USER STORY DOCUMENTATION-LINUX

NAME : P.SANJANA REDDY

EMP ID:MT4020

Table of contents

	mounted /xfs directory. Each file has got a name as follows: .file_N where N is a number from 1 to 40	
8	Configure a HTTP server, which can be accessed through http://station.domain40.example.com. Please download the released page from http://ip/dir/example.htm	34-36
9	Copy /etc/fstab to /var/tmp name admin, the user1 could read, write and modify it, while user2 without any permission.	36-39
10	Install MongoDB in Linux and run CRUD command, create a cron job and move data from Mongo	40-45

USER STORY 1

Create user /home/modak

DESCRIPTION:

a.change home directory /opt/modak

b. uid and gid for the user modak

c.change username and primary group for the user modak to MODAK

d.uid and gid should be same

Explanation:

Created user /home/modak

- Used useradd to add the user
- Then checked whether user added or not in the /etc/passwd directory.

```
[root@sanjana /]# useradd modak
[root@sanjana /]# cat passwd | frep modak
bash: frep: command not found
cat: passwd: No such file or directory
[root@sanjana /]# cat /etc/passwd | grep modak
modak:x:3003:3004::/home/modak:/bin/bash
[root@sanjana /]# _
```

Changed home directory /opt/modak

- Changed the home directory using usermod
- **Syntax: usermod -d directory user**
- Then checked whether it changed or not.

```
[root@sanjana /]# usermod -d /opt/modak modak
```

```
[root@sanjana /]# cat /etc/passwd | grep modak
modak:x:3003:3004::/home/modak:/bin/bash
[root@sanjana /]# usermod -d /opt/modak modak
[root@sanjana /]# cat /etc/passwd | grep modak
modak:x:3003:3004::/opt/modak:/bin/bash
[root@sanjana /]# _
```

uid and gid for the user modak

- Used id command to get uid and gid of user
- **Syntax: id user_name**

```
modak:x:3003:3004::/opt/modak:/bin/bash
[root@sanjana /]# id modak
uid=3003(modak) gid=3004(modak) groups=3004(modak)
```

Changed username and primary group for the user modak to MODAK

- We can change username and primary group using usermod
- **Syntax:usermod –g groupname username**
- **Syntax:usermod –l newname oldname**

```
[root@sanjana /]# groupadd MODAK
[root@sanjana /]# cat /etc/group | grep MODAK
MODAK:x:3005:
[root@sanjana /]# usermod -g MODAK modak
[root@sanjana /]# usermod -l  MODAK modak
[root@sanjana /]# cat /etc/passwd | grep MODAK
MODAK:x:3003:3005::/opt/modak:/bin/bash
[root@sanjana /]# id MODAK
uid=3003(MODAK) gid=3005(MODAK) groups=3005(MODAK)
[root@sanjana /]#
```

- We can also use groupmod commad to change primary group.
- **Syntax : groupmod –n new_group user_name**

```
[root@sanjana /]# groupmod -n MODAK1 modak1
```

Changed uid and gid of MODAK (uid and gid should be same)

- Changed userid using usermod
- Changed gid using groupmod
- Then checked by using id command

```
[root@sanjana /]# groupmod  -g 1234 MODAK
[root@sanjana /]# usermod -u  1234 MODAK
[root@sanjana /]# id MODAK
uid=1234(MODAK) gid=1234(MODAK) groups=1234(MODAK)
[root@sanjana /]#
```

USER STORY 2

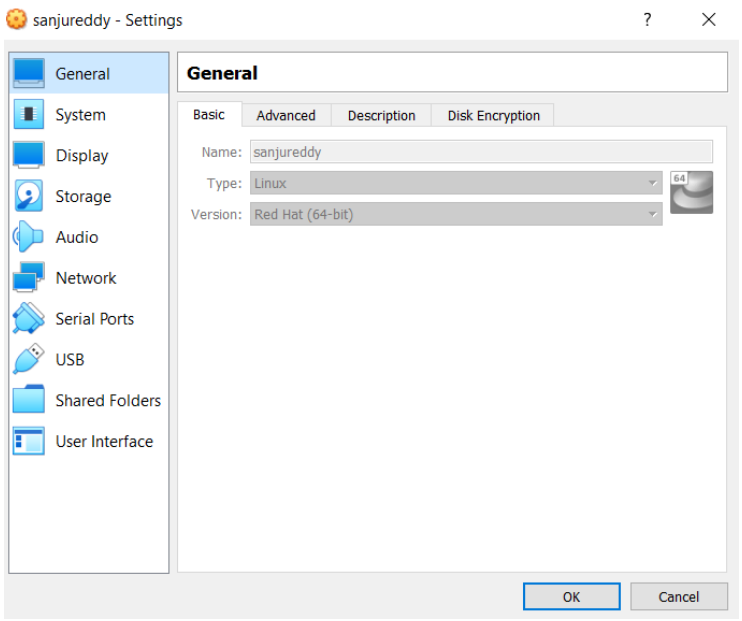
Create a file system and mount, unmount it.

DESCRIPTION:

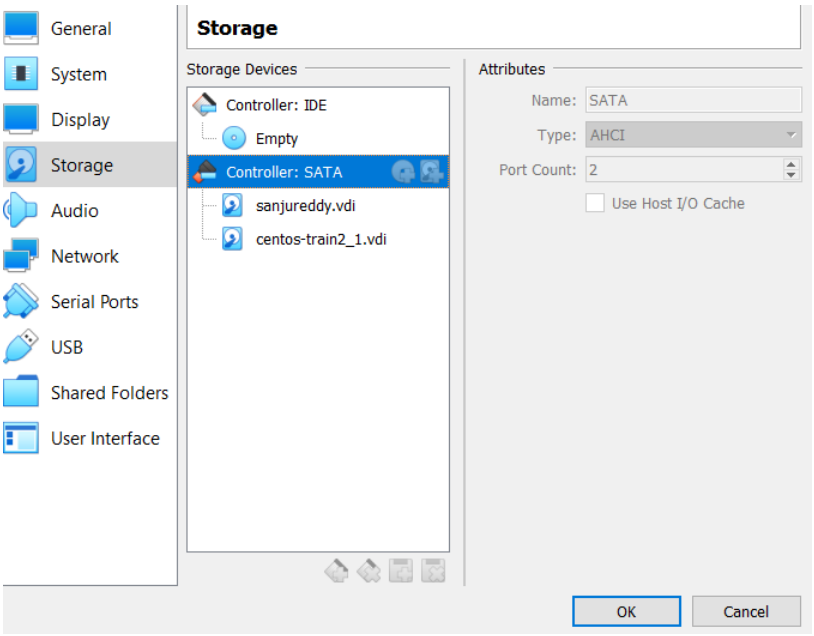
- Create a file system then mount it.
- Add the file system in /etc/fstab file. without editing fstab file remove and unmount filesystem and then restart.

Created a harddisk:

- Goto settings.



- Goto storage.



- Create new harddisk .
- Check whether new harddisk created or not.
    - Use fdisk -l to view partition tables of all attached devices in our system.
    - **Syntax: fdisk -l**

```
[root@sanjana ~]# fdisk -l
Disk /dev/sda: 128 GiB, 137438953472 bytes, 268435456 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xd27b4a58

Device     Boot   Start       End   Sectors  Size Id Type
/dev/sda1  *       2048   2099199   2097152    1G 83 Linux
/dev/sda2       2099200 268435455 266336256  127G 8e Linux LVM


Disk /dev/sdb: 8 GiB, 8589934592 bytes, 16777216 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes




Disk /dev/mapper/cs-root: 70 GiB, 75161927680 bytes, 146800640 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/mapper/cs-swap: 3 GiB, 3263168512 bytes, 6373376 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
```

Created a fileSystem.

