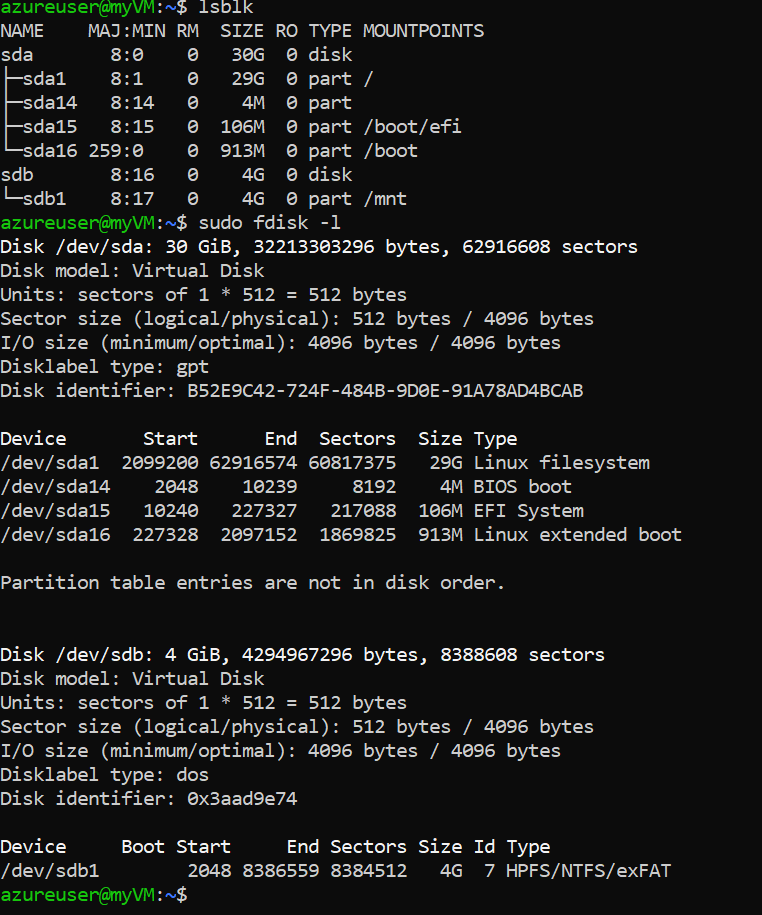
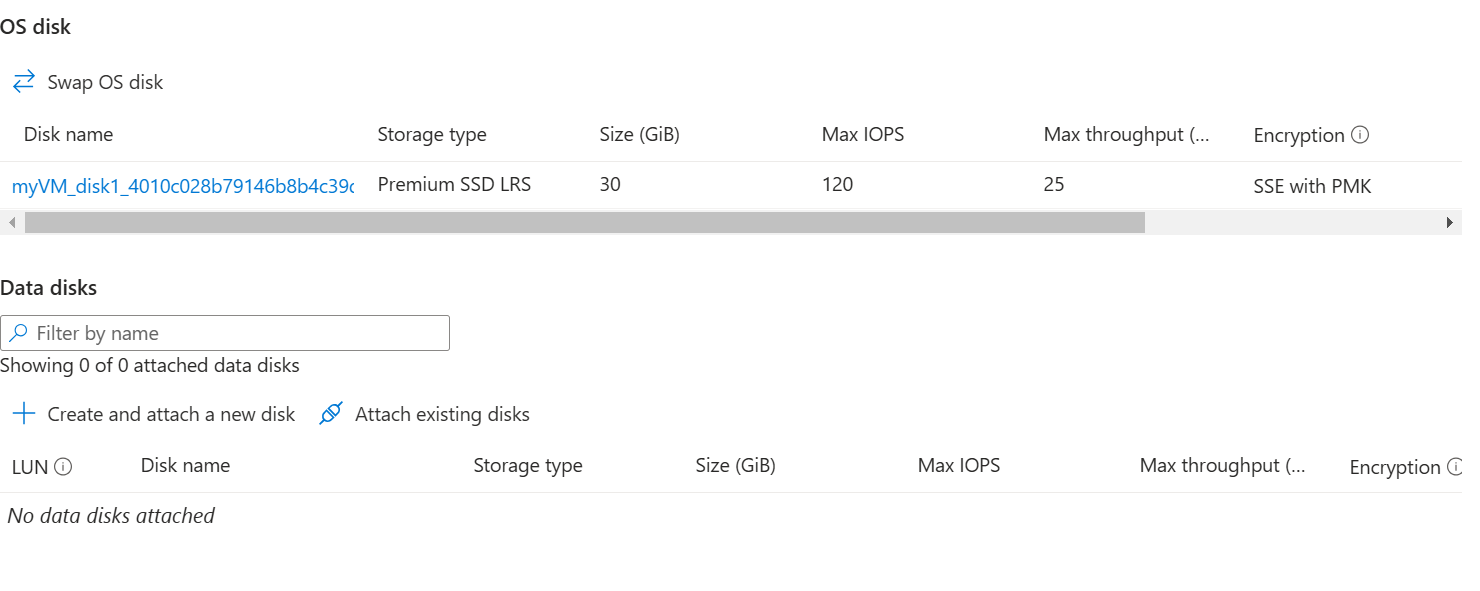
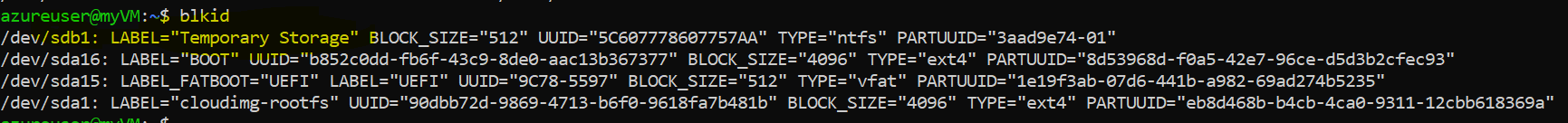
