Linux

Mostly Red Hat and it's clones.

CentOS 7 GSSAPI module

The GSSPI module has been built as a replacement for the aging mod_auth_kerb. Its aim is to use only GSSAPI calls and be as much as possible agnostic of the actual mechanism used.

Installing packages

```
yum install -y epel-release
yum install -y krb5-workstation krb5-devel krb5-libs mod_auth_gssapi mod_session
```

Prepare a `/etc/krb5.conf` against the AD environment

```
includedir /etc/krb5.conf.d/

[logging]
 default = FILE:/var/log/krb5libs.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind.log

[libdefaults]
  default_realm = SAMPLE.COM
  default_tkt_enctypes = aes256-cts-hmac-sha1-96 aes256-cts aes128-cts-hmac-sha1-96 aes128-cts rc4-hmac
  default_tgs_enctypes = aes256-cts-hmac-sha1-96 aes256-cts aes128-cts-hmac-sha1-96 aes128-cts rc4-hmac
  permitted_enctypes = aes256-cts-hmac-sha1-96 aes256-cts aes128-cts-hmac-sha1-96 caes128-cts rc4-hmac
  forwardable = true
  dns_lookup_realm = false
  dns_lookup_kdc = false

[realms]
  SAMPLE.COM  = {
    kdc = domainc1.sample.com
    default_domain = SAMPLE.COM
}

[domain_realm]
  .SAMPLE.COM = SAMPLE.COM
  SAMPLE.COM = SAMPLE.COM
```

Check the time. Kerberos is extremely time sensitive.

```
ntpdate domainc1.sample.com
```

Test the login

```
kinit Administrator@SAMPLE.COM
klist
kdestroy
```

CentOS Linux 8 to CentOS Stream

Step 1: Enable CentOS Stream Repo

```
dnf install centos-release-stream
```

Step 2: Set CentOS Stream repo as the default

```
dnf swap centos-{linux,stream}-repos
```

Step 3: Synchronize installed packages to the latest versions

```
dnf distro-sync
```

When complete, reboot and your CentOS 8 Linux is now CentOS Steam.

Good bye CentOS. I will miss you.

Custom CentOS ISO

There are many reason to create a custom installer. This example was created to add the console output to the 1st serial port to deploy CentOS on a headless device. This process works for CentOS 7 and 8/Stream.

Before you begin, make sure you have genisoimage and syslinux installed

```
yum install -y genisoimage syslinux
```

1. Download .iso from a CentOS mirror. See https://www.centos.org/download/mirrors/ for a list of mirrors nearest to you.

```
wget http://linux.cc.lehigh.edu/centos/7.6.1810/isos/x86_64/CentOS-7-x86_64-Minimal-1810.iso
```

2. Mount .iso to a directory

```
sudo mount -o loop -t iso9660 CentOS-7-x86_64-Minimal-1810.iso centos_iso
```

3. copy its contents to another working directory:

```
cp -rf centos_iso/ centos_customized_iso
```

4. edit centos_customized_iso/isolinux/isolinux.cfg, overwrite its contents with the following. DO NOT alter the LABEL.

```
default linux
prompt 1
timeout 50

label linux
kernel vmlinuz
append initrd=initrd.img inst.stage2=hd:LABEL=CentOS\x207\x20x86_64 console=tty0
console=ttyS0,115200n8

label text
kernel vmlinuz
append initrd=initrd.img text inst.stage2=hd:LABEL=CentOS\x207\x20x86_64 console=tty0
console=ttyS0,115200n8

label check
kernel vmlinuz
```

```
append initrd=initrd.img inst.stage2=hd:LABEL=CentOS\x207\x20x86_64 rd.live.check quiet
```

5. prepare .iso again from centos_customized_iso directory:

```
sudo mkisofs -r -V “CentOS 7 x86_64” -cache-inodes -J -l -b isolinux/isolinux.bin -c isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-info-table -o custom.iso centos_customized_iso
```

6. run isohybrid on the new .iso (otherwise it won’t boot from USB)

```
isohybrid custom.iso
```

7. Make sure .iso is proper with a format like this, with “file” command:

```
file custom.iso
```

The output will be similar to this:

```
custom.iso: DOS/MBR boot sector ISO 9660 CD-ROM filesystem data (DOS/MBR boot sector) ‘CentOS 7 x86_64’ (bootable); partition 1 : ID=0x17, active, start-CHS (0x0,0,1), end-CHS (0x2ae,63,32), startsector 0, 1406976 sectors
```

8. Burn this to USB with dd command. Be sure you are outputting to the correct device. A mistake here will wipe data or your root.

```
dd if=custom.iso of=/dev/sdb bs=1MB
```

9. Insert this USB to your machine and boot from USB, and in the boot: prompt, type “linux text” as the install command

```
boot: linux text
```

To further customize and automate the installation, add a Kickstart to the append.

