### **Step 1: Installing mdadm Tool and Examine Drives**

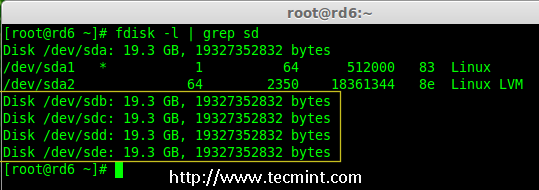
**1.** If you’re following our last two Raid articles (**Part 2** and P**art 3**), where we’ve already shown how to install ‘**mdadm**‘ tool. If you’re new to this article, let me explain that ‘**mdadm**‘ is a tool to create and manage Raid in Linux systems, let’s install the tool using following command according to your Linux distribution.

# yum install mdadm [on RedHat systems]

# apt-get install mdadm [on Debain systems]

**2.** After installing the tool, now it’s time to verify the attached four drives that we are going to use for raid creation using the following ‘**fdisk**‘ command.

# fdisk -l | grep sd

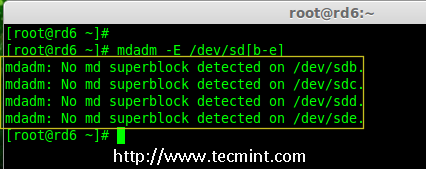


Check Disks in Linux

**3.** Before creating a RAID drives, always examine our disk drives whether there is any RAID is already created on the disks.

# mdadm -E /dev/sd[b-e]

# mdadm --examine /dev/sdb /dev/sdc /dev/sdd /dev/sde



Check Raid on Disk

**Note:** In the above image depicts that there is no any super-block detected or no RAID is defined in four disk drives. We may move further to start creating RAID 6.

### **Step 2: Drive Partitioning for RAID 6**

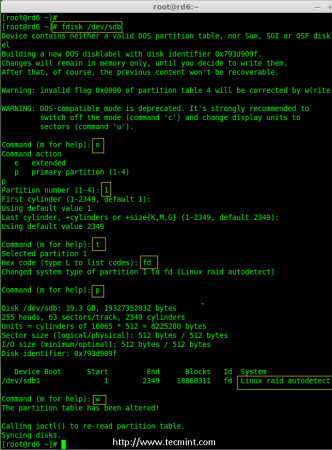
**4.** Now create partitions for raid on ‘**/dev/sdb**‘, ‘**/dev/sdc**‘, ‘**/dev/sdd**‘ and ‘**/dev/sde**‘ with the help of following **fdisk** command. Here, we will show how to create partition on **sdb** drive and later same steps to be followed for rest of the drives.

###### **Create /dev/sdb Partition**

# fdisk /dev/sdb

Please follow the instructions as shown below for creating partition.

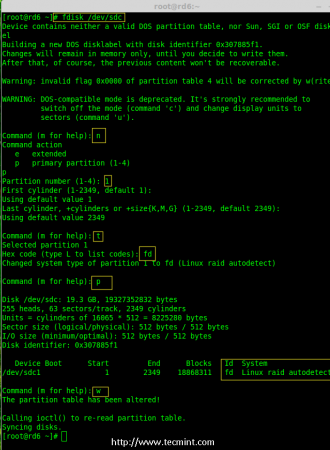
1. Press ‘**n**‘ for creating new partition.
2. Then choose ‘**P**‘ for Primary partition.
3. Next choose the partition number as **1**.
4. Define the default value by just pressing two times **Enter** key.
5. Next press ‘**P**‘ to print the defined partition.
6. Press ‘**L**‘ to list all available types.
7. Type ‘**t**‘ to choose the partitions.
8. Choose ‘**fd**‘ for Linux raid auto and press Enter to apply.
9. Then again use ‘**P**‘ to print the changes what we have made.
10. Use ‘**w**‘ to write the changes.



