# How to connect server to satellite server?

- subscription-manager status
- yum repolist
- cd /etc/yum.repos.d/
- rpm -qa | grep katello-ca-consumer
- subscription-manager remove --all
- subscription-manager unregister
- subscription-manager clean
- rpm -Uvh http://10.21.3.1/pub/katello-ca-consumer-jptrcolxqasrv01.bmotfll.com-1.0-1.noarch.rpm
- subscription-manager register --org="BMO_FRDC_PureApp" --activationkey="1-COL_PureApp"
- yum repolist
- subscription-manager status

# PATCHING PROCESS

1. Collect prechecks from prechecks.sh script.

2. Take snap shots for all VM's.

3. Suppress the alerts as per change schedule.

Run the below commands as a root user

## Pre-Patch:

1. service --status-all > pre-patch-services #RHEL 6

1. systemctl list-units --type service |grep running > pre-patch-services #RHEL 7

2. rpm -qa |sort > oldrpm_`date +"%m%d%Y"`.txt

## PATCHING:

3. yum -y update --security |tee yum_updates_`date +"%m%d%Y"`.txt

4. yum history

4. who

5. reboot

## Post-Patch:

6. uptime; uname -a

7. service --status-all > post-patch-services && diff pre-patch-services post-patch-services |sort #RHEL 6

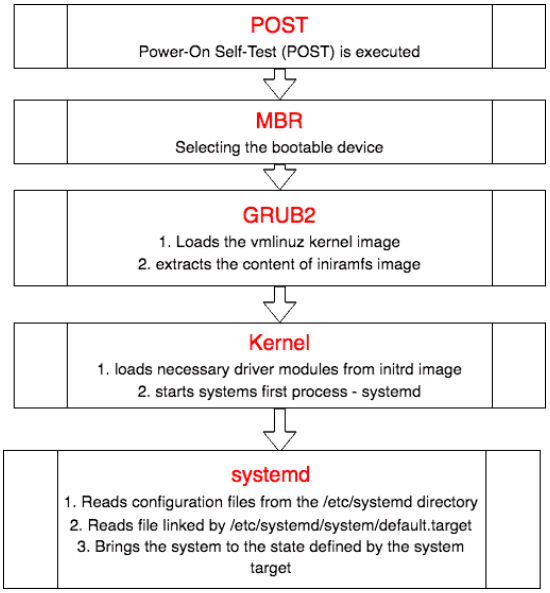
7. systemctl list-units --type service |grep running > post-patch-services && diff pre-patch-services post-patch-services |sort #RHEL 7

# BOOTING PROCESS

The following steps summarize how the boot procedure happens in RHEL/CentOS 7.

1. The computer's BIOS performs POST.
2. BIOS reads the MBR for the bootloader.
3. GRUB 2 bootloader loads the vmlinuz kernel image.
4. GRUB 2 extracts the contents of the initramfs image.
5. The kernel loads driver modules from initramfs.
6. Kernel starts the system's first process, systemd.
7. The systemd process takes over. It:

    - Reads configuration files from the /etc/systemd directory
    - Reads file linked by /etc/systemd/system/default.target
    - Brings the system to the state defined by the system target
    - Executes /etc/rc.local




## 1. POST (Power on Self Test)

From the system firmware, which can be the modern Universal Extended Firmware Interface (**UEFI**) or the classical Basic Input Output System (**BIOS**), the Power-On Self-Test (POST) is executed, and the hardware that is required to start the system is initialized.

## 2. Selecting the bootable device (With MBR)

– Master Boot Record (MBR) is the first 512 bytes of the boot drive that is read into memory by the BIOS.

– The next 64 bytes contain the partition table for the disk. The last two bytes are the “Magic Number” which is used for error detection.


– MBR discovers the bootable device and loads the GRUB2 boot loader into memory and transfers control over to it.

## 3. Loading the boot loader (GRUB2)

– The default bootloader program used on RHEL 7 is GRUB 2. GRUB stands for **GRand Unified Bootloader**. GRUB 2 replaces the older GRUB bootloader also called as legacy GRUB.
– The GRUB 2 configuration file is located at /boot/grub2/grub.cfg (Do not edit this file directly).

```
### BEGIN /etc/grub.d/10_linux ###
menuentry 'CentOS Linux (3.10.0-693.21.1.el7.x86_64) 7 (Core)' --class centos --class gnu-li
nux --class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-3.10.0-123.8.1.el7.
x86_64-advanced-0f790447-ebef-4ca0-b229-d0aa1985d57f' {
        load_video
        set gfxpayload=keep
        insmod gzio
        insmod part_msdos
        insmod xfs
        set root='hd0,msdos1'
        if [ x$feature_platform_search_hint = xy ]; then
          search --no-floppy --fs-uuid --set=root --hint='hd0,msdos1'  0f790447-ebef-4ca0-b2
29-d0aa1985d57f
        else
          search --no-floppy --fs-uuid --set=root 0f790447-ebef-4ca0-b229-d0aa1985d57f
        fi
        linux16 /boot/vmlinuz-3.10.0-693.21.1.el7.x86_64 root=UUID=0f790447-ebef-4ca0-b229-d
0aa1985d57f ro console=ttyS0,115200 console=tty0 console=ttyS0,115200n8 vconsole.font=latarc
yrheb-sun16 crashkernel=auto  vconsole.keymap=us LANG=en_US.UTF-8
        initrd16 /boot/initramfs-3.10.0-693.21.1.el7.x86_64.img
}
```

– GRUB 2 menu-configuration settings are taken from /etc/default/grub when generating grub.cfg.
– Sample /etc/default/grub file :

```
# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel/swap crashkernel=auto rd.lvm.lv=rhel/root rhgb quiet
net.ifnames=0"
GRUB_DISABLE_RECOVERY="true"
```

– If changes are made to any of these parameters, you need to run **grub2-mkconfig** to re-generate the /boot/grub2/grub.cfg file.

```
# grub2-mkconfig –o /boot/grub2/grub.cfg
```

– GRUB2 searches the compressed kernel image file also called as vmlinuz in the /boot directory.
– GRUB2 loads the vmlinuz kernel image file into memory and extracts the contents of the initramfs image file into a temporary, memory-based file system (tmpfs).
– The initial RAM disk (initrd) is an initial root file system that is mounted before the real root file system.

