ACLs

Saturday, March 3, 2018 9:15 AM

Packages Required	Acl
Config Files	
Port Numbers	
Service Name	

Task: Installing ACL

install ACL
#yum install -y acl

create file/folder for acl access
#touch /home/aclAclTest.txt

give proper permissions to the file/folder.
#chmod 777 /home/aclAclTest.txt

setting ACL settings for file "aclAclTest.txt"
#setfacl -m u:<user_name>:r </home/aclAclTest.txt>
m = modify
u = username
r = read access
user_name = u1
#ls -l //'+' is added in the end of the permissions

verifying permissions
#getfacl /home/aclAclTest.txt

accessing through user u1
[u1@systemName] #cat /home/aclAclTest.txt

accessing through OTHER user
[demo@systemName] #cat /home/aclAclTest.txt
ERROR

==========

Task: in a group called avengers, where multiple users & amongst multiple users, user THOR shouldn't have write permissions but only read permission.

#getfacl /top.txt

#setfacl -m “g:avengers:rwx” /top.txt //will allow all users of avengers to r/w/x file

#setfacl -m “u:thor:r” /top.txt //this will allow thor to only read file, being in avengers group.

#setfacl -b /top.txt //remove all entries

==========

apply acl for /etc/passwd file

#setfacl -m u:user1:rw /etc/passwd

checking the acl is applied or not

#getfacl /etc/passwd

applying acl on the group

create users & add them into a group

#setfacl -m g:hr:rw /etc/passwd

==========

There are two types of ACLs:

1. Access ACLs: Access ACLs are used for granting permissions on any file or directory.
2. Default ACLs: Default ACLs are used for granting/setting access control list on a specific directory only.

Difference between Access ACL and Default ACL:

1. Default ACL can be used on directory level only.
2. Any sub directory or file created within that directory will inherit the ACLs from its parent directory. On the other hand a file inherits the default ACLs as its access ACLs.
3. We make use of "–d" for setting default ACLs and Default ACLs are optional.

https://www.google.com/amp/s/www.tecmint.com/secure-files-using-acls-in-linux/amp/

Anacron

Tuesday, March 19, 2019 2:26 PM

Packages Required	
Config Files	/etc/anacrontab
Port Numbers	
Service	
Commands	ls -l /var/spool/anacron/
Firewall rules	

- Anacron is used to run commands periodically with a frequency defined in days. It works a little different from cron; assumes that a machine will not be powered on all the time.
- It is appropriate for running daily, weekly, and monthly scheduled jobs normally run by cron, on machines that will not run 24-7 such as laptops and desktops machines.
- However, if you use anacron, you can be assured that the next time you power on the desktop/laptop again, the backup script will be executed.

How Anacron Works in Linux:

- anacron jobs are listed in /etc/anacrontab and jobs can be scheduled using the format below

```
period   delay   job-identifier   command
```

- From the above format:
  - **period** – this is the frequency of job execution specified in days or as @daily, @weekly, or @monthly for once per day, week, or month. You can as well use numbers: 1 – daily, 7 – weekly, 30 – monthly and N – number of days.
  - **delay** – it's the number of minutes to wait before executing a job.
  - **job-id** – it's the distinctive name for the job written in log files.
  - **Command** – it's the command or shell script to be executed

This is what practically happens:

- Anacron will check if a job has been executed within the specified period in the period field. If not, it executes the command specified in the command field after waiting the number of minutes specified in the delay field.
- Once the job has been executed, it records the date in a timestamp file in the /var/spool/anacron directory with the name specified in the job-id (timestamp file name) field.

Cron	Anacron
It's a daemon	It's not a daemon
Appropriate for server machines	Appropriate for desktop/laptop machines
Enables you to run scheduled jobs every minute	Only enables you to run scheduled jobs on daily basis
Doesn't executed a scheduled job when the machine if off	If the machine if off when a scheduled job is due, it will execute a scheduled job when the machine is powered on the next time
Can be used by both normal users and root	Can only be used by root unless otherwise (enabled for normal users with specific configs)

Ansible

Friday, February 8, 2019 2:46 PM

https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-ansible-on-centos-7

https://devopsmates.com/ansible-installation-and-configuration-on-redhatcentos-7/

Apache web server installation

Saturday, March 3, 2018 9:16 AM

Packages Required	Httpd (apache web server), mod_ssl (for secure websites), elinks (text based browser)
Config Files	/etc/httpd/conf/httpd.conf
Port Numbers	TCP ports 80, 443, 488, 8008, 8009, 8443
Service Name	Httpd
Commands	yum install –y httpd mod_ssl elinks; yum groupinstall "Web Server"
Firewall rules	#iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT #iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT # service iptables save # service iptables restart

Link

1. set disable/permissive firewall
2. modify firewall
   firewall-cmd --add-server http
   firewall-cmd --add-server https

0r turn off firewall

3. "#yum install -y httpd" OR yum group install *Basic Web Server* -y
4. start httpd service
   systemctl enable httpd.service
   systemctl start httpd.service
   systemctl status httpd.service
5. check the log files
   # ll /var/log/httpd
6. check configuration files
   #ll /etc/httpd/conf/
7. testing server, create an index.html file in /var/www/html/www.jeetu.com/index.html

#vim /var/www/html/index.html
hello world

thanks
Jeetu
---
8. open firefox

type -> localhost
the data will be displayed that u have written in index.html file

==========

Apache:

#yum install -y httpd

Check /etc/httpd/conf/httpd.conf file

create an index file at /var/www/html/<index.html>

==========

Apache webserver

#ifconfig
#yum list all
#yum install -y httpd*
#vim /etc/httpd/conf/httpd.conf
SHIFT+g-> go to last
press i
<VirtualHost server-ip>
ServerAdmin root@mylinux
DocumentRoot /var/www/html
DirectoryIndex index.html
</VirtualHost>
:wq!
#cd /var/www/html/
#httpd -t
#vim index.html
i
Hi Jeetu Here
:wq!
#service httpd restart

GOTO web browser & type ur system IP
Hi Jeetu Here

#cd /var/www/html
#mkdir share
#vim /etc/httpd/conf/httpd.conf
IN THE END, above </VirtualHost> add the below
<Directory "/var/www/html/share">
Options Indexes
Order Allow,Deny
Allow from all
</directory>
</VirtualHost>
#service httpd restart
#cd share
#touch test{1,2,3,4,5,6}.txt

GOTO web browser & type ur system IP
http://192.168.0.100/share

Adding security
#vim /etc/httpd/conf/httpd.conf
in the last above '</directory>' add 1 line
AllowOverride Authconfig
:wq!
#vi .htaccess
AuthName "secure file"
AuthType basic
AuthUserFile /etc/httpd/htpass
Require valid-user
</directory>
:wq!

create a user for this
#useradd authUser
#passwd authUser

#htpasswd -c /etc/httpd/htpadd authUser
#xxx
#xxx

#service httpd restart

goto client & refresh the page
UserName: authUser
PWD: xxx

==========

Apache - 1. basic localhost site

04 April 2020 12:51 AM

#create default site

---

install httpd
enable
start
status

vim /var/www/html/index.html
//write HTML code
//save & quit

restart httpd
status httpd

web browser -> localhost (refresh)

Apache - 2. multiple sites

04 April 2020 12:51 AM

#create multiple site in apache
---
create 2 folders at
1. /var/www/site1
   - create index.html file with some code
2. /var/www/site2
   - create index.html file with some code

edit vim /etc/httpd/conf/httpd.conf file with below code:

```
<VirtualHost 127.0.0.1:80>
  DocumentRoot /var/www/site1
  ServerName www.site1.org
</VirtualHost>
<VirtualHost 127.0.0.1:80>
    DocumentRoot /var/www/site2
    ServerName www.site2.org
</VirtualHost>
```

Make site1, site2 entries in /etc/hosts file

```
127.0.0.1    www.site1.org
127.0.0.1    www.site2.org
```

restart httpd service

Apache - 3. HTTPS/SSL site

04 April 2020 12:52 AM

#create HTTPS/SSL based site.

---

1. **create a new site.**

    #mkdir /var/www/www.jeetusingh.in

2. **configure site as per ur requirements**

    #cp /var/www/html/index.html /var/www/www.jeetusingh.in/

    #vim /var/www/www.jeetusingh.in/index.html

3. **make entry in /etc/hosts file**

    192.168.10.12 www.jeetusingh.in

4. **make site entry in httpd config file**

    #vim /etc/httpd/conf/httpd.conf

        <VirtualHost 192.168.10.12:80>

            DocumentRoot /var/www/www.jeetusingh.in

            ServerName www.jeetusingh.in

        </VirtualHost>

5. restart & verify if site is working.

5.b **Install SSL package**

    #yum install -y mod_ssl openssh*

6. **create SSL certificate for site**

    #cd /etc/pki/tls/certs/

    #ls

7. **Make ".key" file**

    #make jeetusingh

        //with or without passphrase

    #ls

8. **create CSR file**

    #openssl rsa -in jeetusingh.key -out jeetusingh.key (press enter)

    output:

        Enter pass phrase for jeetusingh.key:

        writing RSA key

    #make jeetusingh.csr (press enter)

    output:

        Country Name (2 letter code) [XX]:IN

        State or Province Name (full name) []:KA

        Locality Name (eg, city) [Default City]:BLR

        Organization Name (eg, company) [Default Company Ltd]:Alpha.corp

        Organizational Unit Name (eg, section) []:Trainer

        Common Name (eg, your name or your server's hostname) []:jeetusingh.in

        Email Address []:root@jeetusingh.in

    #ls

        output:

            jeetusingh.csr

9. **generate certificate**:
#openssl x509 -in jeetusingh.csr -out jeetusingh.crt -req -signkey jeetusingh.key -days 365
---
output:
Signature ok
subject=/C=IN/ST=KA/L=BLR/O=Alpha.corp/OU=Trainer/CN=jeetusingh.in/emailAddress=root@jeetusingh.in
Getting Private key

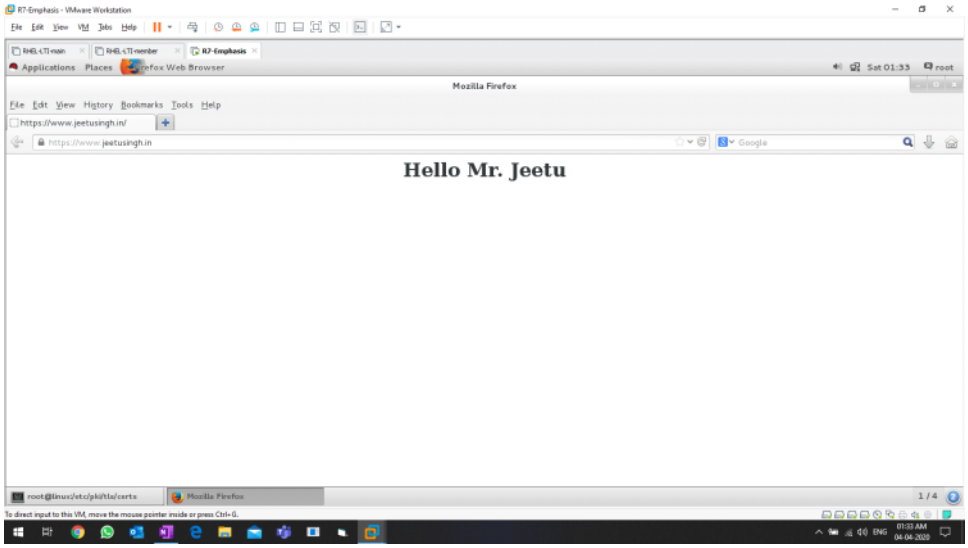
10. **config ssl.conf file**
#vim /etc/httpd/conf.d/ssl.conf
- uncomment line 59:
  DocumentRoot "/var/www/html"
- uncomment line 60 & write site name with 443:
  ServerName www.jeetusingh.in:443
- online 100, add .crt certificate
  SSLCertificateFile /etc/pki/tls/certs/jeetusingh.crt
- add .key file on 108 line
  SSLCertificateKeyFile /etc/pki/tls/certs/jeetusingh.key

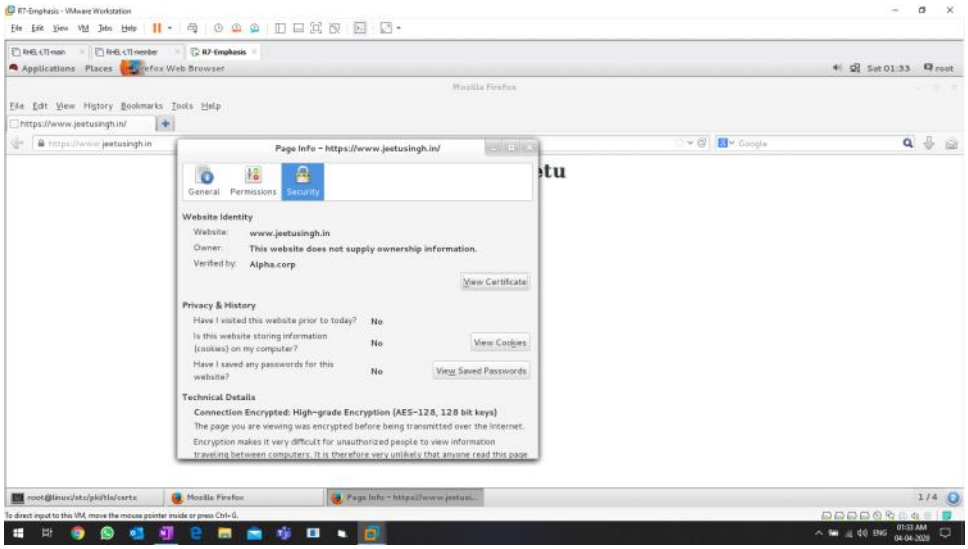
:wq!

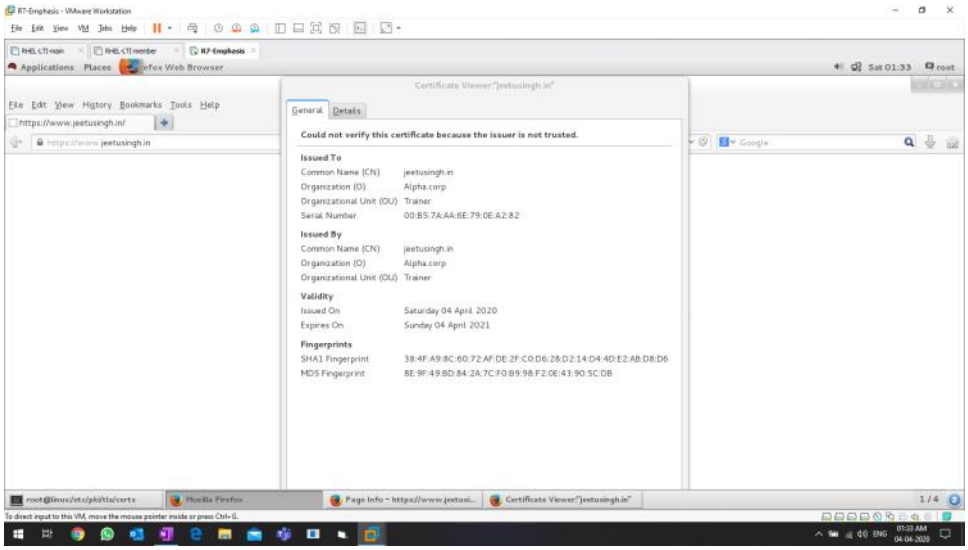
11. **restart httpd service**
#systemctl enable httpd
#systemctl start httpd
#systemctl restart httpd
#systemctl status httpd

12. **open web browser**
https://www.jeetusingh.in

Screen shots:




Page Info – https://www.jeetusingh.in/
General
Permissions
Security
Website Identity
Website: www.jeetusingh.in
Owner: This website does not supply ownership information.
Verified by: Alpha.corp
View Certificate
Privacy & History
Have I visited this website prior to today? No
Is this website storing information (cookies) on my computer? No
View Cookies
Have I saved any passwords for this website? No
View Saved Passwords
Technical Details
Connection Encrypted: High-grade Encryption (AES-128, 128 bit keys)
The page you are viewing was encrypted before being transmitted over the Internet.
Encryption makes it very difficult for unauthorized people to view information traveling between computers. It is therefore very unlikely that anyone read this page


Certificate Viewer:"jeetusingh.in"
General
Details
Could not verify this certificate because the issuer is not trusted.
Issued To
Common Name (CN) jeetusingh.in
Organization (O) Alpha.corp
Organizational Unit (OU) Trainer
Serial Number 00:B5:7A:AA:6E:79:0E:A2:82
Issued By
Common Name (CN) jeetusingh.in
Organization (O) Alpha.corp
Organizational Unit (OU) Trainer
Validity
Issued On Saturday 04 April 2020
Expires On Sunday 04 April 2021
Fingerprints
SHA1 Fingerprint 38:4F:A9:8C:60:72:AF:DE:2F:C0:D6:2B:D2:14:D4:4D:E2:AB:D8:D6
MD5 Fingerprint 8E:9F:49:BD:84:2A:7C:F0:B9:98:F2:0E:43:90:5C:0B

DNS - R7 - working (**use it**)

05 April 2020 03:17 PM

- Domain name service
- Uses port 53
- For the configuration, static IP is "must".
- Requires two systems
  - Svr - 10.10.10.10/8
  - Mem1 - 10.10.10.11/8
- Domain name used: **alpha.corp**
- Required files
  - Main config file - **/etc/named.conf**
  - Forward lookup zone file - **/var/names/forward.alpha.corp** --> *need to be created*
  - Forward lookup zone file - **/var/names/reverse.alpha.corp** --> *need to be created*

Install DNS packages	#yum install -y bind bind-utils
Edit configuration file	#vim /etc/named.conf line 11, add your DNS server IP after 127.0.0.0; 10.10.10.10; Line 17, allow-query {localhost, any;} Save & quit
Enable, start & status 'named' service	#systemctl enable named #systemctl start named #systemctl status named

Create zones in the /etc/named.conf file	At the end of the file, append below code.
	
	zone "alpha.corp" IN {
	type master;
	file "forward.alpha.corp";
	allow-update {none;};
	};
	
	zone "10.10.10.in-addr-apra" IN {
	type master;
	file "reverse.alpha.corp";
	allow-update {none;};
	};
	
	Save & quit

Switch to /var/named directory	#cd /var/named //look for 'named.localhost' file create a copy of it as 'forward.alpha.corp' # cp named.localhost forward.alpha.corp Edit # vim forward.alpha.corp TTL 1D @ IN SOA @ svr.alpha.corp. (0 ; serial 1D ; refresh 1H ; retry 1W ; expire 3H) ; minimum @ IN NS svr.alpha.corp. @ IN A 10.10.10.10 svr IN A 10.10.10.10 mem1 IN A 10.10.10.11 Save & quit
Create a copy of forward.alpha.corp as reverse.alpha.corp	#cp forward.alpha.corp reverse.alpha.corp
Edit reverse.alpha.corp file	#vim reverse.alpha.corp file

	$TTL 1D @ IN SOA @ svr.alpha.corp. (0 ; serial 1D ; refresh 1H ; retry 1W ; expire 3H) ; minimum @ IN NS svr.alpha.corp. @ IN PTR alpha.corp. @ IN A 10.10.10.10 svr IN A 10.10.10.10 mem1 IN A 10.10.10.11 10 IN PTR svr.alpha.corp. 11 IN PTR mem1.alpha.corp. Save & quit
Change the OWNERSHIP of both files, else it won't work.	#chown root:named forward.alpha.corp #chown root:named reverse.alpha.corp
Check the DNS config file:	#named-checkconf -z /etc/named.conf & look for "loaded serial 0", means all OK, else rectify /etc/named.conf file.
Check the zone config files	[root@svr named]# named-checkzone forward /var/named/forward.alpha.corp *zone forward/IN: loaded serial 0* *OK* [root@svr named]# named-checkzone reverse /var/named/reverse.alpha.corp *zone reverse/IN: loaded serial 0* *OK*
If all OK, restart & status named service	# systemctl restart named # systemctl status named
Remove manual entries from /etc/hosts, if any	
On svr.alpha.corp, make changes in	#vim /etc/resolv.conf search alpha.corp nameserver 10.10.10.10
Ping svr with name.	#ping svr.alpha.local

ON CLIENT SYSTEM:

Open resolv.conf & make only below entry, rest remove it.	#vim /etc/resolv.conf search alpha.corp
Verifying all good	#dig alpha.corp

```
[root@mem1 Desktop]# dig alpha.corp

; <<>> DiG 9.9.4-RedHat-9.9.4-14.el7 <<>> alpha.corp
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 16510
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;alpha.corp.                    IN      A

;; ANSWER SECTION:
alpha.corp.             86400   IN      A       10.10.10.10

;; AUTHORITY SECTION:
alpha.corp.             86400   IN      NS      svr.alpha.corp.

;; ADDITIONAL SECTION:
svr.alpha.corp.         86400   IN      A       10.10.10.10

;; Query time: 0 msec
;; SERVER: 10.10.10.10#53(10.10.10.10)
;; WHEN: Sun Apr 05 16:21:42 IST 2020
;; MSG SIZE  rcvd: 89

[root@mem1 Desktop]#
```

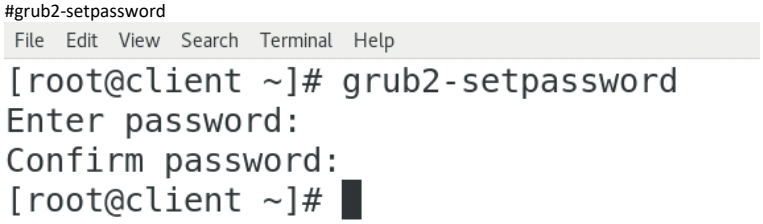
Password protecting GRUB2

28 November 2020 02:31 PM

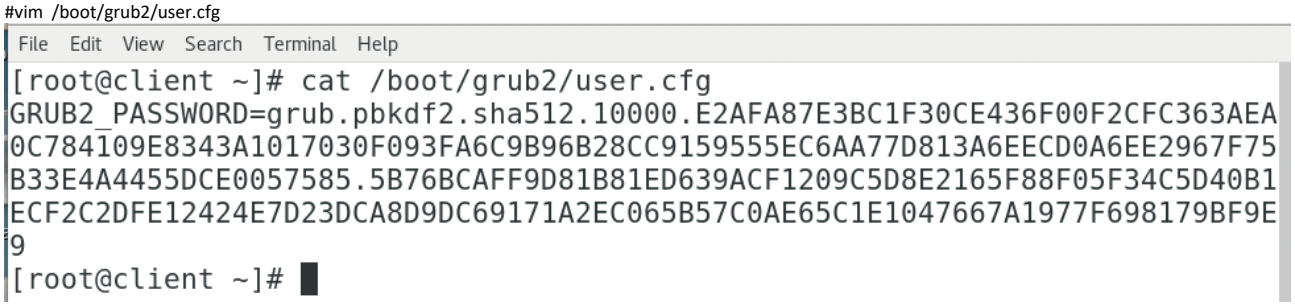
1. Remove "-unrestricted" from the "CLASS" declaration in the "/etc/grub.d/10_linux"

   Note: Keep it (line 29) as it is.

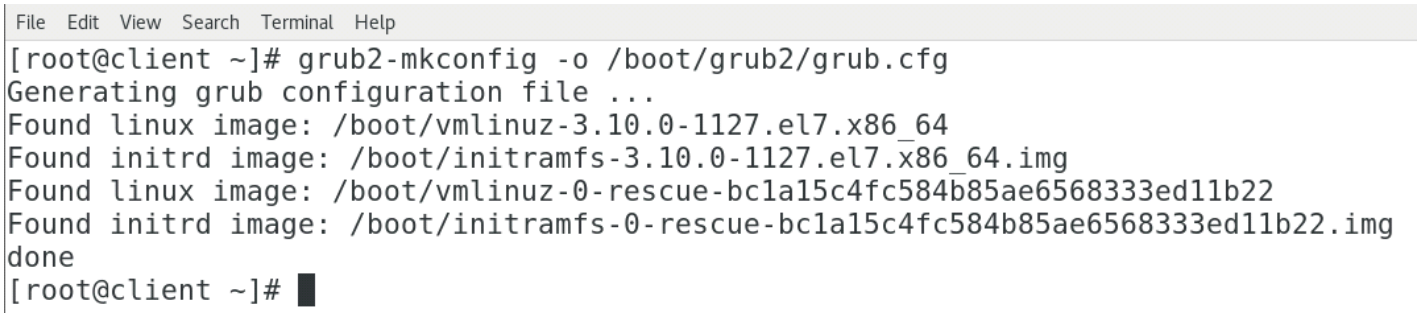
2. Set the password for the root user.
   #grub2-setpassword



3. This creates the below file with encrypted pwd in it.
   #vim /boot/grub2/user.cfg

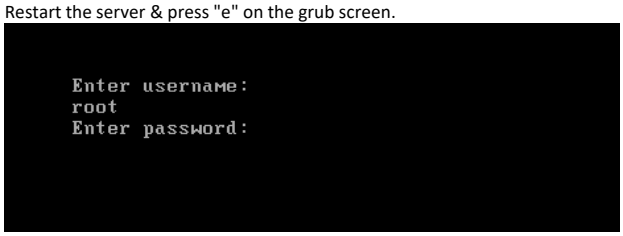


4. Recreate the grub configuration file



   "-o" means output into a file.

5. Restart the server & press "e" on the grub screen.



To remove the password from the GRUB screen.

1. Remove the "/boot/grub2/user.cfg" file OR just RENAME it (tested, working), then restart & press "e" in GRUB.
2. ADD "-unrestricted" from the "CLASS" declaration in the "/etc/grub.d/10_linux"

Mail server - R7 (verified)

05 April 2020 10:54 PM

1. Install POSTFIX server. -> for sending mails
2. Install DOVECOT server. -> for receiving mails.
3. Install squirrel server. -> for accessing

Mail server = used for sending & receiving mails.

Mail server = svr.alpha.corp
Domain = alpha.corp
IP = 10.10.10.10/8

Mail server
- Postfix
  - Uses SMTP
  - For sending mails
- Dovecot
  - Uses POP3/IMAP
  - For receiving mails

Requirement: DNS installed & resolving IP <-> NAME.

- In the existing DNS forward lookup zone, add MX record in order to work with mail server.

ON DNS server:

#vim /var/named/forward.corp

Note: add following content below NS line.

Svr	IN	MX	10	svr.alpha.corp

:wq!
#systemctl restart named
#systemctl status named

POSTFIX CONFIGURATION START:

Install/check post fix

#rpm -q postfix

Configure postfix mail.cf file

#vim /etc/postfix/main.cf

Line 75:	myhostname = svr.alpha.corp
Line 84:	mydomain = alpha.corp
Line 101:	myorigin = $mydomain
Line 116:	inet_interface = all (comment rest inet_* lines)
Line 168:	mydestination = <at the extreme right>, $mydomain
Line 268:	mynetwork = 10.0.0.0/8, 127...
Line 424:	home_mailbox = Maildir/

:wq!

#systemctl enable postfix

#systemctl start postfix
#systemctl status postfix

TO VERIFY POSTFIX IS RUNNING/WORKING:

#telnet localhost smtp
ehlo localhost (hit-enter)
..
mail from:<thor> (write as it is with <> & hit-enter)
..ok
rcpt to:<loki> (to whom you want to send mail)
..ok
data (hit-enter)
Hello Loki,
//write something.
//something - something.
.
quit

A mail file has been created in the loki's home directory:
#cd /home/loki/MailDir/new (hit-enter)
#ls (hit-enter)
#cat <press-tab> (hit-enter)
You can see the mail you had sent earlier. Means its working.

DOVECOT CONFIGURATION START:

Install/check dovecot package:
#yum install -y dovecot

Configure dovecot config file:
#vim /etc/dovecot/dovecot.conf (hit-enter)

Line 24:	Uncomment it. protocol = imap pop3 lmtp

:wq!

Configuring other supportive files:
#vim /etc/dovecot/conf.d/10-mail.conf (hit-enter)
Line 24:
mail_location = maildir :~/Maildir
:wq!

#vim /etc/dovecot/conf.d/10-auth.conf (hit-enter)
Line 10: uncomment the line & change YES -> NO
Disable_plaintext_auth = no
Line 101:
auth_mechanisms = plain login
:wq!

#vim /etc/dovecot/conf.d/10-master.conf (hit-enter)
Line 89: uncomment & add.
user = postfix
group = postfix
:wq!

Systemctl enable dovecot

Systemctl start dovecot
Systemctl status dovecot

TO VERIFY DOVECOT IS RUNNING/WORKING:

#telnet localhost pop3

..
user loki (hit-enter)
..
pass loki (hit-enter)
..
list (hit-enter)
+OK 1 messages.
retr 1(hit-enter)
..
..
..
. (hit-enter)
quit (hit-enter)

Hence, dovecot & postfix working properly.

TO ACCESS MAILS (IN GUI) - Out of TOC (LTI)

Download squirrel web-mail , extract & rename it to '/webmail'
install squirrel web-mail
install php*

##########Squirrel mail repo##########

SquirrelMail package is not available in the official CentOS 7 repositories. hence we will have to enable EPEL repository with the following command

#yum -y install epel-release

Install SquirrelMail on CentOS 7

Now, the SquirrelMail installation is pretty simple and it can be installed through CentOS package manager

#yum -y install squirrelmail php

##########

Move it to

mv /webmail /usr/share
cd /usr/share/webmail/config
#ls
#chmod +x conf.pl
command >> 1
[squirrel mail] Alpha Corp
>> 4
]: Alpha Admins,
>> 8
]: BLR
>>S --> save
>>R --> Return

Command >> 2
>> 1
[Example.com]: alpha.corp
>> A
IMAP >> 4
[localhost]: svr.alpha.corp
Smtp]: >> B
>>4
]: svr.alpha.corp
>> S
>>R
>>Q

#yum install http php -y
#vim /etc/httpd/conf/httpd.conf
Append at bottom.

Alias /webmail /usr/share/squirrel
<Directory /usr/share/squirrel>
Options Indexes FollowSymLinks
RewriteEngine on
AllowOverride All
DirectoryIndex index.php
Order allow,deny
Allow from all
</Directory>
:wq!

#systemctl enable httpd
#systemctl start httpd
#systemctl status httpd

Create users to access the content:
#useradd user1
#useradd user2
#passwd user1
#passwd user2

Access mail over
http://<IP-Address>/webmail

Apache with SSL/TLS

Monday, December 9, 2019 3:36 PM

Steps:

1. Install httpd, mod_ssl packages.
2. Create SSL/Self sign certificate
3. Configure required configuration files.

1. Create multiple site with required configuration:
   a. #yum install -y httpd mod_ssl
2. Create 2 directories in /var/www/html/ location.
   a. #mkdir www.new1.com & create a dummy page inside it.
   b. #mkdir www.new2.com & create a dummy page inside it.
3. Configure /etc/httpd/conf/httpd.conf file & append below lines:

```
<VirtualHost *:80>
  DocumentRoot   /var/www/html/<www.new1.com -Folder>
  ServerName     www.new1.com
</VirtualHost>
<VirtualHost *:80>
  DocumentRoot   /var/www/html/<www.new2.com -Folder>
  ServerName     www.new2.com
</VirtualHost>
:wq!
```

4. **To redirect to HTTPS** (Always on SSL/TLS):
   [URL: https://www.server-world.info/en/note?os=CentOS_7&p=httpd&f=7]

   # vi /etc/httpd/conf.d/vhost.conf

```
<VirtualHost *:80>
  DocumentRoot /var/www/html
  ServerName www.srv.world
  RewriteEngine On
  RewriteCond %{HTTPS} off
  RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
</VirtualHost>
```

5. #systemctl stop httpd
6. #systemctl start httpd
7. #systemctl restart httpd
8. Clear the cache memory & verify.

X11

Tuesday, December 10, 2019 5:30 PM

https://developer.ibm.com/tutorials/l-lpic1-106-1/

Zombie Process creation - verified

Monday, December 9, 2019 10:26 PM

What is a Zombie Process?

- A zombie or a defunct process in Linux is a process that has been completed, but its entry still remains in the process table due to lack of correspondence between the parent and child processes.
- Usually, a parent process keeps a check on the status of its child processes through the wait() function.
- When the child process has finished, the wait function signals the parent to completely exit the process from the memory. However, if the parent fails to call the wait function for any of its children, the child process remains alive in the system as a dead or zombie process.
- These zombie processes might accumulate, in large numbers, on your system and affect its performance.

Creating a Zombie-Process

```
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
int main ()
{
pid_t child_pid;child_pid = fork ();
if (child_pid > 0) {
sleep (60);
}
else {
exit (0);
}
return 0;
}
```

Save this file as zombie.c

Open the Terminal and run the following command to compile this program:
#cc zombie.c -o zombie

Now run the zombie program through the following command:
#./zombie

```
[root@client Desktop]# cat zombie.c
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
int main ()
{
pid_t child_pid;child_pid = fork ();
if (child_pid > 0) {
sleep (60);
}
else {
exit (0);
}
return 0;
}
[root@client Desktop]# cc zombie.c -o zombie
[root@client Desktop]# ./zombie
```

```
top - 22:49:25 up  7:56,  3 users,  load average: 0.05, 0.03, 0.05
Tasks: 260 total,   1 running, 258 sleeping,   0 stopped,   1 zombie
%Cpu(s):  0.7 us,  0.7 sy,  0.0 ni, 98.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  3865552 total,  1927160 free,   869472 used,  1068920 buff/cache
KiB Swap:  5242876 total,  5242876 free,        0 used.  2607280 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
22663 root      20   0  556304  20964  12852 S   1.0  0.5   0:07.23 gnome-terminal-
12049 root      20   0 1692912 149680  39680 S   0.7  3.9   1:52.50 gnome-shell
  795 dbus      20   0  101984   3504   1460 S   0.3  0.1   0:04.07 dbus-daemon
 6391 root      20   0  260620   1476   1080 S   0.3  0.0   0:30.08 pcscd
23143 root      20   0       0      0      0 S   0.3  0.0   0:00.04 kworker/0:0
    1 root      20   0  193628   6776   4000 S   0.0  0.2   0:07.94 systemd
    2 root      20   0       0      0      0 S   0.0  0.0   0:00.06 kthreadd
    3 root      20   0       0      0      0 S   0.0  0.0   0:00.09 ksoftirqd/0
    7 root      rt   0       0      0      0 S   0.0  0.0   0:00.95 migration/0
    8 root      20   0       0      0      0 S   0.0  0.0   0:00.00 rcu_bh
    9 root      20   0       0      0      0 S   0.0  0.0   0:08.20 rcu_sched
   10 root      rt   0       0      0      0 S   0.0  0.0   0:04.18 watchdog/0
   11 root      rt   0       0      0      0 S   0.0  0.0   0:00.42 watchdog/1
   12 root      rt   0       0      0      0 S   0.0  0.0   0:01.13 migration/1
   13 root      20   0       0      0      0 S   0.0  0.0   0:00.81 ksoftirqd/1
   17 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 khelper
   18 root      20   0       0      0      0 S   0.0  0.0   0:00.00 kdevtmpfs
   19 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 netns
   20 root      20   0       0      0      0 S   0.0  0.0   0:00.01 khungtaskd
   21 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 writeback
   22 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kintegrityd
   23 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 bioset
   24 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kblockd
   25 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 md
   31 root      20   0       0      0      0 S   0.0  0.0   0:00.00 kswapd0
```

APT

Friday, June 7, 2019 11:23 AM

- Advance Packaging Tool
- For Debian:
    - Ubuntu
    - Linux Mint

- **Search for a string:**
  #apt-cache <string>

- **To install packages:**
  #apt-get install -y <package-name>

- **To remove a package(leaving configuration):**
  #apt-get remove <package-name>

- **To remove a package (removing config too):**
  #apt-get purge <package-name>

- **Display information about package:**
  #apt-cache show <package-name>

Basic Linux

Saturday, March 3, 2018 9:22 AM

shell = shell prompt id '#' or '$'
kernal = operates device driver

Operating system = Kernal + shell (always)

user -> shell(BASH) -> O.S -> kernal(vmlinuz) -> hardware

(user lang) (machine lang) gen error lang

GNU = Gnome Not Unix
GNOME = Gnu Network Object Model Environment
Thus they call it "GNU Linux"

development purpose = fedora, suse (as dont have to register)

desktop environment

default desktop in linux = GNOME (default), KDE (K desktop env)
popular desktop = Xfce, E17 on internet

wine = WINdows Emulator (to run win apps)

cedega = for installing games on linux (with WINE)

LAMP (in RHEL7) =
L = linux
A = Apache (web server)
M = Mysql/MariaDB (database)
P = PHP/Python/Perl

server = 10.128.1.185

partitions
---
1. primary
2. extended

Max 4 partitions on a drive

[P P P P]

[P P P E {L1,L2}]

1st sector on HDD = MBR (Master Boot Record)
1 sector = 512 bytes (size cannot be changed)

446 bytes = to store Boot loader (GRUB2) {GRand Unified Bootloader- select O.S to boot, when

more than 2 are present}
64 bytes = partition table = (16x4 [number of partition])
2 bytes = magic bytes (it passes, only it boots)

HOW TO CREATE MORE THAN 4 PARTITIONS ON HDD

/dev/sda = Scsi Disk A

/boot = linux kernal, grub, boot loader {always keep in diff. partition} = 1GB
= Primary Partition
/ = linux operating system (commands, configuration files) = 25GB
= Primary Partition
swap = extention memory given to the RAM from HDD = twice than RAM
= Primary Partition
/home = automatically system will create Extended partition = 1GB
= Extended partition
holds users home directories

Xfsdump backup

Tuesday, December 25, 2018 8:28 PM

- There are several tools available for backup of Linux system. Best known amongst admins is RSYNC.
- It efficiently copies and sync files to or from a remote system.
- Faster than SCP (secure Copy)
- It consumes less bandwidth
- Basic syntax of rsync command
  - # rsync options source destination
- Installing rsync
  - #yum install -y rsync
- Copy locally
  - #rsync -zvh <archive.tar> /location-to-store-backup //z=compress, v=verbose, h=human readable
- Incremental backup using rsync
  - #rsync -avzh <archive.tar> /location-to-store-backup //a=archive,
- Taking backup on remote server
  - #rsync -avz <local-archive> <root@remote-ip:/remote-location>
- Taking backup from remote server
  - #rsync -avz <root@remote-ip:/remote-location> <local-archive>

https://www.tecmint.com/rsync-local-remote-file-synchronization-commands/

==========
BACKUP USING DUMP (link) (Not for XFS file systems)
==========

- The dump command is the most commonly used tool for performing backups on UNIX systems.
- The dump package is included in Red Hat Linux. If it was not installed by default when you first set up your Linux system, you can install it from the dump RPM file located on Red Hat Linux installation CD.

Command	Description
dump	Creates backup archives of whole disk partitions or selected directories.
restore	Can be used to restore an entire archive or individual files from an archive to the hard drive.
rmt	A program used by the dump and restore commands to copy files across the network. You should never need to use this command directly.

- Creating a backup with dump (not for XFS file systems)
  - # dump <options> <arguments> <filesystem>

To create a backup of XFS file system (link):(video)

- #xfsdump -f <where-to-backup> <what-to-backup>
- #xfsdump -l 0 -f /var/<dir-name> /<source-backup-dir-name>
  - Ex: #xfsdump -l 0 -f /var/backup/b-name /boot
    - Level name: full
    - Media name: file

Working /etc backup(link):	• mkdir d1 • cd d1/ • ls • xfsdump -0f myetcdump /dev/mapper/rhel-root -s etc • history

For selective restore	1. This exercise enables you to navigate through the backup file “mybackup”, and allows you to restore selected files instead of perfuming a full restore: # cd /tmp

	# xfsrestore –if mybackup /tmp 2. Once you get the restore prompt, you may ls and cd commands to navigate and list the files inside the backup file. 3. Select the files you want to restore by using the add command: # add yum.conf //similarly 'add' names of files & directories 4. Once the selection is complete, write extract to restore the selected files.

To take incremental backup for the same

- #xfsdump -l 0 -f <where-to-backup> <what-to-backup>

To view the backup

- #xfsdump -I // I = capital 'i'

To restore the backup

- #xfsrestore -f <what-to-restore-'xfsdump backup name'> <where-to-backup>

Step 1:

```
[root@client d1]# xfsrestore -if myetcdump /root/Desktop/restore/
xfsrestore: using file dump (drive_simple) strategy
xfsrestore: version 3.1.3 (dump format 3.0) - type ^C for status and control
xfsrestore: searching media for dump
xfsrestore: examining media file 0
xfsrestore: dump description:
xfsrestore: hostname: client
xfsrestore: mount point: /
xfsrestore: volume: /dev/mapper/rhel-root
xfsrestore: session time: Sat Jan 12 23:53:58 2019
xfsrestore: level: 0
xfsrestore: session label: "full-etc"
xfsrestore: media label: "file"
xfsrestore: file system id: 2a6ded55-7e2b-4d6c-85ab-d848155884ca
xfsrestore: session id: 482a15fb-6b53-4003-89b7-25441a6da518
```

Step 2:

```
 -> exit
the following commands are available:
        pwd
        ls [ <path> ]
        cd [ <path> ]
        add [ <path> ]
        delete [ <path> ]
        extract
        quit
        help

 -> pwd
cwd is fs root

 -> cd /etc

 -> pwd
etc

 -> ls
        36701523 locale.conf
        71400982 gnome-vfs-2.0/
         2667388 gtk-3.0/
        36182357 at-spi2/
         2667646 speech-dispatcher/
```

Step 3:

```
         33557197 pki/
        100663950 libreport/
              170 X11/
         67149962 fonts/
         33554564 mtab
         33554563 crypttab
         36509895 fstab

 -> add passwd

 -> add shadow

 -> add X11

 -> add yum

 -> extract

 ---------------------------------- end dialog -------------------------------

xfsrestore: restoring non-directory files
xfsrestore: restore complete: 152 seconds elapsed
xfsrestore: Restore Summary:
xfsrestore:   stream 0 /root/Desktop/d1/myetcdump OK (success)
xfsrestore: Restore Status: SUCCESS
[root@client d1]#
```

Step 4: verify in the directory, extracted files must be there.

Boot

Thursday, March 14, 2019 10:42 AM

Power On --> MBR --> GRUB --> Kernel --> Runlevel --> Login screen

Power On

MBR

- Master Boot Record
- 1st sector of HDD
- Performs POST test
- Consists of boot loader machine code.
- 1st stage of boot loader & it's job is to locate 2nd stage of boot loader.

GRUB

- Grand Unified Boot loader.
- 2nd stage of boot loader.
- This is a graphical scree for various O.S or Kernel.
- After the selection, it locates corresponding files in /boot directory.
- Kernel name is VMLINUZ.
- Then it loads "initramfs" which loads kernel modules & drivers.

KERNEL

- Initialize & configure systems memory & hardware.
- Then it decompresses the initramfs in a specific location "/sysroot" & loads drivers..
- Then it initialize virtual drives (Like RAID, LVM) before completing initramfs & free up the memory.
- Kernel creates a root device & then kernel is loaded in memory as READONLY.
- Then, it initiates "/sbin/init" program.

/sbin/init

- It get starts & becomes the first process by kernel.
- Then it runs /etc/rc.d/rc/sysinit, which sets up ENV path, start swap, checks file system.
- Then it loads specific runlevel, listed in /etc/rc.d/rc<x>.d/dir
- Then it starts source function lib, /etc/rc.d/init.d/functions for system, which configure how to start, kill, determine PID of a program.
- It starts BG process by looking at runlevel in /etc/inittab.
- In runlevel 5, "UPSTART" runs a script called /etc/x11/prefdm.
    - Prefdm executes X display manager.

LOGIN SCREEN

Bonding / NIC Teaming

Saturday, June 15, 2019 12:00 AM

WORKING TEAMING COMMANDS:

==========

0. man teamd.conf
1. nmcli connection add con-name team type team ifname team config '{"runner": {"name": "loadbalance"}}'
2. nmcli connection show
3. nmcli connection add con-name team-slave1 type team-slave ifname eno33554992 master team
4. nmcli connection show
5. nmcli connection add con-name team-slave2 type team-slave ifname eno50332216 master team
6. nmcli connection show
7. teamdctl team state
8. nmcli connection modify team ipv4.addresses 192.168.100.100/24 ipv4.method manual connection.autoconnect yes
9. nmcli connection up team
10. ifconfig | grep inet 11. ping 192.168.100.100

==========

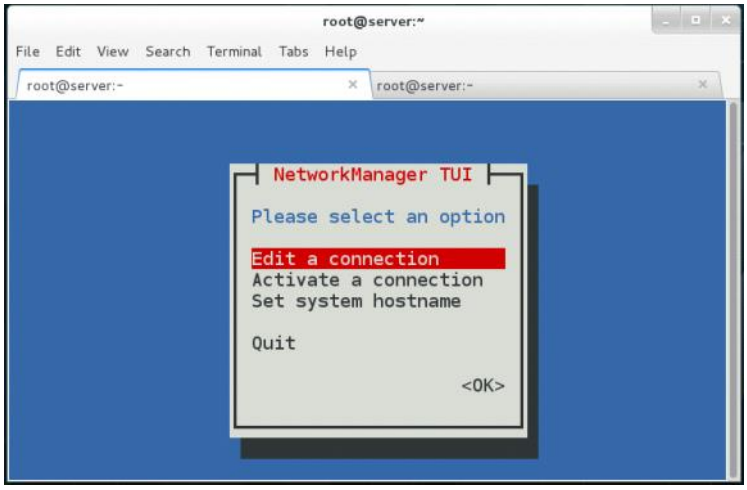
1. Adding 2NICS in the VM
2. By default, bonding is disabled. Check it
   #lsmod | grep bonding

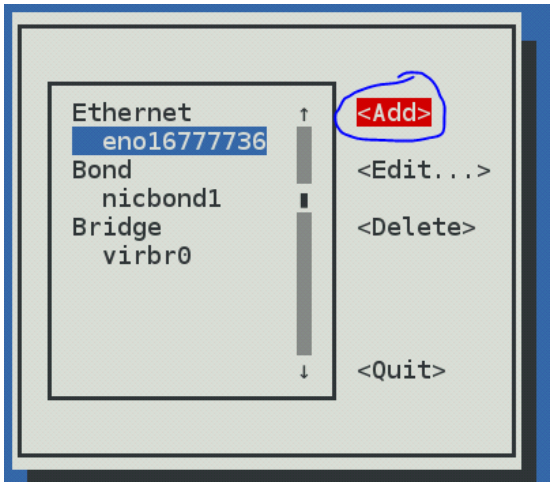
3. Enable the bonding & making it permanent.
   #modprobe --first-time bonding

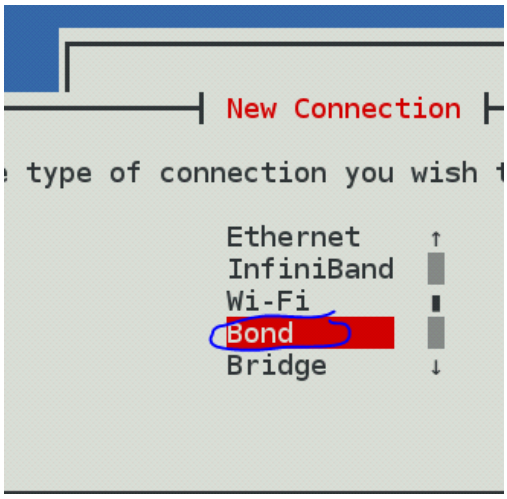
4. To make it permanent, create a ".conf" file
   #vim /etc/modules-load.d/bonding.conf
   #enter below line in the above file:
   bonding

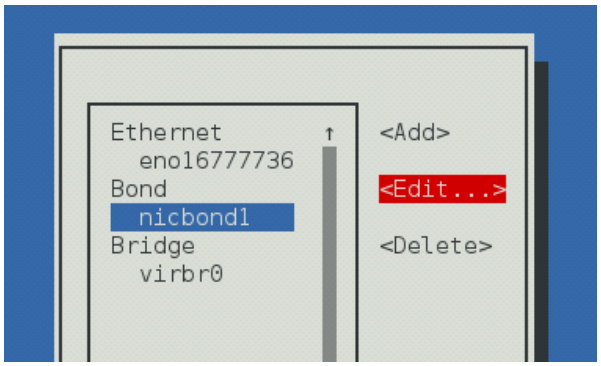
5. Check if its present else reboot the system:
   #lsmod | grep bonding

6. Open NMTUI & add NIC to a new connection.
   #nmtui

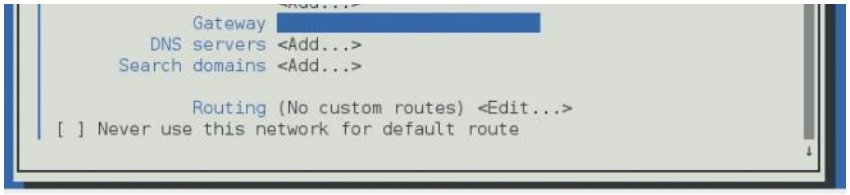



Ethernet
eno16777736
Bond
nicbond1
Bridge
virbr0
<Add>
<Edit...>
<Delete>
<Quit>


New Connection
type of connection you wish t
Ethernet
InfiniBand
Wi-Fi
Bond
Bridge


Ethernet
eno16777736
Bond
nicbond1
Bridge
virbr0
<Add>
<Edit...>
<Delete>


Edit connection
Profile name nicbond1
Device bond0
BOND
Slaves
<Hide>
Ethernet connection 2
Ethernet connection 1
<Add>
<Edit...>
<Delete>
Mode <Round-robin>
Link monitoring <MII (recommended)>
Monitoring frequency 100 ms
Link up delay 0 ms
Link down delay 0 ms
IPv4 CONFIGURATION <Manual>
<Hide>
Addresses 192.168.10.100/32 <Remove>
<Add...>
Gateway
DNS servers <Add...>
Search domains <Add...>



Save & quit

Make the new NIC up
#ip link set dev bond0 up

Display IP
#ip addr show bond0

Bulk user script using shell

Saturday, March 3, 2018 9:23 AM

MultiUser creating script in linux

First create a file which contains all the user name. Something like this:

u1
u2
u3
u4
u5
Save the file as userlist.txt.

