**Linux and Asterisk Administrative Configuration Guide for Reference:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S no** | **Configuration** | **Type** | **Purpose** | **Page no** |
| 1 | NTP | M | Time synchronization with all client WS | d |
| 2 | Samba 3-5 | L |  | d |
| 3 | NFS 5-8 | L |  | d |
| 4 | FTP 8-9 | L |  | d |
| 5 | Monitorix tool 9-13 | M |  | d |
| 6 | Nload | L |  | d |
| 7 | Htop | L |  | d |
| 8 | iftop | L |  | d |
| 9 | Asterisk GUI | M |  | d |
| 10 | SSHPASS | L |  | d |
| 11 | PXE & KS server | M |  | d |
| 12 | WEB Admin | M |  | d |
| 13 | Script for Back-up | M |  | d |
| 14 | hiding partation------------ | L |  | d |
| 15 | ALSA drivers | L |  | d |
| 16 | Grub for dual os | M |  | d |
| 17 | TCP dump | L |  | d |
| 18 | Audit file & folder | M |  | d |
| 19 | dstat | L |  | d |
| 20 | Switch firm ware update | M |  | d |
| 21 | SSh permissions | L |  | d |
| 22 | SSH password less login | L |  | d |
| 23 | Iptables | L |  | d |
| 24 | ESC disable | L |  | d |
| 25 | Tune admin | L |  | d |
| 26 | Port Bonding | M |  | d |
| 27 | Remote Wallpaper change | L |  | d |
| 28 | Mounting win partation on centos | L |  | d |
| 29 | OS Upgrade commands in Ubuntu | L |  | d |
| 30 | Changing UDP packet size in Centos | L |  | d |
| 31 | Asterisk related commands | M |  | d |
| 32 | Repo server |  |  | P |
| 33 | VLAN conf |  |  | d |
| 34 | Linux soft-firewall conf |  |  | P |
| 35 | LDAP for Linux |  |  | P |

**NTP Server and Client configuration**

**NTP Server Side conf:**

First, install and enable ntpd on 192.168.1.5:  
# yum install ntp  
# chkconfig ntpd on  
Now open /etc/ntp.conf:  
# vi /etc/ntp.conf  
Make sure the following line exits:  
**restrict default ignore**  
Above will deny all access to any machine, server or client. However, you need to specifically authorized policy settings. Set it as follows:

**restrict 202.54.1.5 mask 255.255.255.245 nomodify notrap noquery**

**server 202.54.1.5**

Replace 202.54.1.5 and mask with actual remote ISP or ntp.org NTP server IP. Save and close the file.

**Configure NTP clients to access your NTP Server**

Now, you need to allow legitimate NTP clients to access the Server. For example, allow 192.168.1.0/24 network to synchronize to this server located at 192.168.1.5. Open /etc/ntp.conf and add policy as follows:

# Hosts on local network are less restricted.

restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap

Update your firewall settings, open /etc/sysconfig/iptables.  
# vi /etc/sysconfig/iptables  
Add the following line, before the final LOG and DROP lines for the RH-Firewall-1-INPUT chain:

|  |
| --- |
| -A RH-Firewall-1-INPUT -s 192.168.1.0**/**24 -m state --state NEW -p udp --dport 123 -j ACCEPT |

Save and close the file. Finally, start ntpd:  
# service ntpd start  
# service iptables restart  
# netstat -tulpn

**NTP Client Side conf:**

1) Login to root privilege

2) Type below commands for installing NTP service.

#apt-get update

#apt-get install ntp

#sudo gedit /etc/ntp.conf   

(in ntp.conf file need to mention NTP server IP address and remove default ntp server names colored in orange)

{

# on 2011-02-08 (LP: #104525). See http://www.pool.ntp.org/join.html for  
# more information.  
server 0.ubuntu.pool.ntp.org   
server 1.ubuntu.pool.ntp.org  
server 2.ubuntu.pool.ntp.org  
server 3.ubuntu.pool.ntp.org

need to delete all these above servers and mention

**server 192.168.1.5 iburst**

**server 192.168.1.5 prefer**

1 # Use Ubuntu's ntp server as a fallback.  
server ntp.ubuntu.com

delet above and mention

**server 192.168.1.5**

}

save and exit

3) Run following command for starting and testing ntp service

#/etc/init.d/ntp start

#service ntp restart

#ntpq -p

#ntpd -update                         (this command will show the values of Jitter and Delay)

#ntpdate –u 19.168.1.5            (this command will manually sync time with NTP server, if not sync automatically)

**Samba Server configuration**

**Server side conf:**

**Part 1: Configuring anonymous share with samba server**

To install the **samba** package,enter the following command:

sudo yum install samba samba-client samba-common

Check the version of installed samba software by using this command:

**smbd --version**

Configure the samba service, so that, it will start automatically at boot time:

sudo chkconfig smb on

sudo chkconfig nmb on

Disable the SELinux:

sudo nano /etc/selinux/config

Change SELinux from enforcing to disabled:

SELINUX=disabled

Add these Iptables rules, so that samba will work perfectly:

sudo iptables -I INPUT 4 -m state --state NEW -m udp -p udp --dport 137 -j ACCEPT

sudo iptables -I INPUT 5 -m state --state NEW -m udp -p udp --dport 138 -j ACCEPT

sudo iptables -I INPUT 6 -m state --state NEW -m tcp -p tcp --dport 139 -j ACCEPT

sudo service iptables save

**Restart the Server!!!**

Go to your Windows machine and use this command in order to check the WORKGROUP name:

**net config workstation**

Backup the smb.conf file, then delete it and create the new one:

sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.bak

sudo rm /etc/samba/smb.conf

sudo touch /etc/samba/smb.conf

sudo nano /etc/samba/smb.conf

Add these lines, in your smb.conf file (or change it according to your requirement):

#======================= Global Settings =====================================

[global]

workgroup = WORKGROUP

security = share

map to guest = bad user

#============================ Share Definitions ==============================

[MyShare]

path = /samba/share

browsable =yes

writable = yes

guest ok = yes

read only = no

Save the smb.conf file and restart the service:

sudo service smb restart

sudo service nmb restart

Check the current permission on the samba share:

cd /samba/

ls -l

Change it, in such a way that everyone can read and write it(Check it, that it is allowed in your environment or not):

sudo chmod -R 0777 share

ls -l

**Part 2: Add and manage users and groups**

Add a group in your CentOS server (in my case smbgrp):

sudo groupadd smbgrp

Create a new share, set the permission on the share:

cd /samba/

sudo mkdir secure

sudo chown -R arbab:smbgrp secure/

ls -l

sudo chmod -R 0770 secure/

ls -l

Add the user to the samba group and create samba password:

sudo usermod -a -G smbgrp arbab

sudo smbpasswd -a arbab

Edit the smb.conf file:

sudo nano /etc/samba/smb.conf

Add the newly created samba share in smb.conf file:

[Secure]

path = /samba/secure

valid users = @smbgrp

guest ok = no

writable = yes

browsable = yes

Restart the samba service:

sudo service smb restart

sudo service nmb restart

Check the syntax error with ***testparm***:

sudo testparm

**Check for the OK value in Load services =OK**

**Configuring NFS in Centos:**

## Setup

An NFS mount is set up between at least two servers. The machine hosting the shared network is called the server, while the ones that connect to it are called ‘clients’.

This tutorial requires 2 servers: one acting as the server and one as the client. We will set up the server machine first, followed by the client. The following IP addresses will refer to each one:

**Master: 12.34.56.789**

**Client: 12.33.44.555**

The system should be set up as root. You can access the root user by typing

sudo su

## Setting Up the NFS Server

### Step One—Download the Required Software

Start off by using apt-get to install the nfs programs.

yum install nfs-utils nfs-utils-lib

Subsequently, run several startup scripts for the NFS server:

chkconfig nfs on

service rpcbind start

service nfs start

### Step Two—Export the Shared Directory

The next step is to decide which directory we want to share with the client server. The chosen directory should then be added to the /etc/exports file, which specifies both the directory to be shared and the details of how it is shared.

Suppose we wanted to share the directory, /home.

We need to export the directory:

vi /etc/exports

Add the following lines to the bottom of the file, sharing the directory with the client:

/home 12.33.44.555(rw,sync,no\_root\_squash,no\_subtree\_check)

These settings accomplish several tasks:

* **rw:** This option allows the client server to both read and write within the shared directory
* **sync:** Sync confirms requests to the shared directory only once the changes have been committed.
* **no\_subtree\_check:** This option prevents the subtree checking. When a shared directory is the subdirectory of a larger filesystem, nfs performs scans of every directory above it, in order to verify its permissions and details. Disabling the subtree check may increase the reliability of NFS, but reduce security.
* **no\_root\_squash:** This phrase allows root to connect to the designated directory

Once you have entered in the settings for each directory, run the following command to export them:

exportfs -a

## Setting Up the NFS Client

### Step One—Download the Required Software

Start off by using apt-get to install the nfs programs.

yum install nfs-utils nfs-utils-lib

### Step Two—Mount the Directories

Once the programs have been downloaded to the the client server, create the directory that will contain the NFS shared files

mkdir -p /mnt/nfs/home

Then go ahead and mount it

mount 12.34.56.789:/home /mnt/nfs/home

You can use the df -h command to check that the directory has been mounted. You will see it last on the list.

df -h

Filesystem Size Used Avail Use% Mounted on

/dev/sda 20G 783M 18G 5% /

12.34.56.789:/home 20G 785M 18G 5% /mnt/nfs/home

Additionally, use the mount command to see the entire list of mounted file systems.

mount

Your list should look something like this:

/dev/sda on / type ext4 (rw,errors=remount-ro)

none on /proc/sys/fs/binfmt\_misc type binfmt\_misc (rw)

sunrpc on /var/lib/nfs/rpc\_pipefs type rpc\_pipefs (rw)

nfsd on /proc/fs/nfsd type nfsd (rw)

12.34.56.789:/home on /mnt/nfs/home type nfs (rw,noatime,nolock,bg,nfsvers=2,intr,tcp,actimeo=1800,addr=12.34.56.789)

## Testing the NFS Mount

Once you have successfully mounted your NFS directory, you can test that it works by creating a file on the Client and checking its availability on the Server.

Create a file in the directory to try it out:

touch /mnt/nfs/home/example

You should then be able to find the files on the Server in the /home.

ls /home

You can ensure that the mount is always active by adding the directory to the fstab file on the client. This will ensure that the mount starts up after the server reboots.

vi /etc/fstab

12.34.56.789:/home /mnt/nfs/home nfs auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo=1800 0 0

You can learn more about the fstab options by typing in:

man nfs

After any subsequent server reboots, you can use a single command to mount directories specified in the fstab file:

mount -a

You can check the mounted directories with the two earlier commands:

df -h

mount

## Removing the NFS Mount

Should you decide to remove a directory, you can unmount it using the umount command:

cd

sudo umount /*directory name*

You can see that the mounts were removed by then looking at the filesystem again.

df -h

You should find your selected mounted directory gone.

**Configuring FTP server in Centos:**

The first two letters of vsftpd stand for "very secure" and the program was built to have strongest protection against possible FTP vulnerabilities.

## Step One—Install vsftpd

You can quickly install vsftpd on your virtual private server in the command line:

sudo yum install vsftpd

We also need to install the FTP client, so that we can connect to an FTP server:

sudo yum install ftp

Once the files finish downloading, vsftpd will be on your VPS. Generally speaking, the virtual private server is already configured with a reasonable amount of security. However, it does provide access to anonymous users.