- Used mkfs to create a filesystem
- **syntax: mkfs -t[type] target_device**

```
[root@sanjana ~]# mkfs.ext4 /dev/sbd
```

- The commonly used Linux file systems are as follows: ext2, ext3, ext4, JFS, ReiserFS, XFS, and Btrfs.
- checked whether new filesystem is created or not uisng df command
- **syntax: df -h**

```
[root@sanjana ~]# df -h
Filesystem           Size  Used Avail Use% Mounted on
devtmpfs             1.4G     0  1.4G   0% /dev
tmpfs                1.4G     0  1.4G   0% /dev/shm
tmpfs                1.4G  8.4M  1.4G   1% /run
tmpfs                1.4G     0  1.4G   0% /sys/fs/cgroup
/dev/mapper/cs-root   70G  2.2G   68G   4% /
/dev/sda1           1014M  212M  803M  21% /boot
/dev/mapper/cs-home   54G  418M   54G   1% /home
```

Created a directory:

- Used mkdir command to create a direcotry.
- **Syntax: mkdir directory_name**

```
[root@sanjana ~]# mkdir /data
```

- Checked whether directory created or not.

Mounted the filesystem.

- Used mount command to mount the filesystem.
- **Syntax:mount device_name directory_name**
- Checked whether mounted or not using lsblk command.

```
[root@sanjana ~]# mount /dev/sdb /data
[root@sanjana ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda           8:0    0  128G  0 disk
├─sda1        8:1    0    1G  0 part /boot
└─sda2        8:2    0  127G  0 part
  ├─cs-root 253:0    0   70G  0 lvm  /
  ├─cs-swap 253:1    0    3G  0 lvm  [SWAP]
  └─cs-home 253:2    0   54G  0 lvm  /home
sdb           8:16   0    8G  0 disk /data
sr0          11:0    1 1024M  0 rom
```

Added the filesystem in /etc/fstab.

- To make the disk mount permanent after every boot we need to create a mount entry to /etc/fstab

- Step-1: Get the uid of filesystem.
    - Used the blkid command to get the uid of filesystem.

```
[root@sanjana ~]# blkid
/dev/sda1: UUID="6a3b8336-e377-47fc-a916-bd7ec539db06" BLOCK_SIZE="512" TYPE="xfs" PARTUUID="d27b4a58-01"
/dev/sda2: UUID="qI60QL-nu5t-gAnQ-rGXi-TeOC-Ec6O-pOnNM0" TYPE="LVM2_member" PARTUUID="d27b4a58-02"
/dev/sdb: UUID="f3688864-e34a-4c09-9707-1b013025172c" BLOCK_SIZE="4096" TYPE="ext4"
/dev/mapper/cs-root: UUID="ed0f9410-2387-421b-ba17-828328e8539b" BLOCK_SIZE="512" TYPE="xfs"
/dev/mapper/cs-swap: UUID="56e71351-1f2a-4003-a400-5cf18434e8bd" TYPE="swap"
/dev/mapper/cs-home: UUID="1553e91d-9778-41b5-bceb-56b16b89c391" BLOCK_SIZE="512" TYPE="xfs"
[root@sanjana ~]# ^C
[root@sanjana ~]# vi /etc/fstab
```

- Step-2: Added the created filesystem in /etc/fstab.
    - Open the /etc/fstab in vi editor
    - Then added the filesystem in the same format as other filesystems.

```
#
# /etc/fstab
# Created by anaconda on Wed Feb 16 12:58:31 2022
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/cs-root     /                       xfs     defaults        0 0
UUID=3939da3b-d321-43c1-b93d-4cc73ed76fff /boot                   xfs     defaults        0 0
/dev/mapper/cs-home     /home                   xfs     defaults        0 0
/dev/mapper/cs-swap     none                    swap    defaults        0 0
UUID=10ce99a5-00d0-465b-a026-b2862e9068af /dir1 ext4  defaults  0 0
UUID=d774294d-2777-4a65-9300-94028513d8a /data  ext4  defaults 0 0
```

Checked whether filesystem added into /etc/fstab.

- Opened the /etc/fstab using cat command to see content present in it.
- **Syntax: cat /etc/fstab**

```
[root@sanjana ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Fri Mar 25 07:03:43 2022
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/cs-root     /                       xfs     defaults        0 0
UUID=6a3b8336-e377-47fc-a916-bd7ec539db06 /boot                   xfs     defaults        0 0
UUID=1553e91d-9778-41b5-bceb-56b16b89c391 /data                   ext4    defaults        0 0
/dev/mapper/cs-home     /home                   xfs     defaults        0 0
/dev/mapper/cs-swap     none                    swap    defaults        0 0
```

Unmount the filesystem:

- Unmounted the filesystem using umount command.
- **Syntax: umount filesystem**

```
[root@sanjana ~]# umount /dev/sdb
```

- Checked whether unmounted or not using lsblk command.
    - before unmounting

```
[root@sanjana ~]# mkdir /data
[root@sanjana ~]# mount /dev/sdb /data
[root@sanjana ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda           8:0    0  128G  0 disk
├─sda1        8:1    0    1G  0 part /boot
└─sda2        8:2    0  127G  0 part
  ├─cs-root 253:0    0   70G  0 lvm  /
  ├─cs-swap 253:1    0    3G  0 lvm  [SWAP]
  └─cs-home 253:2    0   54G  0 lvm  /home
sdb           8:16   0    8G  0 disk /data
sr0          11:0    1 1024M  0 rom
```

    - After unmounting

```
[root@sanjana ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda           8:0    0  128G  0 disk
├─sda1        8:1    0    1G  0 part /boot
└─sda2        8:2    0  127G  0 part
  ├─cs-root 253:0    0   70G  0 lvm  /
  ├─cs-swap 253:1    0    3G  0 lvm  [SWAP]
  └─cs-home 253:2    0   54G  0 lvm  /home
sdb           8:16   0    8G  0 disk
sr0          11:0    1 1024M  0 rom
```

- Once the file system is attached, the mount point becomes the root directory of the mounted file system.

Reboot the System:

- By using systemctl command reboot the system
- **Syntax: Systemctl reboot**

```
[root@sanjana ~]# systemctl reboot
```

Result:

- After rebooting the system , the system entered into the emergency mode.

```
You are in emergency mode. After logging in, type "journalctl -xb" to view
system logs, "systemctl reboot" to reboot, "systemctl default" or "exit"
to boot into default mode.
Give root password for maintenance
(or press Control-D to continue):
```

- If you want your system not to enter into emergency mode , removed the added filesystem in /etc/fstab.

```
#
# /etc/fstab
# Created by anaconda on Fri Mar 25 07:03:43 2022
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/cs-root     /                       xfs     defaults        0 0
UUID=6a3b8336-e377-47fc-a916-bd7ec539db06 /boot                   xfs     defaults        0 0
/dev/mapper/cs-home     /home                   xfs     defaults        0 0
/dev/mapper/cs-swap     none                    swap    defaults        0 0
~
~
~
```

- Then again rebooted the system and checked whether the system entered into emergency mode or not.
- The system doesn't entered into emergency mode.

```
CentOS Stream 8
Kernel 4.18.0-373.el8.x86_64 on an x86_64

sanjana login:
```

USER STORY 3

Write a script in Linux to get an alert when the disk size reaches 90%

DESCRIPTION:

- Write a script in Linux to get an alert when the disk size reaches 90%

Installing mailx:

- Before installing mailx, updated all installed packages that are available in this repository.
- **Syntax: yum -y update**

```
[sanjana@sanjana /]$ sudo yum -y update
```

```
Upgraded:
  audit-3.0.7-3.el8.x86_64
  dnf-4.7.0-8.el8.noarch
  expat-2.2.5-8.el8.x86_64
  glibc-gconv-extra-2.28-196.el8.x86_64
  libnfsidmap-1:2.3.3-51.el8.x86_64
  python3-dnf-4.7.0-8.el8.noarch
  python3-libdnf-0.63.0-8.el8.x86_64
  samba-client-libs-4.15.5-5.el8.x86_64
  yum-4.7.0-8.el8.noarch

Complete!
```

- Installed the mailx using yum command.

- **Syntax: yum install -y mailx**

```
[sanjana@sanjana /]$ sudo yum install -y mailx
ast metadata expiration check: 13:16:03 ago on
```

```
--------------------------------------------------------------------------------
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :
  Installing       : mailx-12.5-29.el8.x86_64
  Running scriptlet: mailx-12.5-29.el8.x86_64
  Verifying        : mailx-12.5-29.el8.x86_64

Installed:
  mailx-12.5-29.el8.x86_64

Complete!
```

- It is a console application that is used for sending and receiving emails.

