NAME: Nikhil Dhavale

ROLL NO: 17

**Experiment No. 3**

**Implementation of Raid levels in Ubuntu**

**What is Raid?**

RAID is a Redundant Array of Inexpensive disks, but nowadays it is called Redundant Array of Independent drives. Earlier it is used to be very costly to buy even a smaller size of disk, but nowadays we can buy a large size of disk with the same amount like before. Raid is just a collection of disks in a pool to become a logical volume.

Raid contains groups or sets or Arrays. A combine of drivers make a group of disks to form a RAID Array or RAID set. It can be a minimum of 2 number of disk connected to a raid controller and make a logical volume or more drives can be in a group. Only one Raid level can be applied in a group of disks. Raid are used when we need excellent performance. According to our selected raid level, performance will differ. Saving our data by fault tolerance & high availability.

1. **There are three ways to create a RAID:**
   1. **Software-RAID:** Where the RAID is created by software.
   2. **Hardware-RAID:** A special controller used to build RAID. Hardware RAID is generallyfaster, and does not place load on the CPU, and hardware RAID can be used with any OS
   3. **FakeRAID:** Since RAID hardware is very expensive, many motherboard manufacturersuse multi-channel controllers with special BIOS features to perform RAID. This is a form of software RAID using special drivers, and it is not necessarily faster than true software

RAID

1. Files stored on arrays may be striped across multiple spindles. Since a high capacity is available due to the availability of more disks, it is possible to create redundancy within the system, so that if a disk fails the contents of a file may be reconstructed from the redundant information. This off course leads to a penalty in capacity (when storing redundant information) and in bandwidth (to update the disk). Four main techniques are available to overcome the lack of reliability of arrays:

Mirroring or shadowing of the contents of disk, which can be a capacity

kill approach to the problem. Each disk within the array is mirrored and a write operation performs a write on two disks - resulting in a 100% capacity overhead. Reads to disks may however be optimised. This solution is aimed at high bandwidth, high availability environments.

Horizontal Hamming Codes: A special means to reconstruct information using an error correction encoding technique. This may be an overkill, as it is complex to compute over a number of disks.

Parity and Reed-Soloman Codes: Also an error correction coding mechanism. Parity may be computed in a number of ways, either horizontally across disks or through the use of an interleaved parity block. Parity information also has to be stored on disk, leading to a 33% capacity cost for parity. Use of wider arrays reduces the capacity cost, but leads to a decrease

in the expected availability and increased reconstruction times. This approach is generally aimed at high bandwidth scientific applications (such as image processing).

Failure Prediction: There is no capacity overhead in this technique, though it is controversial in nature, as its use cannot be justified if all errors or failures can be forecast correctly.

**4)Installation and partition in ubuntu**

**Requirements**

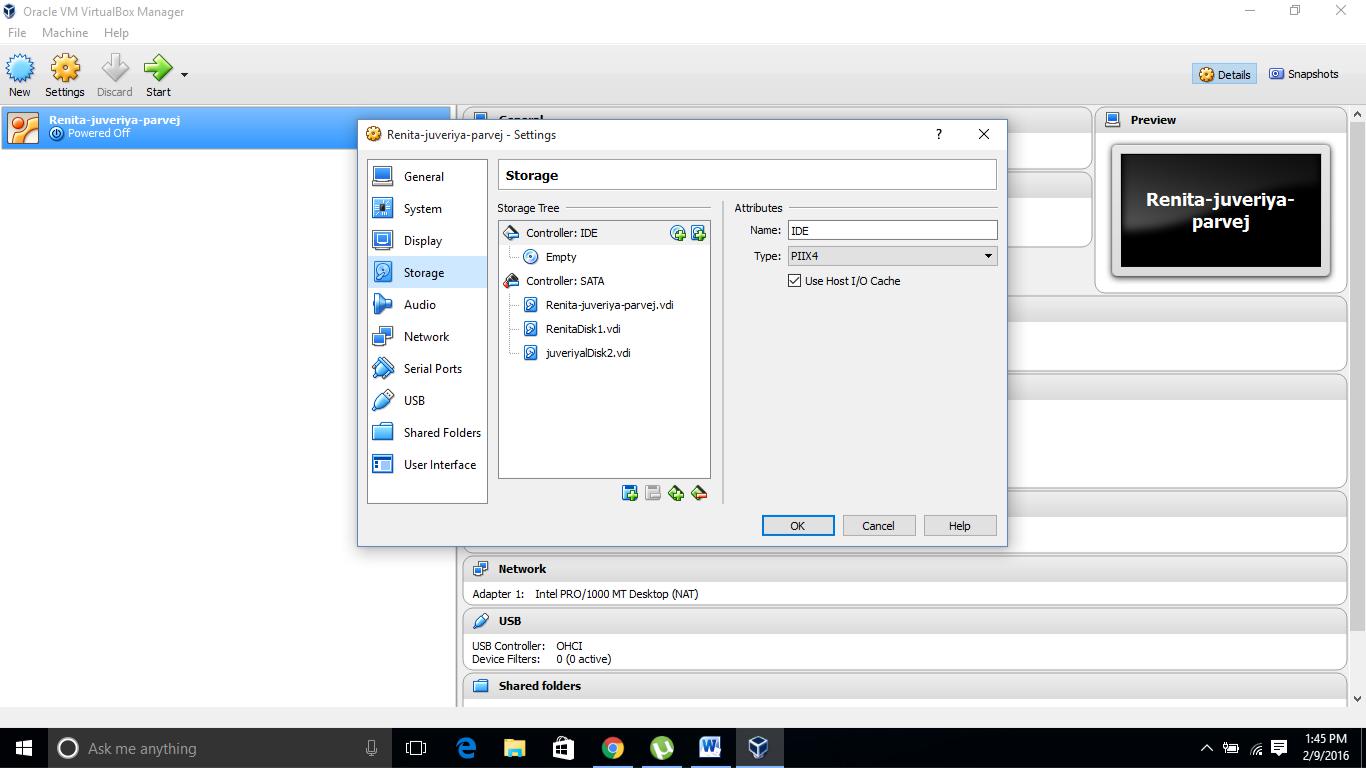
1)If you're building a server, the server install ISO includes the necessary options.

2)If you're building a desktop then you need the "Alternate" install ISO for Ubuntu. Read Getting Ubuntu Alternate Install disk and How to do a Ubuntu Alternate Install How to Burn an ISO

Enough drives to meet the requirements of the RAID.

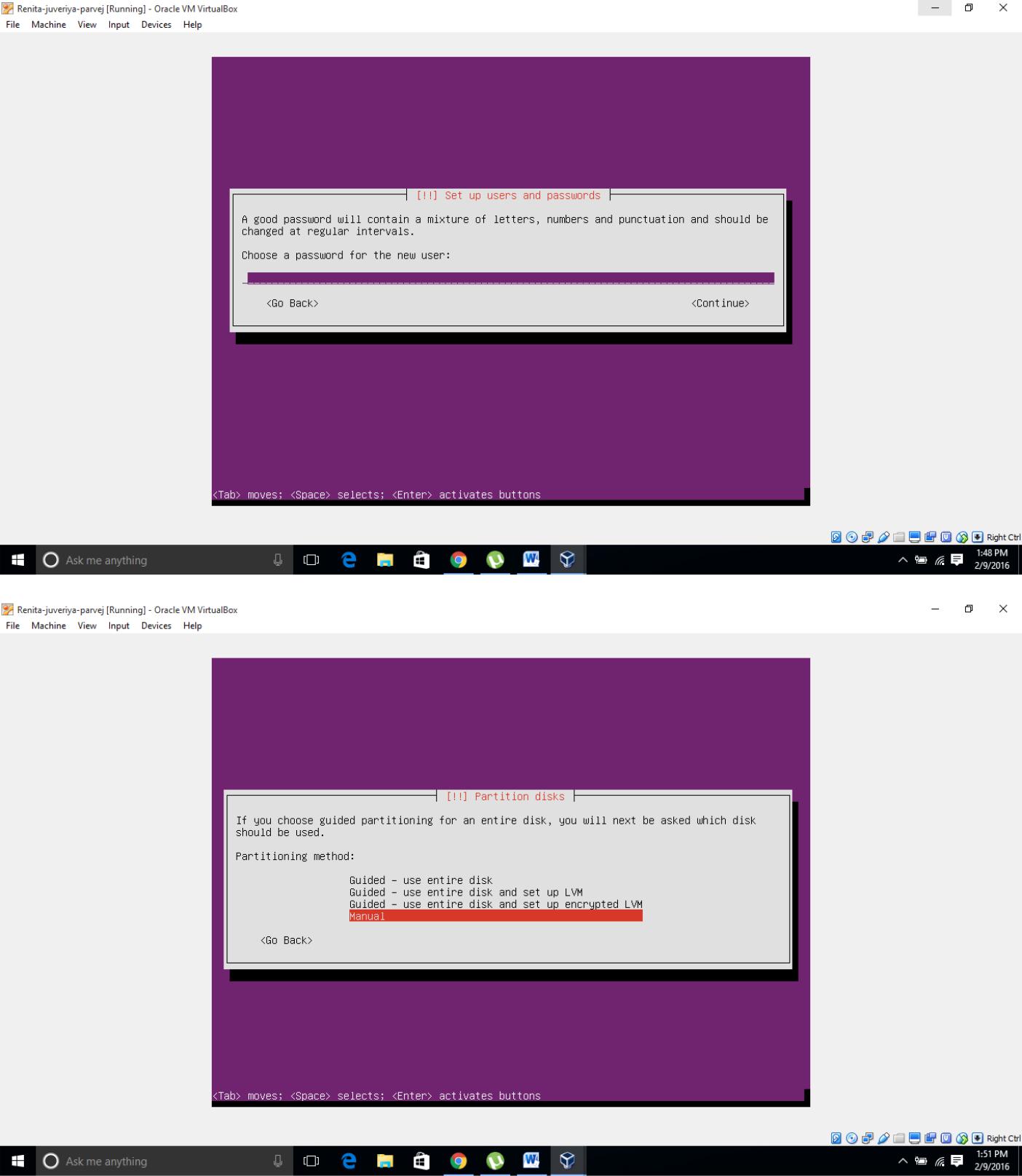
3) As we are using raid levels on ubuntu server