## Step Two—Configure VSFTP

Once VSFTP is installed, you can adjust the configuration.

Open up the configuration file:

sudo vi /etc/vsftpd/vsftpd.conf

One primary change you need to make is to change the Anonymous\_enable to No:

anonymous\_enable=NO

Prior to this change, vsftpd allowed anonymous, unidentified users to access the VPS's files. This is useful if you are seeking to distribute information widely, but may be considered a serious security issue in most other cases. After that, uncomment the local\_enable option, changing it to yes.

local\_enable=YES

Finish up by uncommenting command to chroot\_local\_user. When this line is set to Yes, all the local users will be jailed within their chroot and will be denied access to any other part of the server.

chroot\_local\_user=YES

Finish up by restarting vsftpd:

sudo service vsftpd restart

In order to ensure that vsftpd runs at boot, run chkconfig:

chkconfig vsftpd on

## Step Three—Access the FTP server

Once you have installed the FTP server and configured it to your liking, you can now access it.

You can reach an FTP server in the browser by typing the domain name into the address bar and logging in with the appropriate ID. Keep in mind, you will only be able to access the user's home directory.

ftp://example.com

Alternatively, you can reach the FTP server through the command line by typing:

ftp example.com

Then you can use the word, "exit," to get out of the FTP shell.

**Installing Monitorix in Centos:**

**Method :1**

# install from [EPEL](https://www.server-world.info/en/note?os=CentOS_6&p=initial_conf&f=6)

[root@dlp ~]#

[yum](https://www.server-world.info/en/command/html/yum.html) --enablerepo=epel -y install monitorix

[root@dlp ~]#

[vi](https://www.server-world.info/en/command/html/vi.html) /etc/monitorix/monitorix.conf

# line 6: change to any title you like

title =

Monitorix

# line 7: change to your own hostname

hostname =

dlp.srv.world

# line 8: backgroud color of admin site

theme\_color =

white

# line 12: change network units to bps (default is Bytes per/sec)

netstats\_in\_bps =

y

<httpd\_builtin>  
    enabled = y  
    host =  
    port = 8080  
    user = nobody  
    group = nobody  
    log\_file = /var/log/monitorix-httpd

# set permittion for admin site

    hosts\_deny =

all

    hosts\_allow =

10.0.0.0/24

[root@dlp ~]#

/etc/rc.d/init.d/monitorix start

[ OK ] monitorix: [ OK ]  
[root@dlp ~]#

[chkconfig](https://www.server-world.info/en/command/html/chkconfig.html) monitorix on

Access to the "http://(Monitorix server's hostname or IP address):8080/monitorix/" from a client which is in the network allowed in config. Then, Monitorix admin site is displayed like follows. Click "OK" to see graphs.

**Method: 2**

# How to install Monitorix system monitoring tool on CentOS 6 VPS

First thing you need to do in order to install Monitorix system monitoring tool on your CentOS 6 VPS is to update all of your system software to the latest version by using the command:

## yum update

After the update is finished you will need to install all the required packages. To install the required packages you should enable the EPEL repository on your [CentOS 6 VPS](https://www.rosehosting.com/centos-vps.html) with the following commands:

For CentOS 6 32bit

## wget http://download.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm

## rpm -ivh epel-release-6-8.noarch.rpm

For CentOS 6 64bit

## wget http://download.fedoraproject.org/pub/epel/6/x86\_64/epel-release-6-8.noarch.rpm

## rpm -ivh epel-release-6-8.noarch.rpm

Once you enable the EPEL repository you can install the required packages using the command:

## yum install perl perl-libwww-perl perl-HTTP-Server-Simple perl-IO-Socket-SSL perl-MIME-Lite perl-CGI perl-DBI perl-XML-Simple perl-MailTools perl-Config-General rrdtool rrdtool-perl

Please note: if the installation of some of the required packages fails, you will need to install that package manually.

Now it is time to download and install the Monitorix system monitoring tool on your server. It can be done with only one command:

## rpm -ivh http://www.monitorix.org/monitorix-x.x.x-1.noarch.rpm

You will need to replace the x.x.x. with the version of the system monitoring tool you are going to install. It is recommended to download and install the latest version of the system monitoring tool. At the moment of writing, the latest Monitorix version available is 3.6.0-1, so to download and install the latest version of Monitorix system monitoring tool you can use the following command:

## rpm -ivh http://www.monitorix.org/monitorix-3.6.0-1.noarch.rpm

Before you start using the Monitorix system monitoring tool you can take a look at the Monitorix configuration file

/etc/monitorix/monitorix.conf

If you want to learn more about how to use the Monitorix and how to set up this powerful system monitoring tool according to your needs, we highly recommend you to read the official Monitorix documentation available at:

<http://www.monitorix.org/documentation.html>

Finally, you can start the service using the command:

## service monitorix start

To access the Monitorix web interface from your favorite web browser you need to navigate to:

http://IP-ADDRESS:8080/monitorix

Do not forget to replace the IP-ADDRESS with your server’s IP address. You should be able to see the welcome Monitorix screen. Select the option you like and press the button OK.

**Method : 3**

## nstallation on a Fedora Linux

Users of newer Fedora versions are able to install Monitorix using directly the official repositories. Therefore a simple command like the following should suffice:

# yum install monitorix

## Installation on a RedHat/CentOS Linux

Install first the [EPEL repository](http://archive.fedoraproject.org/pub/epel/) and then install Monitorix using yum:

# yum install monitorix

If yum fails installing one of the packages (probably in EL4 or EL5 only), then you could try to get them manually from this additional repository:

* [Repoforge repository for Red Hat, CentOS and Fedora](http://pkgs.repoforge.org/)

Although these repositories are considered by many in the community stable and safe, please follow these recommendations:

* Keep your systems clean from permanent installations of third party repositories.
* Check both repositories to make sure you are downloading the most up to date version of each package.
* Download and install each package satisfying manually all its dependencies.

Once succesfully installed, take a look into the configuration file /etc/monitorix/monitorix.conf to set the options according your system and enable or disable graphs.

Finally start Monitorix with:

# service monitorix start

At this point, Monitorix will start gathering the system information based on its configuration and after some minutes, you should be able to see the results from your favorite browser pointing it at:

<http://localhost:8080/monitorix/>

## Notes after installation

If your system has SELinux enabled it is possible your Monitorix graphs won't be visible and you get lot of messages in /var/log/audit/audit.log and/or in /var/log/messages about access denied to the RRD database files.

Take the proper actions to remedy this situation.

**Installing Nload tool:**

## Install nload on a CentOS/RHEL/Red Hat/Fedora Linux

First, [turn on EPEL repo on a CentOS or RHEL](https://www.cyberciti.biz/faq/fedora-sl-centos-redhat6-enable-epel-repo/) based system. Type the following [yum command](https://www.cyberciti.biz/faq/rhel-centos-fedora-linux-yum-command-howto/) to install nload:  
# yum install nload

## Install nload on a Debian or Ubuntu Linux

Type the following [apt-get command](https://www.cyberciti.biz/tips/linux-debian-package-management-cheat-sheet.html):  
$ sudo apt-get install nload

## How do I use nload to display the current network usage?

The basic syntax is:

|  |
| --- |
| nload  nload device  nload **[**options**]** device1 device2 |

Just type the following command:  
$ nload  
$ nload eth0  
$ nload em0 em2

**How to install Htop:**

To install **Htop** on **RHEL 7/6/5** and **CentOS 7/6/5**, your system must have **EPEL** repository installed and enabled, to do so run the following commands on your respective distributions to install and enable it for your system architecture (**32bit** or **64bit**).

**-------------- For RHEL/CentOS 6 --------------**

# wget http://download.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm

# rpm -ihv epel-release-6-8.noarch.rpm

**-------------- For RHEL/CentOS 5 --------------**

# wget http://download.fedoraproject.org/pub/epel/5/i386/epel-release-5-4.noarch.rpm

# rpm -ihv epel-release-5-4.noarch.rpm

#### On RHEL/CentOS – 64-bit OS

**-------------- For RHEL/CentOS 7 --------------**

# wget http://dl.fedoraproject.org/pub/epel/7/x86\_64/e/epel-release-7-9.noarch.rpm

# rpm -ihv epel-release-7-9.noarch.rpm

**-------------- For RHEL/CentOS 6 --------------**

# wget http://download.fedoraproject.org/pub/epel/6/x86\_64/epel-release-6-8.noarch.rpm

# rpm -ihv epel-release-6-8.noarch.rpm

**-------------- For RHEL/CentOS 5 --------------**

# wget http://download.fedoraproject.org/pub/epel/5/x86\_64/epel-release-5-4.noarch.rpm

# rpm -ihv epel-release-5-4.noarch.rpm

Once **EPEL** repository has been installed, you can hit the following [yum command](http://www.tecmint.com/20-linux-yum-yellowdog-updater-modified-commands-for-package-mangement/) to fetch and install the htop package as shown.

# yum install htop

### Compile and Install Htop from Source Packages

To install **Htop 2.0.2** version, you must have **Development Tools** and **Ncurses** installed on your system, to do so run the following series of commands on your respective distributions.

#### On RHEL/CentOS and Fedora

# yum groupinstall "Development Tools"

# yum install ncurses ncurses-devel

# wget http://hisham.hm/htop/releases/2.0.2/htop-2.0.2.tar.gz

# tar xvfvz htop-2.0.2.tar.gz

# cd htop-2.0.2

Next, run the **configure** and **make** script to install and compile htop.

# ./configure

# make

# make install

**Install Iftop:**

Installation of the **libpcap** and **libncurses** library with **YUM** command as shown below for error-free iftop installation.

# yum -y install libpcap libpcap-devel ncurses ncurses-devel

### Download and Install IFTOP

Download **iftop** from it’s website with **Wget** command as shown below.

# wget http://www.ex-parrot.com/pdw/iftop/download/iftop-0.17.tar.gz

Follow the below all commands to install **iftop**.

# tar -zxvf iftop-0.17.tar.gz

# cd iftop-0.17

# ./configure

# make

# make install

### Basic usage of Iftop

Once installation done, go to your console and type **iftop** command.

# iftop

**Installation of Asterisk GUI:**

Installing Asterisk Graphical User Interface - AsteriskNow  
  
This tutorial is meant to guide you during the installation of "*Asterisk GUI*".The [Digium](http://www.digium.com/)'s Asterisk GUI is available only for version 1.4 of Asterisk. This tool is part of Digium's most recent project - [AsteriskNow](http://www.asterisknow.org/).

[Back to Tutorials](http://www.asteriskguru.com/tutorials/)

1.9. Installation of Asterisk GUI

Installing Asterisk Graphical User Interface - AsteriskNow  
  
This tutorial is meant to guide you during the installation of "*Asterisk GUI*".The [Digium](http://www.digium.com/)'s Asterisk GUI is available only for version 1.4 of Asterisk. This tool is part of Digium's most recent project - [AsteriskNow](http://www.asterisknow.org/).

