**What is Sticky Bit?**

**Sticky Bit** is mainly used on folders in order to avoid deletion of a folder and its content by other users though they having write permissions on the folder contents. If Sticky bit is enabled on a folder, the folder contents are deleted by only owner who created them and the root user. No one else can delete other users data in this folder(Where sticky bit is set). This is a security measure to avoid deletion of critical folders and their content(sub-folders and files), though other users have full permissions.

**Learn Sticky Bit with examples:**

**Example:** Create a project(A folder) where people will try to dump files for sharing, but they should not delete the files created by other users.

**How can I setup Sticky Bit for a Folder?**

Sticky Bit can be set in two ways

1. **Symbolic way (t,represents sticky bit)**
2. **Numerical/octal way (1, Sticky Bit bit as value 1)**

Use [chmod command](http://www.linuxnix.com/2011/10/chmod-command-explained-linuxunix.html) to set Sticky Bit on Folder: **/opt/dump/**

**Symbolic way:**

**chmod o+t /opt/dump/**  
or  
**chmod +t /opt/dump/**

Let me explain above command, We are setting Sticky Bit(+t) to folder /opt/dump by using chmod command.

**Numerical way:**

**chmod 1757 /opt/dump/**

Here in 1757, 1 indicates Sticky Bit set, 7 for full permissions for owner, 5 for read and execute permissions for group, and full permissions for others.

**Checking if a folder is set with Sticky Bit or not?**

Use ls –l to check if the x in others permissions field is replaced by **t or T**

For example: /opt/dump/ listing before and after Sticky Bit set

**Before Sticky Bit set:**

**ls -l**

**total 8**

**-rwxr-xrwx 1 xyz xyzgroup 148 Dec 22 03:46 /opt/dump/**

**After Sticky Bit set:**

**ls -l**

**total 8**

**-rwxr-xrwt 1 xyz xyzgroup 148 Dec 22 03:46 /opt/dump/**

**Some FAQ’s related to Sticky Bit:**

Now sticky bit is set, lets check if user “temp” can delete this folder which is created xyz user.

**$ rm -rf /opt/dump**

**rm: cannot remove `/opt/dump’: Operation not permitted**

**$ ls -l /opt**

**total 8**

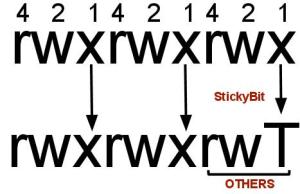
**drwxrwxrwt 4 xyz xyzgroup 4096 2012-01-01 17:37 dump**  
**$**

if you observe other user is unable to delete the folder /opt/dump. And now content in this folder such as files and folders can be deleted by their respective owners who created them. No one can delete other users data in this folder though they have full permissions.

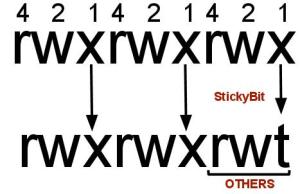
**I am seeing “T” ie Capital s in the file permissions, what’s that?**

After setting Sticky Bit to a file/folder, if you see ‘T’ in the file permission area that indicates the file/folder does not have executable permissions for all users on that particular file/folder.

**Sticky bit without Executable permissions:**

[](http://i1.wp.com/www.linuxnix.com/wp-content/uploads/2012/01/Untitleddrawing-8.jpg)

so if you want executable permissions, Apply executable permissions to the file.  
**chmod o+x /opt/dump/**  
ls -l command output:  
-rwxr-xrwt 1 xyz xyzgroup 0 Dec 5 11:24 /opt/dump/  
**Sticky bit with Executable permissions:**

[](http://i1.wp.com/www.linuxnix.com/wp-content/uploads/2012/01/Untitleddrawing-7.jpg)

 you should see a smaller ‘t’ in the executable permission position.

**How can I find all the Sticky Bit set files in Linux/Unix.**

**find / -perm +1000**

The above [find command](http://www.linuxnix.com/2012/04/learn-linuxunix-find-command-60-practical-examples-part-ii.html) will check all the files which is set with Sticky Bit bit(1000).

**Can I set Sticky Bit for files?**

Yes, but most of the time it’s not required.

**How can I remove Sticky Bit bit on a file/folder?**

**chmod o-t /opt/dump/**

Post your thoughts on this.

# Configuring Kdump on a Red Hat Enterprise Linux 5.3 system

During installation, you are prompted to configure Kdump. However the configuration options are limited to enabling it and setting the Kdump memory size. Enabling Kdump at that time guarantees a local Kdump to the **/var/crash** directory.

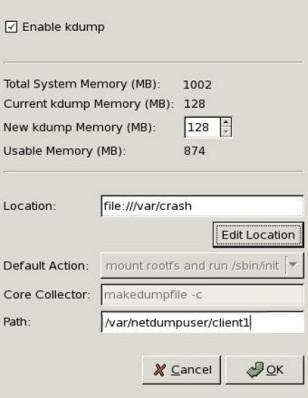
For a remote setup, configure Kdump on an installed system. The following steps configure the client machine Kdump to the remote net dump server that you set up in the previous section.

1. Start the Kdump graphical interface by typing the following command in an X Window System environment (using ssh -X for example):

# system-config-kdump

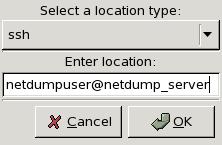
1. Check **Enable kdump**, as follows:
2. In the **Path** box, enter the path to write the crash dumps to the net dump server. This example uses var/netdumpuser/client1

Figure 1. Enable kdump checkbox



1. Click **Edit Location.** In the window, select ssh from the **Select a location type** menu. Enter <username>@<hostname of the dump server> in the **Enter Location** field.

Figure 2. Select a location type



This example uses netdumpuser@netdump\_server. Click **OK**.

1. On the **system-config-kdump** window, verify that the values in the **Location** field and the **Path** field are correct and click **OK**.

The output on your system is similar to the following example:

[root@x206f ~]# system-config-kdump

Stopping kdump: [ OK ]

Detected change(s) the following file(s):

/etc/kdump.conf

Rebuilding /boot/initrd-2.6.18-128.el5kdump.img

/etc/kdump.conf: Could not create netdumpuser@netdump\_server:/var/networkdump/client1, you

probably need to run "service kdump propagate"

Failed to run mkdumprd

Starting kdump: [FAILED]

1. Enter the following command to allow authentication with the net dump server and set up a public transfer:

# service kdump propagate

1. Either reboot the system or restart the **kdump** service, using one of the following commands:
   * If you are enabling Kdump for the first time, reboot the system so that memory is reserved for the Kdump kernel. Enter the following command:
   * # reboot
   * If Kdump is already enabled, restart the **kdump** service by entering the following command:
   * # service kdump restart
2. Verify that **Kdump** daemon is running by entering the following command:

# service kdump status

You should see the following output:

Kdump is operational

1. Verify that the public key authentication works by entering the following command:

# ssh netdumpuser@netdump\_server

You should now be logged into **netdump\_server** as netdumpuser without a password.

# Testing Kdump

Follow this procedure to test if the Kdump setup is working.

Initiating a Kdump causes the client system to reboot. Before starting this procedure, ensure that the client system is ready to be rebooted. Also ensure that the remote server has enough disk space to save the dump file.

On the client system:

1. Write any outstanding data to disk by entering the following command:
2. # sync
3. Trigger the crash dump by entering the following command:
4. # echo c > /proc/sysrq-trigger
5. The client system becomes unresponsive.
6. A crash dump file is created in the **/var/netdumpuser/client1** directory on the remote server. It might take several minutes for the file to be created. Enter the following command to view the crash dump file:
7. # ls /var/netdumpuser/client1/\*
8. /var/netdumpuser/client1/2009-07-16-13:22:

vmcore

When the dumping is done, the client system reboots to its regular kernel. You can find a record of the remote Kdump in the **/var/log/messages** directory, similar to the following entry:

Jul 16 13:22:13 client1 kdump[3070]: Saving 3970 MB crash dump to

ssh://netdumpuser@192.168.0.15/var/netdumpuser/client1/2009-07-16-13:22

# RHEL 5 Linux : configure Kdump on Red Hat Enterprise Linux 5

## Installing required packages

RHEL 5 has the Kdump packages installed by default. If for any reason they are not installed, you need to install the packages “kexec-tools-.rpm” and “system-config-kdump-.rpm” with the following commands:

|  |  |
| --- | --- |
|  | # rpm -ivh kexec-tools-.rpm system-config-kdump-.rpm  or, if your system is registered at the Red Hat Network, by running  # yum install kexec-tools system-config-kdump  Configuration of Kdump  First you need to enable Kdump. There is a configuration dialog available which can be started under a graphical environment by using:  # system-config-kdump  Please check the option box “Enable kdump” at the top of the Dialog.  Next, you have to define the memory to reserve for Kdump In the dialog you see the memory information for your system and the usable memory for Kdump. On most systems a value of “128MB” Kdump memory should be enough.  Finally, you need to define a location where to store the dump file. You have the choice between “file”, “nfs”, “ssh”, “raw”, “ext2″, and “ext3″. This setup is straight forward, please configure the kdump as it fit’s best into your environment. The simplest configuration for the location is “file:///var/crash“.  You need to take care that you have enough disk space on the configured location, at least the physically memory of the system which is expected to dumped.  After you have configured kdump, you need to reboot the system to activate the settings.  More information about the configuration can be found in the file “/usr/share/doc/kexec-tools-\*/kexec-kdump-howto.txt“ |

## Checking the configuration

To make sure that the configuration is working, you can test by using the magic SysRq feature of the kernel.

**WARNING**: Please make sure that no other users are logged into the system and that all work is saved before following the next steps, otherwise this may lead to data loss.

First you need to enable it with the following command:

# echo 1 > /proc/sys/kerne/sysrq

Next you should sync the data of your hard disks to minimize the risk of lost data by

# echo s > /proc/sysrq-trigger

And finally you can force the system to “crash” by

# echo c > /proc/sysrq-trigger

You should see some panic output and the system will restart into the kdump kernel to save the crash dump data. This will take some time depending on the amount of memory of your system and the speed of the device the dump is written to. After the dump is finished the system will reboot back to the normal service.

If you follow the example above you should now find the core file at “/var/crash//vmcore” which indicating the the setup is working.

### Here are the steps to creating LVM mirroring

**Step 1.** Create a two partition using fdisk command( you can use partition or whole disk). Before creating partition, make sure that, how many cylinders has been used and how many are free.

Example 1: **Checking the free space to create partition.**

[root@localhost ~]# fdisk -l

Disk /dev/sda: 10.7 GB, 10737418240 bytes  
255 heads, 63 sectors/track, 1305 cylinders  
Units = cylinders of 16065 \* 512 = 8225280 bytes

Device Boot Start End Blocks Id System  
/dev/sda1 \* 1 765 6144831 83 Linux  
/dev/sda2 766 892 1020127+ 82 Linux swap / Solaris

Above command shows total cylinders 1305 and used cylinders are 892. It means still you have free cylinders, hence we can create partition.

Example: 2 **Creating new 3 partition using fdisk**.

[root@localhost ~]# fdisk /dev/sda  
Press n  
Press p  
Press “Enter” for default starting cylinder”  
Enter 100MB+  
Now Change the partition type to 83 and finally reboot the system.  
Similarly create one more partition of 100MB.  
Note: Make sure that partition ID must be 8e, while creating partitions.

**Step 2.** Create Physical Volumes

Example: 3

[root@localhost ~]# pvcreate /dev/sda[5,6,7]  
Physical volume “/dev/sda5″ successfully created  
Physical volume “/dev/sda6″ successfully created  
Physical volume “/dev/sda7″ successfully created

Above command will initialize partition as lvm partition. Note: Here LVM will assign PV UUID to the partition only.

**Step 3.** Create Volume Group

Example: 4

[root@localhost ~]# vgcreate datavg /dev/sda[5,6,7]  
Volume group “datavg” successfully created

Above command will crate volume group by name datavg. LVM create VGDA(Volume Group Descriptor Area. The VGDA Contains information about Volume Group. LVM Create PE (Physical Extent)

**Step 4.** Creating Mirrored Logical Volumes.

[root@localhost ~]# lvcreate -L 50M -m1 -n mirrorlv datavg  
Rounding up size to full physical extent 52.00 MB  
Logical volume “mirrorlv” created

While creating mirrored volumes in particular volume group, you have to specify the number of copies of the data required, –m argument of the lvcreate command. For creating 1 mirror copy of data, you have to specify the –m1

In above command, we have created 50MB of size logical volume with 1 copy of data to be maintained.

**Step 5.** Creating File system.

[root@localhost ~]# mkfs /dev/datavg/mirrorlv  
mke2fs 1.39 (29-May-2006)  
Filesystem label=  
OS type: Linux  
Block size=1024 (log=0)  
Fragment size=1024 (log=0)  
13328 inodes, 53248 blocks  
2662 blocks (5.00%) reserved for the super user  
First data block=1  
Maximum filesystem blocks=54525952  
7 block groups  
8192 blocks per group, 8192 fragments per group  
1904 inodes per group  
Superblock backups stored on blocks:  
8193, 24577, 40961

Writing inode tables: done  
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 38 mounts or  
180 days, whichever comes first. Use tune2fs -c or -i to override.  
[root@localhost ~]#

**Step 6.** Mounting Filesystem.

Most commonly used method for mounting filesystem is either manually using mount command or by adding entries in /etd/fstab, so that filesystem mount during boot time.