Now create the following bash file:
for i in `more userlist.txt `
do
adduser $i
done

Save the file and exit.
#chmod 755 userlist.txt

```
#!/bin/sh
for i in `more userlist.txt `
do
echo $i
echo $i"123" | passwd –-stdin "$i"
echo; echo "User $username’s password changed!"
done
```

CLI to GUI

Saturday, March 3, 2018 9:17 AM

1. Install yum server

2. Check the avialable list of pacakges
    #yum group list

3. Install all pacakges related to GUI in bulk
    #yum groupinstall 'Server with GUI' -y
        OR
    #yum groupinstall 'X Window System' 'GNOME'

4. once this is done, change default CLI target to GRAPHICAL target
    #systemctl set-default graphical.target

5. Once it's update, check the GNOME version
    #gnome-shell --version

Cluster

Thursday, February 28, 2019 11:28 AM

https://www.tecmint.com/what-is-clustering-and-advantages-disadvantages-of-clustering-in-linux/

Capg

Thursday, December 20, 2018 10:49 PM

/etc/rc.d/rc.sysinit

- This file is executed by /etc/system/sysinit script
- This initiates the following
  - Random
    - It starts a secure random-number generator to provide numbers for use in encryption. It uses /dev/random file to generate high quality numbers.
  - Dumper
    - If the '/var/dumps' directory rc.sysinit initiates dumper utility, which will capture dumps in /var/dumps, when a process is terminated abnormally.
  - Rc.local
    - Runs local configuration
  - Tinit
    - Terminal initialization
    - It runs as a background process
    - It is used for login on the terminal console(s)

/etc/rc

- This is responsible for starting/stopping the service when run level changes.

Udev

- udev is the device manager for the Linux 2.6 kernel series.
- It manages device nodes in /dev while adding or removing any device including firmware.
- File location is : #cd /etc/udev/rules.d

<u>Nohup</u>

- Means "No Hangup"
- This will help in preventing a process to get stopped after the user logs off from the system.
- Example:
  - #login to Jeetu user
  - #nohup yes&
  - #logout from Jeetu user & switch to root user
  - #ps -u <Jeetu>

<u>Bash profile</u>

- .bash_profile is meant for login shells and interactive sessions.
- .bashrc is used for non-interactive sessions.

1. .bash_history = holds history (#echo $HISTFILE, #echo $HISTFILESIZE)
2. .bashrc = it holds non interactive sessions & aliases
3. .bash_logout = it gets executed when user logout from the linux
4. .bash_profile = whenever terminal in invoked, this is called. Used for environment variables.

Yum server

#rpm -q createrepo python-deltarpm deltarpm vsftpd

#mkdir /var/ftp/pub/yumserver

#cp -var * /var/ftp/pub/yumserver

#create client file @ /etc/yum.repos.d/yumserver.repo

#createrepo -v /var/ftp/pub/yumserver/

- It's a python program that creates several XML files
  - Repomd.xml
  - Primary.xml.[gz]
  - Filelists.xml.[gz]
  - Other.xml.[gz]
  - Groups.xml.[gz]

#Enable, start & status VSFTPD service

#chkconfig --list vsftpd

#chkconfig --add vsftpd

#chkconfig vsftpd on

#chkconfig --list vsftpd

#service vsftpd start/restart

#service vsftpd status

#yum clean all

#yum update all

#yum repolist

Adding grub passwd in RHEL 6 (link)

#grub-md5-crypt

#type pwd twice & copy output

#vim /boot/grub/grub.conf

Add below above "splashimage"

Password --md5 <output-of-grub-md5-crypt>

CUPS service

Sunday, December 23, 2018 3:23 PM

- Common Unix Printing System (CUPS)

Packages Required	yum install cups ghostscript.x86_64 hplip-common.x86_64
Config Files	/etc/cups/cupsd.conf; /etc/cups/cups-browsed.conf
Port Numbers	631
Service	# service cups stop ### For CentOS/RHEL 6 # systemctl stop cups ### For CentOS/RHEL 7 # systemctl enable cups-browsed # systemctl start cups-browsed
Commands	R7Config
Firewall rules	FW Traffic

Installation & configuration of CUPS:

Install Cups
[root@dlp ~]# yum -y install cups

Configure Cups
[root@dlp ~]# vi /etc/cups/cupsd.conf

# line 18: change	Listen 631
# line 31: add access permission	<Location /> Order allow,deny Allow 10.0.0.0/24 </Location>
# line 37: add access permission	<Location /admin> Order allow,deny Allow 10.0.0.0/24 </Location>
# line 43: add access permission	<Location /admin/conf> AuthType Default Require user @SYSTEM Order allow,deny Allow 10.0.0.0/24 </Location>
# add at the last: specify certificates	ServerCertificate /etc/pki/tls/certs/server.crt ServerKey /etc/pki/tls/certs/server.key

[root@dlp ~]# /etc/rc.d/init.d/cups start

[root@dlp ~]# chkconfig cups on

Accessing CUPS service: http://<server-IP>:631

Username : root
Password : <root's password>

File types in Linux

Saturday, November 30, 2019 2:21 PM

-	Regular file
D	Directory
C	character device file
S	local socket file
P	Names pipe
L	Symbolic link

Regular files:

- Commonly used
- Includes all text, images, binary files, shared libraries.
- Can be also created by "touch" command.
- Denoted by "-".

Directory:

- Denoted by "d"

Character device file:

- Character & block device files allow users & program to communicate with hardware.
- Denoted by "c".
- Ex:-
  - #ls -ld /dev/vmmon
    crw------- 1 root root 10, 165 Jan 4 10:13 /dev/vmmon

Block devices:

- Similar to "character device".
- Govern hardware device as hard disks, memories.
- Denoted by "b"
- Ex:-
  - #ls -ld /dev/sda
    brw-rw---- 1 root disk 8, 0 Jan 4 10:12 /dev/sda

Sockets:

- Usually used for communication between process.
- Used by syslogs, X windows etc.
- Ex:-
  - ls -ld /dev/log
    srw-rw-rw- 1 root root 0 Jan 4 10:13 /dev/log

Named pipes:

- Similar to local devices.
- Allows communication between two local process.
- Can be created by "mknod" cmd & can be deleted by "rm" cmd.

Symbolic link:

- Administrator can assign a file or directory multiple identities.
- Symbolic link can be though of as a pointer to an original file.
- There are two types of symbolic links:
  - hard links
  - soft links
- Denoted by "l".

- Cmd to create it is "ls -l".
- To remove link, "unlink" or to delete "rm".

Converting file systems

Thursday, December 5, 2019 12:02 PM

Converting Ext2 to Ext3

To change an ext2 file system to ext3 enabling the journal feature, use the command.

- # tune2fs -j /dev/hdXX

Converting Ext2 to Ext4

- To convert from old ext2 to new ext4 file system with latest journaling feature. Run the following command.
- # tune2fs -O dir_index,has_journal,uninit_bg /dev/hdXX

Next do a complete file system check with e2fsck command to fix and repair.

- # e2fsck -pf /dev/hdXX

Converting Ext3 to Ext4

To enable the ext4 features on an existing ext3 filesystem, use the command.

- # tune2fs -O extents,uninit_bg,dir_index /dev/hdXX

After running this command we MUST run fsck to fix up some on-disk structures that tune2fs has modified.

- # e2fsck -pf /dev/hdXX

Firewall cmds - RHEL 7

Thursday, December 5, 2019 1:28 PM

Get firewall based zones	#firewall-cmd --get-zones
Query the default zone	# firewall-cmd --list-all
Query the external zone	# firewall-cmd --zone=external --list-all
Zones manipulation	# firewall-cmd --set-default=external
add the samba service to the external zone	# firewall-cmd --zone=external --add-service=samba
Reload firewall	# firewall-cmd --reload
Check the services allowed in the external zone:	# firewall-cmd --zone=external --list-services
Add samba service permanently	# firewall-cmd --permanent --zone=external --add-service=samba
Remove samba service permanently	# firewall-cmd --permanent --zone=external --remove-service=samba
Adding a port number permanently	# firewall-cmd --permanent --zone=external --add-port=139/tcp
Listing ports	firewall-cmd --zone=external --list-ports
Display firewall service is running	# firewall-cmd --state
To list details of default zone	# firewall-cmd --get-default-zone
To add/remove interfaces to zones	# firewall-cmd --zone=public --change-interface=eth1
To add a series of ports used by service (ftp)	#firewall-cmd --permanent --add-port=21/tcp --add-port=3000-3500/tcp

The panic mode
The panic mode is a mode that should be used only in situations where there are really serious problems with the network environment. When this mode is active, all existing connections are discarded, and all incoming and outgoing packets are be dropped. It can be enabled running:

firewall-cmd --panic-on

To exit panic mode, the command is:
firewall-cmd --panic-off

It's even possible to query the panic mode status, running:
firewall-cmd --query-panic

VSFTPD

Friday, December 6, 2019 1:33 PM

Step 1: Installing FTP Server
yum install vsftpd
systemctl start vsftpd
systemctl enable vsftpd

firewall-cmd --zone=public --permanent --add-port=21/tcp
firewall-cmd --zone=public --permanent --add-service=ftp
firewall-cmd --reload

Step 2: Configuring FTP Server
cp /etc/vsftpd/vsftpd.conf /etc/vsftpd/vsftpd.conf.orig
Next, open the config file above and set the following options with these corresponding values:

```
anonymous_enable=NO          # disable  anonymous login
local_enable=YES             # permit local logins
write_enable=YES             # enable FTP commands which change the filesystem
local_umask=022              # value of umask for file creation for local users
dirmessage_enable=YES        # enable showing of messages when users first enter a new directory
xferlog_enable=YES           # a log file will be maintained detailing uploads and downloads
connect_from_port_20=YES     # use port 20 (ftp-data) on the server machine for PORT style connections
xferlog_std_format=YES       # keep standard log file format
listen=NO                    # prevent vsftpd from running in standalone mode
listen_ipv6=YES              # vsftpd will listen on an IPv6 socket instead of an IPv4 one
pam_service_name=vsftpd      # name of the PAM service vsftpd will use
userlist_enable=YES          # enable vsftpd to load a list of usernames
tcp_wrappers=YES             # turn on tcp wrappers
```

configure FTP to allow/deny FTP access to users based on the user list file /etc/vsftpd.userlist

- userlist_enable=YES # vsftpd will load a list of usernames, from the filename given by userlist_file
- userlist_file=/etc/vsftpd.userlist # stores usernames.
- userlist_deny=NO

Now add these two following options to restrict FTP users to their Home directories.

- chroot_local_user=YES
- allow_writeable_chroot=YES

Securing FTP Server with SELinux

- # setsebool -P ftp_home_dir on
- # semanage boolean -m ftpd_full_access --on
- # systemctl restart vsftpd

Step 4: Testing FTP Server
useradd -m -c “Ravi Saive, CEO” -s /bin/bash ravi
passwd ravi

Afterwards, we have to add the user ravi to the file /etc/vsftpd.userlist using the echo command as follows:

```
# echo "ravi" | tee -a /etc/vsftpd.userlist
# cat /etc/vsftpd.userlist
```

Time to test if our settings above are working correctly

```
# ftp 192.168.56.10
Connected to 192.168.56.10  (192.168.56.10).
220 Welcome to TecMint.com FTP service.
Name (192.168.56.10:root) : anonymous
530 Permission denied.
Login failed.
ftp>
```

also test if a user not listed in the file /etc/vsftpd.userlist

```
# ftp 192.168.56.10
Connected to 192.168.56.10  (192.168.56.10).
220 Welcome to TecMint.com FTP service.
Name (192.168.56.10:root) : aaronkilik
530 Permission denied.
Login failed.
ftp>
```

https://www.tecmint.com/install-ftp-server-in-centos-7/

DHCP SERVER

Saturday, March 3, 2018 9:20 AM

Packages Required	dhcp
Config Files	/var/lib/dhcp; /etc/dhcp/dhcpd.conf
Port Numbers	UDP/67 (server), UDP/68 (client)
Service Name	# systemctl restart network; # service network restart
Commands	
Firewall rules	$IPTABLES -I INPUT -i $LAN_IFACE -p udp --dport 67:68 --sport \ 67:68 -j ACCEPT

RedHat

CREATING DHCP SERVER IN LINUX

[root@dlp ~]# yum -y install dhcp
[root@dlp ~]# vi /etc/dhcp/dhcpd.conf

create new

- # specify domain name
  - option domain-name "srv.world";
- # specify name server's hostname or IP address
  - option domain-name-servers dlp.srv.world;
- # default lease time
  - default-lease-time 600;
- # max lease time
  - max-lease-time 7200;

```
# this DHCP server to be declared valid
        authoritative;

# specify network address and subnet mask
        subnet 10.0.0.0 netmask 255.255.255.0 {
                # specify the range of lease IP address
                range dynamic-bootp 10.0.0.200 10.0.0.254;
                # specify broadcast address
                option broadcast-address 10.0.0.255;
                # specify default gateway
                option routers 10.0.0.1;
        }
```

[root@dlp ~]# systemctl start dhcpd
[root@dlp ~]# systemctl enable dhcpd

[2] If Firewalld is running, allow DHCP service. DHCP Server uses 67/UDP.
[root@dlp ~]# firewall-cmd --add-service=dhcp --permanent
success
[root@dlp ~]# firewall-cmd --reload
success

DHCP server - Verified

Wednesday, December 11, 2019 2:07 PM

Server:

IP address:	192.168.10.10
Package:	#yum install -y dhcp
Copy Configuration:	# cp /usr/share/doc/dhcp-4.1.1/dhcpd.conf.sample /etc/dhcp/dhcpd.conf

Edit config file:	/etc/dhcp/dhcpd.conf
	---------- 	
	From line : 7
	option domain-name "training.local";
	option domain-name-servers ns1.example.org, ns2.example.org;
	default-lease-time 600;
	max-lease-time 7200;
	
	From line : 27
	subnet 192.168.10.0 netmask 255.255.255.0 {
	option routers 192.168.10.254;
	option subnet-mask 255.255.255.0;
	option domain-search "training.local";
	option domain-name-servers 192.168.10.1;
	option time-offset -18000; # Eastern Standard Time
	range 192.168.10.100 192.168.10.150;
	}
	
	#To make static IP for a host using MAC address (ifconfig)
	host client {
	option host-name "client.training.local";
	hardware ethernet 00:0c:29:e4:19:d7;
	fixed-address 192.168.10.149;
	}
	
	Enable & Restart service
	#systemctl stop dhcp
	#systemctl restart dhcp

- Good ece35

Client:
#vim /etc/sysconfig/network-scripts/ifcfg-ens33

DEVICE=eth1
BOOTPROTO=dhcp
TYPE=Ethernet
ONBOOT=yes

systemctl restart network

Link: https://tecadmin.net/configuring-dhcp-server-on-centos-redhat/

Squid Proxy - R7 (working)

Wednesday, December 11, 2019 3:03 PM

Install squid package:

#yum install -y squid
systemctl start squid
systemctl enable squid
systemctl status squid

some important file locations you should be aware of:

- Squid configuration file: /etc/squid/squid.conf
- Squid Access log: /var/log/squid/access.log
- Squid Cache log: /var/log/squid/cache.log

Install squid package

yum install -y squid

Enable squid service

systemctl enable squid
systemctl start squid
systemctl status squid

To view logs

#tail -f /var/log/squid/access.log

Go to client systems (windows/Linux)

#Change the proxy settings using <IP>:3128 and open any site.

Jump back to the squid server & view the packets flowing from SQUID server.

Edit the configuration file:

#cp /etc/squid/squid.conf /etc/squid/squid.conf-backup
#vim /etc/squid/squid.conf

NOTE: ADD EVERYTHING AT THE TOP LINES.
Line 5: acl badsites url_regex "/etc/squid/badsites"
Line 6: http_access deny badsites
Line 7: acl mylocalnet src 192.168.6.0/24
Line 8: http_access allow mylocalnet

Line 66: http_access allow localnet --> COMMENT THIS LINE

:wq!

Create a new file:

#vim /etc/squid/badsites

Facebook.com
Youtube.com
:wq!

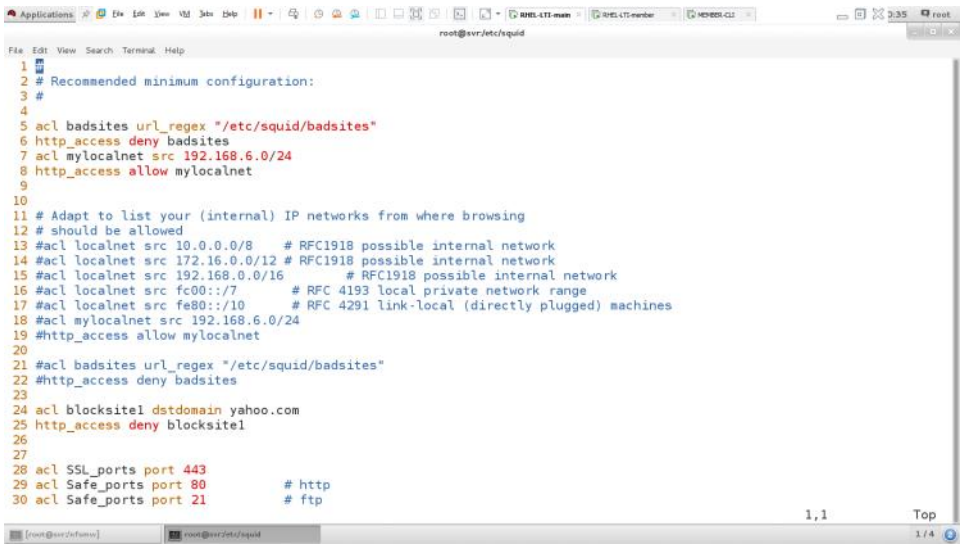
Restart & status of squid server:

systemctl restart squid
systemctl status squid

For safe side:

#chown root:squid /etc/squid/badsites

#chmod 777 /etc/squid/badsites



Ext2 to ext3 - tested

Monday, September 23, 2019 8:17 PM

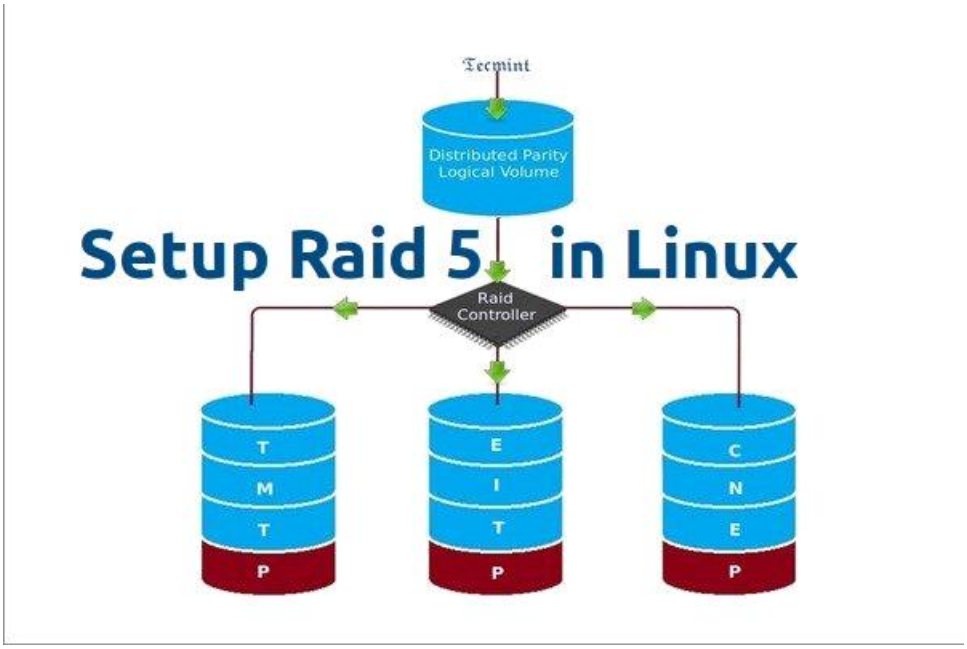
1. Create a new disk (/dev/sda4) with ext2 file system.
2. Verify.

To convert the ext2 file system to ext3 follow:

1. Unmount the disk
    a. #umount /dev/sda4
2. Enabling journaling using tune2fs command
    a. #tune2fs -j /dev/sda4
3. Now modify the entry in the /etc/fstab file
    a. #vim /etc/fstab
    /dev/sda4 /data ext3 defaults 1 2
4. Mount & verify
    a. #mount -a
    b. #mount

RAID 5 - tested

Monday, September 23, 2019 10:14 PM



What is Parity?

Parity is a simplest common method of detecting errors in data storage. Parity stores information in each disks, Let's say we have 4 disks, in 4 disks one disk space will be split to all disks to store the parity information's. If any one of the disks fails still we can get the data by rebuilding from parity information after replacing the failed disk.

Pros and Cons of RAID 5

- Gives better performance
- Support Redundancy and Fault tolerance.
- Support hot spare options.
- Will loose a single disk capacity for using parity information.
- No data loss if a single disk fails. We can rebuilt from parity after replacing the failed disk.
- Suits for transaction oriented environment as the reading will be faster.
- Due to parity overhead, writing will be slow.
- Rebuild takes long time.

mdadm is a package which allow us to configure and manage RAID devices in Linux. By default there is no configuration file is available for RAID, we must save the configuration file after creating and configuring RAID setup in separate file called mdadm.conf.

1. Install the packages:
   #yum install mdadm

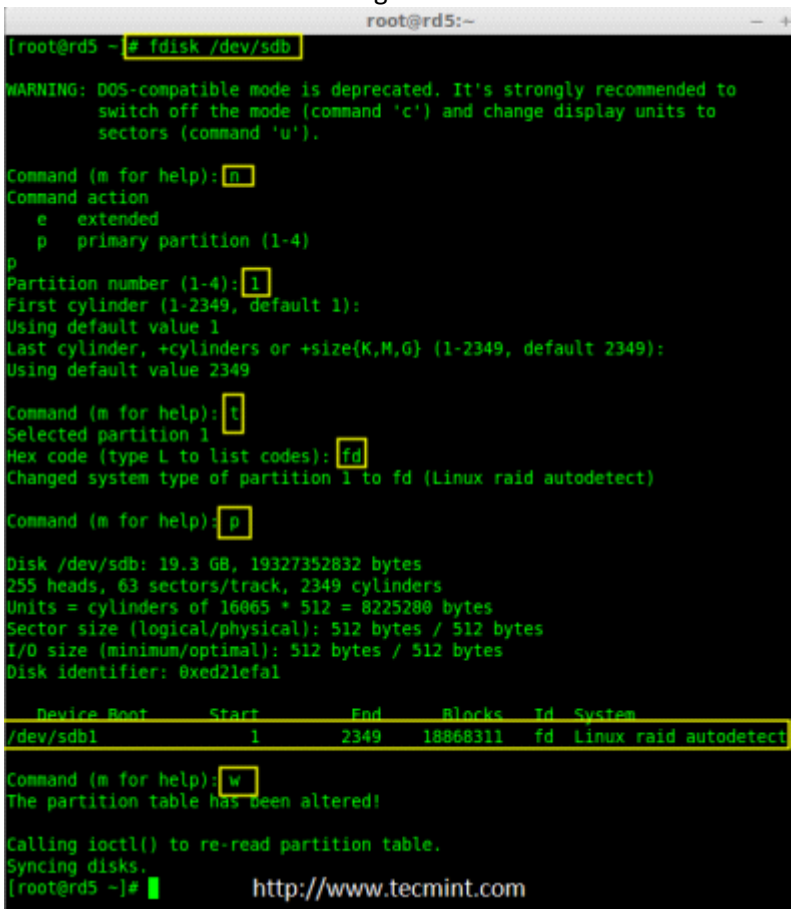
2. After the 'mdadm' package installation
   # fdisk -l | grep sd

3. Now it's time to examine the attached three drives for any existing RAID blocks on these drives using following command.
   # mdadm -E /dev/sd[b-d] OR
   # mdadm --examine /dev/sdb /dev/sdc /dev/sdd

4. Partitioning the Disks for RAID
   # fdisk /dev/sdb
   # fdisk /dev/sdc

fdisk /dev/sdd

5. Create /dev/sdb Partition
   Please follow the below instructions to create partition on /dev/sdb drive.
   - Press ‘n‘ for creating new partition.
   - Then choose ‘P‘ for Primary partition. Here we are choosing Primary because there is no partitions defined yet.
   - Then choose ‘1‘ to be the first partition. By default it will be 1.
   - Here for cylinder size we don’t have to choose the specified size because we need the whole partition for RAID so just Press Enter two times to choose the default full size.
   - Next press ‘p‘ to print the created partition.
   - Change the Type, If we need to know the every available types Press ‘L‘.
   - Here, we are selecting ‘fd‘ as my type is RAID.
   - Next press ‘p‘ to print the defined partition.
   - Then again use ‘p‘ to print the changes what we have made.
   - Use ‘w‘ to write the changes.

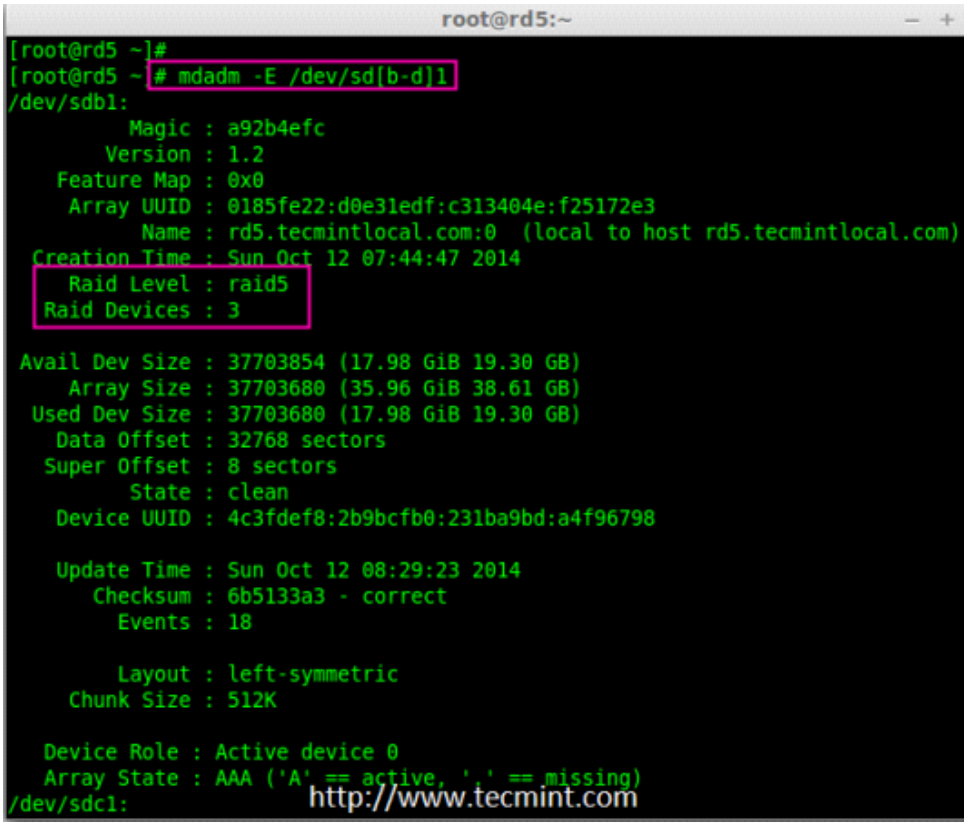


6. After creating partitions, check for changes in all three drives sdb, sdc, & sdd.
   # mdadm --examine /dev/sdb /dev/sdc /dev/sdd

7. Creating md device md0
   Now create a Raid device ‘md0‘ (i.e. /dev/md0) and include raid level on all newly created partitions (sdb1, sdc1 and sdd1) using below command.
   # mdadm --create /dev/md0 --level=5 --raid-devices=3 /dev/sdb1 /dev/sdc1 /dev/sdd1

8. After creating raid device, check and verify the RAID, devices included and RAID Level from the mdstat output.
   # cat /proc/mdstat

9. After creation of raid, Verify the raid devices using the following command.
   # mdadm -E /dev/sd[b-d]1



10. Next, verify the RAID array to assume that the devices which we've included in the RAID level are running and started to re-sync.
    # mdadm --detail /dev/md0

11. Now create a directory under '/mnt' then mount the created filesystem under /mnt/raid5 and check the files under mount point, you will see lost+found directory.
    # mkdir /mnt/raid5
    # mount /dev/md0 /mnt/raid5/
    # ls -l /mnt/raid5/

12. The mount point will differ according to your environment.
    # vim /etc/fstab
    /dev/md0 /mnt/raid5 ext4 defaults 0 0

13. Save Raid 5 Configuration
    # mdadm --detail --scan --verbose >> /etc/mdadm.conf

URL: https://www.tecmint.com/create-raid-5-in-linux/

Disable the user list - R7 verified

Wednesday, November 13, 2019 11:52 AM

1. Create the gdm profile which contains the following lines:
   /etc/dconf/profile/gdm
   user-db:user
   system-db:gdm
   file-db:/usr/share/gdm/greeter-dconf-defaults
   gdm is the name of a dconf database.

2. Create a gdm keyfile for machine-wide settings in /etc/dconf/db/gdm.d/00-login-screen:
   [org/gnome/login-screen]
   # Do not show the user list
   disable-user-list=true

3. Update the system databases:
   # **dconf update**

From <https://help.gnome.org/admin/system-admin-guide/stable/login-userlist-disable.html.en>

Direct Login

Saturday, March 3, 2018 9:21 AM

RHEL 7
---

open the file & edit the lines under [daemon]
vim /etc/gdm/custom.conf
[daemon]
AutoLoginEnable=true
AutoLogin=root
:wq!
Reboot
==========

RHEL 6

#cd /etc/pam.d
#ls | grep gdm
gdm
gdm-autologin
gdm-fingerprint
gdm-password

edit both files "gdm" & "gdm-password"
delete or comment '#' lines where "pam_succeed---" is written
==========

R7

Link: https://gooroo.io/GoorooThink/Article/16500/How-to-RemoveDisable-User-list-on-GDM-Login-Screen-in-RHEL7-/20580#.XDtqL3duLDc

[root@mail gdm]# vim /etc/dconf/db/gdm.d/01-cusstom-gdm-settings
[root@mail gdm]# cat /etc/dconf/db/gdm.d/01-cusstom-gdm-settings
[org/gnome/login-screen] disable-user-list=true
[root@mail gdm]#
Update the dconf database
[root@mail gdm]# dconf update
Restart gdm.
[root@mail gdm]# systemctl restart gdm.service
Root Cause
In RHEL7/CentOS 7 gdm loads its configuration in its own dconf database. Which is separate from the user dconf database that is accessible from the GUI with dconf-editor. So in order to modify gdm login screen create a custom file in the /etc/dconf/db/gdm.d and add the custom entries there.
Diagnostic Steps
If above command doesn't disable userlist then make sure that "/etc/dconf/db/gdm" have following permission after executing dconf update command:
/etc/dconf/db/gdm -rw-r--r--. 1 root root 5139 Jul 9 22:24 /etc/dconf/db/gdm

Disk Quota - R7 (verified)

Sunday, January 13, 2019 10:40 PM

Steps to create disk Quota

1. Create an additional disk to quota management.
   a. Using basic or LVM partitioning method.
2. Enable quota management for users & groups.
   a. Editing fstab file (with usrquota,grpquota entries for specific partitions).
   b. It works with "user quota" & " group quota".
3. Remount file system
4. Create quota DB & generate disk usage table
5. Assign quota policy

Step1:

```
# /etc/fstab
# Created by anaconda on Mon Nov 26 15:58:37 2018
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=93070753-15a8-432a-9c39-15192e7806b2 /                       xfs     defaults
1 1
UUID=987fe292-114c-4012-8680-c6f4db95c5be /boot                   xfs     defaults
1 2
UUID=92a4bedc-5caa-465e-b252-2df773aa5587 swap                    swap    defaults
0 0
/dev/sda4       /quota  xfs     defaults,usrquota,grpquota        0 0
~
~
```

Step2:

- Remount the partitions
  - #mount -o remount /<partition_name>
    - Output should get on "mount | grep /partition_name"
      - type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
      - That means quota is disabled & needs to be enabled at boot time.
  - Reconfiguring Kernel Boot Options for XFS File Systems
    - Make a backup of /etc/default/grub:
      - #cp /etc/default/grub /etc/default/grub-backup
    - Modify /etc/default/grub

Here is the default file.

```
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=cl/root rd.lvm.lv=cl/swap rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
```

We want to modify the following line –

```
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=cl/root rd.lvm.lv=cl/swap rhgb quiet"
```

to

```
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=cl/root rd.lvm.lv
=cl/swap rhgb quiet rootflags=usrquota,grpquota"
```

- After this, we need to reconfigure the grub:
  - #cp /boot/grub2/grub.cfg /boot/grub2/grub.cfg.backup //take backup
    else boot failure.
  - grub2-mkconfig -o /boot/grub2/grub.cfg

```
[root@jeetu Desktop]# grub2-mkconfig -o /boot/grub2/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-3.10.0-123.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-123.el7.x86_64.img
Found linux image: /boot/vmlinuz-0-rescue-2285b46dedf0454b91eb75a811e18f88
Found initrd image: /boot/initramfs-0-rescue-2285b46dedf0454b91eb75a811e18f88.img
done
```

  - Then Reboot.
  - After that, verify:

```
[root@jeetu Desktop]# mount | grep /quota
/dev/sda4 on /quota type xfs (rw,relatime,seclabel,attr2,inode64,usrquota,grpquota)
[root@jeetu Desktop]#
```

  - This means the usrquota & grpquota parameters has been passed via grub.
  - Now verify that, quota is enabled using "quotaon -ap"

```
File  Edit  View  Search  Terminal  Help
[root@jeetu Desktop]# quotaon -ap
group quota on / (/dev/sda3) is on
user quota on / (/dev/sda3) is on
group quota on /quota (/dev/sda4) is on
user quota on /quota (/dev/sda4) is on
[root@jeetu Desktop]#
```

  - Reboot again for stability.
  - Verify the data is available in the /etc/mtab file:

```
File  Edit  View  Search  Terminal  Help
[root@jeetu Desktop]# cat /etc/mtab | grep /quota
/dev/sda4 /quota xfs rw,seclabel,relatime,attr2,inode64,usrquota,grpquota 0 0
[root@jeetu Desktop]#
```

  - Create user & group quota files in the partition.

```
root@jeetu:~/D
File  Edit  View  Search  Terminal  Help
[root@jeetu Desktop]# touch /quota/aquota.user
[root@jeetu Desktop]# chmod 666 /quota/aquota.user
[root@jeetu Desktop]# touch /quota/aquota.group
[root@jeetu Desktop]# chmod 666 /quota/aquota.group
[root@jeetu Desktop]#
```

- And reboot
- Turn On quota on :
  - #quotaon -af //man quotaon
- Edit "user" quota using
  - #edquota -u <username>

```
root@jeetu:~/Desktop                                  root@jeetu:~/Desktop
Disk quotas for user tomar (uid 1001):
  Filesystem                   blocks       soft       hard     inodes     soft     hard
  /dev/sda3                        12          0          0          8        0        0
  /dev/sda4                         0          0          0          0        0        0
~
~
~
```

  - Blocks: 1K
  - Inodes: number of entries in the directory
  - Soft: max number of blocks/inodes, before user get warning & grace period countdown begins. 0 means no limits.
  - Hard: max number of blocks/inodes. 0 means no limits
  - Edit the quota:

```
File  Edit  View  Search  Terminal  Tabs  Help
root@jeetu:~/Desktop                                  tomar@jeetu:/quota
Disk quotas for user tomar (uid 1001):
  Filesystem                   blocks       soft       hard     inodes     soft     hard
  /dev/sda3                        20          0          0         13        0        0
  /dev/sda4                     15000      15000      15000          2        0        0
~
~
```

    - Only soft & hard = 15000 (means 15Mb)
    - This means the user will not be able to exceed more than 15Mb in /dev/sda4 drive.
  - Now enable quota:
    - #quotacheck -a
  - Give proper permissions to the partition:
    - #chmod 777 /quota
- Switch to <username>
  - $cd /quota
  - $touch testFile.txt //try creating test file.
- Execute below command:
  - #dd if=/dev/zero of=/quota/tomarFile bs=16M count=10 //this command will try to create a file greater that given (15M) in size.

```
[tomar@jeetu quota]$ dd if=/dev/zero of=/quota/tomarFile bs=16M count=10
dd: error writing '/quota/tomarFile': Disk quota exceeded
1+0 records in
0+0 records out
15360000 bytes (15 MB) copied, 0.0139255 s, 1.1 GB/s
[tomar@jeetu quota]$ quota
Disk quotas for user tomar (uid 1001):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
      /dev/sda4   15000*  15000   15000               2       0       0
[tomar@jeetu quota]$
```

- On root, verify quota:

```
[root@jeetu Desktop]# quota -u tomar
Disk quotas for user tomar (uid 1001):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
      /dev/sda4   15000*  15000   15000               2       0       0
[root@jeetu Desktop]#
```

- Verify file size in the quota partition:
  - #du -m <partition> //m= MB

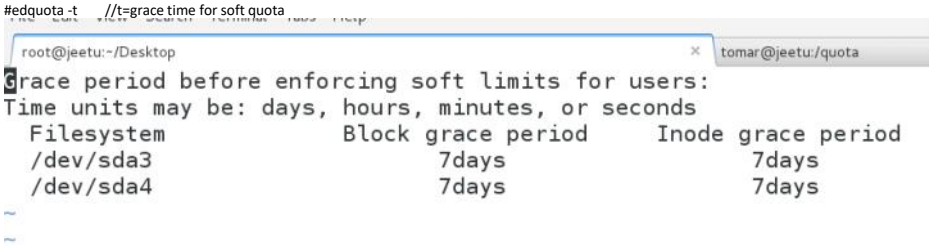
```
[tomar@jeetu quota]$ du -m /quota/*
0       /quota/aquota.group
0       /quota/aquota.user
0       /quota/t1
15      /quota/tomarFile
[tomar@jeetu quota]$
```

- To check the quota limit:

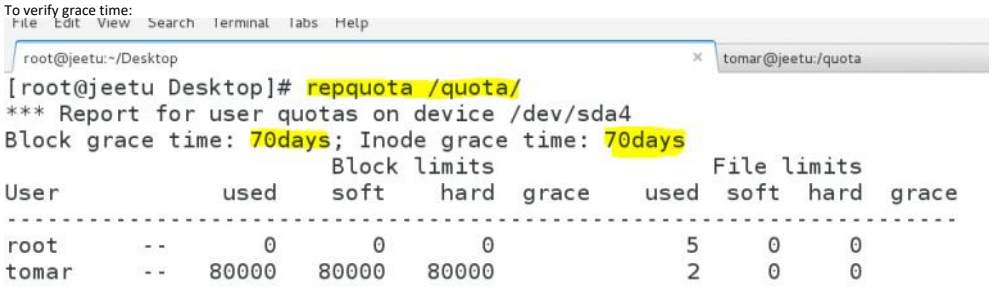
```
root@jeetu:~/Desktop                                  tomar@jeetu:/quota
[root@jeetu Desktop]# repquota /quota
*** Report for user quotas on device /dev/sda4
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      --       0       0       0              5     0     0
tomar     --   80000   80000   80000              2     0     0
```

- To change GRACE time for any quota:

- #edquota -t //t=grace time for soft quota



- To verify grace time:



- To see all quotas:
  - #repquota -aug

==========
==========
Link: https://www.linuxtechi.com/enable-user-group-disk-quota-on-centos-7-rhel-7/

When output gives "noquota":

https://www.tutorialspoint.com/linux_admin/linux_admin_quota_management.htm

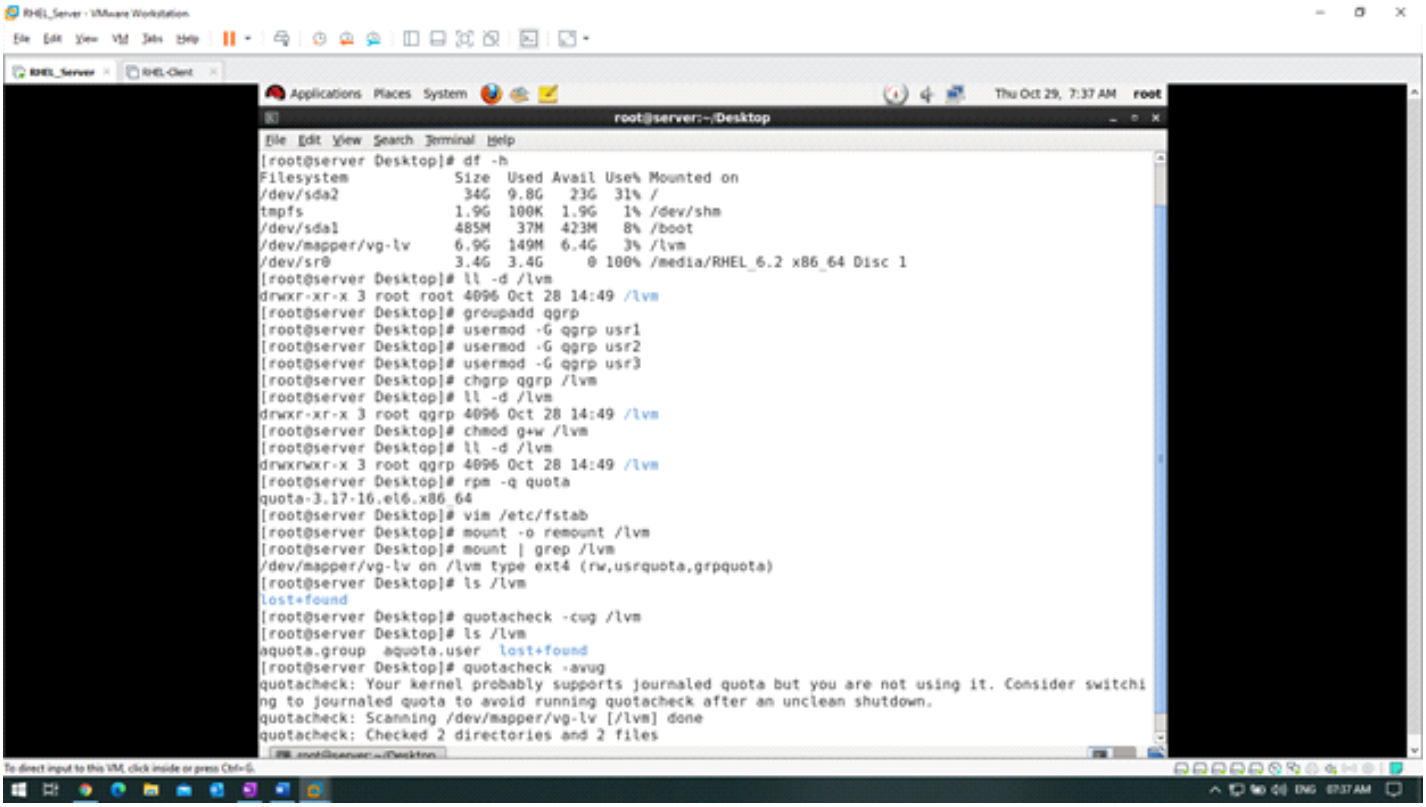
https://www.youtube.com/watch?v=y6LgbQeXKPE

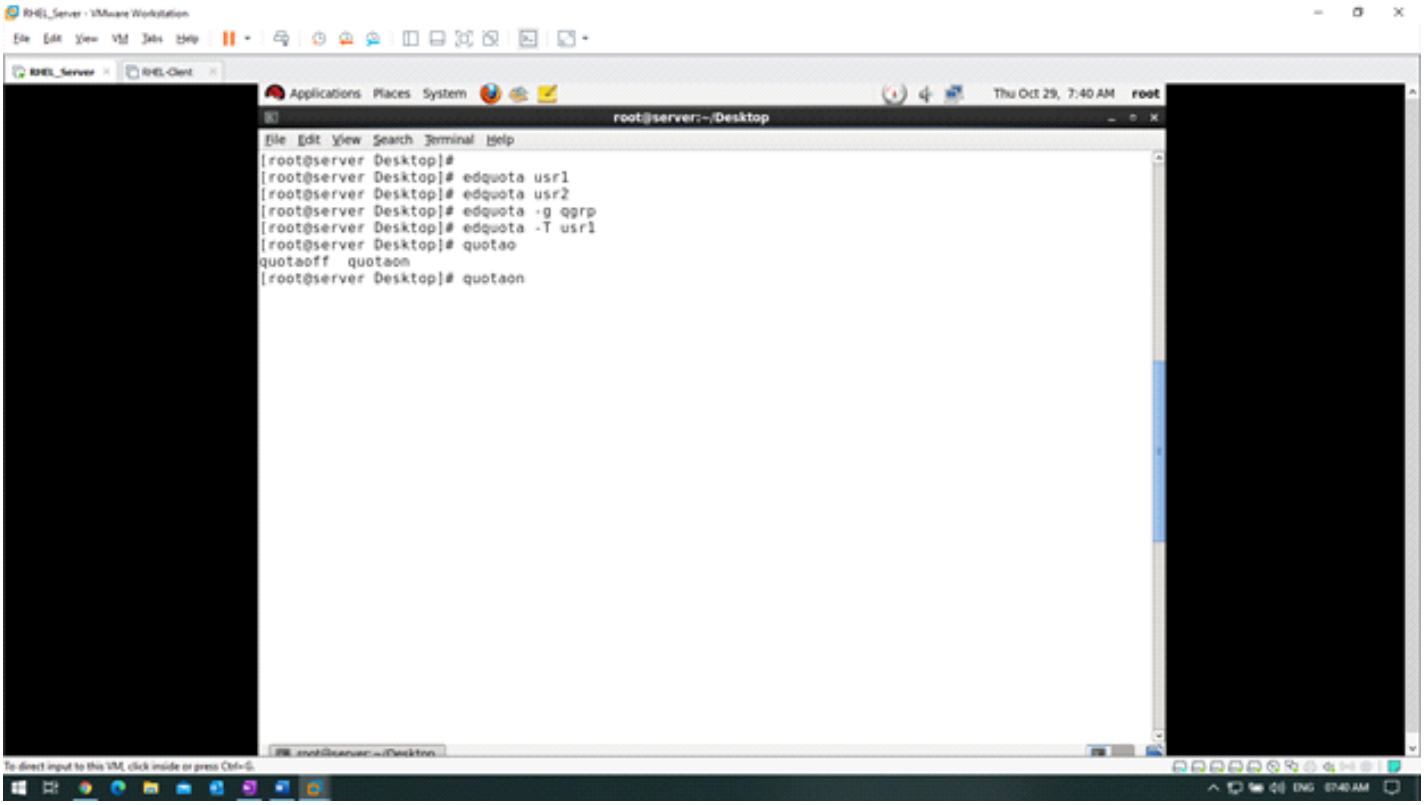
http://www.yolinux.com/TUTORIALS/LinuxTutorialQuotas.html

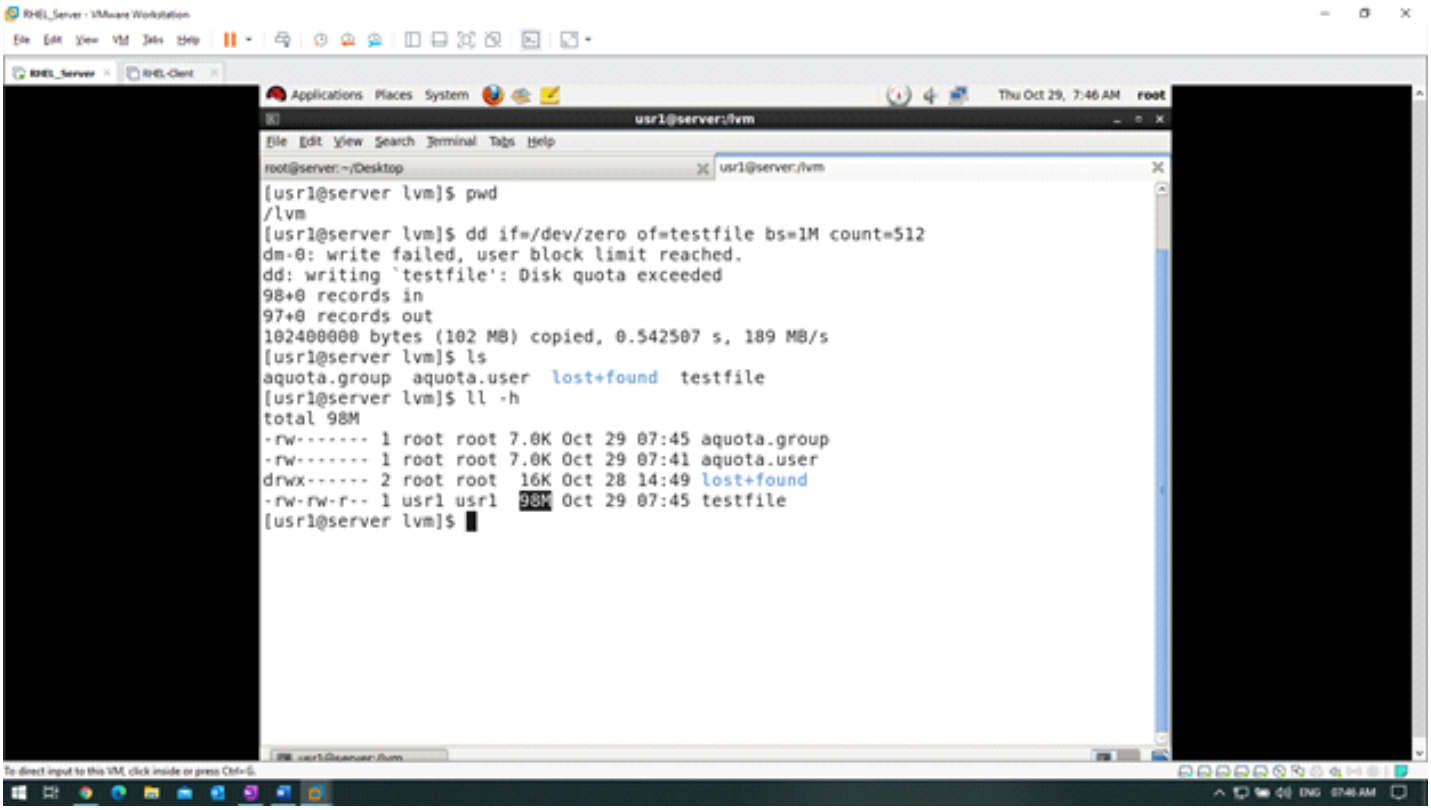
Disk Quota - R6 (verified)

29 October 2020 08:03 AM

Quota on /lvm:







Cmds:

List partition	#df -h
Listing permissions	#ll -d /lvm
Create group	#groupadd qgrp
Adding using usr1 & usr2 to the group	#usermod -G qgrp usr1 #usermod -G qgrp usr2
Change group owner of /lvm	#chgrp qgrp /lvm
Listing permissions	#ll -d /lvm
Allow write permission for group on /lvm	#chmod g+w /lvm
Checking quota package is installed.	#rpm -q quota
Allow user & group quota in /lvm	#vim /etc/fstab /dev/vg/lv /lvm ext4 defaults,usrquota,grpquota 0 0
Remount file system	#mount -o remount /lvm
Listing the changes	#mount | grep /lvm
Listing the folder	#ls /lvm
Creating quota on /lvm	#quotacheck -cug /lvm
Listing the folder	#ls /lvm
Create disk quota database	#quotacheck -avug a :- This option is used to check all quota enabled partitions v :- This option is used to print real time updates as command proceeds u :- This option is used to check user disk quota information g :- This option is used to check group disk quota information
Setting quota value for user (usr1)	#edquota usr1 Column Description 1 Partition where this quota will apply 2 No. of blocks currently used by this user

	3 Soft block size limit for user 4 Hard block size limit for user 5 No. of inodes currently used by this user 6 Soft inodes limit for user 7 hard inodes limit for user Set hardquota = 100000 (in kb), means 100MB.
Setting quota for group	#edquota -g qgrp
Enabling quota on /lvm	#quotaon /lvm
Switch to usr1 user	#su – usr1
Switch to /lvm from usr1	#cd /lvm
Create a file of 512MB	#dd if=/dev/zero of=testFile bs=1M count=512
Verify the file size that got created	#ll -h
The end of user quota.	

