

Adding storage

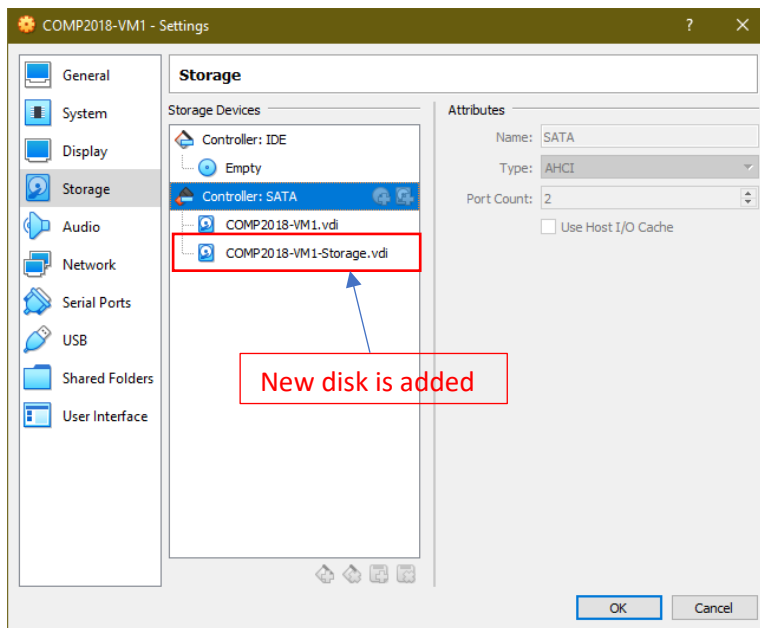
Step 1: Add a disk to your system

In VMware or VirtualBox you can add additional virtual disks to your system. Your VM needs to be powered off. Go and edit your VM settings:

VMware: VM > Settings

VirtualBox: Machine > Settings

For both VMware and VirtualBox the process of adding a new virtual disk is very similar. You select the controller, and you add a virtual disk. Follow the process for each software to add an additional disk to your system. This disk can be any size you like (I choose 5GB for demonstration purposes). The result in VirtualBox looks like the screenshot below.



Step 2: Find the logical name of the disk

There are many ways you can explore hardware in Linux. You can try the following commands to find the name of the new disk you have added:

```
sudo lshw -C disk
```

You can also use this command to see what partition structure on the disk currently looks like (hint, new disk is not yet partitioned!).

```
lsblk
```

the command `lsblk` also shows all the partitions on the disks so you don't delete partitions by mistake or create partitions on wrong disk. Removing the wrong partition or formatting the wrong disk can cause data loss and potentially damage to your Linux system. So please make sure you are working on the correct disk and partition.

In my system the new disk seems to show up as `/dev/sdb` (this may vary in your system).

```
ali@ers20059995:~$ sudo lshw -C disk
...
<output shortened for clarity>
...
*-disk
   description: ATA Disk
   product: VBOX HARDDISK
   vendor: VirtualBox
   physical id: 0.0.0
   bus info: scsi@3:0.0.0
   logical name: /dev/sdb
   version: 1.0
   serial: VB629488bf-f526c9e0
   size: 5GiB (5368MB)
   configuration: ansiversion=5 logicalsectorsize=512 sectorsize=512
```

Step 3: Create a partition on the disk

For CLI in most Linux and Unix systems fdisk is the included utility. There may be other utilities included with your system depending on the distro, but here we will use fdisk since this is widely available on most systems.

```
ali@ers20059995:~$ sudo fdisk /dev/sdb

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.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x958ef744.

Command (m for help):
```

You can type “m” to see all available options. For example you can try using “F” option to list free unpartitioned space on the disk.

You can go ahead and use option “n” to add a new partition. You will be prompted to select from various choices that you can make. Since this is a dedicated disk to our “datastore” we can just pick the default options in the following order. Note that I picked mostly the default options here due to our very happy circumstances on our lab system. You may need to choose other options depending on the situation.

```
Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (1-4, default 1): 1
First sector (2048-10485759, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-10485759, default 10485759):

Created a new partition 1 of type 'Linux' and of size 5 GiB.
```

Now that you have defined the new partition, the changes need to be written to the disk. The option “w” does exactly that and will write out changes to the disk. The fdisk utility will also exit to the terminal.

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

Use the lsblk command again and observe the changes to your disk. You can also note the partition name.

Step 4: formatting the partition

This step is short and easy. Just use the mkfs utility as the following. You can read the manpages for mkfs.

```
ali@ers20059995:~$ sudo mkfs -t ext4 /dev/sdb1
mke2fs 1.45.5 (07-Jan-2020)
Creating filesystem with 1310464 4k blocks and 327680 inodes
Filesystem UUID: 1797bef0-6378-4c0a-b412-c96ddc3045ce
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
ali@ers20059995:~$
```

Step 5: mounting the storage volume

We need to create a “mount point”. Mount point is simply an empty directory anywhere in your system that you have access to. You can create mountpoints depending on your needs and your desire.