Syntax:

Mount [option]

Example

[root@localhost ~]# mount /dev/datavg/mirrorlv /database

- See more at: http://linoxide.com/linux-how-to/identify-linux-lvm-mirror/#sthash.Mm5yedLm.dpuf

# Logical Volume Manager (LVM);

LVM is a tool for logical volume management which is used to allocating disks, striping, mirroring and resizing logical volumes. With LVM, a hard drive or set of hard drives is allocated to one or more physical volumes. LVM physical volumes can be placed on other block devices which might span two or more disks

Now create physical volumes using the command ***pvcreate***.

[root@server ~]# pvcreate /dev/sdb1 /dev/sdb2 /dev/sdb3   
  Physical volume "/dev/sdb1" successfully created  
  Physical volume "/dev/sdb2" successfully created  
  Physical volume "/dev/sdb3" successfully created

Create a new volume group called ***vg1*** using two physical volumes ***/dev/sdb1*** and ***/dev/sdb2*** using the command ***vgcreate***.

[root@server ~]# vgcreate vg1 /dev/sdb1 /dev/sdb2   
  Volume group "vg1" successfully created

To create logical volume use the command ***lvcreate***. Let us create a logical volume called ***lv1*** with size ***200MB***.

[root@server ~]# lvcreate -L 200M vg1 -n lv1  
  Logical volume "lv1" created

### Format and Mount the logical volume

Now format the newly created logical volume and mount it in the ***/mnt*** directory or wherever you want. [root@server ~]# mkfs.ext4 /dev/vg1/lv1

And mount the logical volume in the ***/mnt*** mount point.

[root@server ~]# mount /dev/vg1/lv1 /mnt/

### Extend Volume Group Size

If you’re running out of the space in the logical volume, you can extend the size of it easily if your physical disk contains free space or with additional physical disk(Hard disk).

Say for example let us extend the volume group ***vg1*** using the physical volume ***/dev/sdb3***. And let us add additonal ***100MB*** to logical volume ***lv1***.

[root@server mnt]# vgextend vg1 /dev/sdb3   
  Volume group "vg1" successfully extended

Then resize the logical vloume ***lv1***.

[root@server mnt]# lvresize -L +100M /dev/vg1/lv1   
  Extending logical volume lv1 to 300.00 MiB  
  Logical volume lv1 successfully resized

Resize the filesystem of logical volume ***lv1***.

[root@server mnt]# resize2fs /dev/vg1/lv1   
resize2fs 1.41.12 (17-May-2010)  
Filesystem at /dev/vg1/lv1 is mounted on /mnt; on-line resizing required  
old desc\_blocks = 1, new\_desc\_blocks = 2  
Performing an on-line resize of /dev/vg1/lv1 to 307200 (1k) blocks.  
The filesystem on /dev/vg1/lv1 is now 307200 blocks long.

Now verify the new size of the logical volume ***lv1***.

[root@server mnt]# lvdisplay /dev/vg1/lv1   
  --- Logical volume ---  
  LV Name                /dev/vg1/lv1  
  VG Name                vg1  
  LV UUID                dgLZ79-JZdn-NUSF-fUS1-YVFk-36qs-iuafhE  
  LV Write Access        read/write  
  LV Status              available  
  # open                 1  
  LV Size                300.00 MiB  
  Current LE             75  
  Segments               3  
  Allocation             inherit  
  Read ahead sectors     auto  
  - currently set to     256  
  Block device           253:0

It’s done. Now the size of the logical volume ***lv1*** is extended by ***100MB***.

### Remove Logical Volume

Come out of the ***/mnt*** mount point, unmount the logical volume ***lv1*** and remove it using command ***lvremove***.

[root@server mnt]# cd ..  
[root@server /]# umount /mnt/  
[root@server /]# lvremove /dev/vg1/lv1   
Do you really want to remove active logical volume lv1? [y/n]: y  
  Logical volume "lv1" successfully removed

### Remove Volume Group

[root@server /]# vgremove /dev/vg1  
  Volume group "vg1" successfully removed

### Remove Physical Volume

[root@server /]# pvremove /dev/sdb1 /dev/sdb2 /dev/sdb3  
  Labels on physical volume "/dev/sdb1" successfully wiped  
  Labels on physical volume "/dev/sdb2" successfully wiped  
  Labels on physical volume "/dev/sdb3" successfully wiped

LVM can be defined as a software disk management on top of physical hard disks, in order to provide features such as making large volumes by combining multiple hard disks, easy resizing of a partition, removing of physical disks without affecting data online, improving the throughput by striping data across multiple physical disks, easy backup management with the help of snapshots etc. LVM also provides, mechanisms that are similar to raid level 1 and raid level 0(mirroring and striping), if you are interested in understanding raid, i will recommend reading the below articles.

## What are the components of LVM(Logical Volume Management)

In order to understand LVM in a easier way, we have divided each and every component of LVM into different parts, and each part is studied separately. In this section let's go through each and every components of LVM.

### 

### (1) Physical disks or Physical volume

As we have discussed earlier, a large partition can be made by combining different hard disks or partitions together. Physical disks or physical volumes are either entire hard disks or partitions. You might ask why partitions?

LVM is a software implementation, hence you can supply a partition of any size, or an entire physical hard disk of any size, or a single big partition made out of a hard disk, as a physical disk.

Either you allocate a hard disk, or a partition, as a physical disk, you first need to label that device as physical volume for LVM. This labeling of the device(a partition or a hard disk) can be done by adding a label at the second sector of the device, by using fdisk command. This kind of labeling the partition or hard disk will make identifying a physical volume easier.

So lets create three different partitions of 100 MB, and label it as LVM physical volume, (in real world you can use three different hard disks as physical volumes.)

Repeat the above mentioned steps three times to create 3 partitions of 100 MB. We will be using these three partitions as Physical volumes for LVM, so lets change the partition label for them to LVM. This can be done by the following method.

In the above shown method, we have changed the partition label type to 8e (which is used to label a partition with Linux LVM type). Do this for all the three partitions we have created, so that all of them can be made to LVM physical volume type.

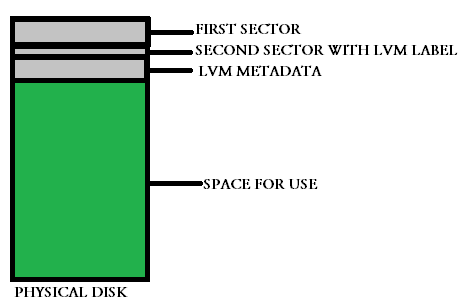
Keep the following things in mind, while changing the type of a partition to LVM in Linux. The below points mentions the reason for changing the type, what does the type change do.

* LVM label on a partition will help the system to identify the disk/partition as LVM physical volume
* The label resides on the second sector (by default a sector is assumed as 512 bytes)
* It contains the size of the physical volume
* It will also tell the location of the LVM metadata on that physical disk/volume (don't worry! we will be discussing lvm metadata on the next section of this tutorial)
* Label also provides a Unique Identification to the physical volume.

To understand what is UUID in a partition, i will recommend reading the below post.

**Read:** [What is UUID in a Partition](http://www.slashroot.in/what-uuid-used-fstab)

Let me make that point clear once again. We have used partitions while creating physical volumes in the above example(3 partitions of 100 MB size), however you can use whole hard disks as physical volumes in LVM. The contents of a physical volume are depicted in the below diagram, to make the above mentioned points about physical volume very clear.



The above shown picture gives a clear idea about how a physical volume is laid out in LVM. All the three physical disks(partitions in our case), has the layout similar to the one shown in the above picture.

However we have only changed the type of the partitions to LVM, in order to create a physical volume we need to take help of a command called as **pvcreate**.

|  |  |
| --- | --- |
| 1  2  3 | [root@localhost ~]# pvcreate /dev/sda{6,7,8}        ---->> physical volume creation    Physical volume "/dev/sda6" successfully created    Physical volume "/dev/sda7" successfully created    Physical volume "/dev/sda8" successfully created |

### (2) Volume Group in LVM

|  |  |
| --- | --- |
| 1  2 | [root@localhost ~]# vgcreate vg00 /dev/sda6 /dev/sda7 /dev/sda8  --->>Volume group creation.    Volume group "vg00" successfully created |

The above command creates a volume group with the three physical volumes (partitions in our case /dev/sda6,sda7,sda8). And a name **vg00** is assigned to that volume group. For activating this volume group, you need to run the command **vgchange**

|  |  |
| --- | --- |
| 1  2 | [root@localhost ~]# vgchange -a y vg00                   ---->>activate volume group vg00    0 logical volume(s) in volume group "vg00" now active |

Let's now see the complete information of the volume group created with **vgdisplay** command.

|  |  |
| --- | --- |
|  |  |

The above command output gives you the complete information about the volume group created. Most importantly, we need to note the below things.

1. Version of LVM used (lvm 2 in our case)
2. Volume group name (vg00)
3. Metadata details
4. Physical extent size (by default its 4 mb in lvm 2, we will be see changing that default physical extent size later)
5. number of physical extents in volume group
6. Free physical extents in the Volume group
7. UUID of the volume group

You can change the Physical extent size from the default value of 4 MB to whatever you like based on the requirement, while creating the volume group. Lets change the physical extent size while creating the volume group. Keep the fact in mind that, you cant change the physical extent size once the volume group is created.

|  |  |
| --- | --- |
| 1 | [root@localhost ~]# vgcreate vg00 /dev/sda6 /dev/sda7 /dev/sda8 -s 34M |

The above command will make the volume group with exactly the same name, but will keep the physical extent size as 34 MB, instead of the default  4MB(which you can verify later on with **vgdisplay** command).

Now lets go ahead with understanding the last component of LVM, which the operating system and the user will be using for storing files and data.

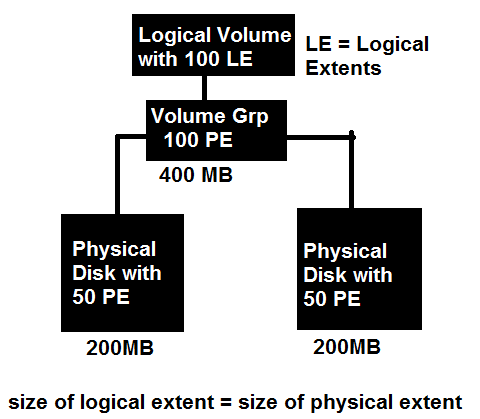
### (3) Logical Volume in LVM

The volume group which we just made, is a pool of disk space, by combining all physical extents of all the physical volumes. We can make partitions from this large volume group by creating logical volumes. These logical volumes are the real place, where the user will be accessing the data.

There are three kinds of logical volumes in LVM. Lets understand each of them first, and then will see methods to create logical volumes from a volume group.

* **Linear Logical Volumes**

Linear logical volumes are nothing but a logical volume made by simply combining multiple physical volumes to one. Let's take an example, to understand linear logical volumes in a much better way. say you have two physical volumes that have 50 physical extents of 4 MB each. Which means one physical volume is of 200M B (50 x 4).



In the above shown example, we have two physical disk with 50 Physical extents of 4 MB each, making a 200 MB disk. While creating a Linear logical volume it will be a combined use of physical extents on both the physical volumes in order.

To the operating system the logical volume created will appear as one single disk with 100 physical extent(although the physical extents are spread on two physical disk linearly.). Another important fact to note here is that, it is not necessary to have two physical volume of same size, in other words, you can have a different number of physical extent in two physical volume that will combine and create one logical volume with the size of both the disks. So for example, you can have one physical volume with 60 physical extent and other physical volume with 40 physical extent, that will combine together and form 100 physical extent in one logical volume.

Let's go ahead and see the commands, that are used while creating a linear volume in LVM.

|  |  |
| --- | --- |
| 1 | [root@myvm1 ~]# lvcreate -L1000 -n example examplevg |

Above command creates a linear logical volume of 1000 MB with the name of **example**, on the volume group **examplevg**. If no unit (M,K,G) is specified along with the -L option in the above command, M is assumed.

Alternatively instead of specifying the size in megabytes, or gigabytes, you can also specify the number of physical extents to be used in that logical volume with the help of -l (lower case letter) option. Lets see a couple of similar example.

|  |  |
| --- | --- |
| 1 | [root@myvm1 ~]# lvcreate -l 100 -n example examplevg |

the above command will create a logical volume by using 100 logical extents(size of logical extent = size of physical extent), to create the logical volume. Now you if you know the physical extent size, then you can easily calculate the volume size in MB. Alternatively, you can also specify which extents to use from which physical volumes in the volume group.

|  |  |
| --- | --- |
| 1 | [root@myvm1 ~]# lvcreate -l 100 -n example examplevg /dev/sda1:25-50 /dev/sdb1:100-176 |

the above command will create a logical volume of 100 extents, and will use the the extents 25 to 50 from /dev/sda1 and then 75 extents from /dev/sdb1(ranging from 100 to 176)

* **Stripped logical volume in LVM**

One of the major strength of logical volumes made out of LVM is that, you can in fact control the way data is structured across the underlying physical disks. Lets see how data is structured when we configure stripped logical volume. Stripped logical volume is very much similar to raid level 0.

In stripped logical volume, data in the underlying physical volume is stripped across the number of physical disks. Similar to raid level 0 which employs striping across the disks, a logical volume with striping will improve both sequential read and writes.

If see the above picture, the data is striped across three different physical disks. First stripe is on the first physical disk, second stripe is on the second and so on.

You can create a striped logical volume with the help of lvcreate command. Creating a striped logical volume can be done by using **-i** option in lvcreate command. The number specified using this option is the number of physical disk, that the logical volume will be stripped. Let's see the below example.

|  |  |
| --- | --- |
| 1 | [root@myvm1 ~]# lvcreate -L 10G -i2 -I64 -n example examplevg |

The above command will create a striped logical volume with the stripe size of 64 kb, and a total size of 10G, and will use 2 physical volume for the stripes, and will be named **example**, on the volume group named **examplevg**.

