RedHat family (Oracle Linux, Alma Linux, Rocky Linux)

- repos, packages
  - rpm, yum and dnf
- Repository proxy server aka satellite server for Redhat family.
- After cloning
- Password change for OL9

repos, packages

package managers

repos, packages

rpm, yum and dnf

Look for packages

Install packages

```
dnf install pkg
dnf localinstall pkg
```

What is installed?

```
dnf list installed
```

Erase packages

```


dnf remove name


rpm --test -e pkg
rpm -e pkg
rpm -e --allmatches --nodeps --noscripts --notriggers --test pkg
```

Repository proxy server aka satellite server for Redhat family.

This setup is done on Oracle Linux v9.5.

Advantages and disadvantages:

```
+ One source of truth
+ Less Internet traffic
+ Local machines will be updated faster, as repository is nearby.
+ Integrity is ensured by checking delivering keys to the clients
+ Easy and simple to manage
? Not tested with several different major version at on the same repository server.
! Repository may be very big, if all versions of packages will be enabled.
```

Preparations

Set up local hostname

```
sudo su
hostnamectl hostname lt58ncp1sat1
exit
# login to ensure hostname is applied
```

Check Internet connectivity

```
curl myip.2dz.fi
```

Carefully observe original repository definition files. It has many repos listed and some of them are disabled. We shall need these. Ensure they are all enabled (`enabled=1`):

```
/etc/yum.repo.d/oracle-linux-ol9.repo
```

```
[ol9_baseos_latest]
name=Oracle Linux 9 BaseOS Latest ($basearch)
baseurl=https://yum$ociregion.$ocidomain/repo/OracleLinux/OL9/baseos/latest/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1

[ol9_appstream]
name=Oracle Linux 9 Application Stream Packages ($basearch)
baseurl=https://yum$ociregion.$ocidomain/repo/OracleLinux/OL9/appstream/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1

[ol9_codeready_builder]
name=Oracle Linux 9 CodeReady Builder ($basearch) - (Unsupported)
baseurl=https://yum$ociregion.$ocidomain/repo/OracleLinux/OL9/codeready/builder/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1
```

and

```
vi /etc/yum.repos.d/uek-ol9.repo
```

```
[ol9_UEKR7]
name=Oracle Linux 9 UEK Release 7 ($basearch)
baseurl=https://yum$ociregion.$ocidomain/repo/OracleLinux/OL9/UEKR7/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1
```

There are additional repositories, which depends on your environment and subscription, such as "Unbreakable Enterprise Kernel (UEK)", "Remote Direct Memory Access (RDMA)", "Virtualization (KVM) Utilities", "Latest Red Hat Compatible Kernel (RHCK) with fixes". Consider enabling them when there is a need for them.

As decscribed in the blog post below, Oracle gives a choice which Kernel to run, depending on that respective repositories should be enabled:

```
https://blog.mythics.com/posts/uek-vs-rhck-kernels
```

Observe enabled repositories, their IDs will be needed later in sync operations

```
dnf repolist
```

```
repo
id
                   repo name
ol9_UEKR7              Oracle Linux 9 UEK Release 7 (x86_64)
ol9_appstream          Oracle Linux 9 Application Stream Packages (x86_64)
ol9_baseos_latest      Oracle Linux 9 BaseOS Latest (x86_64)
ol9_codeready_builder  Oracle Linux 9 CodeReady Builder (x86_64) - (Unsupported)
```

As primary repositories are verified, let's perform a system update and reboot.

```
dnf update
shutdown -r now
```

Install necessary utils

```
dnf install \
    createrepo \
    yum-utils \
    wget \
    tmux \
    tcpdump
```

Configure and attach disk for repositories. Bear in mind that directory will grow, consider using LVM.

```
hostname
lsblk
df -h
export dir="/mnt/$(hostname)-data/data/www/repos/"
mkdir -p ${dir}/keys/
echo "hello-hello" > ${dir}/hello
```

Install webserver to serve packages

```
dnf install nginx

systemctl start nginx
systemctl enable nginx
systemctl status nginx
ss -ntap | grep nginx

vi /etc/nginx/conf.d/$(hostname).conf
```

```
server {
    listen 80 default_server;

    server_name lt58ncp1sat1;

    root /mnt/lt58ncp1sat1-data/data/www/repos/;
    index index.html;

    location / {
        autoindex on;
    }
}
```