LVM with cache

Create a logical volume with nvme cache.

```
vgcreate storage /dev/sdb /dev/nvme0n1
lvcreate -L 6T -n lv storage /dev/sdb
lvcreate -L 900G -n lv_cache storage /dev/nvme0n1
lvcreate -L 9G -n lv_meta storage /dev/nvme0n1

lvconvert --type cache-pool --cachemode writethrough --poolmetadata storage/lv_cache_meta storage/lv_cache

lvconvert --type cache --cachepool storage/lv_cache storage/lv
```

Move RHEL Users

Step 1, on source

Run the following command**s** as root source) system which has users configured

```
ID_minimum=500
for f in /etc/{passwd,group}; do awk -F: -vID=$ID_minimum '$3>=ID && $1!="nfsnobody"' $f |sort -nt: -k3 >
${f#/etc/}.bak; done
while read line; do grep -w "^${line%%:*}" /etc/shadow; done <passwd.bak >shadow.bak
while read line; do grep -w "^${line%%:*}" /etc/gshadow; done <group.bak >gshadow.bak
```

After running the above, 4 new files will be in the current directory (`passwd.bak`, `group.bak`, `shadow.bak`, and `gshadow.bak`). Inspect them and then transfer to the new *destination* system.

Step 2, on destination

Run the following command as root on the destination system in a directory containing the four `.bak` files.

```
for f in {passwd,group,shadow,gshadow}.bak; do cat $f >>/etc/${f%.bak}; done
```

Step 3, on destination

Run the following **final compound command destination system in the same directory as the previous step**

```
for uidgid in $(cut -d: -f3,4 passwd.bak); do
    dir=$(awk -F: /$uidgid/{print\$6} passwd.bak)
    mkdir -vm700 "$dir"; cp -r /etc/skel/.[[:alpha:]]* "$dir"
    chown -R $uidgid "$dir"; ls -ld "$dir"
done
```

This final command will setup home directories for the users.

NFS on ZFS HA Cluster

Build a high-available dual-controller storage array using open-source technologies.
This solution should be capable of presenting shared storage to NFS client and can be expanded to iSCSI and FCoE.

What is needed:

- 2 nodes with A LOT of memory.
- Fast storage that is directly attached to each node.
- Root access.
- The ZFS filesystem configured via the ZFS on Linux repository.
- Familiarity with basic ZFS operations: zpool/zfs filesystem creation and modification.
- Understanding of network under RHEL Linux.

These steps describe the construction of a two host, single JBOD cluster that manages a single ZFS pool.

On both nodes:

Get EPEL and ZFS repositories

```
yum install -y epel-release
yum install -y http://download.zfsonlinux.org/epel/zfs-release.el7_8.noarch.rpm
yum update -y
yum install -y kernel-devel zfs
systemctl preset zfs-import-cache zfs-import-scan zfs-mount zfs-share zfs-zed zfs.target
systemctl enable zfs-import-scan
systemctl start zfs-import-scan
```

Add some clustering requirements:

```
yum install -y pcs fence-agents-all device-mapper-multipath nfs-utils
touch /etc/multipath.conf
systemctl start multipathd
systemctl enable multipathd
```

Add each node to the /etc/hosts file.

```
echo "172.16.100.142 node1" >> /etc/hosts
echo "172.16.100.144 node2" >> /etc/hosts
```

Set the hacluster password:

```
passwd hacluster wljgnFAW4fgwEGF21
```

Add needed heartbeat files

```
cd /usr/lib/ocf/resource.d/heartbeat/
wget https://raw.githubusercontent.com/clusterapps/stmf-ha/master/heartbeat/ZFS
wget https://github.com/ClusterLabs/resource-agents/raw/master/heartbeat/iSCSITarget
wget https://github.com/ClusterLabs/resource-agents/raw/master/heartbeat/iSCSILogicalUnit
chmod a+x ./ZFS
chmod a+x ./iSCSILogicalUnit
chmod a+x ./iSCSITarget
```