You might remember that, you can create a volume group from different sized physical volumes. In that case the maximum size of a striped logical volume can only be equal to the size of the smallest physical volume.

* **Mirrored Logical Volume**

Mirrored logical volumes are very much similar to raid level 1 in which data is mirrored across the number of disks used. It provides better redundancy for the data.

In easy words, you can say that an exact replica copy of one physical disk is present on the other physical disk. While creating logical volume with mirrors, you can give multiple mirrors for the data. Mirroring of a logical volume requires at least three physical volumes.

1. The first physical volume will act as the first mirror leg
2. The second physical volume will act as the second mirror leg
3. The third physical volume will be saving logs related to the mirroring

In the above example, the first two physical volume acts as two mirrored legs, and the third one stores the log, of what is synced in mirrors and what is not synced in the mirrors.

Let's now go ahead and create a logical volume with mirroring.

|  |  |
| --- | --- |
| 1 | [root@myvm1 ~]# lvcreate -L 10G -m1 -n mirrorexample examplevg |

In the above command, -L option is used to specify the size of the logical volume, -m option is used to specify number of mirrors for the logical volume(in our case we have used one mirror), and the name of the logical volume is **mirrorexample** and is created from a volume group known as **examplevg**.

Specifying the **m1** option will create two copies of the file system, one is the linear volume and the other is the mirror of the volume. Hence if you want two mirrors then you need to specify -**m2** option with **lvcreate** command.

You can always convert a linear volume to a mirrored volume or a mirrored volume to a linear volume by using the below method.

|  |  |
| --- | --- |
| 1 | [root@myvm1 ~]# lvconvert -m1 /dev/vg00/lv0 |

the above command can be used to convert the linear logical volume to a mirrored logical volume with one mirror, and its name is /dev/vg00/lv0.

|  |  |
| --- | --- |
| 1 | [root@myvm1 ~]# lvconvert -m0 /dev/vg00/lv0 |

With the help of the above command, you can remove mirroring from a logical volume named **/dev/vg00/lv0.**

**Note:** In all the above shown example's of creating logical volume of any type, you need to first format the logical volume with your required file system, similar to the formatting done on any normal partition, and mount them to use.

Let's now have a look at different commands of LVM and see what they does. We will also be seeing methods used to resize an existing logical volume and much more.

## How to resize an LVM, logical volume?

Resizing of a logical volume can be done by simple **lvextend** and **lvreduce** commands in Linux. We will be going through both increasing as well as decreasing the size of an LVM in Linux.

The first step that needs to be taken before resizing is to unmount the logical volume if its currently being used. Let's begin by increasing the size of the LVM.

|  |  |
| --- | --- |
| 1  2  3  4 | [root@localhost ~]# lvextend -L +50M /dev/vg00/lv00  -->> I am increasing logical volume to 50mb.    Rounding up size to full physical extent 52.00 MB    Extending logical volume lv00 to 204.00 MB    Logical volume lv00 successfully resized |

The above command, can be used to increase the size of the logical volume by 50 M. -L is used to specify the amount of size to increase, and then the last argument is of course the logical volume name.

Now let's make the volume group active before using the resized logical volume with the help of the below command.

|  |  |
| --- | --- |
| 1  2 | [root@localhost ~]# vgchange -a y vg00      --->>activating change.    1 logical volume(s) in volume group "vg00" now active |

Now before mounting the resized Logical volume, lets format the newly added space with the help of resizefs command.

|  |  |
| --- | --- |
| 1  2  3  4  5 | [root@localhost ~]# resize2fs /dev/vg00/lv00  resize2fs 1.39 (29-May-2006)  Filesystem at /dev/vg00/lv00 is mounted on /slashroot.in; on-line resizing required  Performing an on-line resize of /dev/vg00/lv00 to 208896 (1k) blocks.  The filesystem on /dev/vg00/lv00 is now 208896 blocks long. |

Now you can simply mount the logical volume, at your required mount point, and start using the increased size of the logical volume.  Let's now see the method used to decrease the size of a logical volume.

Its always recommended to do an fsck file system check before reducing the size of the logical volume. After running **fsck**, you need to first unmount the file system, and then reduce the size of the file system with the help of resize2fs command as shown below.

|  |  |
| --- | --- |
| 1  2  3  4 | [root@localhost ~]# resize2fs /dev/vg00/lv00 100M  -->>resize filesystem  resize2fs 1.39 (29-May-2006)  Resizing the filesystem on /dev/vg00/lv00 to 102400 (1k) blocks.  The filesystem on /dev/vg00/lv00 is now 102400 blocks long. |

Note the fact the 100M specified in the above command, is the final size of the file system after reduction(it is not the size by which the file system will get reduced to, but is the final size of the file system).

Now you can reduce the size of he logical volume by **lvreduce** command as shown below.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | [root@localhost ~]# lvreduce /dev/vg00/lv00 -L 100M  -->>I am reducing logical volume to 100mb.    WARNING: Reducing active logical volume to 100.00 MB    THIS MAY DESTROY YOUR DATA (filesystem etc.)  Do you really want to reduce lv00? [y/n]: y    Reducing logical volume lv00 to 100.00 MB    Logical volume lv00 successfully resized |

In the above command also the logical volume is reduced to 100 M not by 100 M.

Now you can simply mount the volume back and confirm the new size. We will be coming up with a second part of this post, which will be discussing some interesting things like the following.

* Take metadata backup of an LVM
* Move a complete LVM from one server to another server
* Mirroring of logical volume in depth as well as mirroring for log
* Taking snapshots
* As well as some other interesting LVM commands.

# http://lawrit.lawr.ucdavis.edu/howto_icon.gifAdding a mirror to an existing LVM volume (proxmox)

Adding a mirror to an existing LVM volume

## Purpose

Mirror a LVM volume, e.g. add a mirror to proxmox's root/swap/data so that locally created VMs are protected from disk failure.

## Prerequisities

2 physical disks.

## Step by step

First, you'll need to add your second disk to the volume group containing the volumes you want to mirror:

pvcreate /dev/sdb1 # /dev/sdb1 is the physical disk to contain the mirrors  
  
vgextend pve /dev/sdb1  # pve is volume group name you wish to extend  
  
lvconvert -m 1 --corelog pve/data # -m 1 (how many mirrors to create) --corelog (keep the log in memory, requires a resync on reboot)

# KVM: Install CentOS / RHEL Using Kickstart File (Automated Installation)

Kickstart is a network installation system for RHEL, Fedora and CentOS Linux distributions. Another good option is Cobbler which is a Linux provisioning server that centralizes and simplifies control of services including DHCP, TFTP, and DNS for the purpose of performing network-based operating systems installs. In this tutorial, I'm going to show you how to use kickstart file to install CentOS.  
Create Kickstart file

An automated installation method to install CentOS / Fedora or RHEL is recommend to automate procedure. Using kickstart, a system administrator can create a single file containing the answers to all the questions that would normally be asked during a typical RHEL Linux installation. Use kickstart GUI tool called "Kickstart Configurator" (run **system-config-kickstart** command to start the tool) to create a file called ks.cfg as follows:

auth --useshadow --enablemd5

bootloader --location=mbr

zerombr

clearpart --all --initlabel

text

firewall --enabled --port=22:tcp

firstboot --disable

keyboard us

network --device eth0 --bootproto static --ip 10.10.21.76 --netmask 255.255.255.240 --gateway 10.10.21.100 --nameserver 10.10.21.1,10.10.21.2 --noipv6

network --device eth1 --bootproto static --ip 123.1.2.6 --netmask 255.255.255.240 --gateway 123.1.2.100 --nameserver 10.10.21.1,10.10.21.2 --hostname centos.nixcraft.in --noipv6

lang en\_US

logging --level=info

url --url=http://mirrors.nixcraft.in/centos/5.5/os/x86\_64/

reboot

rootpw --iscrypted $1$somepassword

selinux --enforcing

skipx

timezone America/New\_York

install

part / --bytes-per-inode=4096 --fstype="ext3" --grow --size=1

part swap --recommended

%packages

@core

--nobase

%post

(

echo '10.0.0.0/8 via 10.10.21.100' > /etc/sysconfig/network-scripts/route-eth0

sed -i 's/LABEL=\//& console=ttyS0/' /etc/grub.conf

echo 'S0:12345:respawn:/sbin/agetty ttyS0 115200' >> /etc/inittab

echo "ttyS0" >> /etc/securetty

echo 'IPV6INIT=no' >> /etc/sysconfig/network

echo 'install ipv6 /bin/true' >> /etc/modprobe.conf

) 1>/root/post\_install.log 2>&1

Upload this file to a web server as ks.cfg. You can use nfs server too.

## virt-install: Install CentOS using Kickstart

Type the following command:  
# virt-install \  
-n centos \  
-r 2048 \  
--vcpus=1 \  
--os-variant=rhel5.4 \  
--accelerate \  
-v \  
-w bridge:br0 \  
-w bridge:br1 \  
--disk path=/emc/kvm/centos.img,size=100 \  
-l http://mirrors.nixcraft.in/centos/5.5/os/x86\_64/ \  
-nographics \  
-x "ks=http://10.10.21.3/static/ks.cfg ksdevice=eth0 ip=10.10.21.76 netmask=255.255.255.240 dns=10.10.21.1 gateway=10.10.21.100"

Red Hat Linux operating system installations can be done via a network connection using a Kickstart server. It is frequently much faster than using CDs and the process can be automated.

Example Kickstart

Get the kickstart cfg from http server and start the install

boot: linux ks=<http://server.com/path/to/kickstart/file>

Get the kickstart cfg from nfs server and start the install

boot: linux ks=nfs:server:/path/to/kickstart/file

Serving the Kickstart file from nfs server through dhcp /etc/dhcpd.conf

next-server 10.10.10.100;

filename "/export/rhinstall/kickstart/ks.cfg"

## Setup a Kickstart Server

01. Install and configure the DHCPD server

02. Install tftp server and enable TFTP service

a. yum install tftp-server

b. Enable TFTP server.

vi /etc/xinetd.d/tftp and change disable to 'no'

c. service xinetd restart

03. Install syslinux if not already installed

a. yum install syslinux

04. Copy needed files from syslinux to the tftpboot directory

cp /usr/lib/syslinux/pxelinux.0 /tftpboot

cp /usr/lib/syslinux/menu.c32 /tftpboot

cp /usr/lib/syslinux/memdisk /tftpboot

cp /usr/lib/syslinux/mboot.c32 /tftpboot

cp /usr/lib/syslinux/chain.c32 /tftpboot

04. Create the directory for your PXE menus

mkdir /tftpboot/pxelinux.cfg

05. For each "Release" and "ARCH" Copy vmlinuz and initrd.img from /images/pxeboot/ directory on "disc 1" of that $Release/$ARCH to /tftpboot/images/RHEL/$ARCH/$RELEASE

mkdir -p /tftpboot/images/RHEL/i386/4.3

mkdir -p /tftpboot/images/RHEL/i386/5.5

mkdir -p /tftpboot/images/RHEL/x86\_64/4.3

mkdir -p /tftpboot/images/RHEL/x86\_64/5.5

For RHEL 5.5 x86\_64, do the following

mount /dev/cdrom /cdrom

cd /cdrom/images/pxeboot

cp vmlinuz initrd.img /tftpboot/images/RHEL/x86\_64/5.5

Do the above for all releases and ARCH you want to kickstart from this server.

06. Add this to your existing or new /etc/dhcpd.conf.  
Note: xxx.xxx.xxx.xxx is the IP address of your PXE server

allow booting;

allow bootp;

option option-128 code 128 = string;

option option-129 code 129 = text;

next-server xxx.xxx.xxx.xxx;

filename "/pxelinux.0";

07. Restart DHCP service

# service dhcpd restart

08. Create Simple or Multilevel PIXIE menu. Create a file called "default" in /tftpboot/pxelinux.cfg directory. A Sample file named "isolinux.cfg" is found on the boot installation media in "isolinux" directory. Copy this file as default and edit this file as per requirement. A sample default file is given bellow.

default menu.c32

prompt 0

timeout 300

ONTIMEOUT local

MENU TITLE PXE Menu

LABEL Pmajic

MENU LABEL Pmajic

kernel images/pmagic/bzImage

append noapic initrd=images/pmagic/initrd.gz root=/dev/ram0 init=/linuxrc ramdisk\_size=100000

label Dos Bootdisk

MENU LABEL ^Dos bootdisk

kernel memdisk

append initrd=images/622c.img

LABEL RHEL 5 x86 eth0

MENU LABEL RHEL 5 x86 eth0

KERNEL images/RHEL/x86/5.5/vmlinuz

APPEND initrd=images/RHEL/x86\_64/5.5/initrd.img ramdisk\_size=10000

ks=nfs:xx.xx.xx.xxx:/ ksdevice=eth1

LABEL RHEL 5 x86\_64 eth0

MENU LABEL RHEL 5 x86\_64 eth0

KERNEL images/RHEL/x86\_64/5.5/vmlinuz

APPEND initrd=images/RHEL/x86\_64/5.5/initrd.img ramdisk\_size=10000

ks=nfs:xx.xx.xx.xxx:/ ksdevice=eth1

09. Install the kickstart Configurator tool. This tool will be helpful to create the kickstart configuration file.

yum install system-config-kickstart

10. Create the kickstart config file. This file can be created using kickstart Configuration Tool. A Sample file anaconda-ks.cfg based on current installation of a system is placed in /root directory. We can also use this /root/anaconda-ks-cfg as the configuration file. Copy this file to the location specified in the default file. Make sure the directory is NFS exported if you are using NFS for installing the OS.