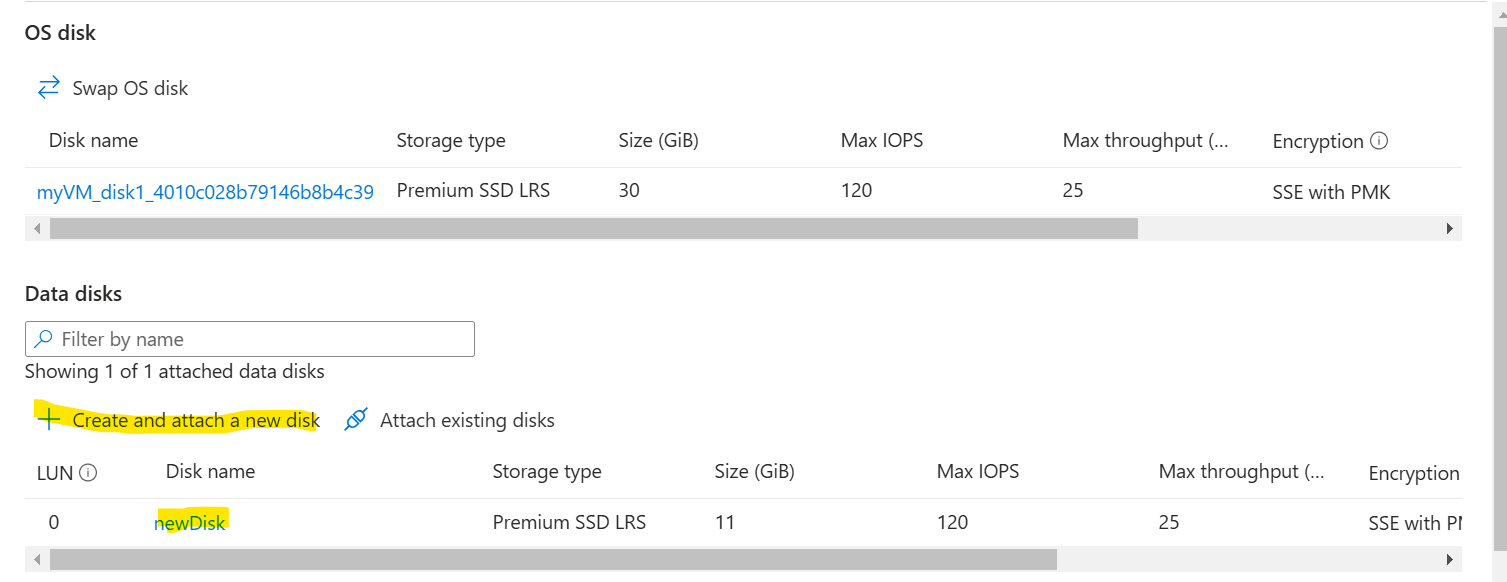
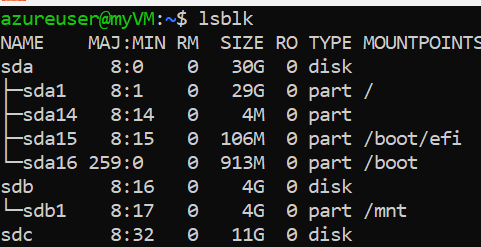
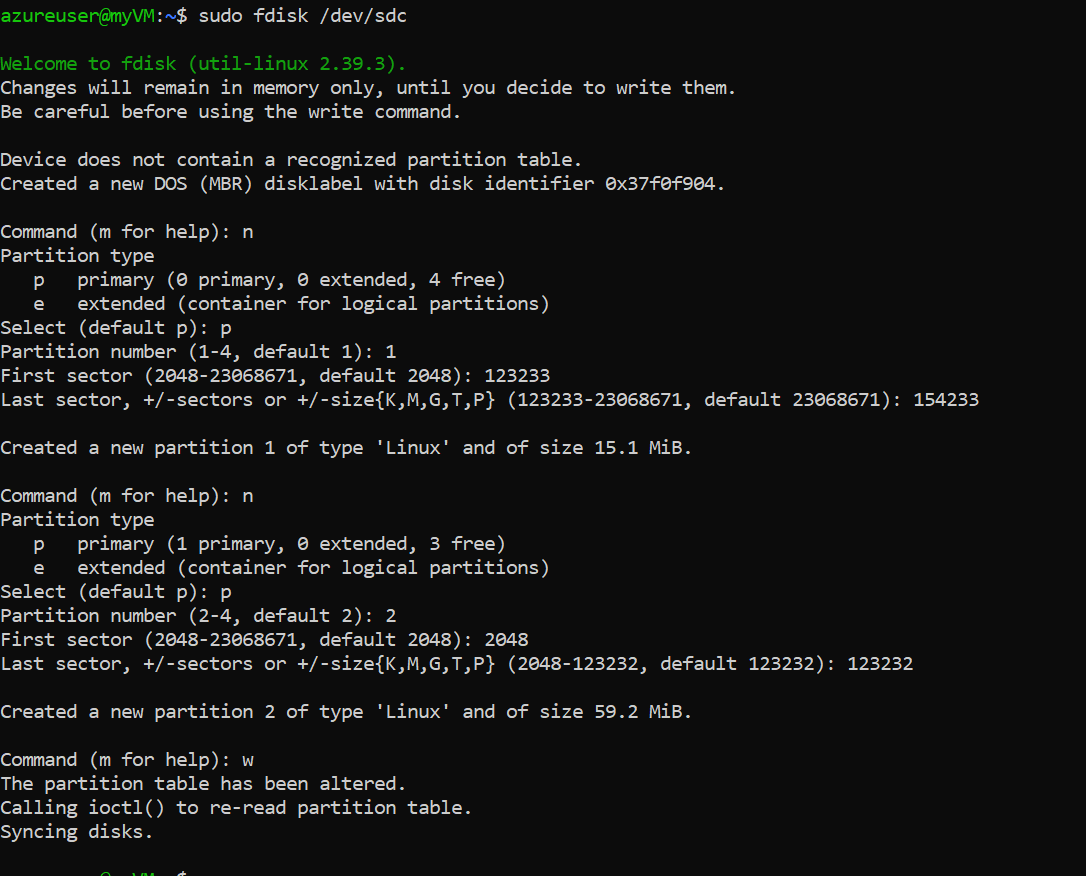
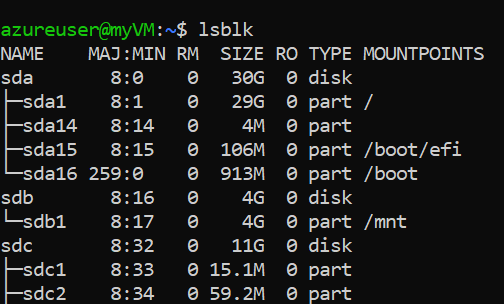