Update the firewall configuration.

```
firewall-cmd --add-service=nfs --permanent
firewall-cmd --add-service=high-availability --permanent
firewall-cmd --permanent --add-service=nfs
firewall-cmd --permanent --add-service=mountd
firewall-cmd --permanent --add-service=rpc-bind
firewall-cmd --reload
```

Enable the services and reboot. This step helps verify all of the services are properly configured on boot.

```
systemctl enable pcsd
systemctl enable corosync
systemctl enable pacemaker
reboot
```

On the Primary

Create the pool. You will need the path to the devices. Get them from:

```
ls -l /dev/disk/by-id/
```

Run zpool create (Your devices will be different!)
This pool has similar qualities to a RAID5+0. You should build what you need.

```
zpool create array1 -o ashift=12 -o autoexpand=on -o autoreplace=on -o cachefile=none \
raidz1 /dev/disk/by-id/scsi-35000c500740a6277 /dev/disk/by-id/scsi-35000c5007411d36b /dev/disk/by-id/scsi-35000c5007411cb1b \
raidz1 /dev/disk/by-id/scsi-35000c50076703a9f /dev/disk/by-id/scsi-35000c50070d3a853 /dev/disk/by-id/scsi-35000c50076701f57 \
raidz1 /dev/disk/by-id/scsi-35000c5007411cfa7 /dev/disk/by-id/scsi-35000c5007411d0bf /dev/disk/by-id/scsi-35000c500740a109f \
log mirror /dev/disk/by-id/scsi-35000c5007670107f /dev/disk/by-id/scsi-35000c5007411d467 spare /dev/disk/by-id/scsi-35000c50076bd0c03
```

Update a few ZFS settings

```
zfs set acltype=posixacl array1
zfs set atime=off array1
zfs set xattr=sa array1
zfs set compression=lz4 array1
```

Authorize Cluster

```
pcs cluster auth node1 node2
pcs cluster setup --start --name NASOne node1 node2
```

Set some cluster properties and ad the resources

```
pcs property set no-quorum-policy=ignore
pcs stonith create fence-array1 fence_scsi pcmk_monitor_action="metadata" pcmk_host_list="node1 node2" \
devices="/dev/mapper/35000c5007670107f,/dev/mapper/35000c5007411d467,/dev/mapper/35000c50076bd0c03" \
meta provides=unfencing --group=group-array1
pcs resource create array1-ip IPaddr2 ip=172.16.100.99 cidr_netmask=24 --group group-array1
pcs resource create array1 ZFS pool="array1" importargs="-d /dev/mapper/" op start timeout="90" op stop timeout="90" --group=group-array1
pcs resource defaults resource-stickiness=100
```

Create and share a ZFS directory.

```
zfs create array1/nfs1
zfs set sharenfs=rw=@172.16.100.0/24,sync,no_root_squash,no_wdelay array1/nfs1
```

Enable and start NFS released services.

```
systemctl enable rpcbind nfs-server
systemctl start rpcbind nfs-server
```

Check the status of the cluster.

```
pcs cluster status
pcs status resources
showmount -e localhost
```

RHEL Move Printers

To migrate all or most the printers from one system to another, run the following commands on the **old system**:

Copy all the files in `/etc/cups/ppd` to a temporary location.

```
scp /etc/cups/ppd/*.ppd newsys:/tmp/
```

Create a list of all the printers on the old system. Copy file to new system

```
lpstat -v > /tmp/printers.txt
scp /tmp/printers.txt newsys:/tmp/printers.txt
```

Copy the lpoptions file, if you have one, to a temporary location.

```
scp /etc/cups/lpoptions newsys:/tmp/lpoptions
```

On the **new system**:

Create a script to read the printers file and create the devices on the new system.

```
cat /tmp/printers.txt \
| sed -e 's/device for //' -e 's/: / /'  \
| ( while read p u; do
      if [ -e /tmp/${p}.ppd ]; then
         echo lpadmin -p ${p} -P /tmp/${p}.ppd -v ${u} -E
      else
         echo lpadmin -p ${p} -v ${u} -E
      fi
  done) > create-prints.sh
```

Verify the create-printers.sh and then run it.

```
chmod +x ./create-prints.sh
./create-prints.sh
```

Copy the CUPS options.

```
cat /tmp/lpoptions >> /etc/cups/lpoptions
```

Verify printers were created.

```
lpstat -v
```

systemd services

Various systemd services.