Set an external SMTP Server to relay E-Mails:

- opened /etc/mail.rc file using vi editor.
- **Syntax:vi /etc/mail.rc**

```
[sanjana@sanjana /]$ sudo vi /etc/mail.rc
```

- Edited the /etc/mail.rc file with following code.

```
# For Linux and BSD, this should be set.
set bsdcompat
set smtp=smtps://smtp.gmail.com:465
set smtp-auth=login
set smtp-auth-user=sanjanareddypeddi@gmail.com
set smtp-auth-password=asuimwdooscoiemz
set ssl-verify=ignore
set nss-config-dir=/etc/pki/nssdb/
```

- The connection to SMTP server is established.

Write script to send alert mail.

- Opened the file using vi editor.

```
[sanjana@sanjana /]$ sudo vi /home/sanjana/check_disk.sh
```

- Wrote script to send alert mail.

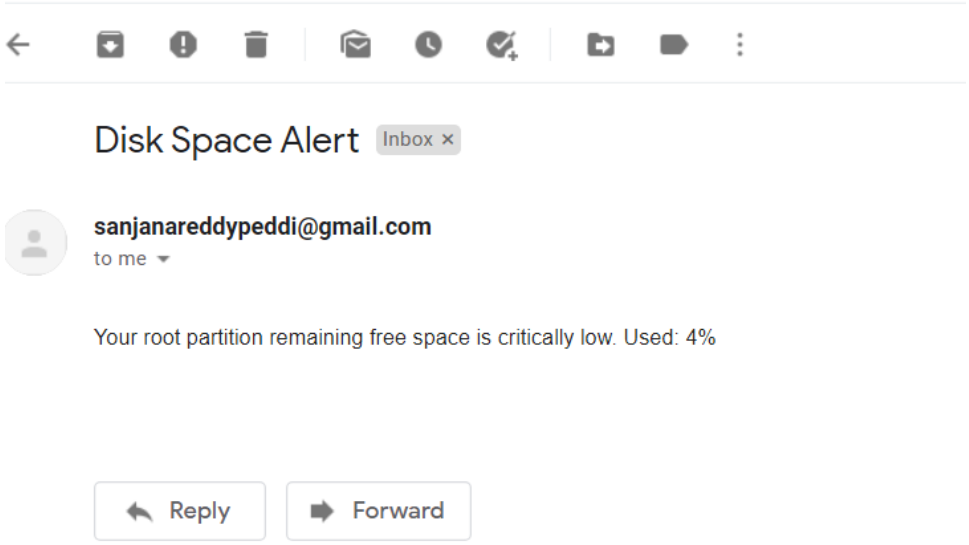
```
#!/bin/bash

CURRENT=$(df / | grep / | awk '{ print $5}' | sed 's/%//g')

THRESHOLD=90

if [ "$CURRENT" -ge "$THRESHOLD" ] ; then

mail -s 'Disk Space Alert' sanjanareddypeddi@gmail.com << EOF

Your root partition remaining free space is critically low. Used: $CURRENT%

EOF

fi
~
~
```

Result:

- Executed the check_disk.sh

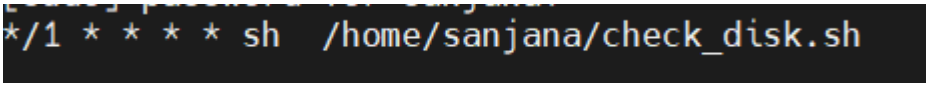
```
[sanjana@sanjana /]$ sudo sh /home/sanjana/check_disk.sh
```

- As memory not extended threshold 90, so mail was not prompted.
- Changed the threshold to 1, so got mail as follow.



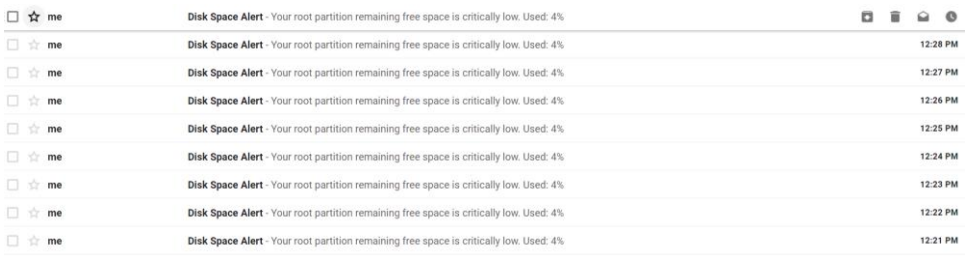
Automated the script execution :

- Created a crontab to execute the script file for every minute.



Result:

- Got mails for every minute.



USER STORY 4

Resetting root password

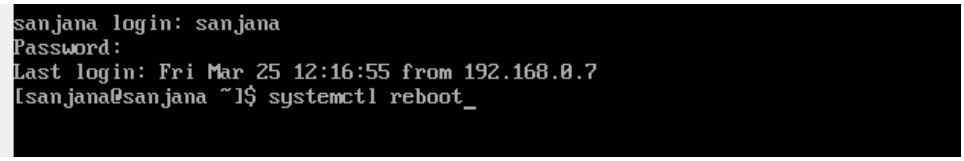
DESCRIPTION:

- Resetting root password of a VM using kernel.

Explanation:

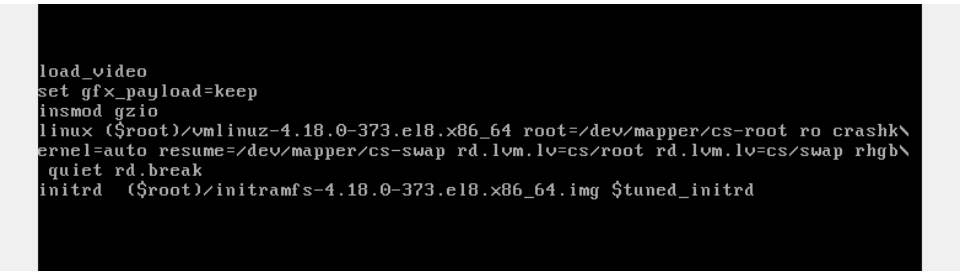
Reboot the system:

- Used the systemctl command to reboot the system.
- **Syntax: systemctl reboot.**



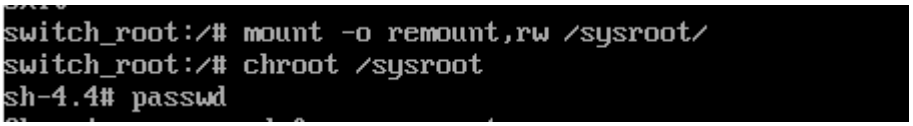
Interrupt boot process:

- While booting process is going on press 'esc'
- Use the arrows to highlight the line that starts with kernel or Linux.
- **Press 'E' :**It will go to end of line.
- **Then add rd.break**.
- Adding rd.break to the end of the line with kernel parameters in Grub stops the start up process before the regular root filesystem is mounted.



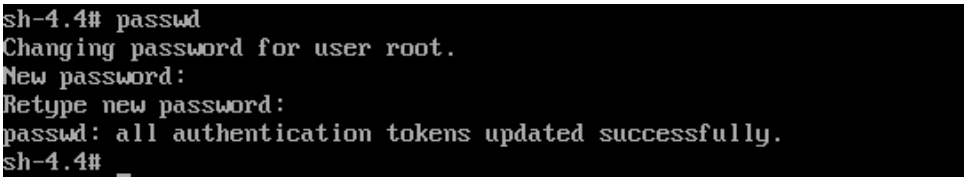
Remount the sysroot directory:

- Initially the sysroot directory is in read only mode, changed it to read and write mode.
- **Syntax: mount -o remount,rw /sysroot**
- Then changed the root directory.
- **syntax: chroot /sysroot**
- The chroot /sysroot command means: "start a new shell in such a way that for that shell the /sysroot directory will appear as /."

