-136

root@lt34xendb001:~# mdadm --detail /dev/md/lt34dbhk0006vm\_4

/dev/md/lt34dbhk0006vm\_4:

Version : 1.1

Creation Time : Thu Jul 10 13:47:07 2014

Raid Level : raid1

Array Size : 58972604 (56.24 GiB 60.39 GB)

Used Dev Size : 58972604 (56.24 GiB 60.39 GB)

Raid Devices : 2

Total Devices : 2

Persistence : Superblock is persistent

Intent Bitmap : Internal

Update Time : Thu Oct 2 20:06:26 2014

State : active

Active Devices : 2

Working Devices : 2

Failed Devices : 0

Spare Devices : 0

Name : SAN:lt34dbhk0006vm\_4

UUID : 3ca238a2:e971c971:c83b7a71:5e1e450d

Events : 2

Number Major Minor RaidDevice State

0 253 42 0 active sync /dev/dm-42

1 253 135 1 active sync /dev/dm-135

>> View LUNs for SAN return (must be done before DMs are removed)

root@lt34xendb001:~# san\_shortinfo | egrep -i '(dm-42|dm-135|dm-123|dm-136|dm-126|dm-137)'

dm-135 | sdip 01:67 3A 97255 57622 | sdjn 01:67 4A 97255 57622 |

dm-137 | sdio 01:66 3A 97255 57622 | sdjm 01:66 4A 97255 57622 |

dm-136 | sdim 01:64 3A 97255 57622 | sdjk 01:64 4A 97255 57622 |

dm-42 | sdid 01:67 3A 97231 57622 | sdjb 01:67 4A 97231 57622 |

dm-126 | sdec 01:66 3A 97231 57622 | sdja 01:66 4A 97231 57622 |

dm-123 | sddd 01:64 3A 97231 57622 | sdiy 01:64 4A 97231 57622 |

root@lt34xendb001:~# vm rmdisk lt34dbhk0006vm lt34dbhk0006vm\_2

20:25:11 Checking farm status before running command 'rmdisk'

20:25:11 Establishing libvirt connection to lt34xendb001 OK

20:25:14 Establishing libvirt connection to lt34xendb002 OK

20:25:17 Verifying free config space on lt34xendb001 lt34xendb002 OK

20:25:23 Establishing libvirt connection to lt34xendb001 OK

20:25:26 Detaching md device /dev/md/lt34dbhk0006vm\_2 from VM lt34dbhk0006vm OK

20:25:30 Establishing libvirt connection to lt34xendb002 OK

20:25:32 Updating definition of lt34dbhk0006vm on lt34xendb002 OK

20:25:32 Stopping md device /dev/md/lt34dbhk0006vm\_2 OK

20:25:32 Removing md device /dev/md/lt34dbhk0006vm\_2 from mdadm.conf OK

20:25:34 /dev/md/lt34dbhk0006vm\_2 removed from lt34dbhk0006vm

root@lt34xendb001:~# vm rmdisk lt34dbhk0006vm lt34dbhk0006vm\_3

20:25:45 Checking farm status before running command 'rmdisk'

20:25:45 Establishing libvirt connection to lt34xendb001 OK

20:25:47 Establishing libvirt connection to lt34xendb002 OK

20:25:50 Verifying free config space on lt34xendb001 lt34xendb002 OK

20:25:56 Establishing libvirt connection to lt34xendb001 OK

20:25:59 Detaching md device /dev/md/lt34dbhk0006vm\_3 from VM lt34dbhk0006vm OK

20:26:04 Establishing libvirt connection to lt34xendb002 OK

20:26:05 Updating definition of lt34dbhk0006vm on lt34xendb002 OK

20:26:05 Stopping md device /dev/md/lt34dbhk0006vm\_3 OK

20:26:06 Removing md device /dev/md/lt34dbhk0006vm\_3 from mdadm.conf OK

20:26:07 /dev/md/lt34dbhk0006vm\_3 removed from lt34dbhk0006vm

root@lt34xendb001:~# vm rmdisk lt34dbhk0006vm lt34dbhk0006vm\_4

20:26:14 Checking farm status before running command 'rmdisk'

20:26:14 Establishing libvirt connection to lt34xendb001 OK

20:26:16 Establishing libvirt connection to lt34xendb002 OK

20:26:19 Verifying free config space on lt34xendb001 lt34xendb002 OK

20:26:25 Establishing libvirt connection to lt34xendb001 OK

20:26:28 Detaching md device /dev/md/lt34dbhk0006vm\_4 from VM lt34dbhk0006vm OK

20:26:32 Establishing libvirt connection to lt34xendb002 OK

20:26:34 Updating definition of lt34dbhk0006vm on lt34xendb002 OK

20:26:34 Stopping md device /dev/md/lt34dbhk0006vm\_4 OK

20:26:34 Removing md device /dev/md/lt34dbhk0006vm\_4 from mdadm.conf OK

20:26:36 /dev/md/lt34dbhk0006vm\_4 removed from lt34dbhk0006vm

Done. Now the replaced LUNs can be returned to the SAN (see TAS000000412164).

SAN return I

**NOTE:** The san\_return command does **NOT** cover this. We have to manually generate a clean pvinfo output to return the LUNs in STORM. Make sure you're using the correct SAN boxes and LUNs.

root@lt34xendb001:~# vm info lt34db0008vm

Name : lt34db0008vm

Domain ID : NR

Status : Running on lt34xendb002

VM Type : PV

Mounts : None

Storage Type : SAN+MD

Image : N/A

Last Backup (UTC) : 2014-10-27 22:03:43

Cur. VCPUs : 6

Max. VCPUs : 12

Cur. Mem [MB] : 10240

Max. Mem [MB] : 20480

XML - Cur. VCPUs : 6

XML - Max. VCPUs : 12

XML - Cur. Mem [MB] : 10240

XML - Max. Mem [MB] : 20480

XML - Target State : 1

XML - Priority : 1

Network device 0 : 00:16:3e:19:c3:da (br\_172025195000)

Network device 1 : 00:16:3e:19:f8:5d (br\_172025248000)

Block device xvda : /dev/md/lt34db0008vm

Block device xvdb : /dev/md/lt34db0008vm\_1

Block device xvdc : /dev/md/lt34db0008vm\_2

Block device xvdd : /dev/md/lt34db0008vm\_3

Block device xvde : /dev/md/lt34db0008vm\_4

Block device xvdf : /dev/md/lt34db0008vm\_5

Block device xvdg : /dev/md/lt34db0008vm\_6

Block device xvdh : /dev/md/lt34db0008vm\_7

Block device xvdi : /dev/md/lt34db0008vm\_8

Block device xvdj : /dev/md/lt34db0008vm\_9

Block device xvdk : /dev/md/lt34db0008vm\_10

Block device xvdl : /dev/md/lt34db0008vm\_11

Block device xvdm : /dev/md/lt34db0008vm\_12

Block device xvdn : /dev/md/lt34db0008vm\_13

Block device xvdo : /dev/md/lt34db0008vm\_14

Block device xvdp : /dev/md/lt34db0008vm\_15

Block device xvdq : /dev/md/lt34db0008vm\_16

Now return the replaced LUNs, in our case xvdc|xvdd|xvde|xvdg|xvdh:

root@lt34xendb001:~# vm info lt34db0008vm | egrep '(xvdc|xvdd|xvde|xvdg|xvdh)'

Block device xvdc : /dev/md/lt34db0008vm\_2

Block device xvdd : /dev/md/lt34db0008vm\_3

Block device xvde : /dev/md/lt34db0008vm\_4

Block device xvdg : /dev/md/lt34db0008vm\_6

Block device xvdh : /dev/md/lt34db0008vm\_7

root@lt34xendb002:~# mdadm --detail /dev/md/lt34db0008vm\_2

/dev/md/lt34db0008vm\_2:

Version : 1.1

Creation Time : Fri Jul 11 11:46:17 2014

Raid Level : raid1

Array Size : 58972604 (56.24 GiB 60.39 GB)

Used Dev Size : 58972604 (56.24 GiB 60.39 GB)

Raid Devices : 2

Total Devices : 2

Persistence : Superblock is persistent

Intent Bitmap : Internal

Update Time : Tue Oct 28 14:19:40 2014

State : active

Active Devices : 2

Working Devices : 2

Failed Devices : 0

Spare Devices : 0

Name : SAN:lt34db0008vm\_2

UUID : c2049538:a3a8b3e2:aabdda14:338e95a3

Events : 4

Number Major Minor RaidDevice State

0 253 127 0 active sync /dev/dm-127

1 253 139 1 active sync /dev/dm-139

root@lt34xendb002:~# mdadm --detail /dev/md/lt34db0008vm\_3

/dev/md/lt34db0008vm\_3:

Version : 1.1

Creation Time : Fri Jul 11 11:49:24 2014

Raid Level : raid1

Array Size : 58972604 (56.24 GiB 60.39 GB)

Used Dev Size : 58972604 (56.24 GiB 60.39 GB)

Raid Devices : 2

Total Devices : 2

Persistence : Superblock is persistent

Intent Bitmap : Internal

Update Time : Tue Oct 28 14:19:43 2014

State : active

Active Devices : 2

Working Devices : 2

Failed Devices : 0

Spare Devices : 0

Name : SAN:lt34db0008vm\_3

UUID : 5126160d:be095b17:a71c0fc1:96496802

Events : 7

Number Major Minor RaidDevice State

0 253 130 0 active sync /dev/dm-130

1 253 144 1 active sync /dev/dm-144

root@lt34xendb002:~# mdadm --detail /dev/md/lt34db0008vm\_4

/dev/md/lt34db0008vm\_4:

Version : 1.1

Creation Time : Fri Jul 11 12:07:59 2014

Raid Level : raid1

Array Size : 58972604 (56.24 GiB 60.39 GB)

Used Dev Size : 58972604 (56.24 GiB 60.39 GB)

Raid Devices : 2

Total Devices : 2

Persistence : Superblock is persistent

Intent Bitmap : Internal

Update Time : Tue Oct 28 14:19:50 2014

State : active

Active Devices : 2

Working Devices : 2

Failed Devices : 0

Spare Devices : 0

Name : SAN:lt34db0008vm\_4

UUID : 8b6a2828:f773b377:0a97b870:990238fd

Events : 2

Number Major Minor RaidDevice State

0 253 132 0 active sync /dev/dm-132

1 253 141 1 active sync /dev/dm-141

root@lt34xendb002:~# mdadm --detail /dev/md/lt34db0008vm\_6

/dev/md/lt34db0008vm\_6:

Version : 1.1

Creation Time : Thu Aug 21 15:26:26 2014

Raid Level : raid1

Array Size : 235890556 (224.96 GiB 241.55 GB)

Used Dev Size : 235890556 (224.96 GiB 241.55 GB)

Raid Devices : 2

Total Devices : 2

Persistence : Superblock is persistent

Intent Bitmap : Internal

Update Time : Tue Oct 28 14:19:54 2014

State : active

Active Devices : 2

Working Devices : 2

Failed Devices : 0

Spare Devices : 0

Name : SAN:lt34db0008vm\_6

UUID : 939c7a34:2f65d126:72cabd2b:d47c23e1

Events : 2

Number Major Minor RaidDevice State

0 253 151 0 active sync /dev/dm-151

1 253 152 1 active sync /dev/dm-152

root@lt34xendb002:~# mdadm --detail /dev/md/lt34db0008vm\_7

/dev/md/lt34db0008vm\_7:

Version : 1.1

Creation Time : Thu Aug 21 15:28:44 2014

Raid Level : raid1

Array Size : 235890556 (224.96 GiB 241.55 GB)

Used Dev Size : 235890556 (224.96 GiB 241.55 GB)

Raid Devices : 2

Total Devices : 2

Persistence : Superblock is persistent

Intent Bitmap : Internal

Update Time : Tue Oct 28 14:19:56 2014

State : active

Active Devices : 2

Working Devices : 2

Failed Devices : 0

Spare Devices : 0

Name : SAN:lt34db0008vm\_7

UUID : 4c6e6b10:3932b0f5:82fd53eb:fb8c4ded

Events : 3

Number Major Minor RaidDevice State

0 253 154 0 active sync /dev/dm-154

1 253 150 1 active sync /dev/dm-150

Now that we know the configuration of eatch DM, we can remove them:

root@lt34xendb002:~# san\_shortinfo | egrep -iw '(dm-151|dm-152|dm-136|dm-142|dm-132|dm-141|dm-130|dm-144|dm127|dm139|dm-154|dm-150)'

dm-136 | sdij 01:6D 3A 97231 57622 | sdjh 01:6D 4A 97231 57622 |

dm-132 | sdii 01:6C 3A 97231 57622 | sdjg 01:6C 4A 97231 57622 |

dm-154 | sdkh 00:B1 3B 97231 230490 | sdkn 00:B1 4B 97231 230490 |

dm-130 | sdih 01:6B 3A 97231 57622 | sdjf 01:6B 4A 97231 57622 |

dm-151 | sdkg 00:B0 3B 97231 230490 | sdkm 00:B0 4B 97231 230490 |

dm-150 | sdke 00:B1 3B 97255 230490 | sdkk 00:B1 4B 97255 230490 |

dm-142 | sdiv 01:6D 3A 97255 57622 | sdjt 01:6D 4A 97255 57622 |

dm-152 | sdkd 00:B0 3B 97255 230490 | sdkj 00:B0 4B 97255 230490 |

dm-141 | sdiu 01:6C 3A 97255 57622 | sdjs 01:6C 4A 97255 57622 |

dm-144 | sdit 01:6B 3A 97255 57622 | sdjr 01:6B 4A 97255 57622 |

root@lt34xendb002:~# san\_shortinfo | egrep -iw '(01:6D|01:6C|00:B1|01:6B|00:B0)'

dm-136 | sdij 01:6D 3A 97231 57622 | sdjh 01:6D 4A 97231 57622 |

dm-132 | sdii 01:6C 3A 97231 57622 | sdjg 01:6C 4A 97231 57622 |

dm-154 | sdkh 00:B1 3B 97231 230490 | sdkn 00:B1 4B 97231 230490 |

dm-130 | sdih 01:6B 3A 97231 57622 | sdjf 01:6B 4A 97231 57622 |

dm-151 | sdkg 00:B0 3B 97231 230490 | sdkm 00:B0 4B 97231 230490 |

dm-150 | sdke 00:B1 3B 97255 230490 | sdkk 00:B1 4B 97255 230490 |

dm-142 | sdiv 01:6D 3A 97255 57622 | sdjt 01:6D 4A 97255 57622 |

dm-152 | sdkd 00:B0 3B 97255 230490 | sdkj 00:B0 4B 97255 230490 |

dm-141 | sdiu 01:6C 3A 97255 57622 | sdjs 01:6C 4A 97255 57622 |

dm-144 | sdit 01:6B 3A 97255 57622 | sdjr 01:6B 4A 97255 57622 |

Remove the disks from the VM:

root@lt34xendb002:~# vm rmdisk lt34db0008vm lt34db0008vm\_2

15:02:46 Checking farm status before running command 'rmdisk'

15:02:46 Establishing libvirt connection to lt34xendb001 OK

15:02:49 Establishing libvirt connection to lt34xendb002 OK

15:02:51 Verifying free config space on lt34xendb001 lt34xendb002 OK

15:02:58 Establishing libvirt connection to lt34xendb002 OK

15:02:59 Detaching md device /dev/md/lt34db0008vm\_2 from VM lt34db0008vm OK

15:03:06 Establishing libvirt connection to lt34xendb001 OK

15:03:13 Updating definition of lt34db0008vm on lt34xendb001 OK

15:03:14 Stopping md device /dev/md/lt34db0008vm\_2 OK

15:03:16 Removing md device /dev/md/lt34db0008vm\_2 from mdadm.conf OK

15:03:24 /dev/md/lt34db0008vm\_2 removed from lt34db0008vm

root@lt34xendb002:~# vm rmdisk lt34db0008vm lt34db0008vm\_3

15:03:58 Checking farm status before running command 'rmdisk'

15:03:58 Establishing libvirt connection to lt34xendb001 OK

15:04:01 Establishing libvirt connection to lt34xendb002 OK

15:04:04 Verifying free config space on lt34xendb001 lt34xendb002 OK

15:04:10 Establishing libvirt connection to lt34xendb002 OK

15:04:12 Detaching md device /dev/md/lt34db0008vm\_3 from VM lt34db0008vm OK

15:04:18 Establishing libvirt connection to lt34xendb001 OK

15:04:21 Updating definition of lt34db0008vm on lt34xendb001 OK

15:04:21 Stopping md device /dev/md/lt34db0008vm\_3 OK

15:04:22 Removing md device /dev/md/lt34db0008vm\_3 from mdadm.conf OK

15:04:41 /dev/md/lt34db0008vm\_3 removed from lt34db0008vm

root@lt34xendb002:~# vm rmdisk lt34db0008vm lt34db0008vm\_4

15:04:53 Checking farm status before running command 'rmdisk'

15:04:53 Establishing libvirt connection to lt34xendb001 OK

15:04:56 Establishing libvirt connection to lt34xendb002 OK

15:04:58 Verifying free config space on lt34xendb001 lt34xendb002 OK

15:05:04 Establishing libvirt connection to lt34xendb002 OK

15:05:06 Detaching md device /dev/md/lt34db0008vm\_4 from VM lt34db0008vm OK

15:05:12 Establishing libvirt connection to lt34xendb001 OK

15:05:15 Updating definition of lt34db0008vm on lt34xendb001 OK

15:05:15 Stopping md device /dev/md/lt34db0008vm\_4 OK

15:05:16 Removing md device /dev/md/lt34db0008vm\_4 from mdadm.conf OK

15:05:18 /dev/md/lt34db0008vm\_4 removed from lt34db0008vm

root@lt34xendb002:~# vm rmdisk lt34db0008vm lt34db0008vm\_5

15:05:23 Checking farm status before running command 'rmdisk'

15:05:23 Establishing libvirt connection to lt34xendb001 OK

15:05:26 Establishing libvirt connection to lt34xendb002 OK

15:05:29 Verifying free config space on lt34xendb001 lt34xendb002 OK

15:05:35 Establishing libvirt connection to lt34xendb002 OK

15:05:37 Detaching md device /dev/md/lt34db0008vm\_5 from VM lt34db0008vm OK

15:07:20 Establishing libvirt connection to lt34xendb001 OK

15:07:24 Updating definition of lt34db0008vm on lt34xendb001 OK

15:07:24 Stopping md device /dev/md/lt34db0008vm\_5 FAILED

15:07:24 ERROR: Could not stop md device /dev/md/lt34db0008vm\_5, aborting

15:07:24 ERROR: Command rmdisk did not complete successfully

root@lt34xendb002:~# vm rmdisk lt34db0008vm lt34db0008vm\_6

15:08:02 Checking farm status before running command 'rmdisk'

15:08:02 Establishing libvirt connection to lt34xendb001 OK

15:08:05 Establishing libvirt connection to lt34xendb002 OK

15:08:08 Verifying free config space on lt34xendb001 lt34xendb002 OK

15:08:14 Establishing libvirt connection to lt34xendb002 OK

15:08:16 Detaching md device /dev/md/lt34db0008vm\_6 from VM lt34db0008vm OK

15:08:33 Establishing libvirt connection to lt34xendb001 OK

15:08:36 Updating definition of lt34db0008vm on lt34xendb001 OK

15:08:37 Stopping md device /dev/md/lt34db0008vm\_6 OK

15:08:38 Removing md device /dev/md/lt34db0008vm\_6 from mdadm.conf OK

15:08:39 /dev/md/lt34db0008vm\_6 removed from lt34db0008vm

root@lt34xendb002:~# vm rmdisk lt34db0008vm lt34db0008vm\_7

15:08:43 Checking farm status before running command 'rmdisk'

15:08:43 Establishing libvirt connection to lt34xendb001 OK

15:08:47 Establishing libvirt connection to lt34xendb002 OK

15:08:49 Verifying free config space on lt34xendb001 lt34xendb002 OK

15:08:56 Establishing libvirt connection to lt34xendb002 OK

15:08:57 Detaching md device /dev/md/lt34db0008vm\_7 from VM lt34db0008vm OK

15:09:03 Establishing libvirt connection to lt34xendb001 OK

15:09:06 Updating definition of lt34db0008vm on lt34xendb001 OK

15:09:06 Stopping md device /dev/md/lt34db0008vm\_7 OK

15:09:07 Removing md device /dev/md/lt34db0008vm\_7 from mdadm.conf OK

15:09:09 /dev/md/lt34db0008vm\_7 removed from lt34db0008vm

As we do know the LUN IDs, verify the DM devices and remove them:

root@lt34xendb001:~# pvinfo -i | egrep -iw '(00:b0|00:b1|01:6a|01:6b|01:6c)' | egrep '(97231|97255)' | grep -i "dm-"

/dev/dm-155 00 00 00 CL4-F-1 00:b1 OPEN-V 00097255

/dev/dm-153 00 00 00 CL4-F-1 00:b1 OPEN-V 00097231

/dev/dm-152 00 00 00 CL4-F-1 00:b0 OPEN-V 00097255

/dev/dm-150 00 00 00 CL4-F-1 00:b0 OPEN-V 00097231

/dev/dm-142 00 00 00 CL4-E-1 01:6c OPEN-V 00097255

/dev/dm-141 00 00 00 CL4-E-1 01:6b OPEN-V 00097255

/dev/dm-132 00 00 00 CL4-E-1 01:6c OPEN-V 00097231

/dev/dm-128 00 00 00 CL4-E-1 01:6b OPEN-V 00097231

/dev/dm-140 00 00 00 CL4-E-1 01:6a OPEN-V 00097255

/dev/dm-129 00 00 00 CL3-E-1 01:6a OPEN-V 00097231

Remove the DMs:

root@lt34xendb001:~# dmsetup info /dev/dm-155

Name: 360060e80167be70000017be7000000b1

State: ACTIVE

Read Ahead: 1024

Tables present: LIVE

Open count: 0

Event number: 0

Major, minor: 253, 155

Number of targets: 1

UUID: mpath-360060e80167be70000017be7000000b1

Note: "dmsetup" works on the WWN or /dev/dm-??? block device file:

root@lt34xendb001:~# dmsetup remove 360060e80167be70000017be7000000b1

root@lt34xendb001:~# dmsetup info /dev/dm-155

Device /dev/dm-155 not found

Command failed

root@lt34xendb001:~# dmsetup info /dev/dm-153

Name: 360060e80167bcf0000017bcf000000b1

State: ACTIVE

Read Ahead: 1024

Tables present: LIVE

Open count: 0

Event number: 0

Major, minor: 253, 153

Number of targets: 1

UUID: mpath-360060e80167bcf0000017bcf000000b1

root@lt34xendb001:~# dmsetup info /dev/dm-153

Device /dev/dm-153 not found

Command failed

As the 2nd command fails, we know that it's really removed from the system. Remove the remaining devices too:

root@lt34xendb001:~# dmsetup remove /dev/dm-152

root@lt34xendb001:~# dmsetup remove /dev/dm-150

root@lt34xendb001:~# dmsetup remove /dev/dm-142

root@lt34xendb001:~# dmsetup remove /dev/dm-141

root@lt34xendb001:~# dmsetup remove /dev/dm-132

root@lt34xendb001:~# dmsetup remove /dev/dm-128

root@lt34xendb001:~# dmsetup remove /dev/dm-140

root@lt34xendb001:~# dmsetup remove /dev/dm-129

SAN return II

**NOTE:** The san\_return command does **NOT** cover this. We have to manually generate a clean pvinfo output to return the LUNs in STORM. Make sure you're using the correct SAN boxes and LUNs.

root@lt34xendb001:~# pvinfo -i | egrep -w -i '(00:b0|00:b1|01:6a|01:6b|01:6c)' | egrep '(97231|97255)'

No supported HBA found!

Unable to get information.

Possible reasons:

HBAAPI not registered / loaded

driver with different capabilities

a.s.o

/dev/sdkn 00 00 00 CL4-F-1 00:b1 OPEN-V 00097255

/dev/sdkm 00 00 00 CL4-F-1 00:b0 OPEN-V 00097255

/dev/sdkj 00 00 00 CL4-F-1 00:b0 OPEN-V 00097231

/dev/sdkk 00 00 00 CL4-F-1 00:b1 OPEN-V 00097231

/dev/sdke 00 00 00 CL4-E-1 01:6b OPEN-V 00097255

/dev/sdkg 00 00 00 CL4-E-1 01:6c OPEN-V 00097255

/dev/sdjt 00 00 00 CL4-E-1 01:6c OPEN-V 00097231

/dev/sdjs 00 00 00 CL4-E-1 01:6b OPEN-V 00097231

/dev/sdjr 00 00 00 CL3-F-1 00:b1 OPEN-V 00097255

/dev/sdjg 00 00 00 CL3-F-1 00:b1 OPEN-V 00097231

/dev/sdjh 00 00 00 CL3-F-1 00:b0 OPEN-V 00097255

/dev/sdjf 00 00 00 CL3-F-1 00:b0 OPEN-V 00097231

/dev/sdiu 00 00 00 CL3-E-1 01:6c OPEN-V 00097255

/dev/sdit 00 00 00 CL3-E-1 01:6b OPEN-V 00097255

/dev/sdii 00 00 00 CL3-E-1 01:6c OPEN-V 00097231

/dev/sdih 00 00 00 CL3-E-1 01:6b OPEN-V 00097231

/dev/sdjq 00 00 00 CL4-E-1 01:6a OPEN-V 00097255

/dev/sdje 00 00 00 CL4-E-1 01:6a OPEN-V 00097231

/dev/sdis 00 00 00 CL3-E-1 01:6a OPEN-V 00097255

/dev/sdig 00 00 00 CL3-E-1 01:6a OPEN-V 00097231

Now clean up the devices and return them in STORM:

root@lt34xendb002:~# for i in sdkn sdkm sdkk sdkj sdkg sdke sdjt sdjs sdjr sdjh sdjg sdjf sdiu sdit sdii sdih sdjq sdje sdis sdig; do echo 1 > /sys/block/$i/device/delete; done

Last edited by Liebl Markus, (Markus.Liebl@partner.bmw.de) , based on work by Kemmerer Walter, (Walter.Kemmerer@partner.bmw.de) , [Kujau Christian, (Christian.CK.Kujau@partner.bmw.de)](https://bsswiki.muc/tiki-user_information.php?userId=55) , Venzke Horst, (Horst.Venzke@partner.bmw.de) , [qxc0474](https://bsswiki.muc/tiki-user_information.php?userId=-1) and Urban Richard, (Richard.Urban@partner.bmw.de) .  
Page last modified on Wednesday 16 of October, 2019 16:09:43 CEST. (Version 25)

* Created by [Kishore Kumar (ext.)](https://atc.bmwgroup.net/confluence/display/~qxz0crd), last modified on [08 Apr 2020](https://atc.bmwgroup.net/confluence/pages/diffpagesbyversion.action?pageId=653793283&selectedPageVersions=3&selectedPageVersions=4)

# How to rename a cluster package

**NOTE: Do not use this procedure for SAP cluster packages! Renaming a SAP package is a nonstandard request that must be handled differently. Notify the L2 team and SAP module responsibles if a request for renaming a SAP package is received.**

## Preparation

* You need to have a downtime for the package in question

### Collect necessary data

* In order to have a smooth package migration, we sample some data which we need for the migration as well as for a fallback if necessary. **Ensure that you run this on the node where the package is active**

# mkdir -p /root/pkg\_ren

# cd /root/pkg\_ren

# hastatus -sum > hastatus\_migration\_pre

# vxdisk -o alldgs list > vxdisk\_o\_alldgs\_list\_pre

# vxdg -q list > vxdg\_q\_list\_pre

# df -hP > df\_hP\_pre

#

# hacf -cftocmd /etc/VRTSvcs/conf/config/ -display | \

grep <old\_package\_name> > <old\_package\_name>.cmd

# df -hP | grep <old\_package\_name> > mountpoints\_<old\_package\_name>.txt

### Modify config files

* As we have now all the necessary information we need, we can go ahead and prepare the new configuration:

# cd /root/pkg\_ren

# cp <old\_package\_name>.cmd <new\_package\_name>.cmd

# sed -i 's/<old\_package\_name>/<new\_package\_name>/g' <new\_package\_name>.cmd

# chmod u+x <new\_package\_name>.cmd

### Create new Mountpoints

* When the package is renamed we also need new mount points to reflect that changes. **Be aware to do that on BOTH nodes!**:

# MOUNTS\_OLD=$( grep global mountpoints\_<old\_package\_name>.txt | awk '{print $6}' )

# MOUNTS\_NEW=$( echo "$MOUNTS\_OLD" | sed s/<old\_package\_name>/<new\_package\_name>/g )

# for i in $MOUNTS\_NEW; do mkdir -p $i; done

# for i in $MOUNTS\_NEW; do ls -d $i; done

### Change cluster scripts:

* Also the cluster scripts need to be changed to reflect the change. **Be aware to do that on BOTH nodes!**:

# cd /lfs/cluster/vcs

# for i in `ls /lfs/cluster/vcs | grep <old\_package\_name>`; do

SCRIPT\_NEW=$( echo "$i" | sed s/<old\_package\_name>/<new\_package\_name>/g )

cat $i | sed s/<old\_package\_name>/<new\_package\_name>/g > $SCRIPT\_NEW

done

# chmod u+x <new\_package\_name>\*

* check the scripts afterwards !