Create /dev/sdb Partition

###### **Create /dev/sdb Partition**

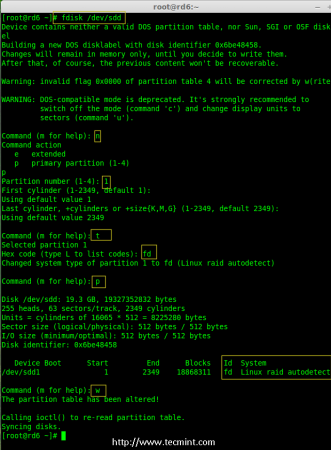
# fdisk /dev/sdc



Create /dev/sdc Partition

###### **Create /dev/sdd Partition**

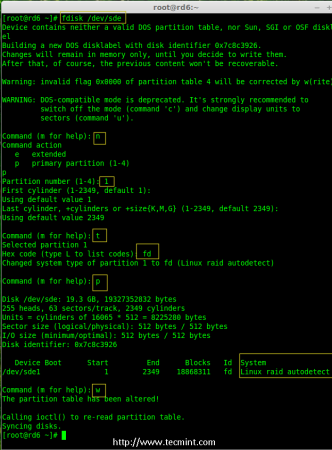
# fdisk /dev/sdd



Create /dev/sdd Partition

###### **Create /dev/sde Partition**

# fdisk /dev/sde



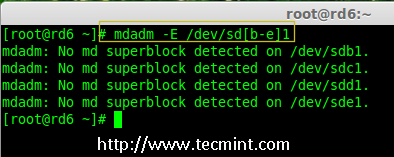
Create /dev/sde Partition

**5.** After creating partitions, it’s always good habit to examine the drives for super-blocks. If super-blocks does not exist than we can go head to create a new RAID setup.

# mdadm -E /dev/sd[b-e]1

or

# mdadm --examine /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1



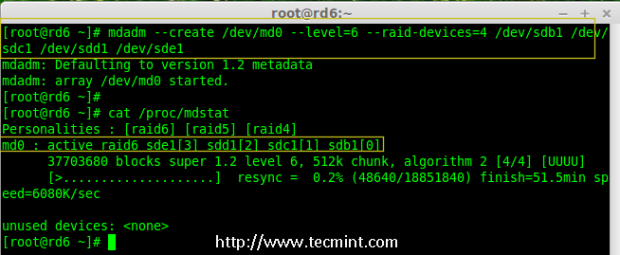
Check Raid on New Partitions

### **Step 3: Creating md device (RAID)**

**6.** Now it’s time to create Raid device ‘**md0**‘ (i.e. **/dev/md0**) and apply raid level on all newly created partitions and confirm the raid using following commands.

# mdadm --create /dev/md0 --level=6 --raid-devices=4 /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1

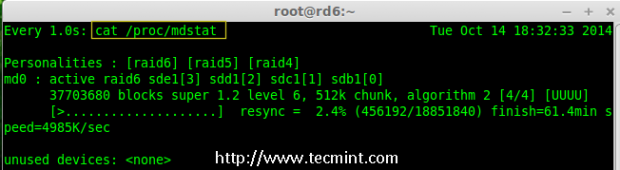
# cat /proc/mdstat



Create Raid 6 Device

**7.** You can also check the current process of raid using **watch** command as shown in the screen grab below.

# watch -n1 cat /proc/mdstat



Check Raid 6 Process

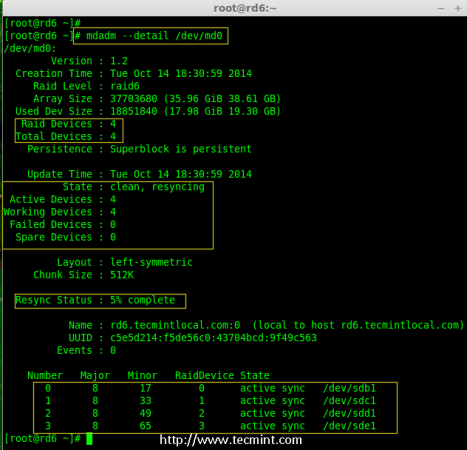
**8.** Verify the raid devices using the following command.