TCP wrapper

Tuesday, November 12, 2019 1:22 PM

Two files which is used for tcp-wrappers
1> ls /etc/hosts.allow
2> ls /etc/hosts.deny

NOTE :- tcp-wrappers will control only that services which are dependent on
----- `libwrap.so` library.

#which vsftpd ---> (it will show the binary location of the vsftpd)

ldd /usr/bin/vsftpd ----> (this ldd command will show the dependency list)

strings /usrlib/libwrap.so | less ----> (to read the library)

man 5 hosts_access ----> (man page for tcp-wrappers)

vim /etc/hosts.deny

vsftpd:ALL ------> deny service to everyone

vsftpd:ALL EXCEPT .example.com

vsftpd:ALL EXCEPT 192.168.1.25

vsftpd:192.168.1.0/255.255.255.0

vsftpd:.yahoo.com

sshd,vsftpd:.yahoo.com ----> (for ssh & vsftpd)

ALL:ALL -----> (this is for all service for all daemon)

DNS server

Saturday, March 3, 2018 9:21 AM

Packages Required	Bind, bind-utils
Config Files	/etc/named.conf; /etc/resolv.conf
Port Numbers	TCP port 53; UDP port 53
Service Name	# service named start # service named stop # service named restart # service named reload # service named status ########## # systemctl restart named # systemctl enable named # systemctl status named
Commands	# dig @127.0.0.1 broker.example.com;
Firewall rules	# firewall-cmd --add-port=53/udp # firewall-cmd --add-port=53/udp --permanent

Install BIND to configure DNS server which resolves domain name or IP address.

[1] Install BIND.

```
[root@dlp ~]# yum -y install bind bind-utils
```

[2] Configure BIND.

This example shows to set with grobal IP address [172.16.0.80/29], Private IP address [10.0.0.0/24], Domain name [srv.world]. However, Please use your own IP addresses and domain-name when you configure your own server. (Actually, [172.16.0.80/29] is for private IP address, though.)

```
[root@dlp ~]# vi /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 {
      # change ( listen all )
      listen-on port 53 { any; };
      # change if not use IPv6
      listen-on-v6 { none; };
      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";
        # query range ( set internal server and so on )
        allow-query       { localhost; 10.0.0.0/24; };
        # transfer range ( set it if you have secondary DNS )
        allow-transfer    { localhost; 10.0.0.0/24; };

        recursion yes;

        dnssec-enable yes;
        dnssec-validation yes;
        dnssec-lookaside auto;

        /* Path to ISC DLV key */
        bindkeys-file "/etc/named.iscdlv.key";

        managed-keys-directory "/var/named/dynamic";

        pid-file "/run/named/named.pid";
        session-keyfile "/run/named/session.key";
};
logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};

# change all from here
view "internal" {
        match-clients {
                localhost;
                10.0.0.0/24;
        };
        zone "." IN {
                type hint;
                file "named.ca";
        };
        zone "srv.world" IN {
                type master;
                file "srv.world.lan";
                allow-update { none; };
        };
        zone "0.0.10.in-addr.arpa" IN {
                type master;
                file "0.0.10.db";
                allow-update { none; };
        };
include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
};
view "external" {
        match-clients { any; };
        allow-query { any; };
        recursion no;
        zone "srv.world" IN {
```

```
        type master;
        file "srv.world.wan";
        allow-update { none; };
    };
    zone "80.0.16.172.in-addr.arpa" IN {
        type master;
        file "80.0.16.172.db";
        allow-update { none; };
    };
};

# allow-query ? query range you permit
# allow-transfer ? the range you permit to transfer zone info
# recursion ? allow or not to search recursively
# view "internal" { *** }; ? write for internal definition
# view "external" { *** }; ? write for external definition
# For How to write for reverse resolving, Write network address reversely like below
# 10.0.0.0/24
# network address? 10.0.0.0
# range of network? 10.0.0.0 - 10.0.0.255
# how to write? 0.0.10.in-addr.arpa
# 172.16.0.80/29
# network address? 172.16.0.80
# range of network? 172.16.0.80 - 172.16.0.87
# how to write? 80.0.16.172.in-addr.arpa
```

DNS BIND server RHEL 7 - verified

02 April 2020 08:26 PM

DNS server: 10.10.10.10/8
Client : 10.10.10.11/8

Required Files:

1. Main config file: /etc/named.conf
2. forward lookup: vim /var/named/alpha.corp.db
3. reverse lookup: vim /var/named/10.0.0.db

On DNS server

Install bind server	#yum -y install bind bind-utils

Main Configure file with zones	#vim /etc/named.conf
	---------- 	
	Line 17: allow-query { localhost; 10.0.0.0/8; };
	Line 58: //configure forward lookup zone
	zone "alpha.corp" IN {
	type master;
	file "/var/named/alpha.corp.db";
	allow-update { none; };
	};
	Line 63: //configure reverse lookup zone
	zone "0.0.0.10 in-addr.arpa" IN {
	type master;
	file "/var/named/10.0.0.db";
	allow-update { none; };
	};
	----------Save it----------
	[Explaination]
	• alpha.corp – Domain name
	• master – Primary DNS
	• fwd.alpha.corp.db – Forward lookup file
	• allow-update – Since this is the primary DNS, it should be none
	• 0.0.10.in-addr.arpa – Reverse lookup name
	• 10.0.0.db -- reverse lookup file
	---------- 	

Create forward lookup Zone Files	#vim /var/named/alpha.corp.db @ IN SOA ns1.alpha.local. root.alpha.local. (1001 ;Serial 3H ;Refresh 15M ;Retry 1W ;Expire 1D ;Minimum TTL) ;Name Server Information @ IN NS ns1.alpha.local. ;IP address of Name Server ns1 IN A 10.10.10.10

	;Mail exchanger alpha.local. IN MX 10 mail.alpha.local. ;A - Record HostName To IP Address www IN A 10.10.10.10 mail IN A 10.10.10.11 ;CNAME record ftp IN CNAME www.alpha.local. --- A – A record NS – Name Server MX – Mail for Exchange CNAME – Canonical Name ---
Create reverse lookup Zone Files	#vim /var/named/10.0.0.db @ IN SOA ns1.alpha.local. root.alpha.local. (1001 ;Serial 3H ;Refresh 15M ;Retry 1W ;Expire 1D ;Minimum TTL) ;Name Server Information @ IN NS ns1.alpha.local. ;Reverse lookup for Name Server 10 IN PTR ns1.alpha.local. ;PTR Record IP address to HostName 10 IN PTR www.alpha.local. 11 IN PTR mail.alpha.local. --- PTR – Pointer SOA – Start of Authority ---
restart bind service	#systemctl restart named
Enable it on system startup	systemctl enable named
Add entry in firewall	#firewall-cmd --permanent --add-port=53/udp #firewall-cmd --reload

On Client system(s).

client machine and add a DNS server ip address in **/etc/resolv.conf**	#vim /etc/resolv.conf nameserver 10.10.10.10
Add DNS entry	#vim /etc/sysconfig/network-scripts/ifcfg-eXX file. DNS1=192.168.0.10
verify the forward lookup.	#dig www.alpha.corp
Confirm the reverse	#dig -x 10.10.10.10

lookup.	

URL: https://www.itzgeek.com/how-tos/linux/centos-how-tos/configure-dns-bind-server-on-centos-7-rhel-7.html

DF command

Monday, January 14, 2019 3:20 PM

- Built-in utility for Linux systems
- df = "disk Filesystem"
- Gives detailed format of file system info.
- Different uses are:

Command	Meaning
#df	Display blocks, disk space, mount point info.
#df -a	Display all file system info
#df -h	Display human readable info.
#df -hT /home	Display info of "/home" directory
#df -k	Display info in Bytes size
#df -m	Display info in MB
#df -h	Display info in GB
#df -i	Display file system inodes
#df -T	Display file system type
#df -t xfs	Display specific file system
#df -x ext3	Display all info excluding "ext3" file system

dpkg

Friday, June 7, 2019 11:22 AM

- **To List all installed packages:**
  #dpkg -l
- **To list file's package:**
  #dpkg -S /path/to/file
- **To list all the packges:**
  #dpgk -L <package-name>
- **Install packages:**
  #dpkg -I packages.deb

DU command

Monday, January 14, 2019 3:27 PM

- du = Disk usage
- Different usages:

Command	details
#du /home/jeetu	Disk usage of /home/jeetu
#du -h /home/jeetu	Disk usage in human readable format
#du -sh /home/jeetu	grand total disk usage
#du -a /home/jeetu	displays the disk usage of all the files and directories
#du -ah /home/jeetu	a=all, h=human readable
#du -k /home/jeetu	k=Kilobyte
#du -mh /home/jeetu	m=MB
#du -ch /home/jeetu	“-c” flag provides a grand total usage disk space at the last line.
#du -ah --exclude="*.txt" /home/jeetu	Excludes "txt" files
#du -ha --time /home/jeetu	The disk usage based on modification of time

https://linuxize.com/post/du-command-in-linux/

File permissions

Thursday, February 7, 2019 9:37 PM

https://www.guru99.com/file-permissions.html

File Lock

Thursday, February 7, 2019 9:48 PM

https://access.redhat.com/documentation/en-US/Red_Hat_Storage/2.0/html/Administration_Guide/ch21s02.html

https://gavv.github.io/articles/file-locks/

https://dmorgan.info/posts/linux-lock-files/ (example for locking a file)

FTP Server

Saturday, March 3, 2018 9:21 AM

Packages Required	vsftpd
Config Files	/etc/vsftpd/vsftpd.conf
Port Numbers	21
Service Name	# systemctl enable vsftpd.service # systemctl start vsftpd.service # systemctl stop vsftpd.service # systemctl restart vsftpd.service
Commands	# systemctl start vsftpd # systemctl enable vsftpd
Firewall rules	# firewall-cmd --zone=public --permanent --add-port=21/tcp # firewall-cmd --zone=public --permanent --add-service=ftp # firewall-cmd --reload

FTP server configuration (rhel_link)

Install ftp
	#yum install -y vsftpd ftp

configure vsftpd
	#vim /etc/vsftpd/vsftpd.conf
	modify following lines
		anonymous_enable=NO
		ascii_upload_enable=YES
		ascii_download_enable=YES
		ftpd_banner=Welcome-----
			add in the last line
				use_localtime=YES

enable & start vsftpd service
	#systemctl enable vsftpd
	#systemctl start vsftpd

firewall service
	#firewall-cmd --permanent --add-port=21/tcp
	#firewall-cmd --permanent --add-service=ftp
	#firewall-cmd --reload

update SELinux
	#setsebool -P ftp_home_dir_on

create FTP user
	#useradd ftpUser
	#passwd ftpUser

to access ftp
	#ftp <ftp_server_ip>

CONNECTING FTP USING FILEZILLA
#yum install filezilla -y
#filezilla
use ftp_server_ip,
ftp_user_name,
ftp_user_passwd
ftp_port_number

==========

Server Side

#yum install vsftpd

#systemctl enable/start/status vsftpd

#useradd ftpuser; passwd ftpuser

#vim /etc/vsftpd/vsftpd.conf

Edit anonymous enable = NO & chroot local user = yes

#ftp <client IP>

ftp> client-ip-username: ftpuser

ftp> mkdir haha

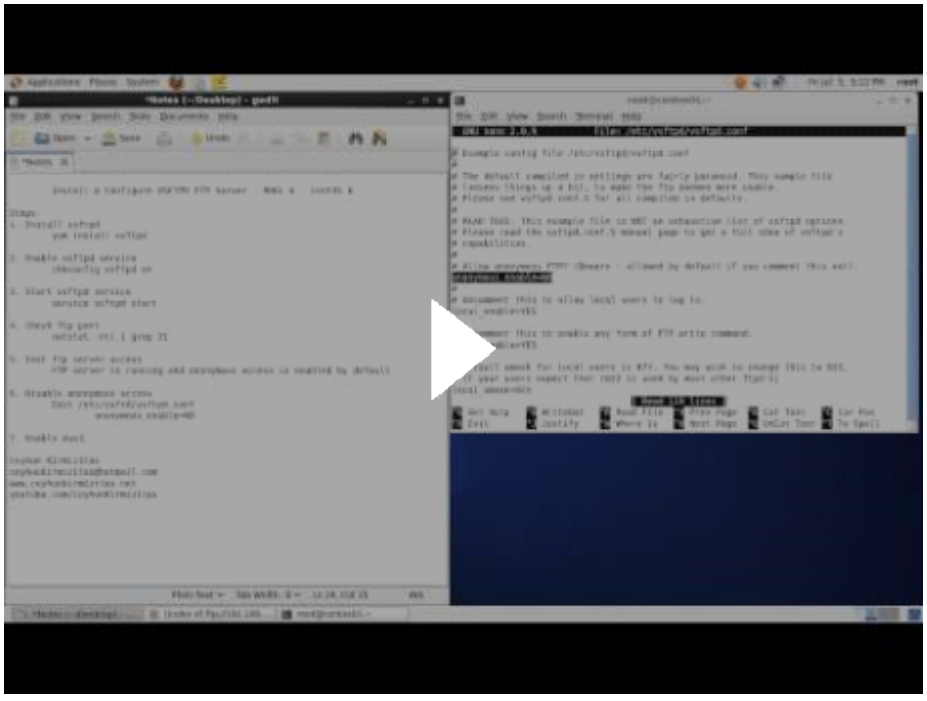
Client side:

#yum install ftp* -y

Edit anonymous enable = NO & chroot local user = yes

Ls; #vim /etc/vsftpd/vsftpd.conf //check config

https://www.linuxnix.com/7-steps-to-install-and-configure-ftp-server/
Install and Configure VSFTPD FTP Server - RHEL 6 - CentOS 6



How to Change FTP Port in Linux

Ftp server - 2 (verified)

Wednesday, December 11, 2019 12:47 PM

Server System:
IP: **192.168.10.10**
Hostname: **server**

Install ftp packages:

#yum install -y ftp*

Create some users for FTP access
#useradd ftpuser
#passwd ftpuser

Client System:
IP: **192.168.10.11**
Hostname: **client**
Install ftp packages:

#yum install -y ftp*

Access ftp from client to server:

#ftp server

```
[root@client ~]# ftp server
Connected to server (192.168.10.10).
220 (vsFTPd 3.0.2)
Name (server:root): ftpuser
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
```

Checking the commands to be access:

```
ftp> help
Commands may be abbreviated.  Commands are:

!               debug           mdir            sendport        site
$               dir             mget            put             size
account         disconnect      mkdir           pwd             status
append          exit            mls             quit            struct
ascii           form            mode            quote           system
bell            get             modtime         recv            sunique
binary          glob            mput            reget           tenex
bye             hash            newer           rstatus         tick
case            help            nmap            rhelp           trace
cd              idle            nlist           rename          type
cdup            image           ntrans          reset           user
chmod           lcd             open            restart         umask
close           ls              prompt          rmdir           verbose
cr              macdef          passive         runique         ?
```

Simply accessing FTP using URL:

on server:

Install required package:
#yum install -y vsftpd

enable, start, status vsftpd service:
#systemctl enable vsftpd
#systemctl start vsftpd
#systemctl status vsftpd

on client

#yum install -y ftp
#firefox ftp://10.10.10.10

File Structure

Monday, December 3, 2018 12:10 PM

UNIX / LINUX file structure

/ -> nameless root, root dir for entire file system.

/bin	essential user command binaries
/boot	boot loader files
/dev	device files
/etc	host specific files
/home	user home dir
/lib	essential shared lib & kernel module
/media	mount point for removable media
/mnt	mount point for temporary files
/opt	add-on application software package
/sbin	system binaries
/srv	data for services provided by this system
/tmp	temp files
/usr	(multi)-user utilities & applpcations (bin, include)
/var	variable files
/root	home dir for root user
/proc	virtual file system documenting kernel & process status as text files

File System Recovery

Monday, December 17, 2018 12:34 PM

[Linux Booting Troubleshooting 1 - Basic System Recovery]

MBR got corrupted

1. Attach ISO, reboot/start Linux into "troubleshooting" mode.
2. Select "**RESCUE A RHEL 7 SYSTEM**".
3. In the dialog box, select 1/continue.
4. Your system will be mounted at "/mnt/sysimage".
5. Once logged-in, execute below command:
    1. Sh-4.2# chroot /mnt/sysimage
6. List all the drives available
    1. Bash-4.2# ls /dev/sd*
7. Identify GRUB command
    1. Bash-4.2# /sbin/grub2-install /dev/sda
8. Exit
9. Reboot/init 6

File Compression and Archiving

Tuesday, December 25, 2018 7:36 PM

- It is useful to store a group of files in one file for easy backup, for transfer to another directory, or for transfer to another computer.
- It is also useful to compress large files; compressed files take up less disk space and download faster via the Internet.
- It is important to understand the distinction between an archive file and a compressed file.
- An archive file is a collection of files and directories stored in one file. The archive file is not compressed — it uses the same amount of disk space as all the individual files and directories combined.
- A compressed file is a collection of files and directories that are stored in one file and stored in a way that uses less disk space than all the individual files and directories combined.
- **File Roller** can compress, decompress, and archive files in common Unix and Linux formats.
    - Application --> Accessories --> Archive Manager

Compressing Files at the Shell Prompt

Compression Tool	File Extension	Decompression Tool
bzip2	.bz2	bunzip2
gzip	.gz	gunzip
zip	.zip	unzip

TAR:

- To create an archive
    - #tar -cvf <archive-name>.tar file1 file2 fil3 … filen //c=create, v=verbose, f=file
- To un-compress tar
    - #tar -xvf <archive-name>.tar //x=eXtract
- To display the content of tar
    - #tar -tvf <archive-name>.tar //t=conten't'

GZIP

- To create a GZIP compressed file
    - #tar -cvzf <archive-name>.tar.gz <list-of-files>
    - #check the file size (ls -l command)
- To uncompress GZIP
    - #tar -xvf <archive-name>.tar.gz

BZIP2

- To create a BZIP2 compressed file (compressing power is more than GZIP)
    - #tar -cvjf <tar-name.bz2> <d1.tar> OR
    - #bzip2 -z <existing-tar-file>.bz2

- To decompress BZIP2 file
    - # tar -xvf <archive-name>.tar.bz2

Swap space Increase - verified(R6)

Tuesday, October 8, 2019 11:00 PM

In this case if (input file) is /dev/zero and of (output file) is the destination where we want to save the file.
Basically, we are going to create 1GB file that will consist of zeros only and use that as a swap later.

```
# dd if=/dev/zero of=/swapfile bs=1M count=1024
```

Next, we need to set up file so it is can be used as swap space. We will use mkswap command for that:

```
# mkswap /swapfile
```

The swap file must be owned by root and have proper permissions:

```
#chown root:root /swapfile
#chmod 0600 /swapfile
```

Once this is done, the only thing left is to activate and start using the swap space.

```
# swapon /swapfile
```

That's it. We got another 1GB of swap space. If we wont to keep the space active after system reboot, we need to add an entry into /etc/fstab file.

```
/swapfile swap swap defaults 0 0
```

To check the swap space/size

```
# free -h
#cat /proc/swaps
```

Swap space increase verified (R6) - 2nd Proirity

Wednesday, October 9, 2019 11:42 AM

Example of Swap Space Creation

Create a new partition and change the type to 82:

```
[root@serverX ~]# fdisk /dev/vda
Command (m for help): n
First sector (12539904-12582911, default 12539904):  Enter
Using default value 12539904
Last sector, +sectors or +size{K,M,G} (12539904-12582911, default 12582911):  Enter
Using default value 12582911

Command (m for help): t
Partition number (1-6): 6
Hex code (type L to list codes): 82
Changed system type of partition 6 to 82 (Linux swap / Solaris)

Command (m for help): w

[root@serverX ~]# reboot
```

Write the swap signature to the device and find the UUID:

```
[root@serverX ~]# mkswap /dev/vda6
[root@serverX ~]# blkid /dev/vda6
/dev/vda6: UUID="4903c440-ffcb-4404-bc09-505c79c7a412" TYPE="swap"
```

Add an entry to **/etc/fstab**:

```
UUID="4903c440-ffcb-4404-bc09-505c79c7a412"  swap  swap  defaults  0 0
```

Activate the swap space, verify it is available and then deactivate the swap space:

```
[root@serverX ~]# swapon -a
swapon -s
/dev/dm-0                               partition       557048  0       -1
/dev/vda6                               partition       21496   0       -2
[root@serverX ~]# swapoff /dev/vda6
```

Sizing total swap space should really be based on the memory workload on the system, not the total amount of physical memory present. However, the table below provides some rough rules of thumb for sizing swap space. For more detailed guidance on sizing swap space, see the Knowledgebase article in the references.

System RAM	*Recommended Minimum Swap Space*
up to 4 GB	at least 2 GB
4 GB to 16 GB	at least 4 GB
16 GB to 64 GB	at least 8 GB
64 GB to 256 GB	at least 16 GB

Table 6.1. Basic Guidance on Swap Space Sizing

dd command in Linux:

28 October 2020 09:54 AM

- dd is a command-line utility for Unix and Unix-like operating systems whose primary purpose is to convert and copy files.
- dd can read and/or write from/to these files, provided that function is implemented in their respective drivers
- dd can be used for tasks such as backing up the boot sector of a hard drive.

Some practical examples on dd command :

1. To backup the entire harddisk:
   # dd if = /dev/sda of = /dev/sdb
   - “if” represents inputfile, and “of” represents output file. So the exact copy of /dev/sda will be available in /dev/sdb.
2. To backup a Partition:
   # dd if=/dev/hda1 of=~/partition.img

3. To create an image of a Hard Disk
   # dd if = /dev/hda of = ~/hdadisk.img

4. To restore using the Hard Disk Image:
   # dd if = hdadisk.img of = /dev/hdb

ISCSI Storage

Wednesday, December 12, 2018 3:57 PM

Requirement

1. Server / target server (IP: 192.168.10.10)
2. Client / iscsi initiator (IP: 192.168.10.11)

On server:

- Install ISCSI target package
  - #yum install -y targetcli
  - #systemctl enable/start/status target.service
- Add the entries in the firewalld (OR disable firewall)
  - #firewall-cmd --permanent --add-port=3260/tcp
  - #firewall-cmd --reload
  - #firewall-cmd --list-port
- Now create a partition (either standard or LVM)
  - Create a 15GB LVM
  - #pvcreate, #vgcreate, #lvcreate
  - Assume /dev/iscsi_vg/iscsi_lv //no mkfs.xfs
- Now make the entries using "targetcli" command:
  - #targetcli
  - /> ls
  - /> /backstores/block create iscsi_disk /dev/iscsi_vg/iscsi_lv
  - .. to be continued ...
- Switch to Client system
  - Install iscsi initiator utility package.
    - #yum install -y iscsi-initiator-utils
  - Enable, start & enable ISCSI service
    - #systemctl enable iscsi.service
    - Verify its running.
  - Open file /etc/iscsi/initiatorname.iscsi & copy IQN ID
    - #cat /etc/iscsi/initiatorname.iscsi
    - Example: InitiatorName=iqn.2018-06.com.server:client2
- Switch back to target server:
  - /> /iscsi create <PASTE-IQN-ID>
- Create ACL for specific system access
  - /> /iscsi/<iqn.2018-06.com.server:client2>/tgp1/acls create iqn.2018-06.com.server:client2
- Create a LUN for backstores
  - /> /iscsi/<iqn.2018-06.com.server:client2>/tgp1/luns create /backstores/block/iscsi_disk
- Create a network portal, so that client can access ISCSI storage.
  - /> /iscsi/ iqn.2018-06.com.server:client2/tgp1/portals create <Client-IP>
- List all details
  - /> ls
  - />Exit
- Restart target service.
  - #systemctl restart target.service

On Client side:

Verify iscsi iscsi service is running or not

- #systemctl enable/start/status iscsi

Discover the target

- #iscsiadm -m discovery -t st -p 192.168.10.10 –discover
- -m = mode
- -t = type
- St = send target
- 192.168.10.10 = target server IP

Expected output:

192.168.10.10:3260,1 iqn.2018-06.com.server:client2

In case it throws error:

```
[root@client ~]# iscsiadm -m discovery -t st 192.168.10.10 -discover
iscsiadm: Please specify portal as <ipaddr>[:<ipport>]
[root@client ~]# iscsiadm -m discovery -t st -p 192.168.10.10 -discover
iscsiadm: cannot make connection to 192.168.10.10: Connection refused
iscsiadm: cannot make connection to 192.168.10.10: Connection refused
iscsiadm: cannot make connection to 192.168.10.10: Connection refused
iscsiadm: cannot make connection to 192.168.10.10: Connection refused
iscsiadm: cannot make connection to 192.168.10.10: Connection refused
iscsiadm: cannot make connection to 192.168.10.10: Connection refused
iscsiadm: connection login retries (reopen_max) 5 exceeded
iscsiadm: Could not perform SendTargets discovery: encountered connection f
ailure
```

Troubleshooting steps:

1. Go to target server
2. Open targetcli
3. /> go to network portal
4. /> /iscsi/<iqn.1994-05.com.redhat:33da29bc587a>/tpg1/portals
5. Delete any existing entry & make an entry for 0.0.0.0:3260
6. /> delete <IP> <port>
7. /> create <0.0.0.0> <3260>
8. /> exit
9. Restart target service & switch to client

After discovery, login to ISCSI storage:

iscsiadm -m node -T iqn.2018-06.com.server:client1.disk1 -l

Expected output:

```
Logging in to [iface: default, target: iqn.2018-06.com.server:client1.disk1, portal:
192.168.1.16,3260] (multiple)
Login to [iface: default, target: iqn.2018-06.com.server:client1.disk1, portal: 192.168.1.16,3260]
successful.
```

Verify the disk is attached or not:

- #dmesg | tail
- fdisk -l /dev/sdb

now create a file system & mount it here.

- #Pvcreate /dev/sdb
- #Vgcreate initiator_vg /dev/sdb
- #lvcreate -L 14G -n initiator_lv initiator_vg
- #lvdisplay
  - /dev/ initiator_vg/ initiator_lv

- #mkdir /iscsi
- #mount -t xfs /dev/ initiator_vg/ initiator_lv /iscsi
- #df -hP /iscsi

  Logout from the iscsi storage:

  #iscsiadm -m node -T iqn.2018-06.com.server:client1.disk1 -u

- Server: https://www.storagetutorials.com/share-storage-iscsi-target-rhel-centos/
- Client: https://www.storagetutorials.com/how-configure-iscsi-initiator-rhel-7-centos/

IPTABLES

Saturday, December 22, 2018 10:47 AM

- Iptables is a rule based firewall and it is pre-installed on most of Linux operating system.
- IPTables is a front-end tool to talk to the kernel and decides the packets to filter.
- IPTables main files are:
  - /etc/init.d/iptables – init script to start|stop|restart and save rulesets.
  - /etc/sysconfig/iptables – where Rulesets are saved.
  - /sbin/iptables – binary.
- How to start, stop and restart Iptabe Firewall.
  - # /etc/init.d/iptables start
  - # /etc/init.d/iptables stop
  - # /etc/init.d/iptables restart
- To start iptables at startup
  - chkconfig --level 345 iptables on
- To save the IPTables service
  - service iptables save
- To list Iptables:
  - iptables -n -L -v --line-numbers // n=numeric format, L=list, v=verbose
- To flush the list:
  - #iptables -F
- To delete specific entry from the Iptables list:
  - #iptables -D INPUT <number-check-from-"iptables -n -L -v --line-numbers">
- To block IP od DOMAIN (facebook.com)
  - #ping facebook.com
  - #whois <FB-IP> | grep CIDR
  - #iptables -A OUTPUT -p tcp -d <CIDR-of-facebook> -j DROP

https://www.cyberciti.biz/faq/rhel-fedorta-linux-iptables-firewall-configuration-tutorial/

IPTABLES-2

Saturday, December 7, 2019 11:59 AM

* iptables contains multiple tables (set of chains)
* tables contains chains (in-built or user defined)
* chain contains rules
* rules are for packets

iptables -> tables -> chains -> rules

IPTABLE

- iptables 4 tables
- filter table
    - INPUT chain = incoming to firewall
    - OUTPUT chain = outgoing from firewall
    - FORWARD chain = packet from other NIC to local system
- NAT table
    - pre-routing chain = Alters packets before routing (i.e Packet translation happens immediately after the packet comes to the system)
    - postrouting chain = Alters packets after routing. (i.e Packet translation happens when the packets are leaving the system.)
    - output chain = NAT for locally generated packets
- Mangle table
    - Iptables's Mangle table is for specialized packet alteration.
    - PREROUTING chain
    - OUTPUT chain
    - FORWARD chain
    - INPUT chain
    - POSTROUTING chain
- Raw table
    - Iptable's Raw table is for configuration exemptions.

IPTABLES RULES

- Rules contain a criteria and a target.
- If the criteria is matched, it goes to the rules 11111 in the target (or) executes the special values mentioned in the target.criteria is not matached, it moves on to the next rule.

Target Values

ACCEPT – Firewall will accept the packet.
DROP – Firewall will drop the packet.
QUEUE – Firewall will pass the packet to the userspace.
RETURN – Firewall will stop executing the next set of rules in the current chain for this packet.

To list iptables filters:	# iptables -t filter --list
to view the mangle table.	# iptables -t mangle --list
to view the nat table.	# iptables -t nat --list

TO list current rules	#iptables -L

Blocking an IP address	#iptables -A INPUT -s <IP-Address> -j DROP
Blocking EMAILs from an IP range	#iptables -A INPUT -s <192.168.10.0/24> -p tcp --destination-port 25 -j DROP
Accepting emails from specific IP	#iptables -A INPUT -s 192.168.10.10 -j ACCEPT
Accepting emails from specific IP (making entry to TOP)	#iptables -I INPUT -s 192.168.10.10 -j ACCEPT
To save Iptables for IPv4	#iptables -save > /etc/iptables/rules.v4
To save iptables for IPv6	#iptables -save > /etc/iptables/rules.v6

Job Schedulers

Monday, December 3, 2018 12:06 PM

AT & CRONTAB jobs

AT jobs
The at command schedules a command to be run once at a particular time.
ex:
at 23:40
at > history > hist.txt
to check list of all the 'at' jobs
ex:
#atq
to delete any 'at' jobs
ex:
#atrm <jobNumber>
CRONTAB
#vim /etc/crontab
The crontab is a list of commands that you want to run on a regular schedule
ex:
#vim /etc/cron.allow
<minutes> <hours> <day(1-31)> <months(1-12)> <days(num/name)>
ex:
30 11 24 6 * shell.sh
crontab –l //list cron jobs
crontab -e // create new job
crontab -r //remove job

Task: Create a Cron job to automate the clearing of a folder during the working hours of Capgemini IMS Trainees (08.00AM to 05.00 PM on weekdays from mon to sat).

Solution: *** 8-9 * * 0-6 rm -rf /root/Desktop/restore/***

journalctl

Monday, June 17, 2019 9:31 PM

#journalctl	List all logs
#journalctl -r	List in reverse order
#journalctl -f	Live logs on terminal
#journalctl --since "2019-06-01 14:10:10"	List logs from given date
#journalctl --until "2019-06-01 14:10:10"	List up to specific time
#journalctl --since "2018-08-30 14:10:10" --until "2018-09-02 12:05:50"	List logs between dates
#journalctl -u sshd.service	Shows the logs of systemd service
#journalctl -k	Lists kernal logs
#journalctl -n 3	List latest 3 log
#journalctl -p crit	List critical. (crit, debug, info, notice, warning, err, alert, emerg)
#journalctl -p crit -n 3	Listing latest 3 critical messages.

GREP

Wednesday, January 2, 2019 1:32 PM

- General regular expression

To search a word ends with "cat" in the in-built dictionary

- #grep 'cat$' /usr/share/dict/words

List all processes by 'jeetu' user

- #ps aux | grep '^jeetu'

List specific word in a file:

- #grep -I 'word' <file-name>

HTOP

Friday, February 8, 2019 4:19 PM

https://www.tecmint.com/install-htop-linux-process-monitoring-for-rhel-centos-fedora/

Kickstart

Monday, December 3, 2018 12:04 PM

How to configure KICKSTART - RHEL 7

---

1. disable SELinux & firewall
2. create directory
   #mkdir /var/www/html/kickstart

---copy rhel7 dvd rom contents in it.---

   #cp /run/media/root/RHEL7...(cd name) /var/www/html/kickstart

OR

   [root@/run/media/root/RHEL7...(cd name)] #cp -var * /var/www/html/kickstart

3. configuring DHCP server
   a. DHCP - system should hv a static ip
   b. Install dhcp server
   c. Configuration of DHCP file

****

   #vim /etc/dhcp/dhcpd.conf
   Allow booting;
   Allow bootp;
   authoritative;
   subnet 192.168.0.0 netmask 255.255.255.0{
   default-lease-time 21600;
   max-lease-time 43200;
   range dynamic-bootp 192.168.0.101 192.168.0.200;
   filename "pxelinux.0";
   next-server 192.168.0.1;
   }

****

d. Enabling DHCP
   #systemctl enable dhcpd // enabling dhcp
   #systemctl start dhcpd // starting dhcp
   #systemctl status dhcpd checking status of dhcp

4. Create an answer file "ks.cfg"
   install - #yum install system-config-kickstart
   & save it in "/var/www/html/kickstart" directory
   now run system-config-kickstart
   #system-config-kickstart
   GUI will be opened
   a. select default on page 1 - basic configuration
   b. install method -
   http
   server - 192.168.0.1
   directory - kickstart
   c. bootloader
   install new boot loader
   d. partition info
   donot clean MBR
   remove all exss..
   layout...
   e. network configuration
   add nw device
   eno16777736

f. firewall
disable all
g. ...
save it in
/var/www/html/kickstart path

5. Configure Apache web server to export "ks.cfg" file to target system
#yum install httpd //install apache server
#vim /etc/httpd/conf/httpd.conf //apache conf file
:set nu
at line 86
append ServerAdmin root@192.168.0.1
at line 96
add ServerAdmin 192.168.0.1:80
at line 164
cursor at 'i'ndex.html, add ks.cfg in between 'DirectoryIndex' & 'index.html'
DirectoryIndex ks.cfg index.html
#systemctl enable httpd
#systemctl start httpd
#systemctl status httpd
application -> firefox web
type in url - http://192.168.0.1/kickstart
ks.cfg page will appear
minimize browser
& install remaining packages, which are left at "step 4, point g"
#vim /root/anaconda-ks.cfg
:set nu
:35,76 w >> /var/www/html/kickstart/ks.cfg
//copy packages to ks.cfg file from line 35 (start package line) to 76 (package ending line)
refresh the browser, all the packages will be list at the end

6. Configure TFTP server
#yum install tftp-server
#mkdir /var/lib/tftpboot/pxelinux.cfg
#cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot
#systemctl enable tftp.socket
#systemctl start tftp.socket
#systemctl status tftp.socket
// tftp.socket is renamed in RHEL7, previsouly in RHEL6 was tftp
7. configure PXE boot environment
#mkdir /var/pxe
#ln -s /var/www/html/kickstart /var/pxe/kickstart
#mkdir /var/lib/tftpboot/kickstart
copy vmlinuz & initrd.img to other dir
#cp /var/pxe/kickstart/images/pxeboot/vmlinuz /var/lib/tftpboot/kickstart
#cp /var/pxe/kickstart/images/pxeboot/initrd.img /var/lib/tftpboot/kickstart
8. Setup banner for target systems
****
***
#vim /var/lib/tftpboot/pxelinux.cfg/default
timeout 1000 //sec
default menu.c32
menu title == Boot Menu ==
label 1
menu label ^ 1) Rhel Installation
kernel kickstart/vmlinuz
append initrd=kickstart/initrd.img ks=http://192.168.0.1/kickstart

****
***
#cp /usr/share/syslinux/menu.c32 /var/lib/tftpboot
#iptables -F //flushing IPv4
#ip6tables -F //flushing IPv6

Kickstart R6

Tuesday, December 25, 2018 9:37 PM

yum install -y vsftpd tftp-server syslinux system-config-kickstart

enable, start vsftpd
#chkconfig vsftpd on
#service vsftpd start

config DHCP

config tftp-server
default tftp boot dir = /var/lib/tftpboot
vim /etc/xinetd.d/tftp
    disable = no

copy all the content of /media/R6../isolinux to tftpboot dir
cp /media/R6/isolinux /var/lib/tftpboot

go to /var/lib/tftpboot
& create a dir pxelinux.cfg

copy
cp /var/lib/tftpboot/isolinux.cfg /var/lib/tftpboot/pxelinux.cfg/default

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

restart
    vsftpd
    firewall
    xinetd

Switch to client
attach DVD to client
boot from DVD & at boot selection page hit tab
thrn append
inst.ks=ftp://192.168.10.11/pub/ks.cfg

Script Examples

18 June 2022 16:31

Find file(s) with more than 1GB & send/transfer it to another backup server.

Script:

```
#listing files more than 1GB
clear
echo "Files greater than 1GB & size"
#find /tmp -type f -size +1G 2> /dev/null # | ls -lh $i | awk '{print $5,$9}'
files=`find /tmp -type f -size +100M -exec du -sh {} \; 2> /dev/null`
#files=`find /tmp -type f -size +100M -exec du -sh {} \; 2> /dev/null` | awk '{print $1,$2}'`
echo $files | awk '{print $2}' 2> /dev/null
echo "Sending files to backup server"
scp $files bkpuser@10.0.0.5:/tmp/
echo "files transferred."
```

Output:

```
Files greater than 1GB & size
/tmp/usr.tar
Sending files to backup server
2.0G: No such file or directory
usr.tar                                                    100% 1947MB  59.4MB/s
00:32
files transferred.
```

Create shell script to automate the file transfer on daily basis and removal of old files with timestamp.

Script:

```
clear
#archive files
date_now=$(date "+%F-%H-%M-%S")
tar -cf /tmp/$date_now.tar /tmp/* 2> /dev/null
echo "files archived with name $date_now.tar"

#scp archived file to another server.
scp /tmp/$date_now.tar bkpuser@10.0.0.5:/tmp/ 2> /dev/null 1> /dev/null
echo "$date_now file copied to another bkpuser@10.0.0.5"
```

Output:

```
files archived with name 2022-06-18-11-45-46.tar
2022-06-18-11-45-46 file copied to another bkpuser@10.0.0.5
```

LDAP Config

Saturday, March 3, 2018 9:21 AM

https://www.youtube.com/watch?v=42tFcFFkk-w

Packages Required	# yum install openldap openldap-clients openldap-servers nss-pam-ldapd
Config Files	/var/lib/ldap; /etc/openldap/ldap.conf; /etc/openldap/slapd.d/
Port Numbers	TCP 389 //default TCP 636 //LDAP with TLS/SSL
Service	# systemctl enable slapd.service # systemctl start slapd.service # systemctl disable slapd.service # systemctl restart slapd.service # systemctl stop slapd.service
Commands	Ldapadd; ldapmodify; ldapsearch; //reference
Firewall rules	# setsebool -P allow_ypbind=0 authlogin_nsswitch_use_ldap=0 # firewall-cmd --add-service=ldap

OpenLDAP

Total hosts: 2
1. LDAP server
   192.168.2.61
   server1
2. LDAP client
   192.168.2.62
   client1

Reqreu
---
1. rhel /centOS
2. both should ping
3. yum enabled
4. iptables stopped

Steps:

At server end
1. install LDAP packages
2. create LDAP passwd
3. edit OpenLDAP config
3b. provide the monitoring priviledges
4. enable & start SLAPD config
5. config the LDAP DB
6. create self-signed certificate
7. create base object in OpenLDAP
8. generate base.ldif file
9. create local users
10. import user in the LDAP DB
11. test the config

Install packages
#yum install -y *openldap* migrationtools

LDAP PWD
#slappasswd
>
>

#cd /etc/openldap/slap.d/cn\=config
#ls -lrt
#ls

select hdb.ldif file for databse edit
& edit these lines
olcSuffix: dc=alpha,dc=local
& at last add
olcRootPW: <slappasswd>
olcTLSCertificateFile: /etc/pki/tls/certs/xxx.pem
olcTLSCertificateKeyFile: /etc/pki/tls/certs/xxxkey.pem
save & quit

Monitoring priviledges
#vim olcDatabase={1}monitor.ldif
edit olcAccess
>cn=Manager,dc=alpha,dc=local

#slaptest -u

enable & start slapd
#systemctl enable slapd
#systemctl start slapd

configure DB for LDAP
#cp -rf /usr/share/openldap-servers/DB_CONFIG.example /var/lib/ldap/DB_CONFIG

edit ldap schemas
#ldapadd -Y EXTERNAL -H ldapi:// -f /etc/openldap/schema/cosine.ldif

#ldapadd -Y EXTERNAL -H ldapi:// -f /etc/openldap/schema/nis.ldif

#ldapadd -Y EXTERNAL -H ldapi:// -f /etc/openldap/schema/inetorgperson.ldif

create self sign certificate
#openssl req -new -x509 -node -out /etc/pki/tls/certs/xxx.pem -keyout
/etc/pki/tls/certs/xxxkey.pem

ll /etc/pki/tls/certs/

create base object - migration tolls are req
#ll /usr/share/migrationtools
#cd /usr/share/migrationtools
#vim migrate_common.ph
line 71: change domain to ur domain &
default_base
line 90: sxt_schema: 1

create base ldap file

Linux Monitoring

Saturday, March 3, 2018 9:22 AM

Linux Monitoring tools
---
1. TOP
    - Table Of Process
2. vmstat
    - Virtual Machine Statistics
3. tcpdump
    - network packet analyser
    - #tcpdump -i <eth0>
    - to check nic card name: #cat /proc/net/dev
4. iotop
    - similar to top
    - monitors & display real time disk i/o & processes.
5. iostat
    - collect & show system i/o storage device statistics.
    - often used to trace device performance for devices, local disks, remote disks(nfs)
6. iptraf-ng
    #iptraf-ng
    - real time lan traffic monitoring
7. psacct / acct
    - monitors each users activity on the system
8. Monit
    - opensource, web based process supervision tool
    - monitors apache, mysqlm mail, ftp, ssh, nginx
9. monitorix
    - monitors system load average, usage, memory allocation, disk drivers health

Logs

Monday, December 3, 2018 11:58 AM

:- **ANALYZING & STORING SYSTEM LOGS**

Root log file
/var/log
configure file
" /etc/rsyslog.conf "
this file contains 2 columns
1. when msg/log is to be generated // SELECTOR
2. where to generate the log for storage // ACTION
Selector - <facility>.<severity Level>
Action - Location
Multiple selectors can be written in a single line using a ';'
:- or manual entry for any left user's /var/log/secure file
#logger -p authpriv.err "ERROR MSG"
:- MONITORING & MANAGING PROCESS

PROCESS STATUS
#ps
Process terminating
#kill
#kill -l //list kill signals
#kill -9 <processID> //killing any process forcefully
to set priority to process while executing it
#nice
to change running process priority
#renice

Log - 2

18 April 2020 07:13 PM

Logs are stored in **/var/log** directory

Systemd-journald daemon provides an improved mgmt that collects logs from kernel, early stage of boot, standard output & errors of daemons as they start up, run or syslog.