**1. Requirements**  
In order to install "*Asterisk GUI*" you'll need:

Concurrent Versions System ([*CVS*](http://www.nongnu.org/cvs/))  
asterisk 1.4 - *and all modules needed to compile and install asterisk 1.4*

**2. Preparation**  
Go to your installation directory (for example '/usr/src') and download the latest version of asterisk-gui. To do this just execute the command:

#yum install svn

***svn checkout http://svn.digium.com/svn/asterisk-gui/trunk asterisk-gui***

By executing this command the sources of Asterisk GUI will be downloaded (see our sample output log [*svn.log*](http://www.asteriskguru.com/tutorials/274498/svn.log)) into ./asterisk-gui/ directory.

cd /usr/src  
svn co http://svn.digium.com/svn/asterisk-gui/branches/2.0 asterisk-gui  
cd asterisk-gui  
sh configure && make && make install

**4. Configuration**  
It is recommended to back up your configuration files before you continue. To achieve this just copy /etc/asterisk under different name:

**cp -r /etc/asterisk /etc/asterisk.backup**

It is recommended to `*install samples*`. This command will modify some of your Asterisk configuration files. After that you'll have finish the configuration modification manually.  
  
There are two files which you should modify:

`manager.conf`

enabled = yes  
webenabled = yes

nano /etc/asterisk/manager.conf  
*Add the following information.  You can substitute [admin] and secret= to whatever username/password you want for logging into the Asterisk GUI*[general]  
enabled=yes  
webenabled=yes  
port=5038  
bindaddr=0.0.0.0  
[admin]  
secret=password  
read=system,call,log,verbose,command,agent,user,config,originate,read,write  
write=system,call,log,verbose,command,agent,user,config,originate,read,write  
  
  
nano /etc/asterisk/http.conf  
*Add the following information*   
  
[general]  
enabled=yes  
bindaddr=0.0.0.0  
bindport=8088  
prefix=gui  
enablestatic=yes  
  
*(ctl-x,y,ENTER)*  
  
  
make checkconfig  
*look for any errors*  
  
asterisk -r  
>restart now  
  
Now go to your web browser and log on to the GUI at 'http://ip:8088/gui/static/config/cfgbasic.html' using the [admin] and 'secret' username/password.  It will take a few seconds to generate the new scripts and prompt you when complete.

**5. Running asterisk-gui**  
In order to load the asterisk-gui, asterisk must restart/reload. You can reload your Asterisk server from your CLI console by executing the command `*reload*`. You can use asterisk-gui from these addresses (*10.100.100.75 is our Asterisk server IP address*):

http://10.100.100.75:8088/asterisk/static/config/cfgbasic.html  
http://10.100.100.75:8088/asterisk/static/config/cfgadvanced.html

You are now done with the Asterisk GUI install.  You can continue with the rest of the install as required as per the following install procedure starting here:

**Installation of SSHPASS:**

# install from [EPEL](https://www.server-world.info/en/note?os=CentOS_6&p=initial_conf&f=6)

[root@dlp ~]#

[yum](https://www.server-world.info/en/command/html/yum.html) --enablerepo=epel -y install sshpass

# -p password : from argument

[cent@dlp ~]$

sshpass -p password ssh 10.0.0.51 hostname

node01.srv.world

# -f file : from file

[cent@dlp ~]$

[echo](https://www.server-world.info/en/command/html/echo.html) 'password' > sshpass.txt

[cent@dlp ~]$

[chmod](https://www.server-world.info/en/command/html/chmod.html) 600 sshpass.txt

[cent@dlp ~]$

sshpass -f sshpass.txt ssh 10.0.0.51 hostname

node01.srv.world

# -e : from env variable

[cent@dlp ~]$

export SSHPASS=password

[cent@dlp ~]$

sshpass -e ssh 10.0.0.51 hostname

node01.srv.world

**PXE Server configuration:**

<https://www.maketecheasier.com/configure-pxe-server-ubuntu/>

In this article we will show you how you can configure a PXE server in Ubuntu 14.04.  
Configure Networking  
  
To get started, you need to first set up your PXE server to use a static IP. To set up a static IP address in your system, you   
  
need to edit the “/etc/network/interfaces” file.  
  
1. Open the “/etc/network/interfaces” file.  
  
sudo nano /etc/network/interfaces  
  
Add/edit as described below:  
  
# The loopback network interface  
auto lo  
iface lo inet loopback  
# The primary network interface  
auto eth0  
iface eth0 inet static  
address 192.168.1.20  
netmask 255.255.255.0  
gateway 192.168.1.1  
dns-nameservers 8.8.8.8  
  
Save the file and exit. This will set its IP address to “192.168.1.20”. Restart the network service.  
  
sudo /etc/init.d/networking restart  
  
Install DHCP, TFTP and NFS:  
  
DHCP, TFTP and NFS are essential components for configuring a PXE server. First you need to update your system and install all   
  
necessary packages.  
  
For this, run the following commands:  
  
sudo apt-get update  
sudo apt-get install isc-dhcp-Server inetutils-inetd tftpd-hpa syslinux nfs-kernel-Server  
  
Configure DHCP Server:  
  
DHCP stands for Dynamic Host Configuration Protocol, and it is used mainly for dynamically distributing network configuration   
  
parameters such as IP addresses for interfaces and services. A DHCP server in PXE environment allow clients to request and   
  
receive an IP address automatically to gain access to the network servers.  
  
1. Edit the “/etc/default/dhcp3-server” file.  
  
sudo nano /etc/default/dhcp3-server  
  
Add/edit as described below:  
  
INTERFACES="eth0"  
  
Save (Ctrl + o) and exit (Ctrl + x) the file.  
  
2. Edit the “/etc/dhcp3/dhcpd.conf” file:  
  
sudo nano /etc/dhcp/dhcpd.conf  
  
Add/edit as described below:  
  
default-lease-time 600;  
max-lease-time 7200;  
subnet 192.168.1.0 netmask 255.255.255.0 {  
range 192.168.1.21 192.168.1.240;  
option subnet-mask 255.255.255.0;  
option routers 192.168.1.20;  
option broadcast-address 192.168.1.255;  
filename "pxelinux.0";  
next-Server 192.168.1.20;  
}  
  
Save the file and exit.  
  
3. Start the DHCP service.  
  
sudo /etc/init.d/isc-dhcp-server start  
  
Configure TFTP Server:  
  
TFTP is a file-transfer protocol which is similar to FTP. It is used where user authentication and directory visibility are not   
  
required. The TFTP server is always listening for PXE clients on the network. When it detects any network PXE client asking for   
  
PXE services, then it provides a network package that contains the boot menu.  
  
1. To configure TFTP, edit the “/etc/inetd.conf” file.  
  
sudo nano /etc/inetd.conf  
  
Add/edit as described below:  
  
tftp dgram udp wait root /usr/sbin/in.tftpd /usr/sbin/in.tftpd -s /var/lib/tftpboot  
  
Save and exit the file.  
  
2. Edit the “/etc/default/tftpd-hpa” file.  
  
sudo nano /etc/default/tftpd-hpa  
  
Add/edit as described below:  
  
TFTP\_USERNAME="tftp"  
TFTP\_DIRECTORY="/var/lib/tftpboot"  
TFTP\_ADDRESS="[:0.0.0.0:]:69"  
TFTP\_OPTIONS="--secure"  
RUN\_DAEMON="yes"  
OPTIONS="-l -s /var/lib/tftpboot"  
  
Save and exit the file.  
  
3. Enable boot service for inetd to automatically start after every system reboot and start tftpd service.  
  
sudo update-inetd --enable BOOT  
sudo service tftpd-hpa start  
  
4. Check status.  
  
sudo netstat -lu  
  
It will show the following output:  
  
Proto Recv-Q Send-Q Local Address Foreign Address State  
udp 0 0 \*:tftp \*:\*  
  
Configure PXE boot files  
  
Now you need the PXE boot file “pxelinux.0” to be present in the TFTP root directory. Make a directory structure for TFTP, and   
  
copy all the bootloader files provided by syslinux from the “/usr/lib/syslinux/” to the “/var/lib/tftpboot/” path by issuing the   
  
following commands:  
  
sudo mkdir /var/lib/tftpboot  
sudo mkdir /var/lib/tftpboot/pxelinux.cfg  
sudo mkdir -p /var/lib/tftpboot/Ubuntu/14.04/amd64/  
sudo cp /usr/lib/syslinux/vesamenu.c32 /var/lib/tftpboot/  
sudo cp /usr/lib/syslinux/pxelinux.0 /var/lib/tftpboot/  
  
Set up PXELINUX configuration file  
  
The PXE configuration file defines the boot menu displayed to the PXE client when it boots up and contacts the TFTP server. By   
  
default, when a PXE client boots up, it will use its own MAC address to specify which configuration file to read, so we need to   
  
create that default file that contains the list of kernels which are available to boot.  
  
Edit the PXE Server configuration file with valid installation options.  
  
To edit “/var/lib/tftpboot/pxelinux.cfg/default,”  
  
sudo nano /var/lib/tftpboot/pxelinux.cfg/default  
  
Add/edit as described below:  
  
DEFAULT vesamenu.c32  
TIMEOUT 100  
PROMPT 0  
MENU INCLUDE pxelinux.cfg/PXE.conf  
NOESCAPE 1  
LABEL Try Ubuntu 14.04 Desktop  
MENU LABEL Try Ubuntu 14.04 Desktop  
kernel Ubuntu/vmlinuz  
append boot=casper netboot=nfs nfsroot=192.168.1.20:/var/lib/tftpboot/Ubuntu/14.04/amd64  
initrd=Ubuntu/initrd.lz quiet splash  
ENDTEXT  
LABEL Install Ubuntu 14.04 Desktop  
MENU LABEL Install Ubuntu 14.04 Desktop  
kernel Ubuntu/vmlinuz  
append boot=casper automatic-ubiquity netboot=nfs nfsroot=192.168.1.20:/var/lib/tftpboot/Ubuntu/14.04/amd64  
initrd=Ubuntu/initrd.lz quiet splash  
ENDTEXT  
  
Save and exit the file.  
  
Edit the “/var/lib/tftpboot/pxelinux.cfg/pxe.conf” file.  
  
sudo nano /var/lib/tftpboot/pxelinux.cfg/pxe.conf  
  
Add/edit as described below:  
  
MENU TITLE PXE Server  
NOESCAPE 1  
ALLOWOPTIONS 1  
PROMPT 0  
MENU WIDTH 80  
MENU ROWS 14  
MENU TABMSGROW 24  
MENU MARGIN 10  
MENU COLOR border 30;44 #ffffffff #00000000 std  
  
Save and exit the file.  
Add Ubuntu 14.04 Desktop Boot Images to PXE Server  
  
For this, Ubuntu kernel and initrd files are required. To get those files, you need the Ubuntu 14.04 Desktop ISO Image. You can   
  
download the Ubuntu 14.04 ISO image in the /mnt folder by issuing the following command:  
  
sudo cd /mnt  
sudo wget http://releases.ubuntu.com/14.04/ubuntu-14.04.3-desktop-amd64.iso  
  
Note: the download URL might change as the ISO image is updated. Check out this website for the latest download link if the above   
  
URL is not working.  
  
Mount the ISO file, and copy all the files to the TFTP folder by issuing the following commands:  
  
sudo mount -o loop /mnt/ubuntu-14.04.3-desktop-amd64.iso /media/  
sudo cp -r /media/\* /var/lib/tftpboot/Ubuntu/14.04/amd64/  
sudo cp -r /media/.disk /var/lib/tftpboot/Ubuntu/14.04/amd64/  
sudo cp /media/casper/initrd.lz /media/casper/vmlinuz /var/lib/tftpboot/Ubuntu/  
  
Configure NFS Server to Export ISO Contents  
  
Now you need to setup Installation Source Mirrors via NFS protocol. You can also use http and ftp for Installation Source   
  
Mirrors. Here I have used NFS to export ISO contents.  
  