Permissions and SElinux

```
chown root:root ${dir}
chmod 775 ${dir}
chcon -Rt httpd_sys_content_t ${dir}
getenforce
nginx -t
nginx -s reload
ss -ntap | grep nginx
sudo -u nginx cat ${dir}/hello
curl http://127.0.0.1/hello
```

Configure firewall and check from externally:

```
firewall-cmd --add-service=http  --permanent
firewall-cmd --add-service=https --permanent

# from another machine
```

```
curl http://192.168.62.151/hello
```

Let's add more repositories (optional)

Add EPEL repos (will download and install `oracle-epel-release-el9`)

```
dnf install epel-release
# dnf install oracle-epel-release-el9
```

Add ClusterControl repo and keys

```
wget http://www.severalnines.com/downloads/cmon/s9s-repo.repo -P /etc/yum.repos.d/
```

and for s9s-tools, add below block to the end of the file:

```
vi /etc/yum.repos.d/s9s-repo.repo
```

```
[s9s-tools]
name=s9s-tools
type=rpm-md
baseurl=http://repo.severalnines.com/s9s-tools/CentOS_9
gpgcheck=1
gpgkey=http://repo.severalnines.com/s9s-tools/CentOS_9/repodata/repomd.xml.key
enabled=1
```

Download keys for repo, they will be needed by clients.

```
wget http://repo.severalnines.com/severalnines-repos.asc -P ${dir}/keys/s9s-repo/
rpm --import ${dir}/keys/s9s-repo/*
wget http://repo.severalnines.com/s9s-tools/CentOS_9/repodata/repomd.xml.key -P
${dir}/keys/s9s-tools/
rpm --import ${dir}/keys/s9s-tools/*
```

Add MariaDB repo

```
curl -LsS https://r.mariadb.com/downloads/mariadb_repo_setup | sudo bash
```

Download keys for repo, they will be needed by clients.

```
mkdir ${dir}/keys/mariadb/
cp /etc/pki/rpm-gpg/MariaDB-Server-GPG-KEY     ${dir}/keys/mariadb/
cp /etc/pki/rpm-gpg/MariaDB-MaxScale-GPG-KEY   ${dir}/keys/mariadb/
```

```
cp /etc/pki/rpm-gpg/MariaDB-Enterprise-GPG-KEY ${dir}/keys/mariadb/
rpm --import ${dir}/keys/mariadb/*
```

Adding Prometheus repo

```
/etc/yum.repos.d/prometheus.repo
```

```
[prometheus]
name=prometheus
baseurl=https://packagecloud.io/prometheus-rpm/release/el/$releasever/$basearch
repo_gpgcheck=1
enabled=1
gpgkey=https://packagecloud.io/prometheus-rpm/release/gpgkey
       https://raw.githubusercontent.com/lest/prometheus-rpm/master/RPM-GPG-KEY-prometheus-rpm
gpgcheck=1
metadata_expire=300
```

Download keys for repo, they will be needed by clients.

```
mkdir ${dir}/keys/prometheus/
wget https://packagecloud.io/prometheus-rpm/release/gpgkey -P ${dir}/keys/prometheus/
wget https://raw.githubusercontent.com/lest/prometheus-rpm/master/RPM-GPG-KEY-prometheus-rpm -
P ${dir}/keys/prometheus/
rpm --import ${dir}/keys/prometheus/*
```

error related to SHA1

```
warning: Signature not supported. Hash algorithm SHA1 not available.
error: /usr/share/nginx/html/repos/keys/prometheus/gpgkey: key 1 import failed.
```

Check imported keys

```
rpm -q --queryformat "%{SUMMARY}\n" $(rpm -q gpg-pubkey)
```

Clean cache and check enabled repositories

```
dnf clean all
dnf repolist
```

Sync repos to the local storage

Allow some time for initial sync, it will take time (hours or overnight, depending on Internet connection). Open `tmux` and use it.

```
tmux
ls -la ${dir}
df -h ${dir}
```

[Shortcut] As we have distro installation media, we may use it to perform initial sync locally of `baseos` and `appstream` repostitories (around 10GB). Using freshly downloaded ISO image will save you some time. Also this option can be considered, when there is no Internet connectivity at all and these basic repositories need to be distributed further within infrastructure. Or, there is no time and depended application need to be deployed quickly. After long-time sync, OS can 'catch on' updates and repositories availability will not be a stopper.