## Actual rename

* When all preparation steps are done, you should have:
  + a new configuration file for the package named "new\_package\_name".cmd which is executable
  + new mountpoints **on both nodes**
  + new start/stop/monitor scripts for the cluster **on both nodes**

### Set downtime

* Before you start, set a downtime for the nodes:

### Stop package

* When you've got the ok to start, at first bring down the package:

# hagrp -offline <old\_package\_name> -any

##verify

# hastatus -sum

### Delete old package

* When the package is down, delete it!

# haconf -makerw

# for i in `hagrp -resources <old\_package\_name>`; do hares -delete $i; done

# hagrp -delete <old\_package\_name>

##and verify

# hares -display -group <old\_package\_name>

# hagrp -display <old\_package\_name>

* last 2 commands should fail as resources and group have all been deleted. If there is still something configured, tidy up before going ahead !

### Create package with new name

* If you are sure that the older package is removed completely, create the new one and make config read only again:

# ./<new\_package\_name>.cmd

# haconf -dump -makero

* Check the output for possible problems and fix them

### Rename disk groups

* At this point it is necessary to change the disk group names. So first check which DGs will have to be renamed:

# grep add <old\_package\_name>.cmd | grep DiskGroup > old\_DGs

# grep add <new\_package\_name>.cmd | grep DiskGroup > new\_DGs

# sdiff old\_DGs new\_DGs | awk '{print "vxdg -n " $9 " import " $3}' | sed -e 's/<old\_pkg\_name>\_//' -e 's/<new\_pkg\_name>\_//'

* Check that output and if OK, just copy and paste to command line
* When this was successful, deport the DGs again:

# for DG in $(awk '{print $3}' new\_DGs); do echo $DG; vxdg deport $DG; done

### Start package

* If the DG import and deport have been successful, we can now start the package

# hagrp -online <new\_package\_name> -sys <node\_package\_was\_online\_before>

* If any trouble, check log files for further information
* Check nagios for problems, if all checks are fine: App team should check the application

### Do a switch test

* After you brought the package online successful, you should switch it to the other node

# hagrp -switch <new\_package\_name> -to <other\_node>

* If you have Problems, check at first if mount points and/or cluster scripts have been created/modified as shown above
* App team shaould check again
* Switch back if needed

### Tidy up

* If everything worked well, tidy up:

1. remove old mount points
2. remove old cluster scripts
3. remove files/scripts generated
4. remove downtime

## Fallback

* In case there are problems which cannot be fixed, we have the possibility to roll back!
* But first of all document in detail what and if possible why it did not work as this is needed for further investigation
* open a PM ticket at least after the fallback !

### Bring package offline

* Bring the package offline if it is running and ensure that it is down:

# hagrp -offline <new\_packagename> -any

# hastatus -sum

* If only resources but not the whole package/service group is online, bring them down and verify:

# hares -clear <res> -sys <node>

# hares -offline <res> -sys <node>

# hastatus -sum

### Remove new package

* The next step is to make the configuration writeable again and delete the new package:

# haconf -makerw

# for i in `hagrp -resources <new\_package\_name>`; do hares -delete $i; done

# hagrp -delete <new\_package\_name>

* If no problems occur the "new" package is gone and we can go ahead and configure the old one again:

# chmod u+x <old\_package\_name>.cmd

# ./<old\_package\_name>.cmd

##verify

# hastatus -sum

##and make config read only again

# haconf -dump -makero

### Rename DGs again

* Now we have to rename the DGs to the old names:

sdiff new\_DGs old\_DGs | awk '{print "vxdg -n " $9 " import " $3}'

* Check that output and if OK, just copy and paste to command line
* When this was successful, deport the DGs again:

# for DG in $(awk '{print $3}' old\_DGs); do echo $DG; vxdg deport $DG; done

### Start package again

* last step is now to bring the package online again:

# hagrp -online <old\_package\_name> - sys <node>

* A switch test to ensure that this is working would be a good idea

Last edited by Wild Thomas, (Thomas.Wild@partner.bmw.de) , based on work by Kemmerer Walter, (Walter.Kemmerer@partner.bmw.de) , Strobel Werner, (Werner.WS.Strobel@partner.bmw.de) , Porezag Dirk, (Dirk.Porezag@partner.bmw.de) , qxc0474 and system .  
Page last modified on Monday 03 of February, 2020 10:56:49 CET. (Version 35)

* Created by [Kishore Kumar (ext.)](https://atc.bmwgroup.net/confluence/display/~qxz0crd), last modified by Unknown User (qxy9685) on [16 Apr 2020](https://atc.bmwgroup.net/confluence/pages/diffpagesbyversion.action?pageId=649682721&selectedPageVersions=7&selectedPageVersions=8)

### Table of contents

* General information
* Task
  + sample text
  + preparation
* resolve package name in CMDB
* Technical doing
  + Check if all SAN devices are visible correctly
  + Check HA status
  + Perform switch
  + [Perform switch for sap / bank servers](https://atc.bmwgroup.net/confluence/display/IAAS18/5.6.1.11+VCS+Switchtest#id-5.6.1.11VCSSwitchtest-SAP&BankSpecificClusternodes-Packageswitchtest)
* Debugging
  + Logfiles

# General information

With a cluster switch test, a package (BMW wording) / Veritas Cluster service group can be switched from one node to another. On one node the package is online, on the other offline.  
  
A use case for a cluster switch is to patch the nodes consecutively.  
  
BEFORE and AFTER each step in this howto, ensure that the single step and the result of this step is approved by the requester in the task.

# Task

## sample text

**Sample task text**

Perform failover test on db package dbipm89 after db patching is complete

## preparation

Check if the package IP-address is linked to a ressource group in CMDB (see below).

# resolve package name in CMDB

In the example above, "dbipm89" is the package name. To get the servers on which this package is running on, you have to resolve the package name in the CMDB.  
  
**First check DNS:**

**dns**

qx33743@lpinstbss1:~> nslookup dbipm89

Server:         160.50.250.2

Address:       160.50.250.2#53

Non-authoritative answer:

Name:   dbipm89.w34

Address: 172.25.195.35

**CMDB:**  
- Server Application Management  
- Schlagwortsuche  
- search for "dbipm89\*"  
- look in the results for "Klasse" "Ressourcen Gruppe", result "RG-XXXX" -> double click  
- on the left pane, click on Failover Cluster  
- in the name column you will find <server1>\_<server2>  
- double click on the "FC-XXXX" entry  
- click on "Server" -> list of servers  
  
HINT: the IP-address of the package MUST be attached to the CMDB ressource group, NOT the server itself.

# Technical doing

## Check if all SAN devices are visible correctly

On both servers:

**Check disk visibility**

lpinstiaas01$ /lfs/scripts/vcs\_sg\_mig.pl CheckDisks -s lp34d11 -t lp34d12 -g dbipm89

OK: all 18 disks of service group dbipm89 are visible on lp34d12

**If some Disks not visible rescan & run check again**

lp34d11/12 $ san\_rescan -o

lpinstbss1$ /lfs/scripts/vcs\_sg\_mig.pl CheckDisks -s lp34d11 -t lp34d12 -g dbipm89

## Check HA status

Login to one of the server:  
hastatus -sum | grep dbipm89  
  
  
hastatus

lp34d11:/home/qx33743 # hastatus -sum | grep dbipm89

B dbipm89         lp34d11             Y         N               OFFLINE

B dbipm89         lp34d12             Y         N               ONLINE

## Perform switch

ATTENTION: this step has to be approved by the requester!  
  
Switch the package from the online-node to the offline-node, on one of both servers:

**hagrp**

Syntax:

hagrp -switch <packet> -to system

Command:

hagrp -switch dbipm89 -to lp34d11

Check the Cluster and ServiceGroup Logfiles under /var/VRTSvcs/log/  
A "tail -f /var/VRTSvcs/log/BMWApplication\_(servicegroup\_name).log" should inform you of any errors encountered when starting the application(s) and should be reporeted to the Change Manager or Application Group responsible for the application.

The VCS log: engine\_A.log  
The ServiceGroup log: BMWApplication\_(servicegroup\_name).log  
Some also have extra MiddleWare Logs (check the Cluster Service Group Start Script under: /lfs/cluster/vcs/<servicegroup>\*\_start

## SAP & Bank Specific and few Bank Cluster nodes - Package switch test

**If  Group dependency is set (only on Cluster with Multiple ResourceGorups):**

Refer below example:

lp10g3p07:~$ sudo hagrp -dep

#Parent      Child      Relationship

ig3ici10     pg3pci00   offline local

The Parent is always the NON-PROD on SAP and the Group which is “sacrificed” when there is a node failure.

Then the NON-PROD Parent must have Downtime as well and be first manually stopped as described below:

SAP Cluster with two  ResourceGroups (SIDs):

1)            Stop the  NON-Prod „Parent“  with  hagrp -offline

2)            Switch the  PROD „Child“ with  hagrp -switch tot he other node

3)            Start the NON-Prod „Parent“  with hagrp -online the other - now free - node

Check with the hastatus command, the package must appear as ONLINE on the secondary node.  
  
Hint: if you type "hastatus" without "-sum" in a new shell, you can read on-line the status of the switch.

## SAP Specific Cluster nodes - Package switch test

**If  Group dependency is set (only on Cluster with Multiple ResourceGorups):**

lp10g3p07:~$ sudo hagrp -dep

#Parent      Child      Relationship

ig3ici10     pg3pci00   offline local

The Parent is always the NON-PROD on SAP and the Group which is “sacrificed” when there is a node failure.

Then the NON-PROD Parent must have Downtime as well and be first manually stopped as described below:

SAP Cluster with two  ResourceGroups (SIDs):

1)            Stop the  NON-Prod „Parent“  with  hagrp -offline

2)            Switch the  PROD „Child“ with  hagrp -switch tot he other node

3)            Start the NON-Prod „Parent“  with hagrp -online the other - now free - node

# Debugging

## Logfiles

**logfiles**

VCS logfiles:

lp34d11:/var/VRTSvcs/log/

VCS package logfiles:

lp34d11:/var/VRTSvcs/log # ls -la \*dbipm89\*

-rw-r--r-- 1 root root 38096 Aug 19 11:44 BMWApplication\_dbipm89.log

Last edited by Wild Thomas, (Thomas.Wild@partner.bmw.de) , based on work by Noyes Geoffrey, (Geoffrey.Noyes@partner.bmw.de) , qxc0474 , Lianas Zisis, (Zisis.Lianas@partner.bmw.de) , Thomas Marcel, (Marcel.Thomas@partner.bmw.de) and system .  
Page last modified on Monday 03 of February, 2020 09:39:59 CET. (Version 26)

* Created by Unknown User (qxy1271), last modified by [Sanjiv Soraganvi (ext.)](https://atc.bmwgroup.net/confluence/display/~qxw9324) on [16 Aug 2021](https://atc.bmwgroup.net/confluence/pages/diffpagesbyversion.action?pageId=649689339&selectedPageVersions=7&selectedPageVersions=8)

# Preparation

## CHECK IP

### TODO

## Request LUN Visibility for new Cluster nodes

Example:  
source server: lpdbsoem5  
oracle package: dboemao  
target cluster: lpdbs91 / lpdbs92

PKG=dboemao

SRC\_HOST=lpdbsoem5

# source server:

lpinstbss1 ~ $ ssh ${SRC\_HOST} "sudo /sbin/san\_shortinfo" | egrep "${PKG}"

VSP\_10567\_01f0 | sdam 01:F0 5P 10567 518602 | sdcc 01:F0 6P 10567 518602 | dgdboemaodb

VSP\_10567\_01f1 | sdan 01:F1 5P 10567 518602 | sdcd 01:F1 6P 10567 518602 | dgdboemaodb

VSP\_10567\_01f2 | sdao 01:F2 5P 10567 518602 | sdce 01:F2 6P 10567 518602 | dgdboemaobkup

VSP\_10567\_01f3 | sdap 01:F3 5P 10567 14405 | sdcf 01:F3 6P 10567 14405 | dgdboemaoredo1

VSP\_10567\_01f4 | sdaq 01:F4 5P 10567 14405 | sdcg 01:F4 6P 10567 14405 | dgdboemaoredo2

VSP\_10582\_01f0 | sdbh 01:F0 6P 10582 518602 | sdr 01:F0 5P 10582 518602 | dgdboemaodb

VSP\_10582\_01f1 | sdbi 01:F1 6P 10582 518602 | sds 01:F1 5P 10582 518602 | dgdboemaodb

VSP\_10582\_01f2 | sdbj 01:F2 6P 10582 518602 | sdt 01:F2 5P 10582 518602 | dgdboemaobkup

VSP\_10582\_01f3 | sdbk 01:F3 6P 10582 14405 | sdu 01:F3 5P 10582 14405 | dgdboemaoredo1

VSP\_10582\_01f4 | sdbl 01:F4 6P 10582 14405 | sdv 01:F4 5P 10582 14405 | dgdboemaoredo2

## Check VxVM Filesystem Layout Version

If you migrate the package from an older version to Veritas Storage Foundation Version 6 or higher, further steps are neccessary.  
  
Check and **note** the current version of the disk layout.

lpinstbss1 ~ $ ssh ${SRC\_HOST}

lpdbsoem5 ~ $ sudo -s

lpdbsoem5:~ # PKG=dboemao

lpdbsoem5:~ # for VOL in $(df -hP -t vxfs | grep ${PKG} | awk '{print $1}'); do /opt/VRTS/bin/fstyp -v ${VOL} | grep -i version; done

magic a501fcf5 version 10 ctime Tue 23 May 2017 03:10:43 PM CEST

magic a501fcf5 version 10 ctime Tue 23 May 2017 03:10:48 PM CEST

magic a501fcf5 version 10 ctime Tue 23 May 2017 03:10:52 PM CEST

magic a501fcf5 version 10 ctime Tue 23 May 2017 03:10:56 PM CEST

## san\_rescan / check visibility

Once the storm order has been completed do a san\_rescan, and check the Visibility on **BOTH Target Cluster Nodes**

# Migrating the Package

## CMDB: check/assign new IP to Package

1. in the CMDB create the new **"Resource Group"** ${PKG} in the target **"Failover Cluster"** name NODE1\_NODE2
2. in the CMDB you will need to assign the new **Package IP** to the **"Resource Group"** created above

See [CMDB Ressource Group and Failover-Group Management](https://atc.bmwgroup.net/confluence/display/IAAS18/RG+Management)  
  
Use the following **checks** to see if the IP has been setup correctly:

lpinstbss1 ~ $ PKG=dboemao

lpinstbss1 ~ $ host ${PKG}

dboemao.bmwgroup.net is an alias for lpdbsoem5.bmwgroup.net. <<< you will need to delete this alias

lpdbsoem5.bmwgroup.net has address 160.46.136.212

lpinstbss1 ~ $ c6000 ${PKG}

.------------------------------------------------.

| dboemao-new |

+---------------+--------------------------------+

| key | value |

+---------------+--------------------------------+

| HOSTNAME | dboemao-new | <<< you will need to rename to ${PKG}

| DOMAIN | bmwgroup.net |

| LOG\_SERVER\_ID | |

| CLUSTER\_NAME | | <<< you will need to assign IP to Cluster Name

| ALIAS | |

| IP\_ADDRESS | 160.50.18.83 |

| NSI | |

| SUBNETMASK | 255.255.255.0 |

| GATEWAY | 160.50.18.1 |

| NET\_ID | 160.50.18.0 |

| BROADCAST | 160.50.18.255 |

| INT\_VLAN | |

| SUB\_AREA | MUC-HA-III |

'---------------+--------------------------------'

## umount, deport, and remove /etc/fstab entries

lpdbsoem5:~ # df -hP -t vxfs | egrep ${PKG}

/dev/vx/dsk/dgdboemaobkup/lvbkup 507G 997M 502G 1% /global/dboemao/bkup

/dev/vx/dsk/dgdboemaodb/lvdb 1013G 496G 513G 50% /global/dboemao/db

/dev/vx/dsk/dgdboemaoredo1/lvredo1 15G 208M 13G 2% /global/dboemao/redo1

/dev/vx/dsk/dgdboemaoredo2/lvredo2 15G 150M 14G 2% /global/dboemao/redo2

lpdbsoem5:~ # egrep ${PKG} /etc/fstab

/dev/vx/dsk/dgdboemaobkup/lvbkup /global/dboemao/bkup vxfs \_netdev 0 0

/dev/vx/dsk/dgdboemaodb/lvdb /global/dboemao/db vxfs mincache=direct,convosync=direct,\_netdev 0 0

/dev/vx/dsk/dgdboemaoredo1/lvredo1 /global/dboemao/redo1 vxfs mincache=direct,convosync=direct,\_netdev 0 0

/dev/vx/dsk/dgdboemaoredo2/lvredo2 /global/dboemao/redo2 vxfs mincache=direct,convosync=direct,\_netdev 0 0

lpdbsoem5:~ # umount /global/dboemao/db

lpdbsoem5:~ # umount /global/dboemao/bkup

lpdbsoem5:~ # umount /global/dboemao/redo1

lpdbsoem5:~ # umount /global/dboemao/redo2

lpdbsoem5:~ # for DG in $(egrep ${PKG} /etc/fstab | cut -f5 -d"/") ; do vxdg deport $DG ; done

lpdbsoem5:~ # vi /etc/fstab

## remove above entries...

## double check: should return nothing

lpdbsoem5:~ # egrep ${PKG} /etc/fstab

lpdbsoem5:~ #

## Update Disk Layout if needed

On the target cluster you will need to stepwise update the Veritas Disk layout to at least version 7.  
  
For example, if the Current Version is 4 then you will need to:  
  
import the diskgroup  
mount the volume in local mode (not cluster)  
stepwise update the Veritas disk layout versions (I think you will need to re-import and mount each time, because the update umounts and deports. ...)  
  
EXAMPLE: upgrading from Version 4 to Version 7: you will need to do the following for each volume of the cluster package.

mkdir /mytemp

vxdg import dgdbapdmw1db

mount.vxfs -t vxfs /dev/vx/dsk/dgdbapdmw1db/lvdb /mytemp

vxupgrade -n 5 /mytemp

mount.vxfs -t vxfs /dev/vx/dsk/dgdbapdmw1db/lvdb /mytemp

vxupgrade -n 6 /mytemp

mount.vxfs -t vxfs /dev/vx/dsk/dgdbapdmw1db/lvdb /mytemp

vxupgrade -n 7 /mytemp

umount /mytemp

vxdg deport dbapdmw1

rmdir /mytemp

Don't forget to unmount the volumes and deport the diskgroup.

## Wait for DNS Push

check that the DNS push is through on **BOTH** install server and your target cluster nodes to reflect the new IP Address

lpinstbss1 ~ $ host ${PKG}

lpdbs92 ~ $ host ${PKG}  
##########How to define new IP of cluster package in bond0 configuration file######

=======================Section added For IP Assignment to Cluster Package=========================  
#nslookup package\_name  
#Check Ping status of cluster package (IP should not ping)  
#will perform steps mentioned below to assign the IP to cluster package

#df -hT | grep -i tdpalt79 (e.g. Cluster package name)  
#ping -c 4 tdpalt79   ( To check ping status of cluster package , this should not ping)  
#nslookup tdpalt79  ( To check IP of cluster package for assignment)  
#ifconfig | grep -i xxx.25.195.62  
#cd /etc/sysconfig/network  
#cat ifcfg-bond0  
#ifconfig -a ( To check all bond0 alias configuration )  
#ls -l \*bond0\*  
#cp -p ifcfg-bond0 old.ifcfg-bond0 ( Before addition of  new IP ..we have to copy old bond0 configuration file )  
#vi ifcfg-bond0  
#cat ifcfg-bond0

#tdpalt79 ( cluster package name) Label Should be any number as per configuration   
LABEL\_23=23 (Need to modify label as per bond0 configuration file)  
IPADDR\_23="xxx.25.195.62" (Need to modify as per bond0 configuration file)  
NETMASK\_23="255.255.255.0" (Need to modify as per bond0 configuration file)  
BROADCAST\_23="xxx.25.195.255" (Need to modify as per bond0 configuration file)  
  
#wicked ifup bond0 ( Execute To activate bond0 interface)  
ifconfig | grep -i xxx.25.195.62 ( Check IP assigned in ifconfig configuration OR not)  
ping 172.25.195.62 ( Check ping status )  
ping tdpalt79 ( check cluster package ping status)

================END ==========================

Last edited by Leonhaeuser Mirko, (Mirko.Leonhaeuser@partner.bmw.de) , based on work by Miko Andras, (Andras.Miko@partner.bmw.de) .  
Page last modified on Monday 27 of January, 2020 11:35:42 CET. (Version 2) , Added section How to define new IP of cluster package in bond0 configuration file~~-~~ -Sanjeev Arora, NTT

# [5.6.1.14 Standalone PKG move to VCS](https://atc.bmwgroup.net/confluence/display/IAAS18/5.6.1.14+Standalone+PKG+move+to+VCS)

* [Edit](https://atc.bmwgroup.net/confluence/pages/editpage.action?pageId=649689155)
* [Save for later](https://atc.bmwgroup.net/confluence)
* [Watch](https://atc.bmwgroup.net/confluence)
* [Share](https://atc.bmwgroup.net/confluence)

1. [Pages](https://atc.bmwgroup.net/confluence/collector/pages.action?key=IAAS18&src=breadcrumbs-collector)
2. **…**
3. [5.6.1 VCS - Veritas Cluster](https://atc.bmwgroup.net/confluence/display/IAAS18/5.6.1+VCS+-+Veritas+Cluster?src=breadcrumbs-parent)

* [[](https://atc.bmwgroup.net/confluence/plugins/confanalytics/analytics.action?key=IAAS18#/analytics/content/page/649689155)Analytics](https://atc.bmwgroup.net/confluence/plugins/confanalytics/analytics.action?key=IAAS18#/analytics/content/page/649689155)
* Created by Unknown User (qxy1271), last modified by [Kishore Kumar (ext.)](https://atc.bmwgroup.net/confluence/display/~qxz0crd) on [08 Apr 2020](https://atc.bmwgroup.net/confluence/pages/diffpagesbyversion.action?pageId=649689155&selectedPageVersions=3&selectedPageVersions=4)

# WSC

**Migration of Oracle Package from Standalone VxVM to a VCS Cluster**

# Preparation

### update san\_tools on target cluster nodes

**CHECK/UPDATE** to the **LATEST SAN\_TOOLS** on the target cluster nodes

### Request LUN Visibility for new Cluster nodes

Example:  
source server: lpdbsoem5  
oracle package: dboemao  
target cluster: lpdbs91 / lpdbs92

PKG=dboemao

SRC\_HOST=lpdbsoem5

# source server:

lpinstiaas01 ~ $ ssh ${SRC\_HOST} "sudo /sbin/san\_shortinfo" | egrep "${PKG}"

VSP\_10567\_01f0 | sdam 01:F0 5P 10567 518602 | sdcc 01:F0 6P 10567 518602 | dgdboemaodb

VSP\_10567\_01f1 | sdan 01:F1 5P 10567 518602 | sdcd 01:F1 6P 10567 518602 | dgdboemaodb

VSP\_10567\_01f2 | sdao 01:F2 5P 10567 518602 | sdce 01:F2 6P 10567 518602 | dgdboemaobkup

VSP\_10567\_01f3 | sdap 01:F3 5P 10567 14405 | sdcf 01:F3 6P 10567 14405 | dgdboemaoredo1

VSP\_10567\_01f4 | sdaq 01:F4 5P 10567 14405 | sdcg 01:F4 6P 10567 14405 | dgdboemaoredo2

VSP\_10582\_01f0 | sdbh 01:F0 6P 10582 518602 | sdr 01:F0 5P 10582 518602 | dgdboemaodb

VSP\_10582\_01f1 | sdbi 01:F1 6P 10582 518602 | sds 01:F1 5P 10582 518602 | dgdboemaodb

VSP\_10582\_01f2 | sdbj 01:F2 6P 10582 518602 | sdt 01:F2 5P 10582 518602 | dgdboemaobkup

VSP\_10582\_01f3 | sdbk 01:F3 6P 10582 14405 | sdu 01:F3 5P 10582 14405 | dgdboemaoredo1

VSP\_10582\_01f4 | sdbl 01:F4 6P 10582 14405 | sdv 01:F4 5P 10582 14405 | dgdboemaoredo2

### create CONFIGURE\_VERITAS template

lpinstiaas01 ~ # PKG=dboemao

lpinstiaas01 ~ # TASK=<task\_number>

lpinstiaas01 ~ # cd /global/instserv/data/CONFIGURE\_VERITAS

lpinstiaas01 /global/instserv/data/CONFIGURE\_VERITAS # ls -l INSTALL\_TEMPLATES | egrep -i oracle

-rw-r--r-- 1 root root 1975 Mar 29 2017 inst\_oracle.cluster

-rw-r--r-- 1 root root 2613 Mar 29 2017 inst\_oracle\_redo.cluster

-rw-r--r-- 1 root root 1836 May 6 2015 inst\_oracle\_redo.standalone

-rw-r--r-- 1 root root 1175 May 6 2015 inst\_oracle.standalone

## copy appropriate template

lpinstiaas01 /global/instserv/data/CONFIGURE\_VERITAS # cp INSTALL\_TEMPLATES/inst\_oracle\_redo.cluster ./${PKG}\_oracle\_redo\_${TASK}.cluster

### edit the file you just created:

# With vi replace with %s/(variable)/new\_value/g

#

# REPLACE (host1) with the physical IP-Name of the first node e.g. lttxtdb01

# REPLACE (host2) with the physical IP-Name of the second node e.g. lttxtdb02

# REPLACE (pkgname) with the virtual IP-Name e.g. itfqdb00

# REPLACE (pkgname\_ip) with the virtual IP e.g. 10.145.92.61

### MAKE sure you have the correct LUN's for each of the DG's

ssh ${TARGET\_NODE}

lpdbs92:~ # san\_shortinfo | egrep "dg${PKG}db"

hitachi\_vspg1k0\_01f0 | sdl 01:F0 8P 10567 518602 | sdb 01:F0 7P 10567 518602 | dgdboemaodb

hitachi\_vspg1k0\_01f1 | sdm 01:F1 8P 10567 518602 | sdc 01:F1 7P 10567 518602 | dgdboemaodb

hitachi\_vspg1k1\_01f0 | sdg 01:F0 7P 10582 518602 | sdq 01:F0 8P 10582 518602 | dgdboemaodb

hitachi\_vspg1k1\_01f1 | sdh 01:F1 7P 10582 518602 | sdr 01:F1 8P 10582 518602 | dgdboemaodb

lpdbs91:~ # san\_shortinfo | egrep "dg${PKG}bkup"

hitachi\_vspg1k0\_01f2 | sdn 01:F2 8P 10567 518602 | sdd 01:F2 7P 10567 518602 | dgdboemaobkup

hitachi\_vspg1k1\_01f2 | sdi 01:F2 7P 10582 518602 | sds 01:F2 8P 10582 518602 | dgdboemaobkup

lpdbs91:~ # san\_shortinfo | egrep "dg${PKG}redo1"

hitachi\_vspg1k0\_01f3 | sdo 01:F3 8P 10567 14405 | sde 01:F3 7P 10567 14405 | dgdboemaoredo1

hitachi\_vspg1k1\_01f3 | sdj 01:F3 7P 10582 14405 | sdt 01:F3 8P 10582 14405 | dgdboemaoredo1

lpdbs91:~ # san\_shortinfo | egrep "dg${PKG}redo2"

hitachi\_vspg1k0\_01f4 | sdp 01:F4 8P 10567 14405 | sdf 01:F4 7P 10567 14405 | dgdboemaoredo2

hitachi\_vspg1k1\_01f4 | sdk 01:F4 7P 10582 14405 | sdu 01:F4 8P 10582 14405 | dgdboemaoredo2

### Check VxVM Filesystem Layout Version

If you migrate the package from an older version to Veritas Storage Foundation Version 6 or higher, further steps are neccessary.  
  