# mdadm -E /dev/sd[b-e]1

**Note:**: The above command will be display the information of the four disks, which is quite long so not possible to post the output or screen grab here.

**9.** Next, verify the RAID array to confirm that the re-syncing is started.

# mdadm --detail /dev/md0

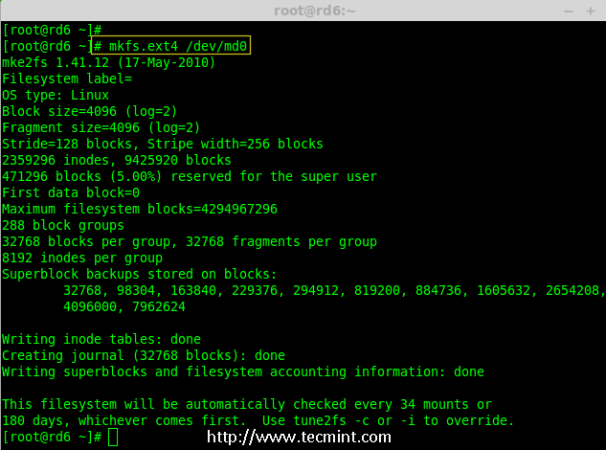


Check Raid 6 Array

### **Step 4: Creating FileSystem on Raid Device**

**10.** Create a filesystem using ext4 for ‘**/dev/md0**‘ and mount it under **/mnt/raid6**. Here we’ve used ext4, but you can use any type of filesystem as per your choice.

# mkfs.ext4 /dev/md0



Create File System on Raid 6

**11.** Mount the created filesystem under **/mnt/raid6** and verify the files under mount point, we can see lost+found directory.

# mkdir /mnt/raid6

# mount /dev/md0 /mnt/raid6/

# ls -l /mnt/raid6/

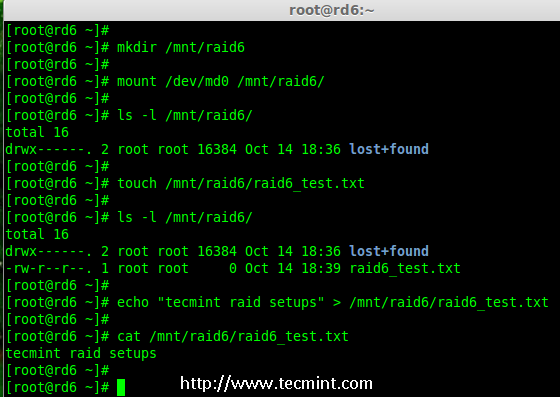
**12.** Create some files under mount point and append some text in any one of the file to verify the content.

# touch /mnt/raid6/raid6\_test.txt

# ls -l /mnt/raid6/

# echo "tecmint raid setups" > /mnt/raid6/raid6\_test.txt

# cat /mnt/raid6/raid6\_test.txt

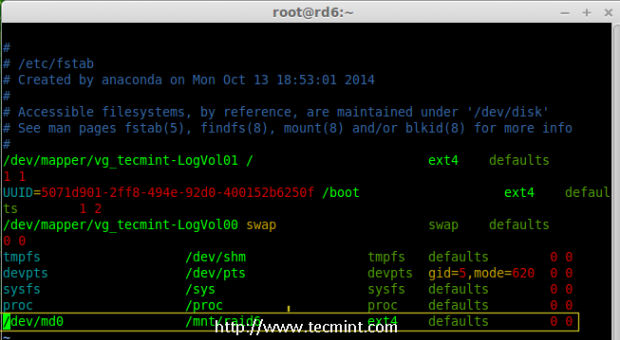


Verify Raid Content

**13.** Add an entry in **/etc/fstab** to auto mount the device at the system startup and append the below entry, mount point may differ according to your environment.