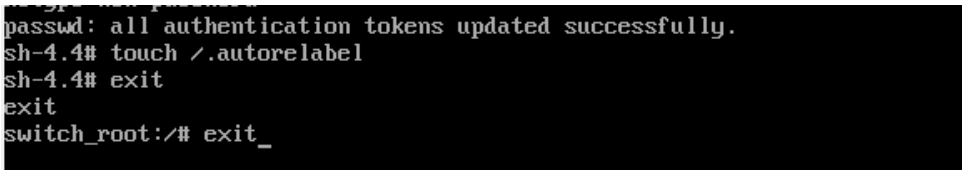


Changed the password:

- Used passwd command to change the root passwd .
- **Syntax: passwd**



- Then execute the command touch /.autorelabel
- The touch /.autorelabel command creates a hidden file named .autorelabel under the root directory. On the next boot, the SELinux subsystem will detect this file, and then relabel all of the files on that system with the correct SELinux(Secured-enhanced linux) contexts.



USER STORY 5

Install the Apache package. Allow it to get documents stored on NFS mounted directories

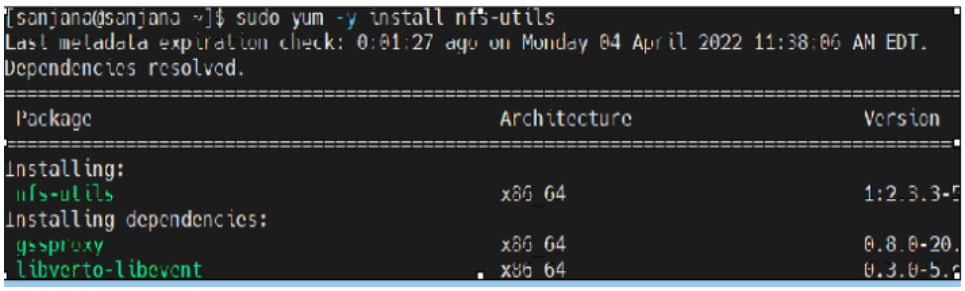
DESCRIPTION:

- Install the Apache package. Allow it to get documents stored on NFS mounted directories.

Explanation:

Installed NFS service:

- Installed NFS service using yum command.
- **Syntax: yum -y install nfs-utils**



Checked whether service installed or not:

- Checked whether service installed or not using yum and grep command.
- All the packages installed were present in the yum list.
- **Syntax: yum list installed | grep nfs-utils**

```
[sanjana@sanjana ~]$ sudo yum  list installed | grep nfs-utils
nfs-utils.x86_64                 1:2.3.3-51.el8                  @baseos
[sanjana@sanjana ~]$
```

Started and enabled the service:

- Started the service using systemctl command.
- **Syntax:systemctl start nfs-server.service**

```
[sanjana@sanjana ~]$ sudo service nfs-server start
Redirecting to /bin/systemctl start nfs-server.service
```

- Enabled the service using systemctl command.
- **Syntax:systemctl enable nfs-server.service**

```
[sanjana@sanjana ~]$ sudo systemctl enable nfs-server.service
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service →
[sanjana@sanjana ~]$ sudo systemctl status nfs-server.service
```

- Checked whether service enabled or not using systemctl command.

```
[sanjana@sanjana ~]$ sudo systemctl status nfs-server.service
● nfs-server.service - NFS server and services
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; ve
   Active: active (exited) since Mon 2022-04-04 11:40:47 EDT; 49s ago
 Main PID: 2969 (code=exited, status=0/SUCCESS)
    Tasks: 0 (limit: 17800)
   Memory: 0B
   CGroup: /system.slice/nfs-server.service
```

Created a directory :

- Created a directory using mkdir command.
- **Syntax: mkdir dir_name**

```
[sanjana@sanjana ~]$ sudo mkdir dir1
```

- Checked whether directory created or not using ls command.

```
[sanjana@sanjana ~]$ ls -l
total 20
-rwxr-xr-x. 1 root    root     279 Mar 29 05:35 check_disk.sh
drwxr-xr-x. 2 root    root       6 Apr  4 11:43 dir1
-rw-rw-r--. 1 sanjana sanjana 1653 Mar 30 12:57 example.html
```

Created some files in directory:

- Created two files using touch command.
- **Syntax: touch file1 file2**

```
[sanjana@sanjana ~]$ cd /home/sanjana/dir1
[sanjana@sanjana dir1]$ touch f1 f2
```

- Wrote the content using cat command.

```
[sanjana@sanjana dir1]$ cat f1
hii vijaya Iam sanju reddy
Whatsuppp brooo
[sanjana@sanjana dir1]$ cat f2
hii sanju reddy iam sathwi reddy
whatsupp dude
[sanjana@sanjana dir1]$
```

Changed the permissions for the directory:

- Changed the permissions for the directory using chmod and chown commands.

- Changed the owner of the directory to nobody using chown command.
- **Syntax : chown -R nobody:dir_name**
- "nobody" means removed all the restrictions of directory.

```
[sanjana@sanjana ~]$ sudo chown -R nobody: dir1
[sanjana@sanjana ~]$ ls -l |grep dir1
drwxrwxrwx. 2 nobody  nobody    26 Apr  4 12:46 dir1
[sanjana@sanjana ~]$
```

- Changed the permissions for the directory using chmod command.
- **Syntax : chmod -R 777 dir_name**

```
[sanjana@sanjana ~]$ sudo chmod -R 777 dir1
[sanjana@sanjana ~]$ ls -l |grep dir1
drwxrwxrwx. 2 nobody  nobody    26 Apr  4 12:46 dir1
[sanjana@sanjana ~]$
```

Created a file to export the NFS:

- The directory created was exported to the client machine using a file.
- First created an export file in the etc directory using vi command.
- **Syntax**: **vi /etc/exports**

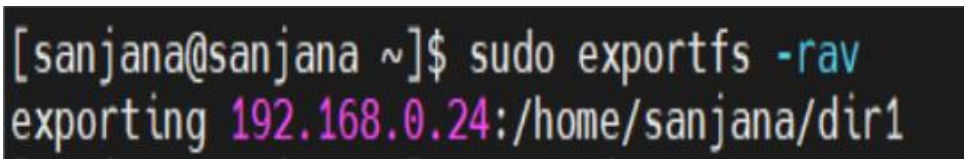
```
[sanjana@sanjana ~]$ vi /etc/exports
[sanjana@sanjana ~]$ cat /etc/exports
dir1 192.168.0.24(rw,sync,no_all_squash,root_squash)
```

- **rw** - allows us to read and write to NFS directory.
- **sync** - requires writing of changes to the disk before any other operations are completed.
- **no_all_squash** - maps all UIDs and GIDs from the client request to the identical UIDs and GIDs of the NFS server.

- **root_squash** - maps requests from the client-side root user to an anonymous UID and GID

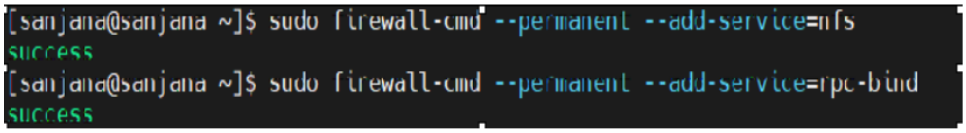
Exported the NFS to the client:

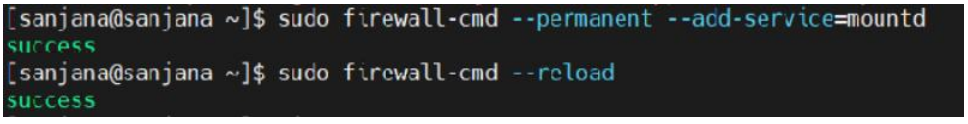
- Exported nfs using exportfs command.
- **Syntax:exportfs -rav**



Modified the firewall:

- The firewall was modified to add the NFS service to the firewall.
- rpc-bind and mountd were also added to the firewall to access the NFS service .
- **Command:** firewall-cmd --permanent --add-service=nfs.
- **Command:** firewall-cmd permanent --add-service=rpc-bind.
- **Command:** firewall-cmd permanent --add-service=mountd.
- **Command:** firewall-cmd --reload.