Check and **note** the current version of the disk layout.

lpinstiaas01 ~ $ ssh ${SRC\_HOST}

lpdbsoem5 ~ $ sudo -s

lpdbsoem5:~ # PKG=dboemao

lpdbsoem5:~ # for VOL in $(df -hP -t vxfs | grep ${PKG} | awk '{print $1}'); do /opt/VRTS/bin/fstyp -v ${VOL} | grep -i version; done

magic a501fcf5 version 10 ctime Tue 23 May 2017 03:10:43 PM CEST

magic a501fcf5 version 10 ctime Tue 23 May 2017 03:10:48 PM CEST

magic a501fcf5 version 10 ctime Tue 23 May 2017 03:10:52 PM CEST

magic a501fcf5 version 10 ctime Tue 23 May 2017 03:10:56 PM CEST

### san\_rescan / check visibility

Once the storm order has been completed do a san\_rescan, and check the Visibility on **BOTH Target Cluster Nodes**

# Migrating the Package

### CMDB: check/assign new IP to Package

1. in the CMDB create the new **"Resource Group"** ${PKG} in the target **"Failover Cluster"** name NODE1\_NODE2
2. in the CMDB you will need to assign the new **Package IP** to the **"Resource Group"** created above

See [CMDB Ressource Group and Failover-Group Management](https://atc.bmwgroup.net/confluence/display/IAAS18/RG+Management)  
  
Use the following **checks** to see if the IP has been setup correctly:

lpinstiaas01 ~ $ PKG=dboemao

lpinstiaas01 ~ $ host ${PKG}

dboemao.bmwgroup.net is an alias for lpdbsoem5.bmwgroup.net. <<< you will need to delete this alias

lpdbsoem5.bmwgroup.net has address 160.46.136.212

lpinstiaas01 ~ $ /lfs/scripts/temp\_c6000 ${PKG}

.------------------------------------------------.

| dboemao-new |

+---------------+--------------------------------+

| key | value |

+---------------+--------------------------------+

| HOSTNAME | dboemao-new | <<< you will need to rename to ${PKG}

| DOMAIN | bmwgroup.net |

| LOG\_SERVER\_ID | |

| CLUSTER\_NAME | | <<< you will need to assign IP to Cluster Name

| ALIAS | |

| IP\_ADDRESS | 160.50.18.83 |

| NSI | |

| SUBNETMASK | 255.255.255.0 |

| GATEWAY | 160.50.18.1 |

| NET\_ID | 160.50.18.0 |

| BROADCAST | 160.50.18.255 |

| INT\_VLAN | |

| SUB\_AREA | MUC-HA-III |

'---------------+--------------------------------'

### umount, deport, and remove /etc/fstab entries

lpdbsoem5:~ # df -hP -t vxfs | egrep ${PKG}

/dev/vx/dsk/dgdboemaobkup/lvbkup 507G 997M 502G 1% /global/dboemao/bkup

/dev/vx/dsk/dgdboemaodb/lvdb 1013G 496G 513G 50% /global/dboemao/db

/dev/vx/dsk/dgdboemaoredo1/lvredo1 15G 208M 13G 2% /global/dboemao/redo1

/dev/vx/dsk/dgdboemaoredo2/lvredo2 15G 150M 14G 2% /global/dboemao/redo2

lpdbsoem5:~ # egrep ${PKG} /etc/fstab

/dev/vx/dsk/dgdboemaobkup/lvbkup /global/dboemao/bkup vxfs \_netdev 0 0

/dev/vx/dsk/dgdboemaodb/lvdb /global/dboemao/db vxfs mincache=direct,convosync=direct,\_netdev 0 0

/dev/vx/dsk/dgdboemaoredo1/lvredo1 /global/dboemao/redo1 vxfs mincache=direct,convosync=direct,\_netdev 0 0

/dev/vx/dsk/dgdboemaoredo2/lvredo2 /global/dboemao/redo2 vxfs mincache=direct,convosync=direct,\_netdev 0 0

lpdbsoem5:~ # umount /global/dboemao/db

lpdbsoem5:~ # umount /global/dboemao/bkup

lpdbsoem5:~ # umount /global/dboemao/redo1

lpdbsoem5:~ # umount /global/dboemao/redo2

lpdbsoem5:~ # for DG in $(egrep ${PKG} /etc/fstab | cut -f5 -d"/") ; do vxdg deport $DG ; done

lpdbsoem5:~ # vi /etc/fstab

## remove above entries...

## double check: should return nothing

lpdbsoem5:~ # egrep ${PKG} /etc/fstab

lpdbsoem5:~ #

### Update Disk Layout if needed

On the target cluster you will need to stepwise update the Veritas Disk layout to at least version 7.  
  
For example, if the Current Version is 4 then you will need to:  
  
import the diskgroup  
mount the volume in local mode (not cluster)  
stepwise update the Veritas disk layout versions (I think you will need to re-import and mount each time, because the update umounts and deports. ...)  
  
EXAMPLE: upgrading from Version 4 to Version 7: you will need to do the following for each volume of the cluster package.

mkdir /mytemp

vxdg import dgdbapdmw1db

mount.vxfs -t vxfs /dev/vx/dsk/dgdbapdmw1db/lvdb /mytemp

vxupgrade -n 5 /mytemp

mount.vxfs -t vxfs /dev/vx/dsk/dgdbapdmw1db/lvdb /mytemp

vxupgrade -n 6 /mytemp

mount.vxfs -t vxfs /dev/vx/dsk/dgdbapdmw1db/lvdb /mytemp

vxupgrade -n 7 /mytemp

umount /mytemp

vxdg deport dbapdmw1

rmdir /mytemp

Don't forget to unmount the volumes and deport the diskgroup.

### Wait for DNS Push

check that the DNS push is through on **BOTH** install server and your target cluster nodes to reflect the new IP Address

lpinstiaas01:~> host ${PKG}

lpdbs92 ~ $ host ${PKG}

### run configure\_veritas.sh

lpinstiaas01:~> instserv configure\_veritas.sh ./${PKG}\_oracle\_redo\_${TASK}.cluster

# check the results:

lpinstiaas01:~> ssh <target cluster node>

hastatus -sum | egrep ${PKG}

## state may be in "PARTIAL": so run:

hagrp -offline ${PKG} -sys ${NODE}

hagrp -online ${PKG} -sys ${NODE}

hastatus -sum | egrep ${PKG}

## Check connection to PKG IP

lpinstiaas01:~> ssh ${PKG}

## you should land on the new Cluster Node: check if the FS's are mounted

lpdbs92 ~ $ sudo df -hP | egrep ${PKG}

### Perform a cluster switch test

hagrp -switch ${PKG} -sys ${OTHER\_NODE}

hastatus -sum

hagrp -switch ${PKG} -sys ${TARGET\_NODE}

hastatus -sum

# CHECK LIST

1. CMDB: IP is connected to just the Cluster Name, and not to any Cluster Node: /lfs/scripts/temp\_c6000 ${PKG}
2. Switch Tests completed successfully
3. ssh to the ${PKG} and you land on the ${TARGET\_NODE}
4. ALL ${PKG} Filesystems are mounted
5. is there a cleanup task??

\*\* **YES:** add LUNs that will need "remove Visibility" to cleanup task notes  
\*\* **NO:** create remove Visibility Storm Order for old server

Last edited by Wild Thomas, (Thomas.Wild@partner.bmw.de) , based on work by Schrocko Winfried, (Winfried.Schrocko@partner.bmw.de) , Thomas Marcel, (Marcel.Thomas@partner.bmw.de) and Kemmerer Walter, (Walter.Kemmerer@partner.bmw.de) .  
Page last modified on Monday 03 of February, 2020 11:43:09 CET. (Version 16)

[Like](https://atc.bmwgroup.net/confluence/display/IAAS18/5.6.1.14+Standalone+PKG+move+to+VCS)Be the first to like this

* Created by [VijayKumarReddy Ayyaluri (ext.)](https://atc.bmwgroup.net/confluence/display/~qxy7563" \o "), last modified on [22 Jan 2023](https://atc.bmwgroup.net/confluence/pages/diffpagesbyversion.action?pageId=848558020&selectedPageVersions=5&selectedPageVersions=6)

# Background

BMW uses the term "package" to refer to all kinds of service or resource groups. A "package" in BMW terminology is a VCS service group in official VCS language. For the migration of package we have three task

1. Preparation Task
2. Execution Task
3. Cleanup Task

### Moving a cluster package includes:

* A) migrating its configuration to the new cluster
* B) moving the data associated with the package to the new cluster There are different ways to move the actual data from the old cluster to the new one. The easiest way is to migrate by re-assigning the existing data disks to the new cluster. However, this is only possible if old and new cluster systems are connected to the same SAN. If this is not the case, the data may either be copied by the middle-ware/application teams or in special cases the SAN team may create a temporary routing between two different SANs and copy the data from the old SAN to the new one. The latter option is similar to the data disk re-assignment procedure from the server point of view; however, instead of getting access to the original disks, the new cluster will get access to a copy of the original disks then.

A package is usually a fail-over service group, this means, it can only run on a single system at a particular point in time. The system or cluster node where the package usually runs is called the primary node.

**Preparations and Prerequisites:**

## Manual Method:

* Before a package to be migrated, the VCS cluster framework must have been installed on the new cluster nodes an the NIC service group for monitoring the network devices must have been defined on this cluster. This is usually the case before a package migration task is received, but it is always a good idea to check by running "hastatus -sum" on one of the new cluster nodes.

**!!NOTE!! :** If the destination cluster is running Veritas Cluster 6 / Veritas Storage Foundation Suite 6, you will need to update the VxFS disk layout versions. These steps are marked To VCS/VxVM 6 below in this document. In Veritas Release 6, you can create and mount only file systems with disk layout Version 7, 8, 9, and 10. You can only local mount disk layout Version 6 only to upgrade to a later disk layout version. Updates can only be performed  
**Caution!!!** : at migrations from newer veritas version to lower veritas version: It may happen that the disk groups have been updated to the latest version, so we can't import them on the target cluster with older veritas version. Always check the veritas version on the source and target cluster. You can use this one-liner to check the DG-version. The max. version for veritas 6.1 is 190, but the default DG version at veritas 7.3 is 240. If the version is not supported on the target cluster please inform the Change Manager, that the migration is not possible.

Check for the disk group version on source and target cluster nodes using below command.

#for x in $(vxdg -q list|awk {'print $1'});do echo -n $x " "; vxdg -q list $x|grep version;done

E.g:

lpcmdbd7:~ # for x in $(vxdg -q list|awk {'print $1'});do echo -n $x " "; vxdg -q list $x|grep version;done  
dgoracle\_lpcmdbd7 version: 240  
dgilcmdb03bkup version: 240  
dgilcmdb03db version: 240  
dgilcmdb03redo1 version: 240  
dgilcmdb03redo2 version: 240  
dgplcmdb01bkup version: 240  
dgplcmdb01db version: 240  
dgplcmdb01redo1 version: 240  
dgplcmdb01redo2 version: 240  
dgplcmdb02bkup version: 240  
dgplcmdb02db version: 240  
dgplcmdb02redo1 version: 240  
dgplcmdb02redo2 version: 240  
lpcmdbd7:~ #

**1.** Check for the resource groups running on source machines which need to be migrated by using below command of VCS:

#hastatus -sum

E.g:

lpcmdbd7:~ # hastatus -sum

-- SYSTEM STATE  
-- System State Frozen

A lpcmdbd7 RUNNING 0  
A lpcmdbd8 RUNNING 0

-- GROUP STATE  
-- Group System Probed AutoDisabled State

B NIC lpcmdbd7 Y N ONLINE  
B NIC lpcmdbd8 Y N ONLINE  
B ilcmdb03 lpcmdbd7 Y N ONLINE  
B ilcmdb03 lpcmdbd8 Y N OFFLINE  
B plcmdb01 lpcmdbd7 Y N ONLINE  
B plcmdb01 lpcmdbd8 Y N OFFLINE  
B plcmdb02 lpcmdbd7 Y N ONLINE  
B plcmdb02 lpcmdbd8 Y N OFFLINE  
lpcmdbd7:~ #

2.

A) Check if Secfs packages are installed on source servers, if yes, we need to install the sam SecFS packages on destination servers.

#hacf -cftocmd /etc/VRTSvcs/conf/config -display | egrep 'service\_group' |grep -i secfs

Example:

lpcmdbd7:~ # hacf -cftocmd /etc/VRTSvcs/conf/config -display | egrep 'plcmdb0[1|2]' |grep -i secfs  
lpcmdbd7:~ #

B) Check if any NFS/NAS share need to mounted on Destination servers. This should be mentioned in the task itself. Always check if these NFS/NAS shares are exported and available on Destination servers. if the NFS/NAS shares are \*\*not available\*\* kindly inform the **requester** of the task.

3. Request LUN Visibility for new Cluster nodes  
If a migration by disk (or "LUN") re-assignment is requested, the existing storage has to be made visible on the new cluster first. In order to accomplish this, a STORM order has to be created via [http://storm.muc](http://storm.muc/)(external link) using the sub-menu "Customer Cockpit > Assign Visibility".You can use the san\_shortinfo toolfor this.

san\_shortinfo | grep "<Service Group name>"

E.g

lpcmdbd7:~ # san\_shortinfo | egrep "plcmdb0[1|2]"  
hitachi\_vsp0\_009a | sdfl 00:9A 3F 86006 57622 | sdsb 00:9A 4F 86006 57622 | dgplcmdb01redo1  
hitachi\_vsp0\_009b | sdfm 00:9B 3F 86006 57622 | sdsc 00:9B 4F 86006 57622 | dgplcmdb01redo2  
hitachi\_vsp0\_00b7 | sdgo 00:B7 3F 86006 1048576 | sdte 00:B7 4F 86006 1048576 | dgplcmdb01bkup  
hitachi\_vsp0\_00b8 | sdgp 00:B8 3F 86006 1048576 | sdtf 00:B8 4F 86006 1048576 | dgplcmdb01bkup  
hitachi\_vsp0\_00b9 | sdgq 00:B9 3F 86006 1048576 | sdtg 00:B9 4F 86006 1048576 | dgplcmdb01bkup  
hitachi\_vsp0\_00ba | sdgr 00:BA 3F 86006 1048576 | sdth 00:BA 4F 86006 1048576 | dgplcmdb01bkup  
hitachi\_vsp0\_00d9 | sdhw 00:D9 3F 86006 518602 | sdum 00:D9 4F 86006 518602 | dgplcmdb01db  
hitachi\_vsp0\_00da | sdhx 00:DA 3F 86006 518602 | sdun 00:DA 4F 86006 518602 | dgplcmdb01db  
hitachi\_vsp0\_00db | sdhy 00:DB 3F 86006 518602 | sduo 00:DB 4F 86006 518602 | dgplcmdb01db  
hitachi\_vsp0\_00dc | sdhz 00:DC 3F 86006 518602 | sdup 00:DC 4F 86006 518602 | dgplcmdb01db  
hitachi\_vsp0\_00dd | sdia 00:DD 3F 86006 518602 | sduq 00:DD 4F 86006 518602 | dgplcmdb01db  
hitachi\_vsp0\_00de | sdib 00:DE 3F 86006 518602 | sdur 00:DE 4F 86006 518602 | dgplcmdb01db  
hitachi\_vsp0\_00df | sdic 00:DF 3F 86006 518602 | sdus 00:DF 4F 86006 518602 | dgplcmdb01db  
hitachi\_vsp0\_00e0 | sdid 00:E0 3F 86006 1048576 | sdut 00:E0 4F 86006 1048576 | dgplcmdb01bkup  
hitachi\_vsp1\_00e9 | sdou 00:E9 4F 86244 1048576 | sdce 00:E9 3F 86244 1048576 | dgplcmdb01bkup  
hitachi\_vsp1\_00ea | sdov 00:EA 4F 86244 1048576 | sdcf 00:EA 3F 86244 1048576 | dgplcmdb01bkup  
hitachi\_vsp1\_00ed | sdoy 00:ED 4F 86244 230490 | sdci 00:ED 3F 86244 230490 | dgplcmdb02db  
hitachi\_vsp1\_00ee | sdoz 00:EE 4F 86244 230490 | sdcj 00:EE 3F 86244 230490 | dgplcmdb02db  
hitachi\_vsp1\_00ef | sdpa 00:EF 4F 86244 230490 | sdck 00:EF 3F 86244 230490 | dgplcmdb02db  
hitachi\_vsp1\_00f0 | sdpb 00:F0 4F 86244 230490 | sdcl 00:F0 3F 86244 230490 | dgplcmdb02db  
hitachi\_vsp1\_00f1 | sdpc 00:F1 4F 86244 230490 | sdcm 00:F1 3F 86244 230490 | dgplcmdb02db  
hitachi\_vsp1\_00f2 | sdpd 00:F2 4F 86244 230490 | sdcn 00:F2 3F 86244 230490 | dgplcmdb02db

4.If LSAN zoning is "Yes" . We need to order fresh disk of same no. and size on the destination farm servers. You can use below command as reference for gathering the details:

#san\_shortinfo | grep "<Service Group name>" | awk '{print $7}'| sort | uniq -c

pcmdbd7:~ # san\_shortinfo | egrep "plcmdb0[1|2]" | awk '{print $7}'| sort | uniq -c  
 36 1048576  
 8 115245  
 16 14405  
 34 230490  
 18 518602  
 4 57622  
lpcmdbd7:~ #

We will Order disk in below number, as we have mirroring in place for the disks.You can use [autosan.pl](http://autosan.pl/) or STORM tool for ordering fresh disk.

So, 18 \* 1024 GB  
4 \* 112GB  
8 \* 14 GB  
17 \* 225 GB  
9 \* 506 GB  
2 \* 56 GB

5.Create the command set for the resource groups , which we will use in execution task for creating the service group. So , post creating the command set file on source server, copy it to destination servers.

# hacf -cftocmd /etc/VRTSvcs/conf/config -display | grep <service\_group name> > /var/tmp/<service\_group>.cmd

Example:  
lpcmdbd7:~ # hacf -cftocmd /etc/VRTSvcs/conf/config -display | grep plcmdb01 > /var/tmp/plcmdb01.cmd

Once you have created the command file for your resource groups and copied it to detaination servers.Login on Destination nodes and create the Mount points mentioned in the command files.

cat <reosurce\_group>.cmd |grep "MountPoint " | awk -F \" '{print $2}' | xargs mkdir -p

Eg.

cat plcmdb01.cmd|grep "MountPoint " | awk -F \" '{print $2}' | xargs mkdir -p

[lpsbcmdbd14:/var/tmp](http://lpsbcmdbd14/var/tmp) # ls -l /global/plcmdb01/  
/global/plcmdb01/:  
total 0  
drwxr-xr-x 1 root root 0 Jun 18 16:07 bkup  
drwxr-xr-x 1 root root 0 Jun 18 16:07 db  
drwxr-xr-x 1 root root 0 Jun 18 16:07 redo1  
drwxr-xr-x 1 root root 0 Jun 18 16:07 redo2

If Ip is getting changed for the resource group, you need to reserve the IP in CMDB and get that attached to resource group.  
Once IP is reserved, change the network entries accordingly in <service\_group.cmd> file, so that resources will be configured accordingly.

**Attention!! When migrating from VCS 7.4.1 to a higher VCS version!!!**

**The Network config is different and not compatible for the \_ip Resources and they must be modified \*manually\* before they are imported on the new system or they will fail to start and cause the RG to fail**

* We have two Methods to follow to start the Package on Target server [ where VCS version is higher than 7.4.1

==> Method 1 # Modify the .cmd file

After copying the .cmd file on Target servers Modify the Attributes

FROM:

|  |
| --- |
| lpdbfips001:/var/tmp # grep -i broadcast  plfasi01.cmd  hares -modify plfasi01\_ip Options "broadcast 10.30.39.255"  lpdbfips001:/var/tmp # vim plfasi01.cmd  // Edit the \_ip Options from " Options " to "  IPOptions "  hares -modify plfasi01\_ip IPOptions "broadcast 10.30.39.255" |

TO

|  |
| --- |
| lpdbfips001:/var/tmp # grep -i broadcast  plfasi01.cmd  hares -modify plfasi01\_ip IPOptions "broadcast 10.30.39.255"  lpdbfips001:/var/tmp # |

[OR](https://atc.bmwgroup.net/confluence/pages/createpage.action?spaceKey=IAAS18&title=OR&linkCreation=true&fromPageId=848558020)

==> Method 2 # [ Perform this Before onlining the Package from Partial state ]

When IMPORTING \*SLES12 VCS IP Resources to a SLES15 VCS 7.4.2\*  cluster, the following modification are required for the IP resource to start (or they will fail and cause a non successful switch):

 => **ADD** the new required value IPOptions value for the broadcast

**hares -modify** ***ilxics10\_ilxics10\_ip IPOptions "broadcast 160.48.100.255"***

 => **REMOVE** the old broadcast value in “Options”

**hares -modify ilxics10\_ilxics10\_ip Options ""**

6.Copy the start/stop/monitor scripts from source to destination servers in the same directory as we have in source servers.

Directory Name : /lfs/cluster/vcs  
Scripts name:  
<service\_group>\_monitor  
<service\_group>\_start  
<service\_group>\_stop

Please note that ownership and permission of the files and content should be identical on destination servers as on source.

Example:

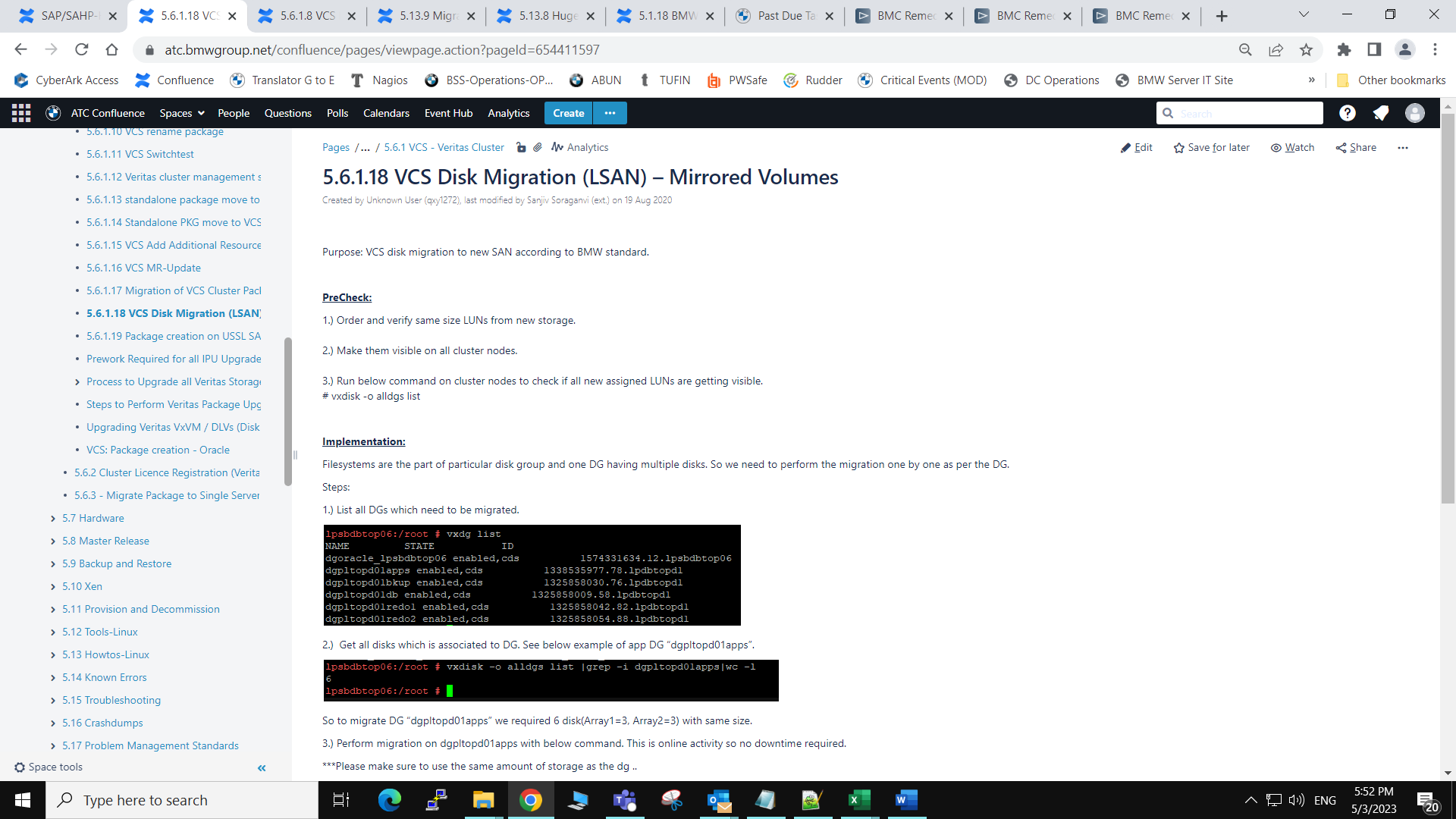
[lpcmdbd7:/lfs/cluster/vcs](http://lpcmdbd7/lfs/cluster/vcs) # ll  
total 64  
drwxr-xr-x 1 root root 344 Oct 7 2015 .  
drwxr-xr-x 1 root root 6 Sep 22 2015 ..  
-rwxr--r-- 1 root root 1381 Mar 27 2008 plcmdb01\_monitor  
-rwxr--r-- 1 root root 5884 Dec 15 2014 plcmdb01\_start  
-rwxr--r-- 1 root root 2914 Dec 15 2014 plcmdb01\_stop  
  
copy the start/stop/monitor scripts from Source to Target server  
NOTE: DO NOT CREATE THE FILES IN TARGET NODE  
[lpcmdbd7:/lfs/cluster/vcs](http://lpcmdbd7/lfs/cluster/vcs) # scp plcmdb0\* qxy1270@[lpsbcmdbd14:/var/tmp/](http://lpsbcmdbd14/var/tmp/)  
SSO Password:  
plcmdb01\_monitor 100% 1381 1.4KB/s 00:00  
plcmdb01\_start 100% 5884 5.8KB/s 00:00  
plcmdb01\_stop 100% 2914 2.9KB/s 00:00

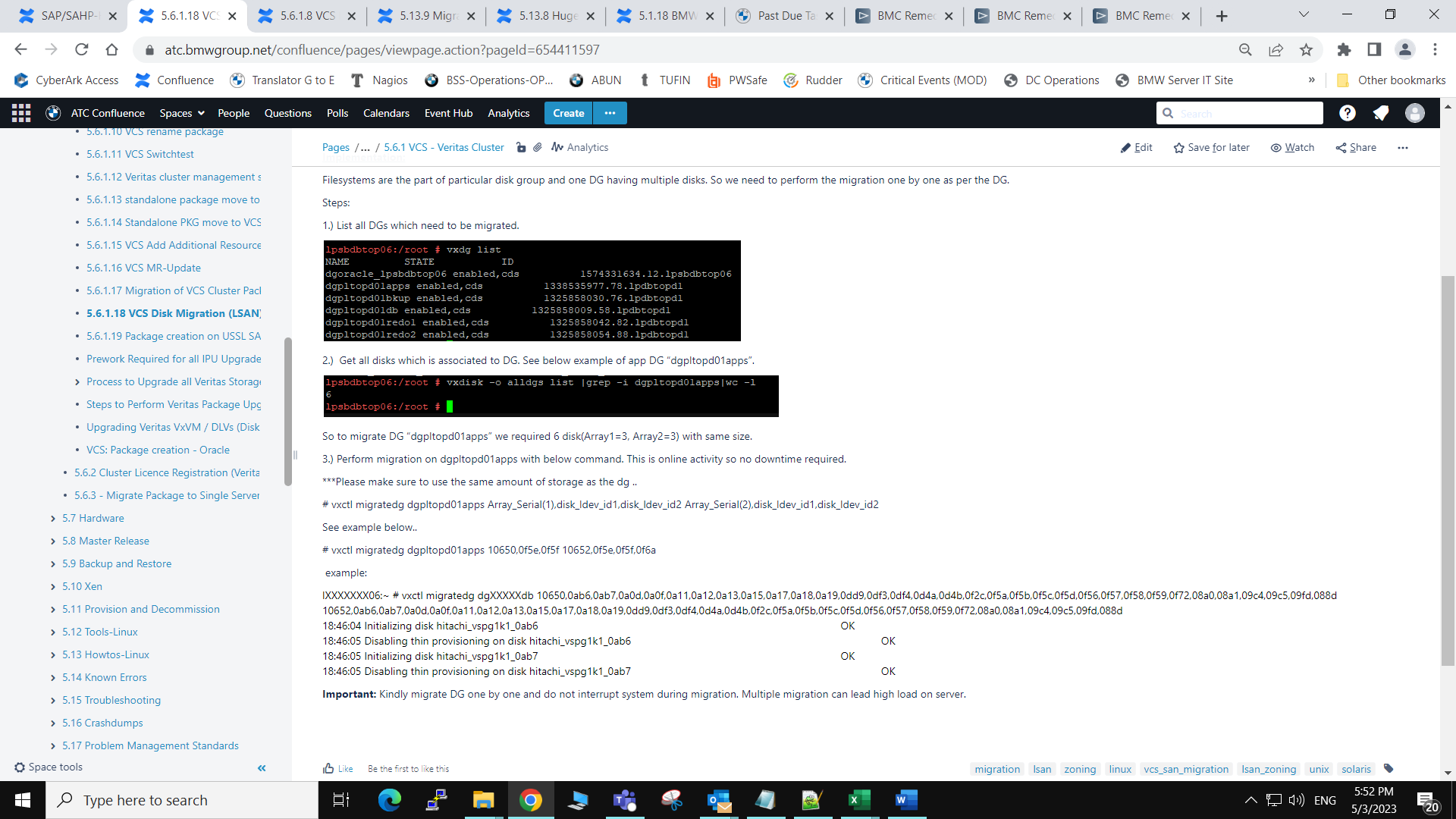
7. Check for the app\_control scripts files in "/lfs/cluster/app\_control/". If these files are present , copy these files on the destination servers.