System log files

/var/log/message	- Most logs are stored here. - Logs related to: • Authentication • Email processing • Periodical jobs • Debugging-related.
/var/log/secure	Security & authentication related messages & errors
/var/log/maillog	Mail server related logs
/var/log/cron	Logs related to periodically executed tasks.
/var/log/boot.log	Messages related to start-up are logged here.

LUKS

Sunday, February 3, 2019 11:20 AM

LUKS

- Linux Unified Key Setup.
- Used for encrypt drives.

Steps LUKS partition:

Create a new partition	#fdisk /dev/sde
Update kernel for updated table	#partprobe

Format partition with LUKS to setup encryption	#cryptsetup luksFormat /dev/sde1
	Format = "F" capital

Setup password	Type any password

Open partition & make entry inside partition	#cryptsetup luksOpen /dev/sde1 <myluks>
	Open = "O" capital

Check the status	#cryptsetup status myluks
Creating file system for newly created partition (myluks)	#mkfs.ext4 /dev/mapper/myluks
Create a new directory	#mkdir /myluks
Mount it	#mount /dev/mapper/myluks /myluks
Check GUI disk utility for encrypted icon	
Check disk for LUKS encrypted entry.	#lsblk

Removing LUKS:

Unmount partition	#umount /myluks
Closing LUKS for the partition	#cryptsetup luksClose myluks
Removing LUKS keys	#cryptsetup luksRemoveKey /dev/sde1
Listing disks	#lsblk

https://www.cyberciti.biz/hardware/howto-linux-hard-disk-encryption-with-luks-cryptsetup-command/

LVM

Tuesday, December 25, 2018 10:57 PM

LVM snapshot: https://www.tecmint.com/take-snapshot-of-logical-volume-and-restore-in-lvm/ https://www.golinuxhub.com/2017/09/understanding-lvm-snapshots-create.html (working, verified)

1. #pvcreate /<disk-1> /<disk-2> /<disk-3>
2. #vgcreate -s 8m vg /<disk-1> /<disk-2> /<disk-3> //with PE=8m
3. #vgdisplay
4. #lvcreate -L <length> -n <name> vg
5. #lvdisplay

lvcreate -L 1GB -s -n tecmint_datas_snap /dev/vg_tecmint_extra/tecmint_datas

OR

lvcreate --size 1G --snapshot --name tecmint_datas_snap /dev/vg_tecmint_extra/tecmint_datas

Both the above commands does the same thing:
-s – Creates Snapshot
-n – Name for snapshot

LVM extent:
1. Attach a physical disk
2. Create a new partition (pv) with "fdisk"
3. Extending Volume Group: vgextend <vg_name> <new-pv>
4. Vgs
5. Pvscan
6. Vgdisplay, check "Free PE…"
7. Extend the LV, #lvextend -L +<PE/GB> /dev/vg1/lv1
8. xfs_grow /dev/vg1/lv1

Extending logical volume in RHEL 6 || 7 [VERIFIED]:

1. 1. create a simple LVM & mount it.	#lvm
2. attach a new raw disk & scan for new partition without reboot	# ls -l /sys/class/scsi_host # grep mpt /sys/class/scsi_host/host*/proc_name # echo "- - -" > /sys/class/scsi_host/host<>/scan
3. create a partition using fdisk (use 8e for LVM partition)	#fdsick </dev/sdx>
4. create physical volume using pvcreate	#pvcreate /dev/sdx1
5. extend the volume group	#vgcreate <vol_grp_name> <pv_name>
6. verify using vgs cmd.	# vgs
7. to extend LV, use "vgdisplay" to see the "FREE PE"	# vgdisplay --> grab "FREE PE" value.

8. extend the logical volume size	# lvextend -l +<FREE_PE_value> </dev/vg/lv>
9. Display	# lvs
10. update the mounted LVM partition	# resize2fs </dev/vg/lv> //RHEL 6 # xfs_grow </dev/vg/lv> //RHEL 7
11. Verify	# df -h

REDUCING logical volume in RHEL 6 [VERIFIED]:

1. Ensure LVM is mounted, verify size using(~15GB)	#df -h
2. Unmount LVM file system	#umount /lvm
3. Check file system for error	#e2fsck -f /dev/vg/lv
4. reduce the file system to desire capacity	#resize2fs /dev/vg/lv 10G
5. reduce LVM to the desired capacity	#lvreduce -L 10G /dev/vg/lv
6. Check file system for error	#e2fsck -f /dev/vg/lv
7. mount again	#mount -a
8. verify	#df -h

Kernel Modules

27 October 2020 10:03 PM

Kernel modules come in different flavours. They are as follows:

- ■■ Device driver: Facilitates communication with a hardware device.
- ■■ Filesystem driver: Required for filesystem IO.
- ■■ Network driver: Used to implement network protocols.
- ■■ System calls: Provides additional functions for adding/modifying system services.
- ■■ Executable loader: Allows additional executable formats to load

Viewing a /lib/modules directory	# ls -F /lib/modules

There are three handy programs that can help with modules:

- ■ dmesg displays the current kernel ring buffer.
- ■ lsmod shows brief module information.
- ■ modinfo provides detailed module data.

Using dmesg with grep to display module messages	# dmesg | grep -i driver
Employing lsmod to display module status	# lsmod
Using modinfo to display detailed module information	# modinfo bridge

Installing Kernel Modules: utilities that can help you load modules into the kernel. They are as follows:

- ■■ insmod - The insmod utility allows you to insert a single module into the Linux kernel
- ■■ modprobe - The modprobe command is easier to use than the insmod utility because you can denote modules by their module name
- ■■ depmod - the depmod utility scans the system, determines any needed modules, reviews the modules' dependencies, and updates the appropriate modules.dep file

Finding module:	#find /lib -name speedstep*
listing module:	#lsmod | grep speedstep-lib
inserting required module speedstep-lib (use output from find cmd)	#insmod <path-output-for-speedstep-lib>
re-inserting required module speedstep-lib (use output from find cmd)	#insmod -f <path-output-for-speedstep-lib>
removing module	#rmmod <path-output-for-speedstep-lib>
listing module information	#modinfo dm_mirror
Using the modprobe utility to remove a module and its dependencies	#modprobe -rv dm_mirror

MySQL in RHEL 6

Sunday, December 23, 2018 7:31 PM

Install MySQL packages

#yum install -y mysql-server

Enable, start & status of MySQL:

#chkconfig mysqld on
#service mysqld start
#service mysqld status

Provide password & other details:

#mysql_secure_installation

Login with Root user:

#mysql -u root -p

Create a new MySQL DB & user

Mysql > create database testdb;
Mysql > create user 'testuser'@'localhost' identified by 'password';
Mysql > grant all on testdb.* to 'testuser' identified by 'password';
Mysql > exit

Login with newly created user & using DB

#mysql -u testuser -p
Mysql > use testdb;
Mysql > create table customers (customer_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY, first_name TEXT, last_name TEXT);

Reference: https://www.linode.com/docs/databases/mysql/how-to-install-mysql-on-centos-6/

NFS Config

Saturday, March 3, 2018 9:23 AM

Packages Required	yum install nfs-utils nfs-utils-lib yum install portmap (not required with NFSv4)
Config Files	/etc/exports
Port Numbers	Port 111 (TCP and UDP) and Port 2049 (TCP and UDP) for the NFS server
Service	# /etc/init.d/portmap start # /etc/init.d/nfs start # chkconfig --level 35 portmap on # chkconfig --level 35 nfs on
Commands	/nfsshare 192.168.0.101(rw,sync,no_root_squash) //on SVR showmount -e 192.168.0.100 //on client mount -t nfs 192.168.0.100:/nfsshare /mnt/nfsshare //on client #systemctl restart nfs-config # systemctl restart nfs-server
Firewall rules	-A RH-Firewall-1-INPUT -s 192.168.1.0/24 -m state --state NEW -p udp --dport 111 -j ACCEPT -A RH-Firewall-1-INPUT -s 192.168.1.0/24 -m state --state NEW -p tcp --dport 111 -j ACCEPT -A RH-Firewall-1-INPUT -s 192.168.1.0/24 -m state --state NEW -p tcp --dport 2049 -j ACCEPT -A RH-Firewall-1-INPUT -s 192.168.1.0/24 -m state --state NEW -p tcp --dport 32803 -j ACCEPT -A RH-Firewall-1-INPUT -s 192.168.1.0/24 -m state --state NEW -p udp --dport 32769 -j ACCEPT -A RH-Firewall-1-INPUT -s 192.168.1.0/24 -m state --state NEW -p tcp --dport 892 -j ACCEPT -A RH-Firewall-1-INPUT -s 192.168.1.0/24 -m state --state NEW -p udp --dport 892 -j ACCEPT -A RH-Firewall-1-INPUT -s 192.168.1.0/24 -m state --state NEW -p tcp --dport 875 -j ACCEPT -A RH-Firewall-1-INPUT -s 192.168.1.0/24 -m state --state NEW -p udp --dport 875 -j ACCEPT -A RH-Firewall-1-INPUT -s 192.168.1.0/24 -m state --state NEW -p tcp --dport 662 -j ACCEPT -A RH-Firewall-1-INPUT -s 192.168.1.0/24 -m state --state NEW -p udp --dport 662 -j ACCEPT # service iptables restart

Network File System (RedHat)
---
1. NFS Server (svr), 192.168.10.10
2. NFS Client (cli), 192.168.10.11
3. same network

Used to share files between
Linux to Linux/UNIX flavors

svr = /project
|- files

Requirements for NFS or any Server
1. nfs package
2. nfs config file /etc/exports
3. services (start nfs, nfs lock, port map)

On Server M/C

IP = 192.168.10.10
1. create a dir /project & some files
2. install package nfs
rpm -qa nfs* / yum install -y nfs*
3. edit /etc/exports /by defaul

/project *(rw,sync)

- * means all systems or
- in place of *, write IP addr for specific user usage.
- or IP_addr/subnet 192.168.10.0/24
- if you want to give read only permissions write - 'ro'
- sync means that what whatever change are made will be reflected in the nfs, will done to all.
- rw means 'read & write' permissions

Now Start the following services
#service nfs start
#service nfslock start
#service portmap start

==========
ON CLIENT M/C

- IP = 192.168.10.11
- check the conectivity between the m/cs
- use cmd on cli machine
#showmount -e <nfs-server-ip-addr>
#showmount -e 192.168.10.10
showmount = shows the mounted dir in the nfs server
-e = shows exports list
---
if it's showing some error, install nfs packages
stop iptables, start services -> nfs, nfslock, rpcbind, portmap
---

- create an empty dir
#mkdir /nfs

- mount the dir
#mount 192.168.10.10:/project /nfs
#mount <nfs_ip>:<svr_mount_path> /<cli_mount_path>

- mounting in fstab file
#<what-to-mount> <where> nfs defaults 0 0
#<server_ip>:<nfs_path> <cli_path> nfs defaults 0 0
#192.168.10.10:/project /nfs nfs defaults 0 0

If you receive error:

```
[root@client ~]# showmount -e 192.168.10.10
clnt_create: RPC: Port mapper failure - Unable to
 receive: errno 113 (No route to host)
```

Make below entry in firewall in both server & client:
firewall-cmd --permanent --add-service=rpc-bind
firewall-cmd --permanent --add-service=mountd
firewall-cmd --permanent --add-port=2049/tcp
firewall-cmd --permanent --add-port=2049/udp
firewall-cmd --reload

On server system:

```
[root@server Desktop]# firewall-cmd --permanent --add-service=rpc-bind
success
[root@server Desktop]# firewall-cmd --permanent --add-service=mountd
success
[root@server Desktop]# firewall-cmd --permanent --add-port=2049/tcp
success
[root@server Desktop]# firewall-cmd --permanent --add-port=2049/udp
success
[root@server Desktop]# firewall-cmd --reload
success
[root@server Desktop]#
```

On client system:

```
[root@client ~]# showmount -e 192.168.10.10
Export list for 192.168.10.10:
/nfs1 192.168.10.11
[root@client ~]#
```

Telnet

Monday, November 11, 2019 1:55 PM

What Is Telnet?
Telnet is a network protocol which is used to connect to remote computers over TCP/IP network. Once you establish a connection to the remote computer, it becomes a virtual terminal and will allow you to communicate with the remote host from your local system.

NFS Config 2

Saturday, March 3, 2018 9:23 AM

NFS Server, IP 10.1.1.100
NFS Client, IP 10.1.1.18

NFS Server configuration
---
install nfs packages
yum install nfs-utils rpcbind

create dir to be shared
#mkdir /nfs

edit /etc/exports
/nfs 10.1.1.18(no_root_squash,rw,sync)

either disable firewall
#systemctl stop firewalld
#systemctl disable firewalld

or

add entry in firewall
firewall-cmd --zone=public --add-port=2049/tcp --permanent
firewall-cmd --reload

start rpcbind daemon
#service rpcbind start
#service nfs start

status rpcbind daemon
#service nfs status

==========
CONFIGURATION ON THE CLIENT SIDE
yum install nfs-utils rpcbind
service rpcbind start

create a mount point for shared dir & mount it
mkdir -p /nfsclient
mount 10.1.1.110:/nfs /nfsclient

test it
cd /nfsclient/
touch NFS.test
ls -l
total 0
-rw-r--r--. 1 root root 0 Dec 11 08:13 NFS.test

==========
GOTO TO SERVER & CHECK
cd /nfs/
ls -l
total 0
-rw-r--r--. 1 root root 0 Dec 11 08:13 NFS.test

NFS Config 3

Saturday, March 3, 2018 9:24 AM

NFS CONFIGURATION
---
1. number of machines required = 1 server, 1 client
2. packages required = nfs-uitls, rpcbind

ON SERVER SIDE

IP = 192.168.10.10

install packages
```
# yum install -y rpcbind nfs-utils
```

create shared directory
```
#mkdir -p /opt/nfs
```

edit "vim /etc/exports" file
```
/opt/nfs 192.168.10.11(no_root_squash,rw,sync)
```

edit firewall or disable it
```
#firewall-cmd --zone=public --add-port=2049/tcp --permanent
#firewall-cmd --reload
```

restart nfs & rpc service
```
#service rpcbind start
#service nfs start
#service nfs status
```

==========
ON CLIENT SIDE

IP ADDRESS = 192.168.10.11

install packages
```
# yum install -y rpcbind nfs-utils
```

create shared directory
```
#mkdir -p /mnt/nfs
```

mount file system
```
#mount 192.168.10.10:/opt/nfs /mnt/nfs
```

then try creating a file in /mnt/nfs location

TO MOUNT IT PERMANENTLY
edit /etc/fstab file
```
#192.168.10.10:/opt/nfs /mnt/nfs nfs defaults 0 0
```

NIS Config

Saturday, March 3, 2018 9:24 AM

SERVER
---
1. yum install rpcbind ypserv ypbind -y

2. vim /etc/sysconfig/network
NISDOMAIN=mydomain.com

3. rpcinfo -u localhost ypserv
    failed as jst now installed

4. /usr/lib64/yp/ypinit -m
    see server hostname is available or not

5. /etc/init.d/rpcbind status

6. /etc/init.d/ypserv start

7. rpcinfo -u localhost ypserv
    startted now

8. useradd user1
  passwd user1

9. cd /var/yp
  make

CLIENT MACHINE
---
1. yum info ypbind rpcbind

2. /etc/init.d/rpcbind status

3. /etc/init.d/rpcbind restart

4. vim /etc/sysconfig/network
    NIDDOMAIN=mydomain.com

5. authconfig-tui
    use NIS
    domain -> mydomain.com
    server -> server.mydomain.com

6. getent passwd
    //user1 will be avilable

if not working TURN OFF firewall & selinux
after this home dir for that user is not there, make it

7. authconfig --enablemkhomedir --update
    if taking too long time, on main server
        #/etc/init.d/ypbind restart
        #/etc/init.d/ypserv restart

NTP

Monday, December 3, 2018 11:45 AM

Packages Required	#ntp
Config Files	/var/log/ntp.log; /etc/ntp.conf
Port Numbers	UDP 123 on OSI 4th (transport) layer
Service	# systemctl start ntpd # systemctl enable ntpd # systemctl status ntpd
Commands	# ntpq -p # date -R # ntpdate -q 0.ro.pool.ntp.org 1.ro.pool.ntp.org
Firewall rules	# firewall-cmd --add-service=ntp --permanent # firewall-cmd --reload

- Network Time Protocol (NTP)
- NTP- is a protocol which runs over **port 123** UDP at Transport Layer.
- Time sync protocol
- To set ntp

#rpm -ivh ntpd*	Install NTP
#ntpdate 1.ro.pool.ntp.org	update time, internet needed.
#timedatectl status	to check status

Installing NTP
#yum install -y ntp

After installation, visit this site for proper NTP details. Select correct continent & country.
https://www.pool.ntp.org/en/
https://www.pool.ntp.org/zone/in

Required files:
#vim /etc/ntp.conf

1. Comment all pool servers available.
2. Re-write the with new server available at (https://www.pool.ntp.org/zone/in)

   server 0.asia.pool.ntp.org
   server 1.asia.pool.ntp.org
   server 2.asia.pool.ntp.org
   server 3.asia.pool.ntp.org

3. To allow specific system to update from NTP site, edit same file with
   a. restrict <IP-RANGE> netmask <mask-value> nomodify notrap
      i. Nomodify & notrap = no other client will be able to update from internet.
4. To save NTP logs, append this line in last:
   a. logfile /var/log/ntp.log
5. Save & quit

Restart NTP service:

1. Chkconfig ntp on
2. Chkconfig --list ntp
3. Service ntp start

4. Service ntp status

Wait for some time to auto-sync & then verify using
#ntpq -p
#date -R

Else update manually
#ntpdate -q

URL: https://www.tecmint.com/install-ntp-server-in-centos/

FSTAB file

16 April 2020 06:53 AM

```
susel:~ # cat /etc/fstab
/dev/sda1            swap                swap       defaults              0 0
/dev/sda2            /                   ext3       acl,user_xattr        1 1
proc                 /proc               proc       defaults              0 0
sysfs                /sys                sysfs      noauto                0 0
debugfs              /sys/kernel/debug   debugfs    noauto                0 0
usbfs                /proc/bus/usb       usbfs      noauto                0 0
devpts               /dev/pts            devpts     mode=0620,gid=5       0 0
# /dev/sr0      /cdrom  iso9660 ro,nosuid,nodev,uid=0    0 0
/dev/sdc1            /novi_disk                    ext3       acl,user_xattr,usr
quota,grpquota        2 0
```

Col1	Col2	Col3	Col4	Col5	Col6
Device	Mount point	File system type	Options	Backup operation	File system check order
• the first field specifies the mount device.	• the second field specifies the mount point	• the third field specifies the file system type	• the fourth field specifies the mount options. You may specify multiple mount options, separated by commas.	• the fifth field contains a 1 if the dump utility should back up a partition or a 0 if it shouldn't. • 1 = means taking DUMP backup • 0 = don't take backup	• the order in which fsck checks the device/partition for errors at boot time. • 0 means that fsck should not check a file system. • The root partition should have a value of 1 , and all others that need to be checked should have a value of 2.

Rsync - R7 - verified

16 April 2020 09:42 PM

- Rsync (**Remote Sync**) is used for copying and synchronizing files and directories remotely as well as locally in Linux/Unix systems.
- It efficiently **copies and sync** files to or from a remote system.
- Supports copying **links, devices, owners, groups** and **permissions**.
- It's **faster than scp** (Secure Copy) because rsync uses remote-update protocol which allows to transfer just the differences between two sets of files. First time, it copies the whole content of a file or a directory from source to destination but from next time, it copies only the changed blocks and bytes to the destination.
- **Rsync consumes less bandwidth** as it uses compression and decompression method while sending and receiving data both ends.

Basic syntax of rsync command

rsync <options> <source> <destination>

Some common options used with rsync commands

- **-v** : verbose
- **-r** : copies data recursively (but don't preserve timestamps and permission while transferring data
- **-a** : archive mode, archive mode allows copying files recursively and it also preserves symbolic links, file permissions, user & group ownerships and timestamps
- **-z** : compress file data
- **-h** : human-readable
- **-e** : specific protocol to use

Install rsync in your Linux machine

yum install rsync

Copy/Sync Files and Directory Locally:

#rsync -zvh backup.tar /tmp/backups/

Copy/Sync a Directory on Local Computer:

#rsync -avzh /root/rpmpkgs /tmp/backups/

Copy/Sync Files and Directory to or From a Server:

#rsync -avz rpmpkgs/ root@192.168.0.101:/home/

Rsync Over SSH:

#rsync -avzhe ssh root@192.168.10.103:/root/Desktop/rpm /root/Desktop/

Show Progress While Transferring Data with rsync:

#rsync -avz --progress . root@192.168.10.101:/root/Desktop/rsync

Set the Max Size of Files to be Transferred:

#rsync -avzhe ssh root@192.168.10.103:/root/Desktop/rpm-new /root/Desktop/

NTP Server Config

Saturday, March 3, 2018 9:24 AM

Configuring NTP server

#yum install -y ntp // install ntp
#vim /etc/ntp.conf // edit ntp conf file

edit "restrict" line
restrict <ip> netmask <mask> nomodify notrap

at last line
logfile /var/log/ntp.log

add file rules or turn it off
#firewall-cmd --add-service=ntp --permanent
#firewall-cmd --reload

#systemctl start ntpd
#systemctl enable ntpd
#systemctl status ntpd

verify server time sync
#ntpq -p
#date -R

if u want to query & sync against a pool of ur choise
#ntpupdate -q 0.ro.pool.ntp.org 1.ro.pool.ntp.org

NMCLI

Monday, December 3, 2018 11:47 AM

[How to Configure and Manage Network Connections Using 'nmcli' Tool]

#nmcli device status	List device status
#nmcli connection	List connections
#nmcli device show	Show device status
#nmcli -t device connection show eno16777736	t = terse/brief intro
#nmcli -p device show eno16777736	p= pretty / good GUI
#nmcli -version	Check NMCLI version

NMCLI: network manager command line tool

The syntax of nmcli is:

```
# nmcli [OPTIONS] OBJECT {COMMAND | help}
```

A good starting point would be to check our devices:

```
# nmcli dev status

DEVICE      TYPE      STATE         CONNECTION
docker0     bridge    connected     docker0
virbr0      bridge    connected     virbr0
enp0s3      ethernet  connected     enp0s3
virbr0-nic  ethernet  disconnected  --
lo          loopback  unmanaged     --
```

Check Network Configuration

We realize that some properties have different values and some others don't exist if it isn't necessary. Let's have a quick look to most important of them.

TYPE, we have ethernet type here. We could have wifi, team, bond and others.

DEVICE, the name of the network device which is associated with this profile.

BOOTPROTO, if it has value "dhcp" then our connection profile takes dynamic IP from dhcp server, if it has value "none" then it takes no dynamic IP and probably whe assign a static IP.

IPADDR, is the static IP we assign to our profile.

PREFIX, the subnet mask. A value of 24 means 255.255.255.0. You can understand better the subnet mask if you write down its binary format. For example values of 16, 24, 26 means that the first 16, 24 or 26 bits respectively are 1 and the rest 0, defining exactly what the network address is and what is the range of ip which can be assigned.

GATEWAY, the gateway IP.

DNS1, DNS2, two dns servers we want to use.

ONBOOT, if it has value "yes" it means, that on boot our computer will read this profile and try to assign it to its device.

Now, let's move on and check our connections:

```
# nmcli con show
```

```
[root@ira network-scripts]# nmcli con show
NAME       UUID                                  TYPE            DEVICE
static1    bfa7a232-f5b5-4d8f-a6fa-f972c60f56b7  802-3-ethernet  --
Myoffice1  2a4a63ec-c6bd-4c76-a92e-0554b7c03817  802-3-ethernet  enp0s3
enp0s3     354faa8a-9efe-4cf0-a03b-d9dd0d7e6311  802-3-ethernet  --
enp0s8     4c00d95c-7fa2-4e5b-968f-4b8609773501  802-3-ethernet  enp0s8
```

The last column of devices helps us understand which connection is "UP" and running and which is not. In the above image you can see the two connections which are active: Myoffice1 and enp0s8.

Hint: If you want to see only the active connections, type:

nmcli con show -a

Hint: You can use the auto-complete hitting Tab when you use nmcli, but is better to use minimal format of the command. Thus, the following commands are equal:

nmcli connection show

nmcli con show

nmcli c s

If I check the ip addresses of my devices:

ip a

```
[root@ira network-scripts]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:50:c4:19 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.40/24 brd 192.168.1.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 2a02:582:5857:8400:a00:27ff:fe50:c419/64 scope global dynamic
       valid_lft 86149sec preferred_lft 86149sec
    inet6 fe80::a00:27ff:fe50:c419/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:33:2a:aa brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.7/24 brd 192.168.1.255 scope global dynamic enp0s8
       valid_lft 1810298sec preferred_lft 1810298sec
    inet6 2a02:582:5857:8400:a00:27ff:fe33:2aaa/64 scope global dynamic
       valid_lft 86149sec preferred_lft 86149sec
    inet6 fe80::a00:27ff:fe33:2aaa/64 scope link
       valid_lft forever preferred_lft forever
```

We can make our first connection profile. The minimum properties we must define are type, ifname and con-name:

type – for the type of connection.

ifname – for the device name which is assigned our connection.

con-name – for the connection name.

Let's make a new ethernet connection with name Myhome1, assigned to device enp0s3:

nmcli con add type ethernet con-name Myhome1 ifname enp0s3

Check its configuration:

cat ifcfg-Myhome1

```
[root@ira network-scripts]# nmcli con add type ethernet con-name Myhome1 ifname enp0s3
Connection 'Myhome1' (6ca1c9f5-0fb5-4505-a185-5e27b7fdeb14) successfully added.
[root@ira network-scripts]# cat ifcfg-Myhome1
TYPE=Ethernet
BOOTPROTO=dhcp
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
NAME=Myhome1
UUID=6ca1c9f5-0fb5-4505-a185-5e27b7fdeb14
DEVICE=enp0s3
ONBOOT=yes
```

As you can see it has **BOOTPROTO=dhcp**, because we didn't give any static ip address.

Hint: We can modify any connection with the "**nmcli con mod**" command. However if you modify a dhcp connection and change it to static don't forget to change its **"ipv4.method**" from "**auto**" to "**manual**". Otherwise you will end up with two IP addresses: one from dhcp server and the static one.

Let's make a new Ethernet connection profile with name **static2**, which will be assigned to device **enp0s3**, with static IP 192.168.1.50, subnet mask 255.255.255.0=24 and gateway 192.168.1.1.

```
# nmcli con add type ethernet con-name static2 ifname enp0s3 ip4 192.168.1.50/24 gw4
192.168.1.1
```

Check its configuration:

```
# cat ifcfg-static2
```

Let's modify the last connection profile and add two dns servers.

```
# nmcli con mod static2 ipv4.dns "8.8.8.8 8.8.4.4"
```

Now let's bring up this connection profile:

```
# nmcli con down static1 ; nmcli con up static2
```

nmcli con show static2

New Link: http://www.linux-commands-examples.com/nmcli

nmcli -t -f RUNNING nm

tells you whether NetworkManager is running or not.

nmcli -t -f STATE nm

shows the overall status of NetworkManager.

nmcli nm wifi off

switches Wi-Fi off.

nmcli -p con list

lists all connections NetworkManager has.

nmcli -f name,autoconnect con list

lists all connections' names and their autoconnect settings.

nmcli con list id "My wired connection"

lists all details of the connection with "My wired connection" name.

nmcli -p con up id "My wired connection" iface eth0

activates the connection with name "My wired connection" on interface eth0. The -p option makes nmcli show progress of the activation.

nmcli con up uuid 6b028a27-6dc9-4411-9886-e9ad1dd43761 ap

00:3A:98:7C:42:D3

connects the Wi-Fi connection with UUID 6b028a27-6dc9-4411-9886-e9ad1dd43761 to the AP with BSSID 00:3A:98:7C:42:D3.

nmcli dev status

shows the status for all devices.

nmcli dev disconnect iface em2

disconnects a connection on interface em2 and marks the device as unavailable for auto-connecting. That's why no connection will automatically be activated on the device until the device's "autoconnect" is set to TRUE or user manually activates a connection.

nmcli -f GENERAL,WIFI-PROPERTIES dev list iface wlan0

lists details for wlan0 interface; only GENERAL and WIFI-PROPERTIES sections will be shown.

nmcli dev wifi

lists available Wi-Fi access points known to NetworkManager.

nmcli dev wifi con "Cafe Hotspot 1" password caffeine name "My cafe"

creates a new connection named "My cafe" and then connects it to "Cafe Hotspot 1" SSID using "caffeine" password. This is mainly useful when connecting to "Cafe Hotspot 1" for the first time. Next time, it is better to use 'nmcli con up id "My cafe"' so that the existing connection profile can be used and no additional is created.

https://arkit.co.in/nmcli-command-network-manager-linux/

To display connections
NMCLI con show
OR
NMCLI dev status

To delete the connection ethernet-eth0
NMCLI con del ethernet-eth0

To create a connection with the name ethernet-eth0, the IPv4 address 192.168.1.10/24 and the default gateway 192.168.1.1, type:
nmcli con add con-name net-eth0 ifname eth0 type ethernet ip4 192.168.1.10/24 gw4 192.168.1.1

To check ip configuration
#ip a ip addr show
#ifconfig

#ip r Ip route show

To get all the information about a connection (here net-eth0), type:
nmcli con show net-eth0

To stop a network connection from working (here net-eth0), type:
nmcli con down net-eth0
nmcli con show

To start a network connection (here net-eth0), type:
nmcli con up net-eth0

Networking

Monday, December 3, 2018 12:00 PM

RedHat NETWORKING

==========

1. /etc/hostname	holds system name
2. /etc/sysconfig/network-scripts/ifcfg-eno16777736	holds ip address, subnet mask, gateway, dns servers etc ethernet card name will only change by udev command
3. /etc/resolve.conf	holds dns server entries
4. /etc/nsswitch.conf	holds name server switch order [name server like dns, ldap, nfs]
to ping a system for 4 times	#ping -c 4 google.com
to ping system after every 10 sec	#ping -i 10 google.com
managing network gui	#nmgui #nmcli
to check default gateway	#netstat -r
to check dns lookup we have 3 utilities	dns looup 1. nslookup <url> 2. hostname <url> 3. dig <url>

Networking RHEL 6

Tuesday, December 25, 2018 3:36 PM

Three categories for network config:

- Interface configuration files
- Interface control scripts
- Network function files

Imp files in network config:

/etc/hosts	resolve host names
/etc/resolv.conf	specifies the IP addresses of DNS servers
/etc/sysconfig/network	specifies routing and host information for all network interfaces
/etc/sysconfig/network-scripts/ifcfg-interface-name	For each network interface, there is a corresponding interface configuration script

Config tools:

1. NMCLI
    - #nmcli dev status
    - #nmcli nm wifi
    - #nmcli con list
    - #nmcli con status
    - #nm-tool
2. #system-config-network (for RHEL 6) // aka NMTUI in R7 OR
3. #system-config-network-tui

Process

Monday, December 3, 2018 11:46 AM

#pstree	//display a tree of process
#ps -aux less	//a=all process, u=all process of other users, x= all terminals
#ps -A	//all process in linux system
#ps -u <username>	//all process by a user
#top -b -n1 > file.txt	//storing data in file
#pgrep <process-name>	//get process id
#kill <PID>	//killing ID
#kill -l	//list of killing terms

Fork: https://www.systutorials.com/docs/linux/man/2-fork/

Partitioning

Monday, December 3, 2018 12:08 PM

naming conventions
- hdd1 = /dev/sda - partition1 (/dev/sda1), -partiton2 (/dev/sda2)
- hdd2 = /dev/sdb - partition1 (/dev/sdb1), -partiton2 (/dev/sdb2)
- hdd3 = /dev/sdc - partition1 (/dev/sdc1)

types of partitions
1. primary partition
2. extended partition

a. Logical Partition

in one hdd, we can create maximum 4 partitions
p1, p2, p3, p4
or
p1, p2, p3, e(L1, L2, L3...) //e=extended,L=Logical

MBR (master boot record)
- o total size = 512 bytes (size cannot be modified)
- o 1st sector on hdd
- o 446 bytes = GRUB (GRand Unified Boot) loader
- o 64 bytes = Partition info (hdd geometry, 16 bytes per partition)
- o 2 bytes = magic bytes (partition validation, partition is perfect(geo)or not)

Creating partition
Query:- create the partition with 5gb size & mount it on /oracle mount point
steps:
1. fdisk = to create partition
2. mkxfs.xfs = to create xfs file system
3. mkdir = create a mount point
4. vim = to add an entry of new partition in /etc/fstab file
5. mount = to mount new partition & verify it

Method:

Step 1:
#fdisk /dev/sda
m = help
p = print table
n = new partition
p = primary partition
partiton number = <3>
w = write & save
init 6 // to reboot for initiate the partition

Step 2:
#mkfs.xfs /dev/sda6

step 3.
#mkdir /oracle

step 4.
#vim /etc/fstab
at end
press o
/dev/sda3 /oracle xfs defaults 1 2
:wq!

step 5.
#mount -a //mount
#mount //verify

#df -h
end

LVM (Advance partitioning)

step a
physical partitions [hdd1], [hdd2], [hdd3]
step b
physical volume (pv)
to create = pvcreate
to display = pvdisplay
to remove = pvremove
step c
volume group (vg) // name = vg1
to create = vgcreate
to display = vgdisplay
to remove = vgremove
step d
logical volume (lv) // name = lv1
to create = lvcreate
to display = lvdisplay
to remove = lvremove
Now full procedure for advance partitoning

---

s1. fdisk
s2. physical partitions [hdd1], [hdd2], [hdd3]
s3. physical volume (pv)
s4. volume group (vg)
s5. logical volume (lv)
s6. mkfs.xfs
s7. mkdir
s8. vim
s9. mount

checking new partition
#fdisk -l
create disks
#fdisk /dev/sda
n // to create a new partition
w // save & quit
#fdisk /dev/sdb
n
w
#fdisk /dev/sdc
n
w
reboot
pv create
#pvcreate /dev/sda7 /dev/sba1 /dev/sdc1
#pvdisplay
vg create
#vgcreate vg1 /dev/sda7 /dev/sdb1 /dev/sdc1
vg1 = name for volume group
#vgdisplay
lv create
#lvcreate -n lv1 -L 20G vg1
n=name
L=size

#lvdisplay
#mkfs.xfs
#mkdir
#vim
#mount

Precision Time Protocol (PTP)

Thursday, February 7, 2019 10:42 AM

http://linuxptp.sourceforge.net/

R6 vs R7

Saturday, March 3, 2018 9:26 AM

[1]ON THE BASIS OF RELEASE DATE.
RELEASE DATE OF RHEL6 IS 10th NOV 2010.
RELEASE DATE OF RHEL7 IS 10TH JUNE 2014.
Hence RHEL7 is latest os.

[2]DIFFERENCE ON THE BASIS OF OPERATING SYSTEM NAMES
If you want to see this use this command
#cat /etc/redhat-release
RHEL6 : REDHAT ENTERPRISE LINUX (SANTIGO)
RHEL7: REDHAT ENTERPRISE LINUX (MAIPO)

[3]KERNEL VERSION
RHEL6: 2.6.32
RHEL7: 3.0.10

[4] OS BOOT TIME
RHEL6: 40 sec
RHEL7: 20 sec

[5]MAXIMUM SIZE OF SINGLE PARTITION
RHEL6: 50TB(EXT4)
RHEL7: 500TB(XFS)

[6] BOOT LOADER
RHEL6: /boot/grub/grub.conf
RHEL7: /boot/grupb2/grub.cfg

[7]PROCESSOR ARCHITECTURE
RHEL6: It support 32bit & 64bit both
RHEL7: It only support 64bit

[8]HOW TO FORMAT OR ASSIGN A FILE SYSTEM IN
RHEL6: #mkfs.ext4 /dev/hda6
RHEL7: #mkfs.xfs /dev/hda6

[9]HOW TO REPAIR A FILE SYSTEM IN
RHEL6: #fsck -y /dev/hda6
RHEL7: #xfs_repair /dev/hda6

[10]COMMAND TO MANAGE NETWORK IN RHEL6 AND RHEL7
RHEL6: #setup
RHEL7: #nmtui

[11]HOSTNAME CONFIGURATION FILE
RHEL6: /etc/sysconfig/network
RHEL7: /etc/hostname

[12]DEFAULT ISO IMAGE MOUNT PATH
RHEL6: /media
RHEL7: /run/media/root

[13]FILE SYSTEM CHECK

RHEL6: e2fsck

RHEL7: xfs_repair

[14]RESIZE A FILE SYSTEM

RHEL6: #resize2fs -p /dev/vg00/lv1

RHEL7: #xfs_growfs /dev/vg00/lv1

[15]TUNE A FILE SYSTEM

RHEL6: tune2fs

RHEL7: xfs_admin

[16]IPTABLES AND FIREWALL

RHEL6: iptables

RHEL7: firewalld

RAID 1 Config

Saturday, March 3, 2018 9:26 AM

Setting up RAID 1 (Mirroring) using 'Two Disks' in Linux

Step 1: Installing Prerequisites and Examine Drives
1. As I said above, we're using mdadm utility for creating and managing RAID in Linux. So, let's install the mdadm software package on Linux using yum or apt-get package manager tool

```
# yum install mdadm
```

2. Once 'mdadm' package has been installed, we need to examine our disk drives whether there is already any raid configured using the following command.

```
# mdadm -E /dev/sd[b-c]
```

Step 2: Drive Partitioning for RAID
3. As I mentioned above, that we're using minimum two partitions /dev/sdb and /dev/sdc for creating RAID1. Let's create partitions on these two drives using 'fdisk' command and change the type to raid during partition creation.

```
# fdisk /dev/sdb
# fdisk /dev/sdc
```

4. Once both the partitions are created successfully, verify the changes on both sdb & sdc drive using the same 'mdadm' command and also confirm the RAID type as shown in the following screen grabs.

```
# mdadm -E /dev/sd[b-c]
```

Step 3: Creating RAID1 Devices
5. Next create RAID1 Device called '/dev/md0' using the following command and verity it.

```
# mdadm --create /dev/md0 --level=mirror --raid-devices=2 /dev/sd[b-c]1
# cat /proc/mdstat
```

6. Next check the raid devices type and raid array using following commands.

```
# mdadm -E /dev/sd[b-c]1
# mdadm --detail /dev/md0
```

Step 4: Creating File System on RAID Device
7. Create file system using ext4 for md0 and mount under /mnt/raid1.

```
# mkfs.ext4 /dev/md0
```

8. Next, mount the newly created filesystem under '/mnt/raid1' and create some files and verify the contents under mount point.

```
# mkdir /mnt/raid1
# mount /dev/md0 /mnt/raid1/
# touch /mnt/raid1/tecmint.txt
# echo "tecmint raid setups" > /mnt/raid1/tecmint.txt
```

9. Open '/etc/fstab' file and add the following line at the bottom of the file.

```
/dev/md0        /mnt/raid1      ext4    defaults        0 0
```

10. Run 'mount -a' to check whether there are any errors in fstab entry.

```
# mount -av
```

11. Next, save the raid configuration manually to 'mdadm.conf' file using the below command.

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

Step 5: Verify Data After Disk Failure

Redirection

Tuesday, September 24, 2019 9:54 PM

Supported Operators

- > Redirect STDOUT to File
- 2> Redirect STDERR to File
- &> Redirect All Output (STDOUT and STDERR)

Redirecting Output to File

- STDOUT and STDERR can be redirected to files.

Examples:

```
* command < file              Send file as a Input to the command.
* command > file              Redirect STDOUT of command to file.
* command >> file             Append STDOUT of command to file.
* command 2> file             Redirect STDERR of command to file.
* command 2>> file            Append STDERR of command to file.
```

```
[stark@client ~]$ ls
haha  pwd-error.txt
[stark@client ~]$ cat h1 > h2
cat: h1: No such file or directory
[stark@client ~]$ cat h1 2> h2
[stark@client ~]$ ls
h2  haha  pwd-error.txt
[stark@client ~]$ cat h2
cat: h1: No such file or directory
[stark@client ~]$ cat h3 > h4 2>&1
[stark@client ~]$ ls
h2  h4  haha  pwd-error.txt
[stark@client ~]$ cat h4
cat: h3: No such file or directory
[stark@client ~]$
```

Root PWD reset - R7

Saturday, March 3, 2018 9:27 AM

Changing root password

---

remove rw, rhgb & type init=/bin/bash & ctrl-X

bash-4.2#passwd root
bash-4.2#df -h
bash-4.2#umount /
bash-4.2#df -h
bash-4.2#mount -o rw /
bash-4.2#mount -o rw,remount /
bash-4.2#passwd root
bash-4.2# <type new passwd twice>

==========

Reset root password

---

1. reboot ur machine
2. now at the boot loader screen, press "e"
3. now move the cursor to the line starts with (linux 16)
4. goto the end of the line & type "rd.break"
5. press "ctrl + x" to boot
6. now emergency mode will be opened
    switch_root # mount -o remount,rw /sysroot
    switch_root # chroot /sysroot/
        sh-4.2# passwd root
        New Password: xxx
        Retype New Password: xxx
        sh-4.2# touch /.autorelabel
        sh-4.2# exit
    switch_root # exit
    logout

    Now machine will be rebooted

Root PWD reset - R6

Tuesday, December 25, 2018 9:26 PM

- Reboot the system.
- Open grub screen & press 'e'
- Select the line where "/vmlinuz…" is written.
- After this add "1" after quite, which will help the system to boot in INIT 1 mode
- Change the passwd using passwd root cmd
- Reboot the system, using init 6

To protect the grub, apply passwd on grub

- Use below command to create encrypted passwd
  - #grub-md5-crypt
  - Enter a secure passwd twice
  - Copy the output
- Open below file:
  - #vim /boot/grub/grub.conf
  - Edit following line above "SPLASHIMAGE"
  - password --md5 <paste-output-of-grub-md5-crypt>
  - Save & quit
- Reboot the system & check by pressing "p"

https://www.tecmint.com/password-protect-grub-in-linux/

Root PWD recovery

Monday, December 17, 2018 5:24 PM

Recover root password:

1. Follow the steps from step 1 to step 5 (i.e, chroot /mnt/sysimage)
2. Open shadow file:
    1. #vi /etc/shadow
    2. Remove the password of ROOT user between ":", ":".
    3. Exit
    4. Reboot
3. After reboot, login directly with root user (w/o password) & change the password using "passwd" command.