Installed httpd:

- Installed httpd using yum command.
- **Syntax: yum install httpd**

```
[root@vijaya /]# yum -y install httpd
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: centos.excellmedia.net
 * extras: centos.excellmedia.net
 * updates: centos.excellmedia.net
Package httpd-2.4.6-97.el7.centos.5.x86_64 already installed and latest version
Nothing to do
[root@vijaya /]#
```

- HTTPd stands for **Hypertext Transfer Protocol daemon**.
- It usually is the main software part of an HTTP server better known as a web server

Created an entry for NFS server in /etc/hosts file

- The ip address and directory path of the server were added to the /etc/hosts file to access the NFS server files.
- **Syntax**: **vi /etc/hos**

```
[root@vijaya vijaya]# vi /etc/hosts
[root@vijaya vijaya]#
[root@vijaya vijaya]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.50.15.233 localhost
192.168.0.20 localhost
```

Created a mount directory to access the files of server machine

- The directory was created using mkdir command
- The directory was created in /var/www/html directory because the httpd can access files present only in html directory
- **Syntax**: **mkdir /var/www/html/directory-name**

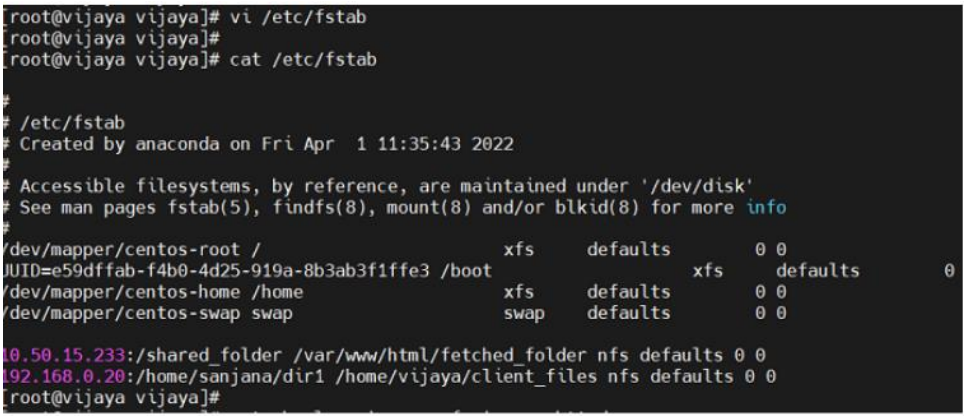
```
[root@vijaya vijaya]# mkdir client_files
[root@vijaya vijaya]# ls
client_files  nfs_dir
```

Mounted NFS onto the directory

- The NFS server was mounted onto the directory using mount command
- **Syntax**: **mount -t nfs host-ip-address:/host-dir-path client-dir-path**

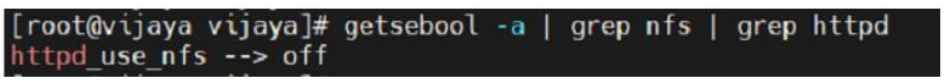
Edited /etc/fstab file

- Added the mount details into fstab file
- **Command**: vi /etc/fstab



Checked the SElinux boolean:

- Used getsebool command.



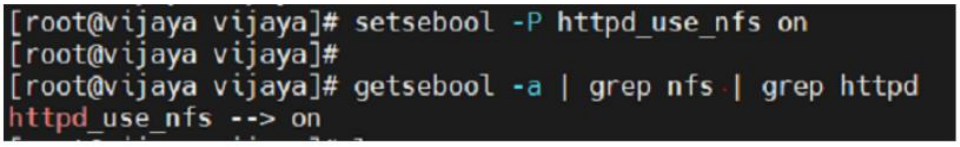
- getsebool **reports where a particular SELinux boolean are on or off.**

Checking the files by accessing through httpd on web browser

- The files were checked whether they can be accessible or not using the ip address of the client machine along with the directory path
- **Syntax** client-ip-address:/directory-path

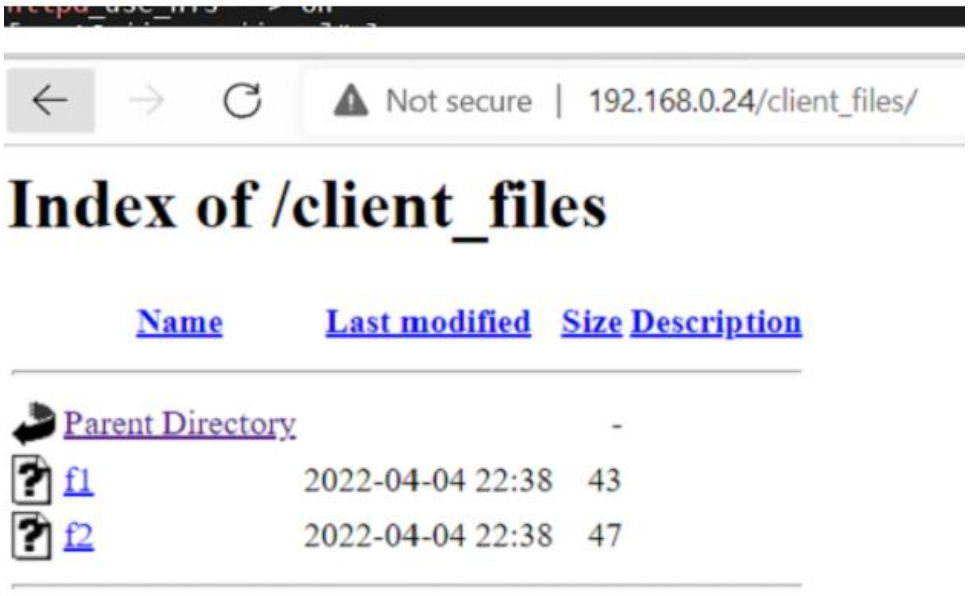
Set the SElinux boolean:

- Used setsebool to change the boolean.
- Initially it is off ,changed to on.
- setsebool **sets the current state of a particular SELinux boolean to a given value**.



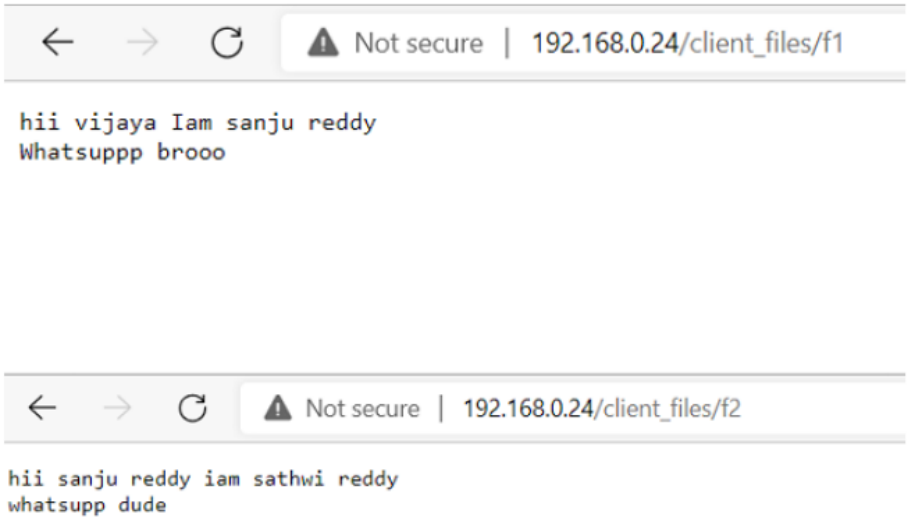
Checking the files by accessing through httpd on web browser

- The files were checked whether they can be accessible or not using the ip address of the client machine along with the directory path
- **Syntax** client-ip-address:/directory-path



Checking the content of files:

- Checked the content of files of servers.



USER STORY 6

Create an ext4 file system on a new logical volume of 100MB called lv_ext4.

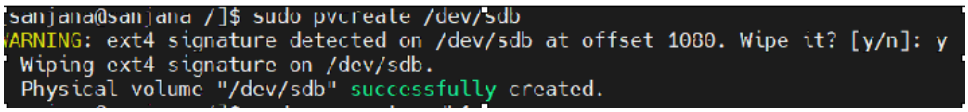
DESCRIPTION:

- Create an ext4 file system on a new logical volume of 100MB called lv_ext4. Mount it permanently under the /ext4 directory. Copy the files previously created into this new space.