E.g:

[lpcmdbd7:/lfs/cluster/vcs](http://lpcmdbd7/lfs/cluster/vcs) # cd /lfs/cluster/app\_control/  
bash: cd: /lfs/cluster/app\_control/: No such file or directory  
[lpcmdbd7:/lfs/cluster/vcs](http://lpcmdbd7/lfs/cluster/vcs) #

Now, your preparation task is completed and you can close it by mentioning the provision code as "DBGNONBILL" and attaching the CI to the TASK.





* Created by Unknown User (qxz13gk), last modified by [Gary Waterworth (ext.)](https://atc.bmwgroup.net/confluence/display/~qxz25l4) on [08 Nov 2022](https://atc.bmwgroup.net/confluence/pages/diffpagesbyversion.action?pageId=1189116547&selectedPageVersions=10&selectedPageVersions=11)

**7.1 => 7.4.1This documentation is for upgrading Veritas VxVM / DLVs (Disk Layout Versions) DGVs (Disk Group Versions). Please read the document carefully before starting the execution**

**Template:**server-unix-std:global|06|Post-SLES12-Veritas-Upgrade-Cleanup

**This is for the following scenarios:**

**1.  Veritas Patch Release or Master Release Upgrade  (with Server/Cluster Downtime).**

       During every MR and IPU update, all Veritas packages need to be updated to the latest versions available in the zypper respositories.

**2.  PRE IPU Upgrades or a Veritas Version Upgrade (when a newer Version of Veritas is installed e.g. VxVM 7.4.1 => 7.4.2 or 7.3.1 => 7.4.1):**

       - BEFORE the IPU or Veritas new Version install, upgrade to the highest DGV and DLV level supported with the \*old\* Veritas Version \*before\* the IPU upgrade

**CAUTION**: Do not upgrade to a higher version, a fall-back will not be possible since the DGs and LVOLS will not function with the older version!

**3.  PRE SAN Import Migrations and SLES12 Re-Installations:**

- BEFORE a SAN Import Migration or SLES12 Re-Install, upgrade to the highest DGV and DLV level supported by Veritas \*before\* the SAN Import Migration or Re-Install

**CAUTION**: Do not upgrade to a higher version, a fall-back will not be possible since the DGs and LVOLS will not function with the older version!

**4.  POST a SLES11=> SLES12 SAN Migration or Re-Installation: Task Template  server-unix-std:global|06|Post-SLES12-Veritas-Upgrade-Cleanup**

  - Since there is no Veritas DLV Version which is compatible between SLES11 Veritas 6.x and SLES12-SP4 Veritas 7.4.1+, this is for a POST SAN/Re-Install action to be done one week after the Migration/Re-Install.

                  The template can also be used for systems which were previously Migrated or Re-Installed, where the necessary DLV and GDV actions were not executed in the past.

                   In this task, no fall back is required and the DLV and DGV should be updated to the highest supported Version.

**How To check the current DLV and DGV Versions on a system:**

1. **Check the Veritas Support Matrix Table at the end of this document**
2. **Use the Script /global/instserv/bin/check\_dlv\_dgv.sh**

**To run a check with the script on e.g.** [**lpgip14.bmwgroup.net**](http://lpgip14.bmwgroup.net/)**:**

[lpinstiaas01]  $  rootscp **/global/instserv/bin/check\_dlv\_dgv.sh**  lpgip14.bmwgroup.[net:/var/tmp/](http://net/var/tmp/)

[lpinstiaas01]  $  rootssh [lpgip14.bmwgroup.net](http://lpgip14.bmwgroup.net/)

**The script will inform you which actions are necessary on the system:**

Checking host lpgip14 SLES Release 12-SP4 VxVM Version 7.4.1 PL: 1409

Supported VxVM Version 7.4.1 PL 1409 on lpgip14

VxVM 7.4.1 supports: DLV 11-15

VxVM 7.4.1 supports: DGV MAX 280

Checking DiskLayout DLVs .. 7.4.1 Max supported DLV is 15

LV /dev/vx/dsk/dgpgipcs0001/lvapps has DLV 13 ERROR NOK: UPDATE TO DLV 15

LV /dev/vx/dsk/dgpgipcs0001/lvsapGIP has DLV 13 ERROR NOK: UPDATE TO DLV 15

LV /dev/vx/dsk/dgpgipcs0002/lvbkup has DLV 13 ERROR NOK: UPDATE TO DLV 15

LV /dev/vx/dsk/dgpgipcs0003/lvmirrlogB has DLV 13 ERROR NOK: UPDATE TO DLV 15

LV /dev/vx/dsk/dgpgipcs0003/lvoriglogA has DLV 13 ERROR NOK: UPDATE TO DLV 15

LV /dev/vx/dsk/dgpgipcs0004/lvmirrlogA has DLV 13 ERROR NOK: UPDATE TO DLV 15

LV /dev/vx/dsk/dgpgipcs0004/lvoriglogB has DLV 13 ERROR NOK: UPDATE TO DLV 15

LV /dev/vx/dsk/dgpgipcs0005/lvGIP has DLV 13 ERROR NOK: UPDATE TO DLV 15

LV /dev/vx/dsk/dgpgipcs0006/lvsapdata1 has DLV 13 ERROR NOK: UPDATE TO DLV 15

LV /dev/vx/dsk/dgpgipcs0007/lvsapdata2 has DLV 13 ERROR NOK: UPDATE TO DLV 15

LV /dev/vx/dsk/dgpgipcs0008/lvsapdata3 has DLV 13 ERROR NOK: UPDATE TO DLV 15

LV /dev/vx/dsk/dgpgipcs0009/lvsapdata4 has DLV 13 ERROR NOK: UPDATE TO DLV 15

LV /dev/vx/dsk/dgpgipcs0010/lvsapdata5 has DLV 13 ERROR NOK: UPDATE TO DLV 15

LV /dev/vx/dsk/dgpgipcs0011/lvsapdata6 has DLV 13 ERROR NOK: UPDATE TO DLV 15

LV /dev/vx/dsk/dgpgipcs0012/lvsapdata7 has DLV 13 ERROR NOK: UPDATE TO DLV 15

LV /dev/vx/dsk/dgpgipcs0013/lvsapdata8 has max DLV 15 OK

Checking DiskGroup DGVs .. 7.4.1 Max supported DGV is 280

DG dgpgipcs0001 has DGV 240 ERROR NOK: UPDATE TO DGV 280

DG dgpgipcs0002 has DGV 240 ERROR NOK: UPDATE TO DGV 280

DG dgpgipcs0003 has DGV 240 ERROR NOK: UPDATE TO DGV 280

DG dgpgipcs0004 has DGV 240 ERROR NOK: UPDATE TO DGV 280

DG dgpgipcs0005 has DGV 240 ERROR NOK: UPDATE TO DGV 280

DG dgpgipcs0006 has DGV 240 ERROR NOK: UPDATE TO DGV 280

DG dgpgipcs0007 has DGV 240 ERROR NOK: UPDATE TO DGV 280

DG dgpgipcs0008 has DGV 240 ERROR NOK: UPDATE TO DGV 280

DG dgpgipcs0009 has DGV 240 ERROR NOK: UPDATE TO DGV 280

DG dgpgipcs0010 has DGV 240 ERROR NOK: UPDATE TO DGV 280

DG dgpgipcs0011 has DGV 240 ERROR NOK: UPDATE TO DGV 280

DG dgpgipcs0012 has DGV 240 ERROR NOK: UPDATE TO DGV 280

DG dgpgipcs0013 has max DGV 280 OK

**Extra Patch Level Check for all versions**

Although a version check is done all IPU's and MR updates need to have ALL installed veritas packages updated to the latest available with the Zypper repositories.

This can be checked separately by running "zypper lu"

**So for the lpgip14 the following actions would be necessary:**

* Patch All Veritas Packages to the latest packages available in the Zypper repositories
* Upgrade all LVOLs to DLV 15 except the one LVOL which already is (lvsapdata8)
* Upgrade all DGs to DGV 280 except for the one DG which already is (dgpgipcs0013)

* **How to Perform the Upgrades:**
* **A) Check the VXVM patch level DLV and DGV version's before upgrading them on the server (check\_dlv\_dgv.sh)**
* **B) Always Set downtime for the server(s) when the servers or resource groups are to be offlined**
* **C) Make sure SAP or Oracle is stopped and confirmation is received from SAP team to proceed ahead. If a cluster offline the resource Groups.**
* **D) Upgrade the Veritas Packages available on the Zypper repository to the latest version available.**

IAAS instructions: [**Steps to Perform Veritas Package Upgrade and MR Update**](https://atc.bmwgroup.net/confluence/display/IAAS18/Steps+to+Perform+Veritas+Package+Upgrade+and+MR+Update)

**E) Upgrade the DGV Versions of the Disk Groups if required in the check above:**

           a) Manually import the DGs which require a DGV (or DVL) upgrade:

         # vxdg import <DG\_NAME>

         Then check with

         # vxdg -q list <DG\_NAME> | egrep -i ^version:| awk '{print $NF}'

         b) Upgrade the DGV version to the highest supported level supported by this Veritas Version (as in the output form  (check\_dlv\_dgv.sh)

**IMPORTANT**: this is done on the Veritas Version BEFORE a IPU, SAN IMPORT or SLES12 Re-Install. Without the -T option, it will upgrade the DGs to the highest supported level of that Veritas version.

         # vxdg upgrade <DG\_NAME>

         Then check with

         # vxdg -q list <DG\_NAME> | egrep -i ^version:| awk '{print $NF}'

If you have many DGs to update, you can use a “for” statement and check at the same time, all should have the highest supported version:

# for DG in dgptupcs00cs02 dgptupcs00cs10 dgptupcs00db02 dgptupcs00db03 dgptupcs00db04 dgptupcs00db05 dgptupcs00db06 dgptupcs00db07 dgptupcs00db10 ; do

   echo "Upgrading $DG"

             vxdg upgrade $DG && sleep 1 && vxdg -q list $DG | egrep -i ^version:| awk '{print "Version: "$NF}'

       done

**F) Upgrade the DLV Version (the Disk Groups must be imported for this):**

1. Create a temporary directory  e.g. # mkdir  /DLVUPGRADE
2. Mount the first vxfs FS in the temporary directory   e.g. # mount /dev/vx/dsk/dgpgipcs0001/lvapps   /DLVUPGRADE )
3. Now upgrade the DLV Version for the vxfs you just mounted. This can \***only**\* be done sequentially .

In the example above lvapps is DLV 10 and must be upgraded to DLV 15:

LV /dev/vx/dsk/dgpgipcs0001/lvapps has DLV 13 ERROR NOK: UPDATE TO DLV 15

This can be done with a “for” statement to save time:

# for DLV in 11 12 13 14 15 ; do

vxupgrade -n $DLV /DLVUPGRADE  &&  sleep 1 && sync -f  /DLVUPGRADE

done

Check that the LVOL is now at the required level and umount when OK:

# /opt/VRTS/bin/fstyp -v /dev/vx/dsk/dgpgipcs0001/lvapps | egrep version

# umount  /DLVUPGRADE

**Important:** Do this for each LVOL which requires an Upgrade (make sure to start with the correct version, some systems have a mix of different DLV versions).

**G)  Continue with the Next step in the Task**

1. If this is an IPU Upgrade, commence with the IPU upgrade here.
2. If this is a SAN Migration start the SAN migration
3. If this is a SLES12 Re-install, commence with the SLES12 Re-install task
4. If no other actions were planned (i.e. TASK server-unix-std:global|06|Post-SLES12-Veritas-Upgrade-Cleanup):
   * If a Standalone, remount the Filesystems (mount -a -t vxfs )
   * If a VCS Cluster, Restart the Resource Groups (hagrp -online)

**H)  Details about the required Veritas VxVM Patchlevel, DLV and DGV Versions:**

**1. Update All Veritas Packages as per the procedure in**

[**Steps to Perform Veritas Package Upgrade and MR Update**](https://atc.bmwgroup.net/confluence/display/IAAS18/Steps+to+Perform+Veritas+Package+Upgrade+and+MR+Update)

**2. Veritas Support Matrix for Disk Layout (DLV )Versions:**

############################################################################################

# DLV Support Installed VxVM Version: Source <https://sort.veritas.com/dgfs_matrix/fs_matrix>

# 5.1 (lpcaps3/4) supprts up to DLV 7? No longer documented @ Semantec

# 6.0 supports DLV 7-9 & Supports Mount only for vxupgrade minimum DLV 6

**# 6.1 supports DLV 7-10 & Supports Mount only for vxupgrade minimum DLV 6  (BMW SLES11-SP4 default, has a max DLV 10!)**

# 6.2 supports DLV 7-10 & Supports Mount only for vxupgrade minimum DLV 6

# 7.0 supports DLV 7-10 & Supports Mount only for vxupgrade minimum DLV 6

# 7.1 supports DLV 7-11 & Supports Mount only for vxupgrade minimum DLV 6

# 7.2 supports DLV 9-12 & Supports Mount only for vxupgrade  DLVs 6-8

# 7.3 supports DLV 9-12 & Supports Mount only for vxupgrade  DLVs 6-8

**# 7.3.1+ supports DLV 9-13 & Supports Mount only for for vxupgrade DLVs 6-8    (BMW SLES12-SP2 default)**

# 7.4.0 supports DLV 10-14 (not used @ BMW) & Supports Mount only for vxupgrade DLVs 6-9

**# 7.4.1 supports DLV 11-15 & Supports Mount only for vxupgrade DLVs 6-10          (BMW SLES12-SP4 default, DLV 10 from SLES11 is not officially supported)**

# 7.4.2 supports DLV 12-16 & Supports Mount only for vxupgrade DLVs 6-11

# 7.4.3 supports DLV 12-16 & Supports Mount only for vxupgrade DLVs 6-11

#

############################################################################################

**3.  Veritas Support Matrix for Disk Group DGV) Versions:**

############################################################################################

# DGV Support Installed VxVM Version: Source <https://sort.veritas.com/dgfs_matrix/dg_matrix>

# NOTE: VxVM 7.4.1 (and possibly lower) cannot modify/extend DGVs less than DG Version 160!

#       All VxVM Versions fully support the use of (but not modification of) all previous

#       DGV versions.

#       NOTE: Some VxVM Version infos are missing in the Veritas link, known deviations have been

#             added below.

#

# 4.0     Supports DGVs up to max      110

# 4.1     Supports DGVs up to max      120

# 5.0     Supports DGVs up to max      130

# 5.0     Supports DGVs up to max      140

# 5.1     Supports DGVs up to max      150   NOTE: Versions 150 and earlier cannot be extended with new disks on VxVM 7.x systems until updated!

# 5.1SP1  Supports DGVs up to max      160

# 6.0     Supports DGVs up to max      170  [DCO version 30 CAUTION larger DCO LOG SPACE REQUIRED]

# 6.0.1   Supports DGVs up to max      180

**# 6.1     Supports DGVs up to max      190**

# 6.2     Supports DGVs up to max      200

# 7.0     Supports DGVs up to max      200

# 7.1     Supports DGVs up to max      220

# 7.2     Supports DGVs up to max      230

# 7.3     Supports DGVs up to max      230

**# 7.3.1   Supports DGVs up to max      240**

# 7.4     Supports DGVs up to max      260

**# 7.4.1   Supports DGVs up to max      280**

# 7.4.2   Supports DGVs up to max      290

# 7.4.3   Supports DGVs up to max      300

#

############################################################################################

* Created by [Dieter Klingbeil (FG-832RG)](https://atc.bmwgroup.net/confluence/display/~q154334), last modified by [VijayKumarReddy Ayyaluri (ext.)](https://atc.bmwgroup.net/confluence/display/~qxy7563" \o ") on [13 Sep 2022](https://atc.bmwgroup.net/confluence/pages/diffpagesbyversion.action?pageId=2606506468&selectedPageVersions=8&selectedPageVersions=9)

To be completed :

→ Template-Names to follow etc...

Below Document explains to Migrate Package from cluster to standalone [single server](https://atc.bmwgroup.net/confluence/pages/createpage.action?spaceKey=IAAS18&title=single+server&linkCreation=true&fromPageId=2606506468)

we used below details for illustration

**Package Name**:  tdkiqaw2

**Cluster [ Source ] :** lt02db001 / lt02db002

**Target  [ standalone /single ]** : li02dbipsq001

Cluster2single-prepare-migrate-package

* LUN-Visibility / LSAN-Zoning
* IP-Reservation -- Each moved package will have a new IP on the target-server.
* NFS-Shares
* Have all the “Rquired-Information” in here

Update Preperation to VCS/VxVM Filesystem Layout Version 6

If you migrate the package from an older version to Veritas VCS Version 6 and/or Veritas Storage Foundation Version 6, further steps are neccessary.

Check and note the current version of the disk layout.

PACKG=dbapdmw1  
for VOL in $(df -hP -t vxfs | grep ${PACKG} | awk '{print $1}'); do /opt/VRTS/bin/fstyp -v ${VOL} | grep -i version; done  
magic a501fcf5  version 6  ctime Sat 06 Sep 2008 01:29:46 PM CEST  
magic a501fcf5  version 6  ctime Sat 06 Sep 2008 01:29:13 PM CEST  
magic a501fcf5  version 6  ctime Sat 06 Sep 2008 01:28:48 PM CEST  
magic a501fcf5  version 6  ctime Sat 06 Sep 2008 01:29:32 PM CEST  
Example for a package named dbapdmw1

Cluster2single-migrate-package:

* Have at “required Information only “Source + Destination” + Task – Numbers of Prep

Execution :

**NOTE:** Is it absolutely essential that the package cannot be started on the old cluster any more after it has been migrated. Simultaneous start on both clusters will very likely result in data corruption! For this reason, the service group is frozen on the old cluster and removed permanently after a grace period of a few days.  
  
Check if the Preparation Task is completed and closed successfully

* Evaluate the Prep Task [ such as Storage Visibility / IP reservation and Hardware issues ]

### Check SecFS

Check, whether the package has SecFS resources:

SUORCE\_NODE:~ $ rpm -qa | grep -i vee

SUORCE\_NODE:~ $

If yes, check, whether SecFS is installed on the target cluster. (See the installation instructions depending on OS-Systems)

Set Downtime for the Source and Target servers to avoid Incident alerts during activity

|  |
| --- |
| lpinstiaas02# /lfs/scripts/set\_downtime <source node 1> -s os\_linux\_cluster\_veritas\_check\_vcs -m 60  lpinstiaas02# /lfs/scripts/set\_downtime <source node 1> -s os\_linux\_cluster\_veritas\_check\_health -m 60  lpinstiaas02# /lfs/scripts/set\_downtime <source node 2> -s os\_linux\_cluster\_veritas\_check\_vcs -m 60  lpinstiaas02# /lfs/scripts/set\_downtime <source node 2> -s os\_linux\_cluster\_veritas\_check\_health -m 60  lpinstiaas02# /lfs/scripts/set\_downtime <target system> -s os\_linux\_fs\_check\_san\_mirror -m 60  lpinstiaas02# /lfs/scripts/set\_downtime <target system> -s os\_linux\_fs\_check\_vxvm -m 60 |

Get Confirmation from oracle on Package status and Proceed with Execution Implementation further

Stop Package on Source Cluster

|  |
| --- |
| lt02db002:~ # hastatus -sum | grep -i tdkiqaw2  B  tdkiqaw2        lt02db001            Y          N               OFFLINE  B  tdkiqaw2        lt02db002            Y          N               ONLINE  lt02db002:~ # hagrp -offline tdkiqaw2 -sys lt02db002  lt02db002:~ # hastatus -sum | grep -i tdkiqaw2  B  tdkiqaw2        lt02db001            Y          N               OFFLINE  B  tdkiqaw2        lt02db002            Y          N               OFFLINE  lt02db002:~ # |

- Freeze Package on Source Cluster

|  |
| --- |
| lt02db002:~ # hagrp -freeze tdkiqaw2 -persistent  VCS WARNING V-16-1-11335 Configuration must be ReadWrite : Use haconf -makerw  lt02db002:~ # haconf -makerw  lt02db002:~ # hagrp -freeze tdkiqaw2 -persistent  lt02db002:~ # haconf -dump -makero  lt02db002:~ # hastatus -sum | grep -i tdkiqaw2  B  tdkiqaw2        lt02db001            Y          N               OFFLINE  B  tdkiqaw2        lt02db002            Y          N               OFFLINE  C  tdkiqaw2  H  tdkiqaw2        Application     tdkiqaw2\_app  H  tdkiqaw2        DiskGroup       tdkiqaw2\_dgtdkiqaw2bkup  H  tdkiqaw2        DiskGroup       tdkiqaw2\_dgtdkiqaw2db  H  tdkiqaw2        DiskGroup       tdkiqaw2\_dgtdkiqaw2redo1  H  tdkiqaw2        DiskGroup       tdkiqaw2\_dgtdkiqaw2redo2  H  tdkiqaw2        IP              tdkiqaw2\_ip  H  tdkiqaw2        Mount           tdkiqaw2\_bkup\_mount  H  tdkiqaw2        Mount           tdkiqaw2\_db\_mount  H  tdkiqaw2        Mount           tdkiqaw2\_redo1\_mount  H  tdkiqaw2        Mount           tdkiqaw2\_redo2\_mount  H  tdkiqaw2        Proxy           tdkiqaw2\_nic\_proxy  lt02db002:~ # |

- Add IPs on target

|  |
| --- |
| li02dbipsq001:~ # cat /etc/sysconfig/network/ifcfg-bond0 | grep -A7 tdkiqaw2  #tdkiqaw2  LABEL\_3=3  IPADDR\_3="160.47.10.131"  NETMASK\_3="255.255.255.0"  NETWORK\_3="160.47.10.0"  BROADCAST\_3="160.47.10.255"  li02dbipsq001:~ # |

- Create mountpoints + fstab – entries on target

|  |
| --- |
| li02dbipsq001:~ # ls -ld /global/tdkiqaw2/redo1 /global/tdkiqaw2/redo2 /global/tdkiqaw2/db /global/tdkiqaw2/bkup  drwxr-xr-x 12 oracle dba 8192 Aug 31 09:57 /global/tdkiqaw2/bkup  drwxr-xr-x  9 oracle dba 8192 Jul 20  2020 /global/tdkiqaw2/db  drwxr-xr-x  5 oracle dba 1024 Jul 20  2020 /global/tdkiqaw2/redo1  drwxr-xr-x  5 oracle dba 1024 Jul 20  2020 /global/tdkiqaw2/redo2  li02dbipsq001:~ #    li02dbipsq001:~ # grep -i tdkiqaw2 /etc/fstab  /dev/vx/dsk/dgtdkiqaw2bkup/lvbkup /global/tdkiqaw2/bkup vxfs \_netdev,nofail,x-systemd.device-timeout=20 0 0  /dev/vx/dsk/dgtdkiqaw2db/lvdb /global/tdkiqaw2/db vxfs mincache=direct,convosync=direct,\_netdev,nofail,x-systemd.device-timeout=20 0 0  /dev/vx/dsk/dgtdkiqaw2redo1/lvredo1 /global/tdkiqaw2/redo1 vxfs mincache=direct,convosync=direct,\_netdev,nofail,x-systemd.device-timeout=20 0 0  /dev/vx/dsk/dgtdkiqaw2redo2/lvredo2 /global/tdkiqaw2/redo2 vxfs mincache=direct,convosync=direct,\_netdev,nofail,x-systemd.device-timeout=20 0 0 |

- import veritas – disc-groups manually

|  |
| --- |
| li02dbipsq001:~ # vxdg **import** dgtdkiqaw2bkup  li02dbipsq001:~ # vxdg **import** dgtdkiqaw2db  li02dbipsq001:~ # vxdg **import** dgtdkiqaw2redo1  li02dbipsq001:~ # vxdg **import** dgtdkiqaw2redo2 |
| li02dbipsq001:~ # mount /global/tdkiqaw2/bkup  li02dbipsq001:~ # mount /global/tdkiqaw2/db  li02dbipsq001:~ # mount /global/tdkiqaw2/redo1  li02dbipsq001:~ # mount /global/tdkiqaw2/redo2    li02dbipsq001:~ # df -hT | grep -i tdkiqaw2  /dev/vx/dsk/dgtdkiqaw2bkup/lvbkup                vxfs      2.5T  1.3T  1.2T  53% /global/tdkiqaw2/bkup  /dev/vx/dsk/dgtdkiqaw2db/lvdb                    vxfs      2.6T  1.6T 1016G  61% /global/tdkiqaw2/db  /dev/vx/dsk/dgtdkiqaw2redo1/lvredo1              vxfs      3.6G  2.9G  637M  83% /global/tdkiqaw2/redo1  /dev/vx/dsk/dgtdkiqaw2redo2/lvredo2              vxfs      3.6G  2.4G  1.1G  70% /global/tdkiqaw2/redo2  li02dbipsq001:~ # |

- Migrate Luns / NFS-Shares

- rename scripts at /lfs/cluster/vcs – if any of this package exist

- Update CMDB entries for IPs

Cluster2single-migrate-cleanup

* Have at “required Information only “Source + Destination” + Task – Numbers of Prep + execution – ‘Task

Actions by OS Operations:

- Remove Package Config on old Cluster

- Remove LUN Visibility on old Cluster (Storm Tool)

- Remove old IP Address (if applicable)

- Umount NFS Shares on old Cluster if requested

- CMDB cleanup -> Delete RG;

- document in Task what you deleted

Input:

Source-Cluster

Destination-Single-Server

Package-Name

If source and target are in the same subnet : the same ip will be used

For all 3 :

Remove:

---- User

User IDs to move:

Remove User IDs at Cleanup: <Y / N >

* Created by [Kishore Kumar (ext.)](https://atc.bmwgroup.net/confluence/display/~qxz0crd), last modified by [VijayKumarReddy Ayyaluri (ext.)](https://atc.bmwgroup.net/confluence/display/~qxy7563" \o ") on [02 May 2023](https://atc.bmwgroup.net/confluence/pages/diffpagesbyversion.action?pageId=524183606&selectedPageVersions=6&selectedPageVersions=7)

This documentation **does not** apply for **virtual** **servers**

| Scope |
| --- |

This is for returning a "Package" Service Group Resource.  
When removing Service Group Resources be **very** careful and check your output before running any delete or destroy command closely. You have to ensure that you are not selecting any other Service Group resources or the System NIC resources!

| Conent |
| --- |

Formal prerequisites

* Who may open this kind of task? Anyone who consulted with the application team or the server owner.
* Which lead time applies? The lead time is five days for task execution on sunday or in the night, and the lead time is one day for other task execution times.
* Which task duration applies? We need at least one hour, but this is often not enough. So please consider the hour as an absolute minimum in the best possible case. If a safety pause is requested in the task, the task duration has to be long enough to cover both, the time occupied by the safety pause and the actual time to do the actual work.