**initramfs**
– The job of the initial RAM file system is to preload the block device modules, such as for IDE, SCSI, or RAID, so that the root file system, on which those modules normally reside, can then be accessed and mounted.
– The initramfs is bound to the kernel and the kernel mounts this initramfs as part of a two-stage boot process.
– The dracut utility creates initramfs whenever a new kernel is installed.
– Use the lsinitrd command to view the contents of the image created by dracut:

```
# lsinitrd | less
```

## 4. Loading the kernel

– The kernel starts the systemd process with a process ID of 1 (PID 1).

```
[root@geeklab ~]# ps -ef | grep systemd
root          1     0  0 10:31 ?        00:00:02 /usr/lib/systemd/systemd --switched-root --system --deserialize 21
```

– It also loads the necessary driver modules from initrd image.
– The boot loader (GRUB2) may present a boot menu to the user, or can be configured to automatically start a default operating system.
– To load Linux, the kernel is loaded together with the initramfs. The initramfs contains kernel modules for all hardware that is required to boot, as well as the initial scripts required to proceed to the next stage of booting.
– On RHEL 7, the initramfs contains a complete operational system (which may be used for troubleshooting purposes).

## 5. Starting systemd

– The kernel starts the systemd process with a process ID of 1 (PID 1).

```
root          1     0  0 02:10 ?        00:00:02 /usr/lib/systemd/systemd --switched-root --system --deserialize
23
```

– systemd is the first process that starts after the system boots, and is the final process that is running when the system shuts down.
– It controls the final stages of booting and prepares the system for use. It also speeds up booting by loading services concurrently.
– systemd reads the file linked by /etc/systemd/system/default.target (for example, /usr/lib/systemd/system/multi-user.target) to determine the default system target (equivalent to run level). The system target file defines the services that systemd starts.
– systemd allows you to manage various types of units on a system, including services (name.service) and targets (name.target), devices (name.device), file system mount points (name.mount), and sockets (name.socket).

**Comparision of SysV Run Levels and Target Units**

| Run Level | Target Units | Description |
|---|---|---|
| 0 | runlevel0.target, poweroff.target | Shut down and power off |
| 1 | runlevel1.target, rescue.target | Set up a rescue shell |
| 2,3,4 | runlevel[234].target, multi- user.target | Set up a nongraphical multi-user shell |
| 5 | runlevel5.target, graphical.target | Set up a graphical multi-user shell |
| 6 | runlevel6.target, reboot.target | Shut down and reboot the system |

systemd brings the system to the state defined by the system target, performing system initialization tasks such as:
1. Setting the host name
2. Initializing the network
3. Initializing SELinux based on its configuration
4. Printing a welcome banner
5. Initializing the system hardware based on kernel boot arguments
6. Mounting the file systems, including virtual file systems such as the /proc file system
7. Cleaning up directories in /var
8. Starting swapping

## View default/current target unit

Use the following command to view which target unit is used by default:

```
# systemctl get-default
graphical.target
```

The **graphical.target** target unit indicates that the system is running in a graphical, multi- user state. This is similar to run level 5 in a SysV init system. You can verify this using the old command **runlevel** :

```
# runlevel
N 5
```

The default target unit is represented by the **/etc/systemd/system/default.target** file. This file is a symbolic link to the current default target unit. For example :

```
# ls -lrt /etc/systemd/system/default.target
lrwxrwxrwx. 1 root root 36 Sep 23 20:01 /etc/systemd/system/default.target ->
/lib/systemd/system/graphical.target
```

## Change default target unit

Use the following command to change the default target unit (for example, to change the default to the multi-user.target unit):

```
# systemctl set-default multi-user.target
Removed symlink /etc/systemd/system/default.target.
Created symlink from /etc/systemd/system/default.target to /usr/lib/systemd/system/multi-user.target.
```

Notice that the default.target symbolic link has changed, and is now pointing to the multi-user.target unit:

```
# ls -lrt /etc/systemd/system/default.target
lrwxrwxrwx. 1 root root 41 Sep 24 11:58 /etc/systemd/system/default.target ->
/usr/lib/systemd/system/multi-user.target
```

# RHEL 6 vs RHEL 7 Difference

| Feature Name | RHEL 6 | RHEL 7 |
|---|---|---|
| **Default File System** | Ext4 | XFS |
| **Kernel Version** | 2.6.xx | 3.10.xx |
| **Release Name** | Santiago | Maipo |
| **Gnome Version** | GNOME 2 | GNOME 3.8 |
| **KDE Version** | KDE 4.1 | KDE 4.6 |
| **Release Date** | Wednesday, November 10, 2010 | Tuesday, June 10, 2014 |
| **NFS Version** | NFS 4 | NFS 4.1. NFS V2 is deprecated in RHEL 7 |
| **Samba Version** | SMB 3.6 | SMB 4.4 |
| **Default Database** | MySQL | MariaDB |
| **Cluster Resource Manager** | Rgmanager | Pacemaker |
| **Network Interface Grouping** | Bonding can be done as Active-Backup, XOR, IEEE and Load Balancing | Team Driver will support multiple types of Teaming methods called Active-Backup, Load-balancing and Broadcast |
| **KDUMP** | Kdump does't support with large RAM Size | RHEL 7 can be supported up to 3TB |
| **Boot Loader** | Grub 0.97<br>/boot/grub/grub.cfg | Grub 2<br>/boot/grub2/grub.conf |
| File System Check | e2fsck<br>-Inode check. Block and size check<br>–Directory Structure check<br>-Directory Link Check<br>-reference count check<br>-Group Summary Check | xfs_replair<br>– Inode blockmap checks<br>-node allocation map checks<br>-Inode size check<br>-Directory check<br>-Path Name check<br>-Link count check |