Potentially this method can be used to maintain air-gapped isolated environments. Yes, updates will not be as often as desired, will be dependent on appearance of distribution images, but fully workable scenario. Attach images to VM, mount, sync, unmount. Yes, will involves manual participation.

Mount media (OracleLinux-R9-U5-x86_64-dvd.iso) to VM via hypervisor dashboard.

```
lsblk
mkdir -p /mnt/rom
mount /dev/sr0 /mnt/rom/
ls -la /mnt/rom/
```

Modify sources for repositories

```
vi /etc/yum.repos.d/oracle-linux-ol9.repo
```

```
[ol9_baseos_latest]
name=Oracle Linux 9 BaseOS Latest ($basearch)
# baseurl=https://yum$ociregion.$ocidomain/repo/OracleLinux/OL9/baseos/latest/$basearch/
baseurl=file:///mnt/rom/BaseOS/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
```

```
gpgcheck=1
enabled=1

[ol9_appstream]
name=Oracle Linux 9 Application Stream Packages ($basearch)
# baseurl=https://yum$ociregion.$ocidomain/repo/OracleLinux/OL9/appstream/$basearch/
baseurl=file:///mnt/rom/AppStream/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1
```

Clone repos from image to VM

```
dnf reposync -g --delete -p ${dir} --repoid=ol9_appstream     --newest-only --download-
metadata
dnf reposync -g --delete -p ${dir} --repoid=ol9_baseos_latest --newest-only --download-
metadata
```

Shortcut: at this point, if you are rushing, manually create a repo and continue dependent deployments. The rest can be scheduled for a night:

```
createrepo ${dir}.
```

Unmount image, revert to the online repositories and sync updates when time permits.

```
umount /mnt/rom
vi /etc/yum.repos.d/oracle-linux-ol9.repo
```

```
[ol9_baseos_latest]
name=Oracle Linux 9 BaseOS Latest ($basearch)
baseurl=https://yum$ociregion.$ocidomain/repo/OracleLinux/OL9/baseos/latest/$basearch/
# baseurl=file:///mnt/rom/BaseOS/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1

[ol9_appstream]
name=Oracle Linux 9 Application Stream Packages ($basearch)
baseurl=https://yum$ociregion.$ocidomain/repo/OracleLinux/OL9/appstream/$basearch/
# baseurl=file:///mnt/rom/AppStream/
```

```
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1
```

Manual sync, but automated recommended (below)

Ensure, you are running sync within `tmux` session, it may take long time. Ensure you have a stable Internet connectivity. Reserve a night for syncing.

```
dnf reposync -g --delete -p ${dir} --repoid=ol9_appstream     --newest-only --download-metadata
dnf reposync -g --delete -p ${dir} --repoid=ol9_baseos_latest --newest-only --download-metadata
createrepo ${dir}
```

Automatic sync of all repositories

To automate process, create script and schedule it on regular basis to update with crontab.

```
sudo su

# cd /usr/local/bin/
cd /root/
vi ./dnf-repos-update.sh
```

modify your repositories' location setting 'dir' variable

```
#!/bin/bash

# Purpose of this script is to perform the synchronization of enabled repositories.
# Inspired by an idea of dedicated repository server within infrastructure.
```

```
#
# Author: Anton TETERIN, Code@InstallAndUse.com
#
# History:
# 2025-02-14  + Init for Oracle Linux v9.5.
#
#
# dir="/mnt/$(hostname)-data/data/www/repos/"
dir="/usr/share/nginx/html/repos"

date
repos=($(dnf repolist | cut -d ' ' -f1))
for repo in "${repos[@]}" ; do
    echo "---[ Performing repo sync for '${repo}'... ]---"
    date
    dnf reposync -g --delete -p ${dir} --repoid=${repo} --newest-only --download-metadata
done

#  If metadata already exists in the output directory, process will be faster.
echo "---[ Creating/updating repo... ]---"
date
createrepo ${dir} --update

echo "---[ Script finished. ]---"
date
```

```
chmod +x ./dnf-repos-update.sh
```