4)As we are using virtual machine to avoid any kind of disk failure or data loss

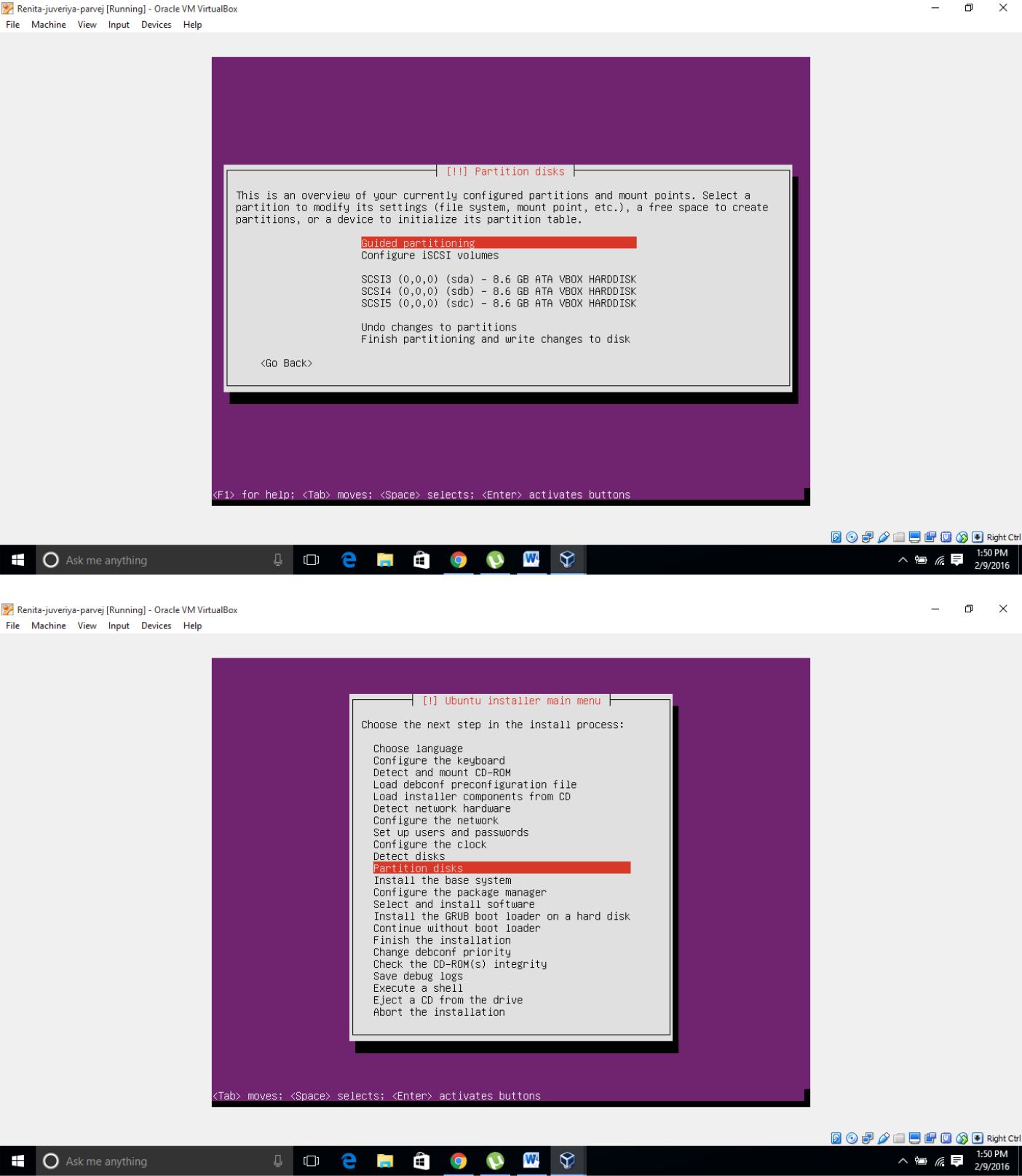


5) So in virtual machine select ubuntu server as Operating system and assign dynamic or select the hard drive space you want .

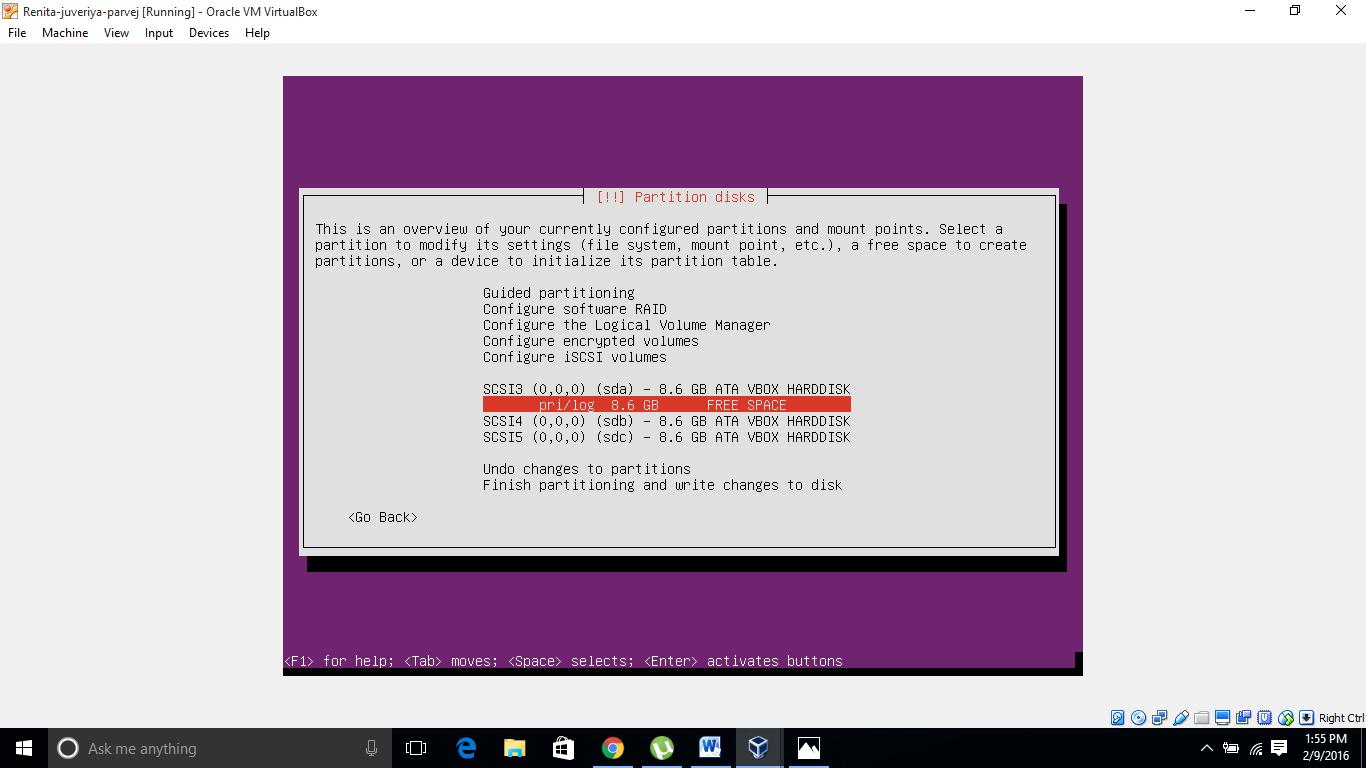
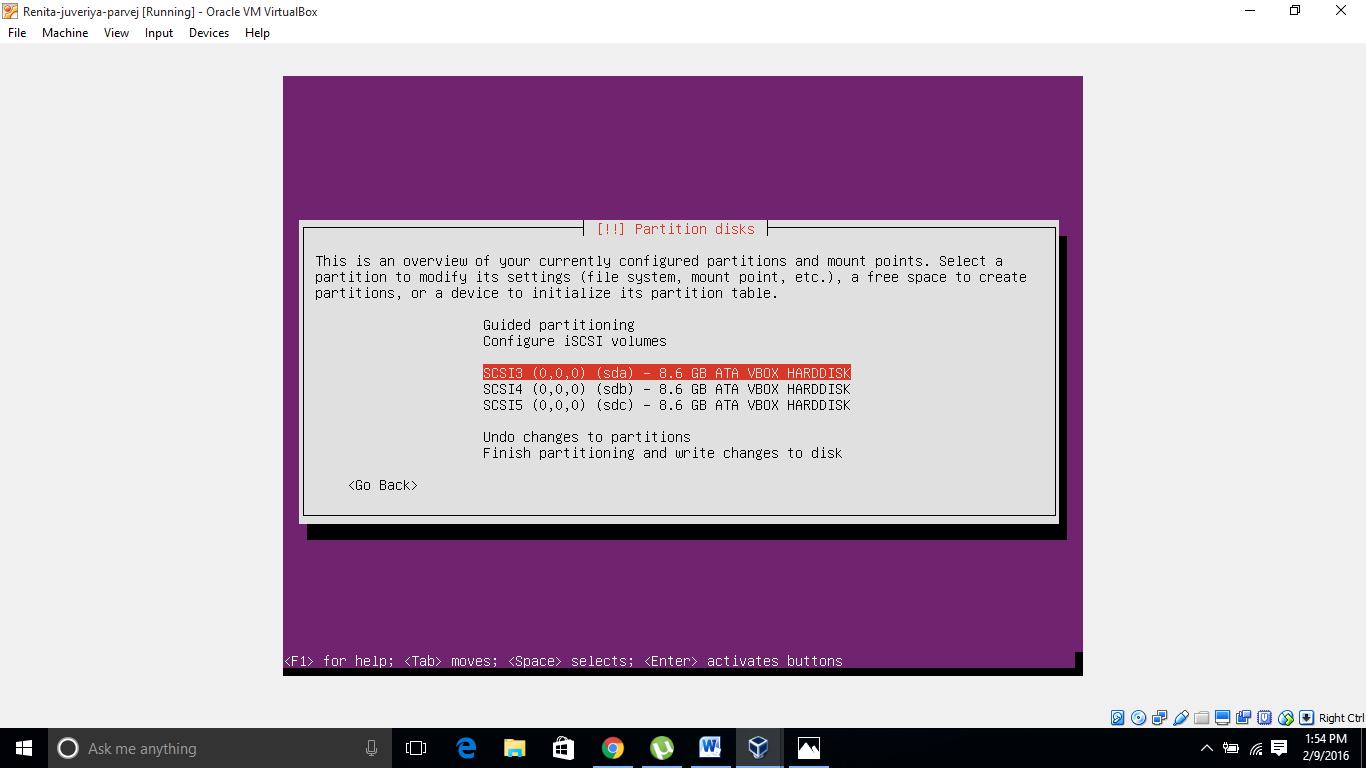
6)start the Virtual machine and while ubuntu server starts select the manual option as to create the raid levels in ubuntu.