| | | |
|---|---|---|
| | | -Freemap check<br>-Super block check |
| **Process ID** | Initd Process ID 1 | Systemd Process ID 1 |
| **Port Security** | Iptables by default service port is enabled when service is switched on. | Firewalld instead of iptables. Iptables support with RHEL 7, but we can't use both of them at the same time. Firewall blocks ports by default you have to enable explicitly. |
| **Boot Time** | 40 Sec | 20 Sec |
| **File System Size** | EXT4 16TB with XFS 100TB | XFS 500TB with EXT4 16TB |
| **Processor Architecture** | 32Bit and 64Bit | Only 64Bit. |
| **Network Configuration Tool** | setup | nmtui |
| **Host name Config File** | /etc/sysconfig/network | /etc/hostname No need to edit hostname file to write permanent hostname simply use hostnamectl command |
| **Interface Name** | eth0 | ens33xxx |
| Managing Services | service sshd start<br>service sshd restart<br>chkconfig sshd on | systemctl start sshd.service<br>systemctl restart sshd.service<br>systemctl enable sshd.service |
| **System Logs** | /var/log/ | /var/log<br>journalctl |
| **Run Levels** | runlevel0 – Power Off<br>runlevel1 – Single User Mode<br>runlevel2 – Multi User without Networking<br>runlevel3 – Multi User CLI<br>runlevel4 – Not USed<br>runlevel5 – GUI Mode<br>runlevel6 – Restart | There is no run levels in RHEL 7. Run levels are called as targets<br>Poweroff.target<br>rescue.target<br>multi-user.target<br>graphical.target<br>reboot.target |
| **UID Information** | Normal User UID will start from 500 to 65534<br>System Users UID will start from 1 to 499 | Normal User UID start from 1000 – 65534<br>System Users UID will start from 1 to 999 because Services are increased compare to RHEL 6 |

| | | |
|---|---|---|
| **By Pass Root Password Prompt** | append 1 or s or init=/bin/bash to Kernel command line | Append rd.break or init=/bin/bash to kernel command line |
| **Rebooting and Poweroff** | poweroff – init 0<br>reboot – init 6 | systemctl poweroff<br>systemctl reboot |
| **YUM Commands** | yum groupinstall<br>yum groupinfo | yum group install<br>yum group info |

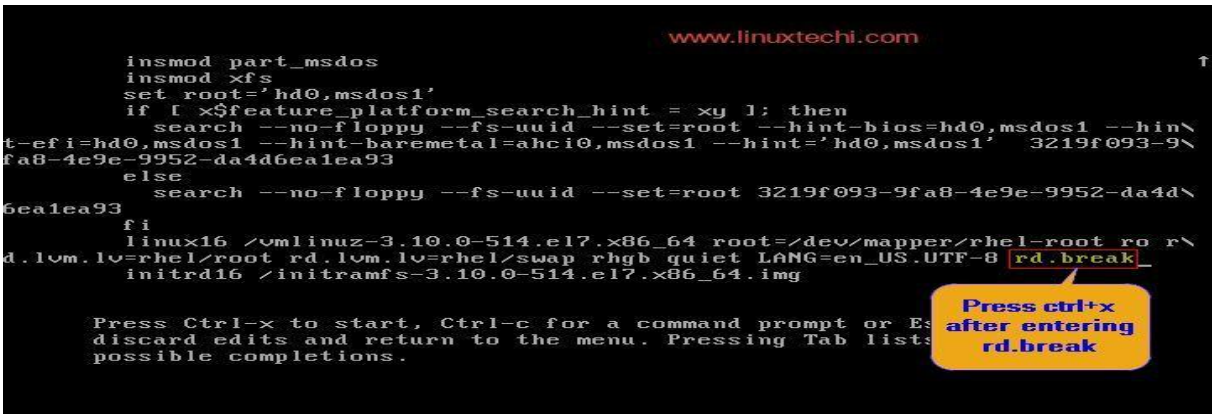
# SINGLE USER MODE

*Method 1:*

**Step1:** Reboot your Server and go to Grub boot loader menu and choose the appropriate kernel, example is shown below:


**Step 2:** Press 'e' and go to the end of line which starts with 'linux16' word.


Type "**rd.break**" at end of line which




begins with linux16 and then press "**ctrl+x**"

In the next window we will get single user mode or emergency mode, something like below:


**Step 3:** Now remount the /sysroot in 'rw' mode

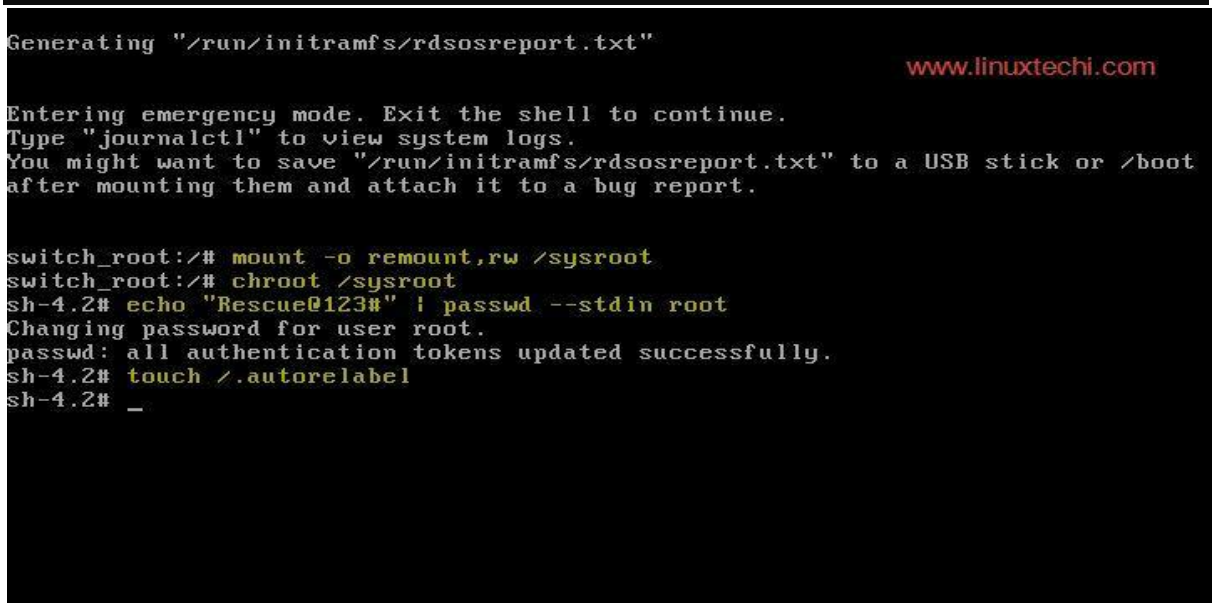
```
switch_root:/# mount -o remount,rw /sysroot

switch_root:/# chroot /sysroot
```

Let's assume we want to reset root password. Run the beneath commands

```
sh-4.2# echo "New-root-password" | passwd --stdin root

sh-4.2# touch /.autorelabel
```



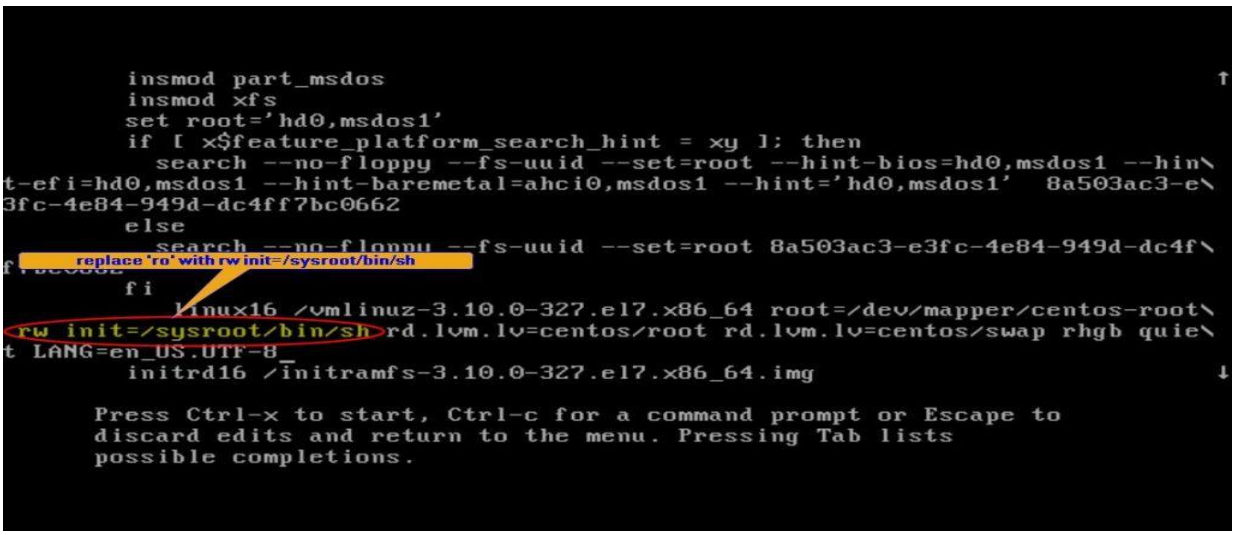

Now you can edit the files and scan corrupted file system with fsck command. Once you are done with troubleshooting Type '**exit**' command two times or '**reboot -f**' to reboot your server.

*Method 2:*

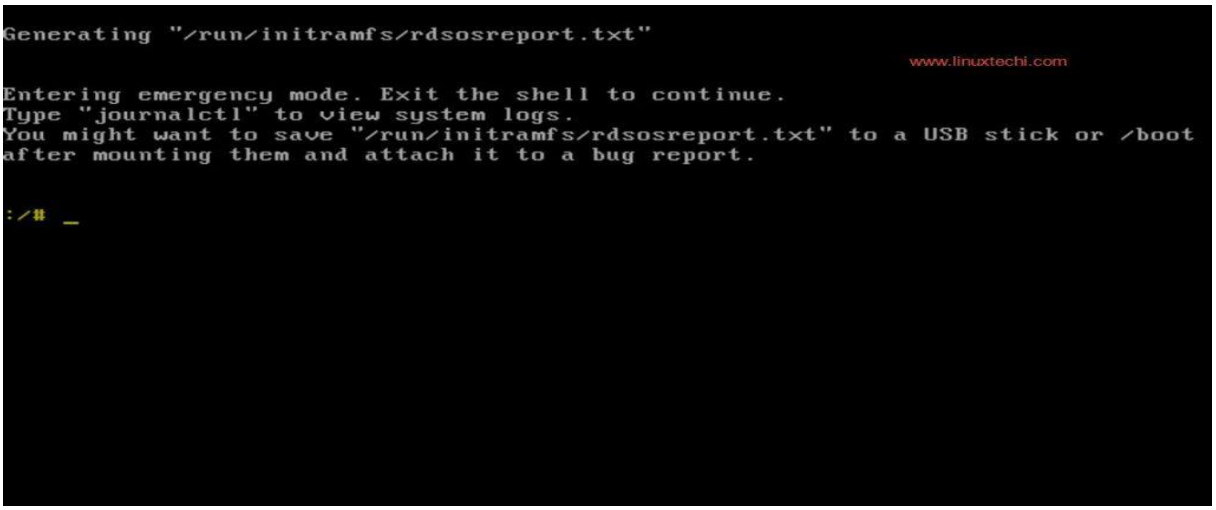
**Step 1:** Reboot the server and go to the grub menu and select the appropriate kernel