For this example, I will create a mount point in my home directory, in this case /home/ali/backups.

```
ali@ers20059995:~$ mkdir backups
```

Once we have our desired location as mount point, we can manually mount the drive.

```
ali@ers20059995:~$ sudo mount /dev/sdb1 ./backups
```

You can use lsblk to verify your new partition is now mounted to your desired location.

```
ali@ers20059995:~$ lsblk
...
<Output shortened for clarity>
...
sdb                8:16    0    5G    0 disk
└─sdb1             8:17    0    5G    0 part /home/ali/backups
```

Use ls -la to see what the permissions for this directory look like. If no permissions are set, by default the volume belongs to root. I like to change the permissions so this device is owned by my user:

```
ali@ers20059995:~$ sudo chown ali:ali ./backups && sudo chmod 755 ./backups
```

Let's create some files on this disk:

```
ali@ers20059995:~$ cd backups
ali@ers20059995:~/backups$ mkdir tesdir
ali@ers20059995:~/backups$ touch tesdir/testfile
```

Use ls commands to verify that your files do exist on this drive. Unmount the drive to observe how the mount point looks when no volume is mounted.

```
ali@ers20059995:~/backups$ cd ..
ali@ers20059995:~$ sudo umount /dev/sdb1
```

```
ali@ers20059995:~$
```

Verify the results using lsblk.

Step 6: Automatic Mounting of the storage volume at boot

According to the manpages, the fstab file “contains descriptive information about the filesystems the system can mount.” Read the man pages for fstab before continuing.

If you read the man pages for fstab, you will find that each line describes a filesystem, and each filesystem has several “fields” that are separated by spaces or tab. You can make note of what each field should contain. Using this information, you can create a line with all the appropriate fields and options that you will need for your new storage volume. In my system, the fstab entry will look like this:

```
/dev/sdb1      /home/ali/backups      ext4      defaults      0      2
```

Add this entry to your fstab file. In this case I use nano with “B” and “w” options (look those up if you like).

Make sure NOT to modify other filesystem entries, or you can potentially break your system!

```
ali@ers20059995:~$ sudo nano -Bw /etc/fstab
```

Once the fstab entry is saved you can either reboot your Linux machine to observe the drive is being mounted automatically, or just use the following command:

```
ali@ers20059995:~$ sudo mount -a
```

Verify that your file system is mounted and is writable by your user. Additionally, you can copy some files into your mount point and use the df command to verify how much storage space you have left on each file systems.

```
ali@ers20059995:~$ cp -r lab2 ./backups
ali@ers20059995:~$ cd backups/
ali@ers20059995:~/backups$ ls -la
total 68
drwxr-xr-x 6 ali ali 4096 Sep 29 23:15 .
drwxr-xr-x 7 ali ali 4096 Sep 29 22:18 ..
drwxr-xr-x 2 ali ali 36864 Sep 29 23:15 lab2
drwx----- 2 root root 16384 Sep 29 22:13 lost+found
drwxrwxr-x 2 ali ali 4096 Sep 29 22:56 tesdir
drwxrwxr-x 2 ali ali 4096 Sep 29 23:14 test2
ali@ers20059995:~/backups$ df -h
Filesystem                Size      Used Avail Use% Mounted on
udev                      950M         0   950M   0% /dev
tmpfs                     199M       1.1M   198M   1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv 15G       5.2G   8.8G  38% /
tmpfs                     994M         0   994M   0% /dev/shm
tmpfs                     5.0M         0    5.0M   0% /run/lock
tmpfs                     994M         0   994M   0% /sys/fs/cgroup
/dev/loop1                 68M        68M         0 100% /snap/lxd/21545
/dev/loop3                 62M        62M         0 100% /snap/core20/1081
/dev/loop2                 71M        71M         0 100% /snap/lxd/21029
/dev/sda2                  976M      299M   610M  33% /boot
/dev/loop0                 56M        56M         0 100% /snap/core18/2128
/dev/loop4                 33M        33M         0 100% /snap/snapd/13170
/dev/loop5                 33M        33M         0 100% /snap/snapd/12883
tmpfs                     199M         0   199M   0% /run/user/1000
/dev/sdb1                  4.9G       21M   4.6G   1% /home/ali/backups
```

Assignment Submission:

Please submit files containing the following to Blackboard as attachments:

1. Screenshot of the output of `lsblk` command showing your device partition layout
2. Screenshot of the output of `df -h` command showing the current mounting point for your storage volume
3. A copy of your `/etc/fstab` file showing that your storage volume is configured to be mounted on startup.