Everlyn Leon

Abstract
Demonstrating the use and features of the EXT4 filesystem such as creating a mounted file system on a Ubuntu Virtual Machine. The EXT4 filesystem is preferred due to being an actively-developed filesystem, compared to a filesystem such as ReiserFS which had its development end in 2008. Although ReiserFS's development ended, it was one of the best filesystems for Linux because of its efficient use of disk space, scalability, and performance when compared to EXT2 and EXT3, but with EXT4's major improvements and backward and forwards compatibility with EXT2 and EXT3, EXT4 is now the preferred filesystem.

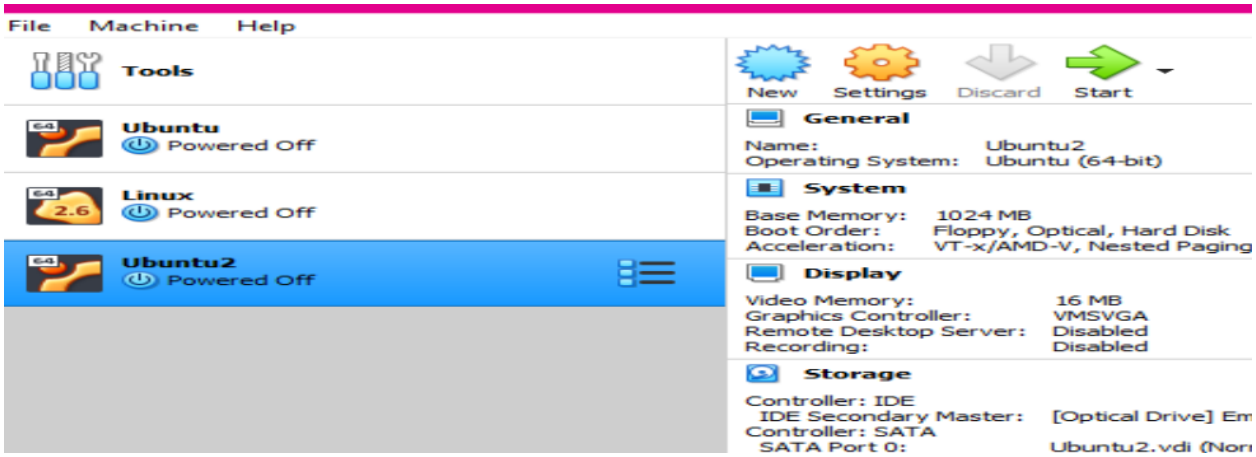
Introduction
Partitioning an Ubuntu Virtual Machine to mount the filesystem named EXT4. A new user will be created and placed into a security group who will have full access to the new file system. The use and features of the EXT4 filesystem will be demonstrated with the new user account, by showing how the user can unmount and mount the new file system. Unauthorized commands and mounting and unmounting of the filesystem by a new user not in the security group will be demonstrated as well.

Summary of Results
A new virtual hard drive will be added to VirtualBox to demonstrate the different usages of the EXT4 filesystem. Here the new Virtual Machine named 'Ubuntu2' was created. After the new VM is created, update the machine via the terminal using the command:

sudo apt-get update

Commands:

sudo	(root privileges	)
apt-get	(retrieve information and packages	)
update	(newer than the existing corresponding files	)



Now the new drive needs to be partitioned. To check the partitions, use the command:

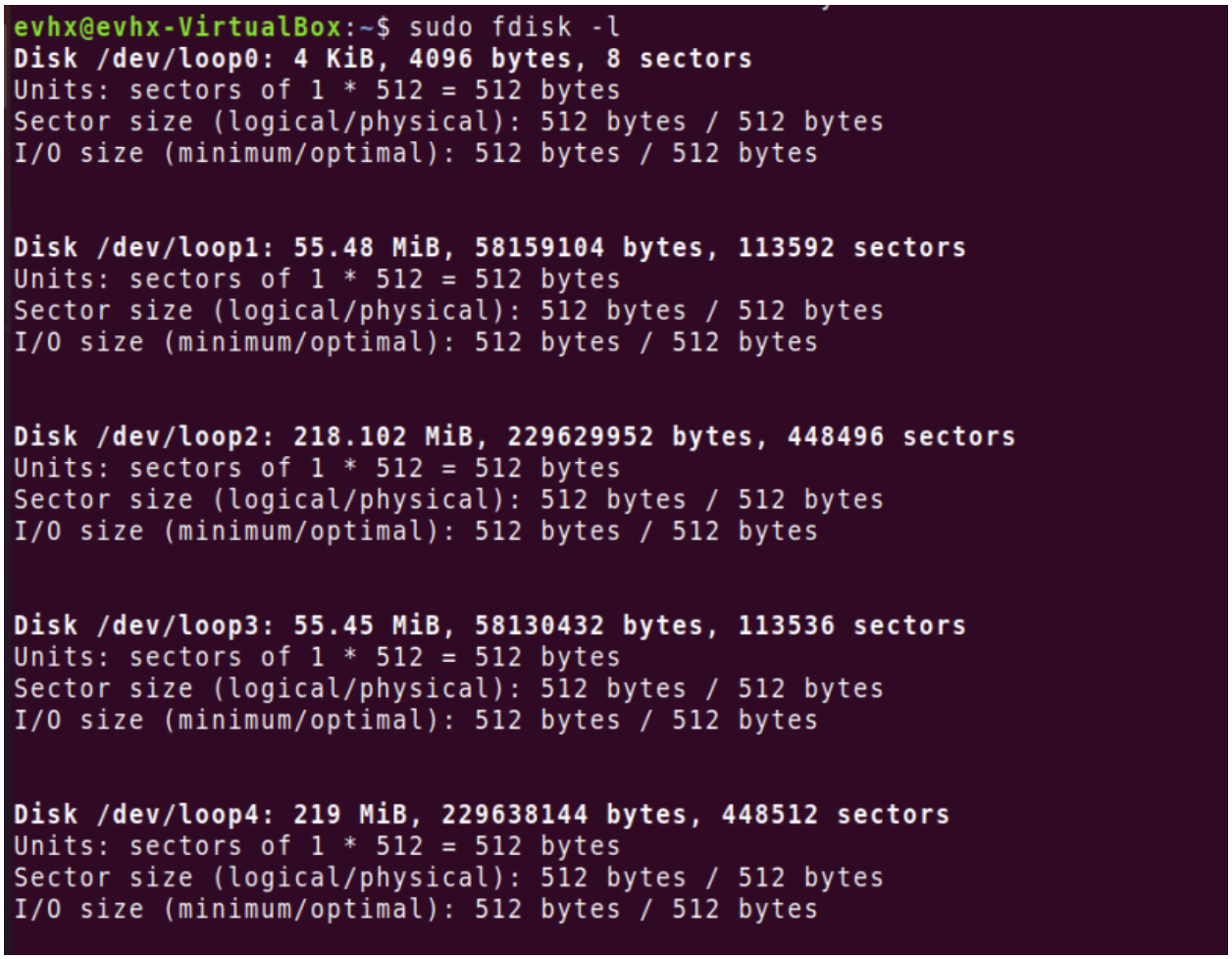
sudo fdisk -l

After checking the partitions, to choose your preferred storage disk location to use as the EXT4 filesystem, use the command:

sudo fdisk /dev/[storage disk location]

Commands:

sudo	(root privileges	)
fdisk	(program for creation and manipulation of partition tables	)
-l	(list information	)
/dev/[preferred location]	(storage disk location	)



```
Disk /dev/loop5: 50.98 MiB, 53432320 bytes, 104360 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/loop6: 32.32 MiB, 33878016 bytes, 66168 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/loop7: 65.1 MiB, 68259840 bytes, 133320 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/sda: 10 GiB, 10737418240 bytes, 20971520 sectors
Disk model: VBOX HARDDISK
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: 0xa215d258

Device     Boot   Start      End  Sectors  Size Id Type
/dev/sda1  *       2048  1050623  1048576  512M  b W95 FAT32
/dev/sda2       1052670 20969471 19916802  9.5G  5 Extended
/dev/sda5       1052672 20969471 19916800  9.5G 83 Linux
```

```
Disk /dev/loop8: 51.4 MiB, 53522432 bytes, 104536 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/loop9: 32.31 MiB, 33869824 bytes, 66152 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/loop10: 65.22 MiB, 68378624 bytes, 133552 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
```

After executing the command:

sudo fdisk /dev/[storage disk location]

Go through the list of commands to create a new partition:

Command-	m	(view list of available commands)
Command-	n	(create the new partition)
Partition type-	p	(primary partition)
Partition number-	1	(partition number)
First sector-	2048	(sector related to size of partition)
Last sector-	1048575	(sector related to size of partition)

To check if the partition was successfully created use the command:

Command-	p	(create the new partition)

```
evhx@evhx-VirtualBox:~$ sudo fdisk /dev/sda1

Welcome to fdisk (util-linux 2.34).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-1048575, default 2048): 2048
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-1048575, default 1048575): 104
8575

Created a new partition 1 of type 'Linux' and of size 511 MiB.

Command (m for help):
```

```
Command (m for help): p
Disk /dev/sda1: 512 MiB, 536870912 bytes, 1048576 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: 0x00000000

Device      Boot Start     End Sectors  Size Id Type
/dev/sda1p1       2048 1048575 1046528  511M 83 Linux
```

To make a new XFS file system, download XFS with the command:

sudo apt install xfsprogs

After the installation of xfsprogs is complete, create the XFS filesystem by using the command:

sudo mkfs.xfs /dev/[preferred location] or
sudo mkfs.xfs -f -L XFS -b size=1024 /dev/[preferred location]

If it turns out that the folder already contains a file system, just unmount it with the command:

sudo umount /dev/[preferred location]

Commands:

sudo	(root privileges	)
apt install	(retrieve installation	)
xfsprogs	(utility package for managing the XFS filesystem	)
mkfs.xfs	(to format the partition as XFS	)
/dev/[preferred location]	(preferred location for the XFS filesystem	)
umount	(unmount file system command	)
-f	(forces overwrite	)
-L XFS	(sets filename to 'XFS'	)
-b size=1024	(sets block size	)

```
evhx@evhx-VirtualBox:~$ sudo apt install xfsprogs
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm11
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libreadline5
Suggested packages:
  xfsdump attr quota
The following NEW packages will be installed:
  libreadline5 xfsprogs
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 873 kB of archives.
```

```
evhx@evhx-VirtualBox:~$ sudo umount /dev/sda1
evhx@evhx-VirtualBox:~$ sudo mkfs.xfs -f -L XFS -b size=1024 /dev/sda1
meta-data=/dev/sda1              isize=512    agcount=4, agsize=131072 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1
data     =                       bsize=1024   blocks=524288, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=1024   blocks=3527, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
evhx@evhx-VirtualBox:~$ sudo mkfs.xfs /dev/sda1
mkfs.xfs: /dev/sda1 appears to contain an existing filesystem (xfs).
```

To create a local mount point for the XFS file system(unless the desired location for the mount point is known), create a new directory using the command:

sudo mkdir /mnt/db

To mount the point to the directory use the command:

sudo mount /dev/[preferred location] /mnt/db

To make sure that the mount point is mounted correctly and see any partitions in the mount location, use the command:

sudo mount | grep /dev/[preferred location]

Commands:

sudo	(root privileges	)
mkdir	(create a directory	)
/mnt/db	(mount point	)
mount	(mount filesystem	)
/dev/[preferred location]	(folder created for the mount point location	)
grep	(search for text from a file or output	)
mount | grep /dev/[preferred location]	(gets the mount point from its location	)

Now a security group will be created to control access to the new filesystem by using the command:
sudo groupadd security_group
sudo groupadd [group_name]

Then the user on the machine named ‘evhx’ will be added to the security group previously made using the command:

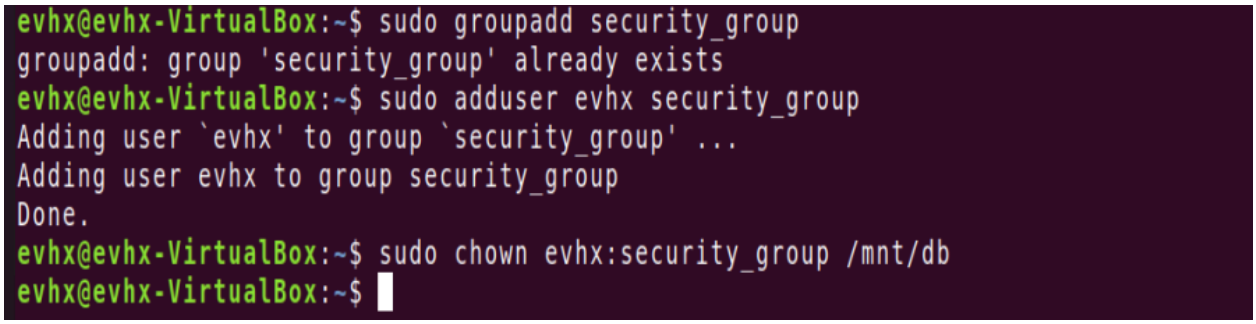
sudo adduser evhx security_group
sudo adduser [username] [group_name]