Explanation:

Created a physical volume:

- A physical volume is a storage device or partition.
- Created a physical volume using 'pvcreate'.



- Checked whether physical volume created or not uisng 'pvdisplay'

```
[sanjana@sanjana /]$ sudo pvdisplay
[sudo] password for sanjana:
  --- Physical volume ---
  PV Name               /dev/sdb
  VG Name               sdb1
  PV Size               8.00 GiB / not usable 4.00 MiB
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              2047
  Free PE               2022
  Allocated PE          25
  PV UUID               6yRT3i-fEiB-9lSd-zIHH-8sbE-IS65-dM3DKM
```

Created a volume group:

- Volume group is the highest level of abstraction.
- created the logical volume using 'vgcreate'.

```
[sanjana@sanjana /]$ sudo vgcreate sdb1 /dev/sdb
[sudo] password for sanjana:
  Volume group "sdb1" successfully created
```

Created a logical volume of 100MB:

- Logical volumes are block devices which are created from the physical extents present in the same volume group.
- Created the logical volume of 100MB using 'lvcreate'.

```
[sanjana@sanjana /]$ sudo lvcreate --size 100M --name lv_ext4 sdb1
  Logical volume "lv_ext4" created.
```

- Checked whether logical volume created or not uisng 'lsblk'.

```
[sanjana@sanjana /]$ lsblk
NAME           MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda              8:0    0  128G  0 disk
├─sda1           8:1    0    1G  0 part /boot
└─sda2           8:2    0  127G  0 part
  ├─cs-root    253:0    0   70G  0 lvm  /
  ├─cs-swap    253:1    0    3G  0 lvm  [SWAP]
  └─cs-home    253:2    0   54G  0 lvm  /home
sdb              8:16   0    8G  0 disk
└─sdb1-lv_ext4 253:3    0  100M  0 lvm
sr0             11:0    1 1024M  0 rom
```

Created the filesystem:

- Used 'mkfs' command to create a filesystem.
- **Syntax : mkfs.[fs type] [target device]**
- It format a disk into specific filesystem.

```
[sanjana@sanjana /]$ sudo mkfs.ext4 /dev/sdb1/lv_ext4
mke2fs 1.45.6 (20-Mar-2020)
Creating filesystem with 102400 1k blocks and 25688 inodes
Filesystem UUID: 8517fa0c-07b3-4f39-ba13-24affa8f74db
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

Mounted the filesystem:

- Created a directory using mkdir command.

```
[sanjana@sanjana ~]$ sudo mkdir /ext4
```

- Used mount command to mount the filesystem.
- **Syntax:mount device_name directory_name**

- Checked whether mounted or not using lsblk command.

```
sanjana@sanjana /]$ sudo mount /dev/sdb1/lv_ext4 /ext4
```

```
[sanjana@sanjana /]$ lsblk
NAME           MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda              8:0    0  128G  0 disk
├─sda1           8:1    0    1G  0 part /boot
└─sda2           8:2    0  127G  0 part
  ├─cs-root    253:0    0   70G  0 lvm  /
  ├─cs-swap    253:1    0    3G  0 lvm  [SWAP]
  └─cs-home    253:2    0   54G  0 lvm  /home
sdb              8:16   0    8G  0 disk
└─sdb1-lv_ext4 253:3    0  100M  0 lvm  /ext4
sr0             11:0    1 1024M  0 rom
```

Added the filesystem in /etc/fstab.

- Step-1: Get the uid of filesystem.
    - Used the blkid command to get the uid of filesystem.

```
[sanjana@sanjana /]$ sudo blkid
/dev/sda1: UUID="6a3b8336-e377-47fc-a916-bd7ec539db06" BLOCK_SIZE="512" TYPE="xfs" PARTUUID="d27b4a58-01"
/dev/sda2: UUID="qI60QL-nu5t-gAnQ-rGXi-TeOC-Ec6O-pOnNM0" TYPE="LVM2_member" PARTUUID="d27b4a58-02"
/dev/sdb: UUID="6yRT3i-fEiB-9lSd-zIHH-8sbE-IS65-dM3DKM" TYPE="LVM2_member"
/dev/mapper/cs-root: UUID="ed0f9410-2387-421b-ba17-828328e8539b" BLOCK_SIZE="512" TYPE="xfs"
/dev/mapper/cs-swap: UUID="56e71351-1f2a-4003-a400-5cf18434e8bd" TYPE="swap"
/dev/mapper/cs-home: UUID="1553e91d-9778-41b5-bceb-56b16b89c391" BLOCK_SIZE="512" TYPE="xfs"
/dev/mapper/sdb1-lv_ext4: UUID="8517fa0c-07b3-4f39-ba13-24affa8f74db" BLOCK_SIZE="1024" TYPE="ext4"
[sanjana@sanjana /]$ ^C
```

- Step-2: Added the created filesystem in /etc/fstab.
    - Open the /etc/fstab in vi editor
    - Then added the filesystem in the same format as other filesystems.

```
[sanjana@sanjana /]$ sudo vi /etc/fstab
```

```
[sanjana@sanjana /]$ sudo cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Fri Mar 25 07:03:43 2022
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/cs-root     /                       xfs     defaults        0 0
UUID=6a3b8336-e377-47fc-a916-bd7ec539db06 /boot                   xfs     defaults        0 0
UUID=8517fa0c-07b3-4f39-ba13-24affa8f74db                         ext4    defaults        0 0
/dev/mapper/cs-home     /home                   xfs     defaults        0 0
/dev/mapper/cs-swap     none                    swap    defaults        0 0
```

Copied the files previously created into this new space:

- Copied the files previously created using 'cp' command.
- **Syntax: cp source_file destination**

```
[sanjana@sanjana ~]$ sudo cp -p * /ext4
```

- cp '-p' option is used to preserve the properties and attributes of a file.

Checked whether files copied or not:

- Used ls command to check whether files or copied or not.

```
[sanjana@sanjana ~]$ ls /ext4
check_disk.sh  f1  f2  f3  f4  f5  lost+found
```

USER STORY 7

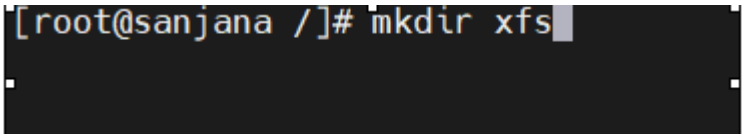
Write a Bash Script to create 40 files

DESCRIPTION:

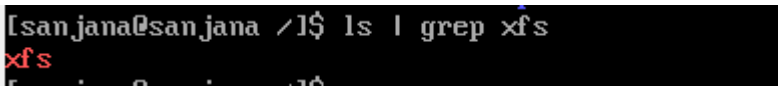
- Write a bash script create 40 files.
- Check whether files created or not.

Created a directory:

- Created a directory /xfs using mkdir command.
- **syntax:mkdir xfs**

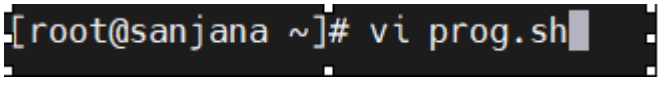


- Checked whether a directory created or not using ls command.



Write a bash script create 40 files.

- Opened prog.sh file using vi editor.



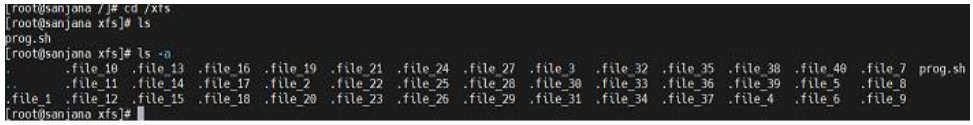
- Updated the bash script in prog.sh file.