To configure the NFS server, you need to edit the “/etc/exports” file.  
  
sudo nano /etc/exports  
  
Add/edit as described below:  
  
/var/lib/tftpboot/Ubuntu/14.04/amd64 \*(ro,async,no\_root\_squash,no\_subtree\_check)  
  
Save and exit the file. For the changes to take effect, export and start NFS service.  
  
sudo exportfs -a  
sudo /etc/init.d/nfs-kernel-server start  
  
Now your PXE Server is ready.

### Centos sever back-up command

 tar -zcvpf /backups/fullbackup.tar.gz --directory=/ --exclude=proc --exclude=sys --exclude=dev/pts --exclude=backups.  
  
cd /  
mkdir backups  
cd backups  
  
nano fullserver.sh  
  
tar -cvpf /backups/fullbackup.tar --directory=/ --exclude=proc --exclude=sys --exclude=dev/pts --exclude=backups .

chmod 750 /backups/fullserver.sh ./backups/fullserver.sh

**Kick Start and PXE server Installation steps**

Dear Sikhamani,

Me and Javed have worked and successfully tested  the concept of Installing Ubuntu OS at a time in multiple new PC/WS, using **Ubuntu PXE Server**, please find the configuration and installation process details mentioned below:

Requirements:

1) IOS Image of Ubuntu 14 (or any ver OS).

2) A WS installed with Ubuntu 14.04 OS or above.

3) LAN setup where client WS can communicate with Server.

Procedure:

1) Need to copy Ubuntu IOS image in the Ubuntu PXE server.

2) Need to configure network interface eth0 of Uuntu PXE server by below command an append the details in file:

**Note: words colored in blue are the variable which we are adding/appending in configuration files:**

 #sudo nano /etc/network/interfaces

*# The loopback network interface*

**auto lo**

**iface lo inet loopback**

***# The primary network interface***

**auto eth0**

**iface eth0 inet static**

**address 192.168.1.240**

**netmask 255.255.255.0**

**gateway 192.168.1.1**

**dns-nameservers 8.8.8.8**

Save the changes and run below command:

**#sudo** **/**etc**/**init.d**/**networking restart

3) Now need to install DHCP, TFTP & NFS packages in Ubuntu PXE server, by using below command;

**#sudo** **apt-get update**

**#sudo** **apt-get install** **isc-dhcp-Server inetutils-inetd tftpd-hpa syslinux nfs-kernel-Server**

  After the packages install successfully, need to configure individual services.

4) Configuring DHCP in Server for IP assigning to the PXE Server clients.

5) Edit the “/etc/default/dhcp3-server”  dhcp configuration file.

**# sudo** **nano** **/**etc**/**default**/**dhcp3-server

  A file will open at the end append this below line save & close:

**INTERFACES="eth0"**

Which will tel server to use Interface eth0 for assigning DNCP IP aggress to PXE client machine while OS installation.

6) Now we need to configure Varibles or rule on DHCP server would function as mentioned below:

Open and edit:

**# sudo** **nano** **/**etc**/**dhcp**/**dhcpd.conf

Add below line at appropriate place mentioned in the configuration file which we will find in file as mentioned below:

 # option definitions common to all supported networks...

option domain-name "**hostname**";   #hostname of your System

option domain-name-servers **hostname**;

**default-lease-time 600;**

**max-lease-time 7200;**

**subnet 192.168.1.0 netmask 255.255.255.0 {**

**range 192.168.1.21 192.168.1.240;**

**option subnet-mask 255.255.255.0;**

**option routers 192.168.1.20;**

**option broadcast-address 192.168.1.255;**

**filename "pxelinux.0";**

**next-Server 192.168.1.20;**

**}**

 [...]

# network, the authoritative directive should be uncommented.

**authoritative;**

Save and close the file and restart the DHCP service using below command:

**#sudo** **/**etc**/**init.d**/**isc-dhcp-server start

7) Now configuring TFTP server in Ubuntu PXE server.

To configure TFTP, edit the “/etc/inetd.conf” file.

**# sudo** **nano** **/**etc**/**inetd.conf

Add below mentioned lines in middile of the file:

**tftp dgram udp wait root /usr/sbin/in.tftpd /usr/sbin/in.tftpd –s /var/lib/tftpboot**

Which tells the server the authentication and the path of OS image file.

Save and exit the file.

Not to configure TFTP service Edit the “/etc/default/tftpd-hpa” file.

**# sudo** **nano** **/**etc**/**default**/**tftpd-hpa

Add below mentioned lines in the file:

**TFTP\_USERNAME="tftp"**

**TFTP\_DIRECTORY="/var/lib/tftpboot"**

**TFTP\_ADDRESS="[:0.0.0.0:]:69"**

**TFTP\_OPTIONS="--secure"**

**RUN\_DAEMON="yes"**

**OPTIONS="-l -s /var/lib/tftpboot"**

Save and exit the file.

Now we need to enable boot service for inetd to automatically start after every system reboot and start tftpd service by below commands.

**# sudo** update-inetd --enable BOOT

**# sudo** service tftpd-hpa start

For conformation weather service is running or not we can use below command to find in in the list

;

**# sudo** **netstat** -lu

8) Now extracting and copying ISO image file to TFTP server nfs share folder path for remote access.

(Note: need to keep Ubuntu ISO image in /home/ubuntu-image location before running below set of commands)

**sudo** **mkdir** **/**var**/**lib**/**tftpboot

**sudo** **mkdir** **/**var**/**lib**/**tftpboot**/**pxelinux.cfg

**sudo** **mkdir** -p **/**var**/**lib**/**tftpboot**/**Ubuntu**/**14.04**/i386/**

**sudo** **cp** **/**usr**/**lib**/**syslinux**/**vesamenu.c32 **/**var**/**lib**/**tftpboot**/**

**sudo** **cp** **/**usr**/**lib**/**syslinux**/**pxelinux.0 **/**var**/**lib**/**tftpboot**/**

**sudo** **mount** -o loop **/home/**ubuntu-image/Ubuntu-14.04-LTS.iso **/**media**/**

**sudo** **cp** -r **/**media**/\*** **/**var**/**lib**/**tftpboot**/**Ubuntu**/**14.04**/i386/**

**sudo** **cp** -r **/**media**/**.disk **/**var**/**lib**/**tftpboot**/**Ubuntu**/**14.04**/i386/**

**sudo** **cp** **/**media**/**casper**/**initrd.lz **/**media**/**casper**/**vmlinuz **/**var**/**lib**/**tftpboot**/**Ubuntu**/**

9) Now setting up PXE server Varibles by editing /var/lib/tftpboot/pxelinux.cfg/default file as metioned below:

Open file by:

**# sudo** **nano** **/**var**/**lib**/**tftpboot**/**pxelinux.cfg**/**default

Add below mentioned line in to the file:

**DEFAULT vesamenu.c32**

**TIMEOUT 100**

**PROMPT 0**

**MENU INCLUDE pxelinux.cfg/PXE.conf**

**NOESCAPE 1**

**LABEL Try Ubuntu 14.04-LTS**

**MENU LABEL Try Ubuntu 14.04-LTS**

**kernel Ubuntu/vmlinuz**

**append boot=casper netboot=nfs nfsroot=192.168.1.240:/var/lib/tftpboot/Ubuntu/14.04/i386 initrd=Ubuntu/initrd.lz quiet splash**

**ENDTEXT**

**LABEL Install Ubuntu 14.04-LTS**

**MENU LABEL Install Ubuntu 14.04-LTS**

**kernel Ubuntu/vmlinuz**

**append boot=casper automatic-ubiquity netboot=nfs nfsroot=192.168.1.240:/var/lib/tftpboot/Ubuntu/14.04/i386 initrd=Ubuntu/initrd.lz quiet splash**

**ENDTEXT**

Save and close the file:

10) Now need to edit 1 more file Edit the “/var/lib/tftpboot/pxelinux.cfg/pxe.conf” to do so use below command.

Edit the “/var/lib/tftpboot/pxelinux.cfg/pxe.conf” file.

**# sudo** **nano** **/**var**/**lib**/**tftpboot**/**pxelinux.cfg**/**pxe.conf

Add/edit as described below:

**MENU TITLE PXE Server**

**NOESCAPE 1**

**ALLOWOPTIONS 1**

**PROMPT 0**

**MENU WIDTH 80**

**MENU ROWS 14**

**MENU TABMSGROW 24**

**MENU MARGIN 10**

**MENU COLOR border 30;44 *#ffffffff #00000000 std***

Save and exit the file.

11) now configuring NFS in Ubuntu PXE server, do blow steps:

To configure the NFS server, you need to edit the “/etc/exports” file.

**# sudo** **nano** **/**etc**/**exports

Add/edit as described below:

**/var/lib/tftpboot/Ubuntu/14.04/amd64 ro,async,no\_root\_squash,no\_subtree\_check)**

Save and exit the file. For the changes to take effect, export and start NFS service by below command.

**# sudo** exportfs -a

**# sudo** **/**etc**/**init.d**/**nfs-kernel-server start

Now your PXE Server is ready for providing OS file for remote installation without using CD/DVD in the new WS.

12) To access Ubuntu PXE server at client end we need to Boot to Network boot mode at Boot menu, it will access our newly configured Ubuntu pxe server and will prompt of 2 options as show in below image, we can chose any of as per our requirement.

There is other Option we were we can install OS unattended (without being preset at client WS by just enabling Network boot) by using Kickstart conf, but there are some complication like:

a) All system needs to be of same brand or Manufacturer.

b) Need Internet access for taking updates else installation will stop with error.

c) one mistake in kickstart.cgf (configuration file) will end up with huge errors.

We are still working on it but possibilities are less, hence request team to share your finding and work on the concern.

**Installation of WEBMIN application:**

Installing Web-min in Linux

To install Webmin in tecmin

# cd /opt

# wget http://www.webmin.com/jcameron-key.asc

# wget http://www.webmin.com/download/rpm/webmin-current.rpm

# rpm --import jcameron-key.asc

# rpm -Uvh webmin-\*.rpm

To start service webmin

# /etc/init.d/webmin start

# /etc/init.d/webmin stop

# /etc/init.d/webmin restart

# /etc/init.d/webmin status

To change password of webmin access

# /usr/libexec/webmin/changepass.pl /etc/webmin root secretpassword

login in browser with https://IP-Address:10000/

Officeal websir on Web min

wget http://prdownloads.sourceforge.net/webadmin/webmin-1.801-1.noarch.rpm

yum -y install perl perl-Net-SSLeay openssl perl-IO-Tty

rpm -U webmin-1.801-1.noarch.rpm

or

If you like to install and update Webmin via RPM, create the /etc/yum.repos.d/webmin.repo file containing :

[Webmin]

name=Webmin Distribution Neutral

#baseurl=http://download.webmin.com/download/yum

mirrorlist=http://download.webmin.com/download/yum/mirrorlist

enabled=1

You should also fetch and install my GPG key with which the packages are signed, with the commands :

wget http://www.webmin.com/jcameron-key.asc

rpm --import jcameron-key.asc

You will now be able to install with the command :

yum install webmin

**Automated Call recording Back-up script**

As you suggested, we have tested the script which I have prepared and its output in testing environment and was able to achieved the goal of Automated backup from system to system and system to EHDD, if EHDD is permanently mounted to system by assigning a cron job at scheduled time.