11. Modify the kickstart configuration file as per requirement. If you are using NFS for installation, Make sure to copy the ISO images of Linux disks to any NFS server and NFS export the directory. This server/directory details need to be specified in the jumpstart configuration file.

12. After creating the KS configuration files and copying the ISO images, the installation can be started.

1. Login to the CentOS server using Root account.
2. Mount the CentOS DVD:  
   mount /dev/cdrom /media
3. Move to the CentOS RPM folder inside the DVD:  
   cd /media/CentOS
4. Run the command bellow to install the TFTP-Server:  
     
   rpm -ivh xinetd-2.3.14-10.el5.i386.rpm  
   rpm -ivh tftp-server-0.49-2.el5.centos.i386.rpm
5. Run the command bellow to install the DHCP server:  
   rpm -ivh dhcp-3.0.5-23.el5.i386.rpm
6. Create new folder for the Kickstart server:  
   mkdir -p /data/kickstart
7. Edit using VI, the file /etc/xinetd.d/tftp and change the following settings:  
   From:  
   disable = yesTo:  
   disable = noFrom:  
   server\_args = -s /tftpbootTo:  
   server\_args = -s /data/kickstart
8. Run the command bellow to start the TFTP server:  
   /sbin/service xinetd start
9. Run the command bellow to start the TFTP server run at startup:  
   chkconfig xinetd on
10. Edit using VI, the file /etc/dhcpd.conf and add the following lines:  
    ddns-update-style none;  
    allow bootp;  
    allow booting;  
    subnet 10.1.1.0 netmask 255.255.255.0 {  
    option routers 10.1.1.254;  
    option domain-name-servers 10.1.1.2;  
    next-server 10.1.1.1;  
    filename "pxelinux.0";  
    range dynamic-bootp 10.1.1.200 10.1.1.210;  
    }Note 1: Replace 10.1.1.0 with the correct network ID.  
    Note 2: Replace 255.255.255.0 with the correct subnet mask.  
    Note 3: Replace 10.1.1.254 with the correct default gateway.  
    Note 4: Replace 10.1.1.1 with the Kickstart server IP address.  
    Note 5: Replace 10.1.1.200 with the first IP of the DHCP pool.  
    Note 6: Replace 10.1.1.210 with the last IP of the DHCP pool.  
    Note 7: Replace 10.1.1.2 with the correct DNS server.
11. Start the DHCP server  
    service dhcpd start
12. Run the command bellow to start the DHCP server run at startup:  
    chkconfig dhcpd on
13. Copy Boot Files  
    cp /usr/lib/syslinux/{pxelinux.0,menu.c32,memdisk,mboot.c32,chain.c32} /data/kickstart
14. Create a folder for the PXE menu files:  
    mkdir -p /data/kickstart/pxelinux.cfg
15. Move to the CentOS DVD root folder:  
    cd /media
16. Copy vmlinuz and initrd.img from the DVD to the images directory:  
    cp /media/images/pxeboot/{vmlinuz,initrd.img} /data/kickstart/images
17. Create the CentOS DVD structure:  
    cp -r CentOS /data/kickstart/  
    cp -r isolinux /data/kickstart/  
    cp -r repodata /data/kickstart/  
    cp -r images /data/kickstart/
18. Create using VI, the file /data/kickstart/pxelinux.cfg/default with the following content:  
    default menu.c32  
    prompt 0  
    MENU TITLE PXE Menu  
    LABEL CentOS  
    MENU LABEL CentOS  
    KERNEL images/vmlinuz  
    append initrd=images/initrd.img vga=normal network ks=nfs:10.1.1.1:/data/kickstart/ks.cfg textNote: Replace 10.1.1.1 with the Kickstart server IP address.
19. Create an unattended installation script /data/kickstart/ks.cfg  
    Note: Make sure the file starts with the following lines:  
    install  
    nfs --server=10.1.1.1 --dir=/data/kickstartNote 1: Replace 10.1.1.1 with the Kickstart server IP address.  
    Note 2: Make sure the lines beginning with “cdrom” and “url” does not exist on the file.  
    Note 3: To review ks.cfg file options, see the link:  
    <http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.4/html/Installation_Guide/s1-kickstart2-options.html>
20. Edit using VI, the file /etc/exports and add the following line:  
    /data/kickstart \*(ro,no\_root\_squash)
21. Start the NFS service:  
    service portmap start  
    service nfs start  
    chkconfig nfs on

--------------------------------------------------------------------------

Swapping:

Whole process is moved from the swap device to the main memory for execution. Process size must be less than or equal to the available main memory. It is easier to implementation and overhead to the system. Swapping systems does not handle the memory more flexibly as compared to the paging systems.

Paging:

Only the required memory pages are moved to main memory from the swap device for execution. Process size does not matter. Gives the concept of the virtual memory. It provides greater flexibility in mapping the virtual address space into the physical memory of the machine. Allows more number of processes to fit in the main memory simultaneously. Allows the greater process size than the available physical memory. Demand paging systems handle the memory more flexibly.

# Understanding UNIX / Linux symbolic (soft) and hard links

Inodes are associated with precisely one directory entry at a time. However, with hard links it is possible to associate multiple directory entries with a single inode. To create a hard link use ln command as follows:  
# ln /root/file1 /root/file2  
# ls -l  
Above commands create a link to file1. Symbolic links refer to:

A symbolic path indicating the abstract location of another file.

Hard links refer to:

The specific location of physical data.

## Hard link vs. Soft link in Linux or UNIX

* Hard links cannot link directories.
* Cannot cross file system boundaries.

Soft or symbolic links are just like hard links. It allows to associate multiple filenames with a single file. However, symbolic links allows:

* To create links between directories.
* Can cross file system boundaries.

These links behave differently when the source of the link is moved or removed.

* Symbolic links are not updated.
* Hard links always refer to the source, even if moved or removed.

## How do I create symbolic link?

You can create symbolic link with ln command:  
$ ln -s /path/to/file1.txt /path/to/file2.txt  
$ ls -ali  
Above command will create a symbolic link to file1.txt.

### Task: Symbolic link creation and deletion

Let us create a directory called foo, enter:  
$ mkdir foo  
$ cd foo  
Copy /etc/resolv.conf file, enter:  
$ cp /etc/resolv.conf .  
View inode number, enter:  
$ ls -ali  
Sample output:

total 152

1048600 drwxr-xr-x 2 vivek vivek 4096 2008-12-09 20:19 .

1015809 drwxrwxrwt 220 root root 143360 2008-12-09 20:19 ..

1048601 -rwxr-xr-x 1 vivek vivek 129 2008-12-09 20:19 resolv.conf

Now create soft link to resolv.conf, enter:  
$ ln -s resolv.conf alink.conf  
$ ls -ali  
  
Sample output:

total 152

1048600 drwxr-xr-x 2 vivek vivek 4096 2008-12-09 20:24 .

1015809 drwxrwxrwt 220 root root 143360 2008-12-09 20:19 ..

1048602 lrwxrwxrwx 1 vivek vivek 11 2008-12-09 20:24 alink.conf -> resolv.conf

1048601 -rwxr-xr-x 1 vivek vivek 129 2008-12-09 20:19 resolv.conf

The reference count of the directory has not changed (total 152). Our symbolic (soft) link is stored in a different inode than the text file (1048602). The information stored in resolv.conf is accessible through the alink.conf file. If we delete the text file resolv.conf, alink.conf becomes a broken link and our data is lost:  
$ rm resolv.conf  
$ ls -ali

**What is the core of Linux Operating System?**

The core of the Linux operating system is Kernel. It is broken down into Shell, Command, Script, and Terminal. Shell is a command Line Interpreter, Command is user Instruction to Computer, Script is collection of commands stored in a file, and Terminal is a command Line Interface.

**What is the basic difference between UNIX and Linux Operating System?**

Linux is free and open-source software (allowing programmers to program with Linux not around it), the kernel of which is created by Linus Torvalds and community. UNIX, on the other hand, is UNIX is copyrighted name only big companies are allowed to use the UNIX copyright and name, so IBM AIX and Sun Solaris and HP-UX all are UNIX operating systems.

**What is an INODE?**

All files have its description stored in a structure called ‘inode’. The inode contains info about the file-size, access and modification time, permission and so on. In addition to descriptions about the file, the inode contains pointers to the data blocks of the file.

**State the syntax of any Linux command.**

The correct syntax of Linux command is Command [options] [arguments]. [Master the Linux command line with this guide](https://www.udemy.com/mastering-the-linux-command-line/?tc=blog.linuxinterviewquestions&utm_source=blog&utm_medium=udemyads&utm_content=post27738&utm_campaign=content-marketing-blog&xref=blog).

Now let’s move on to the meatier questions that are more likely to be asked:

**What is the difference between TCP and UDP?**

The basic difference is that TCP establishes a connection before sending data and this allows it to control the dataflow and guarantee that all packets get delivered. UDP simply chucks datagrams onto the wire and if some get lost or arrive in bad order there’s no way to request a resend. However UDP has low network overhead so some services such as DNS resolution, SNMP, DHCP, RIP and VOIP use UDP for its speed and any errors are usually dealt with on the application layer rather than network layer.

**How does DNS resolution work?**

A client application requests an IP address from the name server usually by connecting to UDP port 53. The name server will attempt to resolve the FQDN based on its resolver library, which may contain authoritative information about the host requested or cached data about that name from an earlier query. If the name server does not already have the answer, it will turn to root name servers to determine the authoritative for the FQDN in question. Then, with that information, it will query the authoritative name servers for that name to determine the IP address.

**What is an MX record?**

An MX record numerically ranks the mail servers you would prefer to receive email for a domain. The MX record with the lowest number is preferred over the others, but you can set multiple email servers with the same value for simple load balancing.

**Please describe the Linux boot-up sequence.**

There are seven steps to the boot-up sequence. 1) BIOS (basic input/output system) – executes the MBR where Boot Loader sits, 2) MBR- Master boot reads Kernel into memory, 3) GRUB (Grand Unified Bootloader) Kernel starts Init process, 4) Kernel – Kernel executes the /sbin/init program.  Init reads inittab, executes rc.sysinit, 5) Init – the rc script than starts services to reach the default run level and 6) Run level programs – these programs are executed from /etc/rc.d/rc\*.dl/

**How do you search for a pattern and then replace it in an entire file?**

You use Sed, or in Vi editor, the search uses character ‘s’ slash the pattern to be searched, slash the pattern to replace it with, slash ‘g’ which stands for entire file.

**How do you list and flush all IPtables?**

First you use the –L switch to view all the currently present rules and then –F to flush them.

**What is a shell? What are their names?**

The shell is the part of the system with which the user interacts. A Unix shell interprets commands such as “pwd”, “cd” or “traceroute” and sends the proper instructions to the actual operating system itself. The shells currently available areAns SH, BASH, CSH, TCSH, NOLOGIN, KSH. Other functions of a shell include scripting capability, path memory, multitasking, and file handling.

**What is a zombie?**

Cheeky answers get bonus points for this one. But in the Linux world, a zombie process is the process  output of ‘ps’ by the presence of ‘Z’ in the STAT column. Zombies are essentially the premature processes whose mature parent processes died without reaping its children. Note that zombies can’t be killed with the usual ‘kill’ signal.

## ****What is SGID?****

**SGID** (**S**et **G**roup **ID** up on execution) is a special type of file permissions given to a file/folder. Normally in Linux/Unix when a program runs, it inherits access permissions from the logged in user. SGID is defined as giving temporary permissions to a user to run a program/file with the permissions of the file group permissions to become member of that group to execute the file**. In simple words users will get file Group’s permissions when executing a Folder/file/program/command.**

SGID is similar to SUID. The difference between both is that SUID assumes owner of the file permissions and SGID assumes group’s permissions when executing a file instead of logged in user inherit permissions.

## ****Learn SGID with examples:****

**Example:** [**Linux Group quota implementation**](http://www.linuxnix.com/2010/03/how-to-linux-group-disk-quota-implementation.html)

When implementing Linux Group quota for group of people SGID plays an important role in checking the quota timer. SGID bit set on folder is used to change their inherit permissions to group’s permissions to make it as single user who is dumping data. So that group members whoever dumps the data the data will be written with group permissions and in turn quota will be reduced centrally for all the users. For clear understanding of this you have to implement group quota from the above link. Without implementation of SGID the quota will not be effective.

## ****How can I setup SGID for a file?****

SGID can be set in two ways

**1) Symbolic way (s)**

**2) Numerical/octal way (2, SGID bit as value 2)**

Use [**chmod** command](http://www.linuxnix.com/2011/10/chmod-command-explained-linuxunix.html) to set SGID on file: file1.txt

**Symbolic way:**

**chmod g+s file1.txt**

Let me explain above command we are setting SGID(+s) to group who owns this file.

**Numerical way:**

**chmod 2750 file1.txt**

**Here in 2750, 2 indicates SGID bitset, 7 for full permissions for owner, 5 for read and execute permissions for group, and no permissions for others.**

**How can I check if a file is set with SGID bit or not?**

Use ls –l to check if the x in group permissions field is replaced by s or S

For example: file1.txt listing before and after SGID set

**Before SGID set:**

**ls -l**

**total 8**

**-rwxr--r-- 1 xyz xyzgroup 148 Dec 22 03:46 file1.txt**

**After SGID set:**

**ls -l**

**total 8**

**-rwxr-sr-- 1 xyz xyzgroup 148 Dec 22 03:46 file1.txt**

**Some FAQ’s related to SGID:**

**Where is SUID used?**

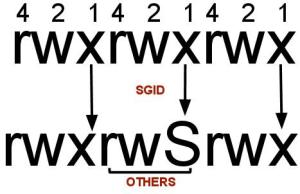
1) When implementing Linux group disk quota.