```
[root@sanjana ~]# cat prog.sh
#!/bin/bash
cd /xfs
N=40
for N in $(seq 40)
do
fallocate -l 2M .file_$N
N=`expr $N - 1`
done
```

- **fallocate command** – Preallocate space to a file.

Result:

- After executing the prog.sh file 40 files got created as follows.



USER STORY 8

Configure a HTTP server and downloaded release and accessed file.

DESCRIPTION:

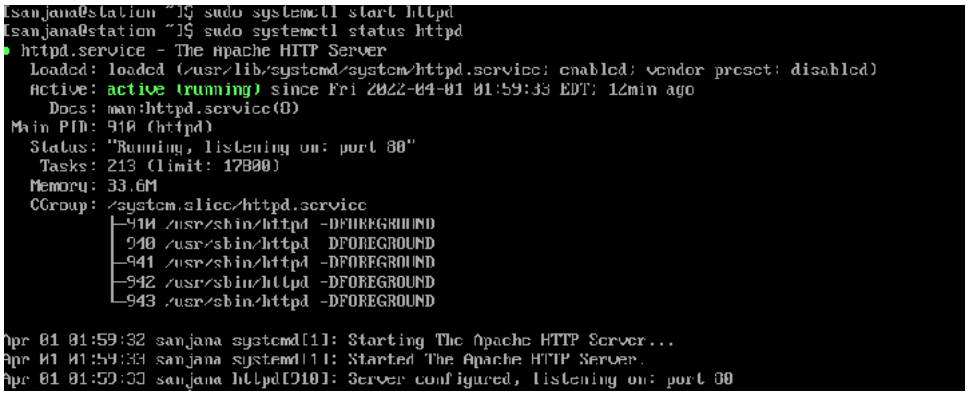
- Configure a HTTP server, which can be accessed through http://station.domain40.example.com. Please download the released page from http://ip/dir/example.html.

Configure the Http Server:

- Initially changed the hostname using 'hostnamectl' command.

- Installed httpd using yum command.

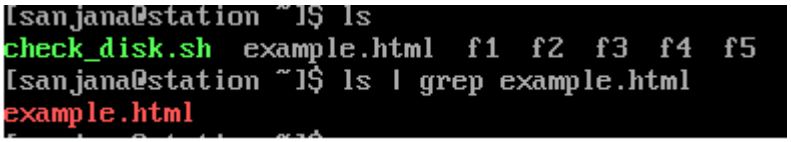
- Then started the httpd using systemctl command.
- Then checked status of httpd whether it is active or not.



Download the released page:

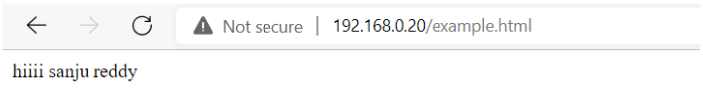
- Used the wget command to download the file.
- Wget is the non-interactive network downloader which is used to download files from the server.

- Downloaded the file example.html.



Accessed the file:

- Accessed the file through http://ip/dir/example.html.



USER STORY 9

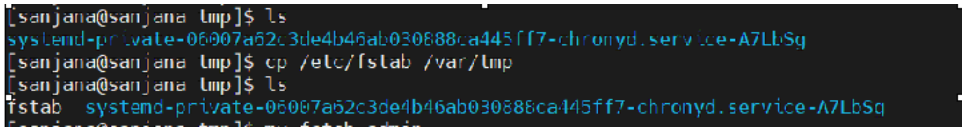
Copy /etc/fstab to /var/tmp

DESCRIPTION:

- Create a user called user2.
- Change the permissions of user2.

Copied /etc/fstab to /var/tmp.

- Used 'cp' command to copy /etc/fstab to /var/tmp.
- **Syntax: cp source destination**
- Then checked whether the file copied or not using 'ls' command.

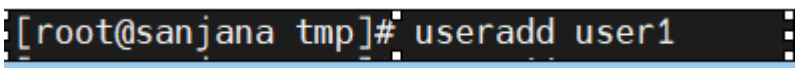


Changed the name of fstab to admin.

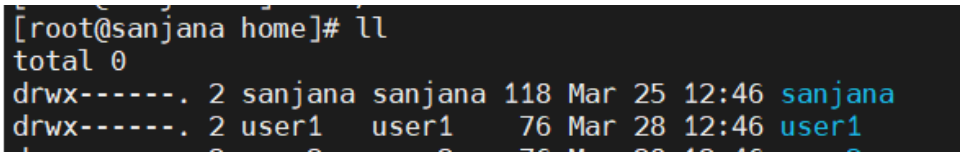
- Used 'mv' command to rename fstab with admin.
- **Syntax: mv old_filename new_filename**
- Checked whether name changed or not with 'ls' command.

Created a user called user1:

- Created user1 using 'useradd' command.
- **Syntax:useradd user1**



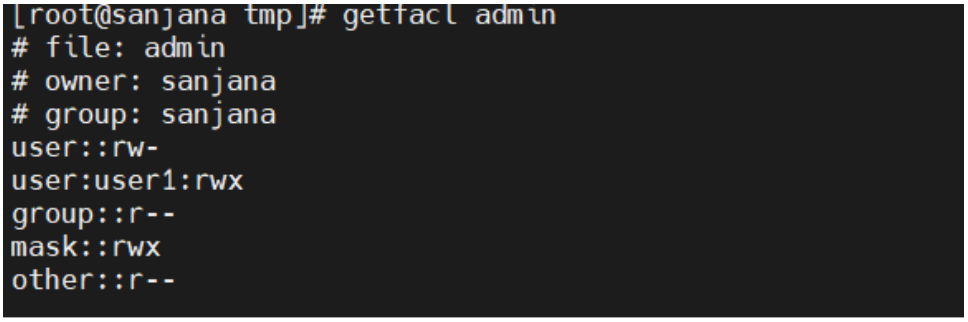
- Checked whether user created or not using 'll' command.



Changed the permissions of user1:

- Set permissions for user1 using 'setfacl' command.
- **Syntax: setfacl -m "u:user_name:rwx" file_name**
- It sets file access control list.

- Used 'getfacl' to check whether permissions are set for user1 or not.
- **Syntax: getfacl file_name**
- For each file, getfacl displays the file name, owner, the group, and the Access Control List.



Result:

- Switched to user1.
- Then read the file admin using cat command.

```
[root@sanjana tmp]# su user1
[user1@sanjana tmp]$ cat admin

#
# /etc/fstab
# Created by anaconda on Fri Mar 25 07:03:43 2022
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/cs-root     /                       xfs     defaults        0 0
UUID=6a3b8336-e377-47fc-a916-bd7ec539db06 /boot                   xfs     defaults        0 0
/dev/mapper/cs-home     /home                   xfs     defaults        0 0
/dev/mapper/cs-swap     none                    swap    defaults        0 0
```

Created a user called user2:

- Created user2 using 'useradd' command.
- **Syntax:useradd user2**

```
[root@sanjana tmp]# useradd user1
```

- Checked whether user created or not using 'll' command.

```
[root@sanjana home]# ll
total 0
drwx------. 2 sanjana sanjana 118 Mar 25 12:46 sanjana
drwx------. 2 user1   user1    76 Mar 28 12:46 user1
drwx------. 2 user2   user2    76 Mar 28 12:46 user2
```

Changed the permissions of user2:

- Set permissions for user2 using 'setfacl' command.
- **Syntax: setfacl -m "u:user_name:rwx" file_name**
- It sets file access control list.
- Gave no permissions for user2

```
[sanjana@sanjana ~]$ setfacl -m "u:user2:---" admin
```

- Used 'getfacl' to check whether permissions are set for user2 or not.
- **Syntax: getfacl file_name**

- For each file, getfacl displays the file name, owner, the group, and the Access Control List.

```
[root@sanjana tmp]# setfacl -m "u:user2:---" admin
[root@sanjana tmp]# getfacl admin
# file: admin
# owner: sanjana
# group: sanjana
user::rw-
user:user1:rwx
user:user2:---
group::r--
mask::rwx
other::r--
```

Result:

- Switched to user2.
- Then read the file admin using cat command.
- Here got permission denied because, set no permission for user2.