**Step 2:** Press 'e' and go the line with starts with 'linux16'

Replace "**ro**" with "**rw init=/sysroot/bin/sh**"




Once done with the changes press 'Ctrl+x'




**Step 3:** Mount the root file system with chroot command

```
:/# chroot /sysroot
```

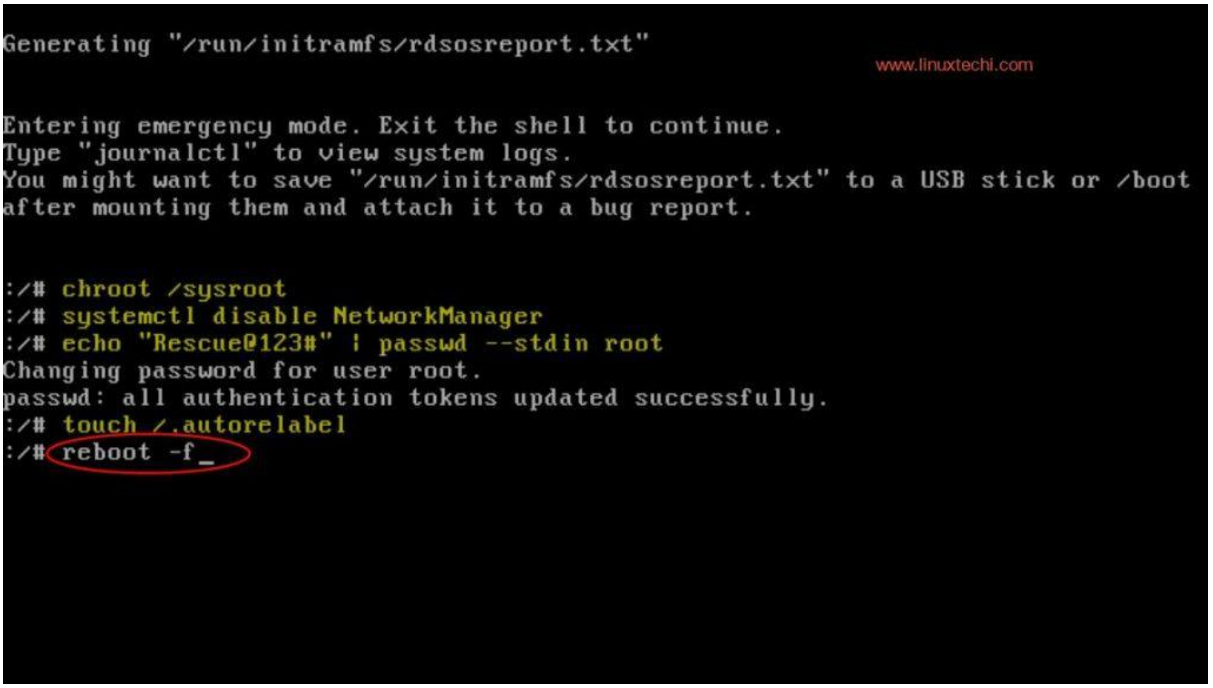
Let's assume I want to disable network manager service and rest root password, run the following commands:

```
:/# systemctl disable NetworkManager

:/# echo "Enter-New-Root-password" | passwd --stdin root

:/# touch /.autorelabel
```

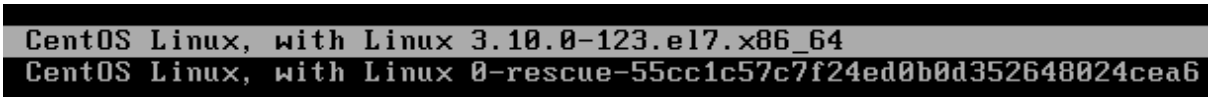
Once you are done with the changes, type "reboot -f" command to reboot your server




**Note:** In production environment, Single user mode is also password protected. By default root password is the single user mode password on CentOS 7 / RHEL 7 Servers.

## Bootup into Emergency mode(target)

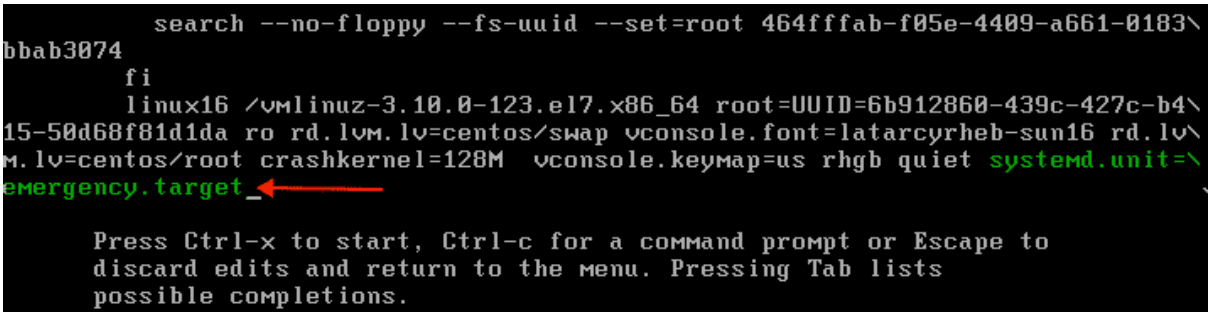
1. During bootup, when the GRUB2 menu shows up, press the e key for edit.