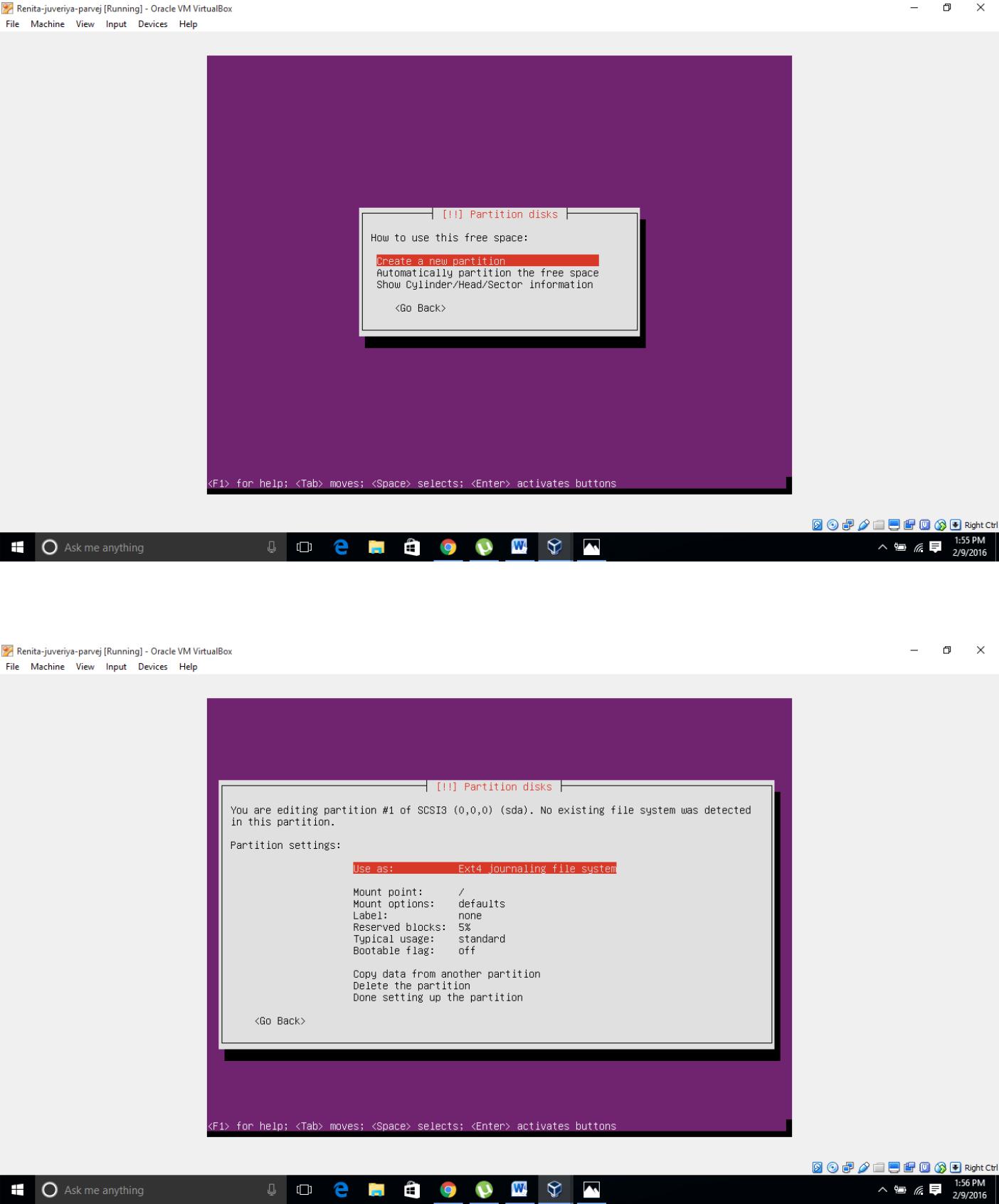


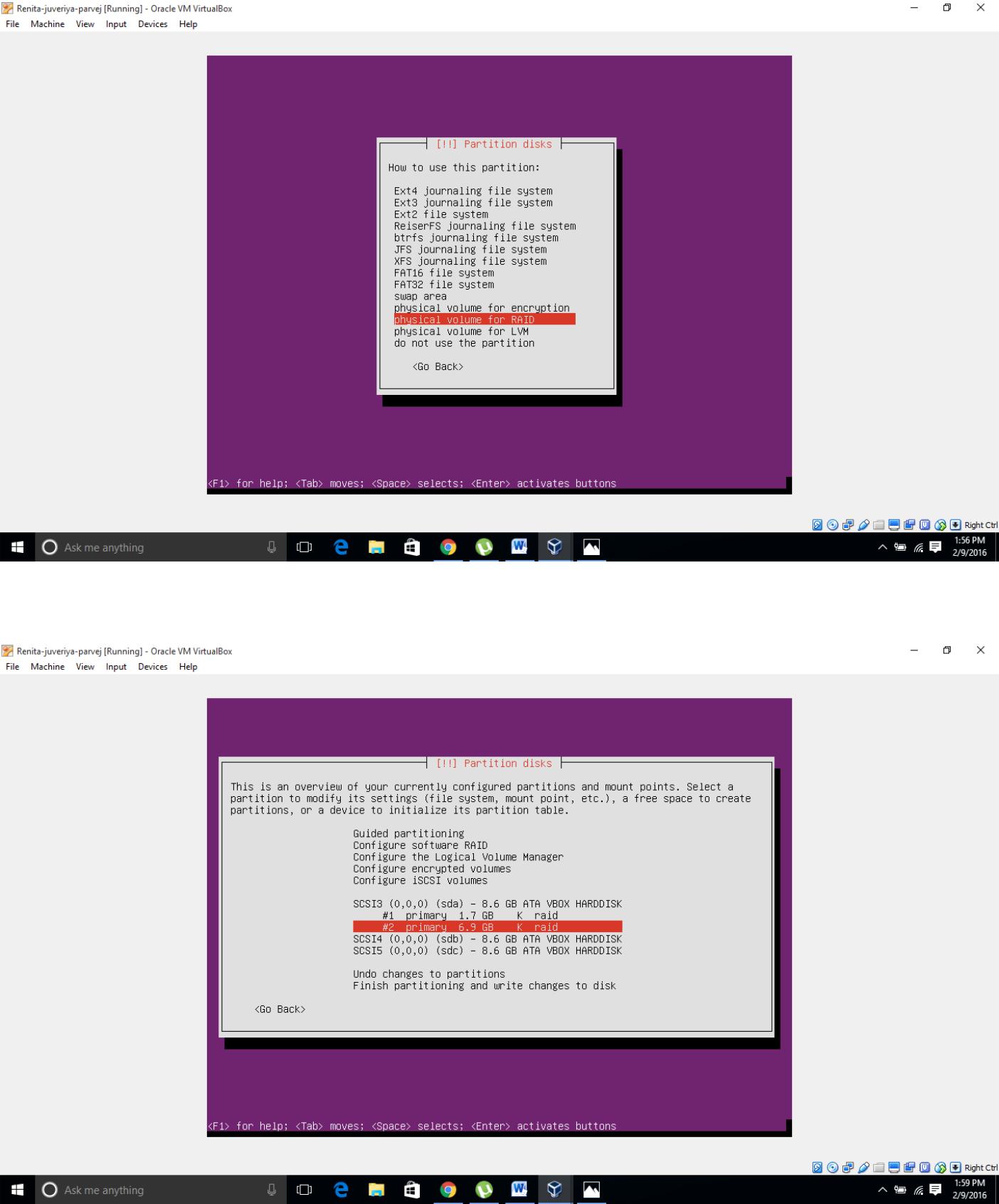
7) Select your hard drive, and agree to "Create a new empty partition table on this device



8) Select the "FREE SPACE" on the 1st drive then select "automatically partition the free space.