# vim /etc/fstab

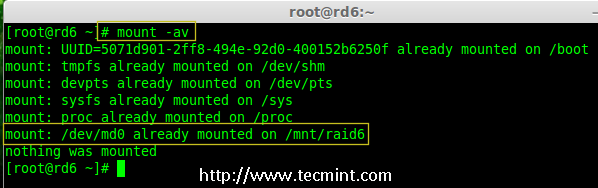
/dev/md0 /mnt/raid6 ext4 defaults 0 0



Automount Raid 6 Device

**14.** Next, execute ‘**mount -a**‘ command to verify whether there is any error in fstab entry.

# mount -av



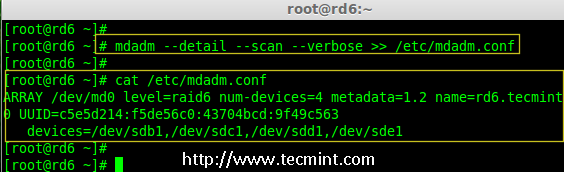
Verify Raid Automount

### **Step 5: Save RAID 6 Configuration**

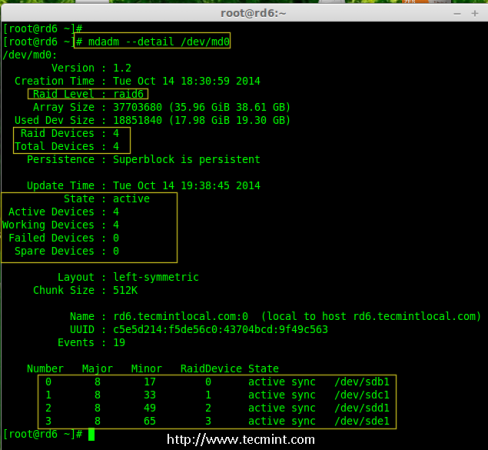
**15.** Please note by default RAID don’t have a config file. We have to save it by manually using below command and then verify the status of device ‘**/dev/md0**‘.

# mdadm --detail --scan --verbose >> /etc/mdadm.conf

# mdadm --detail /dev/md0



Save Raid 6 Configuration



Check Raid 6 Status

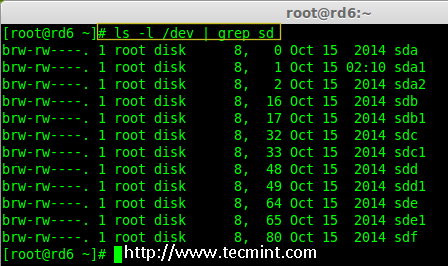
### **Step 6: Adding a Spare Drives**

**16.** Now it has **4** disks and there are two parity information’s available. In some cases, if any one of the disk fails we can get the data, because there is double parity in RAID 6.

May be if the second disk fails, we can add a new one before loosing third disk. It is possible to add a spare drive while creating our RAID set, But I have not defined the spare drive while creating our raid set. But, we can add a spare drive after any drive failure or while creating the RAID set. Now we have already created the RAID set now let me add a spare drive for demonstration.

For the demonstration purpose, I’ve hot-plugged a new HDD disk (i.e. **/dev/sdf**), let’s verify the attached disk.

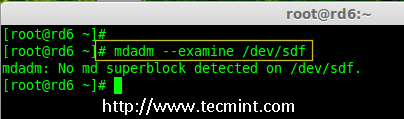
# ls -l /dev/ | grep sd



Check New Disk

**17.** Now again confirm the new attached disk for any raid is already configured or not using the same **mdadm** command.

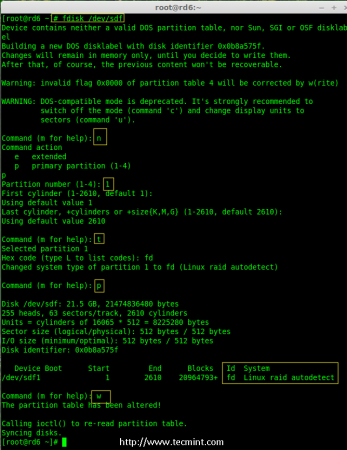
# mdadm --examine /dev/sdf



Check Raid on New Disk

**Note:** As usual, like we’ve created partitions for four disks earlier, similarly we’ve to create new partition on the new plugged disk using **fdisk** command.