Oracle WebLogic

/etc/systemd/system/wls_nodemanager.service

```
[Unit]
Description=WebLogic nodemanager service

[Service]
Type=simple
WorkingDirectory=/u02/oracle/domains/base_domain
ExecStart=//u02/oracle/domains/base_domain/bin/startNodeManager.sh
ExecStop=/u02/oracle/domains/base_domain/bin/stopNodeManager.sh
User=oracle
Group=oinstall
KillMode=process
LimitNOFILE=65535

[Install]
WantedBy=multi-user.target
```

/etc/systemd/system/wls_adminserver.service

```
[Unit]
Description=WebLogic Adminserver service

[Service]
Type=simple
WorkingDirectory=/u02/oracle/domains/base_domain
ExecStart=/u02/oracle/domains/base_domain/startWebLogic.sh
ExecStop=/u02/oracle/domains/base_domain/bin/stopWebLogic.sh
User=oracle
Group=oinstall
KillMode=process
```

```
LimitNOFILE=65535

[Install]
WantedBy=multi-user.target
```

tmux

Session Control (from the command line)

`tmux`	Start a new session
`tmux new -s <session-name>`	Start a new session with the name chosen
`tmux ls`	List all sessions
`tmux attach -t <target-session>`	Re-attach a detached session
`tmux attach -d -t <target-session>`	Re-attach a detached session (and detach it from elsewhere)
`tmux kill-session -t <target-session>`	Delete session

Pane Control

`Ctrl` `b` , `"`	Split pane horizontally
`Ctrl` `b` , `%`	Split pane vertically
`Ctrl` `b` , `o`	Next pane
`Ctrl` `b` , `;`	Previous pane
`Ctrl` `b` , `q`	Show pane numbers
`Ctrl` `b` , `z`	Toggle pane zoom
`Ctrl` `b` , `!`	Convert pane into a window
`Ctrl` `b` , `x`	Kill current pane
`Ctrl` `b` , `Ctrl` `O`	Swap panes
`Ctrl` `b` , `t`	Display clock
`Ctrl` `b` , `q`	Transpose two letters (delete and paste)
`Ctrl` `b` , `{`	Move to the previous pane
`Ctrl` `b` , `}`	Move to the next pane
`Ctrl` `b` , `Space`	Toggle between pane layouts
`Ctrl` `b` , `↑`	Resize pane (make taller)
`Ctrl` `b` , `↓`	Resize pane (make smaller)
`Ctrl` `b` , `←`	Resize pane (make wider)
`Ctrl` `b` , `→`	Resize pane (make narrower)

Window Control

Ctrl b , c	Create new window
Ctrl b , d	Detach from session
Ctrl b , ,	Rename current window
Ctrl b , &	Close current window
Ctrl b , w	List windows
Ctrl b , p	Previous window
Ctrl b , n	Next window

Copy-Mode (Emacs)

Ctrl b , [	Enter copy mode
Ctrl b , M-<	Bottom of history
Ctrl b , M->	Top of history
Ctrl b , M-m	Back to indentation
Ctrl b , M-w	Copy selection
Ctrl b , M-y	Paste selection
Ctrl b , Ctrl g	Clear selection
Ctrl b , M-R	Cursor to top line
Ctrl b , M-r	Cursor to middle line
Ctrl b , ↑	Cursor Up
Ctrl b , ↓	Cursor Down
Ctrl b , ←	Cursor Left
Ctrl b , →	Cursor Right

Copy-Mode (vi)

Ctrl b , [	Enter copy mode
Ctrl b , G	Bottom of history
Ctrl b , g	Top of history
Ctrl b , Enter	Copy selection
Ctrl b , p	Paste selection
Ctrl b , k	Cursor Up
Ctrl b , j	Cursor Down
Ctrl b , h	Cursor Left
Ctrl b , l	Cursor Right

FFMPEG on EL 8

```
dnf install -y epel-release
dnf install -y https://download1.rpmfusion.org/free/el/rpmfusion-free-release-8.noarch.rpm
https://download1.rpmfusion.org/nonfree/el/rpmfusion-nonfree-release-8.noarch.rpm -y &&
dnf install http://rpmfind.net/linux/centos/8-stream/PowerTools/x86_64/os/Packages/SDL2-2.0.10-
2.el8.x86_64.rpm &&
dnf install -y ffmpeg ffmpeg-devel
```