**I am seeing “S” ie Capital s in the file permissions, what’s that?**

After setting SUID or SGID to a file/folder if you see ‘S’ in the file permission area that indicates that the file/folder does not have executable permissions for that user or group on that particular file/folder.

**chmod g+s file1.txt**

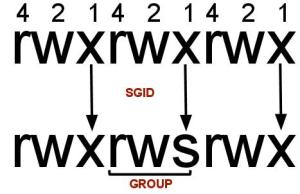
output:  
-rwxrwSr-x 1 surendra surendra 0 Dec 27 11:24 file1.txt

[](http://i0.wp.com/www.linuxnix.com/wp-content/uploads/2011/12/Untitleddrawing-6.jpg)

so if you want executable permissions too, apply executable permissions to the file.

**chmod g+x file1.txt**

output:  
-rwxrwsr-x 1 surendra surendra 0 Dec 5 11:24 file1.txt

[](http://i1.wp.com/www.linuxnix.com/wp-content/uploads/2011/12/Untitleddrawing-3.jpg)

you should see a smaller 's' in the executable permission position.

**How can I find all the SGID set files in Linux/Unix.**

[**find**](http://www.linuxnix.com/2012/04/learn-linuxunix-find-command-50-practical-examples-part-i.html) **/ -perm +2000**

The above find command will check all the files which is set with SGID bit(2000).

**Can I set SGID for folders?**

Yes, you can if it’s required (you should remember one thing, that Linux treats everything as a file)

How can I remove SGID bit on a file/folder?

**chmod g-s file1.txt**

**Sticky Bit(t)**: Sticky bit is very simple and effective file permission; it increases security of a file/directory which is shared with other users. When sticky bit is enabled, only user (owner) of that file can remove or rename the file even if other users have full (rwx) permissions on that file. In the case of a directory, only user (owner) of the directory or the owner of the file in that directory can remove or rename the file. Mainly sticky bit is used on directories on which multiple users have access like /tmp. By default sticky bit is set on /tmp in Redhat Enterprise Linux 6(RHEL6).

In the above example we can see that there is a “t” at execute permission for others. Sticky bit can be enabled using “chmod” command. Let’s take some examples of Sticky Bit.

* Add sticky bit permission on a directory with all permissions using symbolic chmod.

[root@PawanS1 ~]# chmod +t Test\_Dir/  
  
[root@PawanS1 ~]# ls -ld Test\_Dir/  
drwxrwxrwt  2  pawan admin  4096 Aug 28 10:22   Test\_Dir/

**SUID (Set User ID) Bit(s)**: Mainly we enable SUID bit on files specially on executable scripts. When SUID bit is enabled on the script/ file, whenever someone executes the file it runs as the user who is owner of that file. It means the file is ensured to run as the owner, even if executed by anyone. This comes handy when you want to give execute rights of a root privileged script to some other user. In RHEL 6, SUID bit is set by default on commands like /usr/bin/passwd, /usr/bin/wall, /usr/bin/ssh-agent, etc. This is the reason a user can change its password itself.

In the above example we can see that there is a “s” at execute permission of user (owner). SUID bit can be enabled using “chmod” command. Let’s take some examples of SUID bit.

* Add SUID bit on a script using symbolic chmod.

[root@PawanS1 ~]# ls -l test\_script.sh  
-rwxr-xr-x 1 root admin 43 Aug 28 11:51  test\_script.sh  
  
[root@PawanS1 ~]# chmod u+s test\_script.sh  
  
[root@PawanS1 ~]# ls -l test\_script.sh  
-rwsr-xr-x 1 root admin 43 Aug 28 11:51  test\_script.sh

* Add SUID bit on a script which does not have execute permission for user (owner) using numeric chmod.

[root@PawanS1 ~]# ls -l my\_script.sh  
-rw-r--r-- 1 root admin 29 Aug 28 11:58   my\_script.sh  
  
[root@PawanS1 ~]# chmod  4644 my\_script.sh  
  
[root@PawanS1 ~]# ls -l my\_script.sh  
-rwSr--r-- 1 root admin 29 Aug 28 11:58   my\_script.sh

**Note**: This time we have a “S” instead of “s” because the script “my\_script.sh does not have execute permission for user.

**SGID (Set Group ID) Bit**: SGID bit is very useful when you have to give access of a directory to a set of users in a group. When SGID bit is enabled on a directory any file/directory created under it by any user have the same group permissions as of the parent directory.

For example, you have created a group named “sales” and you have added three user pawan, siddharth, ramswaroop and usaid in group “sales”. Now you want that every file created by any of these four users under directory “/Sales” can be accessible by any of these users.

1. To do this first you have to create a directory “/Sales” and then change group owner and group permission to sales and rwx respectively.

drwxrwsr-x 2 root root 4096 Aug 28 12:31 /Sales/

In the above example we can see that there is a “s” at execute permission of group. Now any file created under directory “/Sales” will have group user sales

Now login as user pawan and create a file in /Sales and check its permissions.

* We can also enable SGID bit using chmod in numeric mode.

[root@PawanS1 ~]# ls -d /Purchase/  
drwxrw-r-x 2 root purchase 4096 Aug 28 12:31 /Purchase/  
  
[root@PawanS1 ~]# chmod 2765 /Purchase/  
  
[root@PawanS1 ~]# ls -d /Purchase/  
drwxrwSr-x 2 root purchase 4096 Aug 28 12:31 /Purchase/

**Note**: This time we have a “S” instead of “s” because directory “/Purchase” does not have execute permission for group.

Below table summarize the chmod for SUID, SGID and Sticky Bit.

|  |  |  |
| --- | --- | --- |
| **Permission** | **Symbolic Mode** | **Numeric Mode** |
| **Sticky Bit** | chmod +t file\_name | chmod 1XXX file\_name |
| **SUID Bit** | chmod u+s file\_name | chmod 4XXX file\_name |
| **SGID Bit** | chmod g+s file\_name | chmod 2XXX file\_name |
| where X is permission for user,group and other | | |

1. How to check Gateway and net mask??

Ans. netstat -ar

2. RSYNC command executed, got error while copying, again the command initiated, whether the file copies from 1st and incremental.

Ans:- The rsync command is incremental copy(check the files with destination and copies rest of the part).

3. In RAID 5, two disks are corrupted what to do ?

Ans:- In RAID 5 , if one disk is corrupted , we can replace that effective drive without any downtime. But here the scenario is  two disks are corrupted , in this case we don’t have much option left. Need to replace all the effective disks and rebuild the OS. If we kept two spare hard disk while creating RAID , the situation will be different.

4.what is hard and soft mounting?

Ans.

    Hard mount:

— If the NFS file system is hard mounted, the NFS daemons will try repeatedly to contact the server. The NFS daemon retries will not time out, will affect system performance, and you cannot interrupt them.

   Soft mount

— If the NFS file system is soft mounted, NFS will try repeatedly to contact the server until either:

A connection is established

The NFS retry threshold is met

The nfstimeout value is reached

When one of these events occurs, control returns to the calling program.

5.what is /proc file system ?

Ans. procfs (or the proc file system  is a special file system in UNIX-like operating systems that presents information about processes and other system information in a hierarchical file-like structure, providing a more convenient and standardized method for dynamically accessing process data held in the kernel than traditional tracing methods or direct access to kernel memory.

6.what is daemon responsible for tracking events in a server?

Ans. syslogd

7.what is difference between raid 0 and raid 1?

Ans. RAID 0, No redundancy

       RAID 1, Redundancy

8.what is kernel panic?

Ans. A kernel panic is an action taken by an operating system upon detecting an internal fatal error from which it cannot safely recover. The term is largely specific to Unix and Unix-like systems; for Microsoft Windows operating systems the equivalent term is "bug check" (or "Blue Screen of Death").

9.what is '0' process?

Ans. parent process id of 'init' (sched process)

10.what is the last service started by init before logging screen appears?

Ans. /etc/rc.local which are the last commands run in initialization process or even booting process

11.difference between LILO &GRUB?

Ans. LILO, Linux Loader, on kernel versions < 2.4,It does'nt support booting from n/w.

     Grub, Grand unified boot loader. on Kernel versions >= 2.6, supports booting from network.

12.Stages of boot loader?

Ans.6 stages.

1. BIOS

2. MBR

3. GRUB

4. Kernel

5. Init

6. Runlevels

13.when two machines are there, one machine MAC & IP address is known,  another machine MAC address is there, how to find IP of another machine using command?

Ans. 'arp' command

14.while 'ping' cmd is used  the system not receiving any o/p?

Ans. ping cmd is blocked in kernel parameters

     ping  cmd uses icmp protocol.

15.Issue is there with hard drives ,dont know  which drive is fault. how to check which drive is fault?

Ans. If it is HP and IBM Servers, it will show amber color for faulty HDD.  otherwise we can see the status of the hard disks in ILO (Integrated Lights   Out)configuration for HP servers and the same remote accessing feature also available for IBM. other wise you can check in dmesg .

16.In production server one drive got failed,how to replace new drive?

Ans. plug and play devices, in raid -1 mirror is used machine can run with one hard disk . we can replace

17.ASR means AUTOMATED SERVER RECOVERY

18. What is Network bonding? Requirements for bonding?

Ans. N/w Bonding will configure on the servers for redundancy. It requires minimum 2 Ethernet Ports/Cards.

20. Different mount options apart from mount command?

Ans. crazy question, i will say /bin/mount., But it is mount command full form. I heared guestmount commnad also there to mount a disk ...but didn't used yet.

22.how to start services apart from 'service' cmd?

Ans. /ect/init.d/<service name> start

23.how to display memory info?

Ans. cat /proc/meminfo

   free -go (To see in GB)

24.how to display hardware information?

Ans. dmidecode |less

     Command: lscpu

List available cpus and their caracteristics , Not available on older distribution

     Command: lshal

         Require HAL (Hardware Abstraction Layer) to be installed . List all hardware visible by HAL

     Command: lshw

          Available on Debian and Ubuntu based distributions by default .Available in the Fedora repositories .Uses many inputs            to detect all hardware: Kernel, HAL, DMI, etc. use ‘-html’ switch that generates hardware reports

     Command: lspci

     Standard command ,List all hardware connected to the PCI bus as detected by the kernel

     Command: lsusb

      Standard command. List all hardware connected to the USB buses as detected by the kernel

     Command: dmidecode

      Standard command, Get the source information from the DMI (a kind of BIOS interface) .List all hardware as reported by       the DMI interface

25. command to check the directory's partition or mount ?

Ans. df -P file/goes/here | tail -1 | cut -d' ' -f 1

26.how to rollback application?

Ans.  insert 'ts\_flags=repackage' in /etc/yum.conf and create file /etc/rpm/macros with an entry'repackage' parameter.

27.how to rollback filesystem?

Ans. If 'Snapshot'(backup) is taken, using superblocks we can rollback file system.

28. Newly attached hard drive is not recognizing how to make it active ?

Ans. After building server with 2 hard disks, again if we attach any hdd, server wont recognizes it. if the server want to recognize it, we need to reboot it and create RIAD fo that HDD also.

If we replaced Faulty hard disk with new one, Server will take 1 hr time to synchronize the data (RIAD 1 and RAID 5).

29.Different raid levels & explanation?

Ans. raid 0, raid1, raid1+0, raid 1+0, raid3, raid 5, raid50, raid6.

30.file system is full? user need some space to run his work ?

Ans. 1. If it is configured with 'lvm' we can extend it, or else we can make use of reserved blocks.

     2. Delete the old files after getting confirmation from user.

31. How to check a package is installed or not?

Ans.  rpm -qa | grep <package name>, it will gives package name and version, if it installed, returns empty screen if not installed.

32.how to check which ports are working?

Ans. netstat cmd

33.how to configure static route?

Ans. vi /etc/sysconfig/network-scripts/ifcfg-route0

34. How to check the processor type ?

Ans. grep "model name" /proc/cpuinfo

35.what is GRUB?

Ans. Grand unified Boot loader. Its boot loader in linux. (kernel versions >=2.6)

36.how to check with hard drives ?

Ans. df or mount

37.difference between rpm and yum ?

Ans.

### 1] What Is RPM?Redhat Package Manager, shortly known as RPM is like setup file, somewhat similar to “.exe” files we have in Windows. RPM files are the packages which will install the program in your computer. While commands are very basic and simple to install and uninstall the program, it sometime gets difficult to actually find the links to download these RPM (Linux is as user friendly as a computer can get). 2] What Is YUM?Yellowdog Updater Modified, shortly known as YUM is like a Library which has all the RPMs indexed in it. Since we’re using Linux, we’re cool, we don’t want to waste our time finding the setup files. Hence, we have an entire Library that already has all the RPMs indexed in it. All we need to do is execute the command “yum install package” where package is the software you’re looking to install. Isn’t it cool? No need to Google around to find your programs. YUM is somewhat like Google for Linux programs. You can even search the possible software with commands like “yum list package” “yum search package” etc.Hence, you can not possibly find any DIFFERENCE between RPM and YUM because they are NOT competitors. They work with each other. You would most likely end up using YUM in order to install, uninstall the programs. In case YUM doesn’t have your program indexed, then you will need to download the RPM package from the internet in order to install it.

38.How to increase physical memory in steps?

Ans. Hard ware Part, Need to get down time form the users or customers , upagrde the memeory . (memory size should be match with old memory modules.)

40.kernel patching?