Technical prerequisites

* **Supported OS**: SLES 9 / SLES 10 / SLES 11 / SLES 12 / SLES 15
* **Supported hardware:**All physical servers [ do not follow this documentation, if you are working on virtual server ]
* The server must have a valid host name, which can be resolved over DNS
* The server should reachable via ssh

Check before you start

Please have a close look into the task if an offline safety pause has been requested. If this is the case, you have to wait for the requested time (normally 5 days) between the package was taken offline and the actual removal of the package.  
If there was no safety pause requested, the actual removal can be started without waiting.

Step-by-step instructions

**Important:** All steps need to be documented closely in the task.

Collect and save data

Before you take the package offline, you collect data needed in later stages of this task or needed in case you get in trouble and want to restore the previous state.

The below examples assume, that you are logged in with root rights into one of the cluster nodes.

**Do not start the package, if it is already offline!** You will find some hints below how to avoid taking a package online.

* Store the package name in a variable to avoid typos:

|  |
| --- |
| vcsnode1:/root # PKGNAME=<**package** name>  vcsnode1:/root # echo PKGNAME=${PKGNAME} | tee /var/tmp/pkgremoval-${PKGNAME} |

* Get the current state of the package:

|  |
| --- |
| vcsnode1:/root # hastatus -sum | sed -e "s/ \*$//g" | grep -i ${PKGNAME} | tee -a /var/tmp/pkgremoval-${PKGNAME} |

* Get the IP of the package:  
  The IP address should have been noted in the task by the requester. If it is missing, you get it with this command:

|  |
| --- |
| vcsnode1:/root # hares -value ${PKGNAME}\_ip Address | tee -a /var/tmp/pkgremoval-${PKGNAME} |

* There is an alternative way to find the IP address if you don't know the name of the IP resource:

|  |
| --- |
| vcsnode1:/root # hagrp -resources ${PKGNAME} | egrep \_ip |

* Collect information for resource group removal, disk group removal and storage return:

|  |
| --- |
| vcsnode1:/root # san\_shortinfo | egrep ${PKGNAME} | tee -a /var/tmp/pkgremoval-shortinfo-${PKGNAME}  vcsnode1:/root # awk '{print $15}' /var/tmp/pkgremoval-shortinfo-${PKGNAME} | sort -u | tee /var/tmp/pkgremoval-dgnames-${PKGNAME}  vcsnode1:/root # awk '{print $6 " " $4}' /var/tmp/pkgremoval-shortinfo-${PKGNAME} | tee /var/tmp/pkgremoval-LDEV-BOX-pairs-${PKGNAME}  vcsnode1:/root # hagrp -resources ${PKGNAME} | tee /var/tmp/pkgremoval-resources-${PKGNAME} |

* We use san\_shortinfo to get a list of LUNs to be returned for our SAN return ticket in StorM (stored in /var/tmp/pkgremoval-shortinfo-${PKGNAME}), a list of disk groups involved (stored in /var/tmp/pkgremoval-dgnames-${PKGNAME}) and a list of LUNs with the respective SAN box for the san\_return script (stored in /var/tmp/pkgremoval-LDEV-BOX-pairs-${PKGNAME}). We further saved all resources (stored in /var/tmp/pkgremoval-resources-${PKGNAME}).
* Get a list of mountpoints for later cleanup:

|  |
| --- |
| vcsnode1:/root # hares -display -attribute MountPoint -group ${PKGNAME} | tee /var/tmp/pkgremoval-mountpoints-${PKGNAME} |

Find dependencies

Sometimes your package consists of multiple resource groups which depend on each other. To find this depencies you use:

vcsnode1:/root # hagrp -dep | tee -a /var/tmp/pkgremoval-${PKGNAME}

Take resource groups offline

We can now take the groups offline one by one in the correct order as dictated by the depencies. Please replace ${NODE} by the cluster node, where the resource group is running at the moment. (You can consult /var/tmp/pkgremoval-${PKGNAME} for this, remind the hastatus command at task start?)

vcsnode1:/root # hagrp -offline ${PKGNAME} -sys ${NODE}

Freeze resource groups

Make the cluster configuration readable and freeze the resource group.

vcsnode1:/root # haconf -makerw

vcsnode1:/root # hagrp -freeze ${PKGNAME} -persistent

Make a safety pause

This is the right moment to make a safety pause. The groups are offline, but can easily be restarted in case of trouble. But you don't want the cluster to linger around with a read-write configuration. So **please check**, if there is a request to make a safety pause and bring the cluster in a consistent state in this case:

vcsnode1:/root # haconf -dump -makero

When safety pause is over, you can make the configuration writeable again with:

vcsnode1:/root # haconf -makerw

vcsnode1:/root # hastatus -sum | sed -e "s/ \*$//g" | grep -i ${PKGNAME}

At this point you need to assure that noone brought the package online again. Please keep in mind, that the name of the resource in DNS might have been moved to another package in the meantime. If you are not absolutely sure what happened in the safety pause, repeat the data collecting steps above with some modified file names and compare the results. Should there be anything strange or unexpected, it's a good idea to ask the task requester for further information.

Remove group dependencies

You retrieved the depencies with the hagrp -dep command above. Now you need to remove the depencies that affect the resources you are to decommission. Please replace the variables ${PARENT} and ${CHILD} by the appropriate resource group names. The ${PARENT} depends on the ${CHILD}.

vcsnode1:/root # hagrp -unlink ${PARENT} ${CHILD}

Remove resources

Now you can delete the resources one by one. Following command lists the resources of a resource group:

vcsnode1:/root # hagrp -resources ${PKGNAME}

As you had already saved the resources in the data saving step above, you can take the list there to remove all resources involved in one step. **Please check** this file for correct content before you use it in the next command.

vcsnode1:/root # for resource in $(cat /var/tmp/pkgremoval-resources-${PKGNAME}); do \

hares -delete ${resource} ; \

done

Please check the result with:

vcsnode1:/root # hastatus -sum | sed -e "s/ \*$//g"

vcsnode1:/root # hagrp -resources ${PKGNAME}

The hastatus command above will show you any resource groups currently going offline. So you would realize that you made an error and have to correct it before you went further. The check avoids causing more harm by ignoring problems.

Remove resource groups

As soon as the resources are deleted, you are able to remove the resource group.

vcsnode1:/root # hagrp -delete ${PKGNAME}

Store configuration into cluster

At this point we can store the configuration into the cluster and make the configuration read-only again.

vcsnode1:/root # haconf -dump -makero

Remove start, stop and monitor scripts

For the commands above, it was sufficient to work on one side of the cluster. The VCS software took the burden to synchronize your configuration changes to the other cluster node. From now on, you have to do some steps on both cluster nodes.  
  
Go to each cluster node and remove the scripts. Don't be confused by the second brace in the rm command. The first brace is a variable, the second brace is a globbing expression, that will remove ${PKGNAME}\_monitor, ${PKGNAME}\_start and ${PKGNAME}\_stop with one command.

vcsnode1:/root # cd /lfs/cluster/vcs

vcsnode1:/root # rm ${PKGNAME}\_{monitor,start,stop}

vcsnode2:/root # cd /lfs/cluster/vcs

vcsnode2:/root # rm ${PKGNAME}\_{monitor,start,stop}

Import and delete disk groups

Ensure, that LUNs are only assigned to the both cluster nodes. If your LUNs are assigned to other hosts, your task may be the badly named second half of a package migration. You can use check\_rid.sh from the install servers to find out, where a LUN is visible. You first use it with a server name to get all LUNs currently assigned to that host. Then you can look up one of your LUNs in the output and use the order number of that LUN to do a second search, this time by order number. All listed disks should only be assigned to your cluster nodes vcsnode1 and vcsnode2. Don't continue here, if there are more than this two hosts.

lpinstbss1:/root # check\_rid.sh vcsnode1

lpinstbss1:/root # check\_rid.sh <order-number>

As noted above, you may not start a package, that is already offline. On the other hand you need to import and start a disk group to destroy it, and this will normally done by one of the resources we removed above.  
  
As we have stored all information before, we could have a look into the file /var/tmp/pkgremoval-dgnames-${PKGNAME} and use it to import the disk groups manually. The following step will be done on one side of the cluster, because we directly change information on the LUNs which are shared between the two cluster nodes.  
  
Before you continue, **please check** if you have a propper /var/tmp/pkgremoval-dgnames-${PKGNAME} and a propper /var/tmp/pkgremoval-shortinfo-${PKGNAME}. If you don't have this files, you will not be able to continue after the disk groups have been deleted, because you need that information later.

vcsnode1:/root # for diskgroup in $(grep -e "^dg." /var/tmp/pkgremoval-dgnames-${PKGNAME}); do \

vxdg import ${diskgroup} ; \

vxdg destroy ${diskgroup} ; \

done

The grep above seems to be a little bit superflous, but it filters out the headings that may be still lingering around in the file.

Delete mountpoints

Now you have to delete the mountpoint on both sides of the cluster. First you need to remove the mountpoints in the deeper levels, then you can remove the higher levels. Please check the content of /var/tmp/pkgremoval-mountpoints-${PKGNAME} before you use it in the loop below. If you are getting error messages, you probably tried to remove the wrong directories, there is some old stuff lingering around or this is a special crafted package. Don't ignore this message, try to find out what happened here.

vcsnode1:/root # cat /var/tmp/pkgremoval-mountpoints-${PKGNAME}

vcsnode1:/root # for dir in $(awk '/^[^#]/ {print $4 "/"}' /var/tmp/pkgremoval-mountpoints-${PKGNAME} | sort -r); do \

rmdir ${dir}; \

done

vcsnode1:/root # rmdir /global/${PKGNAME}/

vcsnode1:/root # scp -p /var/tmp/pkgremoval-mountpoints-${PKGNAME} your-account@vcsnode2:/var/tmp

vcsnode2:/root # chown root.root /var/tmp/pkgremoval-mountpoints-${PKGNAME}

vcsnode2:/root # for dir in $(awk '/^[^#]/ {print $4 "/"}' /var/tmp/pkgremoval-mountpoints-${PKGNAME} | sort -r); do \

rmdir ${dir}; \

done

vcsnode2:/root # rmdir /global/${PKGNAME}/

Deconfigure disk on OS level

The easiest way to deconfigure the disks from the system is to use our san\_return script which is available at the install server. You copy the file /var/tmp/pkgremoval-LDEV-BOX-pairs-${PKGNAME} with an uniqe name onto the install server. The install server lpinstiaas01 is assumed in this example. Please be aware, that the deconfiguration has to be done **on both sides** of the cluster.

lpinstiaas01:~> instserv

[INSTSERV][USER][lpinstiaas01] ~ $ PKGNAME=<package name>

[INSTSERV][USER][lpinstiaas01] ~ $ rootscp -p vcsnode1:/var/tmp/pkgremoval-LDEV-BOX-pairs-${PKGNAME} /var/tmp/pkgremoval-TAS000000004711

[INSTSERV][USER][lpinstiaas01] ~ $ san\_return -f /var/tmp/pkgremoval-TAS000000004711 -s vcsnode1 -t TAS000000004711

[INSTSERV][USER][lpinstiaas01] ~ $ san\_return -f /var/tmp/pkgremoval-TAS000000004711 -s vcsnode2 -t TAS000000004711

To deconfigure the disks manually, please have a look into the page [Return SAN VxVM + LVM](https://atc.bmwgroup.net/confluence/pages/viewpage.action?pageId=521112291).  
  
**For SLES12 systems** use the new san\_return script on lpinstiaas01:~> instserv /global/instserv/bin/san\_return (will do both nodes in a single run).

Return IPs

Please check the IP address to return from the proper install server with a ping command before you continue. **If you get an answer you may not delete the IP address.** Try to find out, why you can reach this address and where it belongs now. Perhaps the functionality of your package has been moved to another package at another server and the shift from old to new was done by moving the IP address. There are many possible causes, why an IP address resurrected.  
The actual work will be done in CMDB. Please have a look into the page [Delete IP address](https://bsswiki.muc/tiki-index.php?page=bss_unix_proc_pic_config_delete_ip) for further details.

Delete Firewall Rules for IP, defined CLI process can be use → [5.13.34 Tufin CLI Commands#5.13.34TufinCLICommands-tufin\_decom\_ip](https://atc.bmwgroup.net/confluence/display/IAAS18/5.13.34+Tufin+CLI+Commands#id-5.13.34TufinCLICommands-5.13.34TufinCLICommands-tufin_decom_ip)

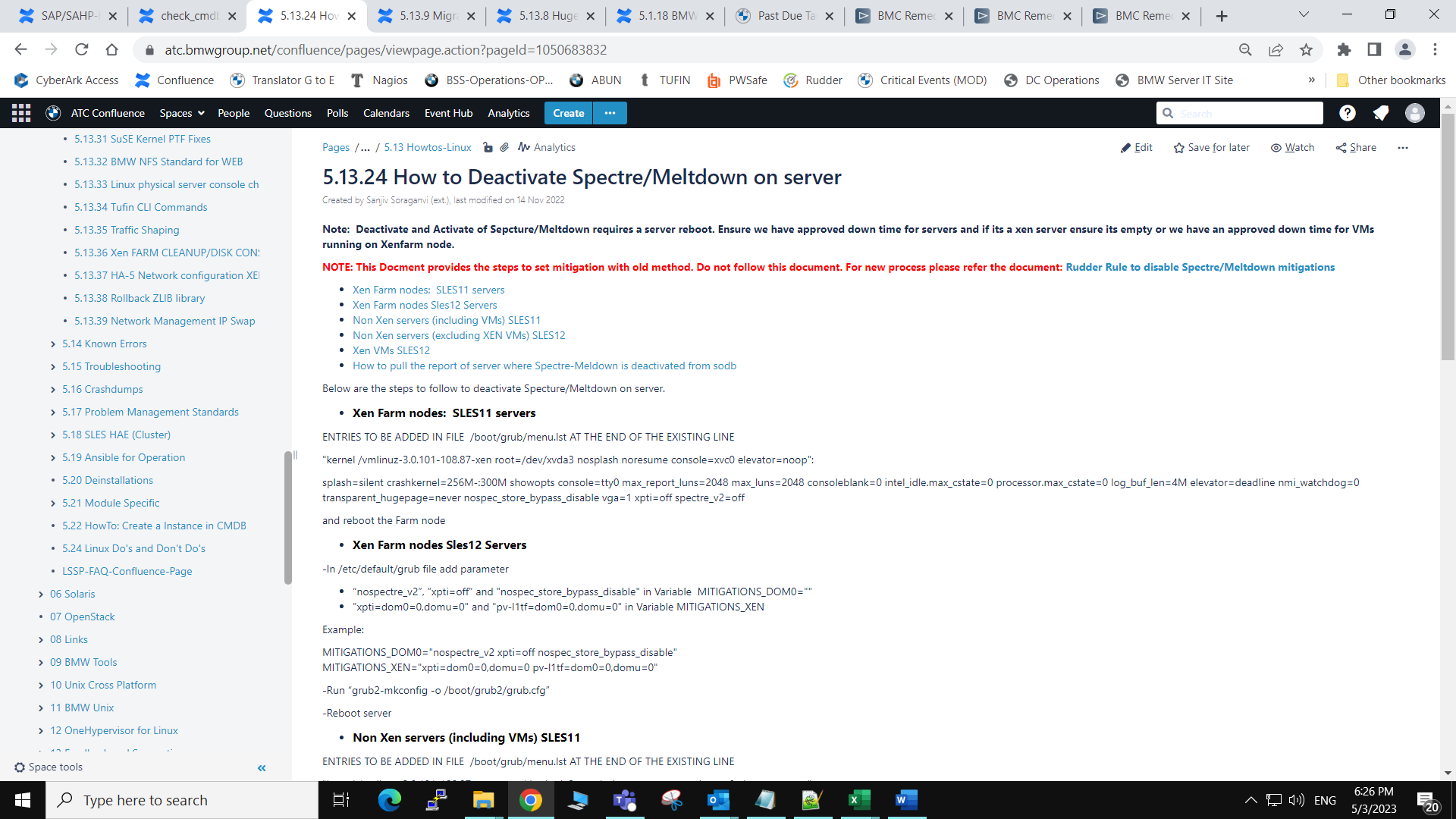
Return Storage

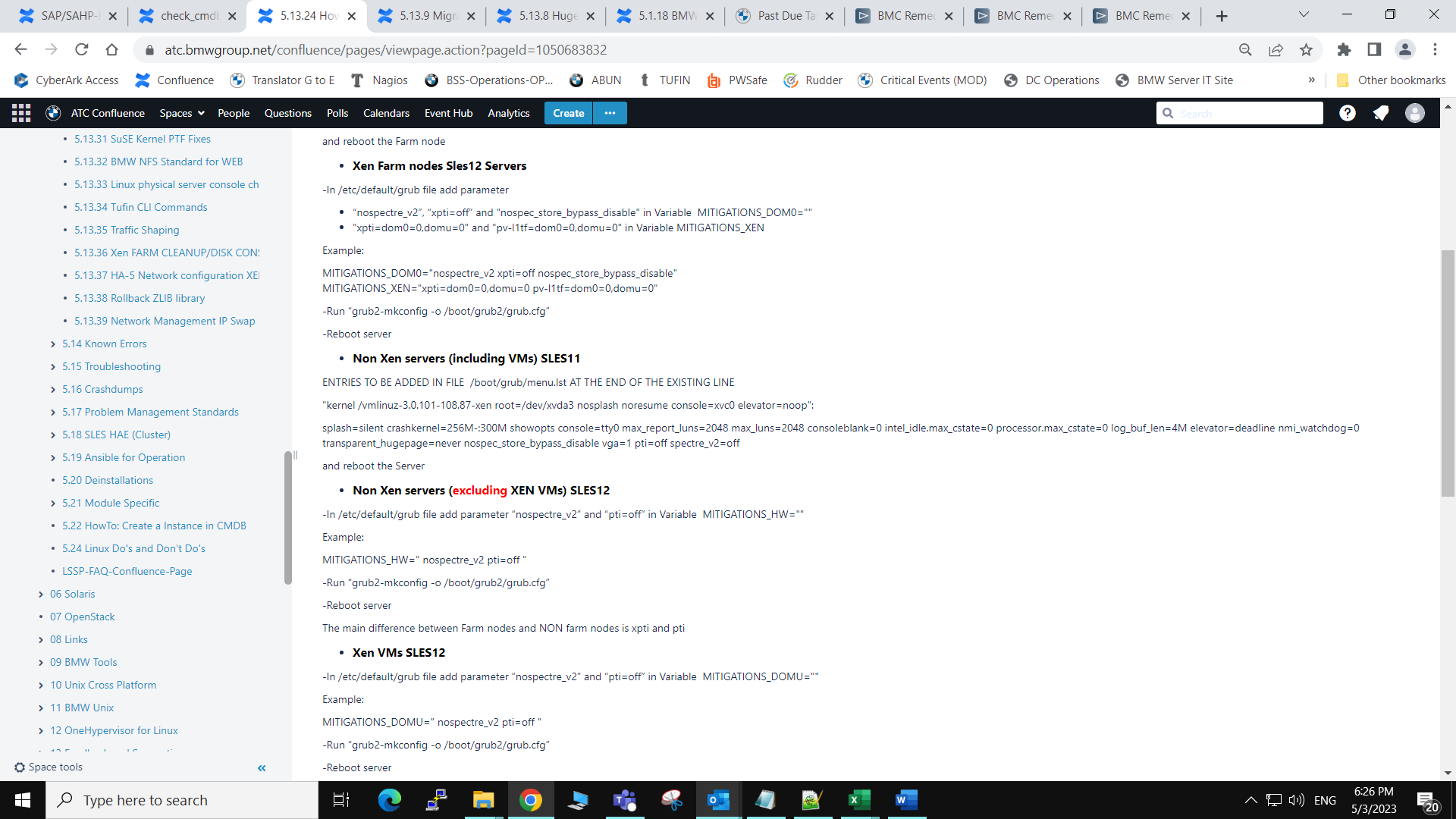
To return the storage you have to create a StorM ticket via [StorM](http://storm.muc/). You will have to create a SAN Return Request. You stored information about LUNs to return in the file /var/tmp/pkgremoval-shortinfo-${PKGNAME} while you collected some data in the steps above. This information will help you to properly select the LUNs to return.  
  
Please copy the complete ticket into your task with a Work Info summary of "SAN Return <san-ticket-no>" and the ticket content as Work Info notes. Please do also check, if at least the task number has been stored in the .donotrescan files. Add this info to the .donotrescan file, if it is still missing.  
  
If you realize at this point, that LUNs are also visible to other hosts than vcsnode1 and vcsnode2, something went very bad (see above). Do not return the storage. Try to find out what happened and why this LUNs have been made available to another server.

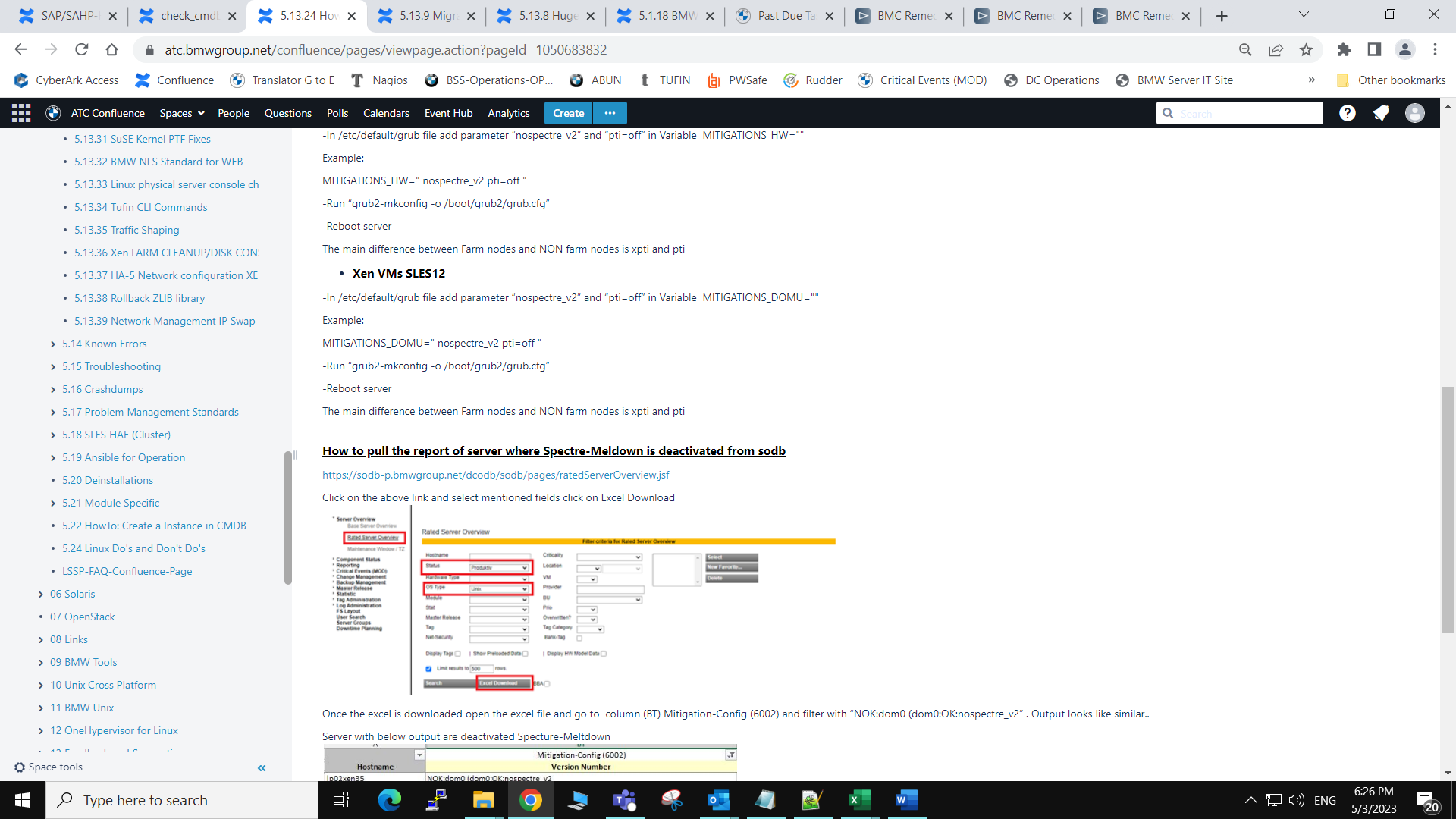
Note: If the cluster nodes will be decomission after package removal then follow the process mentioned in below link for SAN return (after paclage removal process completed)

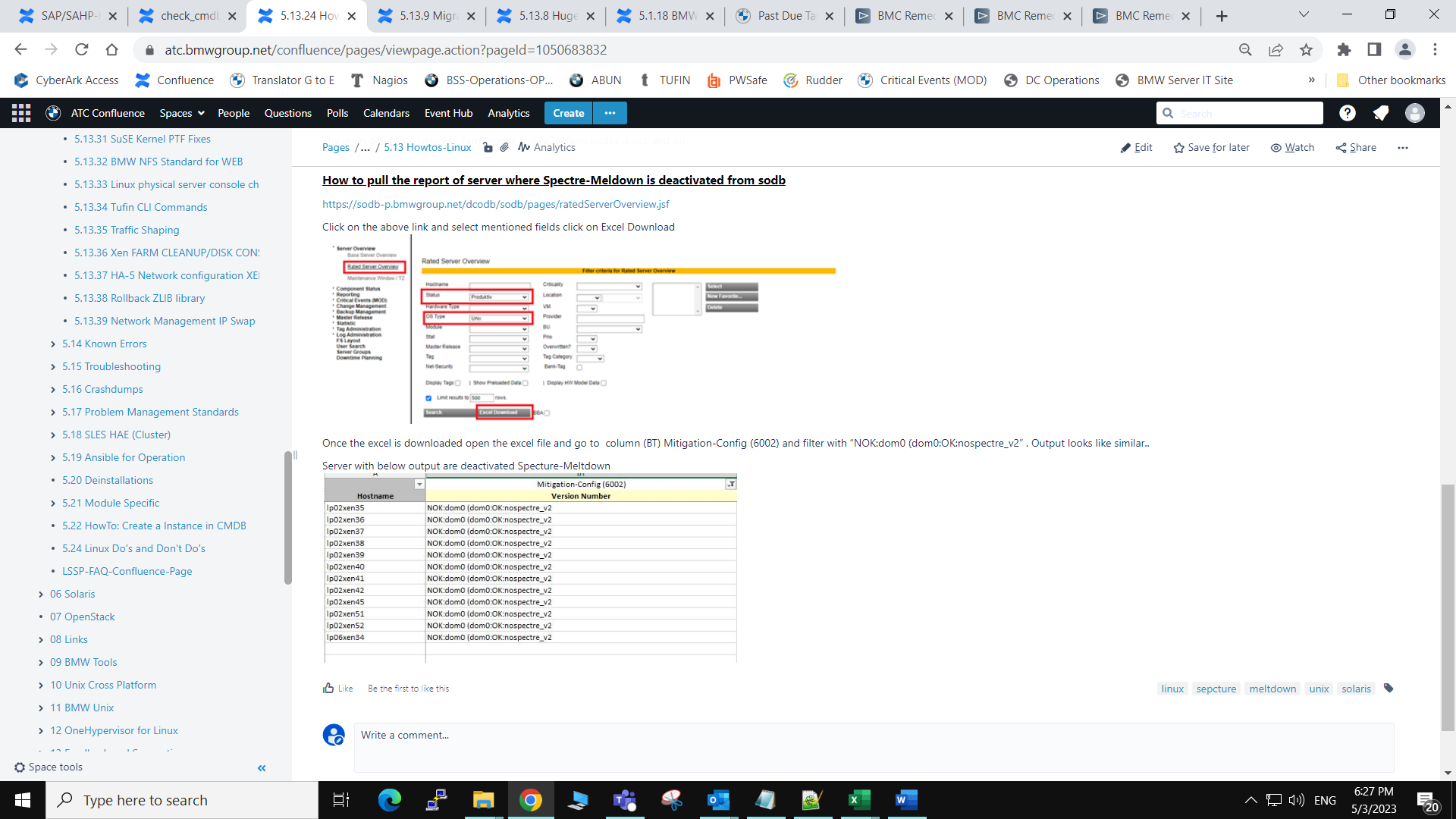
[5.11.1.3 Return Change SAN-Storage](https://atc.bmwgroup.net/confluence/display/IAAS18/5.11.1.3+Return+Change+SAN-Storage)