Encrypt GRUB password (link)

- Login with 'root'
- Use below command to create encrypted command
    1. #grub2-mkpasswd-pbkdf2
        i. Enter a new password twice
    2. Now edit grub file (above line 72)
        i. Set superusers="root"
        ii. Export superusers
        iii. Password_pbkdf2 root <Enter-grub2-mkpasswd-pbkdf2-you-created-in step"1.i">
        iv. Reboot the system

Rsyslog log server

Sunday, December 23, 2018 12:03 PM

Packages Required	# yum install rsyslog
Config Files	/etc/rsyslog.conf, TCP is indicated by @@, UDP by @
Port Numbers	514
Service	# service rsyslog restart # systemctl start rsyslog
Commands	rsyslogd -v <client-send-log-to-SVR>
Firewall rules	# semanage port -l | grep syslog # semanage port -l | grep 514 # firewall-cmd --zone=zone --add-port=10514/tcp

- Remote logging is provided by Rsyslog.
- Works on port number 514
- Requires at least 2 systems
    1. Server - for storing logs
    2. Client

========VERIFIED==========================

AT SERVER SIDE

- Install rsyslog server
    - **#yum install -y rsyslog**
- Configure /etc/rsyslog.conf file
    - Uncomment
      $ModLoad imtcp
      $InputTCPServerRun 514

- **Restart the rsyslog service**

AT CLIENT SYSTEM:

- Install Rsyslog server
    - #yum install -y rsyslog
- Configure /etc/rsyslog.conf file
    - Add (for TCP protocol)
      . @@<server-name>:514
    - Add (for UDP protocol)
      . @<server-name>:514

- Restart the rsyslog service

==========

AT CLIENT SYSTEM:

- Install Rsyslog server
  - #yum install -y rsyslog

- Edit configuration file:
  - #vim /etc/rsyslog.conf
  - Uncomment below line to start logging.

```
$WorkDirectory /var/lib/rsyslog
$ActionQueueFileName fwdRule1
$ActionQueueMaxDiskSpace 2g
$ActionQueueSaveOnShutdown on
$ActionQueueType LinkedList
$ActionResumeRetryCount -1
*.* @@<server-IP-address>:514
```

  - @ means UDP
  - @@ means TCP

- Save, quit file & restart the rsyslog service
  - #service rsyslog restart

AT SERVER SIDE

- Install rsyslog server
  - #yum install -y rsyslog

- Configure rsyslog file for collecting logs
  - #vim /etc/rsyslog.conf
    - Search for the below lines

```
$ModLoad imtcp
$InputTCPServerRun 514
:FROMHOST-IP, isequal, "192.168.0.101" /var/log/rhel6.log
```

- Save, quit file & restart the rsyslog service
  - #service rsyslog restart

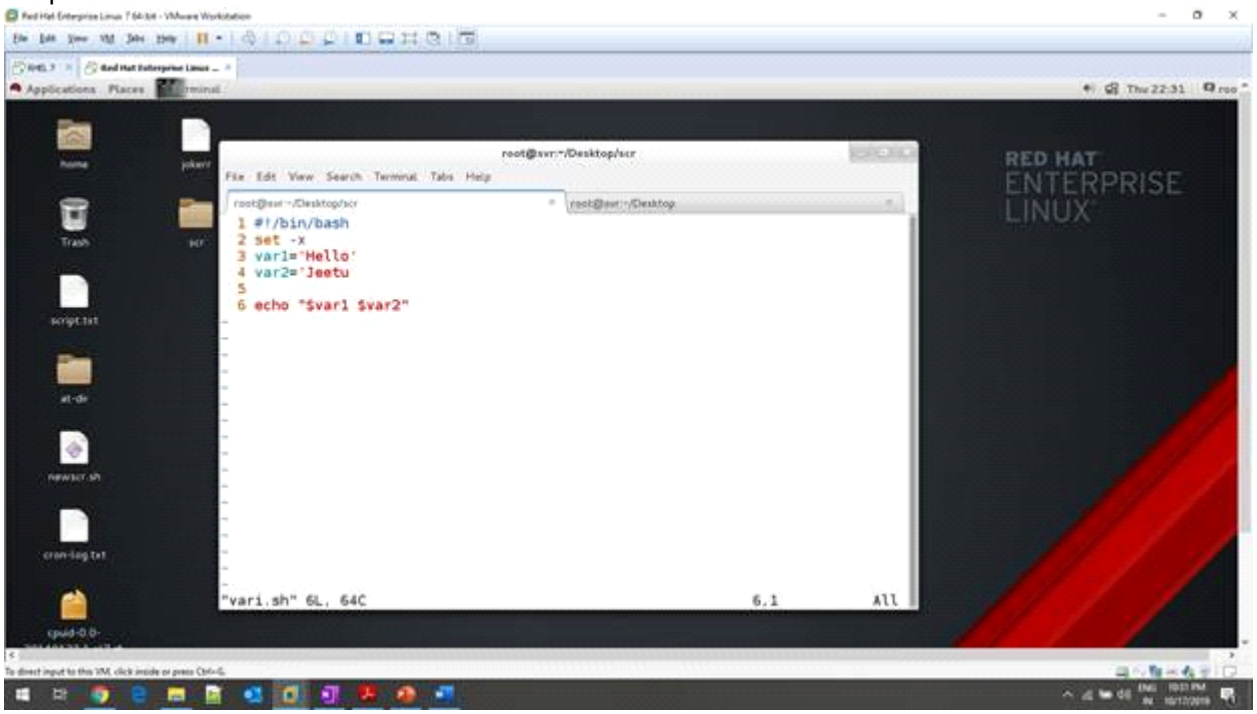
After this

- Switch to client system.
- Execute below command to manually check logs
  - #logger "<any message>"
- & verify @server using below command:
  - #tail /var/log/messages

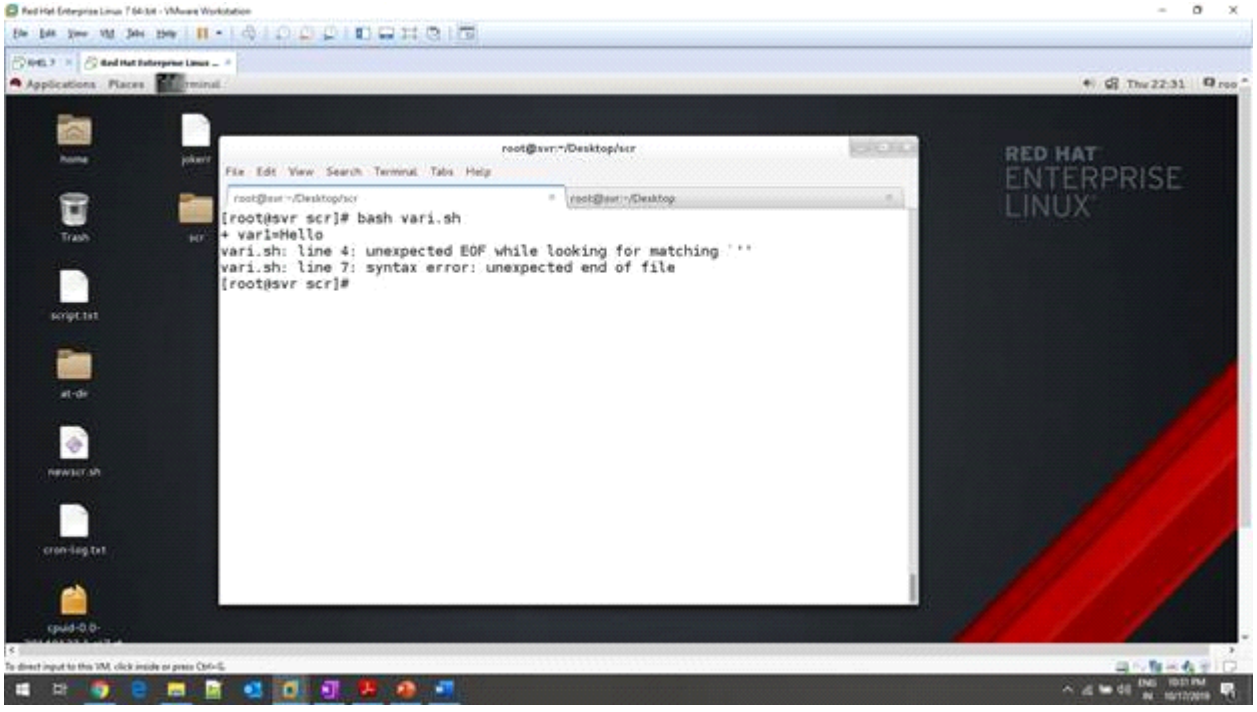
Shell Scripts basics

Thursday, October 17, 2019 11:44 PM

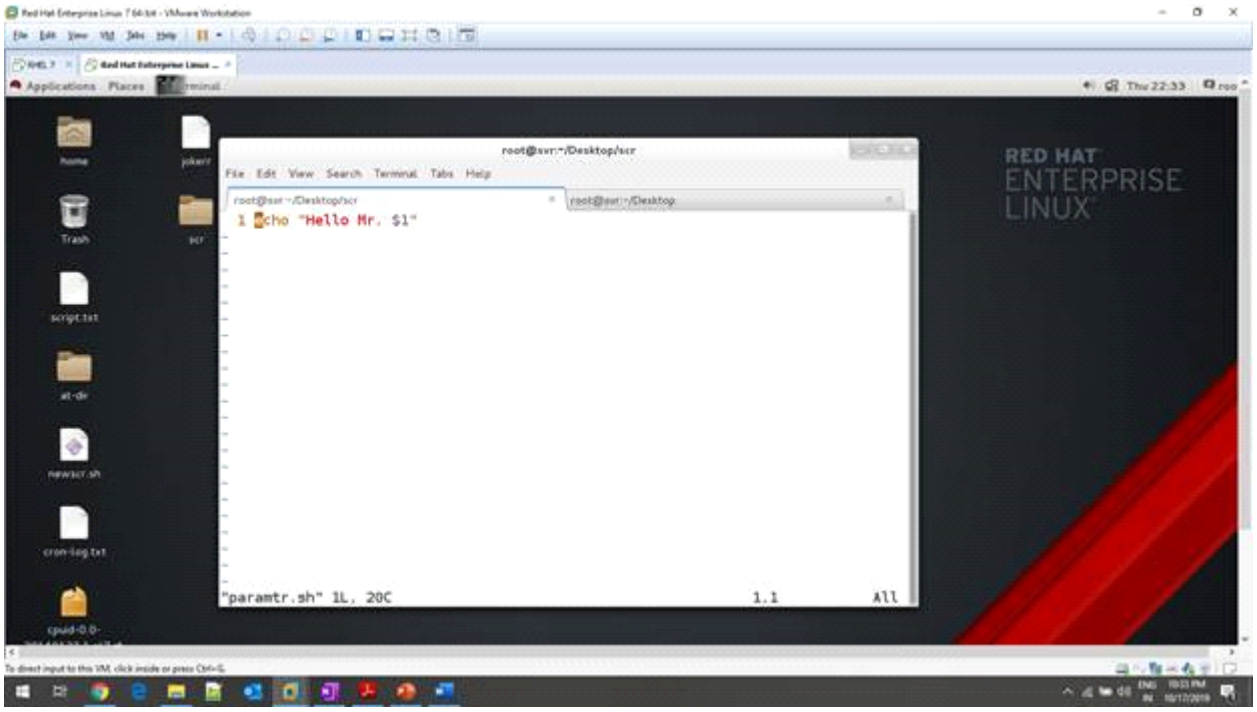
1. Variables print



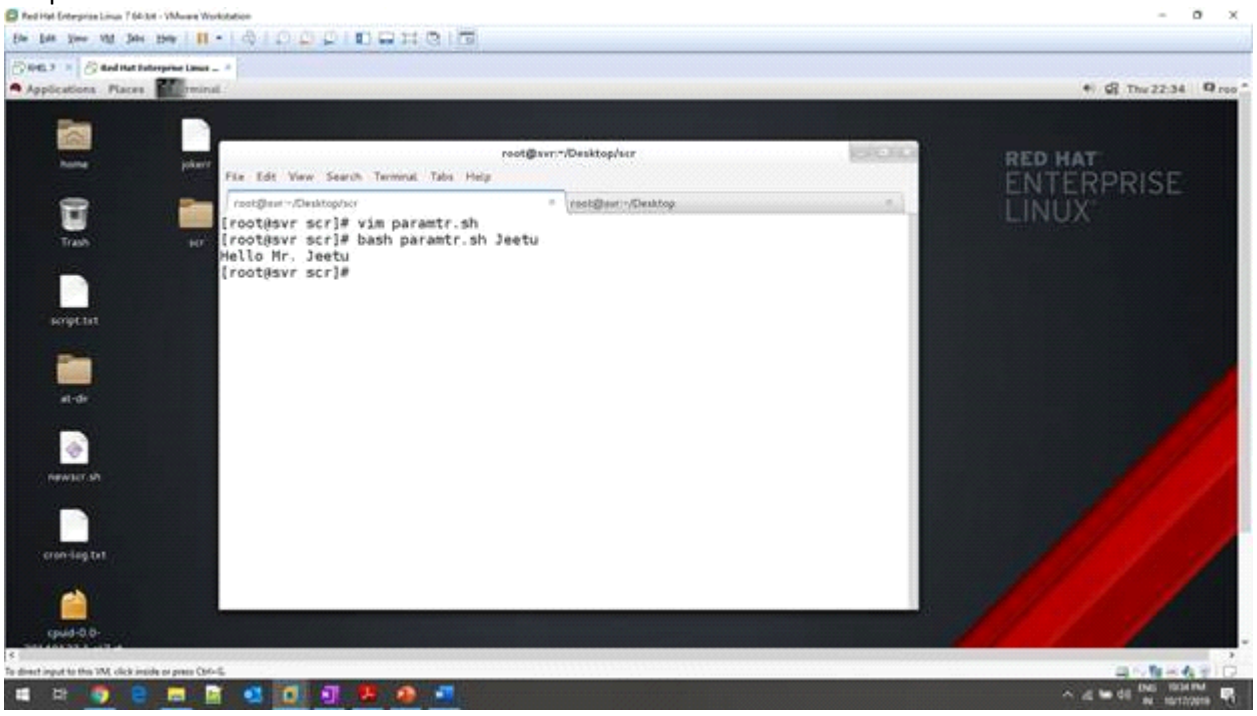
Output with error detection:



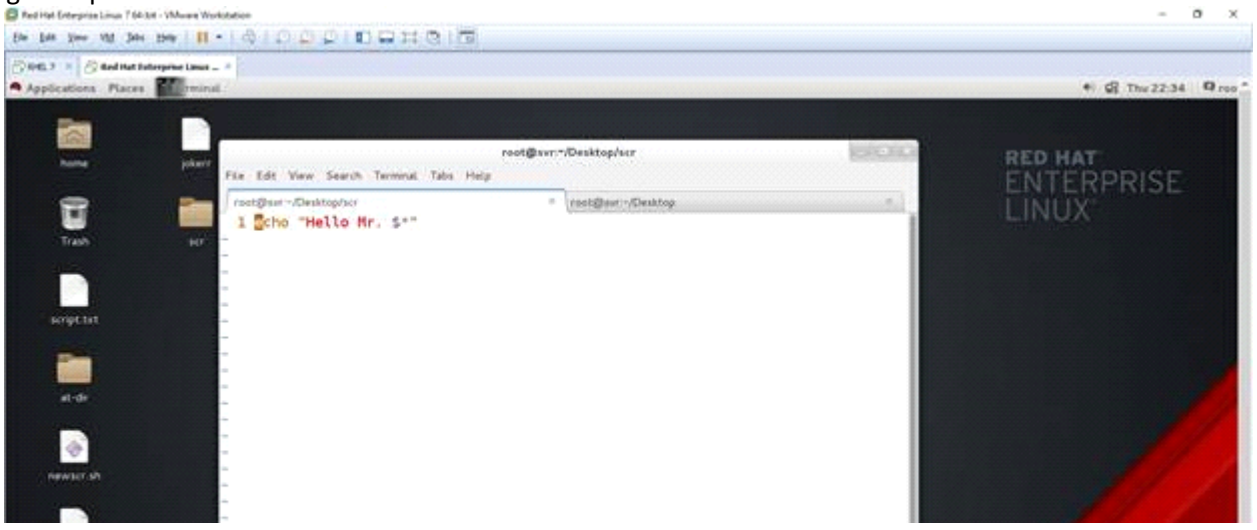
2. Passing parameters



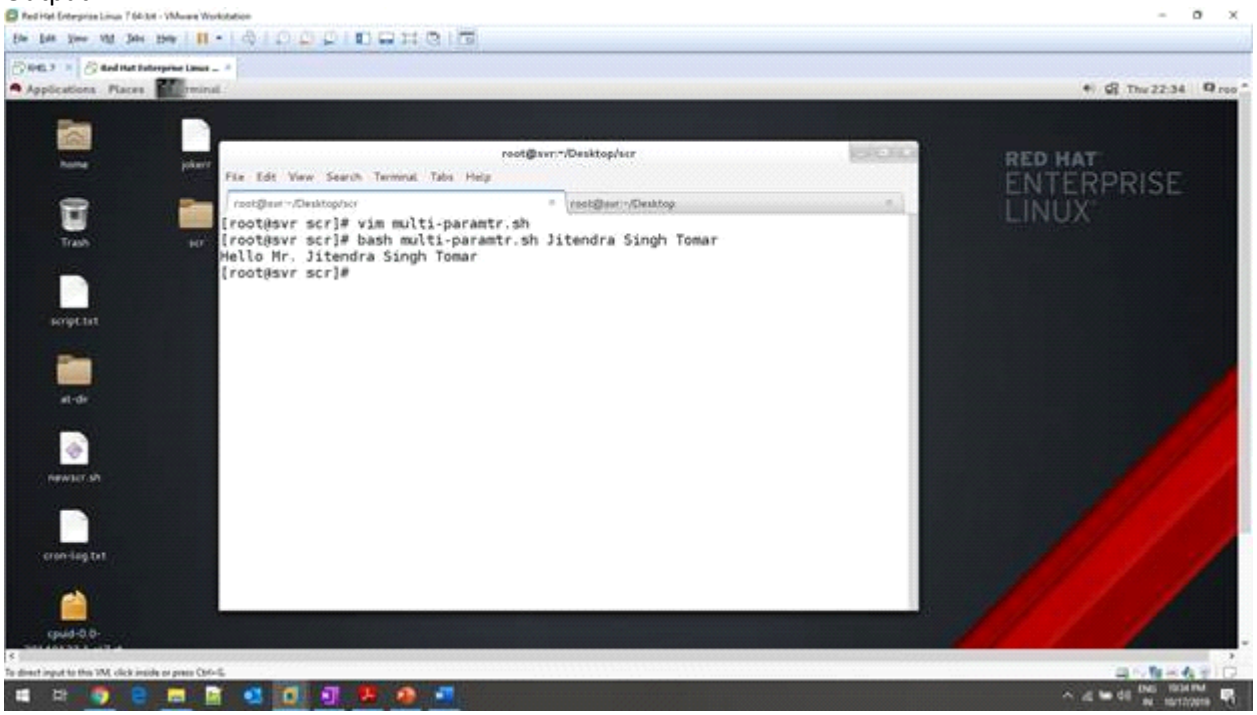
Output



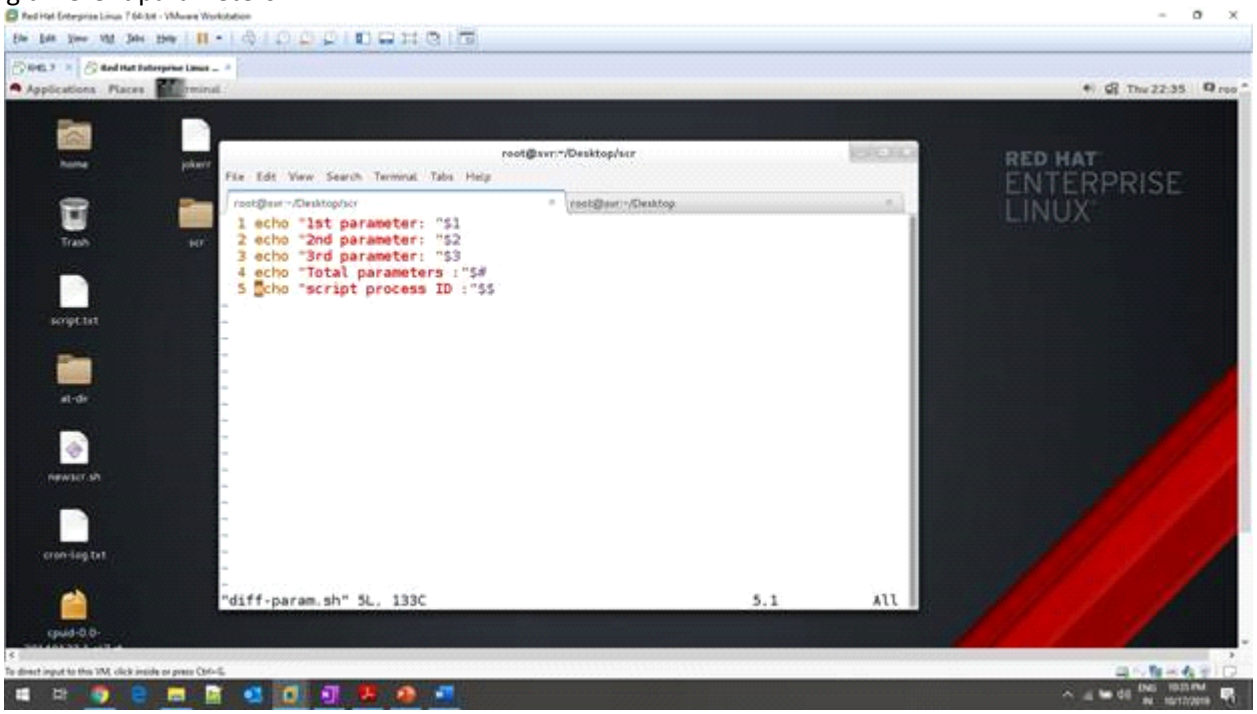
3. Passing multiple variables



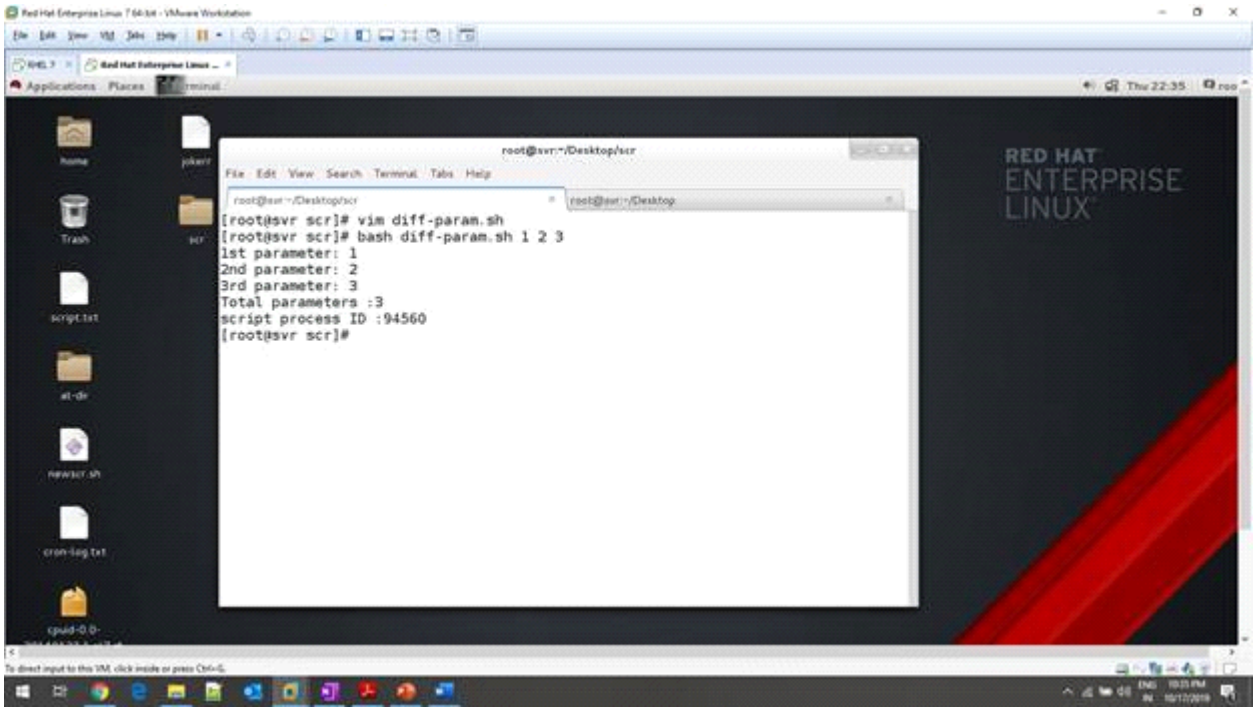
Output:



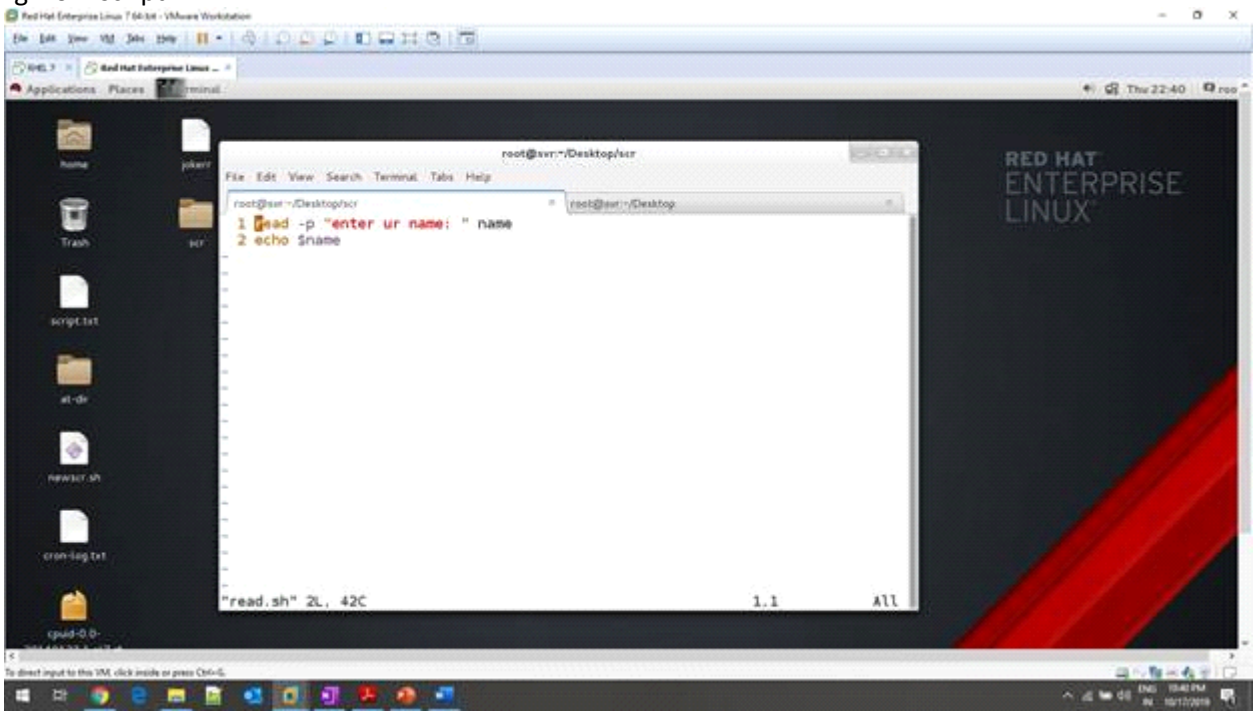
4. Passing different parameters



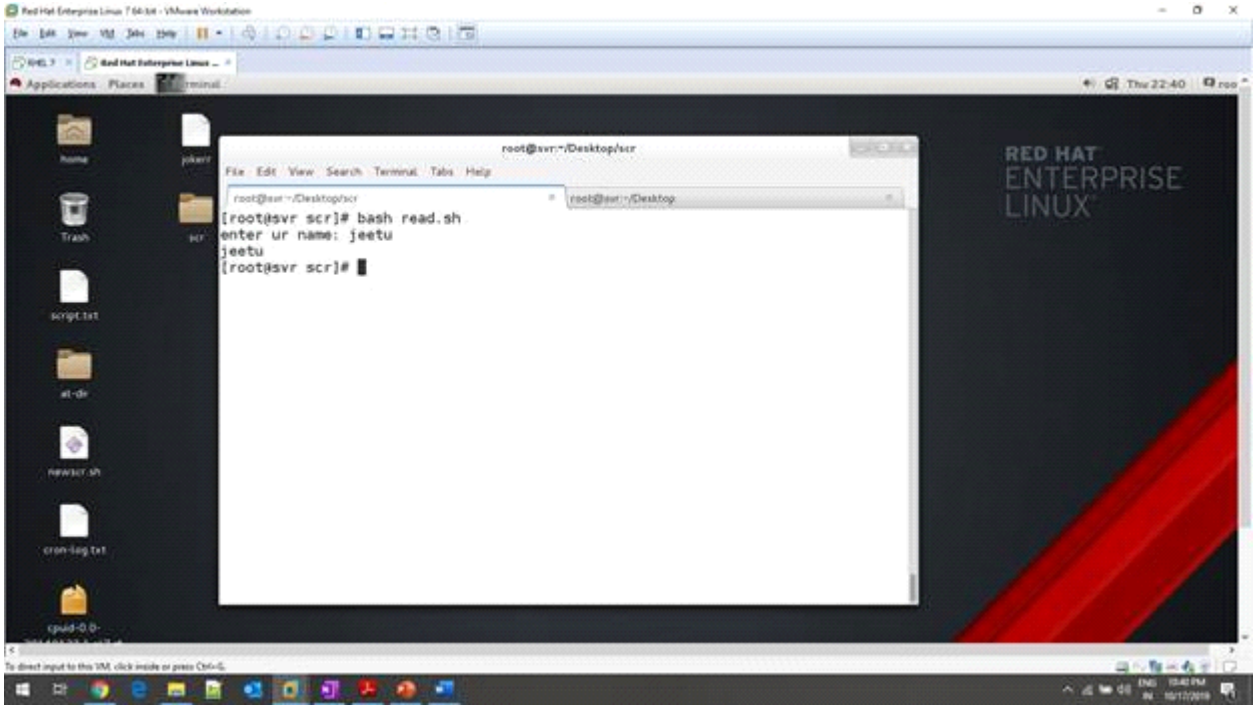
Output



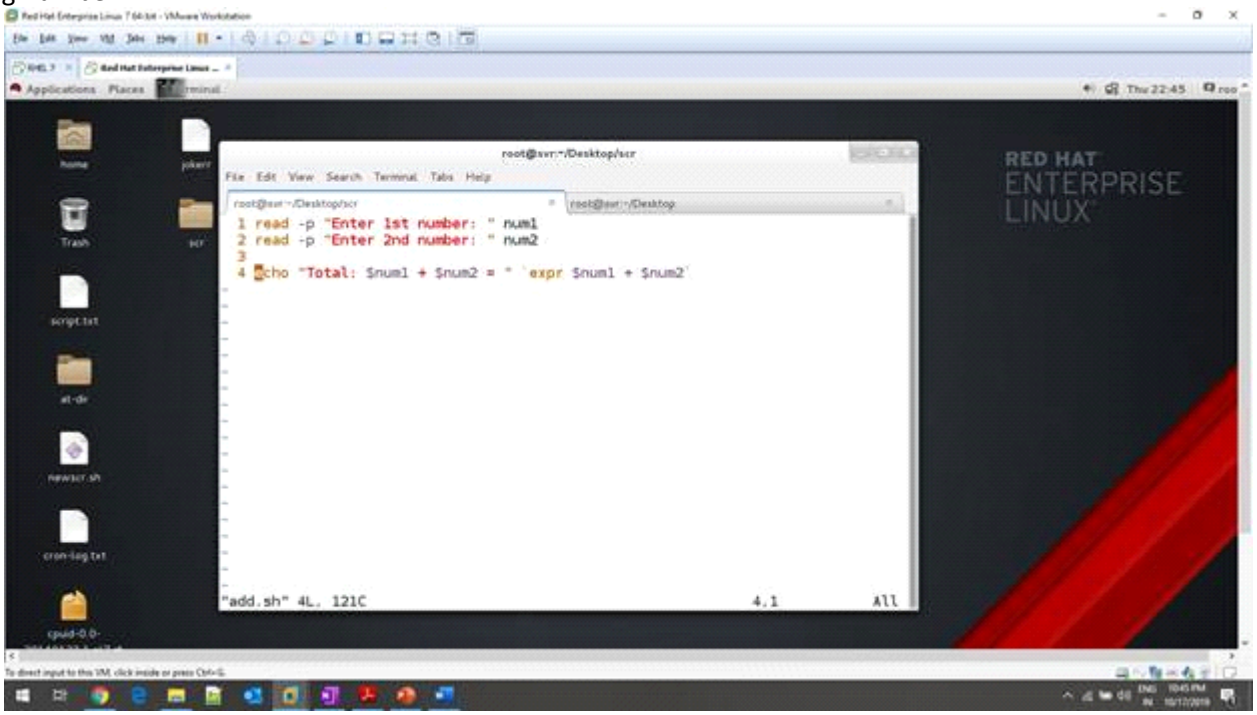
5. Reading from script



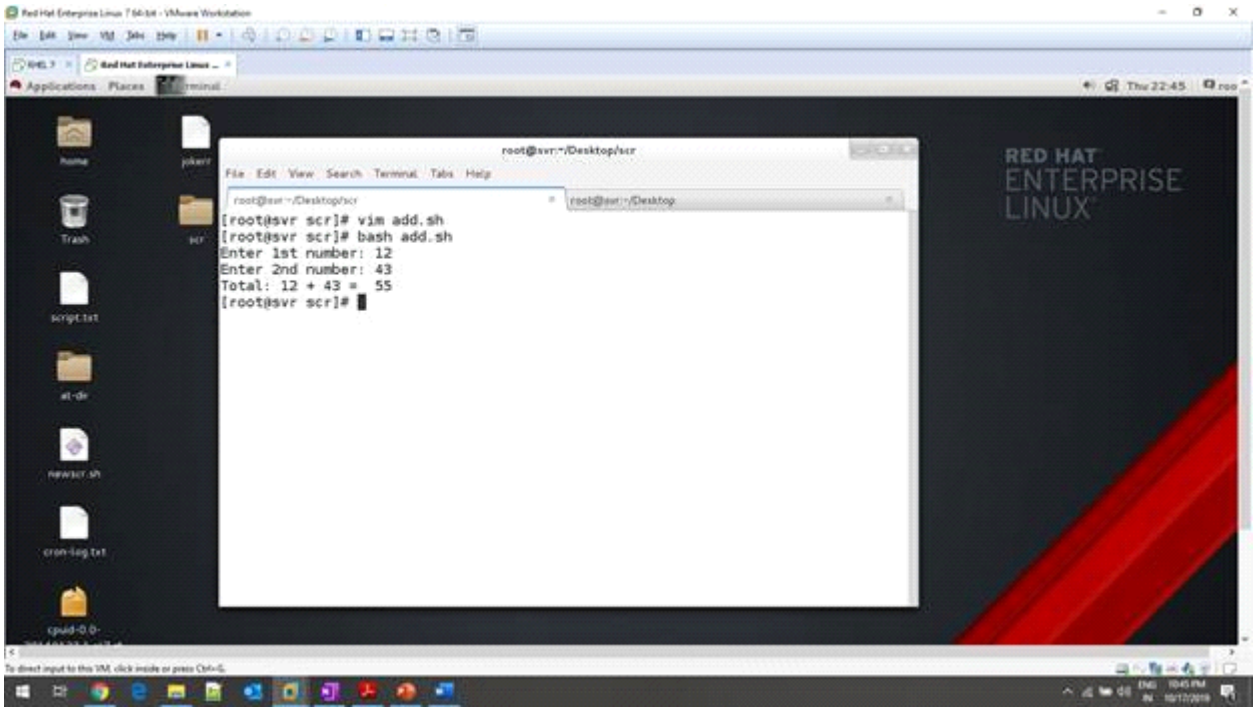
Output



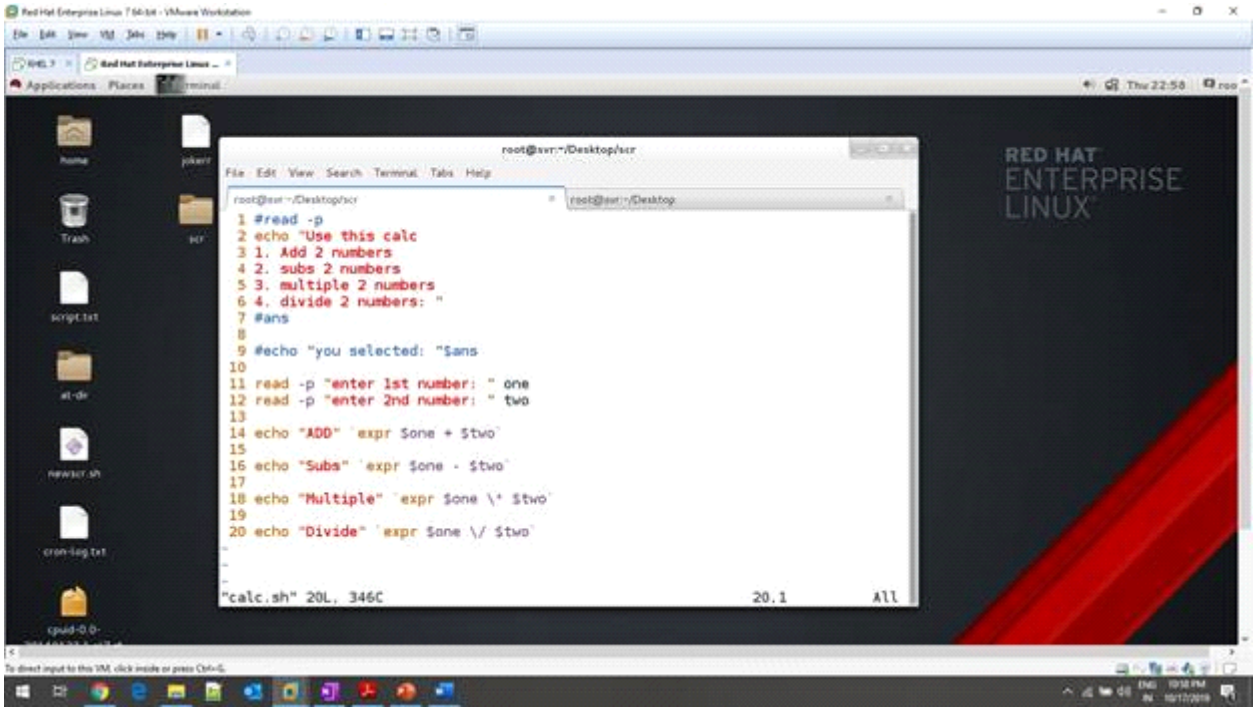
6. Adding number



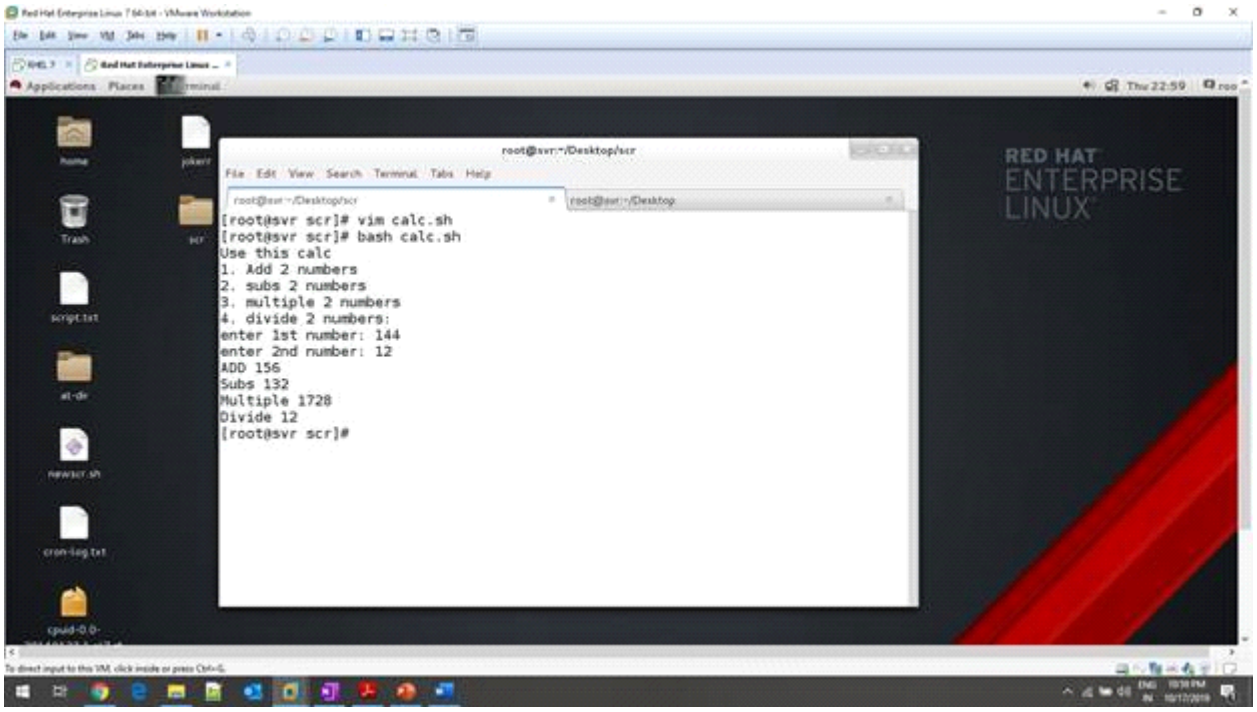
Output



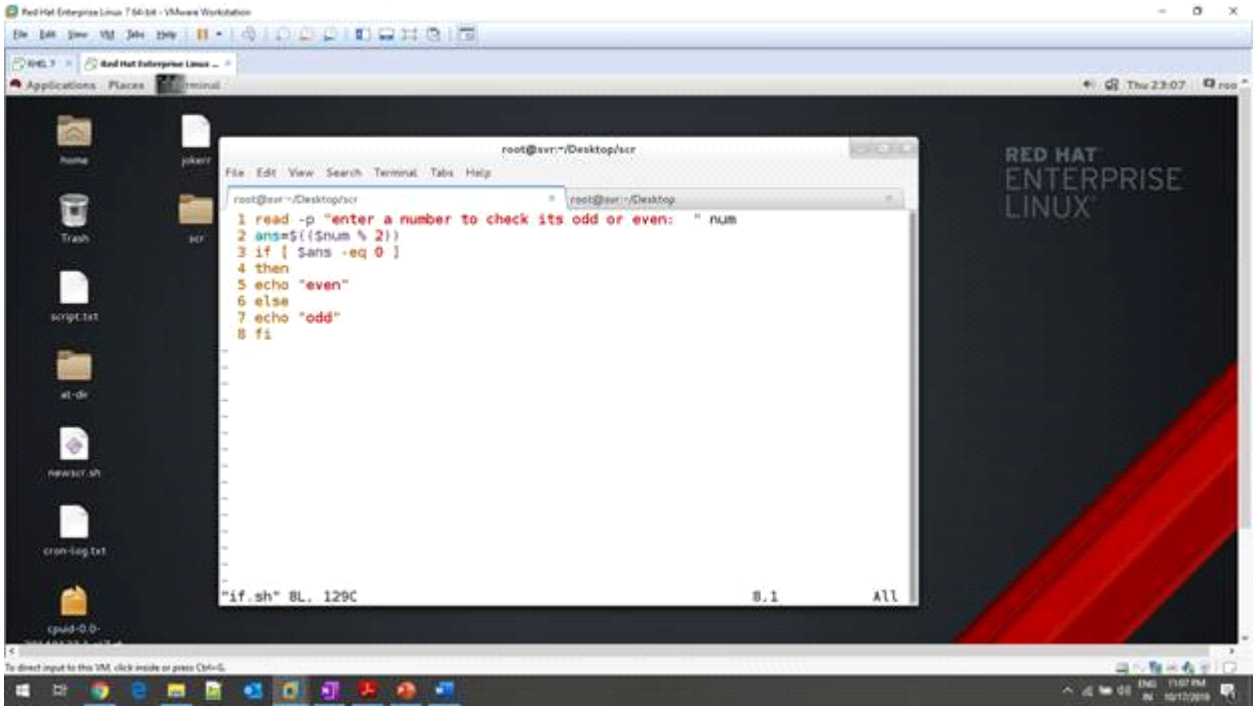
7. Calculator



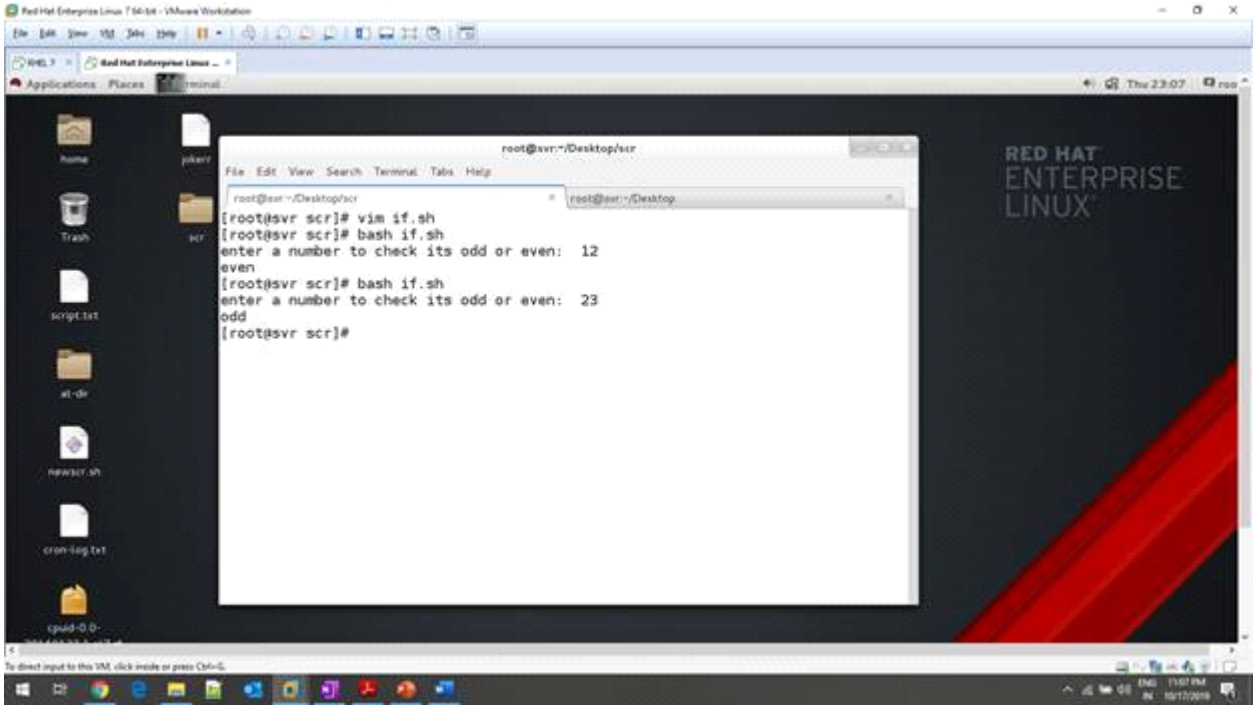
Output



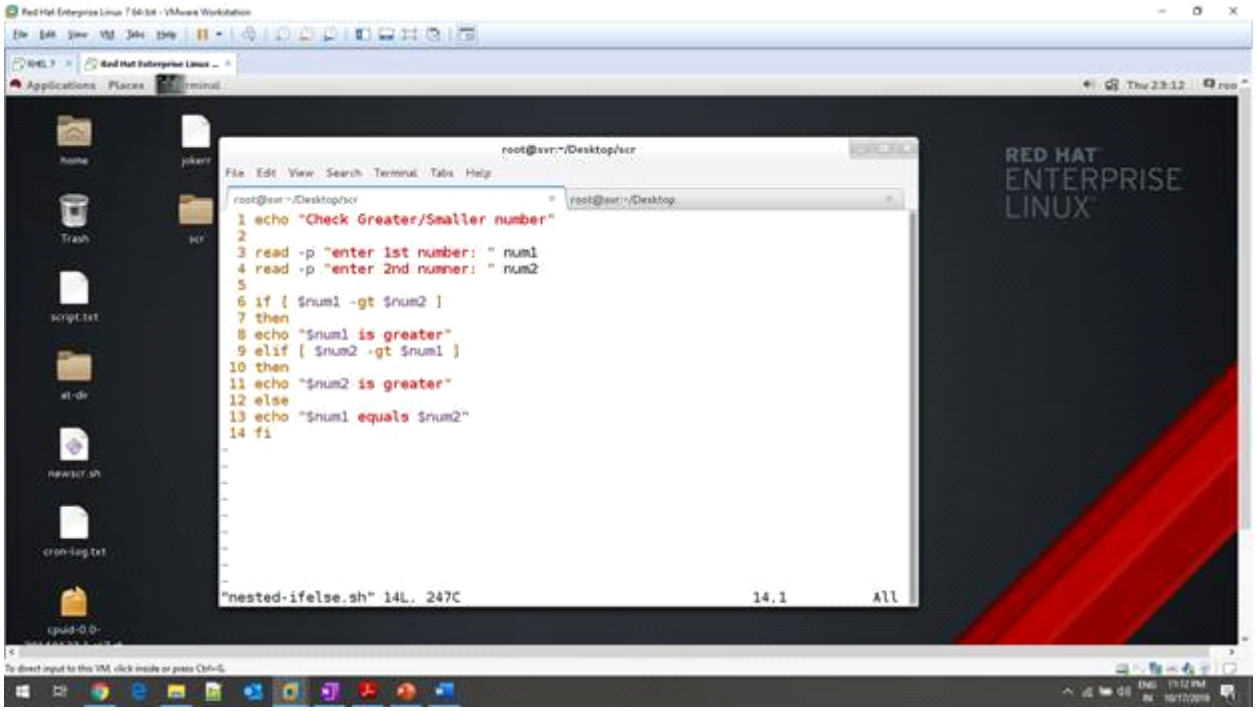
8. Odd/Even



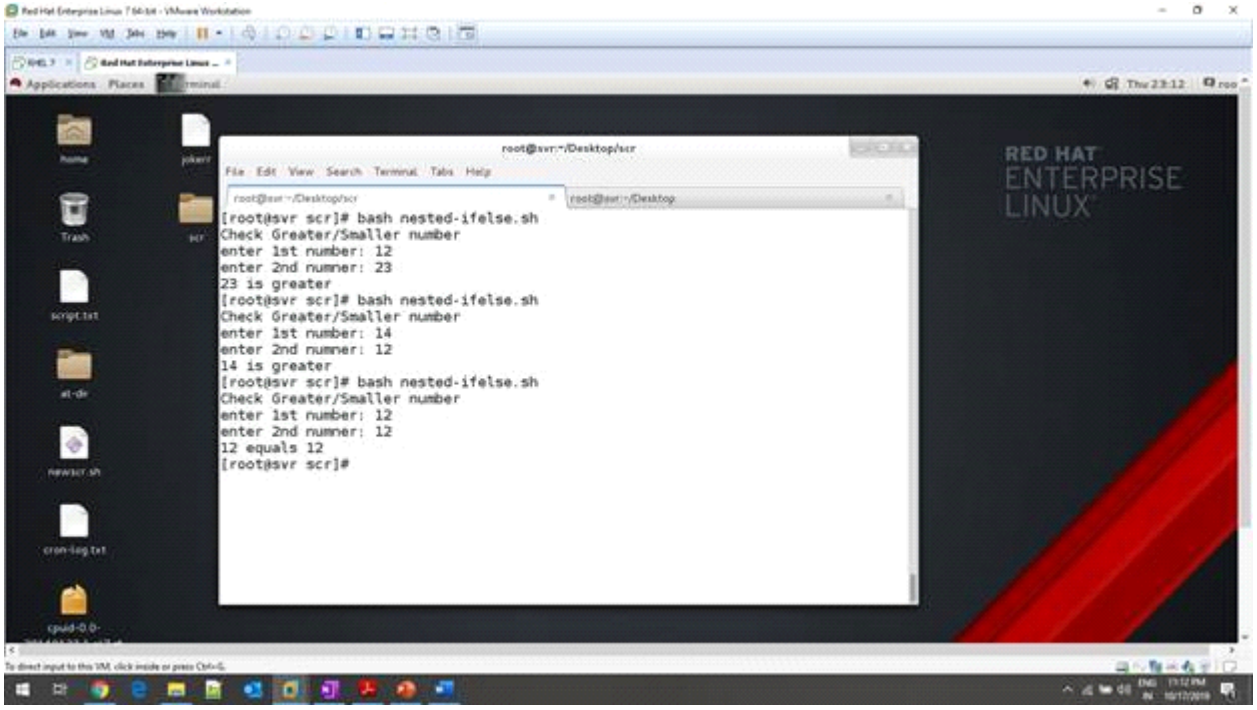
Output



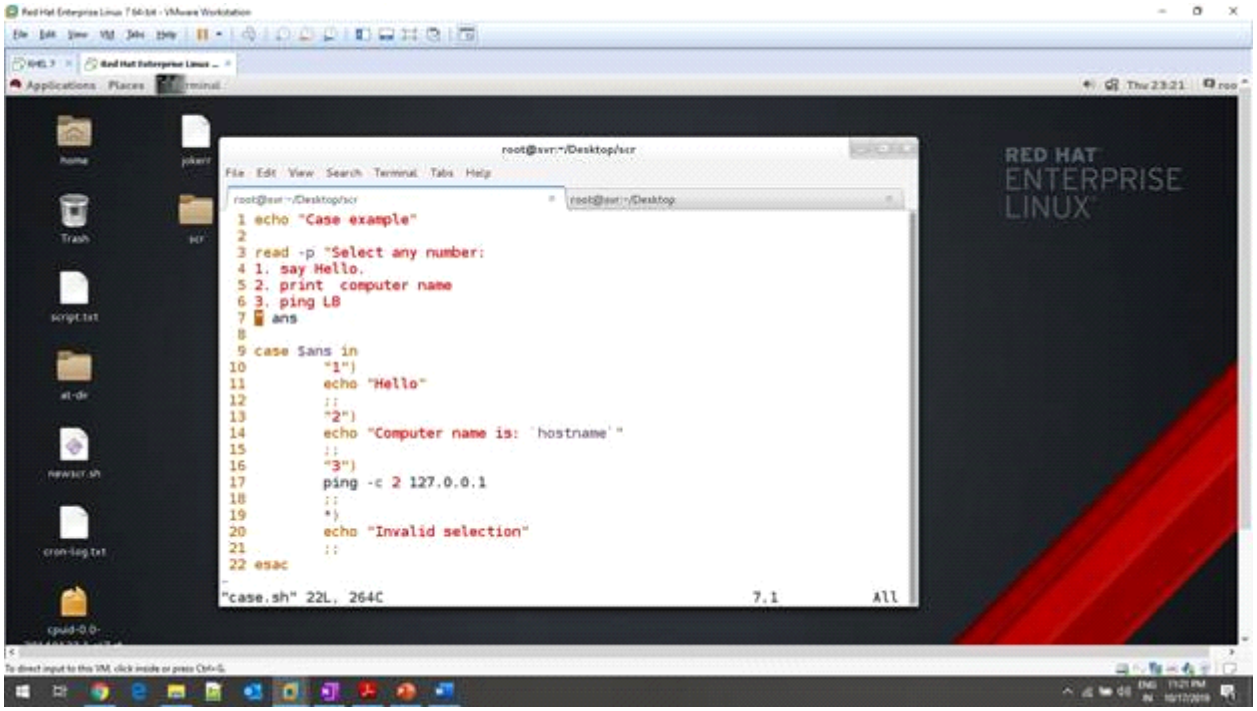
9. Greater/smaller



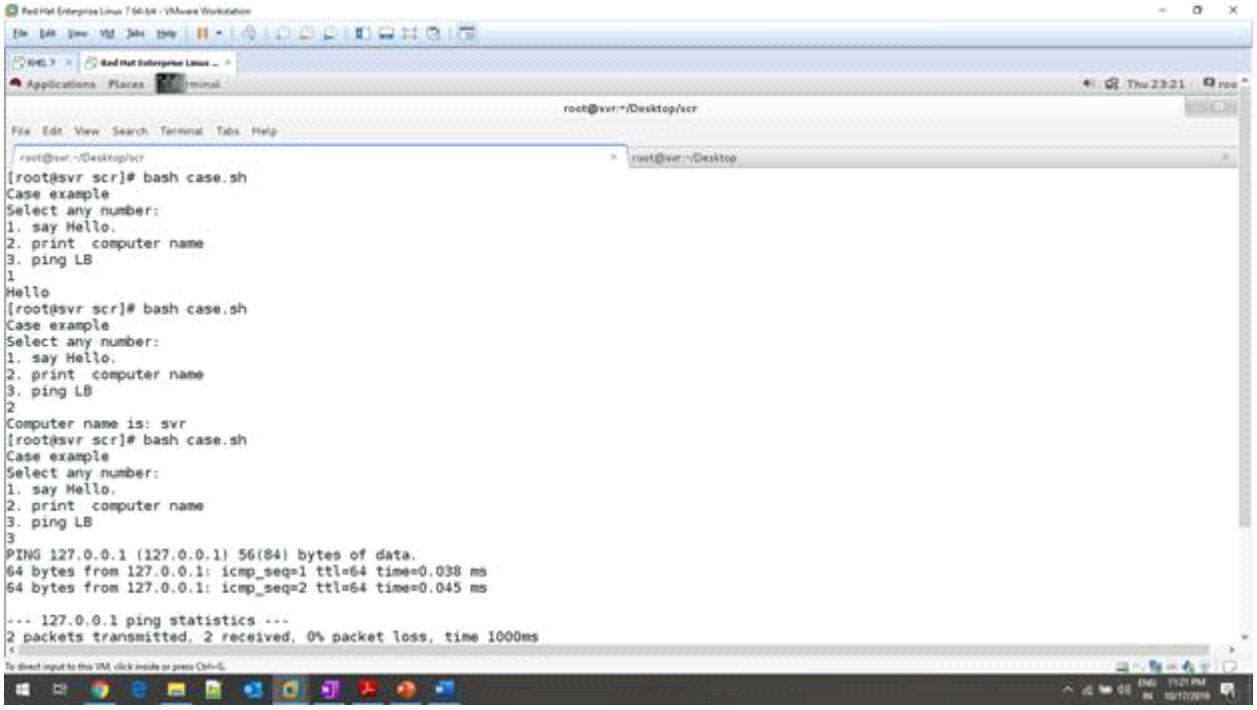
Output



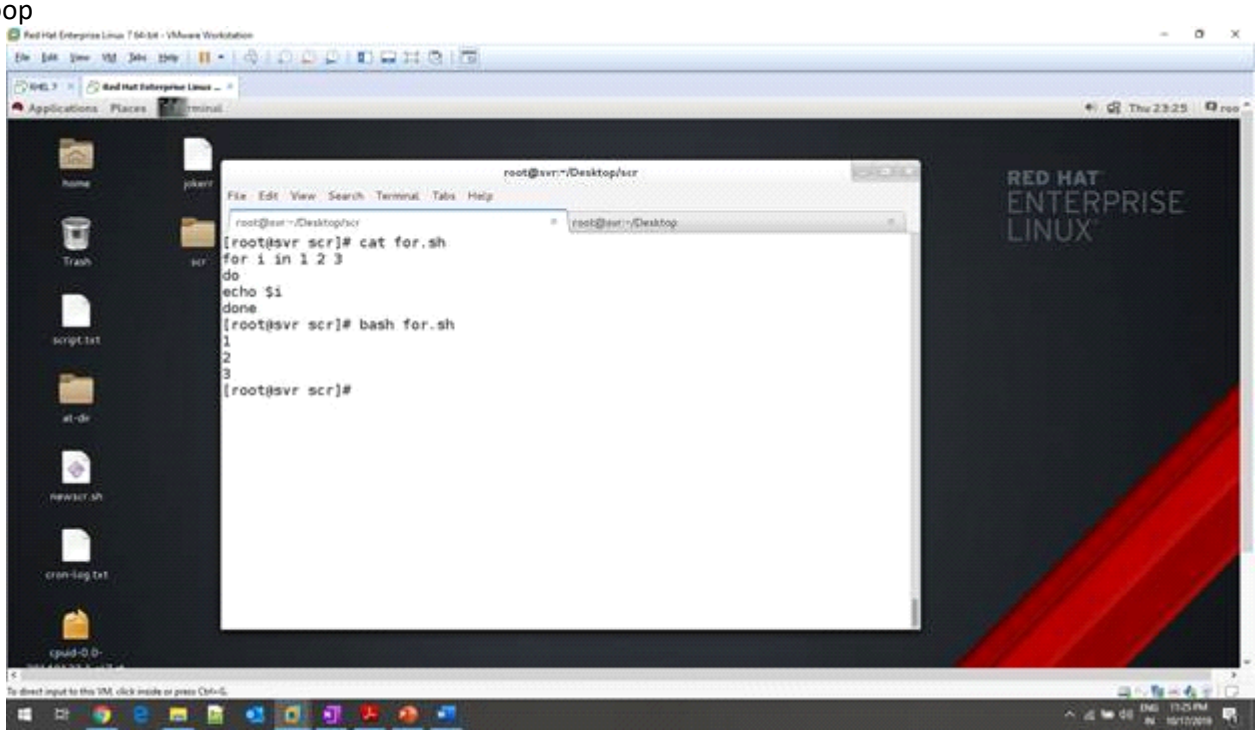
10. CASE statement



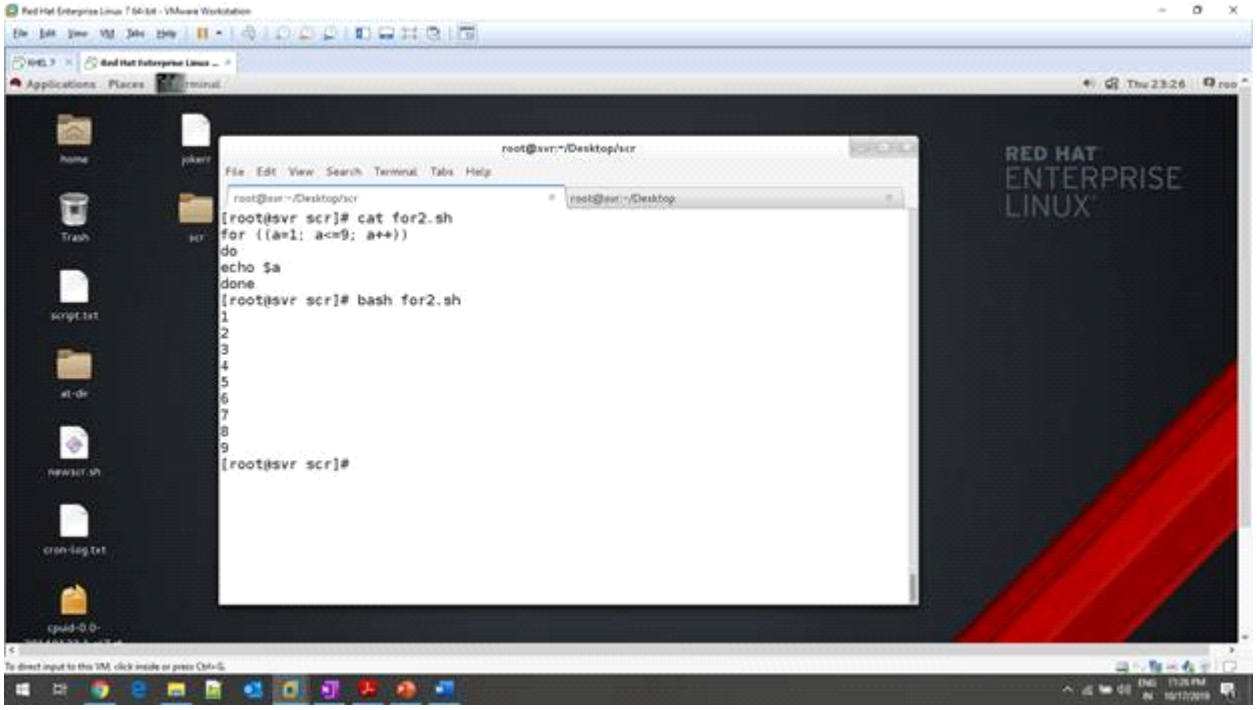
Output



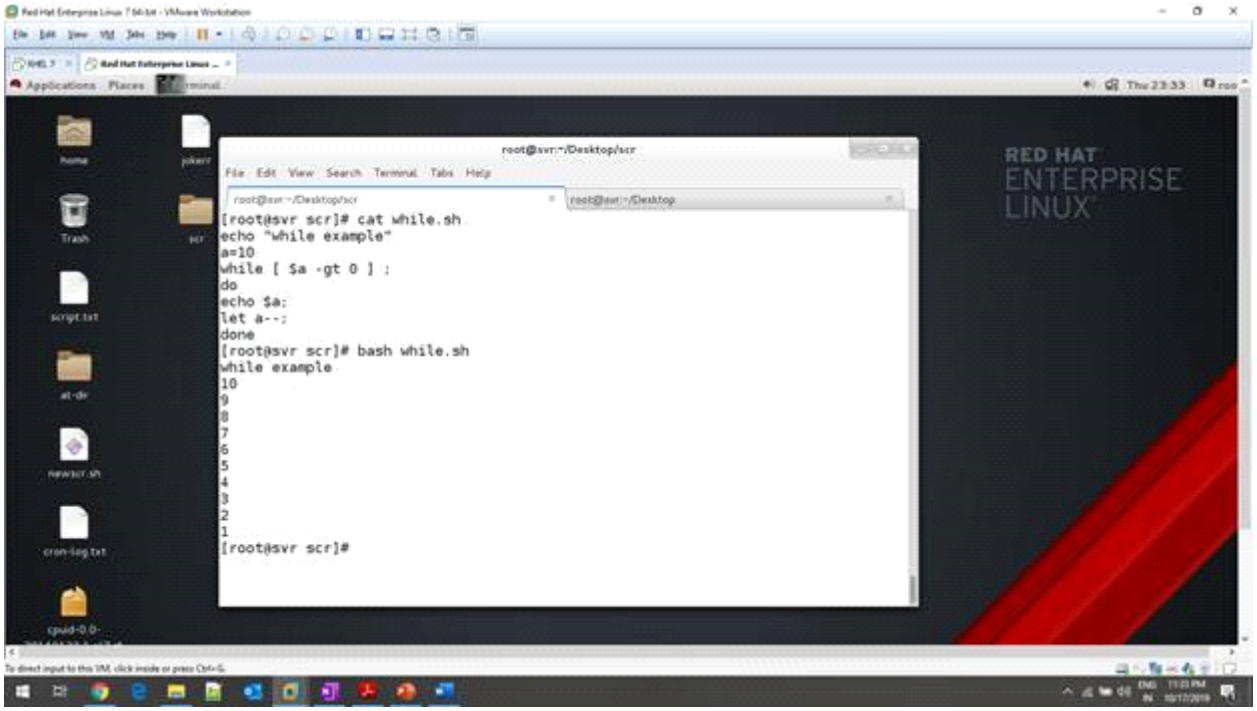
11. For loop



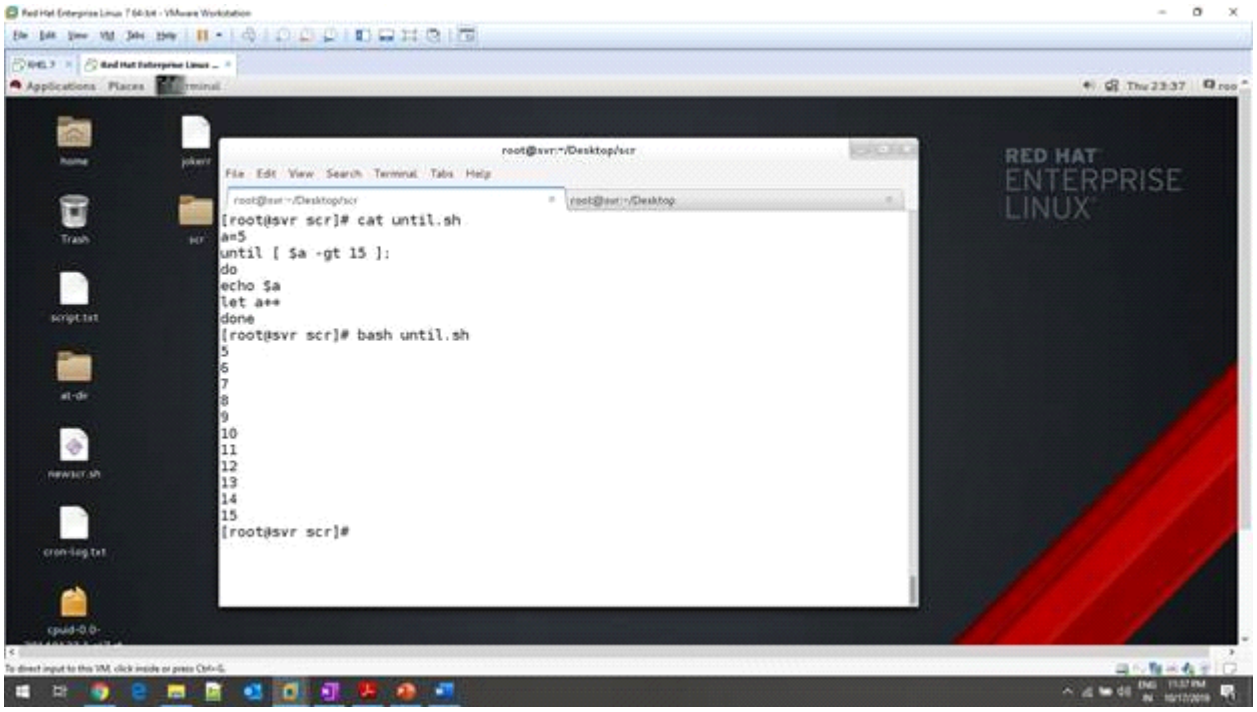
12. For – 2



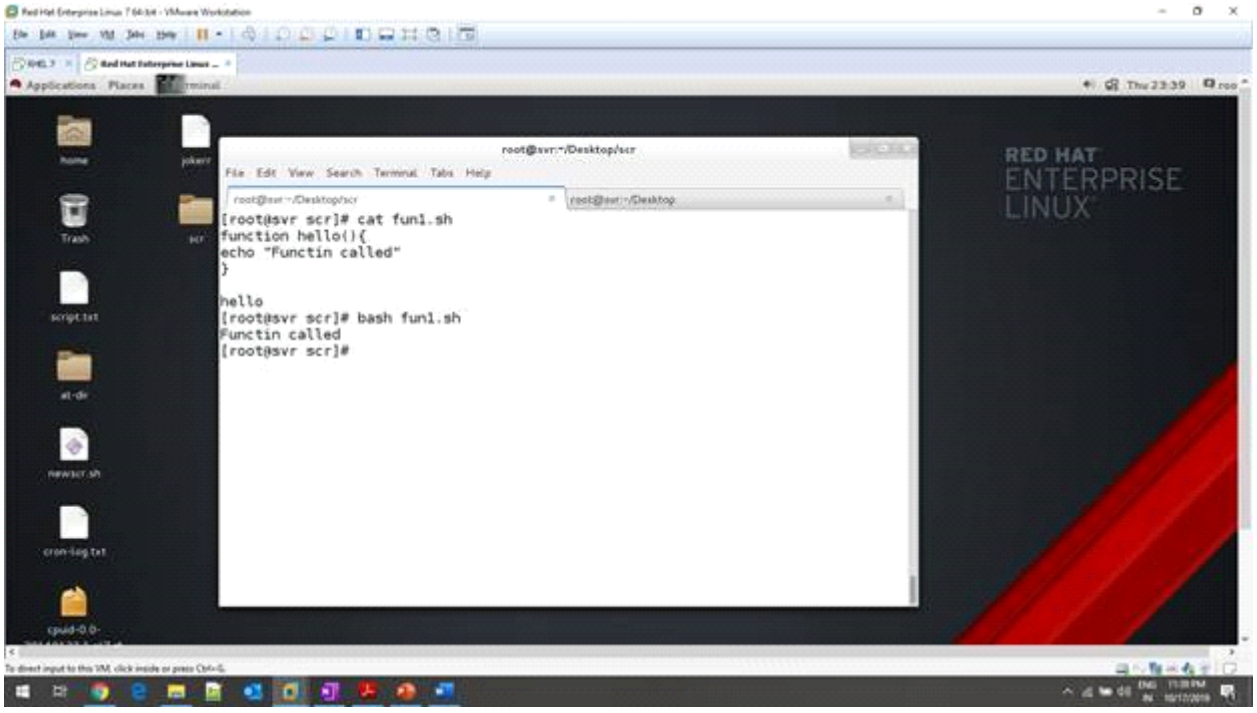
13. While



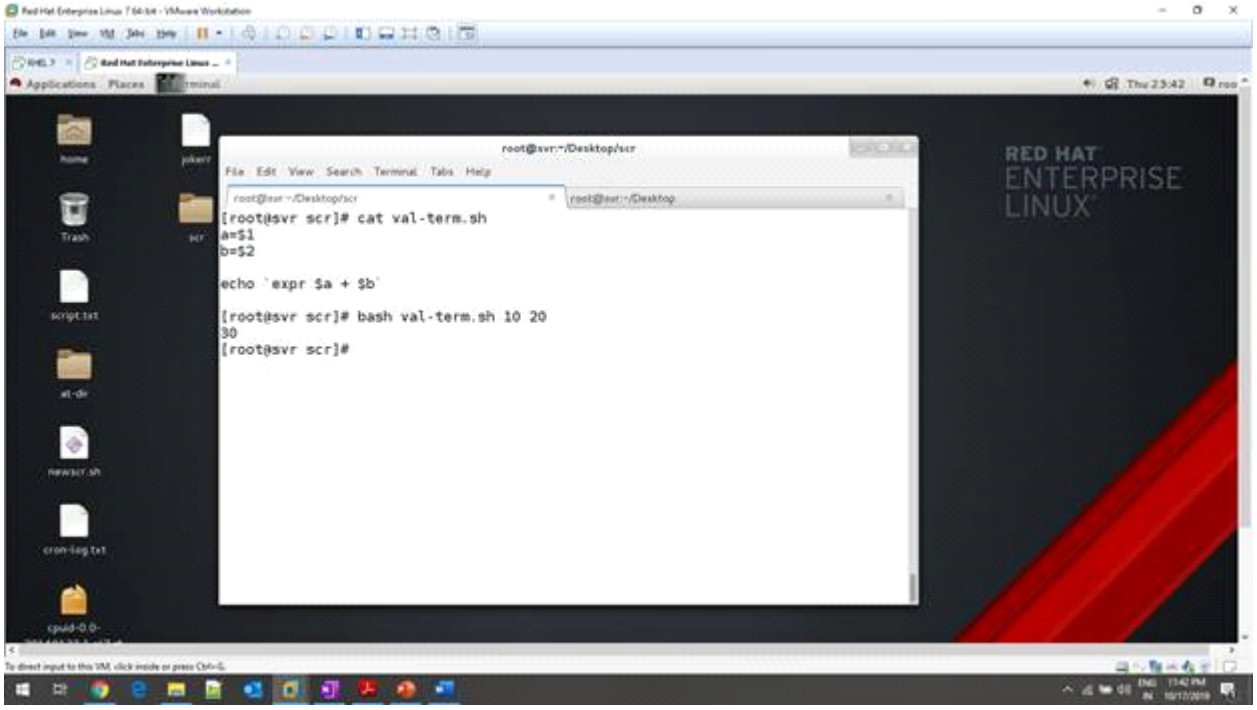
14. Until (at 1c)



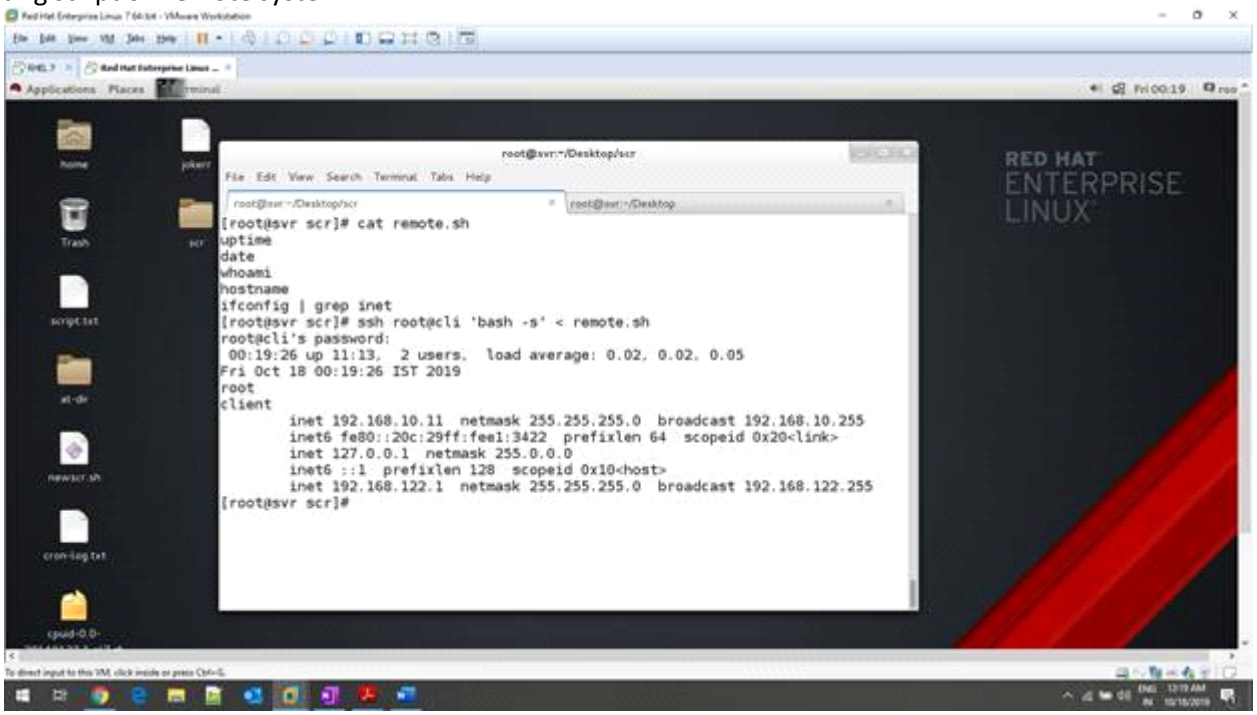
15. Functions



16. Passing values from terminal



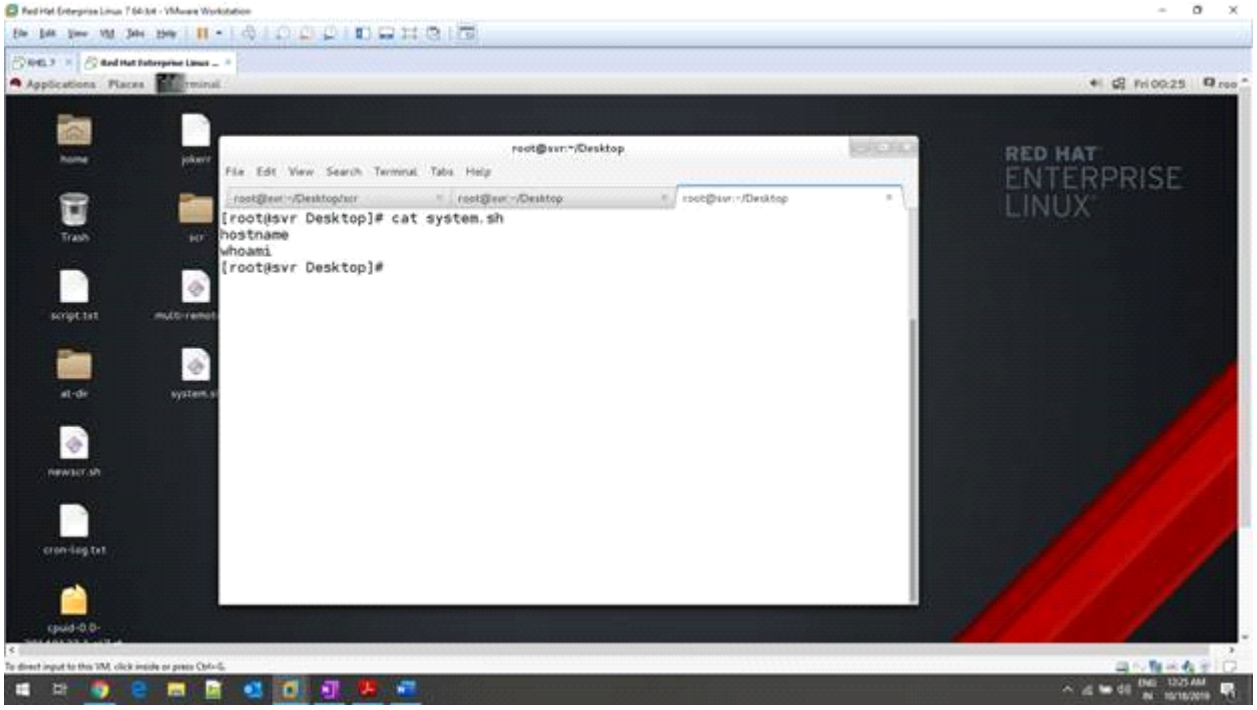
17. Executing script on remote system



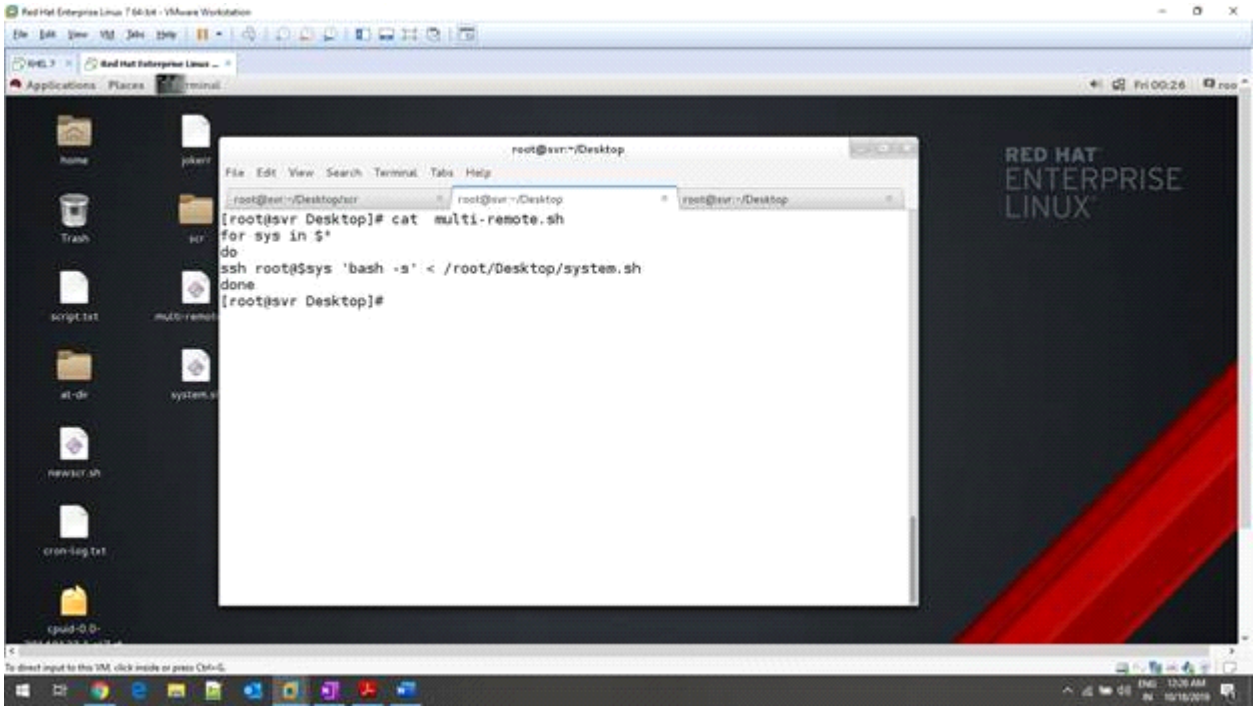
18. Executing script on different system with SSH

    Script: system.sh (cmds-> hostname, whoami)

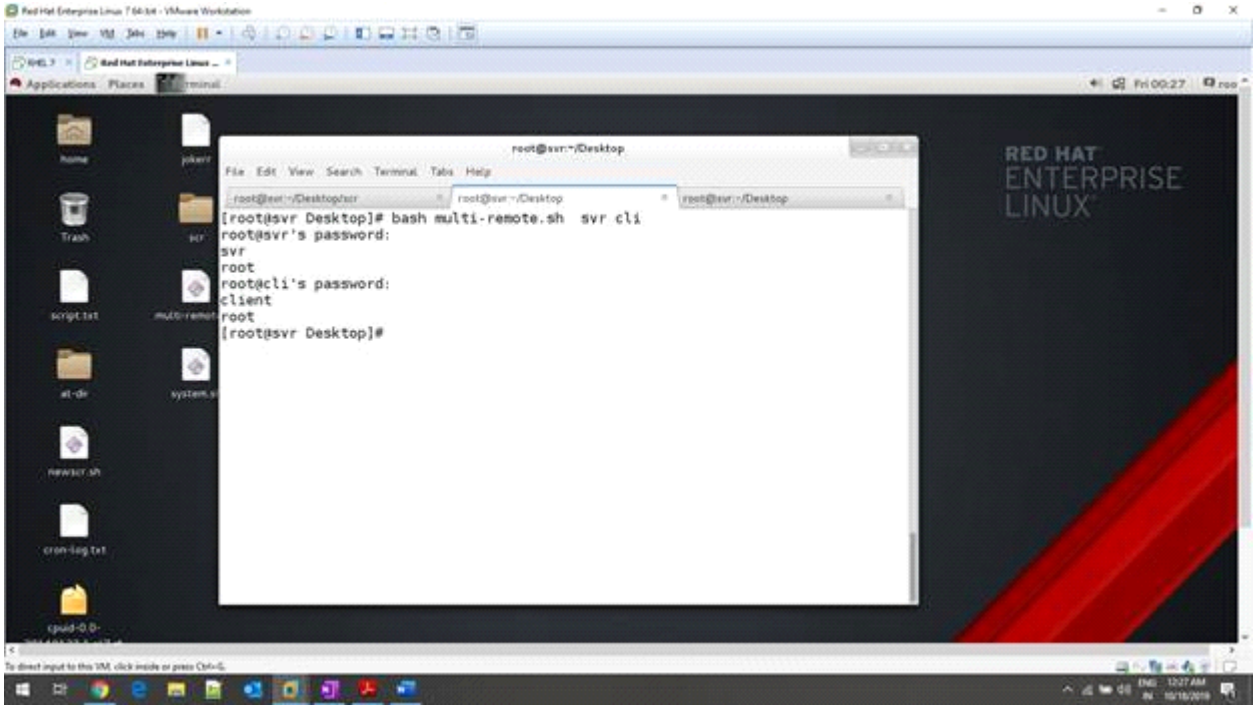
    Other cmd to exe: ps -eo cmd,pid,ppid,%mem,%cpu --sort=-%mem | head -n 6



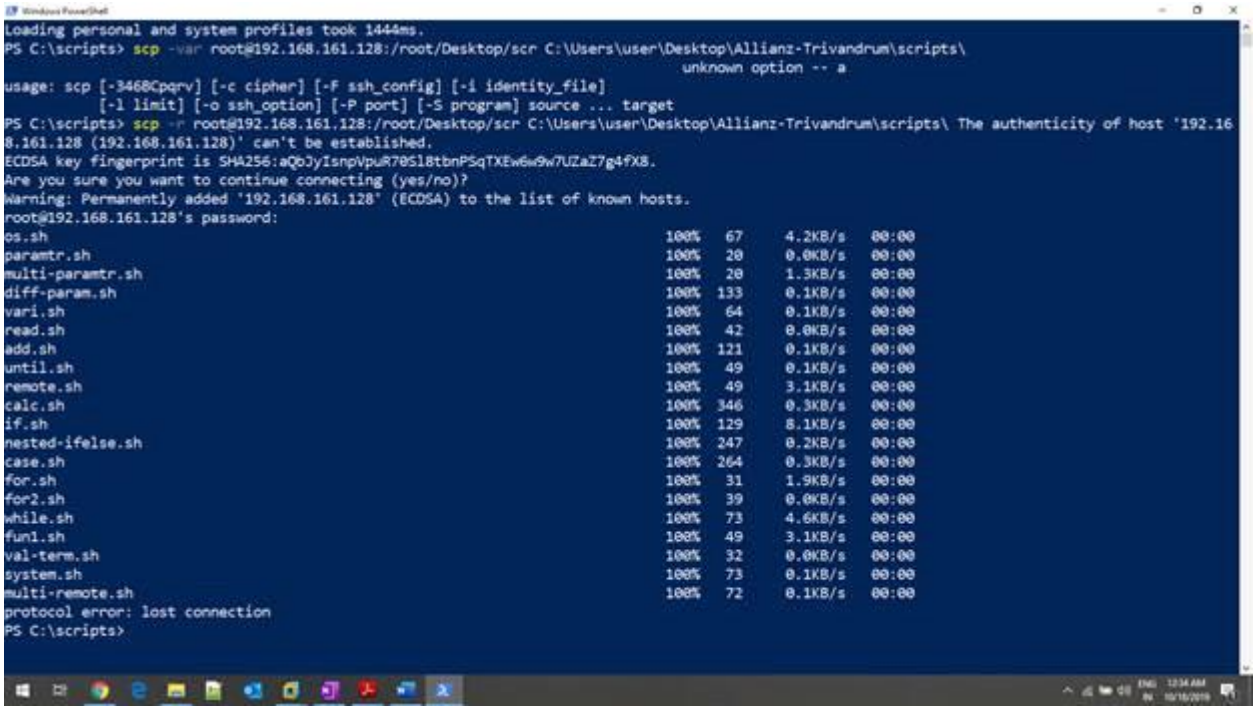
Actual script (multi-sys.sh) to execute on all systems:



Output



Scp win to linux



RHEL 8

Thursday, January 3, 2019 12:30 PM

https://computingforgeeks.com/red-hat-enterprise-linux-rhel-8-new-features/

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8-beta/html/managing_systems_using_the_cockpit_web_interface/getting-started-with-cockpit_system-management-using-cockpit

https://linuxconfig.org/install-gnome-on-redhat-8

Samba server

Saturday, March 3, 2018 9:27 AM

Packages Required	samba-libs samba-winbind-krb5-locator samba-winbind-modules samba-vfs-glusterfs samba-winbind samba-client samba-common samba-winbind-clients Samba cifs-utils
Config Files	/etc/samba/smb.conf
Port Numbers	netbios-ns – 137/tcp # NETBIOS Name Service netbios-dgm – 138/tcp # NETBIOS Datagram Service netbios-ssn – 139/tcp # NETBIOS session service microsoft-ds – 445/tcp # if you are using Active Directory Other ports: Port 389 (TCP) – for LDAP (Active Directory Mode) Port 445 (TCP) – NetBIOS was moved to 445 after 2000 and beyond, (CIFS) Port 901 (TCP) – for SWAT service (not related to client communication)
Service	# setsebool -P samba_export_all_ro=1 samba_export_all_rw=1 # getsebool –a | grep samba_export # semanage fcontext –at samba_share_t "/finance(/.*)?" # restorecon /finance
Commands	https://www.tecmint.com/setup-samba-file-sharing-for-linux-windows-clients/
Firewall rules	# firewall-cmd --permanent --add-port=137/tcp # firewall-cmd --permanent --add-port=138/tcp # firewall-cmd --permanent --add-port=139/tcp # firewall-cmd --permanent --add-port=445/tcp //or # firewall-cmd --permanent --add-service=samba # firewall-cmd --reload

Creating Samba server

Installing samba server packages

#yum install -y samba samba-client

Make a directory for sharing

#mkdir /home/shareDir

Give permissions

#chmod 777 /home/shareDir

Edit samba configuration file

near line 66: add follows

unix charset = UTF-8

dos charset = CP932

```
# line 90: change (Windows' default)
        workgroup = WORKGROUP

# line 96: uncomment and change IP address you allow
        hosts allow = 127. 10.0.0.

# line 126: add ( no auth )
        security = user
        passdb backend = tdbsam
        map to guest = Bad User

# add follows to the end
        [Share]                                 # any name you like
    path = /home/share              # shared directory
    writable = yes                          # writable
    guest ok = yes                          # guest allowed
    guest only = yes                # guest only
    create mode = 0777              # fully accessed file
    directory mode = 0777

#create a group & a user for group

        ========================================
Samba Configuration - another config
----------------------
a. packages required
        a. samba client
        b. cisf utils

b. creating dir to be mounted
        #mkdir /myshare

c. configuring user conf file for user & password
        #vim /root/pass.txt

d. configuring /etc/fstab
        #vim /etc/fstab
        at last
        //192.168.0.1/multi                 /myshare          cifs
        credentials=/root/pass.txt,multiuser,sec=ntlmssp          0 0
```

Samba working (R7)

Friday, October 18, 2019 7:30 AM

Total systems required: 2
Samba server: RHEL 7 (192.168.161.128 - NAT - Pinging to windows)
Samba client: win10 (192.168.1.19 - NAT - pinging to Linux)

Samba server configuration:

1. Install required packages
   #yum install -y samba*
   #yum install -y samba samba-client cifs-utils

2. Create a group to share
   Group: grp1
   Users in the group: u1, u2
   Directory to share: /myshare
   Permission on directory: 777
   Group owner for directory: grp1

3. Add firewall rule for samba
   #firewall-cmd --permanent --add-service=samba
   #firewall-cmd --reload

4. Configure samba config file
   #vim /etc/samba/smb.conf

   #append below content in the last line
   [myshare]
   comment=directory to share data between win & Linux
   browsable=yes
   path=/myshare
   valid user=@grp1
   writable=yes

   :wq!

5. Verify the configuration
   #testparm

   #<press enter>

6. Add samba users:
   #smbpasswd -a <user1>
   #smbpasswd -a <user2>

7. Restart the samba service
   #systemctl enable smb
   #systemctl start smb
   #systemctl status smb

8. Verify smb for client
   #smblient -L localhost -U <user1>
   #smbclient -L localhost -U <user2>

9. Mount SAMBA share for windows
   Go to THIS PC.
   Click on MAP NETWORK DRIVE
   In FOLDER:
   \\<linux-samba-IP-address>\<samba-share>
   \\192.168.161.128\myshare
   Select CONNECT USING DIFFERENT CREDENTIALS

10. Mount samba to Linux client
    #mount //<samba-server-IP>/<samba-share> /media/samba -o username=user1

    #mount //192.168.10.10/myshare /media/samba -o username=user1

Linux Storage

Monday, October 28, 2019 8:53 PM

What is Block Storage?
Block storage is another name for what the Linux kernel calls a block device. A block device is a piece of hardware that can be used to store data, like a traditional spinning hard disk drive (HDD), solid state drive (SSD), flash memory stick, etc. It is called a block device because the kernel interfaces with the hardware by referencing fixed-size blocks, or chunks of space.

So basically, block storage is what you think of as regular disk storage on a computer. Once it is set up, it basically acts as an extension of the current filesystem tree, and you can write to or read information from the drive seamlessly.

What are Disk Partitions?
Disk partitions are a way of breaking up a storage drive into smaller usable units. A partition is a section of a storage drive that can be treated in much the same way as a drive itself.

Partitioning allows you to segment the available space and use each partition for a different purpose. This gives the user a lot of flexibility allowing them to potentially segment their installation for easy upgrading, multiple operating systems, swap space, or specialized filesystems.

While disks can be formatted and used without partitioning, some operating systems expect to find a partition table, even if there is only a single partition written to the disk. It is generally recommended to partition new drives for greater flexibility down the road.

MBR vs GPT
When partitioning a disk, it is important to know what partitioning format will be used. This generally comes down to a choice between MBR (Master Boot Record) and GPT (GUID Partition Table).

MBR is the traditional partitioning system, which has been in use for over 30 years. Because of its age, it has some serious limitations. For instance, it cannot be used for disks over 2TB in size, and can only have a maximum of four primary partitions. Because of this, the fourth partition is typically set up as an “extended partition”, in which “logical partitions” can be created. This allows you to subdivide the last partition to effectively allow additional partitions.

GPT is a more modern partitioning scheme that attempts to resolve some of the issues inherent with MBR. Systems running GPT can have many more partitions per disk. This is usually only limited by the restrictions imposed by the operating system itself. Additionally, the disk size limitation does not exist with GPT and the partition table information is available in multiple locations to guard against corruption. GPT can also write a “protective MBR” which tells MBR-only tools that the disk is being used.

Formatting and Filesystems
While the Linux kernel can recognize a raw disk, the drive cannot be used as-is. To use it, it must be formatted. Formatting is the process of writing a filesystem to the disk and preparing it for file operations. A filesystem is the system that structures data and controls how information is written to and retrieved from the underlying disk. Without a filesystem, you could not use the storage device for any file-related operations.

There are many different filesystem formats, each with trade-offs across a number of different dimensions, including operating system support. On a basic level, they all present the user with a similar representation of the disk, but the features that each supports and the mechanisms used to

enable user and maintenance operations can be very different.

Some of the more popular filesystems for Linux are:

- **Ext4:** The most popular default filesystem is Ext4, or the fourth version of the extended filesystem. The Ext4 filesystem is journaled, backwards compatible with legacy systems, incredibly stable, and has mature support and tooling. It is a good choice if you have no specialized needs.
- **XFS**: XFS specializes in performance and large data files. It formats quickly and has good throughput characteristics when handling large files and when working with large disks. It also has live snapshotting features. XFS uses metadata journaling as opposed to journaling both the metadata and data. This leads to fast performance, but can potentially lead to data corruption in the event of an abrupt power loss.
- **Btrfs:** Btrfs is modern, feature-rich copy-on-write filesystem. This architecture allows for some volume management functionality to be integrated within the filesystem layer, including snapshots, cloning, volumes, etc. Btrfs still runs into some problems when dealing with full disks. There is some debate over its readiness for production workloads and many system administrators are waiting for the filesystem to reach greater maturity.
- **ZFS**: ZFS is a copy-on-write filesystem and volume manager with a robust and mature feature set. It has great data integrity features, can handle large filesystem sizes, has typical volume features like snapshotting and cloning, and can organize volumes into RAID and RAID-like arrays for redundancy and performance purposes. In terms of use on Linux, ZFS has a controversial history due to licensing concerns. Ubuntu is now shipping a binary kernel module for it however, and Debian includes the source code in its repositories. Support across other distributions is yet to be determined.

How Linux Manages Storage Devices

Device Files in /dev

In Linux, almost everything is represented by a file. This includes hardware like storage drives, which are represented on the system as files in the /dev directory. Typically, files representing storage devices start with sd or hd followed by a letter. For instance, the first drive on a server is usually something like /dev/sda.

Partitions on these drives also have files within /dev, represented by appending the partition number to the end of the drive name. For example, the first partition on the drive from the previous example would be /dev/sda1.

While the /dev/sd* and /dev/hd* device files represent the traditional way to refer to drives and partitions, there is a significant disadvantage of in using these values by themselves. The Linux kernel decides which device gets which name on each boot, so this can lead to confusing scenarios where your devices change device nodes.

To work around this issue, the /dev/disk directory contains subdirectories corresponding with different, more persistent ways to identify disks and partitions on the system. These contain symbolic links that are created at boot back to the correct /dev/[sh]da* files. The links are named according to the directory's identifying trait (for example, by partition label in for the /dev/disk/by-partlabel directory). These links will always point to the correct devices, so they can be used as static identifiers for storage spaces.