**Details on how script & cron job works in live environment:**

**1.Script to execute Call recording Backup of previous day  from 1.5 to 1.8.**

a)Login s root and create as script file named “daily-backup” by following commands:

**#nano /home/daily-backup.sh**          (create a file in home folder and append below lines and save it)

#!/bin/sh

Bkpdate=$(date +%Y-%m-%d --date="1 day ago")

#########Daily call recording backup for 104###########

scp -r /var/www/html/calls/$Bkpdate @192.168.1.8:/var/www/html/calls

du -h /var/www/html/calls/ > /home/Daily-backup-report.txt

Save and close the file......

(du –h this command will prepare report of backup call recording folder size and save it to Daily-backup-report.txt file for documentation)

**#chmod 777 /home/daily-backup.sh**           ( run this command to give full permission to execute in shell no matter who runs it)

(Note: need to monitor disc space of 1.8 server before 12 AM and make free space for backup)

**Now Script is ready need to assign it in cron job, to do so we need to do below steps:**

**#crontab –e**                       (cron job file will open need to append below lines to schedule a job)

At the last line mention below job entry to check job can use "crontab -l" command

**##########Daily-backup of Call recordings from 1.5 to 1.8##########**

**45 00 \* \* \* /home/daily-backup.sh**

The above line will execute “**daily-backup.sh**” script at 12:45 AM ( I am mentioning 12:45 am time to make sure that non-of the call is missed) and will copy previous day call recording from 1.5 to 1.8 server.

**2.** Same way we can make a script & assign a cron job in 1.8 to copy call recordings to EHDD which is permanently connected and mounted (we can also disconnect the EHDD after backup but only thing is we need to reconnect it daily night make sure disc space is available and remount it. Need to take care that it should be the only EHDD connected to the 1.8 server to avoid mount point issue)

**Script for backup call recording from 1.8 to EHDD is below:**

**#nano /home/daily-backup-ehdd.sh**                           (and append below lines and save it)

#!/bin/sh

#########Daily call recording backup for EHDD###########

mount –a /dev/sdc1 /mnt/BKP

Bkpdate=$(date +%Y-%m-%d --date="1 day ago")

cp -r /var/www/html/calls/$Bkpdate /mnt/BKP/BackupCalls/2016-Backup-calls/

du -h /var/www/html/calls/ > /home/Daily-backup-report-1-8.txt            (this command will prepare report of backup call recording folder size and save it to Daily-backup-report-1-8.txt file for documentation)

**Now Script is ready need to assign it in cron job, to do so we need to do below steps:**

**#crontab –e**                       (cron job file will open need to append below lines to schedule a job)

At the last line mention below job entry

**##########Daily-backup of Call recordings from 1.8 to EHDD##########**

**45 01 \* \* \* /home/daily-backup-ehdd.sh**

The above line will execute “**daily-backup-ehdd.sh**” script at 1:45 AM.

**3.**The task of doing Automated backup for call recording to **floor WS** cannot be done automated because:

1) Every WS is assigned with unique IP

2) Every week we change WS if the partition get filled

3) We have enabled the hiding of partition for Audit policy requirement, so to enable it we need to manually mount it  in Disc-utility.

Nevertheless for time being we can perform this activity manually by command for as usual:

Login to 1.8 server and enter below command if you are sure of the floor WS partition path.

Note: Floor WS were we are trying to do backup should be installed with SSH service else below command will not work as we already know.

**# scp -r /var/www/html/calls/2016-06-09 @192.168.1.140:/media/Backup/104/JUNE-2016**

**“Provide root password of floor WS when prompted for password”**

Above command source is 1.8 server and destination is 192.168.1.140 i.e. WS-60 to partition called /media/Backup and to the folder called /104/JUNE-2016, which is in “/media/Backup” partition.

Sir, kindly cross verify above procedure for any flaws and if You permit we will try it on live server tonight or demonstrate it to you.

**Hiding Sysytem Partation in Ubuntu:**

As per your Update & suggestion worked on hiding call recording Back-up partition which is available in floor WS, was able to find below mentioned opting to hide partition for all users including root privileged.

**To Hide Back-up partition in Ubuntu 12.04 follow below steps:**  
  
Open Terminal and login as root  
  
type fdisk command to view details of the partition to be hide  
  
# sudo fdisk -l  
  
it will display some thing like mentioned below:  
  
  Device Boot      Start         End      Blocks   Id  System  
/dev/sda1   \*        2048   120000511    59999232   83  Linux  
/dev/sda2       120005550   976768064   428381257+   7  HPFS/NTFS/exFAT  
  
now we want to hide /dev/sda2 which is our back-up drive or Partition.  
  
to do so follow below procedure:  
  
create a file named 99-hide-disks.rules in root folder  
  
# sudo gedit 99-hide-disks.rules  
  
in this file copy paste these below lines  
  
KERNEL=="sda2", ENV{UDISKS\_PRESENTATION\_HIDE}="1"  
KERNEL=="sda2", ENV{UDISKS\_IGNORE}="1"  
  
and save the file and close it.  
  
Now copy the file to /etc/udev/rules.d/ with the command,  
  
# sudo cp 99-hide-disks.rules  /etc/udev/rules.d/  
  
now reboot the system to see the effects i.e. Back-up partition will be hidden until we mount it manually from Disk Utility.  
 **To Hide Back-up partition in Ubuntu 14.04 follow below steps:**  
  
    **Press the super/windows key  
    Type and open disks utility  
    Select the partition you want to hide so it becomes orange  
    select the cog/settings/advanced settings or Mount/Unmount settings icon.   
    Click Edit Mount Options  
    Turn auto mount off  
    Uncheck both Mount at Startup and Show in user interface  
    Apply and save the changes.  
    Save & Reboot**  
  
Now when you login to system and open File option Back-up Drive or partition will not be visible until you go to Disc utility and enable the partition.  
  
**Disc Utility and  Select the partition you want to hide so it becomes orange**  
    **select the cog/settings/advanced settings or Mount/Unmount settings icon.   
    Click Edit Mount Options  
    Turn auto mount On  
    Uncheck both Mount at Startup and Show in user interface  
    Apply and save the changes.  
    Unmount and remount the drive or partition, then it will display the Back-up partition .**  
Same has been configured in WS-90 (Ubuntu 12.04) and WS-46 (Ubuntu 14.04) and was successfully hide the partition, team please cross check and suggest for any changes or enhancements.

**ALSA Drivers iNstallation**

Type 2 method

sudo apt-get --purge remove linux-sound-base alsa-base alsa-utils

sudo apt-get install linux-sound-base alsa-base alsa-utils gdm ubuntu-desktop

**GRUB changes required for Booting Dual OS in Ubuntu:**

**sudo add-apt-repository ppa:yannubuntu/boot-repair && sudo apt-get update**

**Press Enter**

**Then type:**

**sudo apt-get install -y boot-repair && boot-repair**

**Find out on what hard disk and what partition WINDOWS is installed.**

**This you can do after opening a terminal and type at the prompt**

**sudo fdisk -l**

**(this must be done with administative rights, so you will be ask for your password). You will now get a table of your partitions with device names like /dev/sda1, /dev/sda2 .. and so on.**

**If you have installed windows it might have type NTFS. This means if the second entry in this table is like this:**

**Device...Boot.....Start...End......Blocks......Size .........Id.....System**

**/dev/sda1..\*...2048..467118879..467116032..222,8G ..7.....Linux**

**/dev/sda2 ..467118080...961144831....494026552..235,6G 7.....HPFS/NTFS/exFAT**

**/dev/sda3 ..1270138878...1953523711..341692417..7,6G 5.....Extended**

**/dev/sda5 ..1270138880...1945182207..337521664..7,6G 83....Linux**

**windows is installed on the second partition of the first hard disk (sda2)**

**(In my case both operating systems were installed on one hard disk with 2 partitions, on the first I had installed UBUNTU and Windows 7 on the second. ) Press C for accessing the GRUB command prompt. At the Grub-Prompt enter the following 3 commands:**

**Grub> set root=(hd0,2)**

**Grub> chainloader +1**

**Grub> boot**

**Notice: In the command set root = (hd0,2) after hd type the (zero-based) number of the hard disk (here 0 ) and after the Comma the Number of the partion (here 2) . If your windows partition ist sdb1 you must type set root=(hd1,1) if it is sdb4 type set root=(hd1,4) etc…**

**Now Windows should boot!**

**Now you might want to have Windows 7 permanently as a menu item in der GRUB Boatloader menu. This you can get as follows:**

**menuentry 'Windows 7' {**

**set root=(hd0,2)**

**chainloader +1**

**boot**

**}**

**Save the file and close it.**

**Open a terminal and type on the command prompt:**

**sudo update-grub**

**Installation of TCMDUMP:**

Many of Linux distributions already shipped with **tcpdump** tool, if in case you don’t have it on systems, you can install it using following Yum command.

**# yum install tcpdump**

**# tcpdump -i eth0**

**# tcpdump -c 5 -i eth0**

**# tcpdump -A -i eth0**

**# tcpdump -D**

The following command with option **-XX** capture the data of each packet, including its link level header in **HEX** and **ASCII** format.

**# tcpdump -XX -i eth0**

#### Capture and Save Packets in a File

As we said, that **tcpdump** has a feature to capture and save the file in a **.pcap** format, to do this just execute command with **-w** option.

**# tcpdump -w 0001.pcap -i eth0**

**Also**

**C:\Program Files\Wireshark\dumpcap.exe -c 50 -i {interface name or number} -w {wherever you want to save the packet capture file}**

**tcpdump -w App-WS-6.pcap -i eth0 dst 192.168.1.194 and port 5060**

**http://www.en.voipforo.com/QoS/QoS\_PacketLoss.php**

**http://www.en.voipforo.com/QoS/QoS\_Bandwith.php**

**http://www.en.voipforo.com/QoS/QoS\_Jitter.php**

**http://www.en.voipforo.com/QoS/QoS\_Latency.php**

**Installing Audit log for file :**

**Auditing Files and folders in Centos**

**Configuring Aduti**

**-w /home/DB-upload-ws-115.sh -k DB-file-115**

**-w /home/daily-DB-Upload.sh -k DB-File-up**

**sudo yum list audit audit-libs**

**auditctl -l**

**ausearch -f /etc/passwd**

**auditctl -a exit,always -F path=cat /home/DB-upload-ws-115.sh -F perm=wa**

**cat /var/log/audit/audit.log | grep TEST**

**auditctl -w /home/DB-upload-ws-115.sh -p war -k password-file**

**sudo auditctl -w /etc/ssh/sshd\_config -p rwxa -k sshconfigchange**

**sudo nano /etc/audit/auditd.confauditctl -l**

**aureport -k -i | grep TEST to check access date**

**Keeping track of all events by a user**

**Here 501 is the uid for user "deepak" which you can check using id command ans shown below**

**# id deepak**

**uid=501(deepak) gid=501(TEST\users) groups=501(TEST\users)**

**# ausearch -ui 501 --interpret**

**Keeping track of all events for a specific date and time**

**# ausearch -ui 501 --start 02/01/14 00:00:00 --end 02/02/14 00:00:00**

**To list all the audit report**

**# aureport**

**aureport -au -i --success for ssh sucess login**

**ausyscall x86\_64 188**

**Auditing of processes under Linux**

**Similiar to using strace, the audit framework has a tool named autrace. It uses the audit framework and adds the right rules to capture information and log it. Using ausearch the gathered information can be displayed.**