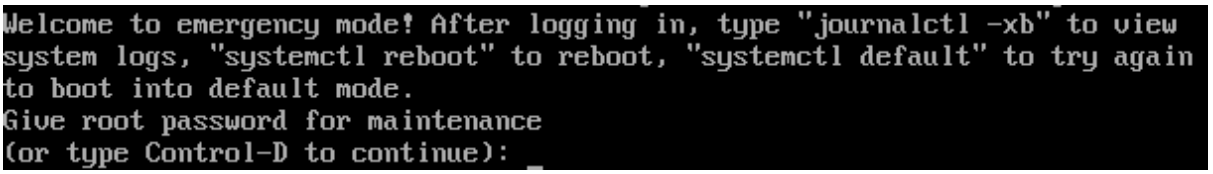

2. Add the following parameter at the end of the linux16 line :

```
systemd.unit=emergency.target
```

Press Ctrl+a (or Home) and Ctrl+e (or End) to jump to the start and end of the line.




3. Press Ctrl+x to boot the system with the parameter.




## Switch to Emergency mode(target)

To switch to Emergency target, simply run following command as root:

```
# systemctl emergency
Broadcast message from root@geeklab on pts/1 (Mon 2016-08-17 00:44:58 EDT):
The system is going down to emergency mode NOW!
```

To prevent systemd from sending informative message:

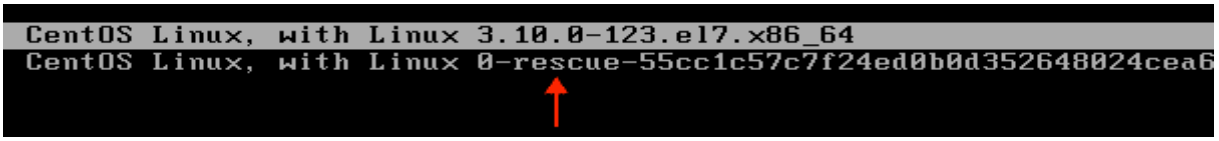
```
# systemctl --no-wall emergency
# systemctl isolate emergency.target
```

## Bootup into Rescue mode(target)

There are 2 ways to get into rescue mode :

**1. Method 1**

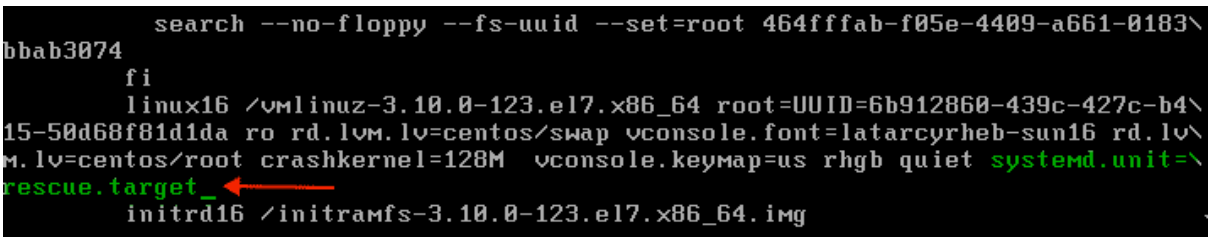
There is a GRUB2 menu option when you boot up the system which can be selected to directly boot into rescue mode.



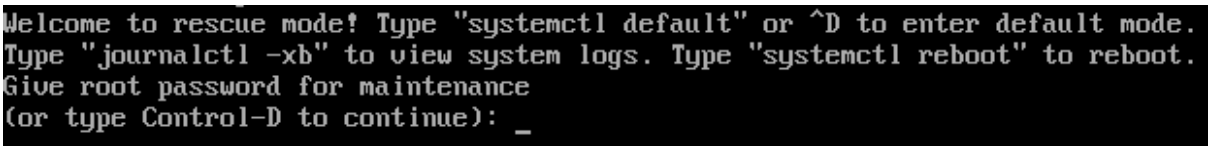

**2. Method 2**
1. During bootup, when the GRUB2 menu shows up, press the e key for edit.

2. Add the following parameter at the end of the linux16 line:

```
systemd.unit=rescue.target
```
Press Ctrl+a (or Home) and Ctrl+e (or End) to jump to the start and end of the line.