Ans.

41.how to mount alternate superblock command?

Ans. mount  sb=alternative superblock  /dev/sda

42.how to create a never expire passwd for user?

Ans. chage -E -1 username

43.tune2fs command used for ?

Ans. for tuning  the filesystem parameters

44.In samba share directory is sharing, everything is access apart  from soft link files ,how to share those soft links shares?

Ans. Apply parameters in share definition

         follow symlinks =yes

         wide links =yes

46.what is the way to set dump and fsck options in /etc/fstab?

Ans.

47.maximum file grow in ext4 filesystem?

Ans.

48.difference between linear and mirror volumes?

Ans.

49.scenerio

# df -h

……………. file system full

#ls -l

two files…..1. one file is occupying full space

# rm -f filename

#df  -h

same output …file system is full

#ls -l

file is not there ..still file system is full.

Ans.

 # lsof | grep "(deleted)" # (or lsof | grep (filename))

If it's safe to do so, take the pid from the lsof command, and do:

Code:

# ls -l /proc/(pid)/fd # to get the fd# for the file

# > /proc/(pid)/fd/(fd#) # to empty the file.

Otherwise the space will be freed when the application using it closes, or the system restarts.

# implies a root prompt, meaning your sysadmin needs to do this.

(S)he will need to install lsof if it's not already installed.

50.what is webstack? how to improve performance and bottlenecks?

Ans.

51. No such file or directory error?

Ans. ls -ld <filename>, error, no such file or directory..

  Reasons: Check it , by logging as a root. might be noral user not having the permissions to view it.

   we will get it, if the directory or file is unmounted or deleted.

52.how to check database and other applications running?

Ans. ps -ef | grep <database name/ application name>

53.mpstat, iostat,vmstat statistics define?

Ans.

   1. iostat – Basic example

Iostat without any argument displays information about the CPU usage, and I/O statistics about all the partitions on the system as shown below.

 #iostat

Linux 2.6.32-100.28.5.el6.x86\_64 (dev-db)       07/09/2013

avg-cpu:  %user   %nice %system %iowait  %steal   %idle

           5.68    0.00    0.52    2.03    0.00   91.76

Device:            tps   Blk\_read/s   Blk\_wrtn/s   Blk\_read   Blk\_wrtn

sda             194.72      1096.66      1598.70 2719068704 3963827344

sda1            178.20       773.45      1329.09 1917686794 3295354888

sda2             16.51       323.19       269.61  801326686  668472456

   vmstat : vmstat by default will display the memory usage (including swap) as shown below.

#vmstat

procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu------

 r  b   swpd   free   buff  cache     si   so    bi    bo   in   cs us sy id wa st

 0  0 305416 260688  29160 2356920    2    2     4     1    0    0  6  1 92  2  0

Procs – r: Total number of processes waiting to run

Procs – b: Total number of busy processes

Memory – swpd: Used virtual memory

Memory – free: Free virtual memory

Memory – buff: Memory used as buffers

Memory – cache: Memory used as cache.

Swap – si: Memory swapped from disk (for every second)

Swap – so: Memory swapped to disk (for every second)

IO – bi: Blocks in. i.e blocks received from device (for every second)

IO – bo: Blocks out. i.e blocks sent to the device (for every second)

System – in: Interrupts per second

System – cs: Context switches

CPU – us, sy, id, wa, st: CPU user time, system time, idle time, wait time

Mpstat – Display basic info

By default mpstat displays CPU statistics.

54.how to list newly attached hardware?

Ans. dmidecode | less

55.In RAID , statistics display  "\_U" .what is the meaning?

Ans. MeAns one of the hard disk is failed.

56.what is initrd? what it does?

Ans. initrd stands for Initial RAM Disk. initrd is used by kernel as temporary root file system until kernel is booted and the real root file system is mounted. It also contains necessary drivers compiled inside, which helps it to access the hard drive partitions, and other hardware parts of the server.

58.where /proc file system rests?

Ans. Linux systems store all data as files. Most users are familiar with the two primary types of files: text and binary. But the /proc/ directory contains another type of file called a virtual file. As such, /proc/ is often referred to as a virtual file system.

59.where tmpfs stores?

Ans. Tmpfs is a file system which keeps all files in virtual memory. Everything in tmpfs is temporary in the sense that no files will be created on your hard drive. If you unmount a tmpfs instance, everything stored therein is lost. tmpfs lives completely in the page cache and on swap, all tmpfs pages currently in memory will show up as cached

60.Daemons in nfs?

Ans. mountd ,nfsd,lockd,statd,nfslogd

61.how to change user access to entire file system?

Ans. chown -R

62.why UMASK is used, where to change its value to be effective on entire OS?

Ans. When user create a file or directory under Linux or UNIX, it creates with a default set of permissions. In most case the system defaults may be open or relaxed for file sharing purpose. For example, if a text file has 666 permissions, it grants read and write permission to everyone. Similarly a directory with 777 permissions, grants read, write, and execute permission to everyone.

You can setup umask in /etc/bashrc or /etc/profile file for all users. By default most Linux machines set it to 0022 (022) or 0002 (002).

Open /etc/profile or ~/.bashrc file, enter:

# vi /etc/profile

OR

# vi ~/.bashrc

Append/modify following line to setup a new umask:

umask 022

Save and close the file. Changes will take effect after next login. All UNIX users can override the system umask defaults in their /etc/profile file, ~/.profile (Korn / Bourne shell) ~/.cshrc file (C shells), ~/.bash\_profile (Bash shell) or ~/.login file (defines the user's environment at login).

63.how to display the list of specific  port ?

Ans. netstat -ntlp | grep <service daemon name>

68.what is SFTP?

Ans.secure file transfer protocol, a network protocol for secure file transfer over a secure shell.

69.Top responding slowly..other option to check load average?

Ans. vmstat

71.Difference between crontab and at ?

list crontab entry?

Ans.Cron command is used to schedule the task daily at the same time repeatedly ,  
"at" command is used toschedule the task only once i.e to run only one time

|  |
| --- |
| Crontab -l |

72.what is MTA?

Ans. Mail Trannsfer Agent

73.while install rpm package, progress reports already use , but  no package name doesn't exist in rpm  list ?

Ans. rpm --reinstall <package name>

74.create user in different directory?

Ans.useradd -m -d  /newdir/<username>  username

## Device Mapper Multipathing

### Procedure for configuring the system with DM-Multipath:

### Install *device-mapper-multipath* rpm

### Edit the multipath.conf configuration file:

### comment out the default blacklist

### change any of the existing defaults as needed

### Start the multipath daemons

### Create the multipath device with the *multipath*

### Install Device Mapper Multipath

# rpm -ivh device-mapper-multipath-0.4.7-8.el5.i386.rpm

warning: device-mapper-multipath-0.4.7-8.el5.i386.rpm: Header V3 DSA signature:

Preparing... ########################################### [100%]

1:device-mapper-multipath########################################### [100%]

### Initial Configuration

### Set *user\_friendly\_name*. The devices will be created as */dev/mapper/mpath[n]*. Uncomment the blacklist.

# vim /etc/multipath.conf

#blacklist {

# devnode "\*"

#}

defaults {

user\_friendly\_names yes

path\_grouping\_policy multibus

}

### Load the needed modul and the startup service.

# modprobe dm-multipath

# /etc/init.d/multipathd start

# chkconfig multipathd on

### Print out the multipathed device.

# multipath -v2

or

# multipath -v3

### Configuration

### Configure device type in config file.

# cat /sys/block/sda/device/vendor

HP

# cat /sys/block/sda/device/model

HSV200

# vim /etc/multipath.conf

devices {

device {

vendor "HP"

product "HSV200"

path\_grouping\_policy multibus

no\_path\_retry "5"

}

}

### Configure multipath device in config file.

# cat /var/lib/multipath/bindings

# Format:

# alias wwid

#

mpath0 3600508b400070aac0000900000080000

# vim /etc/multipath.conf

multipaths {

multipath {

wwid 3600508b400070aac0000900000080000

alias mpath0

path\_grouping\_policy multibus

path\_checker readsector0

path\_selector "round-robin 0"

failback "5"

rr\_weight priorities

no\_path\_retry "5"

}

}

### Set not mutipathed devices on the blacklist. (f.e. local Raid-Devices, Volume Groups)

# vim /etc/multipath.conf

devnode\_blacklist {

devnode "^cciss!c[0-9]d[0-9]\*"

devnode "^vg\*"

}

### Show Configured Multipaths.

# dmsetup ls --target=multipath

mpath0 (253, 1)

# multipath -ll

mpath0 (3600508b400070aac0000900000080000) dm-1 HP,HSV200

[size=10G][features=1 queue\_if\_no\_path][hwhandler=0]

\\_ round-robin 0 [prio=4][active]

\\_ 0:0:0:1 sda 8:0 [active][ready]

\\_ 0:0:1:1 sdb 8:16 [active][ready]

\\_ 1:0:0:1 sdc 8:32 [active][ready]

\\_ 1:0:1:1 sdd 8:48 [active][ready]

### Format and mount Device

### Fdisk cannot be used with /dev/mapper/[dev\_name] devices. Use fdisk on the underlying disks and execute the following command when device-mapper multipath maps the device to create a /dev/mapper/mpath[n] device for the partition.

# fdisk /dev/sda

# kpartx -a /dev/mapper/mpath0

# ls /dev/mapper/\*

mpath0 mpath0p1

# mkfs.ext3 /dev/mapper/mpath0p1

# mount /dev/mapper/mpath0p1 /mnt/san

### After that */dev/mapper/mpath0p1* is the first partition on the multipathed device.

## Multipathing with mdadm on Linux

### The md multipathing solution is only a failover solution what means that only one path is used at one time and no load balancing is made. Start the MD Multipathing Service

# chkconfig mdmpd on

# /etc/init.d/mdmpd start

### On the first Node (if it is a shared device) Make Label on Disk

# fdisk /dev/sda

Disk /dev/sdt: 42.9 GB, 42949672960 bytes

64 heads, 32 sectors/track, 40960 cylinders

Units = cylinders of 2048 \* 512 = 1048576 bytes

Device Boot Start End Blocks Id System

/dev/sdt1 1 40960 41943024 fd Linux raid autodetect

# partprobe

### Bind multiple paths together

# mdadm --create /dev/md4 --level=multipath --raid-devices=4 /dev/sdq1 /dev/sdr1 /dev/sds1 /dev/sdt1

### Get UUID

# mdadm --detail /dev/md4

UUID : b13031b5:64c5868f:1e68b273:cb36724e

### Set md configuration in config file

# vim /etc/mdadm.conf

# Multiple Paths to RAC SAN

DEVICE /dev/sd[qrst]1

ARRAY /dev/md4 uuid=b13031b5:64c5868f:1e68b273:cb36724e

# cat /proc/mdstat

### On the second Node (Copy the */etc/mdadm.conf* from the first node)

# mdadm -As

# cat /proc/mdstat

### Restore a failed path

# mdadm /dev/md1 -f /dev/sdt1 -r /dev/sdt1 -a /dev/sdt1

### difference between ext2,ext3 and ext4

Ext3 file system is nothing but next version of ext2 file system with journaling support.

 Ext3 has been structurally implemented same as ext2 so they have same data structures.

 The most important difference between Ext2 and Ext3 is that Ext3 supports journaling which allows fast recovery from disk problems.

 You also get reliability and a better performance with ext3. Ext3 is designed to take care of both metadata and data.

The main benefit of ext3 is that it allows Journaling .

Journaling has a dedicated area in the file system, where all the changes are tracked. When the system crashes, the possibility of file system corruption is less because of journaling.

There are three types of journaling available in ext3 file system.

Journal – Metadata and content are saved in the journal.

Ordered – Only metadata is saved in the journal. Metadata are journaled only after writing the content to disk. This is the default.

Write back – Only metadata is saved in the journal. Metadata might be journaled either before or after the content is written to the disk.

you can convert  ext2 to ext3 without  having any data loss

ext4

In ext4, it supports journaling and also has the option of turning the journaling feature “off”.

Several other new features are introduced in ext4: multi block allocation, delayed allocation, Journal checksum, fast fsck and etc.

All you need to know is that these new features have improved the performance and reliability of the file system when compared to ext3.

Supports huge individual file size and overall file system size.

You can also mount the   existing ext3  filesystem as  ext4 file system without having to  upgrade it

  How to create a Volume Group (VG).

  How to create a Logical Volume (LV) and mount the file system.  
How to add a disk to a Volume Group

  How to increase the size of a logical volume without OnlineJFS

  How to remove a Logical Volume

  How to reduce the size of a logical volume without OnlineJFS (advanced JFS)

  How to remove a disk from a volume group

  How to remove a volume group

  How to increase the primary swap

  How to create a secondary boot disk LVM Mirroring

  How to mirror a logical volumeXII) How to unmirror a logical volume

  XIII) How to create a mirrored boot disk

  XIV) How to mirror a logical volume on a specific physical volume Physical Volume Group

  XV) How to create a Physical Volume Group (PVG)

  XVI) How to use PVG to mirror logical volumes on specific physical volumes.

**What is LVM?**  
  
LVM stands for Logical Volume Manager. LVM, is a storage management solution that allows administrators to divide hard drive space into physical volumes (PV), which can then be combined into logical volume groups (VG), which are then divided into logical volumes (LV) on which the filesystem and mount point are created.

**Q: - What are the steps to create LVM?**  
  
- Create physical volumes by “pvcreate” command

#pvcreate /dev/sda2  
- Add physical volume to volume group by “vgcreate” command

#vgcreate VLG0 /dev/sda2  
- Create logical volume from volume group by “lvcreate” command.