**Perform a trace:**

**root@host:~# autrace /bin/ls /tmp**

**autrace cannot be run with rules loaded.**

**Please delete all rules using ‘auditctl -D’ if you really wanted to**

**run this command.**

**root@host:~# auditctl -D**

**No rules**

**root@host:~# autrace /bin/ls /tmp**

**Waiting to execute: /bin/ls**

**atop.d mc-root mongodb-27017.sock suds**

**Cleaning up…**

**Trace complete. You can locate the records with ‘ausearch -i -p 20314’**

**ausearch –start recent -p 21023 –raw | aureport –file –summary**

**Audit file access per user**

**The audit framework can be used to monitor syscalls, including access to files. If you want to know what files a particular user ID accessed, use a rule like this:**

**auditctl -a exit,always -F arch=x86\_64 -S open -F auid=80**

**-F arch=x86\_64 Define what architecture is used (uname -m), to monitor the right syscall (some system calls are ambiguous between archtectures).**

**-S open Select the “open” syscall**

**-F auid=80 The related user ID**

**This kind of information is really useful for intrusion detection, but also when performing forensics on a Linux system.**

**Alternatively use below for reference:**

There are two main parts to the audit system:

1. The audit kernel component intercepts system calls from user applications, records events, and sends these audit messages to the audit daemon
2. The auditd daemon collects the information from the kernel and creates entries in a log file

The audit system uses the following packages: audit and audit-libs. These packages are installed by default on a new CentOS 7 Droplet (and a new CentOS 6 Droplet). It is good to verify that you have them installed on your server using:

* sudo yum list audit audit-libs

You should see both the packages under Installed Packages in the output:

Installed Packages

audit.x86\_64

audit-libs.x86\_64

## Configuring Audit

The main configuration file for auditd is /etc/audit/auditd.conf. This file consists of configuration parameters that include where to log events, how to deal with full disks, and log rotation. To edit this file, you need to use sudo:

* sudo nano /etc/audit/auditd.conf

For example, to increase the number of audit log files kept on your server to 10, edit the following option:

/etc/audit/auditd.conf

num\_logs = 10

You can also configure the maximum log file size in MB and what action to take once the size is reached:

/etc/audit/auditd.conf

max\_log\_file = 30

max\_log\_file\_action = ROTATE

When you make changes to the configuration, you need to restart the auditd service using:

* sudo service auditd restart

for the changes to take effect.

The other configuration file is /etc/audit/rules.d/audit.rules. (If you are on CentOS 6, the file is /etc/audit/audit.rules instead.) It is used for permanently adding auditing rules.

When auditd is running, audit messages will be recorded in the file /var/log/audit/audit.log.

## Understanding Audit Log Files

By default, the audit system logs audit messages to the /var/log/audit/audit.log file. Audit log files carry a lot of useful information, but reading and understanding the log files can seem difficult for many users due to the sheer amount of information provided, the abbreviations and codes used, etc. In this section, we will try to understand some of the fields in a typical audit message in the audit log files.

**'Note:** If auditd is not running for whatever reason, audit messages will be sent to rsyslog.

For this example, let us assume we have an audit rule configured on the server with the label (key) sshconfigchange to log every access or modification to the file /etc/ssh/sshd\_config. If you wish, you can add this rule temporarily using:

* sudo auditctl -w /etc/ssh/sshd\_config -p rwxa -k sshconfigchange

Running the following command to view the sshd\_config file creates a new **event** in the audit log file:

* sudo cat /etc/ssh/sshd\_config

This event in the audit.log file looks as follows:

/var/log/audit/audit.log

**Installation of Dstat tool:**

# yum install dstat

# dstat -c --top-cpu -d --top-bio --top-latency

To send the output to a csv file for later use we can issue the following command:

# dstat --output /tmp/sampleoutput.csv -cdn

**SSH Permissions Access ACL list configuration:**

**utmpdump /var/log/wtmp | cat to check logins**

**cat /var/log/messages | grep shutdown to check shutdown message log**

**/var/log/messages**

**Actually, that's the broadcast address (also represented as the MAC address ff:ff:ff:ff:ff:ff depending on the protocol used) for IPs 192.168.1.1 thru 192.168.1.254--if you send a message to this IP (192.168.1.255) all the machines on the LAN will receive the message. So, it could be generated internally or externally (since you receive all broadcast messages, even the ones that you sent).**

**Typically, in a Windows based LAN (or \*nix based running Samba), it's generated by one machine trying to resolve a local name using the NetBIOS Name Service on port 137. For instance, when you ping a non-existant machine on your LAN, say "toad", your computer broadcasts (to the entire 192.168.1 LAN) a query via the NBNS that tries to resolve "toad" to an IP or MAC address, at that point it might try to look it up in the DNS--which will fail, since no machine named "toad" exists. Of course, if "toad" does exist it will reply with it's location.**

**192.168.1.255 is your computers sending local broadcasts, obviously you have your network setup as 192.168.1.X (probably 192.168.1.1 = router .2 =box etc etc) it is also a private subnet accessible only internally, this is why most manufacturers set the default dhcp scope to 192 etc etc.**

**An attacker could generate packets with this ip address included in hopes of gaining access, and though you can get pass the properties of NAT and access ones filtered server the process is extremely slow and bothersome. If are providing no services and have a private internal IP you are pretty safe. If you forward services you may want to look more into the security of those services.**

**/etc/hosts.allow**

**/etc/hosts.deny**

**Access controls that should be enforced by tcp-wrappers are defined here.**

**Further details are described in hosts\_access(5).**

**http://www.debian-administration.org/articles/87 offers these examples:**

**# /etc/hosts.allow**

**sshd: 1.2.3.0/255.255.255.0**

**sshd: 192.168.0.0/255.255.255.0**

**# /etc/hosts.deny**

**sshd: ALL**

**SSH login without password**

**http://www.linuxproblem.org/art\_9.html**

**Your aim**

You want to use Linux and OpenSSH to automate your tasks. Therefore you need an **automatic** login from host A / user a to Host B / user b. You don't want to enter any passwords, because you want to call ssh from a within a shell script.

**How to do it**

First log in on A as user a and generate a pair of authentication keys. Do not enter a passphrase:

a@A:~> ssh-keygen -t rsa

Generating public/private rsa key pair.

Enter file in which to save the key (/home/a/.ssh/id\_rsa):

Created directory '/home/a/.ssh'.

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /home/a/.ssh/id\_rsa.

Your public key has been saved in /home/a/.ssh/id\_rsa.pub.

The key fingerprint is:

3e:4f:05:79:3a:9f:96:7c:3b:ad:e9:58:37:bc:37:e4 a@A

Now use ssh to create a directory ~/.ssh as user b on B. (The directory may already exist, which is fine):

a@A:~> ssh b@B mkdir -p .ssh

b@B's password:

Finally append a's new public key to b@B:.ssh/authorized\_keys and enter b's password one last time:

a@A:~> cat .ssh/id\_rsa.pub | ssh b@B 'cat >> .ssh/authorized\_keys'

b@B's password:

From now on you can log into B as b from A as a without password:

a@A:~> ssh b@B

**A note** from one of our readers: Depending on your version of SSH you might also have to do the following changes:

* Put the public key in .ssh/authorized\_keys2
* Change the permissions of .ssh to 700
* Change the permissions of .ssh/authorized\_keys2 to 640

Real world steps

**#ssh-keygen -t rsa**

**Then just “enter” for all options of password**

**# ssh 192.168.1.55 mkdir -p .ssh**

**When prompted give 1.55 root password**

**# cat .ssh/id\_rsa.pub | ssh 192.168.1.55 'cat >> .ssh/authorized\_keys'**

**When prompted give 1.55 root password**

**# ssh 192.168.1.55 "chmod 700 .ssh; chmod 640 .ssh/authorized\_keys"**

**When prompted give 1.55 root password this is the last time we would give password**

**#ssh 192.168.1.55**

**Now it will not prompt you for password it will direct login.**

**IPtables and its usage**

**The following table explains the output columns.**

|  |  |
| --- | --- |
| remote | The sources defined at ntp.conf. '\*' indicates the current and best source; '+' indicates that these sources are available as NTP source. Sources with - are considered unusable. |
| refid | The IP address of the clock with which the remote server clock is synchronized with. |
| st | Stratum |
| t | Type. 'u' is for unicast. Other values may include local, multicast, broadcast. |
| when | The time elapsed (in seconds) since the last contact with the server. |
| poll | Polling frequency with the server in seconds. |
| reach | An octal value that indicates whether there are any errors in communication with the server. The value 377 indicates 100% success. |
| delay | The round trip time between our server and the remote server. |
| offset | The time difference between our server and the remote server in milliseconds. |
| jitter | The average time difference in milliseconds between two samples. |

\

To allow local ntp server in firewall and deny rest of ntp servers

# iptables -A INPUT -s 192.168.1.0/24 -p udp --dport 123 -j ACCEPT  
# iptables -A INPUT -p udp --dport 123 -j DROP

**Esc Disable commands**

Create a file and append below line

# nano /home/ psmri-hubli-89/xmap-1 (for disabling esc key)

# xmodmap -e 'keycode 9 = '

Save n exit

#ln -s /home/psmri-hubli-89/xmap-1 /usr/local/bin/xmap-1

#chmod +x /home/psmri-hubli-89/xmap-1

#xmap-1

Put it in a startup application or in cron job

Manual commands

xmodmap -e 'keycode 9 = '

xmodmap -e 'keycode 14 = '

**Installation of tuned admin tool:**

1. Installing tuned

# yum install tuned

2. To view a list of available tuning profiles

 [root@myCentOS ~]# tuned-adm list

Available profiles:

- laptop-ac-powersave

- server-powersave

- laptop-battery-powersave

- desktop-powersave

- virtual-host

- virtual-guest

- enterprise-storage

- throughput-performance

- latency-performance

- spindown-disk

- default

3. Tuning to a specific profile

# tuned-adm profile latency-performance

Switching to profile 'latency-performance'

Applying deadline elevator: dm-0 dm-1 dm-2 sda [ OK ]

Applying ktune sysctl settings:

/etc/ktune.d/tunedadm.conf: [ OK ]

Calling '/etc/ktune.d/tunedadm.sh start': [ OK ]

Applying sysctl settings from /etc/sysctl.conf

Starting tuned: [ OK ]

4. Checking current tuned profile used and its status

# tuned-adm active

Current active profile: latency-performance

Service tuned: enabled, running

Service ktune: enabled, running

5. Turning off the tuned daemon

# tuned-adm off

# Configuring Interface Bonding on CentOS/RHEL/OEL 6.x

In my [previous article](http://www.tokiwinter.com/configuring-interface-bonding-on-debian-wheezy/) I wrote about configuring configuring network interface bonding under Debian Wheezy. Here, I’ll briefly outline the steps required to get the same configuration running under recent RHEL-flavoured distributions – namely CentOS 6.4 in my case.

I will be bonding **eth0** and **eth1** into a bond named **bond0**. Ensure that you’re connected to your host via a console. I’ll be using active-backup (i.e. failover) bonding, but there are other options available – see the Debian article for links to reference material for those.