Execute a script or manually do `createrepo ${dir}`.

```
./dnf-repos-update.sh
```

Crontab:

```
sudo su
crontab -l
crontab -e
```

Set it to update at night (i.e. 1st minute on each 3-rd hour = daily at 03h01min)

```
1 3 * * * /root/dnf-repos-update.sh
```

After creating or modifying the cron, I copy-paste the full command in the terminal and manually run it to test for syntax errors. Remember, path MUST be absolute in the cron's configs.

```
crontab -l
```

```
1 3 * * * /root/dnf-repos-update.sh
```

To give an idea of storage usage to estimate download time for coffee and lunch breaks:

```
[root@lt58ncp1sat1 ~]# date
Fri Feb 14 06:30:05 PM EET 2025

[root@lt58ncp1sat1 ~]# du -h ${dir} --max-depth=1
875M    /mnt/lt58ncp1sat1-data/data/www/repos/mariadb-main
73M     /mnt/lt58ncp1sat1-data/data/www/repos/mariadb-maxscale
8.4M    /mnt/lt58ncp1sat1-data/data/www/repos/mariadb-tools
755M    /mnt/lt58ncp1sat1-data/data/www/repos/ol9_UEKR7
3.9G    /mnt/lt58ncp1sat1-data/data/www/repos/ol9_baseos_latest
26G     /mnt/lt58ncp1sat1-data/data/www/repos/ol9_appstream
19G     /mnt/lt58ncp1sat1-data/data/www/repos/ol9_codeready_builder
341M    /mnt/lt58ncp1sat1-data/data/www/repos/s9s-repo
2.4M    /mnt/lt58ncp1sat1-data/data/www/repos/s9s-tools
45G     /mnt/lt58ncp1sat1-data/data/www/repos/ol9_developer_EPEL
95G     /mnt/lt58ncp1sat1-data/data/www/repos/

[root@lt58ncp1sat1 ~]# df -h ${dir}
Filesystem                                              Size  Used Avail Use% Mounted on
/dev/mapper/vg--lt58ncp1sat1--data-lv--lt58ncp1sat1--data  150G   96G   54G  65%
/mnt/lt58ncp1sat1-data
```

Reserve a day for configuration and initial sync, begin to configure afternoon and leave a sync running over night. In case of instable Internet connectivity, use 'always true loop' around the script.

```
while true; do /root/dnf-repos-update.sh; done
```

After some time, I would like to demonsrate again the update process

Before update:

```
[root@lt58ncp1sat1 ~]# date
Sun Mar 16 11:58:42 AM EET 2025
[root@lt58ncp1sat1 ~]# export dir="/mnt/$(hostname)-data/data/www/repos/"
[root@lt58ncp1sat1 ~]# du -h ${dir} --max-depth=1
875M    /mnt/lt58ncp1sat1-data/data/www/repos/mariadb-main
73M     /mnt/lt58ncp1sat1-data/data/www/repos/mariadb-maxscale
8.4M    /mnt/lt58ncp1sat1-data/data/www/repos/mariadb-tools
755M    /mnt/lt58ncp1sat1-data/data/www/repos/ol9_UEKR7
3.9G    /mnt/lt58ncp1sat1-data/data/www/repos/ol9_baseos_latest
26G     /mnt/lt58ncp1sat1-data/data/www/repos/ol9_appstream
19G     /mnt/lt58ncp1sat1-data/data/www/repos/ol9_codeready_builder
341M    /mnt/lt58ncp1sat1-data/data/www/repos/s9s-repo
2.4M    /mnt/lt58ncp1sat1-data/data/www/repos/s9s-tools
45G     /mnt/lt58ncp1sat1-data/data/www/repos/ol9_developer_EPEL
32K     /mnt/lt58ncp1sat1-data/data/www/repos/keys
100M    /mnt/lt58ncp1sat1-data/data/www/repos/repodata
95G     /mnt/lt58ncp1sat1-data/data/www/repos/
[root@lt58ncp1sat1 ~]# df -h ${dir}
Filesystem                                              Size  Used Avail Use% Mounted on
/dev/mapper/vg--lt58ncp1sat1--data-lv--lt58ncp1sat1--data  150G   96G   54G  65%
/mnt/lt58ncp1sat1-data
```