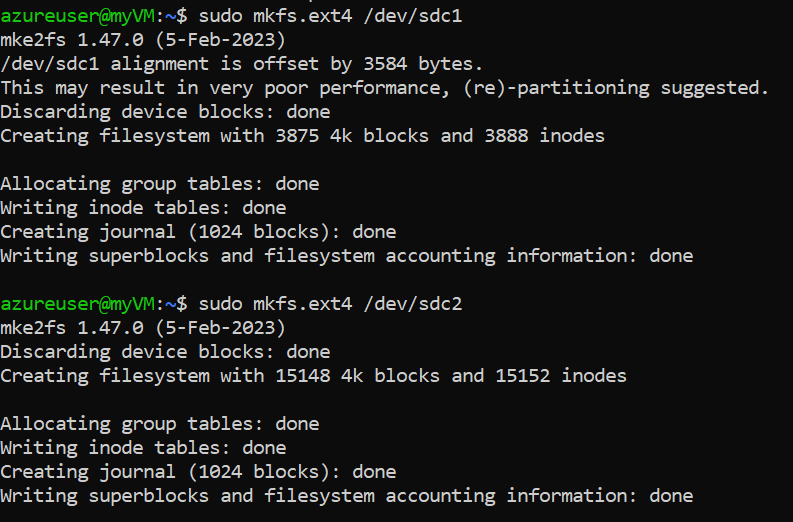
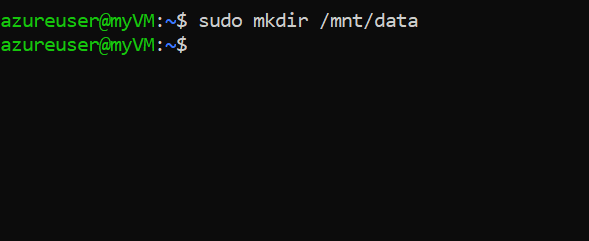
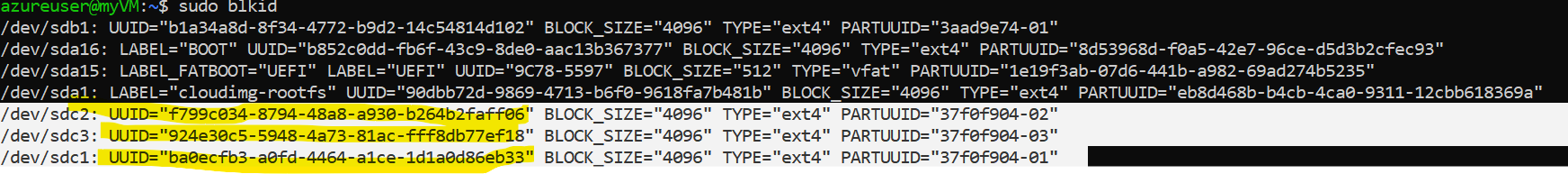