First, create the **ifcfg-bond0** configuration file:

# cd /etc/sysconfig/network-scripts

# vi ifcfg-bond0

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | # cd /etc/sysconfig/network-scripts  # vi ifcfg-bond0  DEVICE=bond0  IPADDR=192.168.122.12  NETMASK=255.255.255.0  GATEWAY=192.168.122.1  NM\_CONTROLLED=no  BOOTPROTO=none  ONBOOT=yes  USERCTL=no |

Substitute relevant values as appropriate for your setup. Next, edit/create the **ifcfg-eth{0,1}** files. Note that these are created as slave interfaces (**SLAVE=yes**) with **bond0** as the master interface (**MASTER=bond0**):

# vi ifcfg-eth0

DEVICE=eth0

USERCTL=no

# vi ifcfg-eth0

DEVICE=eth0

USERCTL=no

ONBOOT=yes

NM\_CONTROLLED=no

MASTER=bond0

SLAVE=yes

BOOTPROTO=none

# sed 's/eth0/eth1/' ifcfg-eth0 > ifcfg-eth1

Note – that if **NM\_CONTROLLED** is set, you should strictly define your **HWADDR** entries too at this step, for each interface. Configure the **bonding** module. **miimon** is the MII link monitoring frequency in milliseconds, **{down,up}delay** are the times, in milliseconds, to wait before disabling or enabling an interface in the bond (to safeguard against flapping), and should be a multiple of the miimon value. Note that **/etc/modprobe.conf** is deprecated in CentOS 6.x so an appropriate file should be created under **/etc/modprobe.d** – in our case, **bonding.conf**:

# vi /etc/modprobe.d/bonding.conf

alias bond0 bonding

options bond0 mode=active-backup miimon=100 downdelay=200 updelay=200

To test, manually load the module (and appropriate options – I see many tutorials with a simple **modprobe bonding** here – you’ll end up with the default bonding mode which is round-robin – not what we want):

|  |  |
| --- | --- |
| 1 | # modprobe bonding mode=active-backup miimon=100 downdelay=200 updelay=200 |

And restart networking:

# service network restart

Verify that all is well with **ifconfig -a**, or more suitably a **cat** on **/proc/net/bonding/bond0**:

|  |
| --- |
| # cat /proc/net/bonding/bond0  Ethernet Channel Bonding Driver: v3.6.0 (September 26, 2009)  Bonding Mode: fault-tolerance (active-backup)  Primary Slave: None  Currently Active Slave: eth0  MII Status: up  MII Polling Interval (ms): 100  Up Delay (ms): 200  Down Delay (ms): 200  Slave Interface: eth0  MII Status: up  Speed: Unknown  Duplex: Unknown  Link Failure Count: 0  Permanent HW addr: 52:54:00:c1:77:fc  Slave queue ID: 0  Slave Interface: eth1  MII Status: up  Speed: 100 Mbps  Duplex: full  Link Failure Count: 0  Permanent HW addr: 52:54:00:f3:11:1e  Slave queue ID: 0 |

Reboot the host at the earliest opportunity to verify that all is well after a reboot.

**Steps to Remove NIC bonding in LINUX CentOS**

**# ifconfig bond0 down**

**# echo "-eth0" > /sys/class/net/bond0/bonding/slaves**

**# echo "-eth1" > /sys/class/net/bond0/bonding/slaves**

**# echo "-bond0" > /sys/class/net/bonding\_masters**

**# rmmod bonding**

**# service network restart**

**DEVICE=bond0**

**IPADDR=192.168.1.10**

**NETWORK=192.168.1.0**

**NETMASK=255.255.255.0**

**USERCTL=no**

**BOOTPROTO=none**

**ONBOOT=yes**

**Changing Wallpaper remotely**

**Script to change wall paper**

**#!/bin/bash**

**SRC\_URI="http://192.168.1.8/calls/IT\_BAnner.jpg"**

**FNAME="/home/$USER/Pictures/temp.jpg"**

**wget "$SRC\_URI" -O "$FNAME"**

**gsettings set org.gnome.desktop.background picture-uri** [**file://$FNAME**](../../../)

**Commands to change Wall paper:**

**ssh -t psmri-hubli-59@192.168.1.141 'bash -s' '/home/wallpaper.sh'**

**ssh -t hmri-hubli-60@192.168.1.141 '/home/wallpaper.sh'**

**Other variable options tested**

**#! /bin/bash**

**SRC\_URI="http://192.168.1.8/calls/IT\_BAnner.jpg"**

**FNAME="/home/$USER/Pictures/black.png"**

**wget "$SRC\_URI" -O "$FNAME"**

**gsettings set org.gnome.desktop.background picture-uri "$FNAME"**

**#!/bin/bash**

**while [[ 1 -eq 1 ]]; do**

**for i in $(echo /usr/share/backgrounds/\*.jpg); do**

**echo $i**

**gsettings set org.gnome.desktop.background picture-uri file:///${i}**

**sleep 60;**

**done**

**done**

**#! /bin/bash**

**SRC\_URI="http://192.168.1.8/calls/IT\_BAnner.jpg"**

**FNAME="file:///home/IT\_BAnner.jpg"**

**wget "$SRC\_URI" -O "$FNAME"**

**gsettings set org.gnome.desktop.background picture-uri "$FNAME"**

**sshpass -p "psmri123" wget http://192.168.1.243/calls/IT\_BAnner.jpg -O /home/**

**su -c "df -h; mpstat -A -I CPU" &> /home/df-h-mpstat-Application && sshpass -p "Call@Bkp18" scp -r /home/df-h-mpstat-Application root@192.168.1.8:/var/www/html/calls/server-utilization-reports/**

**su -c "df -h; mpstat -A -I CPU" &> /home/df-h-mpstat-Application && sshpass -p "Call@Bkp18" scp -r /home/df-h-mpstat-Application root@192.168.1.8:/var/www/html/calls/server-utilization-reports/**

**last | head -25 &> /home/last-less-Application && sshpass -p "Call@Bkp18" scp -r /home/last-less-Application root@192.168.1.8:/var/www/html/calls/server-login-reports/**

**#!/bin/bash**

**SRC\_URI="http://192.168.1.8/calls/IT\_BAnner.jpg"**

**FNAME="/home/IT\_BAnner.jpg"**

**wget "$SRC\_URI" -O "$FNAME"**

**gsettings set org.gnome.desktop.background picture-uri "$FNAME"**

**Mounting Win Partation and transfer data to Windows machine vi ssh**

**#!/bin/sh**

**vi /home/daily-DB-Uplaoad.sh**

**[root@localhost ~]# vi /home/DB-upload-ws-115.sh**

**https://wiki.centos.org/TipsAndTricks/WindowsShares**

**#Daily DB Upload to HO DB Server#**

**scp -vr /var/www/html/DIALER/\* root@175.101.1.228:/DataBase-Backup/Hubli/MCTS/test/**

**/home/daily-DB-Uplaoad.sh**

**sshpass -p "zolt123$" scp -r /var/www/html/DIALER/\* root@175.101.1.228:/DataBase-Backup/Hubli/MCTS/**

**mount -t cifs //192.168.1.85/Hubli -o username=test,password=Networking@123$ /mnt/share**

**#!/bin/sh**

**#Daily DB Upload to WS-115 Hubli folder#**

**mount -t cifs '//192.168.1.85/Hubli/Daily DB Backup' -o username=test,password=Networking@123$ /mnt/share**

**cp /var/www/html/DIALER/\* /mnt/share/**

**umount -t cifs //192.168.1.85/Hubli/Daily DB Backup**

**############Daily DB Upload to WS-115 Hubli folder###############**

**00 1 \* \* \* /home/DB-upload-ws-115.sh**

**server 3.in.pool.ntp.org**

**server 1.asia.pool.ntp.org**

**server 3.asia.pool.ntp.org**

**#!/bin/sh**

**#Daily DB Upload to WS-115 Hubli folder#**

**Bkpdate=$(date +%Y-%m-%d)**

**mount -t cifs //192.168.1.85/Hubli/DB-Bkp-Test/104 -o username=test,password=Networking@123$ /mnt/share**

**cp /var/www/html/Daily\_Backup/"$Bkpdate"hmridata.sql.zip /mnt/share/**

**cp /var/www/html/Daily\_Backup/"$Bkpdate"convoxccs.sql.zip /mnt/share/**

**sshpass -p "zolt123$" Bkpdate=$(date +%Y-%m-%d) && scp -r /var/www/html/Daily\_Backup/"$Bkpdate"hmridata.sql.zip** [**root@175.101.1.228:/DataBase-Backup/Hubli/104/**](mailto:root@175.101.1.228:/DataBase-Backup/Hubli/104/)

**Script to extract login report in Server:**

**#!/bin/sh**

**last | less >> /home/last-less-Mcts-App.txt && sshpass -p "Call@Bkp18" scp -r /home/last-less-Mcts-App.txt root@192.168.1.8:/var/www/html/calls/server-login-reports/**

**#!/bin/sh**

**last | less >> /home/last-less-Mcts-Bkp.txt && sshpass -p "Call@Bkp18" scp -r /home/last-less-Mcts-Bkp.txt root@192.168.1.8:/var/www/html/calls/server-login-reports/**

**#!/bin/sh**

**last | less >> /home/last-less-Application.txt && sshpass -p "Call@Bkp18" scp -r /home/last-less-Application.txt root@192.168.1.8:/var/www/html/calls/server-login-reports/**

**#!/bin/sh**

**last | less >> /home/last-less-App-Bkp.txt && sshpass -p "Call@Bkp18" scp -r /home/last-less-App-Bkp.txt root@192.168.1.8:/var/www/html/calls/server-login-reports/**

**#!/bin/sh**

**last | less > /var/www/html/calls/server-login-reports/last-less-Backup.txt**

**##########Daily 1.55 Server Login Logout report##########**

**00 6 \* \* \* /home/last-less.sh**

**chmod u+x /home/last-less.sh && chmod 777 /home/last-less.sh**

**OS Upgrade commands in Ubuntu**

**lsb\_release -a**

**sudo apt-get dist-upgrade**

**sudo apt-get install update-manager-core**

**his process may take some time. Once it finishes, use the dist-upgrade command, which will perform upgrades involving changing dependencies, adding or removing new packages as necessary. This will handle a set of upgrades which may have been held back by apt-get upgrade:**

**sudo apt-get dist-upgrade**

**Again, answer y when prompted to continue, and wait for upgrades to finish.**

**Partation mounting issue**

**http://askubuntu.com/questions/840433/ubuntu-16-04-unable-to-mount-location**

**Changing UDP packet size in Centos:**

**smartctl -t long /dev/sdX**

**smartctl -a /dev/sdX | less**

**http://unix.stackexchange.com/questions/92076/how-can-i-check-for-bad-blocks-on-an-lvm-physical-volume**

**sysctl -w net.core.rmem\_max=26214400**

**On Linux, you can change the UDP buffer size (e.g. to 26214400) by (as root):**

**sysctl -w net.core.rmem\_max=26214400**

**The default buffer size on Linux is 131071.**

**You can also make it permanent by adding this line to /etc/sysctl.conf:**

**net.core.rmem\_max=26214400**

**Asterisk related commands:**

**2. Asterisk 1.4.0 CLI commands**

**Introduction**

**Asterisk CLI supports large variety of commands which can be used for testing, configuration and monitoring. In this tutorial we will describe all commands available at the standard Asterisk version 1.4.0. We will divide this tutorial into few sections in order to facilitate the reading.**