And after update

```
[root@lt58ncp1sat1 ~]# date
Sun Mar 16 01:01:56 PM EET 2025
[root@lt58ncp1sat1 ~]# export dir="/mnt/$(hostname)-data/data/www/repos/"
[root@lt58ncp1sat1 ~]# du -h ${dir} --max-depth=1
875M    /mnt/lt58ncp1sat1-data/data/www/repos/mariadb-main
73M     /mnt/lt58ncp1sat1-data/data/www/repos/mariadb-maxscale
8.4M    /mnt/lt58ncp1sat1-data/data/www/repos/mariadb-tools
760M    /mnt/lt58ncp1sat1-data/data/www/repos/ol9_UEKR7
3.9G    /mnt/lt58ncp1sat1-data/data/www/repos/ol9_baseos_latest
26G     /mnt/lt58ncp1sat1-data/data/www/repos/ol9_appstream
20G     /mnt/lt58ncp1sat1-data/data/www/repos/ol9_codeready_builder
345M    /mnt/lt58ncp1sat1-data/data/www/repos/s9s-repo
2.4M    /mnt/lt58ncp1sat1-data/data/www/repos/s9s-tools
45G     /mnt/lt58ncp1sat1-data/data/www/repos/ol9_developer_EPEL
32K     /mnt/lt58ncp1sat1-data/data/www/repos/keys
100M    /mnt/lt58ncp1sat1-data/data/www/repos/repodata
```

```
96G     /mnt/lt58ncp1sat1-data/data/www/repos/
[root@lt58ncp1sat1 ~]#  df -h ${dir}
Filesystem                                                Size  Used Avail Use% Mounted on
/dev/mapper/vg--lt58ncp1sat1--data-lv--lt58ncp1sat1--data  150G   97G   54G  65%
/mnt/lt58ncp1sat1-data
```

Some time later...

```
[root@lt58sat1 anton]# export dir="/mnt/lt58ncp1sat1-data/data/www/repos/"
[root@lt58sat1 anton]# date
Fri Jul 18 06:36:56 PM EEST 2025
[root@lt58sat1 anton]# du -h ${dir} --max-depth=1
968M    /mnt/lt58ncp1sat1-data/data/www/repos/mariadb-main
78M     /mnt/lt58ncp1sat1-data/data/www/repos/mariadb-maxscale
8.4M    /mnt/lt58ncp1sat1-data/data/www/repos/mariadb-tools
801M    /mnt/lt58ncp1sat1-data/data/www/repos/ol9_UEKR7
4.1G    /mnt/lt58ncp1sat1-data/data/www/repos/ol9_baseos_latest
27G     /mnt/lt58ncp1sat1-data/data/www/repos/ol9_appstream
20G     /mnt/lt58ncp1sat1-data/data/www/repos/ol9_codeready_builder
363M    /mnt/lt58ncp1sat1-data/data/www/repos/s9s-repo
2.5M    /mnt/lt58ncp1sat1-data/data/www/repos/s9s-tools
47G     /mnt/lt58ncp1sat1-data/data/www/repos/ol9_developer_EPEL
32K     /mnt/lt58ncp1sat1-data/data/www/repos/keys
103M    /mnt/lt58ncp1sat1-data/data/www/repos/repodata
99G     /mnt/lt58ncp1sat1-data/data/www/repos/
[root@lt58sat1 anton]# df -h ${dir}
Filesystem                                                Size  Used Avail Use% Mounted on
/dev/mapper/vg--lt58ncp1sat1--data-lv--lt58ncp1sat1--data  150G  100G   51G  67% /mnt/lt58ncp1sat1-data
```

Client configuration

Remove **all** old repos! Do not leave leftovers.

```
sudo su
ls -la /etc/yum.repos.d/
rm /etc/yum.repos.d/*
```

Configure DNS records locally for repository server, if not present in the zone.

```
vi /etc/hosts
```

```
192.168.56.109  lt58ncp1sat1
```

Check and modify local name resolution, if needed. Check.

```
ping lt58ncp1sat1
curl http://lt58ncp1sat1/hello
```