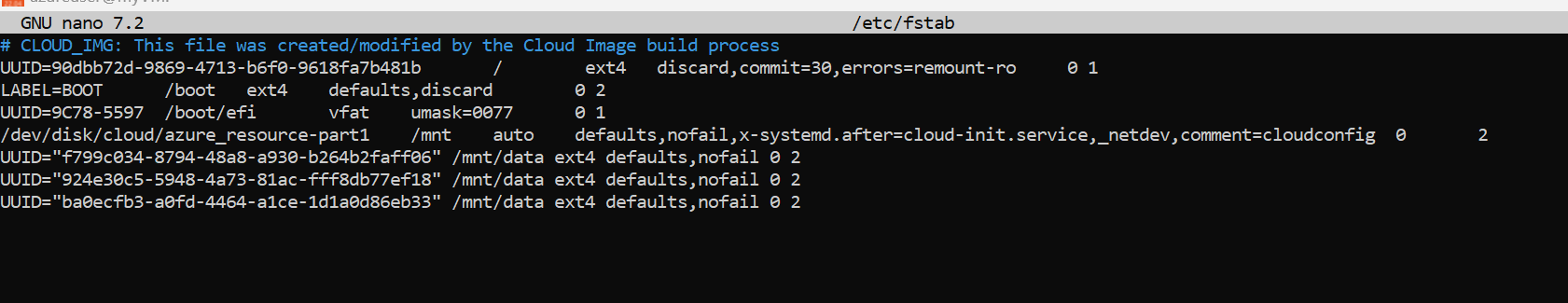
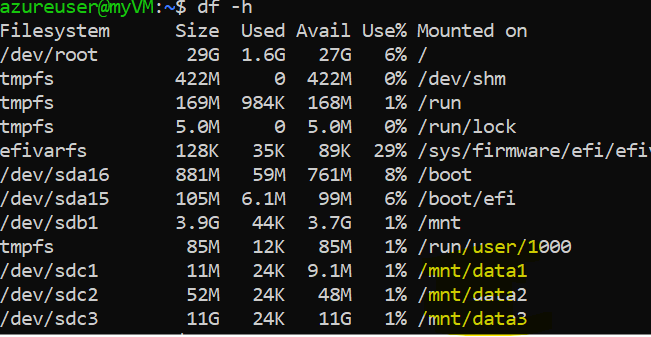
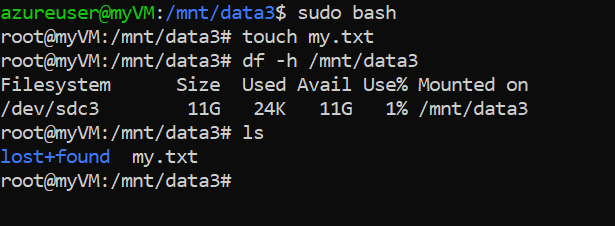
**Attach and format Linux Disk**