To give this user and their group ownership privileges for the file system, use the command:

sudo chown evhx:security_group /mnt/db
sudo chown [username]:[group_name] [mount point]

Commands:

sudo	(root privileges	)
groupadd	(add a group	)
adduser	(add a user	)
chown	(changes user/group ownership of a file or directory	)
/mnt/db	(mount location of the file system	)



Now lets create a new user named ‘will’ using the command:

sudo useradd will
sudo adduser [username]

The user ‘will’ will also be added to the security group previously made with the command:

sudo adduser will security_group
sudo adduser [username] [group_name]

To give the newly created user ‘will’ the ability to use sudo, use the command:

sudo usermod -aG sudo will
sudo usermod -aG sudo [username]

To check if will is in the security group and allowed access to sudo, use the command

groups will
groups [username]

Which displays:
will : will sudo security_group

Commands:

sudo	(root privileges	)
adduser	(add a user	)
usermod	(change properties of a user/calls program	)
-a	(add to the group	)
G	(specify the group	)
usermod -aG sudo [username]	(grant sudo privileges to a user	)
groups [username]	(shows what groups the user belongs in	)

```
evhx@evhx-VirtualBox:~$ sudo adduser will
Adding user `will' ...
Adding new group `will' (1004) ...
Adding new user `will' (1003) with group `will' ...
Creating home directory `/home/will' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for will
Enter the new value, or press ENTER for the default
        Full Name []: will
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
evhx@evhx-VirtualBox:~$ sudo adduser will security_group
Adding user `will' to group `security_group' ...
Adding user will to group security_group
Done.
evhx@evhx-VirtualBox:~$ sudo usermod -aG sudo will
evhx@evhx-VirtualBox:~$ groups will
will : will sudo security_group
```

To allow the previously created user named ‘will’ be able to mount and unmount the filesystem, its privileges to the mount point need to be changed, as previously shown with the ‘evhx’ user, using the command:

sudo chown will:security_group /mnt/db
sudo chown [username]:[group_name] [mount point]

Or to give these privileges solely to the user without the group, use the command:

sudo chown -R will /mnt/db
sudo chown -R [username] [mount point]

Those commands just give the user ‘will’ permissions over that specific mount point, but to only allow the user ‘will’ to be able to execute the ‘mount’ and ‘umount’ commands, we have to restrict the users shell with commands:

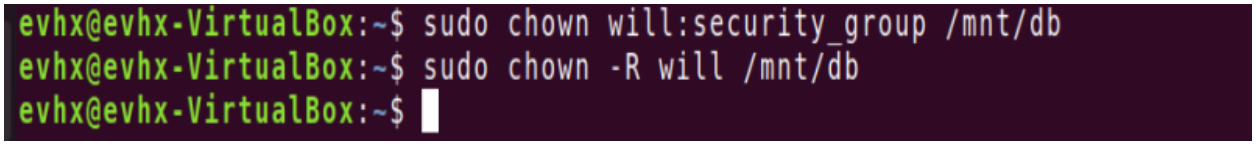
sudo usermod -s /bin/rbash will
sudo mkdir /home/will/programs

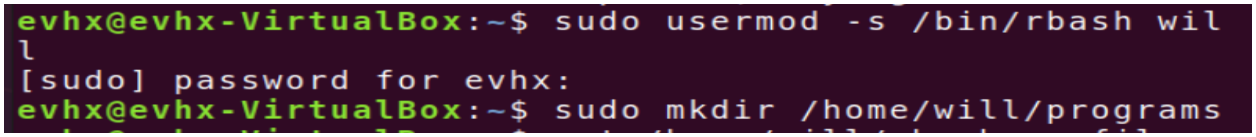
To limit the commands that can be used by the user ‘bill’, use the commands:

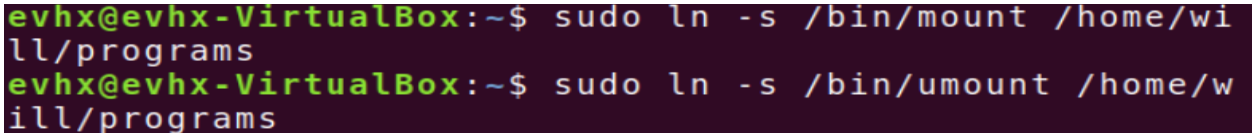
sudo ln -s /bin/mount /home/will/programs
sudo ln -s /bin/umount /home/will/programs

Commands:

sudo	(root privileges)
chown	(changes user/group ownership of a file or directory)
-R	(recursively)
usermod	(change properties of a user/calls program)
mkdir	(create a directory)
ln -s	(creating links between files (-s for symbolic)







To check if all these set permissions have worked accordingly, we can log in as the new created user.