#lvcreate -L 1G -n LVM1 VLG0  
Now create file system on /dev/sda2 partition by “mke2fs” command.

#mke2fs -j /dev/VLG0/LVM1

**Q: - What is the difference between LVM and RAID?**  
  
RAID provides redundancy but LVM doesn’t provide Redundancy.

**Q: - What are LVM1 and LVM2?**  
  
LVM1 and LVM2 are the versions of LVM.  
LVM2 uses device mapper driver contained in 2.6 kernel version.  
LVM 1 was included in the 2.4 series kernels.

**Q: - What is Volume group (VG)?**  
  
The Volume Group is the highest level abstraction used within the LVM. It gathers together a collection of Logical Volumes and Physical Volumes into one administrative unit.

**Q: - What is physical extent (PE)?**  
  
Each physical volume is divided chunks of data, known as physical extents; these extents have the same size as the logical extents for the volume group.

**Q: - What is logical extent (LE)?**  
  
Each logical volume is split into chunks of data, known as logical extents. The extent size is the same for all logical volumes in the volume group.

**Q: - Explain LVM snapshot?**  
  
LVM snapshots allow the administrator to create a new block device which presents an exact copy of a logical volume, frozen at some point in time.

### monitoring commands

### iostat

The [iostat](http://www.cyberciti.biz/tips/linux-disk-performance-monitoring-howto.html) command shows in detail what your storage subsystem is up to. You usually use iostat to monitor how well your storage sub-systems are working in general and to spot slow input/output problems before your clients notice that the server is running slowly. Trust me, you want to spot these problems before your users do!

### meminfo and free

[Meminfo](http://www.redhat.com/advice/tips/meminfo.html) gives you a detailed list of what's going on in memory. Typically you access meminfo's data by using another program such as cat or grep. For example,

cat /proc/meminfo

gives you the details of what's going on in your server’s memory at any given moment.

For a quick “just the facts” look at memory, you can use the free command. In short,free gives you the overview; meminfo gives you the details.

### mpstat

The [mpstat](http://linuxcommand.org/man_pages/mpstat1.html) command reports on the activities of each of the available CPUs on a multi-processor server. These days, thanks to [multi-core processors](http://h30565.www3.hp.com/t5/Feature-Articles/What-Does-x86-Need-to-Compete-With-RISC/ba-p/1222), that’s almost all servers.mpstat also reports on the average activities of all your server's CPUs. It enables you to display overall CPU statistics per system or per processor. This overview can alert you to possible application problems before they get to the point of annoying users.

### netstat

[Netstat](http://www.thegeekstuff.com/2010/03/netstat-command-examples/), like ps, is a Linux tool that administrators use every day. It displays a lot of network related information, such as socket usage, routing, interface, protocol, network statistics, and more. Some of the most commonly used options are:

-a Show all socket information

-r Show routing information

-i Show network interface statistics

-s Show network protocol statistics

### nmon

[Nmon](http://nmon.sourceforge.net/pmwiki.php), short for Nigel's Monitor, is a popular [open-source](http://h30565.www3.hp.com/t5/Feature-Articles/Crafting-Small-Business-Open-Source-Policies/ba-p/1368) tool to monitor Linux systems performance. Nmon watches the performance information for several subsystems, such as processor utilization, memory utilization, run queue information, disk I/O statistics, network I/O statistics, paging activity, and process metrics. You can then view nmon's real-time system measurements via its curses “graphical” interface.

To run nmon, you start the tool from the shell. Once up, you select the subsystems to monitor by typing in its one-key commands. For example, to get CPU, memory, and disk statistics, you type c, m, and d. You can also use nmon with the -f flag to save performance statistics to a CSV file for later analysis.

For day to day server monitoring I find nmon to be the single most useful program in my Linux system management tool-kit.

### pmap

The [pmap](http://linuxcommand.org/man_pages/mpstat1.html) command reports the amount of memory that your server's processes are using. You can use this tool to determine which processes on the server are being allocated memory and whether any of these processes are being piggy with RAM.

### ps and pstree

The [ps](http://www.linux.ie/newusers/beginners-linux-guide/ps.php) and [pstree](http://www.linfo.org/pstree.html) commands are two of the Linux administrator’s best friends. They both provide a list of all currently running processes. Ps tells you how much memory and processor time the server’s programs are using. Pstree shows less information, but highlights which processes are the children of other processes. Armed with this information, you can spot out–of-control processes and kill them off with Linux's “take no prisoners” [kill](http://linux.about.com/library/cmd/blcmdl_kill.htm) command.

### sar

The [sar](http://www.thegeekstuff.com/2011/03/sar-exampl) program is a Swiss-army knife of a system monitoring tool. The sar command is actually made up of three programs: sar, which displays the data, and sa1 and sa2, which collect and store it. Once installed, sar creates a detailed overview of CPU utilization, memory paging, network I/O and transfer statistics, process creation activity, and [storage device](http://h30565.www3.hp.com/t5/Feature-Articles/Securing-Data-at-Rest-with-Encrypted-Portable-Drives/ba-p/243) activity. The big difference between sar and nmon is that the former is better at long-term system monitoring, while I find nmon to be better at giving me a quick read on my server's status.

### strace

[strace](http://www.hokstad.com/5-simple-ways-to-troubleshoot-using-strace.html) is often thought of a programmer's debugging tool, but it's more than that. It intercepts and records the system calls that are called by a process. This makes it a useful diagnostic, instructional, and debugging tool. For example, you can use strace to find out which configuration file a program is actually using when it starts up.

Strace does have one flaw though. When it's checking out a specific process, that process' performance will fall through the floor. Thus, I only use strace when I already have a darned good reason to think that that program is causing trouble.

### tcpdump

[Tcpdump](http://danielmiessler.com/study/tcpdump/) is a simple, robust network monitoring utility. Its basic protocol analyzing capability enables you to get a rough view of [what is happening on your network](http://h30565.www3.hp.com/t5/Feature-Articles/Understanding-Syslog-Managers-and-How-to-Use-Them/ba-p/1488). To really dig into what's going on with your network however, you'll want to use [Wireshark](http://ww/) (see below).

### top

The [top](http://adminlinux.blogspot.com/2009/06/how-do-i-use-linux-top-command.html) command shows what's going on with your active processes. By default, it displays the most CPU-intensive tasks running on the server and updates the list every five seconds. You can sort the processes by PID (Process ID); age, newest first; time, by cumulative time; and resident memory usage and total time it's been using the CPU since startup. I find this a fast and easy way to see if any process is starting to go out of control and about to get into trouble.

### uptime

Use [uptime](http://www.computerhope.com/unix/uptime.htm) to see how long the server has been running and how many users are logged on. It also gives you an overview of the average server load. The optimal value of the load is 1 or less, which means that each process has immediate access to the CPU and there are no CPU cycles lost.

### vmstat

For the most part, you use [vmstat](http://www.linuxjournal.com/article/8178) to monitor what's going on with virtual memory. Linux constantly uses virtual memory to get the best possible storage performance.

If your applications are taking up too much memory you get excessive page-outs — programs moving from RAM to your system's swap space, which is on the hard drive. Your server can reach a point where it's spending more time managing memory paging than running your applications, a condition called thrashing. When your computer is thrashing, its performance falls through the floor. Vmstat, which can display either average data or actual samples, can help you spot memory pig programs and processes before they bring your server to a crawl.

### Wireshark

[Wireshark](http://www.wireshark.org/), formerly known as Ethereal (and still often referred to that way), is tcpdump's big brother, though it is more sophisticated and with far more advanced protocol analyzing and reporting. Wireshark has both a GUI interface and a shell interface. If you do any serious network administration, you must use ethereal. And, if you're using Wireshark/ethereal, I highly recommend Chris Sander's [Practical Packet Analysis](http://www.amazon.com/gp/product/B002N3M6RC/ref=as_li_ss_tl?ie=UTF8&tag=thegroovycorpora&linkCode=as2&camp=1789&creative), a great book on how to get the most out of this useful program.

This has only been a 10,000 foot overview of some of Linux's most valuable system monitoring programs. Still, if you can master these programs you'll be well on your way to being a top Linux system administrator.

### NFS Server Interview Questions And Answers for linux admin

**Q: - Explain this entry /shared 192.168.1.0/255.255.255.0(sync,rw)**

allows all systems with 192.168.1.\* IP addresses read-write access to the /shared/ directory:

**Q: - What will happened if a space is given inbetween allowed\_hosts and (options)**  
  
If a space is included, the options are applied to any and all IP addresses, which can be quite dangerous if write permission is granted.

**Q: - What is the role of "sync" option for NFS server**  
  
If sync is specified, the server waits until the request is written to disk before responding to the client. The sync option is recommended because it follows the NFS protocol.

**Q: - How to retrieve a list of clients connected to the NFS server ?**  
  
To retrieve a list of clients connected to the NFS server, use the showmount command  
from a shell prompt. To also show the directories the clients are connected to, use the  
showmount -a command.

**Q: - Name of Configuration file for NFS Server ?**  
  
/etc/exports

**Q: - What is meaning of "no\_root\_squash" option ?**  
  
Treat remote root user as local root. Do not map requests from root to the anony-  
mous user and group ID.

**Q: - What is NFS ?**  
  
NFS stands for Network File System. NFS was originally developed by Sun Microsystems in the 1980's. NFS allows remote hosts to mount file systems over a network and interact with those file systems as though they are mounted locally. This enables system administrators to consolidate resources onto centralized servers on the network.

**Q: - Which NFS versions are available ?**  
  
NFS Version 2  
NFS Version 3  
NFS Version 4

**Q: - What is different between NFS Version 2 & 3 ?**  
  
nfs 2 default 8kb transfer rate,it did not check the authentication at the time connection.client wants to access unauthorized file it shows error messages like "write error","read error" nfs 3 32kb transfer rate. It check at the time connection- ACL Support

**Q: - Can we grant access by Username and password for nfs share?**

No, access is granted only for IP address.

**Q: - What is the role of "all\_squash" option?**

Treat all client users as anonymous users. Map all user and group IDs to the anonymous user and group ID.

**Q: - What is the role of "root\_squash" option?**

All requests from the user root are translated or mapped as if they came from the user anonymous (default).

**Q: - Explain option "all\_squash"?**

The UID and GID of exported files are mapped to the user anonymous. It is good for public directories.

**Q: - Explain "exportfs" command?**

The exportfs command is used to maintain the current table of exported file systems for NFS.

**Q: - Explain command "/usr/sbin/exportfs -f"?**

It will flush everything out of the kernels export table. Any clients that are active will get new entries added by mountd when they make their next request.

**Q: - Which option is used with exportfs command to display the current export list, also displays the list of export options?**

exportfs -v

**Q: - Which option is used with exportfs command to re-export all directories?**

exportfs -r

**Q: - How you will export directory (/data) to host 192.168.1.51, allowing asynchronous writes without adding the entry in /etc/exports file?**

 # exportfs -o async 192.168.1.51:/data

**Q: - Is rpc.mountd daemon supports TCP\_WRAPPERS?**

Yes, The rpc.mountd daemon is protected by the tcp\_wrappers. You have to give the clients access to rpc.mountd if they should be allowed to use NFS Server.

**Q: - Explain "nfsstat" command?**

The nfsstat command displays the statistics about NFS client and NFS server activity.

**Q: - What do you understand by "nfsstat -o all -234" command?**

It will Show all information about all versions of NFS.

**Q: - What do you understand by "nfsstat --nfs --server -3" command?**

It will show statistics for NFS version 3 server.

**Q: - Can NFS share mounted on Window XP and Justify your answer?**

No, Window XP operating system doesn’t support nfs protocol.

**Q: - 192.168.1.51:/data is exported by NFS Server and i want to add this NFS share to client /etc/fstab file. How you will add this entry in /etc/fstab file?**

# device                      mount-point     fs-type     options      dump   fsckorder

192.168.1.51:/data            /mnt           nfs               rw              0         0

**Q: - Explain "Soft Mounting" option at NFS Client?**  
  
if a file request fails, the NFS client will report an error to the process on the client machine requesting the file access. if it cannot be satisfied (for example, the server is down), then it quits. This is called soft mounting.

**Q: - Explain "Hard Mounting" option at NFS Client?**

If a file request fails, the NFS client will report an error to the process on the client machine requesting the file access. if it cannot be satisfied, then it will not quit until the request is satisfied. This is called Hard mounting.

**Q: - What is "portmap"?**

The portmapper keeps a list of what services are running on what ports. This list is used by a connecting machine to see what ports it wants to talk to access certain services.

**Q: - How you will check "portmap" service is running or not?**

rpcinfo -p

**Q: - I am unable to mount a NFS share. How will you trace out the reason?**

Firstly, check that you have permissions to mount nfs share or not. Check /etc/exports file.

Secondly you can get RPC error: Program Not Registered (or another "RPC" error)

For this check your NFS server and portmap service running or not by "rpcinfo -p"

**Q: - Can I modify export permissions without needing to remount clients in order to have them take effect?**

Yes. The safest thing to do is edit /etc/exports and run "exportfs -r".