Edit client's repo file.

```
vi /etc/yum.repos.d/lt58ncp1sat1.repo
```

```
[ol9_baseos_latest]
name=Oracle Linux 9 BaseOS Latest
baseurl=http://lt58ncp1sat1/ol9_baseos_latest/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle

[ol9_appstream]
name=Oracle Linux 9 Application Stream Packages
baseurl=http://lt58ncp1sat1/ol9_appstream/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle

[ol9_codeready_builder]
name=Oracle Linux 9 CodeReady Builder (x86_64) - (Unsupported)
baseurl=http://lt58ncp1sat1/ol9_codeready_builder/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle

[ol9_UEKR7]
name=Oracle Linux 9 UEK Release 7 (x86_64)
baseurl=http://lt58ncp1sat1/ol9_UEKR7/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle

[ol9_developer_EPEL]
name=Oracle Linux 9 EPEL Packages for Development
baseurl=http://lt58ncp1sat1/ol9_developer_EPEL/
```

```
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle


[mariadb-main]
name=MariaDB Server
baseurl=http://lt58ncp1sat1/mariadb-main/
enabled=1
gpgcheck=1
gpgkey=http://lt58ncp1sat1/keys/mariadb/MariaDB-Server-GPG-KEY

[mariadb-maxscale]
name=MariaDB MaxScale
baseurl=http://lt58ncp1sat1/mariadb-maxscale/
enabled=1
gpgcheck=1
gpgkey=http://lt58ncp1sat1/keys/mariadb/MariaDB-MaxScale-GPG-KEY

[mariadb-tools]
name=MariaDB Tools
baseurl=http://lt58ncp1sat1/mariadb-tools/
enabled=1
gpgcheck=1
gpgkey=http://lt58ncp1sat1/keys/mariadb/MariaDB-Enterprise-GPG-KEY


[s9s-repo]
name = Severalnines Release Repository
baseurl=http://lt58ncp1sat1/s9s-repo/
enabled=1
gpgcheck=1
gpgkey=http://lt58ncp1sat1/keys/s9s-repo/severalnines-repos.asc

[s9s-tools]
name = Severalnines Tools Repository
baseurl=http://lt58ncp1sat1/s9s-tools/
enabled=1
gpgcheck=1
gpgkey=http://lt58ncp1sat1/keys/s9s-tools/repomd.xml.key
```

```
[prometheus]
name=prometheus
baseurl=http://lt58ncp1sat1/prometheus/
enabled=1
gpgkey=http://lt58ncp1sat1/keys/prometheus/gpgkey
       http://lt58ncp1sat1/keys/prometheus/RPM-GPG-KEY-prometheus-rpm
gpgcheck=1
```

Check configured repos

```
dnf clean all
dnf repolist
```

Check availability of packages and list installed ones

```
dnf list --installed
```

While fetching metadata, updating, downloading packages, you may observe traffic from satellite:

```
sudo su
tcpdump -n port 80
```

Update OS

```
[root@lt58ncp1dbn3 anton]# dnf update
Last metadata expiration check: 0:06:10 ago on Fri 14 Feb 2025 09:00:17 PM EET.
Dependencies resolved.
Nothing to do.
Complete!
```

Happy admin :-)

After cloning

After the clone, there is a need to make machine 'unique':

Ensure, it is a new machine, it's uptime should be low.

```
uptime
```

Check network interfaces

```
ip -br a
```

Set new hostname

```
sudo su
hostnamectl hostname (new_hostname)
```

Remove current server's SSH keys and generate new ones to avoid confusion on connecting

```
rm -rf /etc/ssh/ssh_host_*
ssh-keygen -A
```

Reboot the machine to ensure it will apply it's hostname in depended services locally and in the network

```
shutdown -r now
```

Possible situations:

You may need to remove old SSH keys, to perform it:

```
$ ssh-keygen -R 10.77.84.133
```

```
# Host 10.77.84.133 found: line 27
/home/anton/.ssh/known_hosts updated.
Original contents retained as /home/anton/.ssh/known_hosts.old
```

Password change for OL9

Reboot in GRUB:

add to end of `linux` line, and initiate boot with CTRL-X

```
rd.break
```

Mount

```
mount -o remount,rw /sysroot
```

Chroot environment

```
chroot /sysroot
passwd root
```

Reboot

```
exit
exit
exit
```