3. Press Ctrl+x to boot the system with the parameter.




## Switch to Rescue mode(target)

To switch to rescue target, simply run following command as root:

```
# systemctl rescue
Broadcast message from root@geeklab on pts/0 (Mon 2016-08-17 00:22:44 EDT):
The system is going down to rescue mode NOW!
```
To prevent systemd from sending informative message:

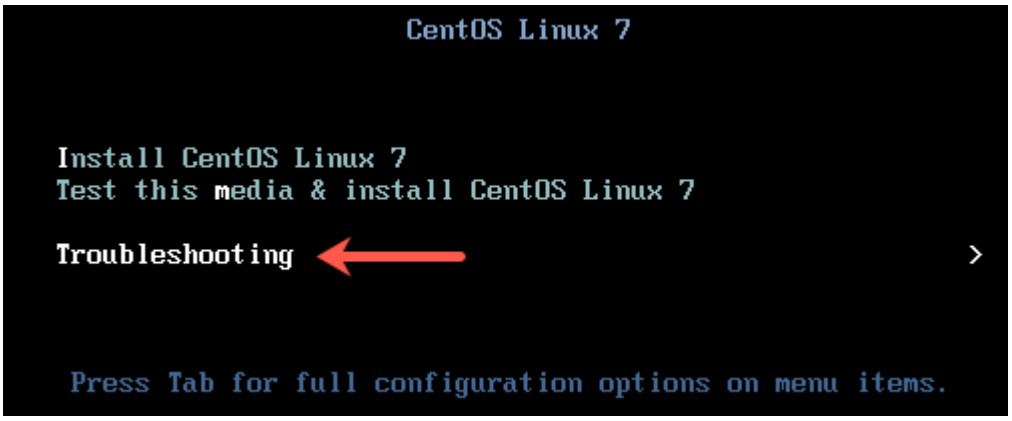
```
# systemctl --no-wall rescue
# systemctl isolate rescue.target
```

## Booting into rescue mode using DVD/ISO

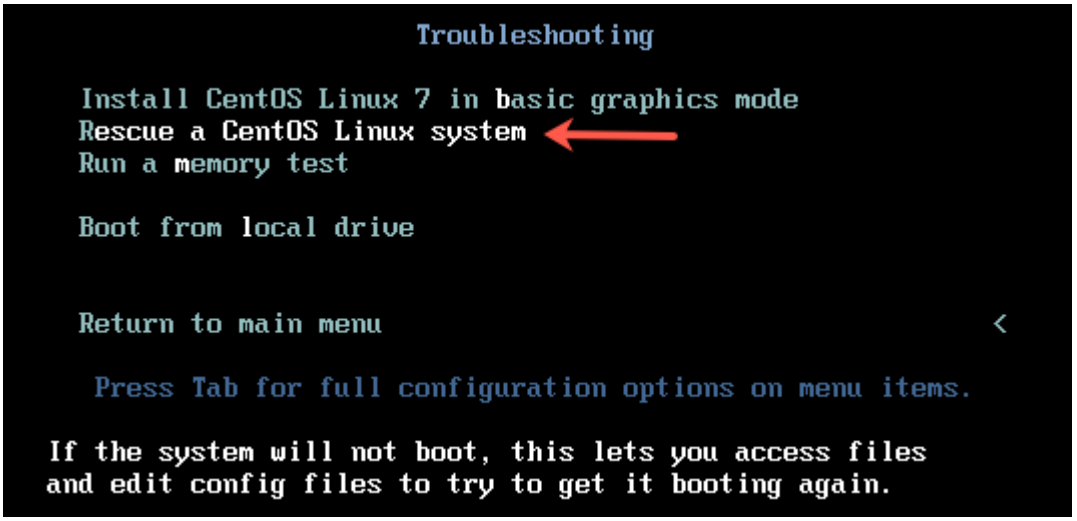
Follow the steps below to boot the system into rescue mode using installation media such as DVD or ISO.

**1. Attach the ISO image**
You can use an actual installation DVD instead of ISO image, but I find using ISO image easy and there is no need to go to the data center to physically insert the DVD into the system. Different virtualization platforms have similar features to attach/mount the ISO image to a VM guest. Make sure you change the boot order to boot from the ISO image.

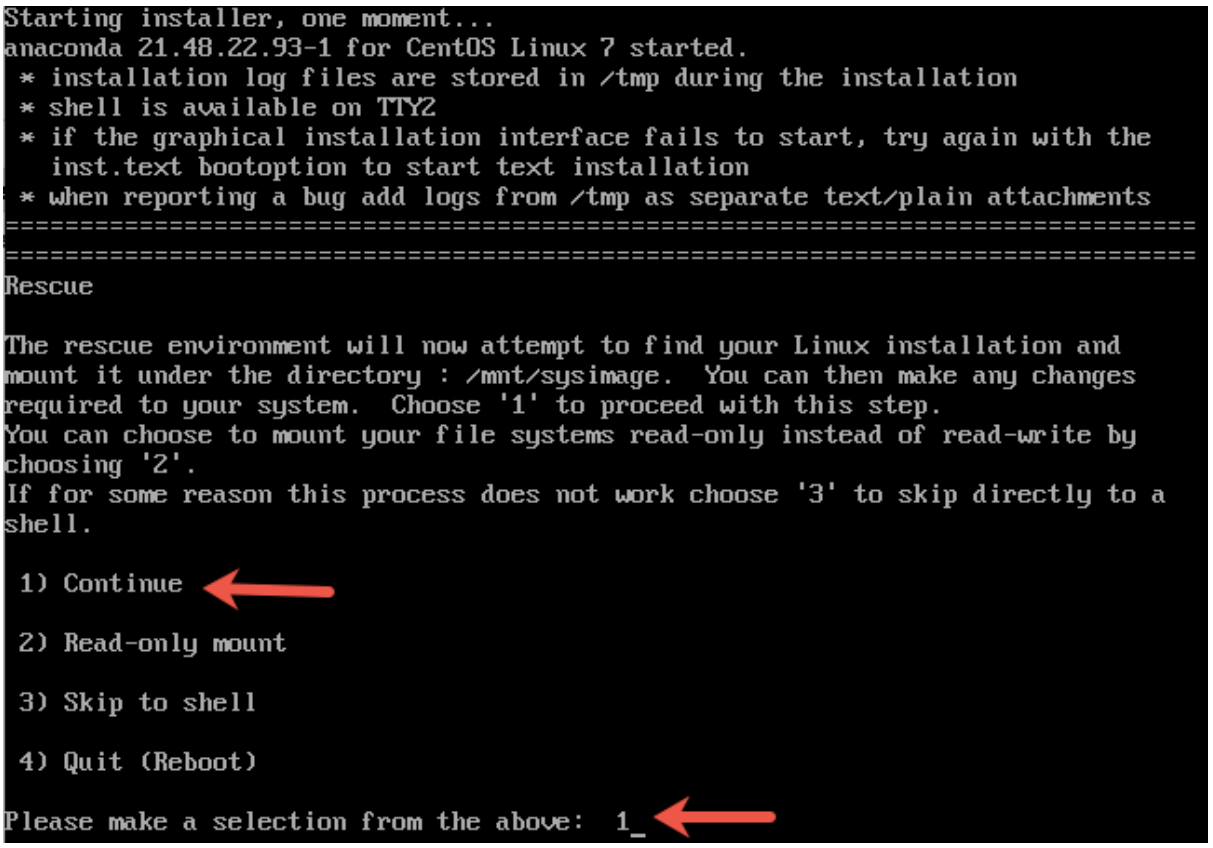
**2. Boot up the system**
Boot up the CentOS 7 system from ISO image. At the boot screen, Select the **Troubleshooting option** at the end of the screen.