1. Connect to Azure VM (SOP is in place). Then rub **lsblk** to check how many block devices (disks) are already connected to the VM. And then run **sudo fdisk -I** command to check the exact partitions. 
2. See the sda shows that the disk size of 30GB is there, which is evident from the Azure portal as well. 
3. The sdb (4GB) is a temporary storage automatically created by the VM at the time of creation. Use **blkid** to verify the same. 
4. Now create and attach disk to the VM from portal. 
5. As soon as you do that, it starts showing up in the VM. 
6. Now, you can create partition inside the VM using below commands,
   1. sudo fdisk /dev/sdc
   2. n 🡪 new partion
   3. give number 1,2 (sdc-1, sdc-2)
   4. w 🡪 write (that means confirm) 
7. Now you can see the sectors have been created. 
8. Now you have to add file system to the partitions using the command, **sudo mkfs.ext4 /dev/sdc1** ****
9. Create a mount point where you will mount this disk,
   1. **sudo mkdir /mnt/data1**
   2. **sudo mkdir mnt/data2**
   3. **sudo mkdir mnt/data3**
   4. ****
10. Mount the disk to the directory
    1. **sudo mount /dev/sdc1 /mnt/data1**
    2. **sudo mount /dev/sdc2 /mnt/data2**
    3. **sudo mount /dev/sdc3 /mnt/data3** ****
11. Now we need to make the disk, persistent. Which means whenever the VM is rebooted, we shouldn’t need to do all these steps and the disk should automatically get mounted.
12. Using the **blkid** command, get the UUID of the blocks. 
13. After that we need to open the file fstab, **sudo nano /etc/fstab,** and add below lines in the file,
    1. UUID="f799c034-8794-48a8-a930-b264b2faff06" /mnt/data ext4 defaults,nofail 0 2
    2. UUID="924e30c5-5948-4a73-81ac-fff8db77ef18" /mnt/data ext4 defaults,nofail 0 2
    3. UUID="ba0ecfb3-a0fd-4464-a1ce-1d1a0d86eb33" /mnt/data ext4 defaults,nofail 0 2 
14. df -h, and you can see the partitions with file system are now mounted on the directories. 
15. Now you can start using them. 