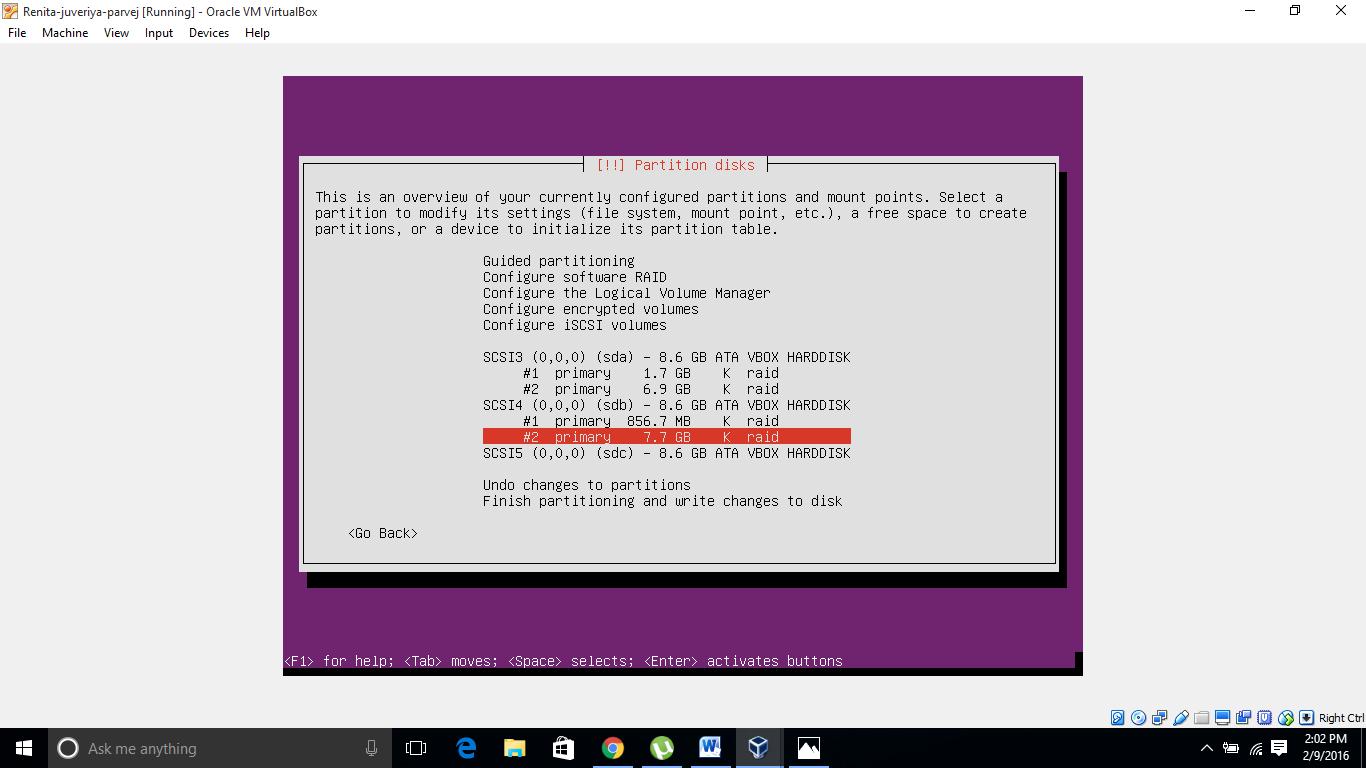




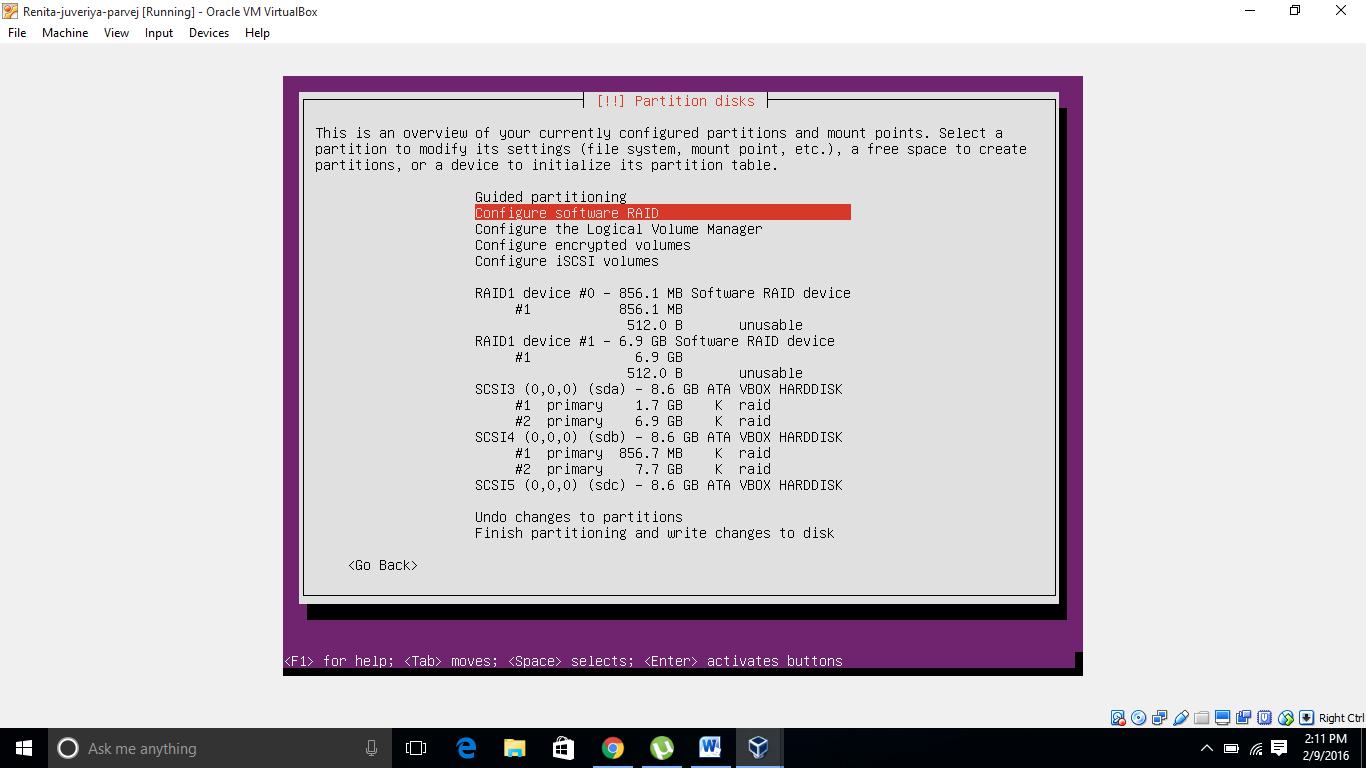
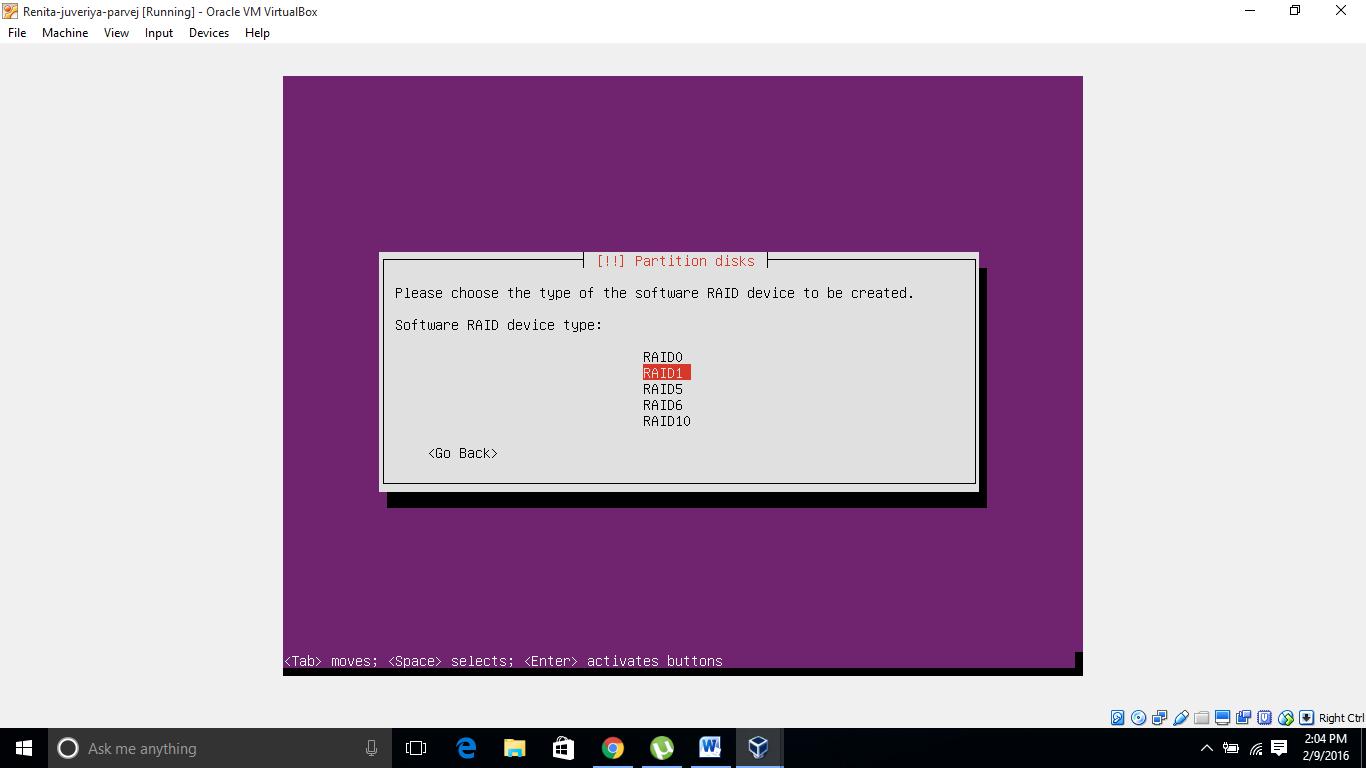