```
[root@sanjana tmp]# su user2
[user2@sanjana tmp]$ cat admin
cat: admin: Permission denied
[user2@sanjana tmp]$
```

USER STORY 10

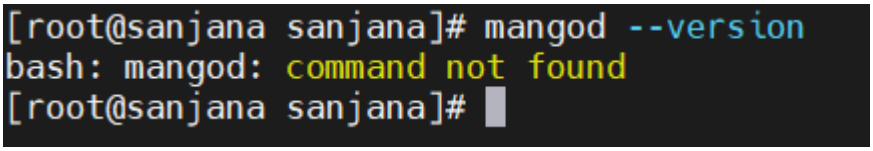
Install MongoDB in Linux and CURD operation.

DESCRIPTION:

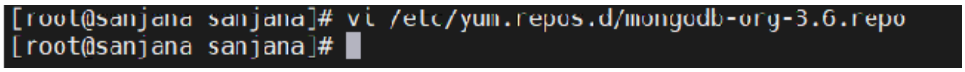
- Install MongoDB in Linux and run CRUD command, create a cron job and move data from Mongo

Created a mongodb-org-3.6.repo file:

- Checked whether mongodb is installed or not.



- To add the repository, created a mongodb-org-3.6.repo file and opened using vi.



- Added the following content to this file.

- Save and exit it.

Install mongodb:

- Installed mongodb using yum command.

```
[root@sanjana sanjana]#  yum install -y mongodb-org
MongoDB Repository
Dependencies resolved.
```

- Mongodb got installed.

```
Installed:
  mongodb-org-3.6.23-1.el8.x86_64
  mongodb-org-server-3.6.23-1.el8.x86_64
  mongodb-org-tools-3.6.23-1.el8.x86_64
  python2-libs-2.7.18-10.module_el8.6.0+1092+a03304bb.x86_64
  python2-pip-wheel-9.0.3-19.module_el8.6.0+987+71f62bb6.noarch
  python2-setuptools-wheel-39.0.1-13.module_el8.5.0+743+cd2f5d28.noarch

Complete!
[root@sanjana sanjana]#
```

- Checked whether mongodb is installed or not.

```
[root@sanjana sanjana]# yum list installed | grep mongodb
mongodb-org.x86_64                  3.6.23-1.el8
mongodb-org-mongos.x86_64           3.6.23-1.el8
mongodb-org-server.x86_64           3.6.23-1.el8
mongodb-org-shell.x86_64            3.6.23-1.el8
mongodb-org-tools.x86_64            3.6.23-1.el8
[root@sanjana sanjana]#
```

Start mongodb:

- Started mongodb using systemctl command.

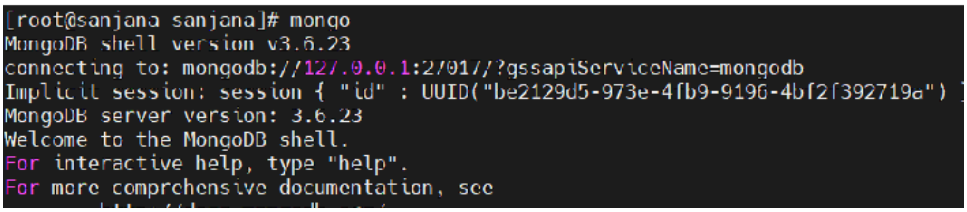
```
[root@sanjana sanjana]# systemctl start mongod
```

- Checked the status of mongodb.

```
[root@sanjana sanjana]# systemctl status mongod
● mongod.service - MongoDB Database Server
   Loaded: loaded (/usr/lib/systemd/system/mongod.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2022-04-01 12:47:58 EDT; 6s ago
     Docs: https://docs.mongodb.org/manual
  Process: 4610 ExecStart=/usr/bin/mongod $OPTIONS (code=exited, status=0/SUCCESS)
  Process: 4608 ExecStartPre=/usr/bin/chmod 0755 /var/run/mongodb (code=exited, status=0/SUCCESS)
  Process: 4607 ExecStartPre=/usr/bin/chown mongod:mongod /var/run/mongodb (code=exited, status=0/SUCCESS)
  Process: 4605 ExecStartPre=/usr/bin/mkdir -p /var/run/mongodb (code=exited, status=0/SUCCESS)
 Main PID: 4613 (mongod)
   Memory: 46.0M
   CGroup: /system.slice/mongod.service
           └─4613 /usr/bin/mongod -f /etc/mongod.conf
```

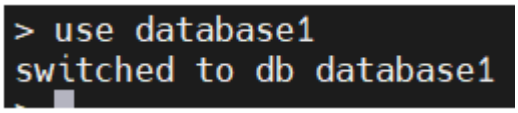
Opened mongodb environment:

- Opened mongodb environment using mongo.

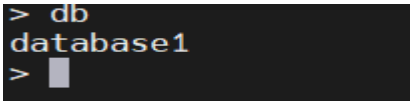


Created a database:

- Created a database

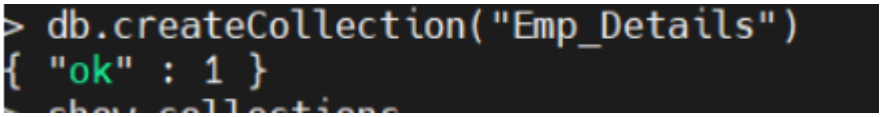


- Checked currently selected database, use the command **db.**

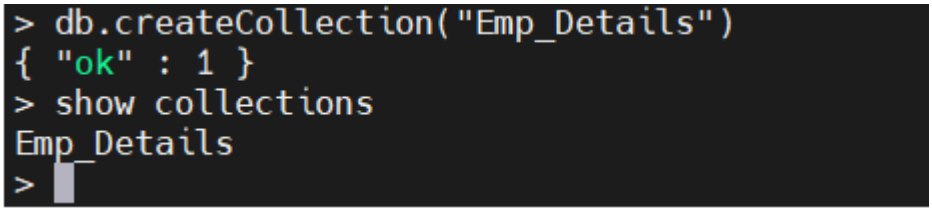


Created Collection:

- Created a Collection.
- MongoDB db.createCollection(name) is used to create collection.



- Checked whether Collection created or not using show command.



Performed Insert operation:

- Inserted the data into Collection.

- **Syntax: db.Colection_name.insert(document)**

```
Emp_Details
> db.Emp_Details.insert(
... {"eid":"MT4020","name":"sanjana","salary":"10l"}
... )
WriteResult({ "nInserted" : 1 })
>
```

- Checked whether data inserted or not using find command.

```
> db.Emp_Details.find()
{ "_id" : ObjectId("62472e9531fc0351db05cebc"), "eid" : "MT4020", "name" : "sanjana", "salary" : "10l" }
>
```

Performed update operation:

- Updated the data using update() method.
- **Syntax:db.collection_name.update(selection_criteria,updated_data)**

```
> db.Emp_Details.update({"name":"sanjana"},{$set:{"name":"sanju reddy"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

- Checked whether data updated or not.
    - Before updating

```
> db.Emp_Details.find()
{ "_id" : ObjectId("62472e9531fc0351db05cebc"), "eid" : "MT4020", "name" : "sanjana", "salary" : "10l" }
>
```

    - After updating.

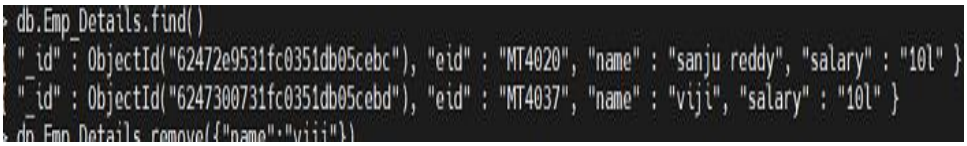
```
> db.Emp_Details.find()
{ "_id" : ObjectId("62472e9531fc0351db05cebc"), "eid" : "MT4020", "name" : "sanju reddy", "salary" : "10l" }
>
```

Performed delete operation:

- Performed deleted operation using remove() function.
- **Syntax:db.collection_name.remove(deletion_Criteria)**

```
> db.Emp_Details.remove({"name":"viji"})
WriteResult({ "nRemoved" : 1 })
>
```

- Checked whether data deleted or not.
    - Before deletion.



    - After deletion.

Created a crontab to transfer data to file:

- Created a crontab to transfer data to file.
- **Syntax:crontab -e**

- 'mangoexport' to export data from a MongoDB instance in a CSV or JSON file format.

Checked the file:

- Checked the file to see whether data present or not.