**Scenario**  
Here are my test setup scenario :

Operating System : RHEL 6

Internal LAN IP of DNS Server : 192.168.10.2

Hostname : server1.howtoc.com

**1. Setup a network-script files :**

[root@server1 ~]# vim /etc/sysconfig/netwprk-scripts/ifcfg-eth0

**2. Setup a hosts file :**

[root@server1 ~]# vim /etc/hosts

**192.168.10.2 server1.howtoc.com server1** # Added by NetworkManager

127.0.0.1 localhost.localdomain localhost

::1 server1.howtoc.com server1 localhost6.localdomain6 localhost6

[root@server1 ~]# vim /etc/sysconfig/network

NETWORKING=yes

**HOSTNAME=server1.howtoc.com**

**3. Add the nameserver in resolve file :**  
[root@server1 ~]# vim /etc/resolve.conf

**search howtoc.com**

**nameserver 192.168.10.2**

**4. Now time to install BIND packages from yum :**

[root@server1 ~]# yum -y install bind\*

[root@server1 ~]# updatedb

*# Find the named.conf file(Main configuration file of BIND)*

[root@server1 ~]# locate named.conf

/etc/named.conf

/usr/share/doc/bind-9.7.0/named.conf.default

/usr/share/doc/bind-9.7.0/sample/etc/named.conf

/usr/share/logwatch/default.conf/services/named.conf

/usr/share/man/man5/named.conf.5.gz

*# Go to below path*

[root@server1 ~]# cd /var/named/chroot/

[root@server1 chroot]# cd etc

[root@server1 etc]# pwd

/var/named/chroot/etc

**5. Copy named.conf file from BIND lib. & Change the group of named.conf :**

[root@server1 etc]# cp /usr/share/doc/bind-9.7.0/named.conf.default named.conf

[root@server1 etc]# chgrp named named.conf

[root@server1 etc]# ll named.conf

f -rw-r--r--. 1 root named 930 Aug 3 07:58 named.conf

**6. Edit the BIND configuration file :**

[root@server1 etc]# vim named.conf

[root@server1 etc]# grep listen named.conf

listen-on port 53 { 127.0.0.1; };

*Comment it # // listen-on-v6 port 53 { ::1; };*

[root@server1 etc]# vim named.conf

[root@server1 etc]# grep listen named.conf

**listen-on port 53 { 127.0.0.1; 192.168.10.2; };**

*Comment it # // listen-on-v6 port 53 { ::1; };*

**7. Restart the name(BIND) service :**

[root@server1 etc]# /etc/init.d/named restart

Stopping named: [ OK ]

Starting named: [ OK ]

**8. Edit the named.conf file & add the zone :**

# vim /etc/named.conf

//

// named.conf

//

// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS

// server as a caching only nameserver (as a localhost DNS resolver only).

//

// See /usr/share/doc/bind\*/sample/ for example named configuration files.

//

options {

**listen-on port 53 { 127.0.0.1;192.168.10.2; };**

**/\*listen-on-v6 port 53 { ::1; };\*/**

directory "/var/named";

dump-file "/var/named/data/cache\_dump.db";

statistics-file "/var/named/data/named\_stats.txt";

memstatistics-file "/var/named/data/named\_mem\_stats.txt";

**allow-query { localhost;192.168.10.0/24; };**

# transfer range ( set it if you have secondary DNS )

**allow-transfer { localhost; 192.168.10.0/24; };,**

recursion yes;

dnssec-enable yes;

dnssec-validation yes;

dnssec-lookaside auto;

/\* Path to ISC DLV key \*/

bindkeys-file "/etc/named.iscdlv.key";

};

logging {

channel default\_debug {

file "data/named.run";

severity dynamic;

};

};

zone "." IN {

type hint;

file "named.ca";

};

**zone "howtoc.com" IN {**

**type master;**

**file "forward.zone";**

**allow-update { none; };**

**};**

**zone "10.168.192.in-addr.arpa" IN {**

**type master;**

**file "reverse.zone";**

**allow-update { none; };**

**};**

**include "/etc/named.rfc1912.zones";**

**9. Now edit the rfc1912.zones which define in named.conf :**

# vim /etc/named.rfc1912.zones

// named.rfc1912.zones:

//

// Provided by Red Hat caching-nameserver package

//

// ISC BIND named zone configuration for zones recommended by

// RFC 1912 section 4.1 : localhost TLDs and address zones

// and http://www.ietf.org/internet-drafts/draft-ietf-dnsop-default-local-zones-02.txt

// (c)2007 R W Franks

//

// See /usr/share/doc/bind\*/sample/ for example named configuration files.

//

**zone "howtoc.com" IN {**

**type master;**

**file "forward.zone";**

**allow-update { none; };**

**};**

zone "localhost" IN {

type master;

file "named.localhost";

allow-update { none; };

};

zone "1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa" IN {

type master;

file "named.loopback";

allow-update { none; };

};

**zone "10.168.192.in-addr.arpa" IN {**

**type master;**

**file "reverse.zone";**

**allow-update { none; };**

**};**

zone "0.in-addr.arpa" IN {

type master;

file "named.empty";

allow-update { none; };

};

**10. Copy the zone file from BIND Lib :**

[root@server1 named]#cp named.localhost forward.zone

[root@server1 named]#cp named.loopback reverse.zone

**11. Edit the forward zone (name to ip Addr) :**

[root@server1 named]#vim /var/named/forward.zone

$TTL 1D

@ IN SOA server1.howtoc.com. root.howtoc.com. (

0 ; serial

1D ; refresh

1H ; retry

1W ; expire

3H ) ; minimum

IN NS server1.howtoc.com.

IN A 192.168.10.2

server1 IN A 192.168.10.2

**12. Edit the reverse zone (ip Addr to name) :**

[root@server1 named]#vim /var/named/reverse.zone

$TTL 1D

@ IN SOA server1.howtoc.com. root.howtoc.com. (

0 ; serial

1D ; refresh

1H ; retry

1W ; expire

3H ) ; minimum

IN NS server1.howtoc.com.

IN PTR howtoc.com.

IN A 255.255.255.0

2 IN PTR server1.howtoc.com.

**12. Change the group permission & restart the service :**

[root@server1 named]#chgrp named forward.zone

[root@server1 named]#chgrp named reverse.zone

[root@server1 named]#/etc/init.d/named restart

**13. Test your DNS server using dig command :**  
*@ forward lookup*

[root@server1 named]# dig server1.howtoc.com

; <<>> DiG 9.7.0-P2-RedHat-9.7.0-5.P2.el6 <<>> server1.howtoc.com

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 50351

;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:

;server1.howtoc.com. IN A

;; ANSWER SECTION:

**server1.howtoc.com. 86400 IN A 192.168.10.2**

**;; AUTHORITY SECTION:**

**howtoc.com. 86400 IN NS server1.howtoc.com.**

[root@server1 named]# dig -x 192.168.10.2

; <<>> DiG 9.7.0-P2-RedHat-9.7.0-5.P2.el6 <<>> -x 192.168.10.2

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45077

;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:

;2.10.168.192.in-addr.arpa. IN PTR

**;; ANSWER SECTION:**

**2.10.168.192.in-addr.arpa. 86400 IN PTR server1.howtoc.com.**

**;; AUTHORITY SECTION:**

**10.168.192.in-addr.arpa. 86400 IN NS server1.howtoc.com.**

**;; ADDITIONAL SECTION:**

**server1.howtoc.com. 86400 IN A 192.168.10.2**

[root@server1 named]# nslookup

> server1.howtoc.com

Server: 192.168.10.2

Address: 192.168.10.2#53

Name: server1.howtoc.com

Address: 192.168.10.2

> 192.168.10.2

Server: 192.168.10.2

Address: 192.168.10.2#53

2.10.168.192.in-addr.arpa name = server1.howtoc.com.

### configure and install yum server in RHEL

In Linux you can install packages through many ways like.

**YUM(Yellow-dog Updater and Modifier)**is another and advanced way of installing the packages in Linux distro's such as Red-hat, Fedora and CenOS.

In RHEL4 installing packages is a tedious process, some times its headache to install all the dependencies. So Red-hat come with a solution to overcome this dependencies problem in most situations, the solution for this is nothing but YUM implementation. This will resolve this dependency issue and other known issues. Here we are going to present some basic way how to use YUM utility to install packages locally(there are so many ways to install packages from different sources either local or remote such as ftp, http).

**Basic YUM implementation locally:**

**Step1 :**Copy the entire OS cd's content to Hard-drive as below.

**#cp -ar /media/cdrom/Server/\* /destinationfolder**

**Ex: server1#cp -ar /media/cdrom/Server/\* /var/ftp/pub/Server/**

Note :  
1. From second cd too Server content in to our /var/ftp/pub/Server/ folder as shown below.  
2. Here please take destination folder as /var/ftp/pub so that we can implement FTP server to share our repository.  
3. If you have RHEL5 DVD then execute below command once.

**server1#cp -ar /media/cdrom/Server/\* /var/ftp/pub/Server/**

**Step2 :**Now change the directory to /var/ftp/pub/Server and install the createrepo package

**server1#cd /var/ftp/pub/Server  
server1#rpm -ivh createrepo\***

**Step3 :** Specify the repository location to YUM.

**server1#createrep -v .**

Note:There is the dot in the above command.  
**Step4 :**Create a file with repo as extension and specify the YUM details in /etc/yum.repos.d folder

**server1#cd /etc/yum.repos.d/  
server1#vi testing.repo**

**Note :** The directory /etc/yum.repos.d/ contaions two .repo files which should be removed or moved to other directory, so that YUM server will check default .repo file it self.Q. What if I don't move or remove the default .repo files from /etc/yum.repos.d/ folder?A: Every time when you try to install packages through YUM, your yum will check all these files for repositories for getting packages which will or will not work and most probably delay is increased in getting those packages from online servers.

The new file which is created contains as follows.

**[server1.linuxnix.com] comment ="**test**"  
baseurl=file:///var/ftp/pub/Server  
gpgcheck=0**

After entering these entries save and exit from the file.

Let me explain what actually these four entries mean.

**[server1.example.com]** ==>This informs what is the repository name.

**comment** ==> Its used to see the information about the repo.

**baseurl** ==> This is the server and path of the repo(here its a local repo so the base url is just a**file:///**  
For example you are creating YUM client through FTP then base url should be like this

**baseurl=ftp://station1.example.com/pub/Server**

**gpgcheck** ==> This is to check the authentication of the repository, which is disabled in this case.  
Local YUM repository is created, now you can install any package you want with yum command. In order to know more about YUM, Please see man pages for YUM.  
Note:In-order to use yum repository we have to clean the yum meta data, so before installing any package first time use yum clean all command as shown below.

**server1#yum clean all**

**Basic YUM Server Repository through FTP server :**

So what about Installing packages remotely by using this repository?

Let us see how to configure client to access this repository. Before doing client configuration we have to share this repository through FTP or HTTP.

**Step1 :** Install vsftpd server on server

**server1#yum install vsftpd**

**Step2 :** Start the ftp service and on it

**server1#service vsftpd restart**

**server1#chkconfig vsftpd on**

That's it on the server side every thing configured properly, Now move on to client machine.

## ****Conflagration Yum client in Linux****

**Step3 :**Remove/move the local repository file from /etc/yum.repo.d/ folder to some other location #mv /etc/yum.repo.d/\* /tmp/

**Step4 :** Create server.repo file in /etc/yum.repo.d/ with following contents

**[server1.linuxnix.com]**

comment ="test"

baseurl=ftp://server1.example.com/pub/Server

gpgcheck=0

Save and exit the file Now start using yum to install packages, as follows.

**client1#yum clean all  
client1#yum install packagename**

**client1#yum install httpd**

To uninstall a package through YUM

**client1#yum remove httpd**

To see the info of a package

**client1#yum info packagename**

To see the package is already installed or not

**client1#rpm -qa grep packagename**

fdisk -l /dev/hdb

Replace /dev/hdb with the device of the hard disk on your system with the swap partition on it. You should see output that looks like this:

Device Boot Start End Blocks Id System

/dev/hdb1 2328 2434 859446 82 Linux swap / Solaris

If the partition isn't marked as swap you will need to alter it by running fdisk and using the 't' menu option. Be careful when working with partitions -- you don't want to delete important partitions by mistake or change the id of your system partition to swap by mistake. All data on a swap partition will be lost, so double-check every change you make. Also note that Solaris uses the same ID as Linux swap space for its partitions, so be careful not to kill your Solaris partitions by mistake.

Once a partition is marked as swap, you need to prepare it using the mkswap (make swap) command as root:

mkswap /dev/hdb1

If you see no errors, your swap space is ready to use. To activate it immediately, type:

swapon /dev/hdb1

You can verify that it is being used by running swapon -s. To mount the swap space automatically at boot time, you must add an entry to the /etc/fstab file, which contains a list of filesystems and swap spaces that need to be mounted at boot up. The format of each line is:

Since swap space is a special type of filesystem, many of these parameters aren't applicable. For swap space, add:

/dev/hdb1 none swap sw 0 0

where /dev/hdb1 is the swap partition. It doesn't have a specific mount point, hence none. It is of type swapwith options of sw, and the last two parameters aren't used so they are entered as 0.

To check that your swap space is being automatically mounted without having to reboot, you can run the swapoff -a command (which turns off all swap spaces) and then swapon -a (which mounts all swap spaces listed in the /etc/fstab file) and then check it with swapon -s.

#### Swap file

As well as the swap partition, Linux also supports a swap file that you can create, prepare, and mount in a fashion similar to that of a swap partition. The advantage of swap files is that you don't need to find an empty partition or repartition a disk to add additional swap space.

To create a swap file, use the dd command to create an empty file. To create a 1GB file, type:

dd if=/dev/zero of=/swapfile bs=1024 count=1048576

/swapfile is the name of the swap file, and the count of 1048576 is the size in kilobytes (i.e. 1GB).

Prepare the swap file using mkswap just as you would a partition, but this time use the name of the swap file:

mkswap /swapfile

And similarly, mount it using the swapon command: swapon /swapfile. The /etc/fstab entry for a swap file would look like this:

/swapfile none swap sw 0 0