Some or all of the following subdirectories may exist under /dev/disk:

- **by-label**: Most filesystems have a labeling mechanism that allows the assignment of arbitrary user-specified names for a disk or partition. This directory consists of links that named after these user-supplied labels.
- **by-uuid:** UUIDs, or universally unique identifiers, are a long, unique string of letters and numbers that can be used as an ID for a storage resource. These are generally not very human-

readable, but are pretty much guaranteed to be unique, even across systems. As such, it might be a good idea to use UUIDs to reference storage that may migrate between systems, since naming collisions are less likely.
- **by-partlabel** and by-partuuid: GPT tables offer their own set of labels and UUIDs, which can also be used for identification. This functions in much the same way as the previous two directories, but uses GPT-specific identifiers.
- **by-id:** This directory contains links generated by the hardware's own serial numbers and the hardware they are attached to. This is not entirely persistent, because the way that the device is connected to the system may change its by-id name.
- **by-path**: Like by-id, this directory relies on the storage devices connection to the system itself. The links here are constructed using the system's interpretation of the hardware used to access the device. This has the same drawbacks as by-id as connecting a device to a different port can alter this value.

Mounting Block Devices
The device file within /dev are used to communicate with the Kernel driver for the device in question. However, a more helpful abstraction is needed in order to treat the device as a segment of available space.

In Linux and other Unix-like operating systems, the entire system, regardless of how many physical devices are involved, is represented by a single unified file tree. As such, when a filesystem on a drive or partition is to be used, it must be hooked into the existing tree. Mounting is the process of attaching a formatted partition or drive to a directory within the Linux filesystem. The drive's contents can then be accessed from that directory.

Drives are almost always mounted on dedicated empty directories (mounting on a non-empty directory means that the directory's usual contents will be inaccessible until the drive is unmounted). There are many different mounting options that can be set to alter the behavior of the mounted device. For example, the drive can be mounted in read-only mode to ensure that its contents won't be altered.

The Filesystem Hierarchy Standard recommends using /mnt or a subdirectory under it for temporarily mounted filesystems. If this matches your use case, this is probably the best place to mount it. It makes no recommendations on where to mount more permanent storage, so you can choose whichever scheme you'd like. In many cases, /mnt or /mnt subdirectories are used for more permanent storage as well.

Making Mounts Permanent with /etc/fstab
Linux systems look at a file called /etc/fstab (filesystem table) to determine which filesystems to mount during the boot process. Filesystems that do not have an entry in this file will not be automatically mounted (the exception being those defined by systemd .mount unit files, although these are not common at the moment).

The /etc/fstab file is fairly simple. Each line represents a different filesystem that should be mounted. This line specifies the block device, the mount point to attach it to, the format of the drive, and the mount options, as well as a few other pieces of information.

What is RAID?
RAID stands for redundant array of independent disks. RAID is a storage management and virtualization technology that allows you to group drives together and manage them as a single unit with additional capabilities.

The characteristics of a RAID array depend on its RAID level, which basically defines how the disks in the array relate to each other. The level chosen has an impact on the performance and redundancy of the set. Some of the more common levels are:

- **RAID 0**: This level indicates drive striping. This means that as data is written to the array, it is split up and distributed among the disks in the set. This offers a performance boost as multiple disks can be written to or read from simultaneously. The downside is that a single drive failure can lose all of the data in the entire array, since no one disk contains enough information about the contents to rebuild.
- **RAID 1**: RAID 1 is basically drive mirroring. Anything written to a RAID 1 array is written to multiple disks. The main advantage is data redundancy, which allows data to survive hard drive lose in either side of the mirror. Because multiple drives contain the same data, usable capacity is reduced half.
- **RAID 5**: RAID 5 stripes data across multiple drives, similar to RAID 0. However, this level also implements a distributed parity across the drives. This basically means that if drive fails, the remaining drives can rebuild the array using the parity information shared between them. The parity information is enough to rebuild any one disk, meaning the array can survive any one disk loss. The parity information reduces the available space in the array by the capacity of one disk.
- **RAID 6**: RAID 6 has the same properties as RAID 5, but provides double parity. This means that RAID 6 arrays can withstand the loss of any 2 drives. The capacity of the array is again affected by the parity amount, meaning that the usable capacity is reduced by two disks worth of space.
- **RAID 10**: RAID 10 is a combination of levels 1 and 0. First, two sets of mirrored arrays are made. Then, data is striped across them. This creates an array that has some redundancy characteristics while providing good performance. This requires quite a few drives however, and the total capacity is half of the combined disk space.

What is LVM?
LVM, or Logical Volume Management, is a system that abstracts the physical characteristics of the underlying storage devices in order to provide increased flexibility and power. LVM allows you to create groups of physical devices and manage it as if it were one single block of space. You can then segment the space as needed into logical volumes, which function as partitions.

LVM is implemented on top of regular partitions, and works around many of the limitations inherent with classical partitions. For instance, using LVM volumes, you can easily expand partitions, create partitions that span multiple drives, take live snapshots of partitions, and moving volumes to different physical disks. LVM can be used in conjunction with RAID to provide flexible management with traditional RAID performance characteristics.

Startup scripts - R7

Monday, November 11, 2019 10:50 AM

1. **Let us first create a sample custom script to be run at system boot automatically.**

```
# vi /var/tmp/test_script.sh
#!/bin/bash
echo "This is a sample script to test auto run during boot" > /var/tmp/script.out
echo "The time the script run was -->  `date`" >> /var/tmp/script.out
```

2. **Add execute permission(if it's not already set).**

```
# chmod +x /var/tmp/test_script.sh
```

3. **Creating new systemd service unit**

```
# vi /etc/systemd/system/sample.service
[Unit]
Description=Description for sample script goes here
After=network.target

[Service]
Type=simple
ExecStart=/var/tmp/test_script.sh
TimeoutStartSec=0

[Install]
WantedBy=default.target
```

After= : If the script needs any other system facilities (networking, etc), modify the [Unit] section to include appropriate After=, Wants=, or Requires= directives.
Type= : Switch Type=simple for Type=idle in the [Service] section to delay execution of the script until all other jobs are dispatched
WantedBy= : target to run the sample script in

4. **Enable the systemd service unit**

```
# systemctl daemon-reload
# systemctl enable sample.service
# systemctl start sample.service
# systemctl reboot
```

After reboot verify the scr

Whatis

Monday, November 11, 2019 11:36 AM

whatis command in Linux is used to get a one-line manual page descriptions. So this command search for the manual pages names and show the manual page description of the specified filename or argument.

Command	Meaning
whatis -d ls	prints the debugging information
whatis -v ls	prints verbose warning messages
whatis -r ls	interprets each of the name as a regular expression
whatis -w ls	interprets each name as a pattern containing shell style wildcards

Link: https://www.geeksforgeeks.org/whatis-command-in-linux-with-examples/

Inode

Monday, November 11, 2019 12:01 PM

Everything is a file, Linux and other Unix-like Operating systems maintain a consistency by treating everything as a file (even the hardware devices). The keyboard, mouse, printers, monitor, hard disk, processes, even the directories are treated as files in Linux. The regular files contain data such as text (text files), music, videos (multimedia files) etc. Set aside the regular data, there are some other data about these files, such as their size, ownership, permissions, timestamp etc. This meta-data about a file is managed with a data structure known as an inode (index node).

An inode is an entry in inode table, containing information (the metadata) about a regular file and directory. An inode is a data structure on a traditional Unix-style file system such as ext3 or ext4. Linux extended filesystems such as ext2 or ext3 maintain an array of these inodes: the inode table. This table contains list of all files in that filesystem. The individual inodes in inode table have a unique number (unique to that filesystem), the inode number. Diving deep into the inode, an inode stores:

- **File type**: regular file, directory, pipe etc.
- **Permissions to that file**: read, write, execute
- **Link count**: The number of hard link relative to an inode
- **User ID**: owner of file
- **Group ID**: group owner
- **Size of file**: or major/minor number in case of some special files
- **Time stamp**: access time, modification time and (inode) change time
- **Attributes**: immutable' for example
- **Access control list**: permissions for special users/groups
- Link to location of file
- Other metadata about the file

SED editor

Saturday, March 3, 2018 9:28 AM

SED can edit/modify rows in a file
Its a case sensitive lang
link -> https://www.youtube.com/watch?v=uVJpq_Tq-fE
SED commands in Linux
---
1. Mode of operation
2. filtering lines by linenumber
3. filtering lines by content
4. removing text within line
5. replacing text within line

#create a file 'my.csv'
1,"test1","des1","fn,mn,ln"
2,"test2","des2","fn,mn,ln"
3,"test3","des3","fn,mn,ln"
4,"test1","des4","fn,mn,ln"
5,"test5","des5","fn,mn,ln"

printing all data of file
#cat my.csv | sed -r ''

don't printing all data of file
#cat my.csv | sed -n -r '' //-n=negate

remove specific lines
#cat my.csv | sed -r '2,4d' //d=delete
show except line 2 to 4(including 4)

print specific lines
#cat my.csv | sed -n -r '2,4p' //p=print
show except line 2 to 4(including 4)

delete line having following expression
#cat my.csv | sed -r '/test1/d'

subsitute a word (1st occurence)
#cat my.csv | sed -r 's/oldWord/newWord/'

subsitute a word (all occurence)
#cat my.csv | sed -r 's/oldWord/newWord/g'

#

SSH config

Saturday, March 3, 2018 9:28 AM

Packages Required	# yum install -y openssh-server openssh-clients
Config Files	/etc/ssh/sshd_config
Port Numbers	22; 2292
Service	# systemctl start sshd # systemctl enable sshd # systemctl reload sshd
Commands	Configuration
Firewall rules	#semanage port -a -t ssh_port_t -p tcp 2292 #semanage port -l | grep ssh #firewall-cmd --permanent --zone=public --add-port=2292/tcp //or #firewall-cmd --permanent --zone=public --add-port=22/tcp # firewall-cmd --permanent --add-service ssh # firewall-cmd --reload

Change port number;

Install SSH package

- Yum install -y openssh-server

Enable, start & status SSH

- Chkconfig sshd on
- Service sshd start
- Service sshd status

Connecting to server

- #ssh -l <username> <server-IP-address>

Secure copy (SCP)

- #scp <Source-Username>@<IP>:/<path> <Destination-Username>@<IP>:/<path>

SCP Configuration

---

1. install OPENSSH package

    #yum install openssh

2. enable, start & status of sshd service

    #yum enable sshd

    #yum start sshd

    #yum status sshd

PERFORM THE SAME IN CLIENT SIDE ALSO

    server m/c

    client m/c

To connect from server m/c to client m/c

<from server>#ssh <client_IP>
provide various password & login

To copy any folder or file from client to server
#scp <local_file> <IP>:/<location>

To copy from server to client
#scp <user>@<serverIP>:/<location>/file <user>@<clientIP>:/<location>/

Ssh url: https://hackertarget.com/ssh-examples-tunnels/

SSH + rsync: https://www.digitalocean.com/community/tutorials/how-to-copy-files-with-rsync-over-ssh

SSH with port 2222

Sunday, October 13, 2019 6:53 PM

1. Open SSH config file:
   #vim /etc/sshd_config
   Search for line (usually its commented)
   Port 22
   & change it to
   Port 2222

   Save & quit the file

Firewall settings:

Add firewall R6	#iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 2222 -j ACCEPT
Add firewall R7	#firewall-cmd --permanent --zone=public --add-port=2222/tcp

2. Restart the SSH service
   #service sshd restart

3. Switch to client & type below comman
   #ssh -p <2222> <user>@<Ipaddress>

SSH PWD-less

Tuesday, May 1, 2018 11:02 AM

RHEL 7 passwd-less SSH(working):
Login to server:
Create a SSH user on both systems
#ssh-keygen
#ssh-copy-id <user>@<client>
#ssh user@client

#ssh-keygen -t rsa -b 4096

SSH Password less Login Using SSH Keygen in 5 Easy Steps

SVR = 192.168.10.10
CLI = 192.168.10.11

1. *On CLI machine*
   a. Create SSH key using
      # ssh-keygen -t rsa
      & provide details
2. *On SVR*, Create a directory for SSH user (jeetu)
      # ssh jeetu@192.168.10.10 mkdir -p .ssh

3. *On CLI*, Upload generated RSA key to SVR
      # cat .ssh/id_rsa | ssh jeetu@192.168.10.10 "cat >> .ssh/authorized_keys"
      # enter CLI passwd

4. *On SVR*, set permissions for the file to access
      # ssh jeetu@192.168.10.10 "chmod 700 .ssh; chmod 640 .ssh/authorized_keys"

5. *On CLI*, login in without password
      # ssh jeetu@192.168.10.10

SELinux

Monday, December 3, 2018 11:59 AM

- Its advance **'access control mechanism**' in Linux.
- Basically developed by US National Security Agency (NSA) to protect systems against tempering & malicious intrusion.
- SELinux implements MAC (Mandatory Access Control), on top of DAC (Discretionary Access Control) which is present in almost all Linux distributions.
    - DAC
        - Traditional security model (user(u), group(g), other(o), read, write & execute on a file or directory.

SELINUX - security built-ON kernel

#cat /etc/sysconfig/selinux

- Security Enhanced Linux
- Security of kernel of an OS
- built-in, by-default enabled

Operates in 3 modes

1. **Enforcing** - (means SELinux is active)
2. **Permissive** - (means partially active - generates log, but won't deny if attacked)
3. **Disabled** - (means SELinux is inactive)

Check SELinux status

#**getenforce**

SELinux status can be changed by using by (temporarily)

#setenforce 0 // set permissive mode
#setenforce 1 // set enforcing mode
#setenforce enforcing
#setenforce permissive

setting SELinux permanetly (disable)

#vim /etc/sysconfig/selinux

if while entering the value in the file any spell mistaken is done, kernel will automatically have permission press e, x, a

:wq!

if permanently SELinux is disable & again we have to give temporary changing status for SELinux it will generate ERROR.

Sticky bit

Tuesday, December 25, 2018 8:53 PM

https://www.thegeekstuff.com/2013/02/sticky-bit/?utm_source=feedburner

- Think of a scenario where you create a Linux directory that can be used by all the users of the Linux system for creating files. Users can create, delete or rename files according to their convenience in this directory. For all those who think that why would such a directory be created? There exists, for example, /tmp directory in the Linux system that can be used by different Linux users to create temporary files.

1. Create a group
2. Add 2 users in the group
3. Create a shared directory & link (chown) with the users.
4. Switch to both users 1 by 1 & create normal files
5. Now try to delete the file of other users. It will get deleted

Now to stop this, we use STICKY BIT

- #chmod +t <dir-name> //to add sticky bit
- #chmod -t <dir-name> //to remove sticky bit

Now try to delete the same file (as in step 5), this time you won't be able to delete the file.

To verify, type "ls -l" command & after permission you will find a "t" is added, which represents it’s a sticky bit.

SUDO user list

Saturday, March 3, 2018 9:28 AM

TO CHECK MULTIPLE SUDO USER IN LINUX MACHINE

#getent passwd | cut -f1 -d: | sudo xargs -L1 sudo -l -U | grep -v 'not allowed'

To check if the users is sudo with commands

#sudo -l -U <username>

To check if the users is sudo or not

#sudo -v //if not output, means it is having sudo permissions.

Or

#sudo ls

TCP Wrapper

Saturday, December 22, 2018 8:42 AM

- It's host-based networking ACL system.
- Used to filter network access to IP server on UNIX/Linux OS.
- It allows host, IP, names to be used as token on which access control will be implemented.
- Used for monitoring the UNIX systems from malicious activities of crackers.
- It provides another layer of security to Linux system.
- Its limited to TCP packets, not UDP (like audio or video) or ICMP (ping).
- To check list of services associated with TCP wrapper, use below command:
  - #***lsof /lib64/libwrap.so.0***
- TCP Wrapper configuration:
  - /etc/hosts.allow
  - /etc/hosts.deny
- TCP Wrapper file format (for allow & deny files):
  - Daemon : Clients
  - ALL : ALL
  - ALL daemons : all system/users
  - Example:
    - In.telnetd : .alpha.corp -> telnet allowed for alpha corp
    - Sshd : 192.168.10.10 -> ssh service is allowed for given IP.
    - In.telnetd : .alpha.corp EXCEPT crack.alpha.corp -> telnet denied for alpha corp
    - Sshd : 192.168.10 EXCEPT 192.168.10.12 -> ssh service is denied for given IP.

TigerVNC

Monday, April 9, 2018 4:40 PM

Packages Required	# yum groupinstall "GNOME Desktop" # yum install tigervnc-server xorg-x11-fonts-Type1
Config Files	Configuration; /etc/systemd/system/
Port Numbers	5900
Service	#systemctl start vncserver@:5.service #systemctl enable vncserver@:5.service
Commands	#cp /lib/systemd/system/vncserver@.service /etc/systemd/system/vncserver@:3.service
Firewall rules	# firewall-cmd --permanent --zone=public --add-port=5903/tcp //or #firewall-cmd --zone=public --permanent --add-service=vnc-server # firewall-cmd --reload

https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-vnc-remote-access-for-the-gnome-desktop-on-centos-7

https://www.thegeekdiary.com/how-to-install-and-configure-vnc-tigervnc-server-in-centos-rhel-7/

Step 1 — Creating Two User Accounts

First, we will create two user accounts. These accounts will remotely connect to our CentOS 7 server from VNC clients.

- joevnc
- janevnc

Run the following command to add a user account for **joevnc**:
sudo useradd -c "User Joe Configured for VNC Access" joevnc
Then run the passwd command to change **joevnc**'s password:
sudo passwd joevnc
The output will ask us for new password. Once supplied, the account will be ready for login:
Changing password for user joevnc.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
Next, create an account for **janevnc**:
sudo useradd -c "User Jane Configured for VNC Access" janevnc
Set the password for **janevnc**:
sudo passwd janevnc

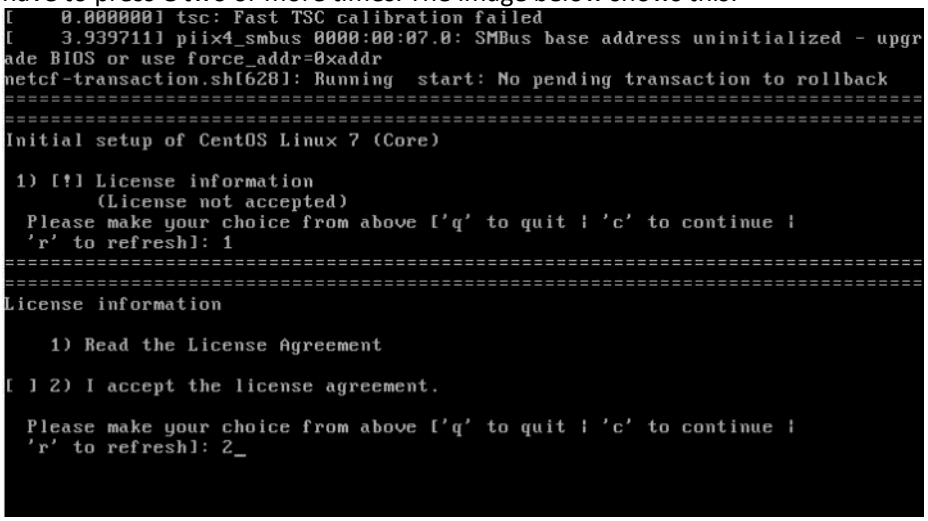
Step 2 — Installing GNOME Desktop

Now we will install GNOME desktop. GNOME is a collaborative effort: it's a collection of free and open source software that makes up a very popular desktop environment. There are other desktop environments like KDE, but GNOME is more popular. Our VNC users will use GNOME to interact with the server from its desktop:
sudo yum groupinstall -y "GNOME Desktop"
Depending on the speed of your network, this can take a few minutes.
Once the package group is installed, reboot the server:
sudo reboot

Troubleshooting — Server Stuck at Boot Phase

Depending on how your server has been set up, when the machine starts up it may remain in the boot phase showing a message like this:
Initial setup of CentOS Linux 7 (core)
1) [!] License information (Licence not accepted)

Please make your choice from above ['q' to quit | 'c' to continue | 'r' to refresh]:
To get past this, press **1** (license read), then **2** (accept licence), and then **C** (to continue). You may have to press **C** two or more times. The image below shows this:



If you don't see this error and the boot process is smooth, all the better – you can move on to the next step.

Step 3 — Installing TigerVNC Server

TigerVNC is the software that will allow us to make a remote desktop connection.
Install the Tiger VNC server:
sudo yum install -y tigervnc-server
This should show output like the following:
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
. . .
Running transaction
Installing : tigervnc-server-1.2.80-0.30.20130314svn5065.el7.x86_64
1/1
Verifying : tigervnc-server-1.2.80-0.30.20130314svn5065.el7.x86_64
1/1
Installed:
tigervnc-server.x86_64 0:1.2.80-0.30.20130314svn5065.el7
Complete!
Now we have VNC server and the GNOME desktop installed. We have also created two user accounts for connecting through VNC.

Step 4 — Configuring VNC Service for Two Clients

VNC server doesn't start automatically when it's first installed. To check this, run the following command:
sudo systemctl status vncserver@:.service
The output will be like this:
vncserver@:.service - Remote desktop service (VNC)
Loaded: loaded (/usr/lib/systemd/system/vncserver@.service; disabled)
Active: inactive (dead)
You can also run this command:
sudo systemctl is-enabled vncserver@.service
This should show output like this:
disabled
So why is it disabled? That's because each user will start a separate instance of the VNC service daemon. In other words, VNC doesn't run as one single process that serves every user request. Each user connecting via VNC will have to start a new instance of the daemon (or the system administrator can automate this).
CentOS 7 uses the systemd daemon to initiate other services. Each service that natively runs under systemd has a *service unit file* that's placed under the /lib/systemd/system directory by the yum installer. Processes that get started automatically at boot time have a link to this service unit file placed in the /etc/systemd/system/ directory.

In our case, a generic service unit file was created in the /lib/systemd/system/ directory, but no link was made under /etc/systemd/system/. To test this, run the following commands:
sudo ls -l /lib/systemd/system/vnc*
You should see:
-rw-r--r--. 1 root root 1744 Jun 10 16:15 /lib/systemd/system/vncserver@.service
Then check under /etc/systemd/system/:
sudo ls -l /etc/systemd/system/*.wants/vnc*
This one doesn't exist:
ls: cannot access /etc/systemd/system/*.wants/vnc*: No such file or directory
So, the first step is to start two new instances of VNC server for our two users. To do this, we will need to make two copies of the generic VNC service unit file under /etc/system/system. In the code snippet below, you're making two copies with two different names:
sudo cp /lib/systemd/system/vncserver@.service /etc/systemd/system/vncserver@:4.service
sudo cp /lib/systemd/system/vncserver@.service /etc/systemd/system/vncserver@:5.service
So why did we add two numbers (along with the colon) in the copied file names?
Again, that comes back to the concept of individual VNC services. VNC by itself runs on port 5900. Since each user will run their own VNC server, each user will have to connect via a separate port. The addition of a number in the file name tells VNC to run that service as a *sub-port* of 5900. So in our case, **joevnc**'s VNC service will run on port 5904 (5900 + 4) and **janevnc**'s will run on 5905 (5900 + 5).
Next edit the service unit file for each client. Open the /etc/systemd/system/vncserver@:4.service file with the **vi** editor:
sudo vi /etc/systemd/system/vncserver@:4.service
A look at the "Quick HowTo" section tells us we have already completed the first step. Now we need to go through the remaining steps. The comments also tell us that VNC is a non-trusted connection. We will talk about this later.
For now, edit the [Service] section of the file, replacing instances of <USER> with joevnc. Also, add the -geometry 1280x1024 clause at the end of the ExecStart parameter. This just tells VNC the screen size it should start in. You will modify two lines in total. Here's what the edited file should look like (note that the entire file is not shown):
```
# The vncserver service unit file
#
# Quick HowTo:
# 1. Copy this file to /etc/systemd/system/vncserver@:<display>.service
# 2. Edit <USER> and vncserver parameters appropriately
#   ("runuser -l <USER> -c /usr/bin/vncserver %i -arg1 -arg2")
# 3. Run `systemctl daemon-reload`
# 4. Run `systemctl enable vncserver@:<display>.service`
#
. . .
[Unit]
Description=Remote desktop service (VNC)
After=syslog.target network.target
[Service]
Type=forking
# Clean any existing files in /tmp/.X11-unix environment
ExecStartPre=/bin/sh -c '/usr/bin/vncserver -kill %i > /dev/null 2>&1 || :'
ExecStart=/sbin/runuser -l joevnc -c "/usr/bin/vncserver %i -geometry 1280x1024"
PIDFile=/home/joevnc/.vnc/%H%i.pid
ExecStop=/bin/sh -c '/usr/bin/vncserver -kill %i > /dev/null 2>&1 || :'
[Install]
WantedBy=multi-user.target
```
Save the file and exit vi.
Similarly, open the /etc/systemd/system/vncserver@:5.service file in vi and make the changes for user **janevnc**:
sudo vi /etc/systemd/system/vncserver@:5.service
Here's just the [Service] section with the changes marked:
```
[Service]
Type=forking
# Clean any existing files in /tmp/.X11-unix environment
```

ExecStartPre=/bin/sh -c '/usr/bin/vncserver -kill %i > /dev/null 2>&1 || :'
ExecStart=/sbin/runuser -l janevnc -c "/usr/bin/vncserver %i -geometry 1280x1024"
PIDFile=/home/janevnc/.vnc/%H%i.pid
ExecStop=/bin/sh -c '/usr/bin/vncserver -kill %i > /dev/null 2>&1 || :'
Next, run the following commands to reload the systemd daemon and also to make sure VNC starts up for two users at boot time.
sudo systemctl daemon-reload
Enable the first server instance:
sudo systemctl enable vncserver@:4.service
Output:
ln -s '/etc/systemd/system/vncserver@:4.service' '/etc/systemd/system/multi-user.target.wants/vncserver@:4.service'
Enable the second server instance:
sudo systemctl enable vncserver@:5.service
Output:
ln -s '/etc/systemd/system/vncserver@:5.service' '/etc/systemd/system/multi-user.target.wants/vncserver@:5.service'
Now you've configured two VNC server instances.

Step 5 — Configuring Your Firewall

Next, we will need to configure the firewall to allow VNC traffic through ports **5904** and **5905** only. CentOS 7 uses Dynamic Firewall through the firewalld daemon; the service doesn't need to restart for changes to take effect.
The firewalld service should start automatically at system boot time, but it's always good to check:
sudo firewall-cmd --state
This should show:
running
If the state is "not running" for any reason, execute the following command to make sure it's running:
sudo systemctl start firewalld
Now add the rules for ports 5904 and 5905:
sudo firewall-cmd --permanent --zone=public --add-port=5904-5905/tcp
Output:
success
Reload the firewall:
sudo firewall-cmd --reload
Output:
success

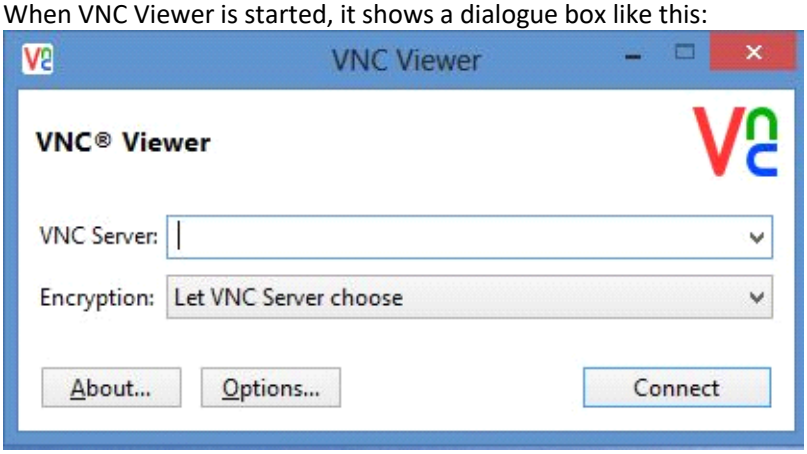
Step 6 — Setting VNC Passwords

We are one step away from seeing VNC in action. In this step, the users will need to set their **VNC passwords**. These are *not* the users' Linux passwords, but the passwords to log in to the VNC sessions.
Open another terminal connection to the CentOS 7 server, and this time log in as **joevnc**.
ssh joevnc@your_server_ip
Execute the following command:
vncserver
As shown in the output below, the server will ask **joevnc** to set up a VNC password. After typing in the password, the program also shows a number of files being created in the user's home directory:
You will require a password to access your desktops.
Password:
Verify:
xauth: file /home/joevnc/.Xauthority does not exist
New 'localhost.localdomain:1 (joevnc)' desktop is localhost.localdomain:1
Creating default startup script /home/joevnc/.vnc/xstartup
Starting applications specified in /home/joevnc/.vnc/xstartup
Log file is /home/joevnc/.vnc/localhost.localdomain:1.log
Let's look at the line New 'localhost.localdomain:1 (joevnc)' desktop is localhost.localdomain:1. **localhost.localdomain** was the server name in our example; in your case it could be different. Note the number after the server name: (**1**, separated by a colon). It's not the number in **joevnc**'s service

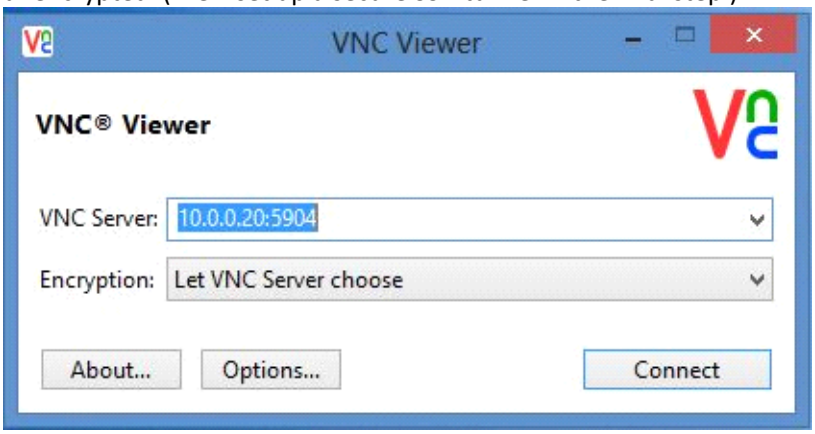
unit file (which was **4**). That's because this is the *display number* **joevnc**'s session will run on in this server, not the port number of the service (5904) itself.
Next open a new terminal session and log in as **janevnc**. Here as well, start the VNC server and set a password for **janevnc**:
vncserver
You should see similar output showing that **janevnc**'s session will run on display **2**.
Finally, reload the services from the **main terminal session**:
sudo systemctl daemon-reload
sudo systemctl restart vncserver@:4.service
sudo systemctl restart vncserver@:5.service

Step 7 — Connecting to Remote Desktops with a VNC Client

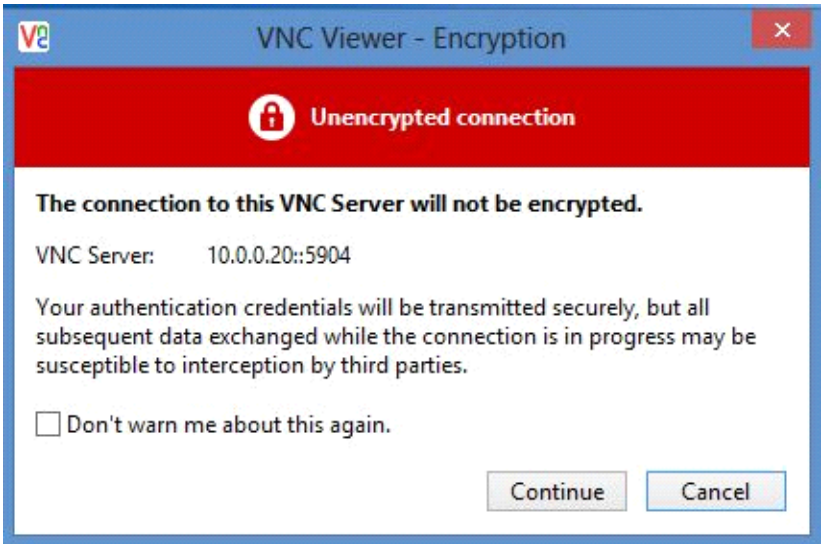
For this tutorial, we will assume users **joevnc** and **janevnc** are trying to connect to the CentOS 7 server from their Windows computers.
They will each need a VNC client for Windows to log into the remote desktop. This client is just like a terminal client like PuTTY, except it shows graphical output. There are various VNC client available, but the one we will use is RealVNC, available here. VNC Viewer for Mac OS X is available for download on the same page, and the Mac version is fairly similar to the Windows one.
When VNC Viewer is started, it shows a dialogue box like this:



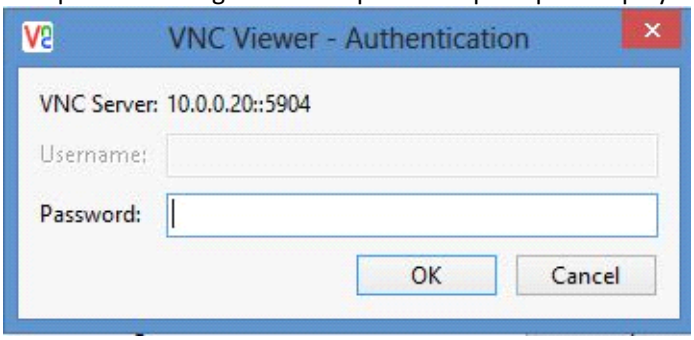
In the **VNC Server** field, add the IP address of your CentOS 7 server. Specify the port number 5904 after the server's IP, separate by a colon (**:**). We used 5904 because that's the VNC service port for **joevnc**.
We have also decided to let VNC Viewer choose the encryption method. This option will only encrypt the password sent across the network. Any subsequent communication with the server will be unencrypted. (We'll set up a secure SSH tunnel in the final step.)



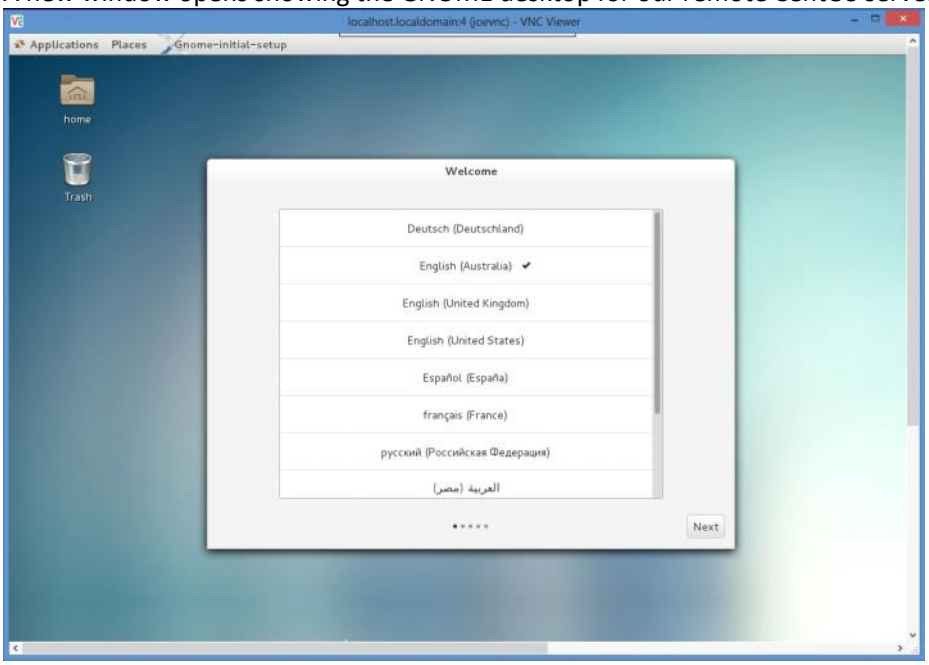
In fact, a warning message shows just that:



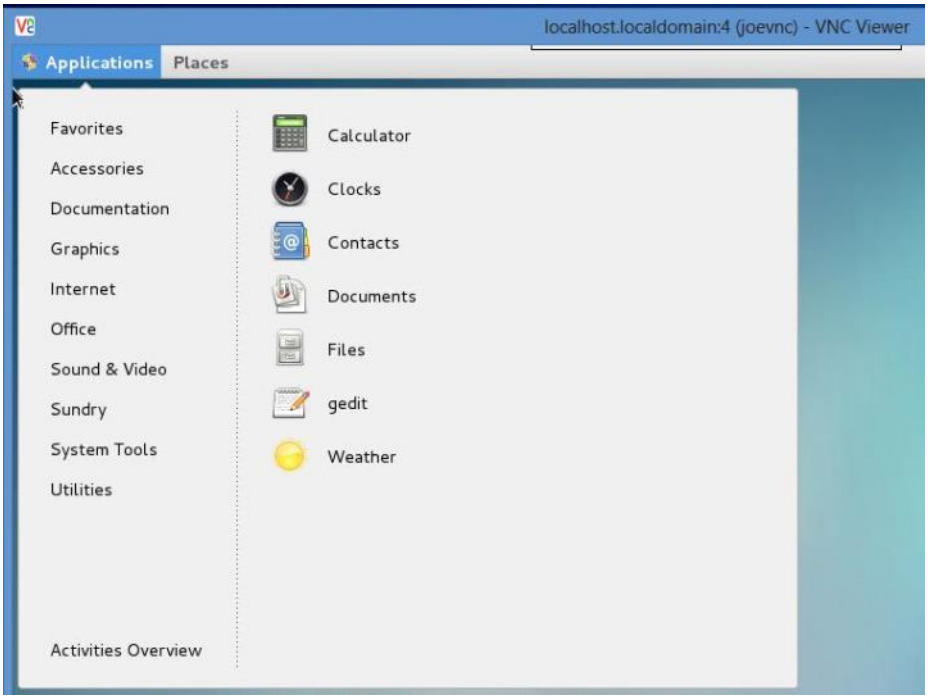
Accept the warning for now. A password prompt is displayed:

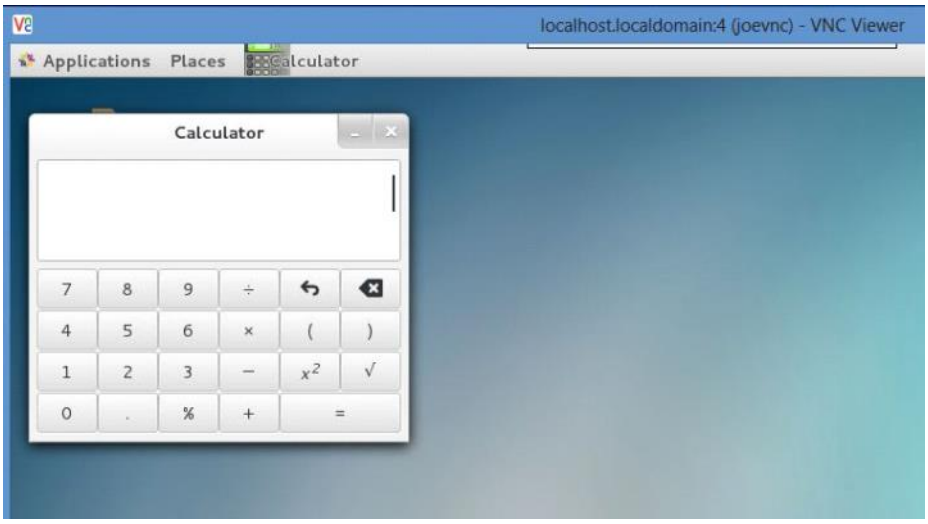


Enter **joevnc**'s VNC password that you set earlier.
A new window opens showing the GNOME desktop for our remote CentOS server:

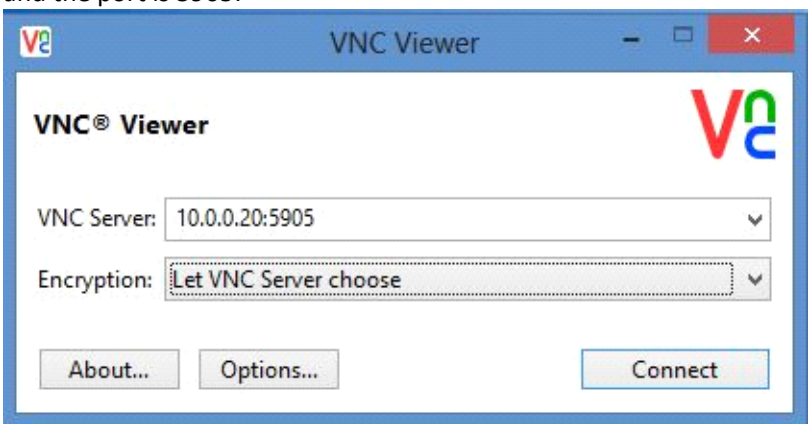


Accept the default welcome message.
Now **joevnc** can start a graphical tool like the GNOME calculator:





You can leave this desktop connection open.
Now **janevnc** can also start another VNC session with the CentOS server. The IP address is the same, and the port is 5905:



From <https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-vnc-remote-access-for-the-gnome-desktop-on-centos-7>

Users & Groups

Saturday, March 3, 2018 9:29 AM

https://www.tldp.org/LDP/lame/LAME/linux-admin-made-easy/shadow-file-formats.html
http://www.yourownlinux.com/2015/08/etc-shadow-file-format-in-linux-explained.html

Managing Users & Groups

Users
---
1. add a user
    adduser <username>
    useradd <username>

2. del a user
    userdel <username>

3. change user password
    passwd <username>

/etc/passwd

UID ranges

Specific UID numbers and ranges of numbers are used for specific purposes by Red Hat Enterprise Linux.

- *UID 0* is always assigned to the superuser account, **root**.
- *UID 1-200* is a range of "system users" assigned statically to system processes by Red Hat.
- *UID 201-999* is a range of "system users" used by system processes that do not own files on the file system. They are typically assigned dynamically from the available pool when the software that needs them is installed. Programs run as these "unprivileged" system users in order to limit their access to just the resources they need to function.
- *UID 1000+* is the range available for assignment to regular users.

Groups

It is usually used either specific permission or restriction to specific person.

1. groupadd <groupname>
    groupadd marketing

2. del a group
    groupdel marketing

3. add a user to a group
#adduser jeetu <trainer>

4. remove a user from a group
#deluser jeetu <group>

5. Add multi users
    #gpasswd -M u1,u2,u2 <group-name>

6. Deleting a user from a particular group:
   #gpasswd -d <username> <groupname>

data stored in
#vim /etc/group

VNC server

Saturday, March 3, 2018 9:29 AM

VNC server configuration
==========WORKING CONFIG==========

a. Install VNC server:
   i. #yum install -y tigervnc* openssh
b. Port numbers used by VNC
   i. 5900, 5901, 5902
c. Edit VNC config file
   i. #vim /etc/sysconfig/vncservers
      1) In the end of the file, uncomment last 2 line and edit these:
         a) VNCSERVERS="2:<username>"
         b) VNCSERVERARGS[2]="-geometry 1024x768"
         c) Save & quit
      2) Login to the user <username>
      3) Type password for VNC, #vncpasswd
      4) Enable, start, status "vncserver" service.
         a) #service vncserver start
         b) #chkconfig vncserver on
         c) #service vncserver status
   ii. Switch to another system
      1) Execute below command
         a) #vncviewer -via <username>@<remote-IP> localhost:2
         b) Provide user password
         c) Provide vncpasswd
      2) That's it.

https://www.dell.com/support/article/in/en/indhs1/sln283098/how-to-install-and-configure-a-vnc-server-on-redhat-enterprise-linux-rhel-6?lang=en
==========

INSTALL DESKTOP PACKAGES
#yum groupinstall "GNOME Desktop"
==========

INSTALL TIGERVNC
#yum install tigervnc-server

update the user's info in the config file
#vim /etc/systemd/system/vncserver@:3.service
<USER> -> username

firewall rule to be enabled
#firewall-cmd --permanent --zone=public --add-port=5903/tcp
#firewall-cmd --reload

set VNC server password
#su - u1
password: pass@word1
verify : pass@word1

start & enable VNC

#systemctl daemon-reload
#systemctl start vncserver@:3.service
#systemctl enable vncserver@:3.service

Access remote desktop throught VNCViewer
#vncviewer <vnc-server-ip> //ubuntu

YUM Installation

Saturday, March 3, 2018 9:29 AM

Yum server installation using cd/iso only

[RHEL 6]

vim /etc/yum.repos.d/yumserver.repo

```
[Yumserver]
name="RHEL 6"
baseurl=file:///media/RHEL_6.2\ x86_64\ Disc\ 1
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

:wq!
```

#yum clean all

==========

[RHEL 7]

vim /etc/yum.repos.d/yumserver.repo

```
[Yumserver]
name="RHEL 7"
baseurl=file:///run/media/root/RHEL-7.0\ Server.x86_64
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

:wq!
```

#yum clean all

Full FTP method - YUM installation

MANAGING PACKAGES USING YUM

---

1. setting up yum server

packages needed for yum server

```
a. createrepo
b. deltarpm
c. python-deltarpm
d. vsftpd
```

check using rpm

```
#rpm -q createrepo
#rpm -q deltarpm
#rpm -q python-deltarpm
#rpm -q vsftpd
```

OR

```
#rpm -q createrepo deltarpm python-deltarpm vsftpd
```

{firstly check all 4 are alredy installed or not, if not install them}

2. create YUM directory

```
#mkdir /var/ftp/pub/yumserver
```

3. copy all the packages from RHEL7 DVD ROM to YUM directory

```
#cp -var * /var/ftp/pub/yumserver/
```

var=verbose,all,recursive

* = all packages

4. creating yum client configuration file

client config file :- contains file from where yum will go & download the packages using ftp
#cd /etc/yum.repos.d
#ls
one file is their, delete that file as it belongs to RHN
#rm pa....
to create a new repo file
#vim yumserver.repo
inside this file write
****
[yumserver]
name='RHEL7 Repo."
baseurl=ftp://192.168.0.1/pub/yumserver
gpgcheck=0
enabled=1
****
gpgcheck=Gnu Privacy guard check, it is asking for some privacy key
5. creating repo
#createrepo -v /var/ftp/pub/yumserver
v=verbose
****
repo is nothing but an index for all the packages, for efficient use
****
6. enable & start ftp service
#systemctl enable vsftpd //enable vsftpd
#systemctl start vsftpd //starting vsftpd
#systemctl status vsftpd //checking for status

7. clean the repo & update the repo
#yum clean all
#yum update all
#yum repolist

YUM Full

Saturday, March 3, 2018 9:30 AM

```
#Give permissions to this file

#chmod 755 yumserver.sh

#Running the file

#./yumserver.sh


#disable SELINUX

setenforce 0


#checking packages

rpm -q createrepo python-deltarpm deltarpm vsftpd


#creating yum directory

mkdir /var/ftp/pub/yumserver


#copy all packages to yumserver
cp -var /run/media/root/RHEL-7.0\ Server.x86_64/Packages/*  /var/ftp/pub/yumserver


#creating repo file

touch /etc/yum.repos.d/yumserver.repo

echo "[yumserver]" >> /etc/yum.repos.d/yumserver.repo

echo "name='RHEL 7 Repo.' " >> /etc/yum.repos.d/yumserver.repo

echo "baseurl=ftp://192.168.0.1/pub/yumserver" >> /etc/yum.repos.d/yumserver.repo

echo "gpgcheck=0" >> /etc/yum.repos.d/yumserver.repo

echo "enabled=1" >> /etc/yum.repos.d/yumserver.repo


#create repo

createrepo -v /var/ftp/pub/yumserver


#enable & start vsftpd
```

```
systemctl enable vsftpd

systemctl start vsftpd

systemctl status vsftpd


#clean & update

yum clean all

yum update all

yum repolist
```

Windows AD domain join

Wednesday, January 9, 2019 4:29 PM

https://www.linuxtechi.com/integrate-rhel7-centos7-windows-active-directory/

realm join --user=<username> <domain.name>

Additional commands

Saturday, January 12, 2019 12:20 PM

To list directory size in human readable format, with max depth (link)

- #du -h --max-depth=1 <Dir_Loc>

To check last login details of all users in a Linux system:

```
[root@jeetu ~]# lastlog -b 0 -t 100
Username         Port     From             Latest
root             :0                        Mon Jan 14 13:20:56 +0530 2019
gdm              :0                        Mon Jan 14 13:20:33 +0530 2019
jeetu            pts/0                     Wed Jan  2 15:08:53 +0530 2019
tomar            pts/1                     Mon Jan 14 12:25:28 +0530 2019
u1               pts/1                     Mon Jan 14 14:14:54 +0530 2019
u2               pts/2                     Mon Jan 14 14:15:35 +0530 2019
u3               pts/3                     Mon Jan 14 14:16:07 +0530 2019
u4               pts/4                     Mon Jan 14 14:17:31 +0530 2019
[root@jeetu ~]#
```

To check last login details using user name:

```
[root@jeetu ~]# lastlog -u root
Username         Port     From             Latest
root             :0                        Mon Jan 14 13:20:56 +0530 2019
[root@jeetu ~]# lastlog -u u1
Username         Port     From             Latest
u1               pts/1                     Mon Jan 14 14:14:54 +0530 2019
[root@jeetu ~]#
```

To check last reboot:

```
File  Edit  View  Search  Terminal  Help
[root@jeetu ~]# last reboot
reboot   system boot  3.10.0-123.el7.x Mon Jan 14 18:50 - 15:15  (-3:-35)
reboot   system boot  3.10.0-123.el7.x Mon Jan 14 17:44 - 12:54  (-4:-49)
reboot   system boot  3.10.0-123.el7.x Mon Jan 14 17:38 - 12:14  (-5:-24)
reboot   system boot  3.10.0-123.el7.x Mon Jan 14 17:09 - 12:07  (-5:-1)
reboot   system boot  3.10.0-123.el7.x Mon Jan 14 16:25 - 11:38  (-4:-47)
reboot   system boot  3.10.0-123.el7.x Tue Nov 27 15:26 - 10:09 (36+18:43)
reboot   system boot  3.10.0-123.el7.x Mon Nov 26 21:46 - 09:55  (12:08)

wtmp begins Mon Nov 26 21:46:35 2018
[root@jeetu ~]#
```

To check the uptime:

#uptime

Zabbix to Securely Monitor Remote Servers

Friday, February 8, 2019 2:36 PM

https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-zabbix-to-securely-monitor-remote-servers-on-centos-7

https://www.howtoforge.com/tutorial/centos-zabbix-system-monitoring/

Patch Management

Wednesday, March 20, 2019 8:01 PM

Like all OSes, every once in a while you need to update the software running on your Linux server. You can do this in one of three ways :

- Download the updated packages and manually install them yourself.
- Use a built-in open source application that comes with the OS distribution.
- Use a third party application that downloads the file and then runs the installation for you.