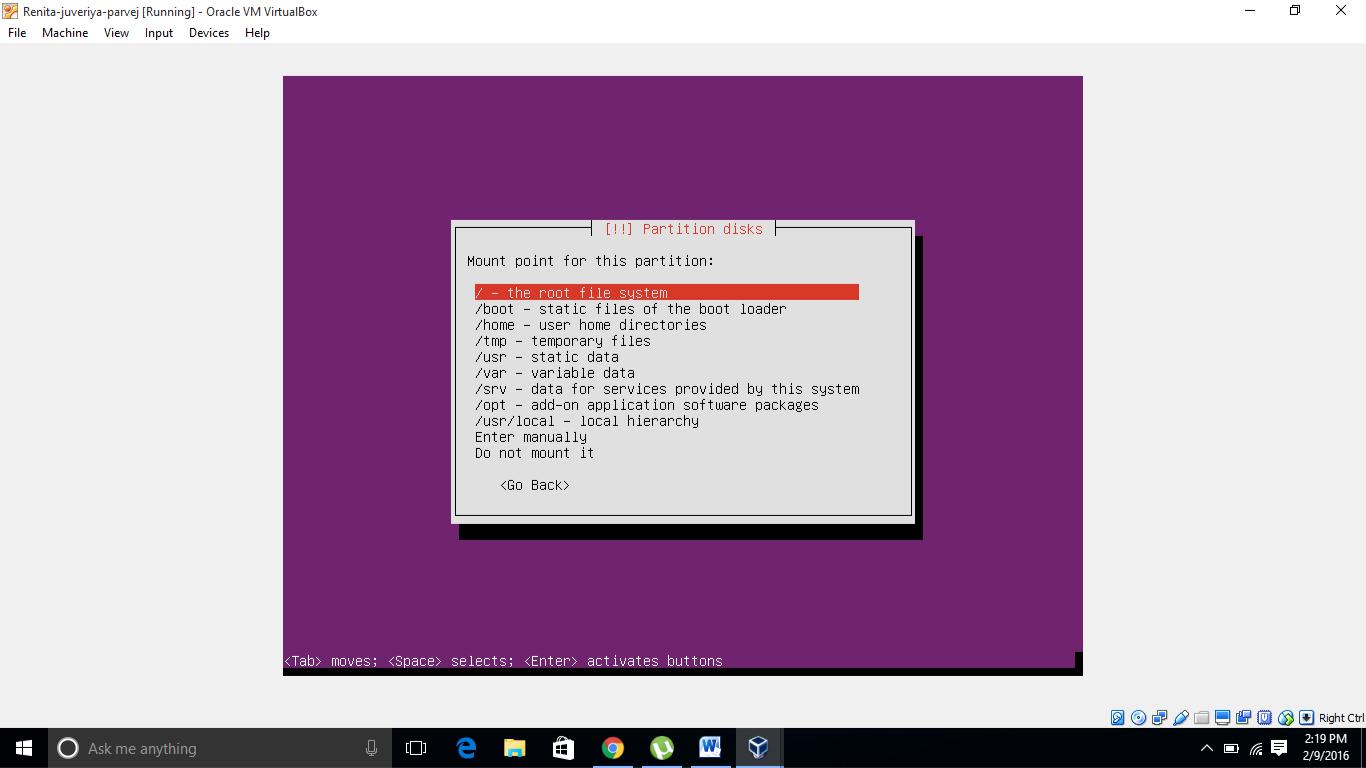
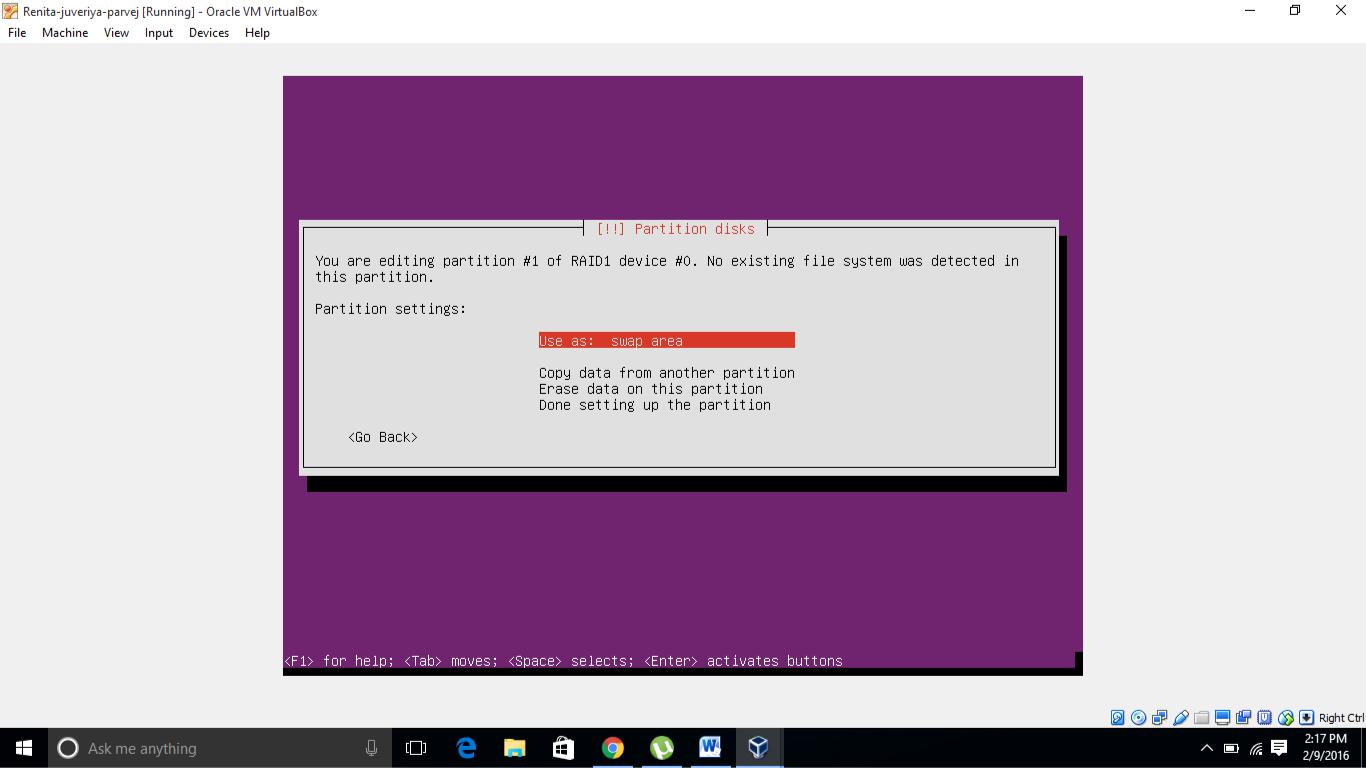
**9)**create the raid for those both disks of particular space.