# fdisk /dev/sdf



Create /dev/sdf Partition

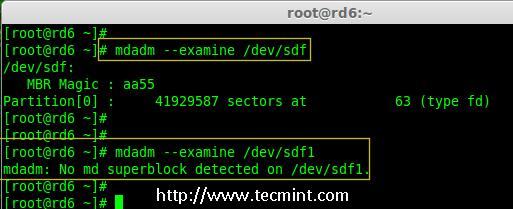
**18.** Again after creating new partition on **/dev/sdf**, confirm the raid on the partition, include the spare drive to the **/dev/md0** raid device and verify the added device.

# mdadm --examine /dev/sdf

# mdadm --examine /dev/sdf1

# mdadm --add /dev/md0 /dev/sdf1

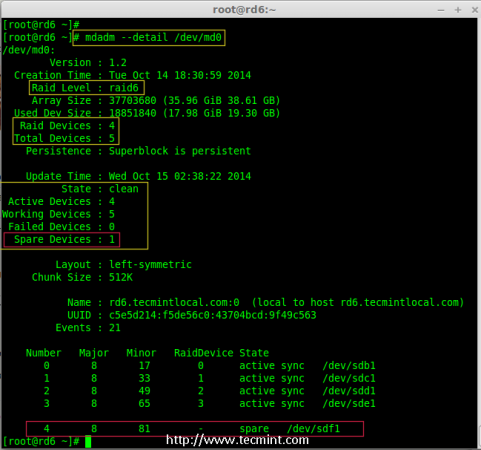
# mdadm --detail /dev/md0



Verify Raid on sdf Partition



Add sdf Partition to Raid



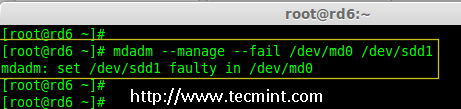
Verify sdf Partition Details

### **Step 7: Check Raid 6 Fault Tolerance**

**19.** Now, let us check whether spare drive works automatically, if anyone of the disk fails in our Array. For testing, I’ve personally marked one of the drive is failed.

Here, we’re going to mark **/dev/sdd1** as failed drive.

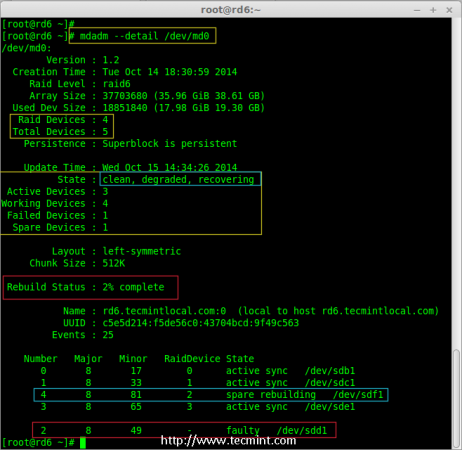
# mdadm --manage --fail /dev/md0 /dev/sdd1



Check Raid 6 Fault Tolerance

**20.** Let me get the details of RAID set now and check whether our spare started to sync.

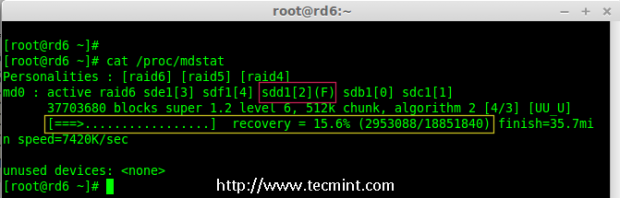
# mdadm --detail /dev/md0



Check Auto Raid Syncing

**Hurray!** Here, we can see the spare got activated and started rebuilding process. At the bottom we can see the faulty drive **/dev/sdd1** listed as faulty. We can monitor build process using following command.

# cat /proc/mdstat



Raid 6 Auto Syncing