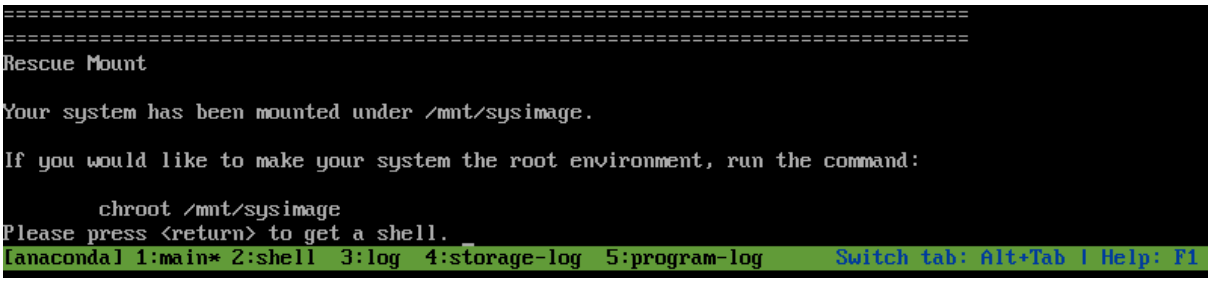

3. At the next screen, select the option **Rescue a CentOS Linux system**.




4. On the next screen, press enter to continue. When asked if you would like Rescue to find your installation, choose Continue.




If you run into trouble detecting your install, retry using the Skip option and manually detect and mount your storage. You would get a message shown in the picture below if the rescue mode has detected the correct installation.

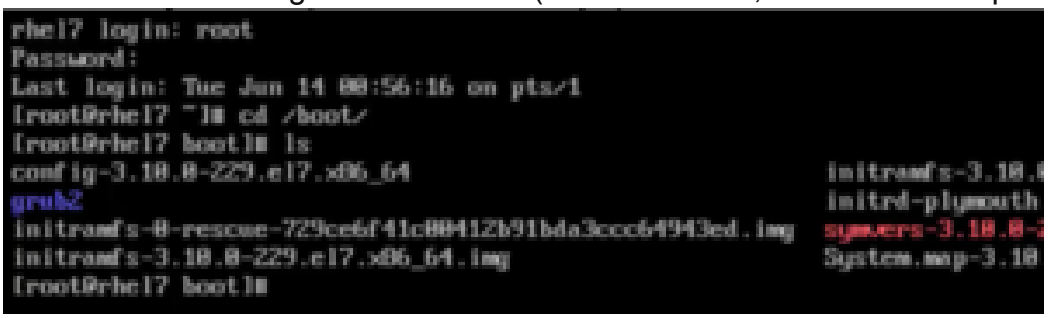



5. Next step is to change your root directory to **/mnt/sysimage** using the chroot command. This makes your system the root environment.

```
# chroot /mnt/sysimage
```

# KERNEL PANIC

This happens mostly because of initramfs.img file missing or corrupted
1.Boot the system in rescue mode
2.Login using the root account
3.Navigate to /boot
4. See if initramfs.img file is available (If it is available, it must be corrupted




5.find kernel version (uname -r)
6.mkinitrd initramfs-kernel_version.img kernel_version (RHEL6)
Or dracut initramfs-kernel_version.img kernel_version (RHEL7)


7.Boot the machine

# Steps to Expand a / filesystem :-

- lsblk
- echo "- - -" > /sys/class/scsi_host/host0/scan
- echo "- - -" > /sys/class/scsi_host/host1/scan
- echo "- - -" > /sys/class/scsi_host/host2/scan
- lsblk
- fdisk -l /dev/sdc
- fdisk /dev/sdc
- fdisk -l /dev/sdc
- pvcreate /dev/sdc1
- vgextend vg_root /dev/sdc1
- lvdisplay
- lvextend -L +4.99G /dev/vg_root/LogVol00
- xfs_growfs /

# Steps to Reduce filesystem size :-

Step:1 Umount the **file system**.
Step:2 check the **file system for** Errors using e2fsck command.

```
    e2fsck -f /dev/mapper/vg_cloud-LogVol00
```

Step:3 **Reduce** or

Step:4 Now **reduce** the **size** using lvreduce command.

```
lvreduce -L 10G /dev/mapper/vg_cloud-LogVol00
```

Step:5 (Optional) **For** the safer side, now check the **reduced file system for** errors.
Step:6 Mount the file system and verify its size