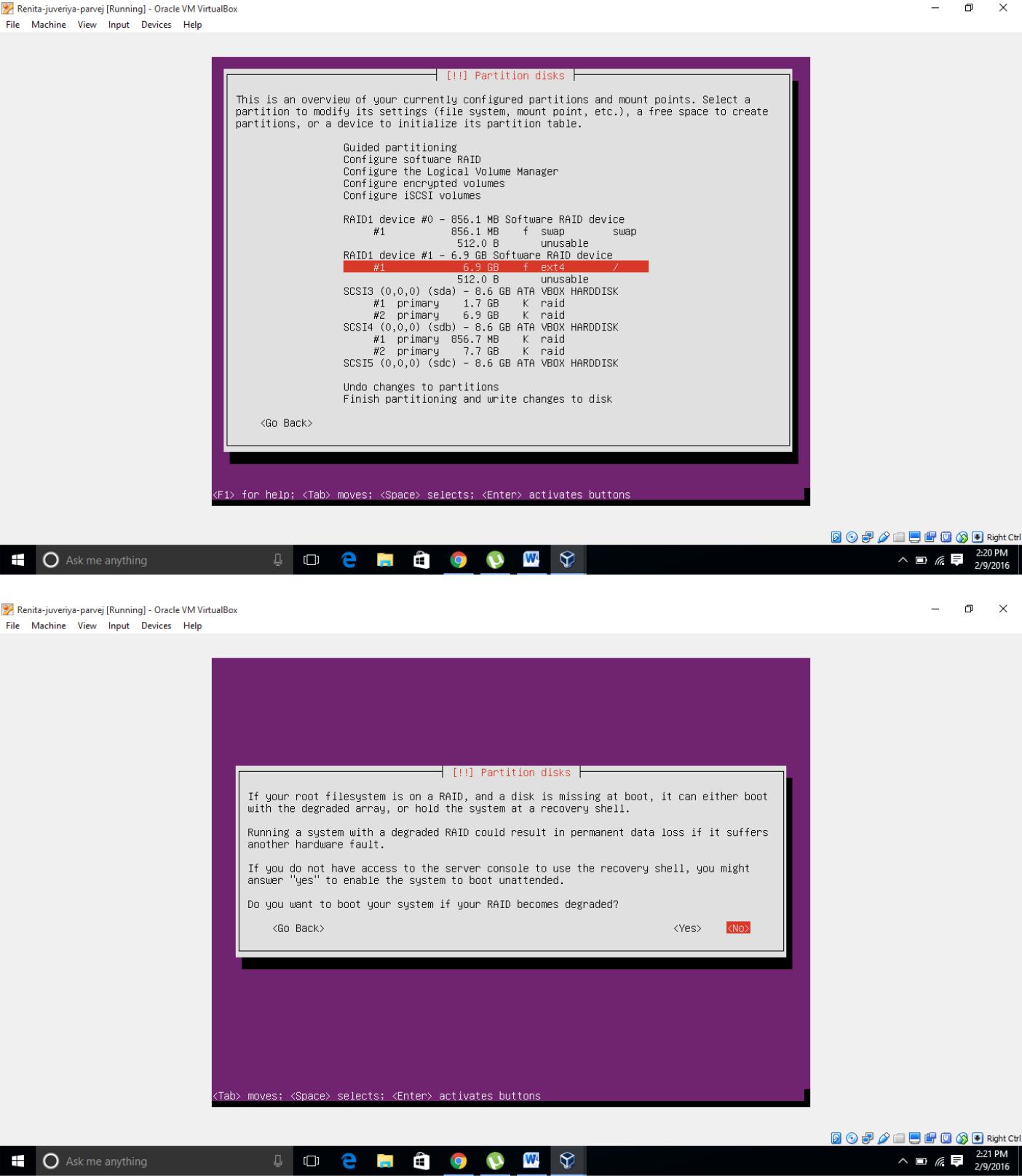
\

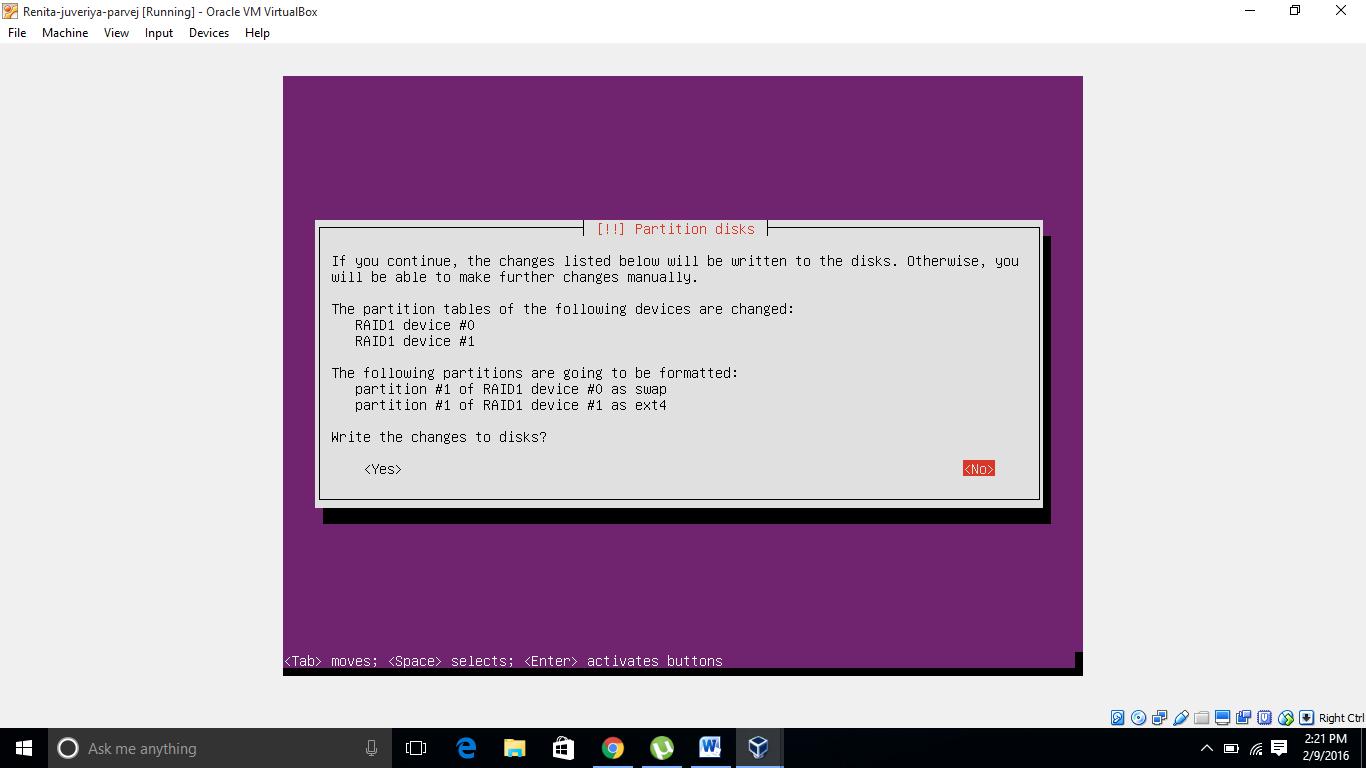


1. Ubuntu will create two partitions as swap,select Raid level .
2. Configuring the RAID
   1. Once you have completed your partitioning in the main "Partition Disks" page select "Configure Software RAID"
   2. Select "Yes"
   3. Select "Create new MD drive"
   4. Select RAID type: RAID 0, RAID 1, RAID 5 or RAID 6
   5. Number of devices. RAID 0 and 1 need 2 drives. 3 for RAID 5 and 4 for RAID 6.
   6. Number of spare devices. Enter 0 if you have no spare drive.
   7. select which partitions to use..
   8. Repeat steps 3 to 7 with each pair of partitions you have created.
   9. Filesystem and mount points will need to be specified for each RAID device. By default they are set to "do not use".
   10. Once done, select finish.