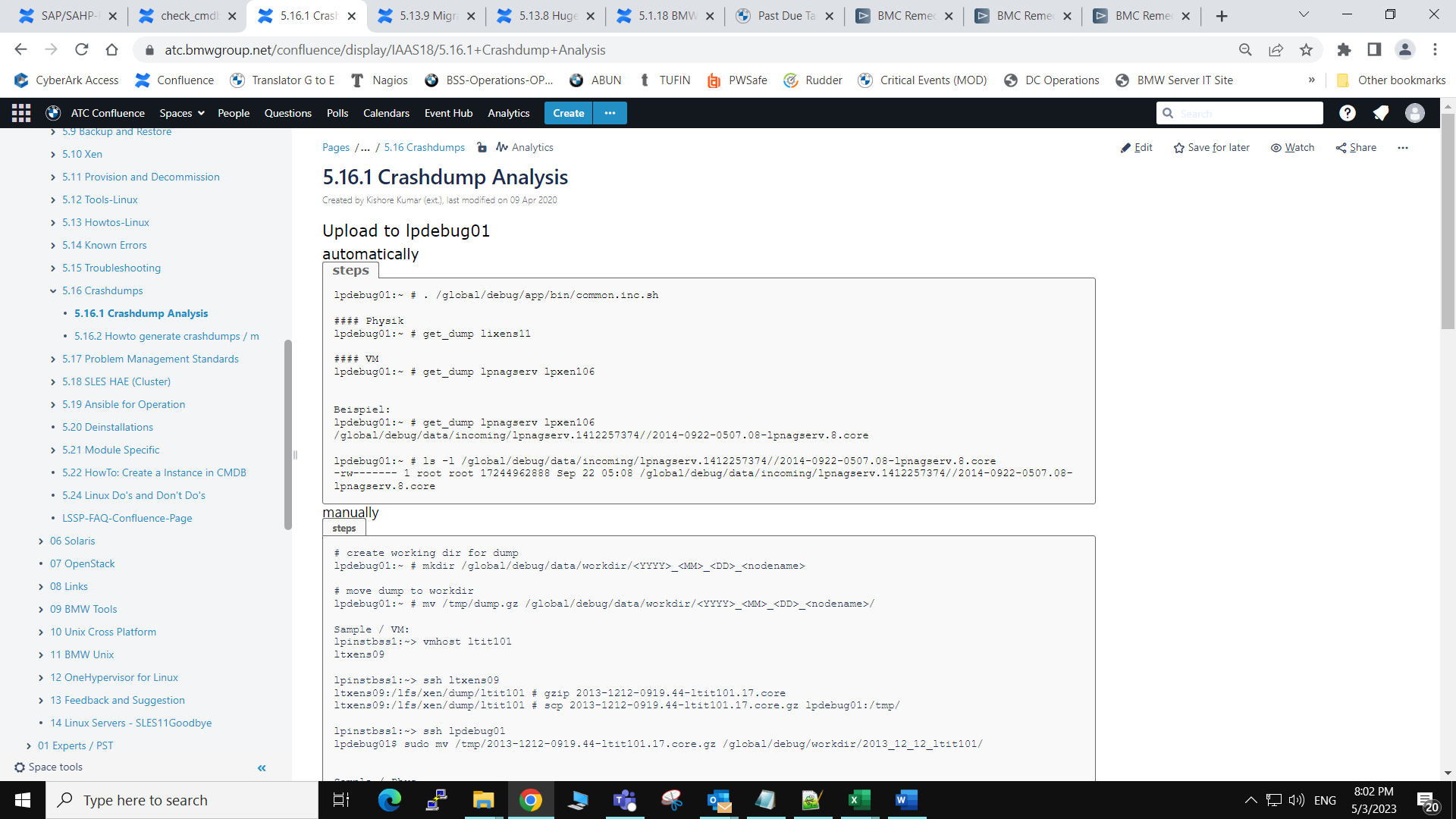
Last edited by Liebl Markus, (Markus.Liebl@partner.bmw.de) , based on work by Schrocko Winfried, (Winfried.Schrocko@partner.bmw.de) , Langer Eberhard, (Eberhard.Langer@partner.bmw.de) , Toth Endre, (Endre.ET.Toth@partner.bmw.de) , Miko Andras, (Andras.Miko@partner.bmw.de) , [qxj5457](https://bsswiki.muc/tiki-user_information.php?userId=-1) , [Kujau Christian, (Christian.CK.Kujau@partner.bmw.de)](https://bsswiki.muc/tiki-user_information.php?userId=55) , Linden Sylvia, (Sylvia.LA.Linden@partner.bmw.de) , Kemmerer Walter, (Walter.Kemmerer@partner.bmw.de) , Knopp Ekkehard, (Ekkehard.Knopp@partner.bmw.de) , Engert Michael, (Michael.Engert@partner.bmw.de) , Noyes Geoffrey, (Geoffrey.Noyes@partner.bmw.de) and [system](https://bsswiki.muc/tiki-user_information.php?userId=-1) .  
Page last modified on Thursday 17 of October, 2019 16:09:10 CEST. (Version 83)

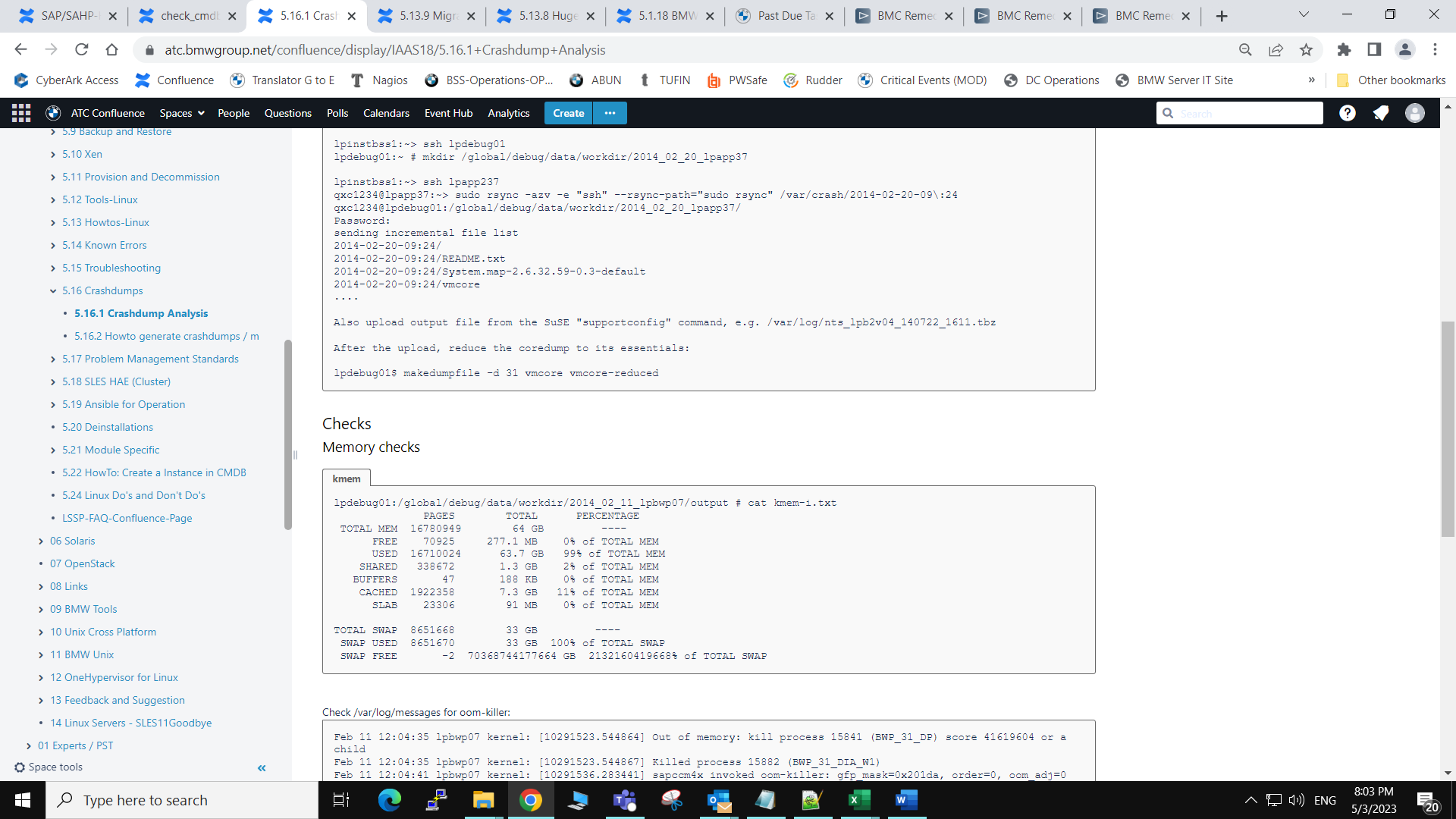


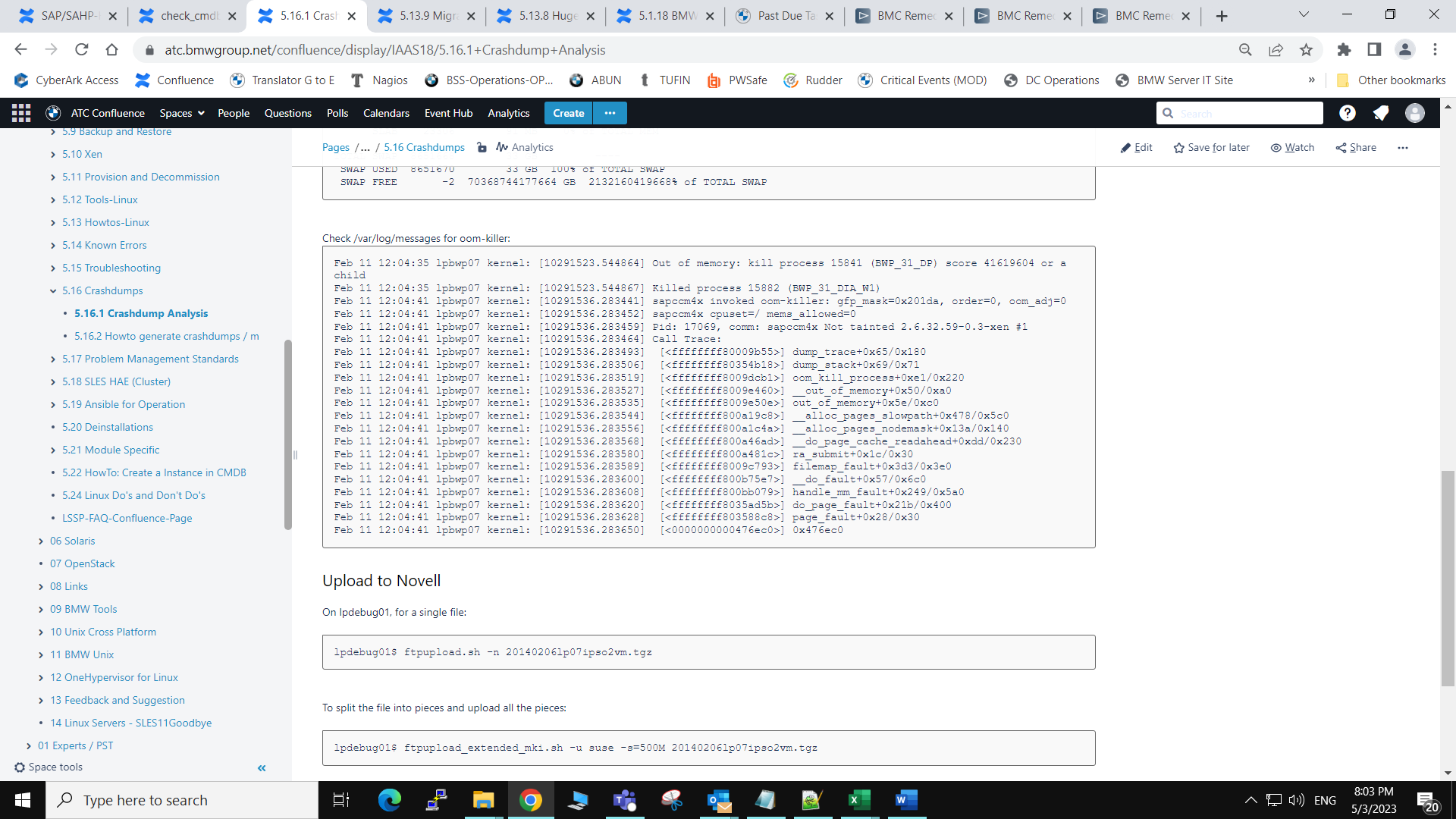


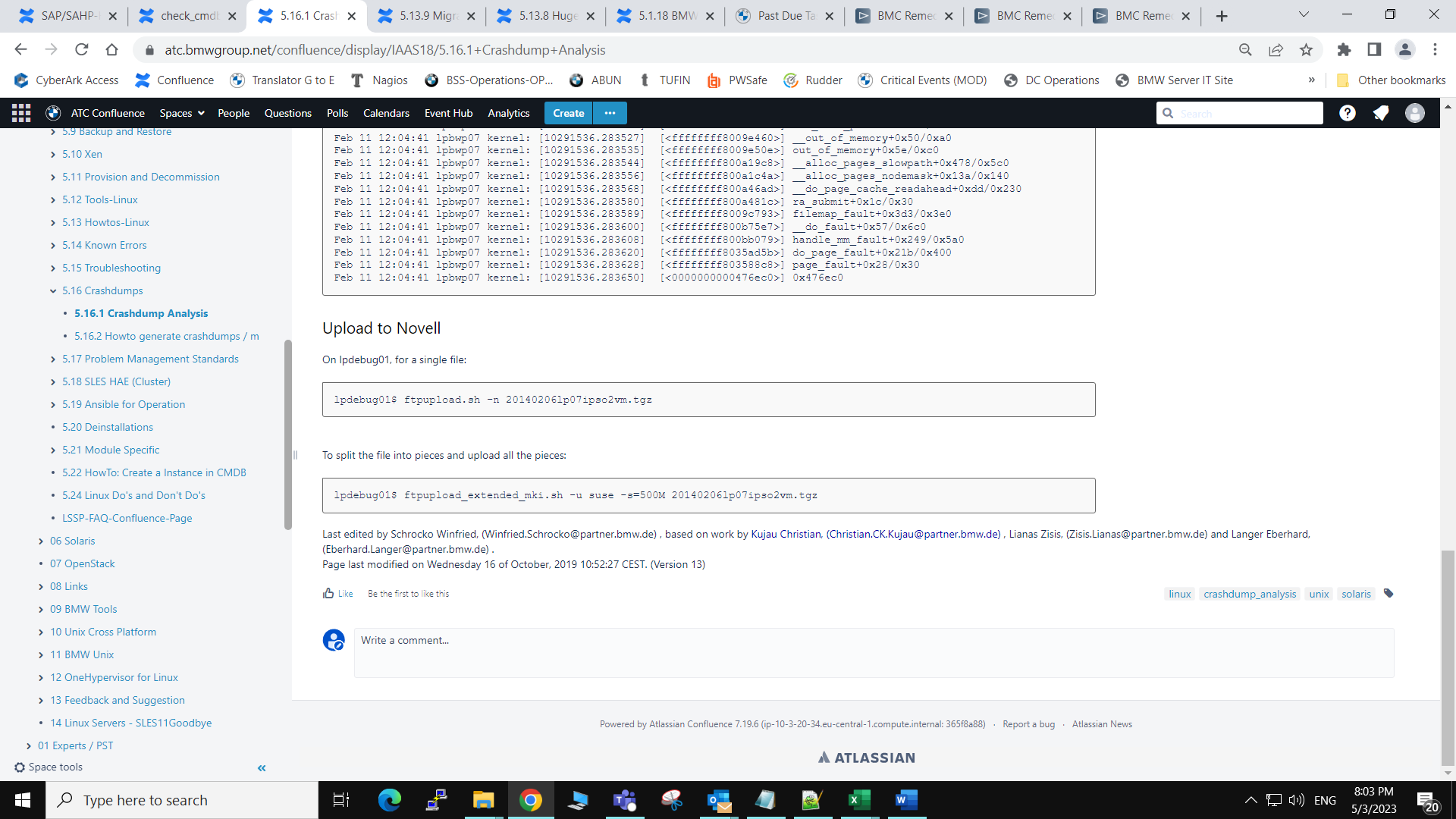


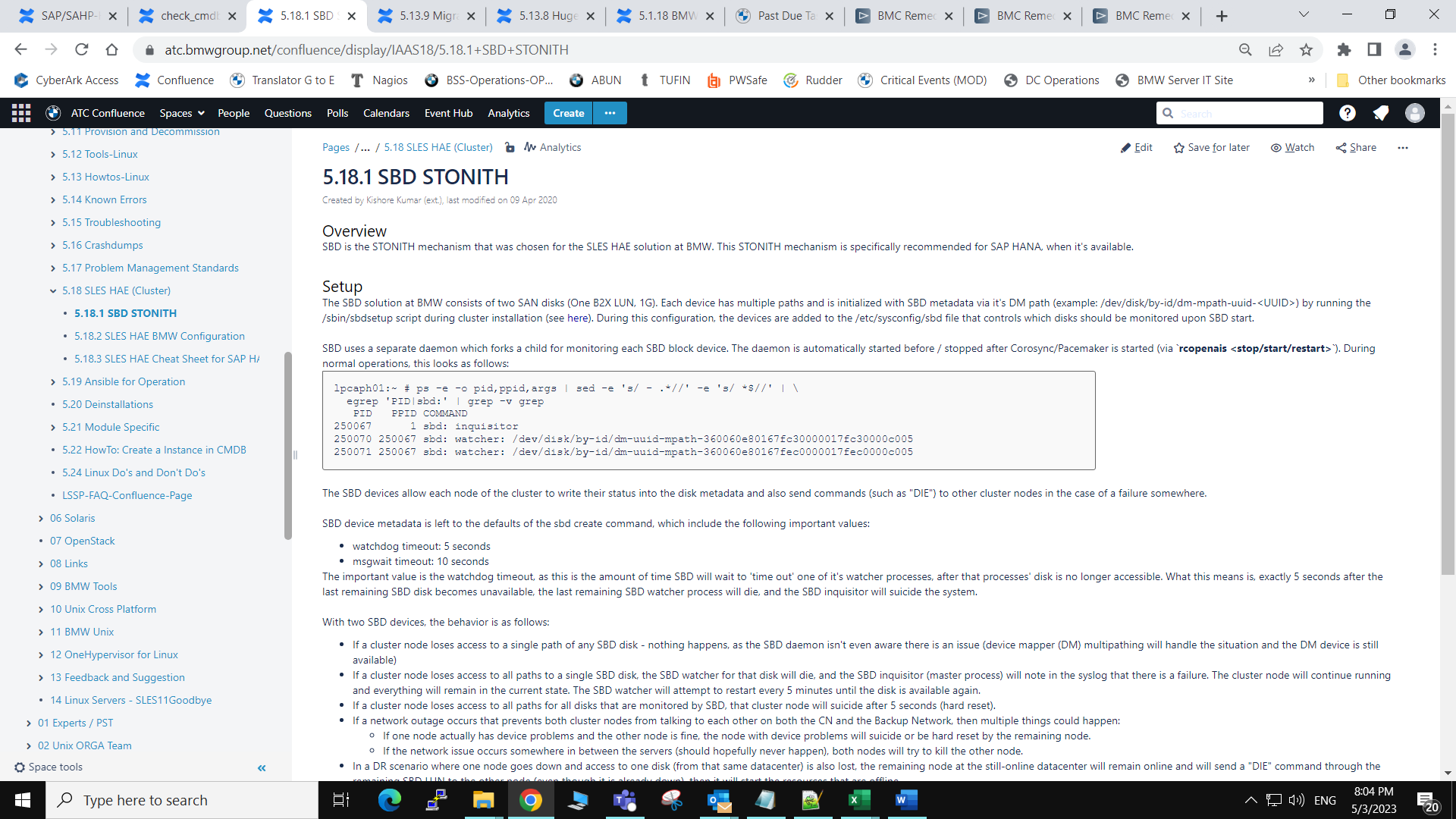














* Created by [Kishore Kumar (ext.)](https://atc.bmwgroup.net/confluence/display/~qxz0crd), last modified on [09 Apr 2020](https://atc.bmwgroup.net/confluence/pages/diffpagesbyversion.action?pageId=521130719&selectedPageVersions=1&selectedPageVersions=2)

Software

As noted in the install guide (see [here](https://bsswiki.muc/tiki-index.php?page=bss_unix_linux_inst_modspec_sap-hana_hae)), the SLES HAE installation is performed by installing the sup pattern BMW-SLESHAE on SLES-11. On SLES-12, the necessary packages are installed by rudder for the relevant modules. SLES HAE comprises the following main components:

* Corosync: a cluster communication engine; see <http://corosync.github.io/corosync>
* Pacemaker: a cluster resource manager; see <http://clusterlabs.org/wiki/Pacemaker.> It runs "on top of" corosync.

For HANA clusters, there are additional module specific cluster agents and tools which can be installed via pattern BMW-SLESHAE\_HANA for SLES-11 (for SLES-12 the packages are installed automatically via Rudder).

Initial Configuration

For SLES-11, basic configuration of the cluster is performed via installing the BMW-CFG-SLESHAE override. For SLES-12, the configuration is not automated yet; the necessary steps are documented in the corresponding install section. In all cases, the configuration involves the following changes:

* Passwordless root login via SSH key is enabled in /etc/ssh/sshd\_config on both nodes, as this is required for the cluster to stay in sync
* csync2 is configured to run via xinetd and will keep the following files in sync between cluster nodes:
  + /etc/booth/booth.conf
  + /etc/corosync/corosync.conf
  + /etc/corosync/authkey
  + /etc/csync2/csync2.cfg
  + /etc/csync2/key\_hagroup
  + /etc/ctdb/nodes
  + /etc/drbd.d
  + /etc/ha.d/[ldirectord.cf](http://ldirectord.cf/)
  + /etc/lvm/lvm.conf
  + /etc/multipath.conf
  + /etc/samba/smb.conf
  + /etc/sysconfig/openais
  + /etc/sysconfig/pacemaker
  + /etc/sysconfig/sbd
* For SLES-11, some files are created on the install server under /global/instserv/data/BMW-CFG-SLESHAE and pushed to a cluster node when the override is installed:
  + /etc/csync2/key\_hagroup (not actually used, but created/shared by default in case it will be used later)
  + root user ssh keypair
* corosync.conf is configured to include the list of cluster nodes as well as each ring device (like a heartbeat in VCS) - one ring is created using the corporate network, and a second ring is created using the backup network. SLES HAE does not support more than 2 rings, so we currently cannot use a third heartbeat-only network as a third ring.
* Cluster firewall settings are defined in /etc/sysconfig/SuSEfirewall2.d/services/cluster
* The hardware watchdog module iTCO\_wdt is blacklisted (via /etc/modprobe.d/BMW-SLESHAE-blacklist) as this is detected on the Lenovo servers but isn't actually an available hardware watchdog device. Instead, the software watchdog driver 'softdog' is set to load on boot (via a boot.BMWwatchdog init script in /etc/init.d/). Once activated, the watchdog module will tell the system to hard reset if the /dev/watchdog file is not either touched every 30 seconds or cleanly reset. Our STONITH Block Device (SBD) cluster fencing setup (see [here](https://bsswiki.muc/tiki-index.php?page=bss_unix_linux_cluster_hae_sbd) and below) will be using /dev/watchdog as soon as the STONITH resources are started. Interrupting SBD uncleanly will result in a failure to touch the watchdog file within the appropriate timeframe and the system will reset.

SBD configuration

During initial configuration, no cluster resources are created, and the default cluster configuration is only a basic set of default properties with default values. After the cluster is initially configured, the /sbin/sbdsetup script should normally be run on one of the cluster nodes to initialize the two SBD devices. At that time, a new STONITH cluster resource along with a clone resource is automatically configured ("sbd-fencing" and "cln-sbd-fencing" - one resource runs on each cluster node). This resource talks between pacemaker and the SBD daemon. At that time, stonith-enabled is also set to true (to enable the overall stonith mechanism in pacemaker). The stonith-timeout value is left at the default of 75s (this value is mostly irrelevant as the actual SBD timeouts are much smaller - 5s for watchdog timeout, 10s for message timeout), so the stonith timeout in pacemaker should never be reached.  
A clone resource is basically an identical copy of an existing resource that can run simultaneously on another node. In this case we need a clone resource of the sbd-fencing resource so that each node can fence the other node.

General admin notes

When the cluster is set to maintenance mode (manual command would be `**crm configure property maintenance-mode=true**`), all cluster resources are set to unmanaged and monitoring is immediately stopped on the resources. Any application level change can then be made without affecting the cluster. Before the cluster maintenance mode is disabled (via `**crm configure property maintenance-mode=false**`), everything on each cluster node (application/resource states, etc) should be in the state it was in prior to enabling maintenance mode, to prevent faults/automatic failover/resource problems). If cluster nodes are rebooted while the cluster is in maintenance mode, the local resource manager will forget the status of any resource that might have been online on this node. Hence, applications must not be restarted manually on rebooted nodes; instead the resources must be restarted via the cluster after disabling maintenance mode.

Logging

Cluster status messages and SBD messages are logged to /var/log/messages, but a concise list of messages can be automatically parsed from both cluster nodes by using the following crm commands:

* **crm history latest**: prints all resource transition messages from the last resource transistion (start, stop, etc)
* **crm history log node**: prints all cluster logs from both cluster nodes (or the node specified) since the last /var/log/messages log rotation.

Further documentattion

The official SLES HAE Administration documentation can be found at the following link: <https://www.suse.com/documentation/sle_ha/book_sleha/data/book_sleha.html>  
On top of this, Pacemaker comes with a decent set of man pages that are installed on all systems where the SLES HAE extension is deployed.

Last edited by Schrocko Winfried, (Winfried.Schrocko@partner.bmw.de) , based on work by Porezag Dirk, (Dirk.Porezag@partner.bmw.de) and Vonderbecke Michael, FG-AM-81 . Page last modified on Thursday 17 of October, 2019 15:52:28 CEST. (Version 17)

* Created by [Kishore Kumar (ext.)](https://atc.bmwgroup.net/confluence/display/~qxz0crd), last modified on [09 Apr 2020](https://atc.bmwgroup.net/confluence/pages/diffpagesbyversion.action?pageId=520971891&selectedPageVersions=1&selectedPageVersions=2)

Table of contents:

# Overview

In the SLES HAE HANA setup, the Pacemaker cluster monitors and manages two SAP HANA instances:

* a master instance that is actively serving client requests
* a slave instance that is kept in sync with the master via database replication and may take over the master role in case the master should fail.

SAP AG and SuSE have jointly developed two agents for managing the SAP HANA processes and replication mode in this setup:

* the SAPHanaTopology agent determines what the current master/slave topology of the HANA system is
* the SAPHana agent actively manages the HANA instances on the two nodes

Both agents are contained in the SAPHanaSR RPM package and can be found in the directory /usr/lib/ocf/resource.d/suse.  
  
The package also contains the script **/usr/share/SAPHanaSR/tests/show\_SAPHanaSR\_attributes** used for printing status details of the HANA resources. Please note: the tool does **NOT** print the actual HANA database status but instead what the cluster **THINKS** the status is. This may not be the actual status, especially if the cluster was/is frozen. Also, more recent versions of the tool require manual inclusion of the /usr/lib/SAPHanaSR/SAPHanaSRTools.pm perl library and extra arguments for execution. Therefore, it is not useful for quick troubleshooting any more.  
  
Instead, it's better to check the node attributes updated by the HANA agents and visible in the "crm\_mon -Ar1" output. Among other things, these attributes will also show what the sync\_state of the replication is (according to the cluster resource monitoring). During normal operations, the corresponding value of hana\_<sid>\_sync\_state should be "PRIM" for the primary and "SOK" for the secondary node.  
  