**1. Creating Partitions**

**Why?**

* A disk is raw storage by default, which means the operating system cannot use it directly.
* Partitions divide the disk into logical segments. This helps in:
  + Organizing data (e.g., separating system files, user data, or application data).
  + Assigning different filesystems or mount points to different sections of the disk.
  + Managing disk space efficiently (e.g., allocating specific space for logs, databases, etc.).

**What happens if you skip it?**

* You can still use the whole disk directly, but you lose flexibility and may face challenges if you want to resize or reconfigure the disk later.

**2. Adding a Filesystem**

**Why?**

* A filesystem is required to organize and store files on the disk in a way that the operating system can understand.
* Different filesystems (e.g., ext4, xfs, ntfs) are optimized for different use cases (e.g., performance, compatibility, journaling).

**What happens if you skip it?**

* The disk or partition remains unusable. The OS cannot read or write data to it.

**3. Creating Mount Points (Directories)**

**Why?**

* In Linux, storage devices are accessed through the filesystem tree. Mount points are directories where these devices are "attached" or integrated into the tree.
* Mount points provide a way to access the data on the disk or partition.
* Each disk or partition must have a unique mount point to avoid conflicts.

**What happens if you skip it?**

* You won't be able to access the partition’s data. Even if the filesystem is created, the partition won't appear anywhere in the directory structure.

**4. Mounting the Partition**

**Why?**

* Mounting connects the disk/partition to a directory so that the operating system and users can access its data.
* It allows the system to use the storage for files, directories, or other purposes.

**What happens if you skip it?**

* The disk/partition won't be usable in the Linux filesystem tree. Even if it's partitioned and formatted, the data won't be accessible.

**5. Adding to /etc/fstab**

**Why?**

* Ensures the disk/partition is automatically mounted on boot.
* Provides consistency, so you don't have to mount it manually every time the system restarts.

**What happens if you skip it?**

* You'll need to manually mount the partition every time the system is rebooted.

**Overall Importance**

These steps are crucial for:

1. **Data Organization**: Properly managing and organizing data on disks.
2. **Flexibility**: Allowing different configurations for different parts of the storage.
3. **Stability**: Ensuring that disks/partitions are mounted consistently and correctly.
4. **Performance**: Choosing the right filesystem and configuration for your use case.

**Analogy**

Think of a disk as a plot of land:

* **Partitioning**: Dividing the land into sections for different purposes (e.g., house, garden, parking).
* **Filesystem**: Creating infrastructure on each section (e.g., roads, plumbing).
* **Mount Points**: Setting up gates/doors to access each section.
* **Mounting**: Attaching the gate/door to the infrastructure so it’s usable.
* **/etc/fstab**: Automating the opening of these gates on boot.

Skipping these steps often results in unorganized storage, inefficiency, or outright inaccessibility of the disk. Let me know if you'd like further clarification! 😊