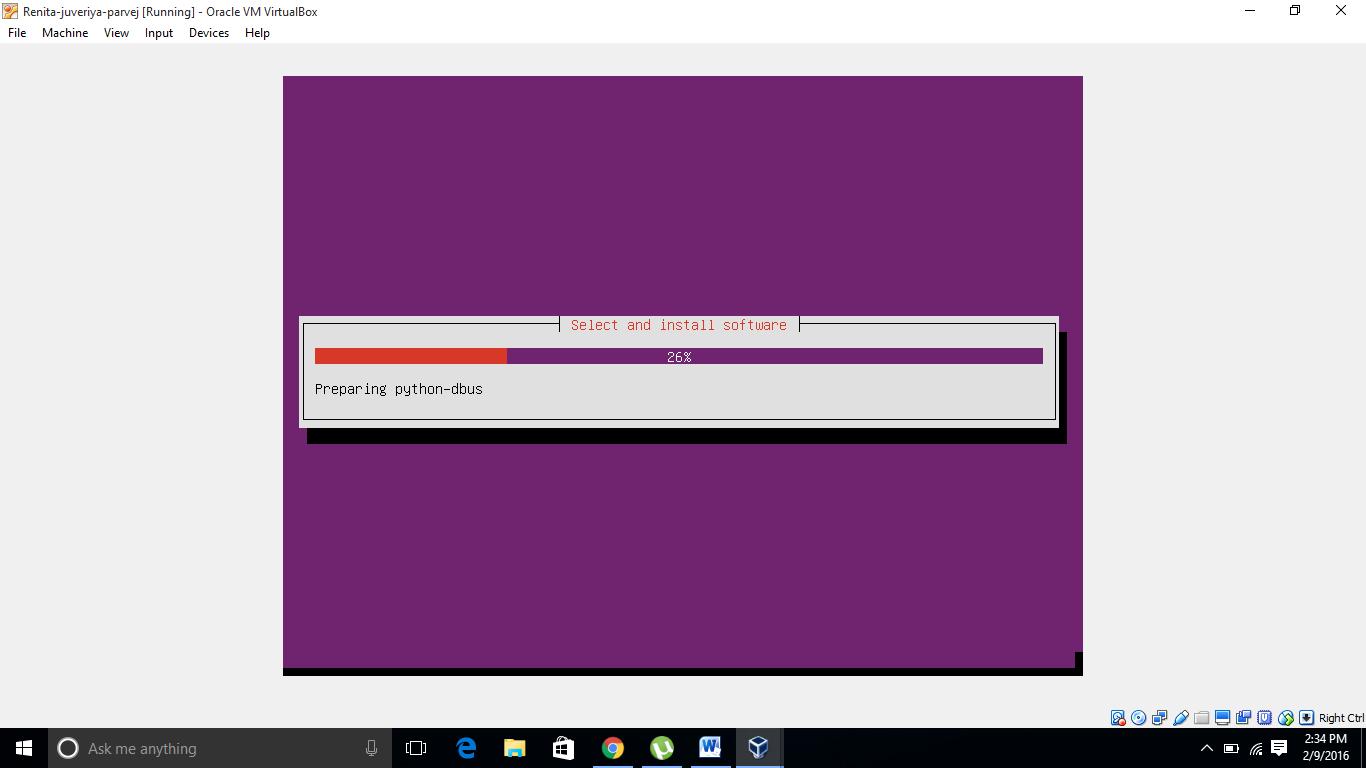


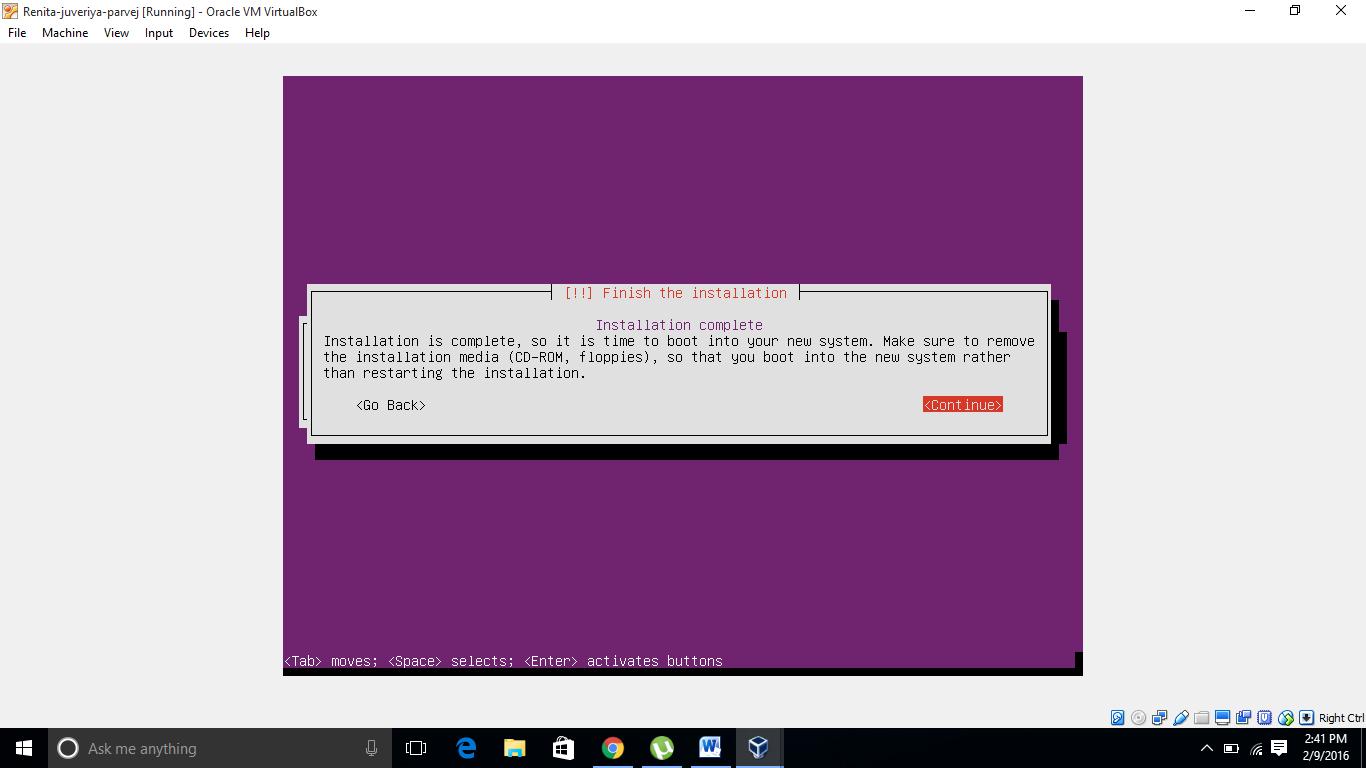






**12)**Select the softwares and all it will be automatically install



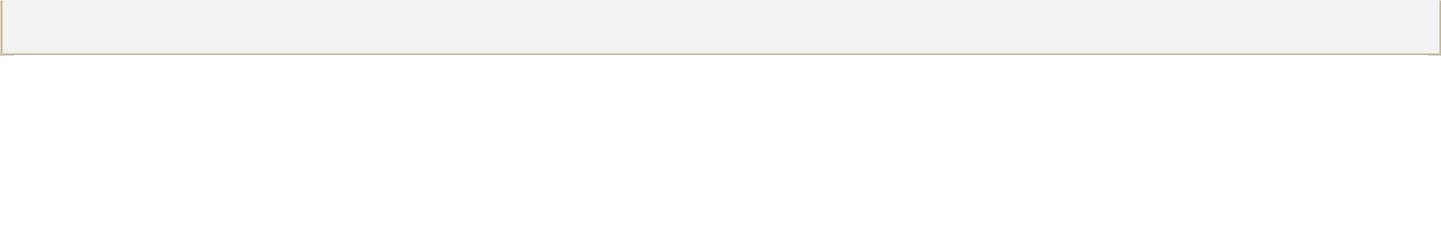


14) **Boot Loader**

In case your next HDD won't boot then simply install Grub to another drive:

A description...

**grub-install /dev/sdb grub-install /dev/sdc**



**Boot from Degraded Disk**

If the default HDD fails then RAID will ask you to boot from a degraded disk. If your server is located in a remote area, the best practice may be to configure this to occur automatically:

* 1. edit /etc/initramfs-tools/conf.d/mdadm
  2. change "BOOT\_DEGRADED=false" to "BOOT\_DEGRADED=true"
* Please provide URL to support claim: (this option is not supported from mdadm-3.2.5-5ubuntu3 / Ubuntu 14.04 onwards)
  1. Additionally, this can be specified on the kernel boot line with the bootdegraded=[true|false]
  2. You also can use #dpkg-reconfigure mdadm rather than CLI!

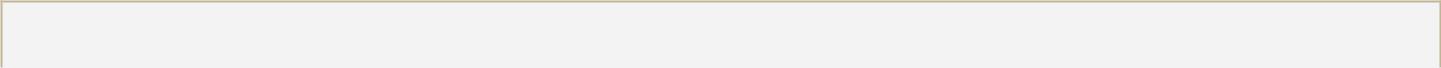
**Verify the RAID**

1. shut-down your server
2. remove the power and cable data of your first drive
3. start your server and see if your server can boot from a degraded disk.

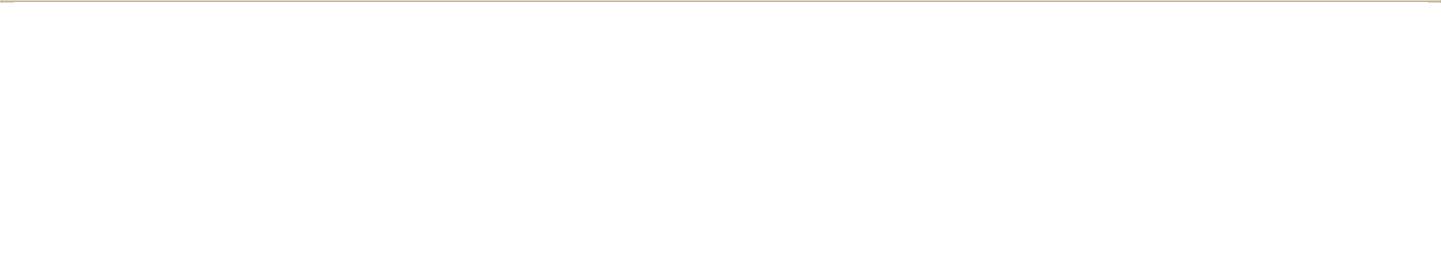
**Troubleshooting**

Swap space doesn't come up, error message in dmesg

Provided the RAID is working fine this can be fixed with:



**sudo update-initramfs -k all -u**

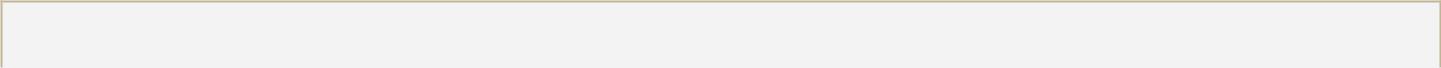


Using the mdadm CLI

For those that want full control over the RAID configuration, the mdadm CLI provides this.

Checking the status of your RAID

Two useful commands to check the status are:



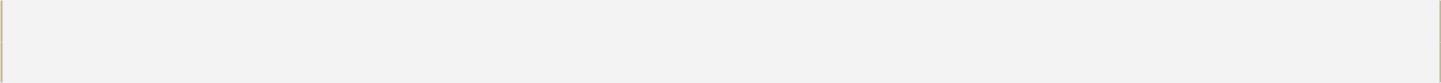
**cat /proc/mdstat**

A description...

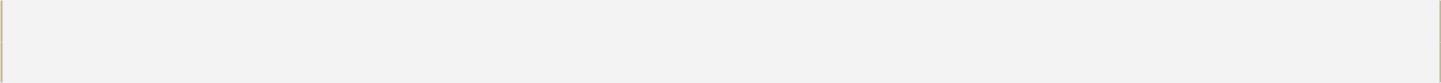
Example output:

A description...

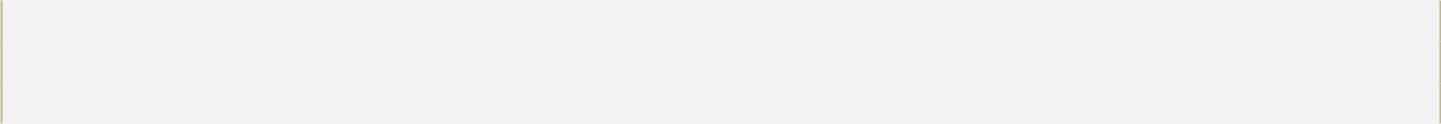
Personalities : [raid1] [raid6] [raid5] [raid4] md5 : active raid1 sda7[0] sdb7[1]



62685504 blocks [2/2] [UU]



md0 : active raid1 sda1[0] sdb1[1] 256896 blocks [2/2] [UU]



md6 : active raid5 sdc1[0] sde1[2] sdd1[1]

A description...