Each SLES HAE HANA setup will have the following resources:

* a clone set **cln-sbd-fencing [sbd-fencing]** implementing the SBD STONITH setup (see [here](https://bsswiki.muc/tiki-index.php?page=bss_unix_linux_cluster_hae_sbd))
* a clone set **cln\_SAPHanaTopology\_<SID><IN> [rsc\_SAPHanaTopology\_<SID><IN]** where <SID> is the SID of the HANA instance and <IN> the instance number (for production instances usually 00). This clone set is managed by the SAPHanaTopology agent (see above).
* a master/slave set **msl\_SAPHana\_<SID><IN> [rsc\_SAPHana\_<SID><IN>]** managed by the SAPHana agent. The state of this master/slave resource should reflect the current roles that the two HANA instances should have.
* an IP resource **rsc\_ip\_<SID><IN>** which is managed by the cluster and should always be online on the node that acts as the current master in the master/slave set described above. The correct placement of the IP is enforced by a cluster collocation rule.
* for cost optimized HANA HAE clusters, there is also a Q/A HANA instance with the name rsc\_SAPDatabase\_<SID><IN> (with the Q/A <SID><IN>!) that can only run on the secondary node and only if this node is NOT the master for production at this time.

The full picture during normal operations should look like this (example without Q/A instance):

root@lpcaph01:~# crm\_mon -1r

Last updated: Tue Jun 21 20:00:29 2016

Last change: Tue Jun 21 19:59:32 2016 by root via crm\_attribute on lpcaph01

Stack: classic openais (with plugin)

Current DC: lpcaph01 - partition with quorum

Version: 1.1.11-3ca8c3b

2 Nodes configured, 2 expected votes

7 Resources configured

Online: [ lpcaph01 lpcaph51 ]

Clone Set: cln-sbd-fencing [sbd-fencing]

Started: [ lpcaph01 lpcaph51 ]

Clone Set: cln\_SAPHanaTopology\_CAP00 [rsc\_SAPHanaTopology\_CAP00]

Started: [ lpcaph01 lpcaph51 ]

Master/Slave Set: msl\_SAPHana\_CAP00 [rsc\_SAPHana\_CAP00]

Masters: [ lpcaph01 ]

Slaves: [ lpcaph51 ]

rsc\_ip\_CAP00 (ocf::heartbeat:IPaddr2): Started lpcaph01

# Cluster Health Check

Run "crm\_mon -1r" and verify that the cluster status looks normal. In particular:

* the SAPHanaTopology clone resource should be started on both nodes
* the Master/Slave set should have one Master (usually the node ending in 01) and one Slave (ususally the node ending in 51)
* the rsc\_ip IP resource should be active on the Master node

If all resources appear as "unmanaged", the cluster functions have been disabled by setting maintenance mode, usually due to application level activity on the affected system. For more details regarding maintenance mode, see the next sections in this Wiki page.  
  
In all other cases where the cluster status looks different, the situation must be resolved in cooperation with SAP basis operations. It is possible that they have removed maintenance mode on the cluster before the system was completely recovered. In this case - and only if SAP basis confirms that the system should be clean at the application level - it usually helps to clear previous errors by running "crm resource cleanup <resource> <host>" as demonstrated in the example below. Besides clearing previous failures of the affected resource, this forces the cluster to re-run its monitor and hence will update its status:

lpcmph51:~ # crm\_mon -1

Last updated: Mon May 14 02:00:08 2018

Last change: Mon May 14 01:59:35 2018 by root via crm\_attribute on lpcmph01

Stack: classic openais (with plugin)

Current DC: lpcmph51 - partition with quorum

Version: 1.1.12-f47ea56

2 Nodes configured, 2 expected votes

7 Resources configured

Online: [ lpcmph01 lpcmph51 ]

Clone Set: cln-sbd-fencing [sbd-fencing]

Started: [ lpcmph01 lpcmph51 ]

Clone Set: cln\_SAPHanaTopology\_CMP00 [rsc\_SAPHanaTopology\_CMP00]

Started: [ lpcmph01 lpcmph51 ]

Master/Slave Set: msl\_SAPHana\_CMP00 [rsc\_SAPHana\_CMP00]

rsc\_SAPHana\_CMP00 (ocf::suse:SAPHana): Master lpcmph01 (Monitoring)

rsc\_ip\_CMP00 (ocf::heartbeat:IPaddr2): Started lpcmph01

Failed actions:

rsc\_SAPHana\_CMP00\_start\_0 on lpcmph51 'unknown error' (1): call=175, status=complete, exit-reason='none', last-rc-change='Fri May 11 12:52:36 2018', queued=0ms, exec=600663ms

lpcmph51:~ # crm resource cleanup rsc\_SAPHana\_CMP00 lpcmph51

Cleaning up rsc\_SAPHana\_CMP00:0 on lpcmph51

Waiting for 1 replies from the CRMd. OK

lpcmph51:~ # crm\_mon -1

Last updated: Mon May 14 02:01:16 2018

Last change: Mon May 14 02:00:40 2018 by root via crm\_attribute on lpcmph01

Stack: classic openais (with plugin)

Current DC: lpcmph51 - partition with quorum

Version: 1.1.12-f47ea56

2 Nodes configured, 2 expected votes

7 Resources configured

Online: [ lpcmph01 lpcmph51 ]

Clone Set: cln-sbd-fencing [sbd-fencing]

Started: [ lpcmph01 lpcmph51 ]

Clone Set: cln\_SAPHanaTopology\_CMP00 [rsc\_SAPHanaTopology\_CMP00]

Started: [ lpcmph01 lpcmph51 ]

Master/Slave Set: msl\_SAPHana\_CMP00 [rsc\_SAPHana\_CMP00]

Masters: [ lpcmph01 ]

Slaves: [ lpcmph51 ]

rsc\_ip\_CMP00 (ocf::heartbeat:IPaddr2): Started lpcmph01

# Standard Operations Tasks

## Setting and removing maintenance mode

Setting the maintenance mode will "freeze" all cluster actions, including resource monitoring. This is necessary for certain maintenance tasks (e.g. SAP upgrades or restarting SAP processes outside of the cluster).

### Setting cluster maintenance mode

Maintenance mode is set by running

crm configure property maintenance-mode=true

or alternatively by using the script /usr/local/bin/freeze\_cluster.

### Removing cluster maintenance mode:

Maintenance mode is removed by running:

crm configure property maintenance-mode=false

or alternatively by using the script /usr/local/bin/unfreeze\_cluster.

### Additional Information

If maintenance mode is set, all instances become "unmanaged":

root@lpcaph01:~# crm\_mon -1r

Last updated: Tue Jun 21 20:11:48 2016

Last change: Tue Jun 21 20:11:43 2016 by root via crm\_attribute on lpcaph01

Stack: classic openais (with plugin)

Current DC: lpcaph01 - partition with quorum

Version: 1.1.11-3ca8c3b

2 Nodes configured, 2 expected votes

7 Resources configured

Online: [ lpcaph01 lpcaph51 ]

Clone Set: cln-sbd-fencing [sbd-fencing] (unmanaged)

sbd-fencing (stonith:external/sbd): Started lpcaph51 (unmanaged)

sbd-fencing (stonith:external/sbd): Started lpcaph01 (unmanaged)

Clone Set: cln\_SAPHanaTopology\_CAP00 [rsc\_SAPHanaTopology\_CAP00] (unmanaged)

rsc\_SAPHanaTopology\_CAP00 (ocf::suse:SAPHanaTopology): Started lpcaph51 (unmanaged)

rsc\_SAPHanaTopology\_CAP00 (ocf::suse:SAPHanaTopology): Started lpcaph01 (unmanaged)

Master/Slave Set: msl\_SAPHana\_CAP00 [rsc\_SAPHana\_CAP00] (unmanaged)

rsc\_SAPHana\_CAP00 (ocf::suse:SAPHana): Started lpcaph51 (unmanaged)

rsc\_SAPHana\_CAP00 (ocf::suse:SAPHana): Master lpcaph01 (unmanaged)

rsc\_ip\_CAP00 (ocf::heartbeat:IPaddr2): Started lpcaph01 (unmanaged)

**Note:** Whenever maintenance mode is set, the cluster will stop monitoring the resources (unlike VCS that still does monitor resources even if a group is frozen!) and it will also not take any actions. A status update for a resource in maintenance mode can be forced by running:

crm resource cleanup <resource> [host]

This will force the cluster to first "forget" the last status information stored for this resource, then execute one monitoring cycle and thus synchronize resource status with reality - but only if the agent is able to derive the correct status. In some situations, this may not be the case. For example, the SAPHana agent will sometimes fail to derive the correct status if the HANA processes are in an inconsistent state or if HANA is unable to start due to a configuration issue.  
  
Below is an example for cleaning up resource status in active (non-maintenance) mode after starting the resource failed. SAP operations then corrected application status (master on lpcaph01, slave on lpcaph51) but due to the previously failed actions the cluster didn't have the current status. Via **crm resource cleanup**, the cluster is forced to delete the error history and status of rsc\_SAPHana\_CAP00 on both nodes, then monitors the current status of the resource and consequently moves the IP resource to the current master node, resulting in a clean final state. If the cluster had been in maintenance mode, the status update would have occurred as well but the IP resource would not have been moved to lpcaph01.

lpcaph01:~ # crm\_mon -1r

Last updated: Wed Jun 22 10:07:12 2016

Last change: Wed Jun 22 09:11:31 2016 by root via crm\_attribute on lpcaph51

Stack: classic openais (with plugin)

Current DC: lpcaph01 - partition with quorum

Version: 1.1.11-3ca8c3b

2 Nodes configured, 2 expected votes

7 Resources configured

Online: [ lpcaph01 lpcaph51 ]

Clone Set: cln-sbd-fencing [sbd-fencing]

Started: [ lpcaph01 lpcaph51 ]

Clone Set: cln\_SAPHanaTopology\_CAP00 [rsc\_SAPHanaTopology\_CAP00]

Started: [ lpcaph01 lpcaph51 ]

rsc\_ip\_CAP00 (ocf::heartbeat:IPaddr2): Started lpcaph51

Failed actions:

rsc\_SAPHana\_CAP00\_start\_0 on lpcaph51 'unknown error' (1): call=227, status=complete, last-rc-change='W016', queued=0ms, exec=26279ms

rsc\_SAPHana\_CAP00\_start\_0 on lpcaph01 'not running' (7): call=118, status=complete, last-rc-change='Wed6', queued=0ms, exec=18531ms

lpcaph01:~ # crm resource cleanup rsc\_SAPHana\_CAP00

lpcaph01:~ # sleep 15 # wait a while for monitoring to complete

lpcaph01:~ # crm\_mon -1r

Last updated: Wed Jun 22 10:26:57 2016

Last change: Wed Jun 22 10:26:19 2016 by root via crm\_attribute on lpcaph01

Stack: classic openais (with plugin)

Current DC: lpcaph01 - partition with quorum

Version: 1.1.11-3ca8c3b

2 Nodes configured, 2 expected votes

7 Resources configured

Online: [ lpcaph01 lpcaph51 ]

Clone Set: cln-sbd-fencing [sbd-fencing]

Started: [ lpcaph01 lpcaph51 ]

Clone Set: cln\_SAPHanaTopology\_CAP00 [rsc\_SAPHanaTopology\_CAP00]

Started: [ lpcaph01 lpcaph51 ]

Master/Slave Set: msl\_SAPHana\_CAP00 [rsc\_SAPHana\_CAP00]

Masters: [ lpcaph01 ]

Slaves: [ lpcaph51 ]

rsc\_ip\_CAP00 (ocf::heartbeat:IPaddr2): Started lpcaph01

Below is an example of what to doe when a monitor has failed actions. I tis case there is no resource named **rsc\_SAPHana\_NWP00\_monitor\_61000**, so what we have to do is cleanup rsc\_SAPHana\_NWP00\_monitor\_61000\_\_.

Failed Actions:

\* rsc\_SAPHana\_NWP00\_monitor\_61000 on lpnwph01 'unknown error' (1): call=135, status=complete, exitreason='none',

last-rc-change='Mon Oct 14 08:23:42 2019', queued=0ms, exec=0ms

\* rsc\_SAPHana\_NWP00\_monitor\_61000 on lpnwph51 'unknown error' (1): call=116, status=complete, exitreason='none',

last-rc-change='Fri Sep 27 12:16:53 2019', queued=0ms, exec=0ms

lpnwph01:~ # crm resource cleanup rsc\_SAPHana\_NWP00

Cleaning up rsc\_SAPHana\_NWP00:0 on lpnwph01, removing fail-count-rsc\_SAPHana\_NWP00

Cleaning up rsc\_SAPHana\_NWP00:0 on lpnwph51, removing fail-count-rsc\_SAPHana\_NWP00

Waiting for 2 replies from the CRMd.. OK

In general, extreme care should be taken when maintenance mode is removed! The cluster will then run a monitor cycle for each resource and enforce a status that is consistent with its current configuration. For example, if "standby" mode has been set for a node, the cluster will stop all resources currently active on this node. The safest approach is to make sure that the status of all resources before removing maintenance mode is the same as at the point in time when maintenance mode was set.  
  
**Note:** SAP operations is allowed to set and remove maintenance mode in order to manage HANA without requiring intervention by BMW Unix operations.

## Switching HANA master role between cluster nodes

**Prerequisites**  
Before performing a manual switch test, make sure the cluster status returned by the "crm\_mon -Ar1" command fulfills the following conditions:

* node1 should be listed with node attribute hana\_<sid>\_sync\_state=PRIM
* node2 should be listed with node attribute hana\_<sid>\_sync\_state=SOK
* the cluster should have no "Failed Actions" in the "crm\_mon" output

If the HANA sync state is different, notify the SAP Basis team that the HANA system replication is not working correctly and therefore the cluster switch test cannot be performed. If there are "Failed Actions", clean up the correspondig resources with "crm resource cleanup" (for details, see the relevant chapter on this Wiki page) until all "Failed Actions" are cleared.  
  
**Performing the switch**  
In order to switch the master role from <node1> to <node2>, perform the following steps:

# check the current cluster status (on either node)

crm\_mon -Ar1

# make sure the prerequisites outlined above are met

# DO NOT CONTINUE IF THE RESOURCE OR REPLICATION STATES ARE DIFFERENT!

# now place the current master node1 into standby mode

# this will force the cluster to promote node2 to be master

crm node standby <node1>

# wait while observing the crm\_mon output until node2 is listed as master

crm\_mon

# repeatedly check until node2 shows hana\_<sid>\_sync\_state=PRIM

crm\_mon -Ar1

# now that node2 is the active master, we can online node1 - it should start as slave

crm node online <node1>

# wait while observing the crm\_mon output until node1 is listed as slave

crm\_mon

# repeatedly check until node1 shows hana\_<sid>\_sync\_state=SOK

crm\_mon -Ar1

**Troubleshooting hints:**  
If the status of the HANA master/slave resource (msl\_SAPHana\_<SID><IN>) is weird (two slaves, no master, etc.), sometimes the SAP team might have to fix the application. First check the output of "crm\_mon -Ar1". If both nodes in that output are in sync\_state "PRIM" or both nodes are "SFAIL" then the SAP team may have to fix the problem on the SAP side. If one node shows "PRIM" then it should also show up as "Master" in the crm\_mon output, the other node might show up as "Slave" in crm\_mon but show as "SFAIL" or "UNDEFINED" or something like that in the show\_SAPHanaSR\_attributes output.  
  
In that case, and if SAP operations claims that they see a different status at the application level, it may help to try the following steps:

1. Ask SAP operations which system is acting as master and which is acting as slave (or use the SAP-HANA commands documented in the section below).
2. Put the cluster node acting as slave in standby mode (**crm node standby <node>**).
3. Restart the Pacemaker cluster services on the slave node (see below).
4. Put the cluster node acting as slave back to online mode (**crm node online <node>**).

The Pacemaker cluster services can be restarted as follows:

rcopenais restart # SLES 11

systemctl restart pacemaker.service # SLES 12

Most cases where inconsistencies between cluster status and application status as reported by SAP operations occur can be fixed by restarting Pacemaker on one or both nodes, but this should be a last resort and only done after SAP operations confirms that this may be done. The restart forces the local resource manager to re-initialize and update the status of all resources from scratch.

## OS maintenance

The proper procedure for doing OS maintenance that might require one or more reboots of individual cluster nodes is as follows:

1. Put the cluster into maintenance mode (see above).
2. Request SAP operations to stop all HANA instances on the cluster and wait until they confirm that application shutdown is complete.
3. Shutdown the cluster processes on both nodes (see below). It is recommended to first shutdown the node that was acting as slave and then the master. If only a single system needs maintenance, it is sufficient to shutdown cluster services on this node, the other node may remain online. NOTE: If cluster services are stopped on SLES 12, make sure to stop ALL components (pacemaker, corosync and sbd) - a partial stop may lead to unexpected reboots if the other node executes a STONITH fencing action when it can't communicate with its partner's pacemaker process.
4. Perform system maintenance. Nodes can be rebooted as necessary.
5. When both nodes are back online and work without issues, the cluster can be taken out of maintenance mode. At this time, HANA should be automatically started by the cluster and come up with the same master/slave setup that was active before application shutdown.

# stop / start / restart pacemaker services

rcopenais [stop / start / restart] # SLES 11

systemctl [stop / start / restart] pacemaker.service corosync.service sbd.service # SLES 12

**Troubleshooting:**  
If the resources do not start automatically, the best option is to restart pacemaker on both cluster nodes (first one node, then the other node).

# Known issues

During the cluster tests it was found that host crashes may lead to empty (or possibly corrupt) HANA configuration files (in one particular instance, for example, the main config file /hana/shared/<SID>/global/hdb/custom/config/global.ini was empty). This will prevent HANA from starting and can be seen in the **crm\_mon** output:

rsc\_SAPHana\_<SID><IN>\_start\_0 (call=<XX>, rc=1, cib-update=<YY>, confirmed=true) unknown error

In order to fix this, first put the cluster into maintenance mode and then tell SAP operations to fix the HANA configuration files and start HANA manually. After HANA is up and running, run

crm resource cleanup rsc\_SAPHana\_<SID><IN>

while still in maintenance mode. The cluster will then remove the old failed status, perform a new monitor cycle and pick up the new status. If this status is as expected (working master/slave setup), the maintenance mode can be removed.

# Some SAP HANA check commands

This is auxiliary information which may be helpful for troubleshooting and supporting SAP operations in some cases. Unix operations is not responsible for the application, however, since there is a tight integration between application and cluster, it may be helpful to check the application status directly.  
  
**NOTE: All commands MUST be executed as <sid>adm, not as root!**

## ****Check HANA process status****

su - <sid>adm

/usr/sap/hostctrl/exe/sapcontrol -nr <IN> -function GetProcessList

Sample output:

lpcaph51:/usr/sap/CAP/HDB00> /usr/sap/hostctrl/exe/sapcontrol -nr 00 -function GetProcessList

21.06.2016 22:36:19

GetProcessList

OK

name, description, dispstatus, textstatus, starttime, elapsedtime, pid

hdbdaemon, HDB Daemon, GREEN, Running, 2016 06 21 19:34:44, 3:01:35, 31212

hdbcompileserver, HDB Compileserver, GREEN, Running, 2016 06 21 19:34:50, 3:01:29, 31272

hdbindexserver, HDB Indexserver, GREEN, Running, 2016 06 21 19:34:54, 3:01:25, 31405

hdbnameserver, HDB Nameserver, GREEN, Running, 2016 06 21 19:34:45, 3:01:34, 31228

hdbpreprocessor, HDB Preprocessor, GREEN, Running, 2016 06 21 19:34:50, 3:01:29, 31274

hdbwebdispatcher, HDB Web Dispatcher, GREEN, Running, 2016 06 21 19:35:40, 3:00:39, 32941

hdbxsengine, HDB XSEngine, GREEN, Running, 2016 06 21 19:34:54, 3:01:25, 31407

## ****Check master/slave replication role of local system****

su - <sid>adm

hdbnsutil -sr\_state

Sample output on system currently acting as master:

lpcaph01:/usr/sap/CAP/HDB00> hdbnsutil -sr\_state

checking for active or inactive nameserver ...

System Replication State

~~~~~~~~~~~~~~~~~~~~~~~~

mode: primary

site id: 1

site name: primary\_database\_site

Host Mappings:

~~~~~~~~~~~~~~

lpcaph01 -> [primary\_database\_site] lpcaph01

lpcaph01 -> [secondary\_database\_site] lpcaph51

done.

Sample output on system currently acting as slave:

lpcaph51:/usr/sap/CAP/HDB00> hdbnsutil -sr\_state

checking for active or inactive nameserver ...

System Replication State

~~~~~~~~~~~~~~~~~~~~~~~~

mode: syncmem

site id: 2

site name: secondary\_database\_site

active primary site: 1

Host Mappings:

~~~~~~~~~~~~~~

lpcaph51 -> [primary\_database\_site] lpcaph01

lpcaph51 -> [secondary\_database\_site] lpcaph51

primary masters:lpcaph01

done.

## ****Check master/slave replication status****

su - <sid>adm

python /usr/sap/<SID>/HDB<IN>/exe/python\_support/systemReplicationStatus.py

Sample output for system running as master (some unimportant columns removed for better readability):

lpcaph01:/usr/sap/CAP/HDB00> python /usr/sap/CAP/HDB00/exe/python\_support/systemReplicationStatus.py

| Host | Port | Service Name | Secondary | Secondary | Replication | Replication | Replication |

| | | | Host | Port | Mode | Status | Status Details |

| -------- | ----- | ------------ | --------- | --------- | ----------- | ----------- | -------------- |

| lpcaph01 | 30007 | xsengine | lpcaph51 | 30007 | SYNCMEM | ACTIVE | |

| lpcaph01 | 30001 | nameserver | lpcaph51 | 30001 | SYNCMEM | ACTIVE | |

| lpcaph01 | 30003 | indexserver | lpcaph51 | 30003 | SYNCMEM | ACTIVE | |

status system replication site "2": ACTIVE

overall system replication status: ACTIVE

Sample output for system running as slave:

lpcaph51:/usr/sap/CAP/HDB00> python /usr/sap/CAP/HDB00/exe/python\_support/systemReplicationStatus.py

this system is either not running or not primary system replication site

Local System Replication State

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

mode: SYNCMEM

site id: 2

site name: secondary\_database\_site

active primary site: 1

primary masters: lpcaph01

## ****Display CRM Configuration****

lpcmph51:~ # crm configure show

node lpcmph01 \

attributes hana\_cmp\_vhost=lpcmph01 hana\_cmp\_site=primary\_database\_site hana\_cmp\_srmode=syncmem lpa\_cmp\_lpt=1568073658 hana\_cmp\_remoteHost=lpcmph51 standby=off hana\_cmp\_op\_mode=delta\_datashipping maintenance=off

node lpcmph51 \

attributes lpa\_cmp\_lpt=10 hana\_cmp\_op\_mode=delta\_datashipping hana\_cmp\_vhost=lpcmph51 hana\_cmp\_site=secondary\_database\_site hana\_cmp\_srmode=syncmem hana\_cmp\_remoteHost=lpcmph01 standby=off maintenance=off

primitive rsc\_SAPHanaTopology\_CMP00 ocf:suse:SAPHanaTopology \

operations $id=rsc\_sap2\_CMP00-operations \

op monitor interval=10 timeout=600 \

op start interval=0 timeout=600 \

op stop interval=0 timeout=300 \

params SID=CMP InstanceNumber=00

primitive rsc\_SAPHana\_CMP00 ocf:suse:SAPHana \

operations $id=rsc\_sap\_CMP00-operations \

op start interval=0 timeout=3600 \

node lpcmph01 \

attributes hana\_cmp\_vhost=lpcmph01 hana\_cmp\_site=primary\_database\_site hana\_cmp\_srmode=syncmem lpa\_cmp\_lpt=1568073658 hana\_cmp\_remoteHost=lpcmp

h51 standby=off hana\_cmp\_op\_mode=delta\_datashipping maintenance=off

node lpcmph51 \

attributes lpa\_cmp\_lpt=10 hana\_cmp\_op\_mode=delta\_datashipping hana\_cmp\_vhost=lpcmph51 hana\_cmp\_site=secondary\_database\_site hana\_cmp\_srmode=syn

cmem hana\_cmp\_remoteHost=lpcmph01 standby=off maintenance=off

primitive rsc\_SAPHanaTopology\_CMP00 ocf:suse:SAPHanaTopology \

operations $id=rsc\_sap2\_CMP00-operations \

op monitor interval=10 timeout=600 \

op start interval=0 timeout=600 \

op stop interval=0 timeout=300 \

params SID=CMP InstanceNumber=00

primitive rsc\_SAPHana\_CMP00 ocf:suse:SAPHana \

operations $id=rsc\_sap\_CMP00-operations \

op start interval=0 timeout=3600 \

op stop interval=0 timeout=3600 \

op promote interval=0 timeout=3600 \

op monitor interval=60 role=Master timeout=700 \

op monitor interval=61 role=Slave timeout=700 \

params SID=CMP InstanceNumber=00 PREFER\_SITE\_TAKEOVER=true DUPLICATE\_PRIMARY\_TIMEOUT=7200 AUTOMATED\_REGISTER=true

primitive rsc\_ip\_CMP00 IPaddr2 \

meta target-role=Started is-managed=true \

operations $id=rsc\_ip\_CMP00-operations \

op monitor interval=10s timeout=20s \

params ip=160.46.110.207

primitive sbd-fencing stonith:external/sbd \

op start interval=0 start-delay=1 time=30 \

op monitor interval=3600 timeout=20

ms msl\_SAPHana\_CMP00 rsc\_SAPHana\_CMP00 \

meta is-managed=true notify=true clone-max=2 clone-node-max=1 target-role=Started interleave=true

clone cln-sbd-fencing sbd-fencing \

meta target-role=Started is-managed=true

clone cln\_SAPHanaTopology\_CMP00 rsc\_SAPHanaTopology\_CMP00 \

meta is-managed=true clone-node-max=1 target-role=Started interleave=true

location cli-prefer-rsc\_SAPHana\_CMP00 rsc\_SAPHana\_CMP00 role=Started inf: lpcmph01

colocation col\_saphana\_ip\_CMP00 2000: rsc\_ip\_CMP00:Started msl\_SAPHana\_CMP00:Master

order ord\_SAPHana\_CMP00 2000: cln\_SAPHanaTopology\_CMP00 msl\_SAPHana\_CMP00

property cib-bootstrap-options: \

stonith-enabled=true \

stonith-timeout=75s \

no-quorum-policy=ignore \

placement-strategy=balanced \

expected-quorum-votes=2 \

dc-version=1.1.12-f47ea56 \

cluster-infrastructure="classic openais (with plugin)" \

maintenance-mode=false \

last-lrm-refresh=1526256037

rsc\_defaults rsc-options: \

resource-stickiness=1000 \

migration-threshold=5000

op\_defaults op-options: \

timeout=600 \

record-pending=true

role basisadmin \

write property:maintenance-mode \

read cib

acl\_target adm \

basisadmin

acl\_target cmpadm \

basisadmin

## A more readable Cluster Status

-1 Display the cluster status once on the console and exit

-f Display resource fail counts, and appends the Migration Summary at the end of the output

-n Group resources by node

-r Display inactive resources

-A show the node attributes

lpnwph51:~ # crm\_mon -1fnAr

Stack: corosync

Current DC: lpnwph51 (version 1.1.15-21.1-e174ec8) - partition with quorum

Last updated: Tue Oct 15 09:35:49 2019

Last change: Tue Oct 15 09:35:29 2019 by root via crm\_attribute on lpnwph01

2 nodes configured

7 resources configured

Node lpnwph01: online

sbd-fencing (stonith:external/sbd): Started

rsc\_ip\_NWP00 (ocf::heartbeat:IPaddr2): Started

rsc\_SAPHanaTopology\_NWP00 (ocf::suse:SAPHanaTopology): Started

rsc\_SAPHana\_NWP00 (ocf::suse:SAPHana): Master

Node lpnwph51: online

rsc\_SAPDatabase\_NWQ10 (ocf::heartbeat:SAPDatabase): Started

rsc\_SAPHanaTopology\_NWP00 (ocf::suse:SAPHanaTopology): Started

rsc\_SAPHana\_NWP00 (ocf::suse:SAPHana): Slave

Inactive resources:

Node Attributes:

\* Node lpnwph01:

+ hana\_nwp\_clone\_state : PROMOTED

+ hana\_nwp\_op\_mode : logreplay

+ hana\_nwp\_remoteHost : lpnwph51

+ hana\_nwp\_roles : 4:P:master1:master:worker:master

+ hana\_nwp\_site : primary\_database\_site

+ hana\_nwp\_srmode : syncmem

+ hana\_nwp\_sync\_state : PRIM

+ hana\_nwp\_version : 2.00.033.00.1535711040

+ hana\_nwp\_vhost : lpnwph01

+ lpa\_nwp\_lpt : 1571124929

+ master-rsc\_SAPHana\_NWP00 : 150

\* Node lpnwph51:

+ hana\_nwp\_clone\_state : DEMOTED

+ hana\_nwp\_op\_mode : logreplay

+ hana\_nwp\_remoteHost : lpnwph01

+ hana\_nwp\_roles : 4:S:master1:master:worker:master

+ hana\_nwp\_site : secondary\_database\_site

+ hana\_nwp\_srmode : syncmem

+ hana\_nwp\_sync\_state : SOK

+ hana\_nwp\_version : 2.00.033.00.1535711040

+ hana\_nwp\_vhost : lpnwph51

+ lpa\_nwp\_lpt : 30

+ master-rsc\_SAPHana\_NWP00 : 100

Migration Summary:

\* Node lpnwph01:

\* Node lpnwph51:

###Other outputs you can try:

-L Display negative location constraints

Negative Location Constraints:

loc\_NWQ\_never\_on\_lpnwph01 prevents rsc\_SAPDatabase\_NWQ10 from running on lpnwph01

-o Display resource operation history

-t Display resource operation history with timing details

-c Display cluster tickets

## Other helpful commands

Run using rootsh:

qxf6163@lpnwph51:~> sudo rootsh SAPHanaSR-showAttr

Global cib-time

--------------------------------

global Tue Oct 15 09:58:07 2019

Hosts clone\_state lpa\_nwp\_lpt node\_state op\_mode remoteHost roles site srmode standby sync\_state version vhost

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

lpnwph01 PROMOTED 1571126287 online logreplay lpnwph51 4:P:master1:master:worker:master primary\_database\_site syncmem off PRIM 2.00.033.00.1535711040 lpnwph01

lpnwph51 DEMOTED 30 online logreplay lpnwph01 4:S:master1:master:worker:master secondary\_database\_site syncmem off SOK 2.00.033.00.1535711040 lpnwph51

Global cib-time

--------------------------------

global Tue Oct 15 09:58:07 2019

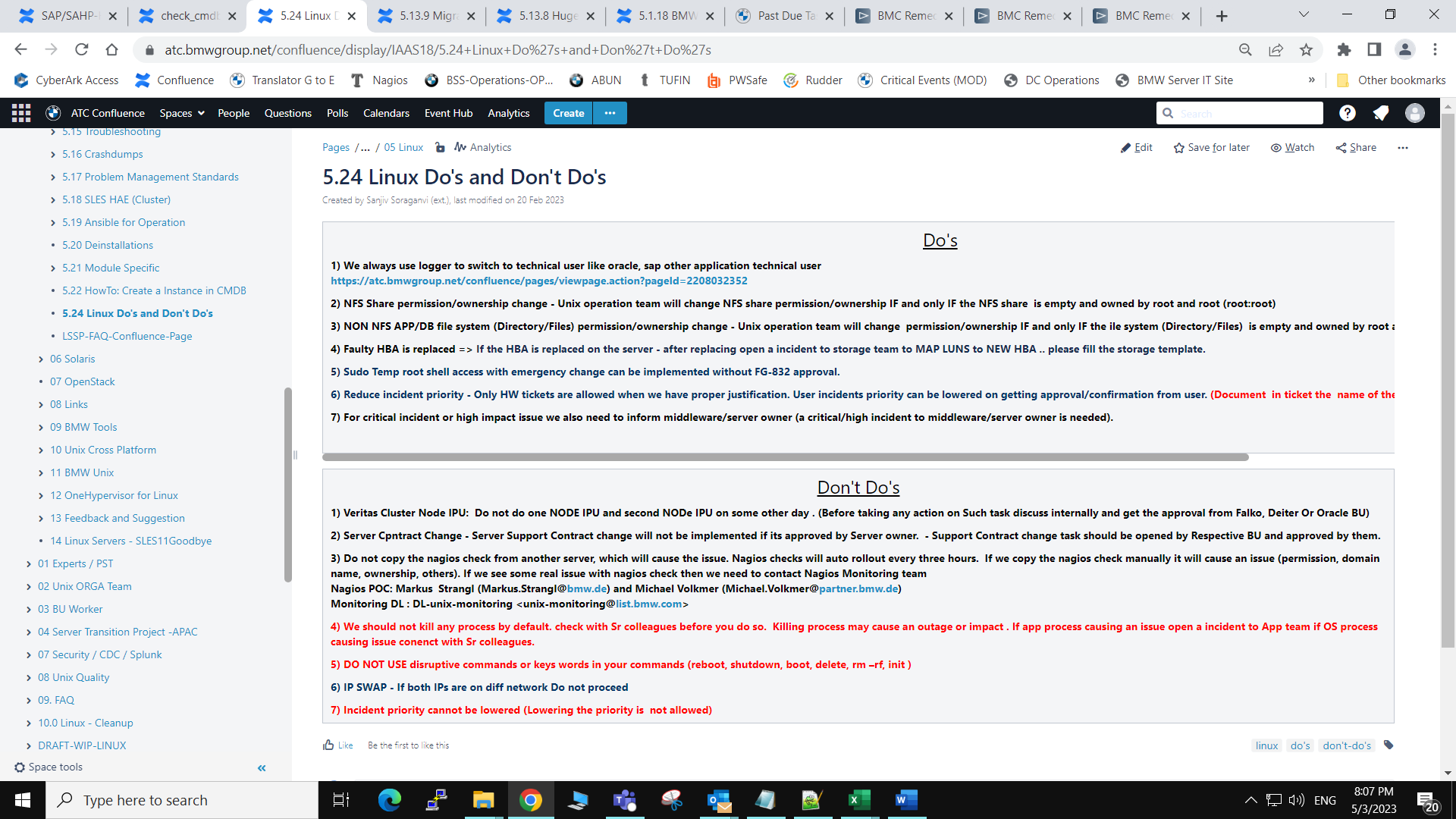
Hosts node\_state standby

----------------------------

lpnwph01 online off

lpnwph51 online off

Last edited by Schrocko Winfried, (Winfried.Schrocko@partner.bmw.de) , based on work by Kemmerer Walter, (Walter.Kemmerer@partner.bmw.de) , Porezag Dirk, (Dirk.Porezag@partner.bmw.de) and Vonderbecke Michael, FG-AM-81 .  
Page last modified on Thursday 17 of October, 2019 16:21:50 CEST. (Version 88)



# [BTRFS Balancing Disable](https://atc.bmwgroup.net/confluence/display/IAAS18/BTRFS+Balancing+Disable)

* Created by Unknown User (qxy1272), last modified on [30 Jan 2020](https://atc.bmwgroup.net/confluence/pages/diffpagesbyversion.action?pageId=621874912&selectedPageVersions=3&selectedPageVersions=4)

1- Please remove the weekly cron for BTRFS balancing .

SERVERX:~ # ll /etc/cron.weekly

total 8

drwxr-xr-x 1 root root   46 May  4  2019 .

drwxr-xr-x 1 root root 3980 Jan 15 01:19 ..

lrwxrwxrwx 1 root root   44 May  4  2019 btrfs-balance -> /usr/share/btrfsmaintenance/btrfs-balance.sh

lrwxrwxrwx 1 root root   41 May  4  2019 btrfs-trim -> /usr/share/btrfsmaintenance/btrfs-trim.sh

SERVERX:~ # rm "/etc/cron.weekly/btrfs-balance"

SERVERX:~ # ll /etc/cron.weekly

total 4

drwxr-xr-x 1 root root   20 Jan 23 17:02 .

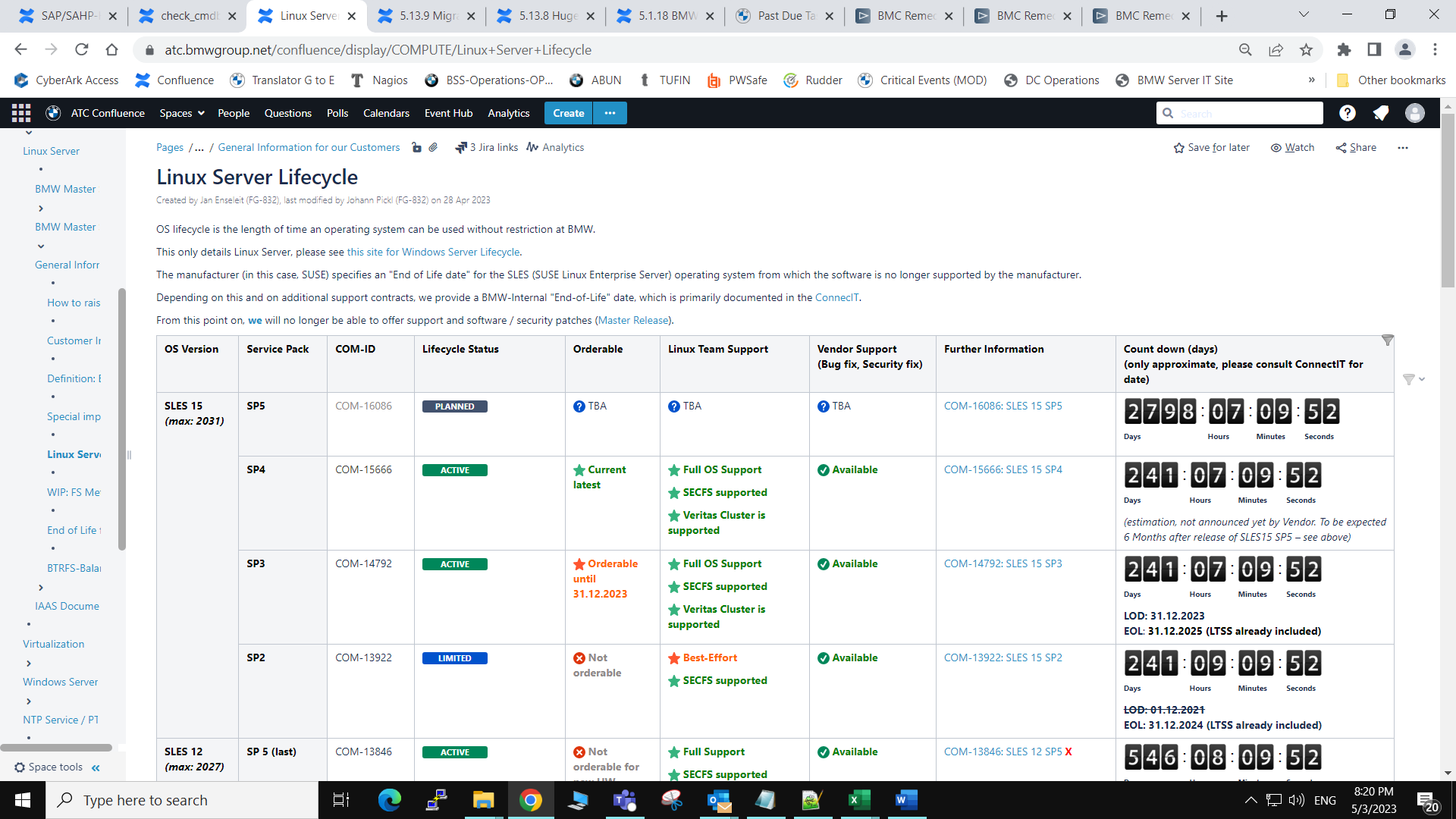
drwxr-xr-x 1 root root 3980 Jan 15 01:19 ..

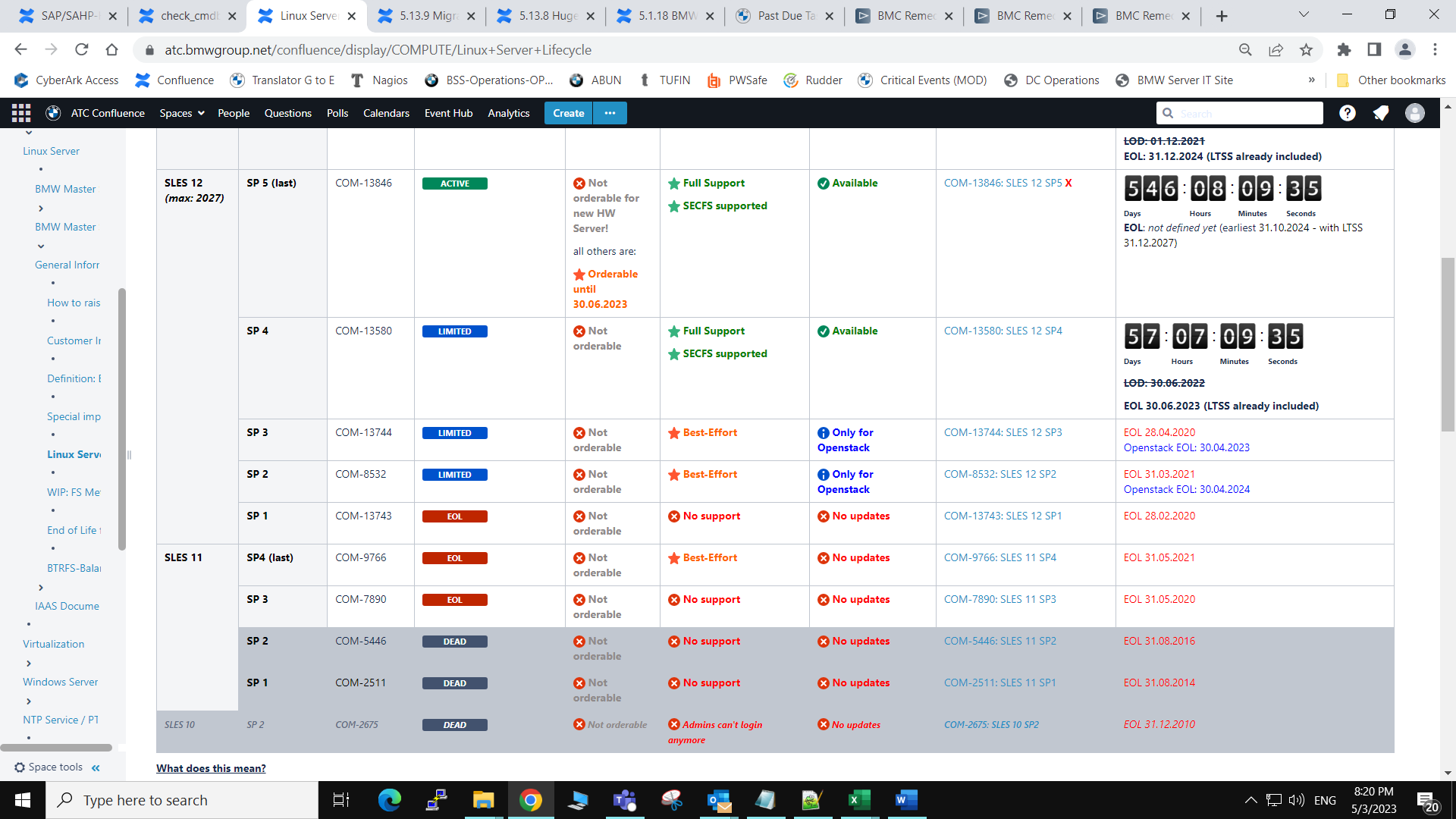
lrwxrwxrwx 1 root root   41 May  4  2019 btrfs-trim -> /usr/share/btrfsmaintenance/btrfs-trim.sh

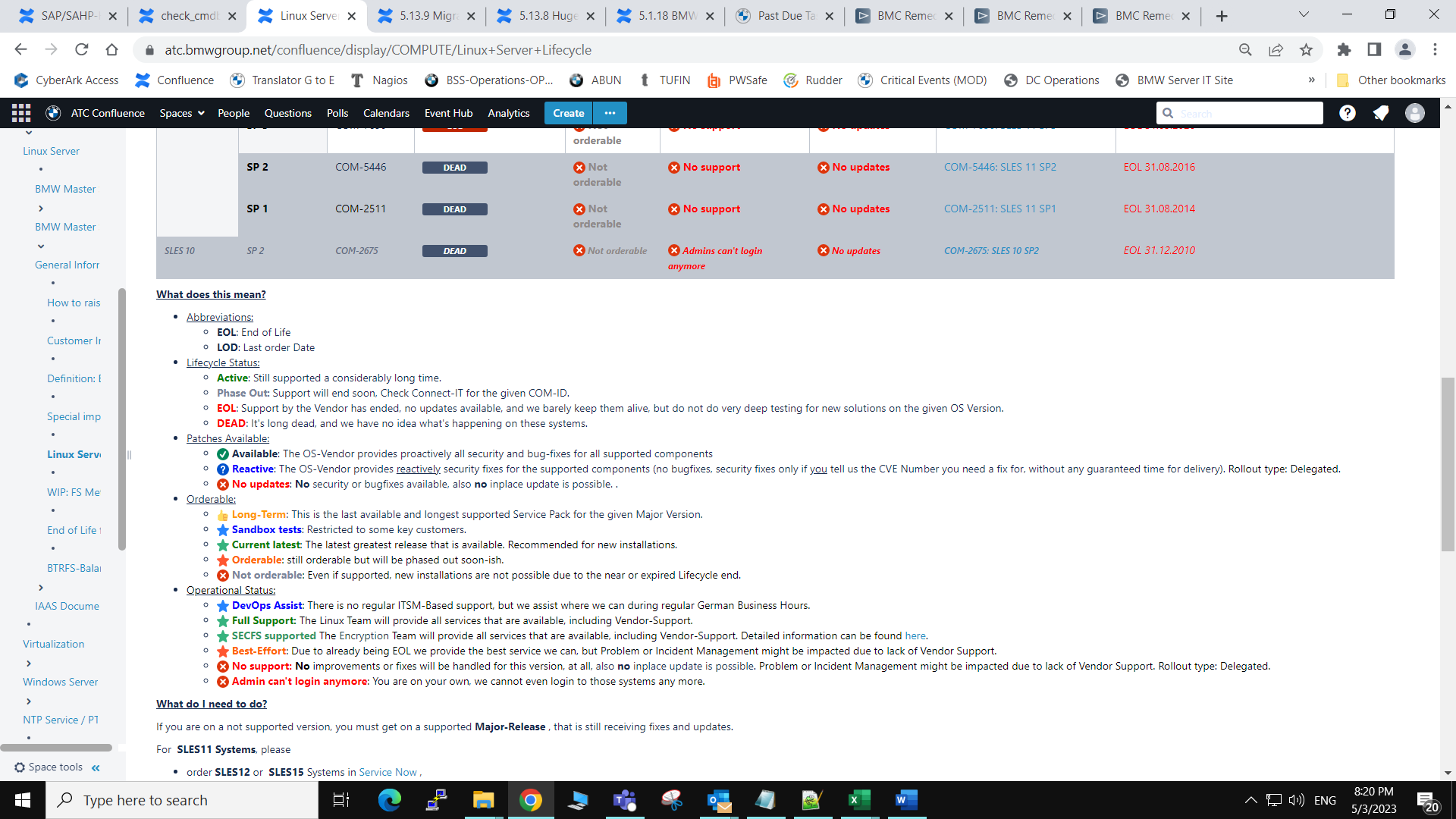
2- Please disable the BTRFS maintenance service

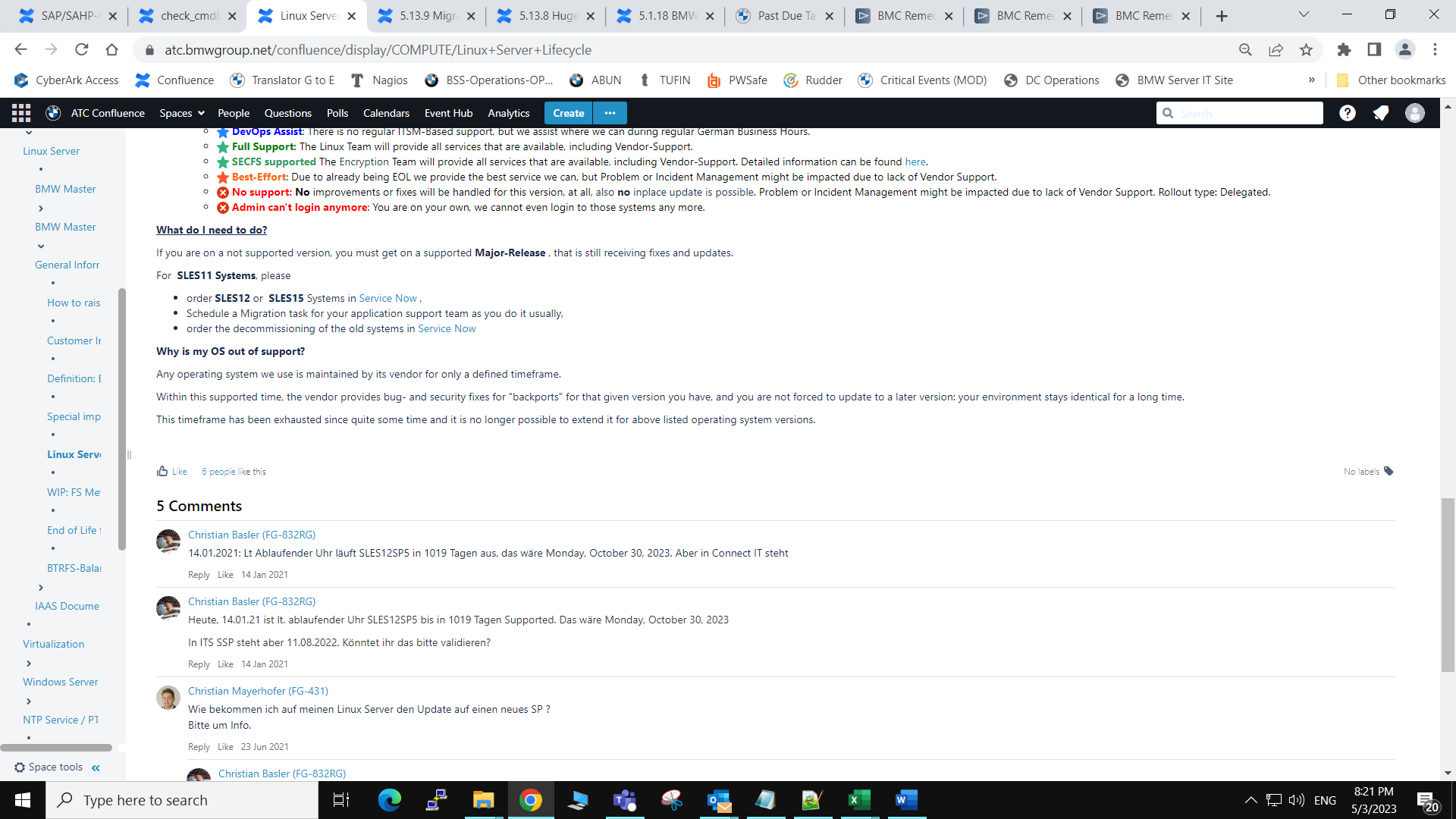
SERVERX:~ # systemctl mask btrfsmaintenance-refresh.service

Created symlink from /etc/systemd/system/btrfsmaintenance-refresh.service to /dev/null.









/global/instserv/home/q545913/ansible/ansible-playbooks/OS\_FIX\_lvm.yml

[INSTSERV][qxz10kv][lpinstiaas02] ~ $ cat /global/instserv/home/q545913/ansible/ansible-playbooks/OS\_FIX\_lvm.yml

---

- name: Ansible playbook to fix the lvm.conf

hosts: all

become: true

gather\_facts: false

connection: ssh

tasks:

- name: searching for active global\_filter

shell: grep -q "^\\s\*global\_filter = \[" /etc/lvm/lvm.conf

ignore\_errors: true

failed\_when: no

changed\_when: no

register: global\_filter\_found

# Active global\_filter found, ensure, that the content is valid

- name: active setting found, ensuring valid content

lineinfile:

path: /etc/lvm/lvm.conf

line: "\\1 \"r|/dev/md/.\*\_[0-9]\*$|\" ]"

backup: true

regexp: '(^\s\*global\_filter = \[)'

backrefs: true

when: global\_filter\_found.rc == 0

- name: searching for commented global\_filter

shell: grep -q "^\\s\*\# \*global\_filter = \[" /etc/lvm/lvm.conf

ignore\_errors: true

failed\_when: no

changed\_when: no

register: commented\_global\_filter\_found

# Uncommented global\_filter found, enabling it with the correct filter

- name: Commented global\_filter found, enabling it with the correct filter

lineinfile:

path: /etc/lvm/lvm.conf

line: "\\1\\3 \"r|/dev/md/.\*\_[0-9]\*$|\" ]"

backup: true

regexp: '(^\s\*)\#( \*)(global\_filter = \[)'

backrefs: true

when: commented\_global\_filter\_found.rc == 0 and global\_filter\_found.rc != 0

# No active global\_filter found, setting up one

- name: No active global\_filter found, setting up one

lineinfile:

path: /etc/lvm/lvm.conf

line: "global\_filter = [ \"r|/dev/md/.\*\_[0-9]\*$|\" ]"

backup: true

insertafter: '^devices {'

when: commented\_global\_filter\_found.rc != 0 and global\_filter\_found.rc != 0

# ensure that all other uncommented instances of global\_filter is removed from the lvm.conf

- name: remove uncommented global\_filter entries

lineinfile:

path: /etc/lvm/lvm.conf

regex: '^\s\*\#\s\*global\_filter = \['

state: absent

backup: true

- name: searching for active auto\_activation\_volume\_list setting

shell: grep -q "^\\s\*auto\_activation\_volume\_list = \[" /etc/lvm/lvm.conf

ignore\_errors: true

failed\_when: no

changed\_when: no

register: auto\_activation\_volume\_list\_found

- name: active auto\_activation\_volume\_list found, ensuring valid content

lineinfile:

path: /etc/lvm/lvm.conf

line: "\\1 \"vglocal\", \"vgDRBD\" ]"

backup: true

regexp: '(^\s\*auto\_activation\_volume\_list = \[)'

backrefs: true

when: auto\_activation\_volume\_list\_found.rc == 0

- name: searching for commented auto\_activation\_volume\_list setting

shell: grep -q "^\\s\*\#\\s\*auto\_activation\_volume\_list = \[" /etc/lvm/lvm.conf

ignore\_errors: true

failed\_when: no

changed\_when: no

register: commented\_auto\_activation\_volume\_list\_found

- name: enable the commented auto\_activation\_volume\_list in the lvm.conf with correct content

lineinfile:

path: /etc/lvm/lvm.conf

line: "\\1\\3 \"vglocal\", \"vgDRBD\" ]"

backup: true

regexp: '(^\s\*)\#( \*)(auto\_activation\_volume\_list = \[)'

backrefs: true

when: commented\_auto\_activation\_volume\_list\_found.rc == 0 and auto\_activation\_volume\_list\_found.rc != 0

- name: enable the auto\_activation\_volume\_list when the line was missing

lineinfile:

path: /etc/lvm/lvm.conf

line: "auto\_activation\_volume\_list = [ \"vglocal\", \"vgDRBD\" ]"

backup: true

insertafter: '^activation {'

when: commented\_auto\_activation\_volume\_list\_found.rc != 0 and auto\_activation\_volume\_list\_found.rc != 0

- name: remove commented auto\_activation\_volume\_list entries

lineinfile:

path: /etc/lvm/lvm.conf

regex: '^\s\*\#\s\*auto\_activation\_volume\_list = \['

backup: true

state: absent

- name: searching for duplicates

shell: grep -c "^\\s\*auto\_activation\_volume\_list = \[" /etc/lvm/lvm.conf

ignore\_errors: true

failed\_when: no

changed\_when: no

register: auto\_activation\_dups

- name: searching for duplicates

shell: grep -c "^\\s\*global\_filter = \[" /etc/lvm/lvm.conf

ignore\_errors: true

failed\_when: no

changed\_when: no

register: global\_filter\_dups

- name: search results

fail:

msg: "DUPLICATES OR ERROR FOUND! MANUAL CHECK REQUIRED!"

when: (global\_filter\_dups.stdout != '1' or auto\_activation\_dups.stdout != '1')

[INSTSERV][qxz10kv][lpinstiaas02] ~ $

[INSTSERV][qxz10kv][lp09instiaas02] /var/tmp/amit/test $ ansiblectl apply --playbook=OS\_FIX\_lvm.yml --hostlist=IAAS\_NTT\_LI\_UX\_SRV-Test-lp09instiaas02