The user ‘will’ was clearly able to mount and unmount the file system using the commands:

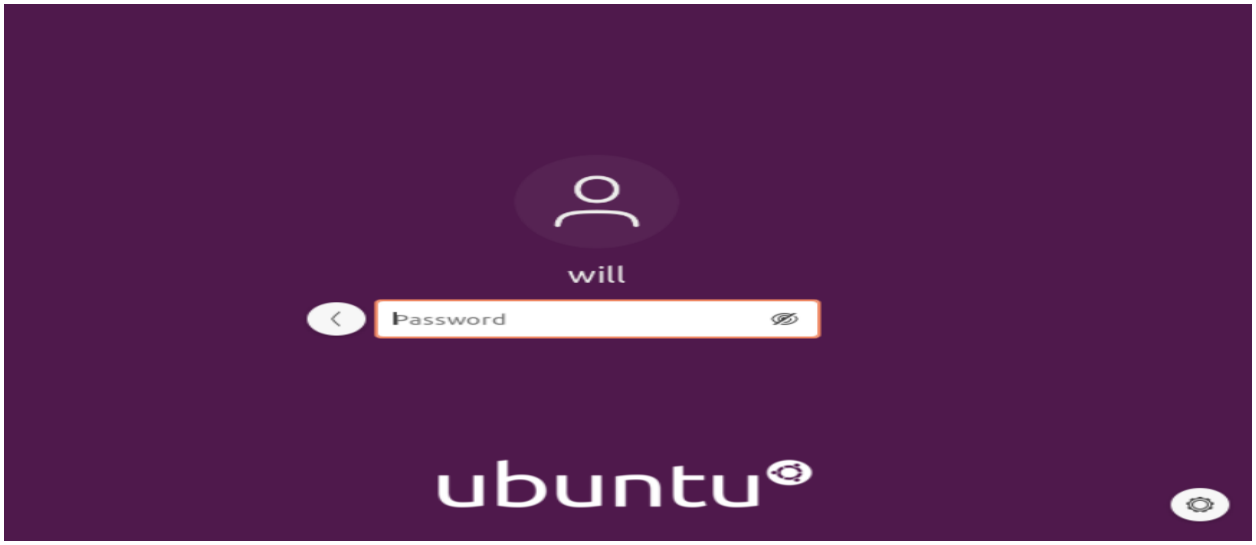
sudo umount /dev/sda1
sudo mount /dev/sda1 /mnt/db

But as shown below, there was not permission to create a text file or write into it, even when using sudo when using the commands:

cat > random.txt
sudo cat > random.txt

Commands:

sudo	(root privileges	)
umount	(unmount the file system	)
mount	(mount filesystem	)
cat >	(create a file + add text input	)




```
will@evhx-VirtualBox:~$ sudo umount /dev/sda1
will@evhx-VirtualBox:~$ sudo mount /dev/sda1 /mnt/db
will@evhx-VirtualBox:~$
```

```
will@evhx-VirtualBox:/mnt/db$ cat > random.txt
bash: random.txt: Permission denied
will@evhx-VirtualBox:/mnt/db$ sudo cat > random.txt
bash: random.txt: Permission denied
will@evhx-VirtualBox:/mnt/db$
```

Conclusion

A mounted EXT4 file system was created on the Ubuntu Virtual Machine, and some key features are shown were its structured data storage, naming, and security by only allowing certain users access rights. EXT4 is the latest of the EXT file systems, which is preferred for its actively-developed nature, compared to a file system like ReiserFS which had its development end in 2008. It is said that ReiserFS fell out of favor after Hans Reiser(ReiserFS's creator) was arrested in 2006 for the first-degree murder of his wife-- Nina Reiser.

Another good file system commonly in use is the JFS or Journaled File System, which is well suited in enterprise environments because of its advanced techniques that are good for very large file systems, which is different from ReiserFS that is better with smaller file systems. Regardless of what file system is in use, a good file system is always welcome for its provided help with the vast amount of files a computer holds.