**General CLI commands**

**! - Execute a shell command**

**abort halt - Cancel a running halt**

**cdr status - Display the CDR status**

**feature show - Lists configured features**

**feature show channels - List status of feature channels**

**file convert - Convert audio file**

**group show channels - Display active channels with group(s)**

**help - Display help list, or specific help on a command**

**indication add - Add the given indication to the country**

**indication remove - Remove the given indication from the country**

**indication show - Display a list of all countries/indications**

**keys init - Initialize RSA key passcodes**

**keys show - Displays RSA key information**

**local show channels - List status of local channels**

**logger mute - Toggle logging output to a console**

**logger reload - Reopens the log files**

**logger rotate - Rotates and reopens the log files**

**logger show channels - List configured log channels**

**meetme - Execute a command on a conference or conferee**

**mixmonitor - Execute a MixMonitor command.**

**moh reload - Music On Hold**

**moh show classes - List MOH classes**

**moh show files - List MOH file-based classes**

**no debug channel (null)**

**originate - Originate a call**

**realtime load - Used to print out RealTime variables.**

**realtime update - Used to update RealTime variables.**

**restart gracefully - Restart Asterisk gracefully**

**restart now - Restart Asterisk immediately**

**restart when convenient - Restart Asterisk at empty call volume**

**sla show - Show status of Shared Line Appearances**

**soft hangup - Request a hangup on a given channel**

**stop gracefully - Gracefully shut down Asterisk**

**stop now - Shut down Asterisk immediately**

**stop when convenient - Shut down Asterisk at empty call volume**

**stun debug - Enable STUN debugging**

**stun debug off - Disable STUN debugging**

**udptl debug - Enable UDPTL debugging**

**udptl debug ip - Enable UDPTL debugging on IP**

**udptl debug off - Disable UDPTL debugging**

**AEL commands**

**ael debug contexts - Enable AEL contexts debug (does nothing)**

**ael debug macros - Enable AEL macros debug (does nothing)**

**ael debug read - Enable AEL read debug (does nothing)**

**ael debug tokens - Enable AEL tokens debug (does nothing)**

**ael nodebug - Disable AEL debug messages**

**ael reload - Reload AEL configuration**

**Agents commands**

**agent logoff - Sets an agent offline**

**agent show - Show status of agents**

**agent show online - Show all online agents**

**AGI commands**

**agi debug - Enable AGI debugging**

**agi debug off - Disable AGI debugging**

**agi dumphtml - Dumps a list of agi commands in html format**

**agi show- List AGI commands or specific help**

**dnsmgr reload - Reloads the DNS manager configuration**

**dnsmgr status - Display the DNS manager status**

**http show status - Display HTTP server status**

**Console commands**

**console active - Sets/displays active console**

**console answer - Answer an incoming console call**

**console autoanswer - Sets/displays autoanswer**

**console boost - Sets/displays mic boost in dB**

**console dial - Dial an extension on the console**

**console flash - Flash a call on the console**

**console hangup - Hangup a call on the console**

**console mute - Disable mic input**

**console send text - Send text to the remote device**

**console transfer - Transfer a call to a different extension**

**console unmute - Enable mic input**

**Core related commands**

**core clear profile - Clear profiling info**

**core set debug channel - Enable/disable debugging on a channel**

**core set debug - Set level of debug chattiness**

**core set debug off - Turns off debug chattiness**

**core set global - Set global dialplan variable**

**core set verbose - Set level of verboseness**

**core show applications - Shows registered dialplan applications**

**core show application - Describe a specific dialplan application**

**core show audio codecs - Displays a list of audio codecs**

**core show channels - Display information on channels**

**core show channel - Display information on a specific channel**

**core show channeltypes - List available channel types**

**core show channeltype - Give more details on that channel type**

**core show codecs - Displays a list of codecs**

**core show codec - Shows a specific codec**

**core show config mappings - Display config mappings (file names to config engines)**

**core show file formats - Displays file formats**

**core show file version - List versions of files used to build Asterisk**

**core show functions - Shows registered dialplan functions**

**core show function - Describe a specific dialplan function**

**core show globals - Show global dialplan variables**

**core show hints - Show dialplan hints**

**core show image codecs - Displays a list of image codecs**

**core show image formats - Displays image formats**

**core show license - Show the license(s) for this copy of Asterisk**

**core show profile - Display profiling info**

**core show switches - Show alternative switches**

**core show threads - Show running threads**

**core show translation - Display translation matrix**

**core show uptime - Show uptime information**

**core show version - Display version info**

**core show video codecs - Displays a list of video codecs**

**core show warranty - Show the warranty (if any) for this copy of Asterisk**

**Database commands**

**database del - Removes database key/value**

**database deltree - Removes database keytree/values**

**database get - Gets database value**

**database put - Adds/updates database value**

**database show - Shows database contents**

**database showkey - Shows database contents**

**Dialplan commands**

**dialplan add extension - Add new extension into context**

**dialplan add ignorepat - Add new ignore pattern**

**dialplan add include - Include context in other context**

**dialplan reload - Reload extensions and \*only\* extensions**

**dialplan remove extension - Remove a specified extension**

**dialplan remove ignorepat - Remove ignore pattern from context**

**dialplan remove include - Remove a specified include from context**

**dialplan save - Save dialplan**

**dialplan show - Show dialplan**

**DUNDI commands**

**dundi debug - Enable DUNDi debugging**

**dundi flush - Flush DUNDi cache**

**dundi lookup - Lookup a number in DUNDi**

**dundi no debug - Disable DUNDi debugging**

**dundi no store history - Disable DUNDi historic records**

**dundi precache - Precache a number in DUNDi**

**dundi query - Query a DUNDi EID**

**dundi show entityid - Display Global Entity ID**

**dundi show mappings - Show DUNDi mappings**

**dundi show peers - Show defined DUNDi peers**

**dundi show peer - Show info on a specific DUNDi peer**

**dundi show precache - Show DUNDi precache**

**dundi show requests - Show DUNDi requests**

**dundi show trans - Show active DUNDi transactions**

**dundi store history - Enable DUNDi historic records**

**GTalk & Jabber commands**

**gtalk reload - Enable Jabber debugging**

**gtalk show channels - Show GoogleTalk Channels**

**jabber debug - Enable Jabber debugging**

**jabber debug off - Disable Jabber debug**

**jabber reload - Enable Jabber debugging**

**jabber show connected - Show state of clients and components**

**jabber test - Shows roster, but is generally used for mog's debugging.**

**IAX2 commands**

**iax2 provision - Provision an IAX device**

**iax2 prune realtime - Prune a cached realtime lookup**

**iax2 reload - Reload IAX configuration**

**iax2 set debug - Enable IAX debugging**

**iax2 set debug jb - Enable IAX jitterbuffer debugging**

**iax2 set debug jb off - Disable IAX jitterbuffer debugging**

**iax2 set debug off - Disable IAX debugging**

**iax2 set debug trunk - Enable IAX trunk debugging**

**iax2 set debug trunk off - Disable IAX trunk debugging**

**iax2 show cache - Display IAX cached dialplan**

**iax2 show channels - List active IAX channels**

**iax2 show firmware - List available IAX firmwares**

**iax2 show netstats - List active IAX channel netstats**

**iax2 show peers - List defined IAX peers**

**iax2 show peer - Show details on specific IAX peer**

**iax2 show provisioning - Display iax provisioning**

**iax2 show registry - Display IAX registration status**

**iax2 show stats - Display IAX statistics**

**iax2 show threads - Display IAX helper thread info**

**iax2 show users - List defined IAX users**

**iax2 test losspct - Set IAX2 incoming frame loss percentage**

**Manager commands**

**manager show command - Show a manager interface command**

**manager show commands - List manager interface commands**

**manager show connected - List connected manager interface users**

**manager show eventq - List manager interface queued events**

**manager show users - List configured manager users**

**manager show user - Display information on a specific manager user**

**MGCP commands**

**mgcp audit endpoint - Audit specified MGCP endpoint**

**mgcp reload - Reload MGCP configuration**

**mgcp set debug - Enable MGCP debugging**

**mgcp set debug off - Disable MGCP debugging**

**mgcp show endpoints - List defined MGCP endpoints**

**Module management**

**module load - Load a module by name**

**module reload - Reload configuration**

**module show - List modules and info**

**module show like - List modules and info**

**module unload - Unload a module by name**

**PRI commands**

**pri debug span - Enables PRI debugging on a span**

**pri intense debug span - Enables REALLY INTENSE PRI debugging**

**pri no debug span - Disables PRI debugging on a span**

**pri set debug file - Sends PRI debug output to the specified file**

**pri show debug - Displays current PRI debug settings**

**pri show spans - Displays PRI Information**

**pri show span - Displays PRI Information**

**pri unset debug file - Ends PRI debug output to file**

**Queue commands**

**queue add member - Add a channel to a specified queue**

**queue remove member - Removes a channel from a specified queue**

**queue show - Show status of a specified queue**

**rtcp debug ip - Enable RTCP debugging on IP**

**rtcp debug - Enable RTCP debugging**

**rtcp debug off - Disable RTCP debugging**

**rtcp stats - Enable RTCP stats**

**rtcp stats off - Disable RTCP stats**

**rtp debug ip - Enable RTP debugging on IP**

**rtp debug - Enable RTP debugging**

**rtp debug off - Disable RTP debugging**

**say load - Set/show the say mode**

**show parkedcalls - Lists parked calls**

**show queue - Show information for target queue**

**show queues - Show the queues**

**SIP commands**

**sip history - Enable SIP history**

**sip history off - Disable SIP history**

**sip notify - Send a notify packet to a SIP peer**

**sip prune realtime - Prune cached Realtime object(s)**

**sip prune realtime peer - Prune cached Realtime peer(s)**

**sip prune realtime user - Prune cached Realtime user(s)**

**sip reload - Reload SIP configuration**

**sip set debug - Enable SIP debugging**

**sip set debug ip - Enable SIP debugging on IP**

**sip set debug off - Disable SIP debugging**

**sip set debug peer - Enable SIP debugging on Peername**

**sip show channels - List active SIP channels**

**sip show channel - Show detailed SIP channel info**

**sip show domains - List our local SIP domains.**

**sip show history - Show SIP dialog history**

**sip show inuse - List all inuse/limits**

**sip show objects - List all SIP object allocations**

**sip show peers - List defined SIP peers**

**sip show peer - Show details on specific SIP peer**

**sip show registry - List SIP registration status**

**sip show settings - Show SIP global settings**

**sip show subscriptions - List active SIP subscriptions**

**sip show users - List defined SIP users**

**sip show user - Show details on specific SIP user**

**Skinny commands**

**skinny reset - Reset Skinny device(s)**

**skinny set debug - Enable Skinny debugging**

**skinny set debug off - Disable Skinny debugging**

**skinny show devices - List defined Skinny devices**

**skinny show lines - List defined Skinny lines per device**

**Voicemail commands**

**voicemail show users - List defined voicemail boxes**

**voicemail show users for - List defined voicemail boxes for target context**

**voicemail show zones - List zone message formats**

**Zaptel commands**

**zap destroy channel - Destroys a channel**

**zap restart - Fully restart zaptel channels**

**zap show cadences - List cadences**

**zap show channels - Show active zapata channels**

**zap show channel - Show information on a channel**

**zap show status - Show all Zaptel cards status**

**MySQL repair command:**

pkill -9 asterisk

asterisk –vr

safe\_asterisk

service dahdi restart

lspci

wanrouter restart

mysqlcheck --repair --databases convoxccs

wanrouter status