976767872 blocks level 5, 64k chunk, algorithm 2 [3/3] [UUU]

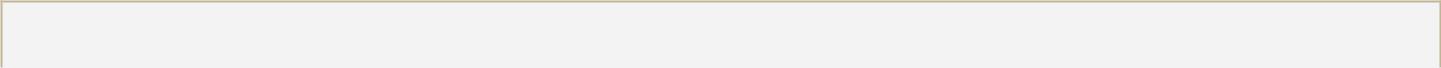


From this information you can see that the available personalities on this machine are "raid1, raid6, raid4, and raid5" which means this machine is set-up to use raid devices configured in a raid1, raid6, raid4 and raid5 configuration.

You can also see in the three example meta devices that there are two raid 1 mirrored meta devices. These are md0 and md5. You can see that md5 is a raid1 array and made up of disk /dev/sda partition 7, and /dev/sdb partition 7, containing 62685504 blocks, with 2 out of 2 disks available and both in sync.

The same can be said of md0 only it is smaller (you can see from the blocks parameter) and is made up of /dev/sda1 and /dev/sdb1.

md6 is different in that we can see it is a raid 5 array, striped across three disks. These are /dev/sdc1, /dev/sde1 and /dev/sdd1, with a 64k "chunk" size or write size. Algorithm 2 shows it is a write algorithm pattern, which is "left disk to right disk" writing across the array. You can see that all three disks are present and in sync.



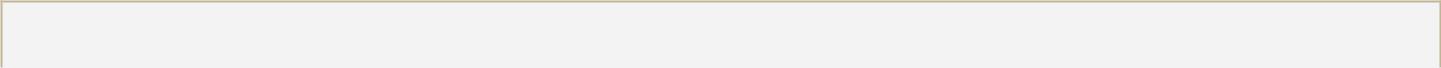
**sudo mdadm --query --detail /dev/md\***

A description...

Replace \* with the partition number.

Disk Array Operation

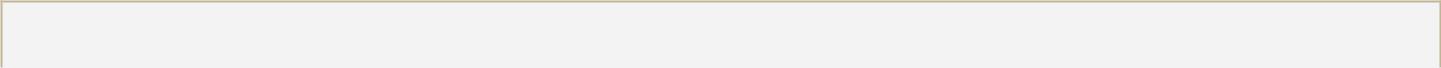
Note: You can add, remove disks, or set them as faulty without stopping an array. 1. To stop an array, type:



**sudo mdadm --stop /dev/md0**



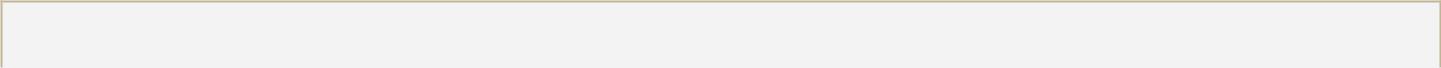
Where /dev/md0 is the array device. 2. To remove a disk from an array:



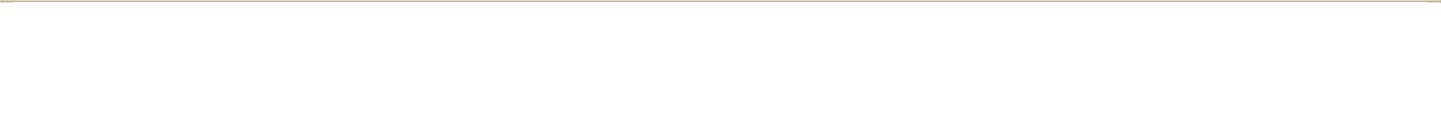
**sudo mdadm --remove /dev/md0 /dev/sda1**

A description...

Where /dev/md0 is the array device and /dev/sda is the faulty disk. 3. Add a disk to an array:

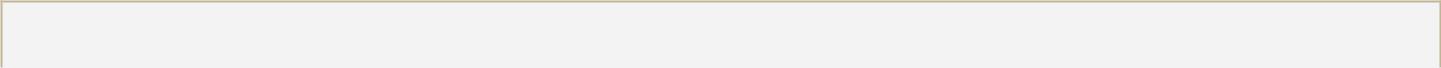


sudo mdadm --add /dev/md0 /dev/sda1

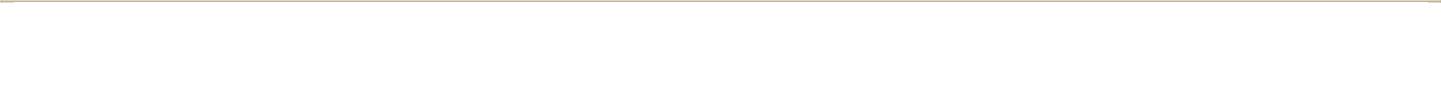


Where /dev/md0 is the array device and /dev/sda is the new disk.**Note**: This is not the same as "growing" the array!

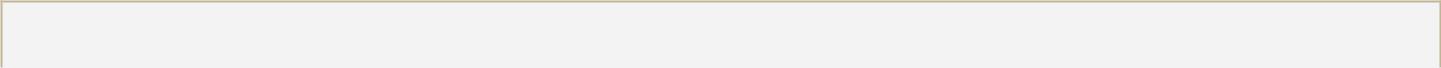
4. Start an Array, to reassemble (start) an array that was previously created:



mdadm --assemble --scan



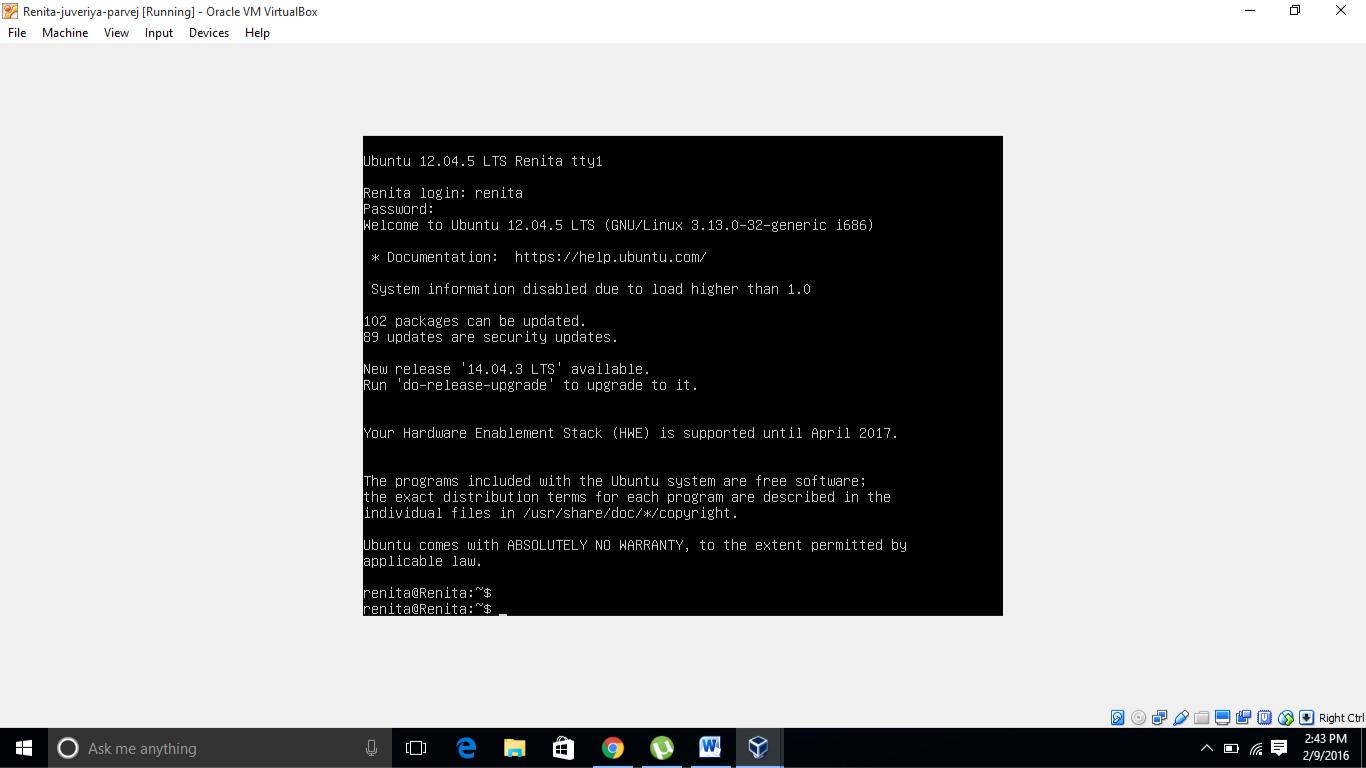
ddadm will scan for defined arrays and start assembling it. 5. To track the status of the array as it gets started:



cat /proc/mdstat

A description...

**15)**Once created and installed all required sofwares ,restart the system



**Conclusion:** That’s it ,wehave implemented all raid levels in Ubuntu using two redundant harddisks with the help of virtual machine.

**Don Bosco Institute Of Technology**

**Department of Information technology**

**References:**

1. <https://help.ubuntu.com/community/Installation/SoftwareRAID>
2. <https://www.youtube.com/watch?v=z84oBqOxsD0>
3. <http://www.tecmint.com/understanding-raid-setup-in-linux/>
4. [www.cs.cf.ac.uk/Dave/Multimedia/node126.html](http://www.cs.cf.ac.uk/Dave/Multimedia/node126.html)