Tar, Gzip, Bzip2

Friday, May 10, 2019 8:37 AM

tar: archive tool
gzip: compression tool (higher compression speed)
bzip2: compression tool (higher compression rate)

to create tar:	#tar -cvf tar-name.tar file1 f2 f3 f4...
to list tar:	#tar -tf tar-name.tar
to count file in tar	#tar -tf tar-name.tar | wc -l
to extract all files from tar	#tar -xf tar-name.tar
to extract a file from tar	#tar -xf tar-name.tar f1 f2 f3 ..
to extract a file from tar in a different location	#tar -C <remote-location> -xf tar-name.tar

GZIP

create gzip	#tar -cvzf tar-name.tar.gz <files> compressing file or directory OR #gzip file.tar compressing a tar only
list the content	#tar -tf tar-name.tar.gz
To see the compression ratio	#gzip -l tar-name.tar.gz
To unzip	#gunzip tar-name.tar.gz

BZIP2

create bzip2	#tar -cvjf tar-name.tar.bz2 <file> compressing file or directory
list the content	#tar -tf tar-name.tar.bz2 compressing a tar only
To unzip	#bunzip tar-name.tar.bz2
Compression file + keeping original file	#bzip2 -k test.txt

AFTER CREATING TAR, GZIP & BZIP2
#ls -l //for matching the sizes
FAST compression

TAR, GZIP, BZIP2:
==========

create a single file from /boot for backup	#dd if=/dev/sda1 of=boot.img
listing the size	#ls #ll -h #in MBs

creating archival of this file	#tar -cvf archive.tar boot.img
creating a copy of it.	#cp archive.tar archive2.tar
listing the size	#ll -h #in MBs
using gzip to compress the file.	#gzip archive.tar
using bzip2 to compress 2nd file.	#bzip2 archive2.zip
listing the size for comparision	#ll -h #ls -l
to decompress using gzip	#gunzip archive.tar.gz
to decompress using bzip2	#bunzip2 archive.tar.bz2
to extract using tar	#tar-xvf archive.tar
To extract a single file from the tar (from /boot) on current location.	#tar --extract --file=boot.tar boot/efi Efi is a folder inside /boot folder.

Password details

Sunday, May 19, 2019 10:23 PM

Shadow passwords and password policy

In the distant past, encrypted passwords were stored in the world-readable /etc/passwd file. This was thought to be reasonably secure until dictionary attacks on encrypted passwords became common . At that point, the encrypted passwords, or " password hashes," were moved to the more secure /etc/shadow file. This new file also allowed password aging and expiration features to be implemented .

There are three pieces of information stored in a modern password hash:

1gCjLa2/Z$6Pu0EK0AzfCjxjv2hoLOB/

1. **1**: The hashing algorithm. The number 1 indicates an MD5 hash. The number 6 appears when a SHA-512 hash is used.
2. **gCjLa2/Z**: The *salt* used to encrypt the hash. This is originally chosen at random. The salt and the unencrypted password are combined and encrypted to create the encrypted password hash. The use of a salt prevents two users with the same password from having identical entries in the **/etc/shadow** file.
3. **6Pu0EK0AzfCjxjv2hoLOB/**: The encrypted hash.

/etc/shadow format

The format of **/etc/shadow** follows (nine colon-separated fields):

```
❶name:❷password:❸lastchange:❹minage:❺maxage:❻warning:❼inactive:❽expire:❾blank
```

❶ The login *name*. This must be a valid account name on the system.
❷ The encrypted *password*. A password field which starts with a exclamation mark means that the password is locked.
❸ The date of the last *password change*, represented as the number of days since 1970.01.01.
❹ The *minimum* number of days before a password may be changed, where 0 means "no minimum age requirement."
❺ The *maximum* number of days before a password must be changed.
❻ The *warning* period that a password is about to expire. Represented in days, where 0 means "no warning given."
❼ The number of days an account remains active after a password has expired. A user may still log into the system and change the password during this period. After the specified number of days, the account is locked, becoming *inactive*.
❽ The account *expiration* date, represented as the number of days since 1970.01.01.
❾ This *blank* field is reserved for future use.

Default Algo:

```
File  Edit  View  Search  Terminal  Help
[root@server Desktop]# #to check default hashing algo
[root@server Desktop]# authconfig --test | grep hashing
 password hashing algorithm is sha512
[root@server Desktop]#
```

To change algorithm:
#authconfig --passalgo=md5 --update

```
[root@client ~]# authconfig --test | grep hashing
 password hashing algorithm is sha512
[root@client ~]# authconfig --passalgo=md5 --update
[root@client ~]# tail -2 /etc/passwd
u2:x:1009:1011::/home/u2:/bin/bash
u3:x:1010:1012::/home/u3:/bin/bash
[root@client ~]# tail -2 /etc/shadow
u2:!!:18233:0:99999:7:::
u3:!!:18233:0:99999:7:::
[root@client ~]# passwd u3
Changing password for user u3.
New password:
BAD PASSWORD: The password is shorter than 7 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@client ~]# tail -2 /etc/passwd
u2:x:1009:1011::/home/u2:/bin/bash
u3:x:1010:1012::/home/u3:/bin/bash
[root@client ~]# tail -2 /etc/shadow
u2:!!:18233:0:99999:7:::
u3:$1$bYsAcCuz$pvRDDGUxX1hcRIrlFNhCW.:18239:0:99999:7:::
[root@client ~]# authconfig --test | grep hashing
 password hashing algorithm is md5
```

Notes : The new algorithm in passwd/shadow files will apply until next execution of passwd command

Oracle Linux 7

Monday, May 27, 2019 10:36 AM

Download link: http://mirrors.kernel.org/oracle/OL7/u6/x86_64/

Syslog Standards

Friday, June 7, 2019 11:54 AM

- Allows logs control centrally.
- Uses facilities & severities to categorize messages.

Number	Keyword	Description
0	kern	kernel messages
1	user	user-level messages
2	mail	mail system
3	daemon	system daemons
4	auth	security/authorization messages
5	syslog	messages generated by syslogd
6	lpr	line printer subsystem
7	news	network news subsystem
8	uucp	UUCP subsystem
9	clock	daemon
10	authpriv	security/authorization messages

Severity levels:

Code	Severity	Keyword	Description
0	Emergency	emerg (panic)	System is unusable
1	Alert	alert	Action must be taken immediately
2	Critical	crit	Critical conditions
3	Error	err (error)	Error conditions
4	Warning	warning (warn)	Warning conditions
5	Notice	notice	Normal but significant condition
6	Info	info	Informational messages
7	Debug	debug	Debug-level messages

- 0 = Emergency = highest severity
- 7 = Debug = lowest severity

To implement, several methods are:

1. Syslogd
2. Rsyslog
3. Syslog-ng

Logging Rules: (both fields are separated by 1 or more spaces or tabs)

1. Selector field
    a. Facility.severities
    b. Mail.*
    c. Mail
    d. Facility.none
    e. Facility_1.severity; Facility_2.severity

2. Action Field
    a. Determines how message is processed.

Example Logging Rule

```
mail.*        /var/log/mail.log
```

Caching & non-caching:

- Caching is good for I/O performance.
- Data might be lost in case of crash.
- Denoted by a hipher (-) sign.

```
mail.info      -/var/log/mail.info
mail.warn      -/var/log/mail.warn
mail.err       /var/log/mail.err
```

Logrotate:

- To rotate , mail, compress, remove logs.
- #/etc/logrotate.conf
- To configure logs:

```
/var/log/debug
/var/log/messages
{
        rotate 4
        weekly
        missingok
        notifempty
        compress
        sharedscripts
        postrotate
               reload rsyslog >/dev/null 2>&1 || true
        endscript
}
```

- To test log rotate:
  #logrotate -fv /etc/logrotate.conf
  F = force
  V = verbose

Special permissions

Friday, June 7, 2019 12:52 PM

SETUID:

- Set user ID.
- It forces the process to start as the owner of the file.
- Examples:
  - #ls -l /usr/bin/passwd
    -rwsr-xr-x 1 root root /usr/bin/passwd
  - #ping
  - #chsh //change shell

#chmod x700 file-name
x = 1 to 7 (below values)

1 = sticky bit
2 = set GUID (SGID)
3 = sticky + SGID
4 = set UID + execution bit (small 's')
5 = set UID + execution bit (small 's') + sticky bit
6 = set UID + execution bit (small 's') + set GUID (SGID)
7 = set UID + execution bit (small 's') + set GUID (SGID) + sticky bit

Octal Permissions

setuid	setgid	sticky	
0	0	0	Value for off
1	1	1	Binary value for on
4	2	1	Base 10 value for on

Adding the Setuid Attribute

```
chmod u+s /path/to/file

chmod 4755 /path/to/file
```

Removing the Setuid Attribute

```
chmod u-s /path/to/file

chmod 0755 /path/to/file
```

Finding Setuid Files

```
find / -perm /4000

# Older style:
find / -perm +4000
```

SETGID:

- Set Group ID

Setgid

- setgid = Set Group ID upon execution.
- `-rwxr-sr-x 1 root tty /usr/bin/wall`

Wall = sends the write data to terminal of all the logged in users.

Finding Setgid Files

```
find / -perm /2000 -ls

# Older style:
find / -perm +2000 -ls
```

Adding the Setgid Attribute

```
chmod g+s /path/to/file
```

```
chmod 2755 /path/to/file
```

Adding the Setuid & Setgid Attributes

```
chmod ug+s /path/to/file
```

```
chmod 6755 /path/to/file
```

Removing the Setgid Attribute

```
chmod g-s /path/to/file
```

```
chmod 0755 /path/to/file
```

Linux Sysprep

Friday, July 5, 2019 11:15 AM

Link: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/virtualization_deployment_and_administration_guide/sect-guest_virtual_machine_disk_access_with_offline_tools-using_virt_sysprep

To install virt-sysprep, enter the following command:
yum install /usr/bin/virt-sysprep

Syslog

Thursday, July 11, 2019 10:02 PM

Syslog files:

Overview of syslog priorities

Code	Priority	Severity
0	emerg	System is unusable.
1	alert	Action must be taken immediately.
2	crit	Critical condition.
3	err	Non-critical error condition.
4	warning	Warning condition.
5	notice	Normal but significant event.
6	info	Informational event.
7	debug	Debugging-level message.

Sample rules section of `rsyslog.conf`

```
#### RULES ####

# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                 /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none                /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                              /var/log/secure

# Log all the mail messages in one place.
mail.*                                                  -/var/log/maillog


# Log cron stuff
cron.*                                                  /var/log/cron

# Everybody gets emergency messages
*.emerg                                                 :omusrmsg:*

# Save news errors of level crit and higher in a special file.
uucp,news.crit                                          /var/log/spooler

# Save boot messages also to boot.log
local7.*                                                /var/log/boot.log
```

Log file rotation:

- Logs are rotated by "logrotate" command.
- Link: https://www.golinuxhub.com/2014/06/how-to-configure-logrotate-for-new-log.html

Analyze a syslog entry

```
❶Feb 11 20:11:48 ❷localhost ❸sshd[1433]: ❹Failed password for student from
 172.25.0.10 port 59344 ssh2
```

❶ The time stamp when the log entry was recorded.
❷ The host from which the log message was sent.
❸ The program or process that sent the log message.
❹ The actual message sent.

Monitor a log file with tail:

#tail - f /path/to/file OR

#tail - f /var/log/secure

Send a syslog message with logger

To send a message to **rsyslogd** that gets recorded in the **/var/log/boot.log** log file, execute:

```
[root;@serverX ~]$ logger -p local7.notice "Log entry created on serverX"
```

Finding events with journalctl:

```
[root@serverX ~]# journalctl
Feb 13 10:01:01 server1 run-parts(/etc/cron.hourly)[8678]: star
Feb 13 10:01:01 server1 run-parts(/etc/cron.hourly)[8682]: fini
Feb 13 10:10:01 server1 systemd[1]: Starting Session 725 of use
Feb 13 10:10:01 server1 systemd[1]: Started Session 725 of user
Feb 13 10:10:01 server1 CROND[8687]: (root) CMD (/usr/lib64/sa/
```

Size on disk:

```
[root@svr Desktop]# journalctl --disk-usage
Journals take up 8.0M on disk.
[root@svr Desktop]#
```

Last 10 entries:

```
[root@svr Desktop]# journalctl -n 5
-- Logs begin at Fri 2019-07-12 01:33:39 IST, end at Thu 2019-07-11 22:30:01 IST. --
Jul 11 22:30:01 svr.allianz.local systemd[1]: Starting Session 101 of user root.
Jul 11 22:30:01 svr.allianz.local systemd[1]: Started Session 101 of user root.
Jul 11 22:30:01 svr.allianz.local CROND[10741]: (root) CMD (rm -rf /tmp/*)
Jul 11 22:30:01 svr.allianz.local CROND[10742]: (root) CMD (/usr/lib64/sa/sa1 1 1)
Jul 11 22:30:01 svr.allianz.local systemd[1]: Failed to mark scope session-100.scope as ab
lines 1-6/6 (END)
```

Current logs & between:

Output all journal entries that got recorded today:

```
[root@serverX ~]# journalctl --since today
```

Output the journal entries from 10th February 2014 20:30:00 to 13th February 2014 12:00:00:

```
[root@serverX ~]# journalctl --since "2014-02-10 20:30:00" --until "2014-02-13 12:00:00"
```

```
[root@svr Desktop]# journalctl -o verbose
-- Logs begin at Fri 2019-07-12 01:33:39 IST, end at Thu 2019-07-11 22:32:01 IST. --
Fri 2019-07-12 01:33:39.550925 IST [s=a70b85bd188d4b90b6eff916b400c124;i=1;b=7acc62e15f69492dba4a
    PRIORITY=6
    _TRANSPORT=driver
    MESSAGE=Runtime journal is using 8.0M (max 188.9M, leaving 283.3M of free 1.8G, current limit
    MESSAGE_ID=ec387f577b844b8fa948f33cad9a75e6
    _PID=353
    _UID=0
    _GID=0
    _COMM=systemd-journal
    _EXE=/usr/lib/systemd/systemd-journald
    _CMDLINE=/usr/lib/systemd/systemd-journald
    _CAP_EFFECTIVE=4402800cf
    _SYSTEMD_CGROUP=/system.slice/systemd-journald.service
    _SYSTEMD_UNIT=systemd-journald.service
    _SYSTEMD_SLICE=system.slice
    _SELINUX_CONTEXT=kernel
    _BOOT_ID=7acc62e15f69492dba4afc8df8c8df8a
    _MACHINE_ID=323ef1628844422aab03f2c79fb6a141
    _HOSTNAME=svr.allianz.local
```

Logs from a process:

```
[root@svr Desktop]# journalctl _PID=1
-- Logs begin at Fri 2019-07-12 01:33:39 IST, end at Thu 2019-07-11 22:34:01 IST. --
Jul 12 01:33:39 svr.allianz.local systemd[1]: Starting udev Kernel Socket.
Jul 12 01:33:39 svr.allianz.local systemd[1]: Listening on udev Kernel Socket.
Jul 12 01:33:39 svr.allianz.local systemd[1]: Starting udev Control Socket.
Jul 12 01:33:39 svr.allianz.local systemd[1]: Listening on udev Control Socket.
Jul 12 01:33:39 svr.allianz.local systemd[1]: Starting Sockets.
Jul 12 01:33:39 svr.allianz.local systemd[1]: Reached target Sockets.
Jul 12 01:33:39 svr.allianz.local systemd[1]: Starting Create list of required static
Jul 12 01:33:39 svr.allianz.local systemd[1]: Starting Apply Kernel Variables...
Jul 12 01:33:39 svr.allianz.local systemd[1]: Starting Swap.
Jul 12 01:33:39 svr.allianz.local systemd[1]: Reached target Swap.
Jul 12 01:33:39 svr.allianz.local systemd[1]: Starting Local File Systems.
```

Output the journal messages with priority **warning** and above on serverX.

```
[root@serverX ~]# journalctl -p warning
```

Logrotate

Thursday, July 11, 2019 10:57 PM

Cmds to copy & store a backup of logs & run logrotate forcefully:

```
mkdir /old-logs
cp -var * /old-logs/
ls
ls | wc -l
logrotate -f /etc/logrotate.conf
ls | wc -l
ll
```

To create own/custom log rotate:

Login to:	#cd /etc/logrotate.d/
Create a new file	#vim custom

Enter these values in the file:	/root/Desktop/hist.txt{
	daily
	rotate 1
	compress
	create
	}

Explanation:

Take the log of this file:	/root/Desktop/hist.txt{
Frequency of logs	Daily
Delete old logs after	rotate 1
New rotated log, plain or compressed	Compress
After creating log, create new file	Create
	}

After running the command check the

1. /var/log
2. Check the desktop

Scan new HDD without reboot - lvm

Friday, July 12, 2019 9:26 AM

To scan newly attached HDD, without reboot

1. Attach a new HDD
2. Execute the below commands:

To list current disks	ls -l /sys/class/scsi_host/
Searching a specific type	`grep mpt /sys/class/scsi_host/host?/proc_name`
Reset buffer values:	# echo "- - -" > /sys/class/scsi_host/host2/scan
Verify	Fdisk -l
Using GUI	Application --> utilities --> disks

Deleted files restore

Monday, July 15, 2019 6:49 PM

https://www.rootusers.com/restore-deleted-file-linux/

Xfsdump

Tuesday, August 13, 2019 8:36 AM

Backup

Xfsdump -l 0 -f where-to-backup what-to-bup

Put session label
#full

Put label device
#file

Restore

Check backup inventory
#xfsrestore -I // Capital i

TEST backup before restore
#xfsrestore -t -f where-to-backup

Restoring to a new dir /lvm
#xfsrestore -f where-to-backup -L full /lvm

Dump

Thursday, October 10, 2019 10:50 PM

Dump

1. Create a new folder.
2. Create few files in it (1000)
3. Execute below command:
   #dump -0a -f /my-backup.dump /root/Desktop/dump-test/

   -0 = zero
   -f = file to store the backup
   /my-backup.dump = backup file name.
   /root/Desktop/dump-test/ = what to backup

Restore

#restore -i -f /mudump.dump
restore > pwd
restore > cd /root/Desktop/dump-test/
restore > ls t5*
restore > add t5*
restore > ls t5*
restore > add t??
restore > extract
restore > Specify next volume # (none if no more volumes): 1
restore > set owner/mode for '.'? [yn] n
restore > quit
#ls -l

RAID 0 - verified

10 February 2020 15:50

URL: https://www.looklinux.com/how-to-configure-raid-0-on-centos-7/

- RAID – or “Redundant Array of Independent Disks” – is a strategy for data storage used on most server setups. Understanding how RAID works, how it can help you meet the needs of your business or organization, and understanding differences between RAID levels is important before setting up your server.

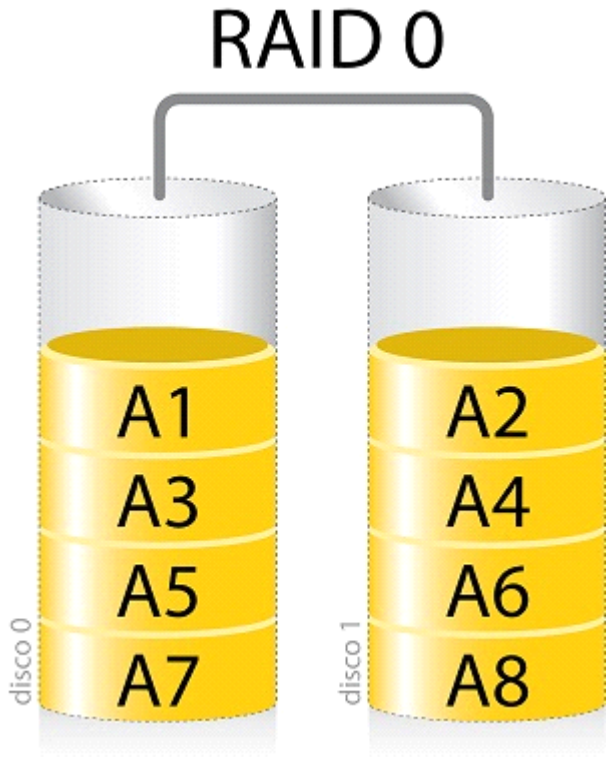
- The disks are connected together to make a logical volume. It offers an excellent performance and this performance will vary depending on the RAID level. The two types of RAID are hardware RAID and software RAID.

RAID Levels

- RAID 0 : Striping
- RAID 1 : Mirroring
- RAID 5 : Striping and Parity
- RAID 6 : Striping with Double Parity
- RAID 10: Combining Mirroring and Striping

RAID Level 0 – Striping

- In RAID 0 data are split up into blocks that get written across all the drive in the array. Using multiple disks at the same time this offers superior I/O performance. RAID 0 consists of striping, without mirroring or parity.
- The capacity of a RAID 0 volume is the sum of the capacities of the disks in the set, the same as with a spanned volume. To configure RAID 0, a minimum of two hard disks are required.



RAID Level 0 – Advantages

- RAID 0 offers great performance, both in read and write operations. There is no overhead caused by parity controls.
- All storage capacity is used, there is no overhead.
- The technology is easy to implement.

RAID Level 0 – Disadvantages

- RAID 0 is not fault-tolerant. If one drive fails, all data in the RAID 0 array are lost. It should not be used for mission-critical systems.

RAID 0 Configuration

- To configure RAID 0 first of all you have to install the mdadm package which is a RAID managing tool in the target system.

```
# yum install mdadm -y
    Loaded plugins: fastestmirror, langpacks
    Loading mirror speeds from cached hostfile
    * base: centos.myfahim.com
    * extras: centos.myfahim.com
    * updates: mirror.ehost.vn
    ....
    ...
    Verifying : mdadm-3.2.6-31.el7.x86_64 2/2
    Updated:
    mdadm.x86_64 0:4.0-5.el7
    Complete
```

After installing mdadm package check if the device is available to configure RAID by typing below command:

```
# lsblk

    NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
    sda 8:0 0 30G 0 disk
    sda1 8:1 0 500M 0 part /boot
    sda2 8:2 0 14.7G 0 part /
    sda3 8:3 0 2.5G 0 part [SWAP]
    sdb 8:16 0 10G 0 disk
    sdc 8:32 0 10G 0 disk
```

You can see the disk is available create a partitioning. Now create the partition for two disk by typing below command:

```
# fdisk /dev/sdb
    Command (m for help): n
    Partition type:
    p primary (0 primary, 0 extended, 4 free)
    e extended
    Select (default p): p
    Partition number (1-4, default 1): 1
    First sector (2048-20971519, default 2048):
    Using default value 2048
    Last sector, +sectors or +size{K,M,G} (2048-20971519, default 20971519): +5G
    Partition 1 of type Linux and of size 5 GiB is set

    Command (m for help): t
    Selected partition 1
    Hex code (type L to list all codes): fd
    Command (m for help): p
```

Check the block details, Execute the following command to check it. The output shown below states that the disks have no previous RAID partition in the target system.

```
# mdadm -E /dev/sd[b-c]1
    mdadm: No md superblock detected on /dev/sdb1.
```

```
mdadm: No md superblock detected on /dev/sdc1.
```

Create md device and select the RAID Level 0 as shown below:

```
# mdadm --create /dev/md0 --level=stripe --raid-devices=2 /dev/sd[b-c]1
    mdadm: Defaulting to version 1.2 metadata
    mdadm: array /dev/md0 started.
```

md device RAID level 0 create successfully. Now check the status of RAID level by typing below command:

```
# mdadm -E /dev/sd[b-c]1
```

You can also verify the status of RAID level by executing the following cat command as shown below.

```
# cat /proc/mdstat
```

To verify the md device status, run the following command.

```
# mdadm --detail /dev/md0
```

Now run the mkfs command to create a filesystem for md device:

```
# mkfs.ext4 /dev/md0
```

Mount permanently md device using blkid command and copy the UUID number. Edit the fstab file using vim editor and enter the copied UUID number in the file like below:

```
# mkdir /mnt/raid0
# blkid /dev/md0
    /dev/md0: UUID= 1de8a17f-bdab-4cf4-9811-fd92af3e9188" TYPE= "ext4"

# vim /etc/fstab
    UUID=1de8a17f-bdab-4cf4-9811-fd92af3e9188 /mnt/raid0 ext4 defaults 0 0
```

The device is mounted successfully and to verify the status of the device, run the following set of commands.

```
# mount -av
# df -h
    Filesystem Size Used Avail Use% Mounted on
    /dev/sda2 15G 4.4G 11G 30% /
    devtmpfs 741M 0 741M 0% /dev
    tmpfs 749M 140K 749M 1% /dev/shm
    tmpfs 749M 8.9M 741M 2% /run
    /dev/sda1 497M 116M 382M 24% /boot
    /dev/md0 5.0G 37M 4.9G 1% /mnt/raid0
```

Cmds:

- yum install -y mdadm
- lsblk
- fdisk /dev/sdb
- fdisk /dev/sdc
- lsblk
- mdadm -E /dev/sd[b-c]1
- mdadm --create /dev/md0 --level=stripe --raid-devices=2 /dev/sd[b-c]1
- mdadm -E /dev/sd[b-c]1
- cat /proc/mdstat
- mdadm --detail /dev/md0
- mkfs.ext4 /dev/md0

- mkdir /raid
- mount /dev/md0 /raid/
- df -h
- mount -av
- history

RPM

13 February 2020 10:52

- RPM (Red Hat Package Manager) is an default open source and most popular package management utility for Red Hat based systems like (RHEL, CentOS and Fedora).
- The tool allows system administrators and users to install, update, uninstall, query, verify and manage system software packages in Unix/Linux operating systems.
- Some Facts about RPM (RedHat Package Manager)
  - RPM is free and released under GPL (General Public License).
  - RPM keeps the information of all the installed packages under /var/lib/rpm database.
  - RPM is the only way to install packages under Linux systems, if you've installed packages using source code, then rpm won't manage it.
  - RPM deals with .rpm files, which contains the actual information about the packages such as: what it is, from where it comes, dependencies info, version info etc.
- There are five basic modes for RPM command
  - Install : It is used to install any RPM package.
  - Remove : It is used to erase, remove or un-install any RPM package.
  - Upgrade : It is used to update the existing RPM package.
  - Verify : It is used to verify an RPM packages.
  - Query : It is used query any RPM package.
- Where to find RPM packages. Below is the list of rpm sites, where you can find and download all RPM packages.
  - http://rpmfind.net
  - http://www.redhat.com
  - http://freshrpms.net/
  - http://rpm.pbone.net/

1. How to Check an RPM Signature Package.
   [root@tecmint]# rpm --checksig pidgin-2.7.9-5.el6.2.i686.rpm
   pidgin-2.7.9-5.el6.2.i686.rpm: rsa sha1 (md5) pgp md5 OK

2. How to Install an RPM Package
   [root@tecmint]# rpm -ivh pidgin-2.7.9-5.el6.2.i686.rpm
   Preparing... ### [100%]
   1:pidgin ### [100%]

3. How to check dependencies of RPM Package before Installing
   [root@tecmint]# rpm -qpR BitTorrent-5.2.2-1-Python2.4.noarch.rpm
   /usr/bin/python2.4
   python >= 2.3
   python(abi) = 2.4
   python-crypto >= 2.0
   RPM command and options
   -q : Query a package
   -p : List capabilities this package provides.
   -R: List capabilities on which this package depends..

4. How to Install a RPM Package Without Dependencies:
   rpm -ivh --nodeps BitTorrent-5.2.2-1-Python2.4.noarch.rpm
   Preparing... ### [100%]
   1:BitTorrent ### [100%]

5. How to check an Installed RPM Package:
   [root@tecmint]# rpm -q BitTorrent
   BitTorrent-5.2.2-1.noarch

6. How to List all files of an installed RPM package:
   [root@tecmint]# rpm -ql BitTorrent
   /usr/bin/bittorrent
   /usr/bin/bittorrent-console
   /usr/bin/bittorrent-curses
   /usr/bin/bittorrent-tracker
   /usr/bin/changetracker-console
   /usr/bin/launchmany-console
   /usr/bin/launchmany-curses

7. How to List Recently Installed RPM Packages:
   [root@tecmint]# rpm -qa --last
   BitTorrent-5.2.2-1.noarch Tue 04 Dec 2012 05:14:06 PM BDT
   pidgin-2.7.9-5.el6.2.i686 Tue 04 Dec 2012 05:13:51 PM BDT
   cyrus-sasl-devel-2.1.23-13.el6_3.1.i686 Tue 04 Dec 2012 04:43:06 PM BDT

8. How to List All Installed RPM Packages
   [root@tecmint]# rpm -qa
   initscripts-9.03.31-2.el6.centos.i686
   polkit-desktop-policy-0.96-2.el6_0.1.noarch

9. How to Upgrade a RPM Package
   [root@tecmint]# rpm -Uvh nx-3.5.0-2.el6.centos.i686.rpm
   Preparing... ### [100%]
   1:nx ### [100%]

10. How to Remove an RPM Package Without Dependencies
    [root@tecmint]# rpm -ev --nodeps vsftpd

11. How to Query an Information (qi) of Installed RPM Package
    [root@tecmint]# rpm -qi vsftpd
    Name : vsftpd Relocations: (not relocatable)
    Version : 2.2.2 Vendor: CentOS

12. Get the Information of RPM Package Before Installing [-qip (query info package)]
    [root@tecmint]# rpm -qip sqlbuddy-1.3.3-1.noarch.rpm
    Name : sqlbuddy Relocations: (not relocatable)
    Version : 1.3.3 Vendor: (none)

13. How to Query documentation of Installed RPM Package[-qdf (query document file)]
    [root@tecmint]# rpm -qdf /usr/bin/vmstat
    /usr/share/doc/procps-3.2.8/BUGS
    /usr/share/doc/procps-3.2.8/COPYING
    /usr/share/doc/procps-3.2.8/COPYING.LIB

14. How to Verify all RPM Packages
    [root@tecmint]# rpm -Va
    S.5....T. c /etc/rc.d/rc.local
    ....T. c /etc/dnsmasq.conf
    ....T. /etc/ld.so.conf.d/kernel-2.6.32-279.5.2.el6.i686.conf

    How To rebuild Corrupted RPM Database
    Sometimes rpm database gets corrupted and stops all the functionality of rpm and other applications on the system. So, at the time we need to rebuild the rpm database and restore it with the help of following command.

15. How To rebuild Corrupted RPM Database
    [root@tecmint]# cd /var/lib
    [root@tecmint]# rm __db*
    [root@tecmint]# rpm --rebuilddb
    [root@tecmint]# rpmdb_verify Packages

rpm command cheat sheet for Linux:

Syntax	Description	Example(s)
rpm -ivh {rpm-file}	Install the package	rpm -ivh mozilla-mail-1.7.5-17.i586.rpm rpm -ivh --test mozilla-mail-1.7.5-17.i586.rpm
rpm -Uvh {rpm-file}	Upgrade package	rpm -Uvh mozilla-mail-1.7.6-12.i586.rpm rpm -Uvh --test mozilla-mail-1.7.6-12.i586.rpm
rpm -ev {package}	Erase/remove/ an installed package	rpm -ev mozilla-mail
rpm -ev --nodeps {package}	Erase/remove/ an installed package without checking for dependencies	rpm -ev --nodeps mozilla-mail
rpm -qa	Display list all installed packages	rpm -qa rpm -qa | less
rpm -qi {package}	Display installed information along with package version and short description	rpm -qi mozilla-mail
rpm -qf {/path/to/file}	Find out what package a file belongs to i.e. find what package owns the file	rpm -qf /etc/passwd rpm -qf /bin/bash
rpm -qc {pacakge-name}	Display list of configuration file(s) for a package	rpm -qc httpd
rpm -qcf {/path/to/file}	Display list of configuration files for a command	rpm -qcf /usr/X11R6/bin/xeyes
rpm -qa --last	Display list of all recently installed RPMs	rpm -qa --last rpm -qa --last | less
rpm -qpR {.rpm-file} rpm -qR {package}	Find out what dependencies a rpm file has	rpm -qpR mediawiki-1.4rc1-4.i586.rpm rpm -qR bash

From <https://www.cyberciti.biz/howto/question/linux/linux-rpm-cheat-sheet.php>

YUM

13 February 2020 11:20

What is YUM?

- YUM (Yellowdog Updater Modified) is an open source command-line as well as graphical based package management tool for RPM (RedHat Package Manager) based Linux systems. It allows users and system administrator to easily install, update, remove or search software packages on a systems.
- YUM uses numerous third party repositories to install packages automatically by resolving their dependencies issues.

1. Install a Package with YUM

```
# yum install firefox
Loaded plugins: fastestmirror
Dependencies Resolved

================================================================================
===============
 Package               Arch       Version                Repository          Size
================================================================================
===============
Updating:
firefox                i686       10.0.6-1.el6.centos    updates             20 M
Updating for dependencies:
 xulrunner             i686       10.0.6-1.el6.centos    updates             12 M

Transaction Summary
================================================================================
===============
Install       0 Package(s)
Upgrade       2 Package(s)

Total download size: 32 M
Is this ok [y/N]: y
Downloading Packages:
(1/2): firefox-10.0.6-1.el6.centos.i686.rpm                          |  20 MB   01:10
(2/2): xulrunner-10.0.6-1.el6.centos.i686.rpm                        |  12 MB   00:52
--------------------------------------------------------------------------------
Total                                          63 kB/s |  32 MB   02:04

Updated:
  firefox.i686 0:10.0.6-1.el6.centos

Dependency Updated:
  xulrunner.i686 0:10.0.6-1.el6.centos

Complete!
```

2. Removing a Package with YUM

```
# yum remove firefox
Loaded plugins: fastestmirror
Setting up Remove Process
Resolving Dependencies
--> Running transaction check
```

```
---> Package firefox.i686 0:10.0.6-1.el6.centos set to be erased
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
========================
 Package                Arch       Version                   Repository          Size
================================================================================
========================
Removing:
 firefox                i686       10.0.6-1.el6.centos       @updates            23 M

Transaction Summary
================================================================================
========================
Remove       1 Package(s)
Reinstall    0 Package(s)
Downgrade    0 Package(s)

Is this ok [y/N]: y
Downloading Packages:
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Erasing       : firefox-10.0.6-1.el6.centos.i686
1/1

Removed:
  firefox.i686 0:10.0.6-1.el6.centos

Complete!
```

3. Updating a Package using YUM
   # yum update mysql

4. List a Package using YUM
   # yum list openssh

5. Search for a Package using YUM
   # yum search vsftpd

6. Get Information of a Package using YUM
   # yum info firefox

7. List all Available Packages using YUM
   # yum list | less

8. List all Installed Packages using YUM
   # yum list installed | less

9. Yum Provides Function
   - Yum provides function is used to find which package a specific file belongs to. For example, if you would like to know the name of the package that has the /etc/httpd/conf/httpd.conf.

```
# yum provides /etc/httpd/conf/httpd.conf
Loaded plugins: fastestmirror
httpd-2.2.3-63.el5.centos.i386 : Apache HTTP Server
Repo        : base
Matched from:
Filename    : /etc/httpd/conf/httpd.conf
```

10. List all available Group Packages
    # yum grouplist

11. Install a Group Packages
    # yum groupinstall 'MySQL Database'

12. Update a Group Packages
    # yum groupupdate 'DNS Name Server'

13. Remove a Group Packages
    # yum groupremove 'DNS Name Server'

14. List Enabled Yum Repositories
    # yum repolist

15. List all Enabled and Disabled Yum Repositories
    # yum repolist all

16. Clean Yum Cache
    # yum clean all

17. View History of Yum
    # yum history

Piping and Redirection

13 February 2020 11:45

URL: https://ryanstutorials.net/linuxtutorial/piping.php

Every program we run on the command line automatically has three data streams connected to it.

- **STDIN** (0) - Standard input (data fed into the program)
- **STDOUT** (1) - Standard output (data printed by the program, defaults to the terminal)
- **STDERR** (2) - Standard error (for error messages, also defaults to the terminal)

Redirecting to a File

The greater than operator (>) indicates to the command line that we wish the programs output (or whatever it sends to STDOUT) to be saved in a file instead of printed to the screen.
Ex:
creating a file: ls > myoutput
append data i file: ls >> myoutput

Redirecting from a File

If we use the less than operator (<) then we can send data the other way. We will read data from the file and feed it into the program via it's STDIN stream.
Ex:
wc -l myoutput
wc -l < myoutput

Redirecting STDERR

[user@bash] ls -l video.mpg blah.foo
ls: cannot access blah.foo: No such file or directory
-rwxr--r-- 1 ryan users 6 May 16 09:14 video.mpg

[user@bash] ls -l video.mpg blah.foo 2> errors.txt
-rwxr--r-- 1 ryan users 6 May 16 09:14 video.mpg

[user@bash] cat errors.txt
ls: cannot access blah.foo: No such file or directory

[user@bash] ls -l video.mpg blah.foo > myoutput 2>&1

Summary:

Save output to a file.	>
Append output to a file.	>>
Read input from a file.	<
Redirect error messages	2>
Send the output from one program as input to another program.	|

X11

14 February 2020 13:00

Video: https://www.youtube.com/watch?v=7y-wycTrl0I

- X is the standard graphical user interface for Linux. Like other graphical user interfaces such as Microsoft Windows and Mac OS, X lets you interact with programs by using a mouse (or other pointing device) to point and click, providing a simple means of communicating with your computer.

Commands:

Installing required packages	#yum install -y xterm* xorg* xclock* xauth*
Install VIM (if needed)	#yum install -y vim*
Configure SSHD_CONFIG file	#vim /etc/ssh/sshd_config
Make required entries in SSHD_CONFIG	On line 18, **AddressFamily inet** On line 117, **X11Forwarding yes** On line 118, **X11DisplayOffset 10** On line 119, **X11useLocalhost yes** Save, quit & restart sshd service

Login to another system & connect to the system (where SSHD_CONFIG is configured) & connect through SSH.
After you login through SSH, execute
#xclock

R7-AD domain join

26 March 2020 12:42 PM

- Ensure that the linux & DC is pinging.

#realm join <domain.name>

- Give DC admin password.

Verify the domain name
#dnsdomainname

POSTFIX mail server -R7

27 March 2020 12:45 PM

Install postfix package	#yum install -y postfix
Verify package	#rpmquery postfix
Configure main.cf file	#vim /etc/postfix/main.cf ---------- line68: [BELOW INTERNET HOST AND DOMAIN NAMES] Line77: myhostname = <linux-system-full-hostname> Line86: mydomain = <domain.name> Line101: myorigin = $mydomain Line115: uncomment [inet_interfaces = all] Save & quit.
Check the configuration	#postfix check
Restart the service	#systemctl restart postfix
Check the status	#systemctl status postfix
Check / install telnet	#yum install telnet; #rpmquery telnet
Check the postfix listening port	#netstat -tulpn | grep 25 ➢ It must show 0.0.0.0:25, means listening all NICs
Login through telnet & initiate email	#telnet <linux-system-full-hostname> 25 >say hello to postfix EHLO <linux-system-full-hostname> >set mail acceptance from a user. MAIL FROM: user1@<linux-system-full-hostname> >set recipient RCPT TO: user2@<linux-system-full-hostname> >set data DATA >fill mail subject

	Subject: mail subject line >hit enter, and write your mail. This is a test mail Regards Jeetu >to close mail add a dot(.) in the end. . >to close the mail. quit ========== [root@linux postfix]# telnet linux.alpha.corp 25 Trying 192.168.10.12... Connected to linux.alpha.corp. Escape character is '^]'. 220 linux.alpha.corp ESMTP Postfix EHLO linux.alpha.corp 250-linux.alpha.corp 250-PIPELINING 250-SIZE 10240000 250-VRFY 250-ETRN 250-ENHANCEDSTATUSCODES 250-8BITMIME 250 DSN MAIL FROM hulk@linux.alpha.corp 501 5.5.4 Syntax: MAIL FROM:<address> MAIL FROM: hulk@linux.alpha.corp 250 2.1.0 Ok RCPT TO: jeetu@linux.alpha.corp 250 2.1.5 Ok DATA 354 End data with <CR><LF>.<CR><LF> Subject: hahah PostFix This is a test email. Regards Alpha.Corp ==========
To verify mail from user1 to user2	#cat /var/spool/mail/user2 EX: Subject: hahah PostFix Message-Id: <20200327071355.8882342FDFEF@linux.alpha.corp> Date: Fri, 27 Mar 2020 12:43:21 +0530 (IST) From: hulk@linux.alpha.corp This is a test email. Regards Alpha.Corp

Add firewall rule	#firewall-cmd --add-service=smtp --permanent #firewall-cmd --reload

Nginx on Centos 7

Tuesday, July 18, 2023 12:30 PM

Step 1: Update Repository Package Lists

sudo yum -y update

```
[root@svr1 ~]# yum update
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirrors.nxtgen.com
 * extras: mirrors.nxtgen.com
 * updates: mirrors.nxtgen.com
Resolving Dependencies
--> Running transaction check
---> Package NetworkManager.x86_64 1:1.18.8-1.el7 will be updated
---> Package NetworkManager.x86_64 1:1.18.8-2.el7_9 will be an update
---> Package NetworkManager-adsl.x86_64 1:1.18.8-1.el7 will be updated
---> Package NetworkManager-adsl.x86_64 1:1.18.8-2.el7_9 will be an update
---> Package NetworkManager-bluetooth.x86_64 1:1.18.8-1.el7 will be updated
---> Package NetworkManager-bluetooth.x86_64 1:1.18.8-2.el7_9 will be an update
```

Step 2: Install Extra Packages for Enterprise Linux (EPEL)

yum install -y epel-release

```
[root@svr1 ~]# yum install -y epel-release
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirrors.nxtgen.com
 * extras: mirrors.nxtgen.com
 * updates: mirrors.nxtgen.com
Resolving Dependencies
--> Running transaction check
---> Package epel-release.noarch 0:7-11 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                        Arch                           Version
================================================================================
Installing:
 epel-release                   noarch                         7-11
```

Step 3: Install Nginx

yum –y install nginx

```
[root@svr1 ~]# yum install nginx -y
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirrors.nxtgen.com
 * epel: epel.excellmedia.net
 * extras: mirrors.nxtgen.com
 * updates: mirrors.nxtgen.com
Resolving Dependencies
--> Running transaction check
---> Package nginx.x86_64 1:1.20.1-10.el7 will be installed
--> Processing Dependency: nginx-filesystem = 1:1.20.1-10.el7 for package: 1:nginx-1.20.1-10.el7.x86_64
--> Processing Dependency: libcrypto.so.1.1(OPENSSL_1_1_0)(64bit) for package: 1:nginx-1.20.1-10.el7.x86_64
--> Processing Dependency: libssl.so.1.1(OPENSSL_1_1_0)(64bit) for package: 1:nginx-1.20.1-10.el7.x86_64
--> Processing Dependency: libssl.so.1.1(OPENSSL_1_1_1)(64bit) for package: 1:nginx-1.20.1-10.el7.x86_64
--> Processing Dependency: nginx-filesystem for package: 1:nginx-1.20.1-10.el7.x86_64
--> Processing Dependency: libcrypto.so.1.1()(64bit) for package: 1:nginx-1.20.1-10.el7.x86_64
--> Processing Dependency: libssl.so.1.1()(64bit) for package: 1:nginx-1.20.1-10.el7.x86_64
--> Running transaction check
---> Package nginx-filesystem.noarch 1:1.20.1-10.el7 will be installed
---> Package openssl11-libs.x86_64 1:1.1.1k-5.el7 will be installed
--> Finished Dependency Resolution
```

Step 4: Start Nginx Service

systemctl enable nginx

systemctl start nginx

systemctl status nginx

```
[root@svr1 ~]# systemctl enable nginx
Created symlink from /etc/systemd/system/multi-user.target.wants/nginx.service to /usr/lib/
[root@svr1 ~]# systemctl start nginx
[root@svr1 ~]# systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2023-07-16 14:58:32 IST; 10ms ago
  Process: 76113 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
  Process: 76110 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
  Process: 76108 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
 Main PID: 76115 (nginx)
    Tasks: 3
   CGroup: /system.slice/nginx.service
           ├─76115 nginx: master process /usr/sbin/nginx
           ├─76116 nginx: worker process
           └─76117 nginx: worker process
```

Step 6: Configure Firewall to Allow Traffic (for this, I have enabled firewall)

firewall-cmd --zone=public --permanent --add-service=http

firewall-cmd --zone=public --permanent --add-service=https

firewall-cmd --reload

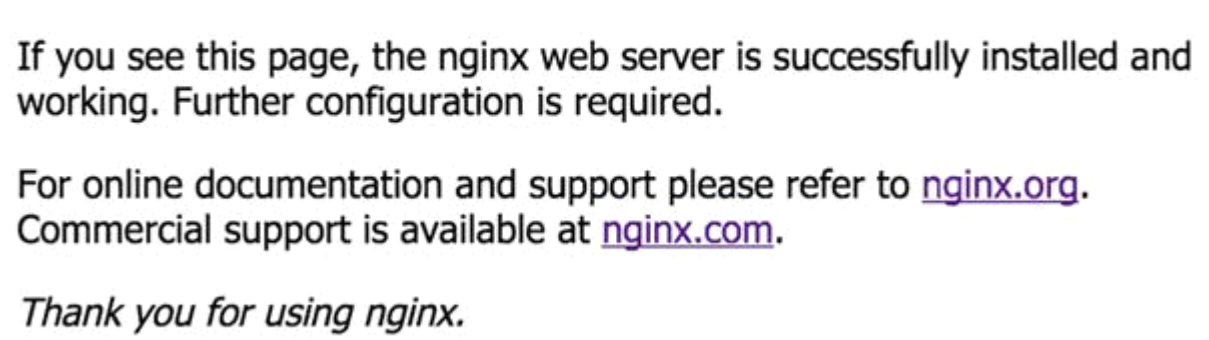
```
[root@svr1 ~]# firewall-cmd --zone=public --permanent --add-service=http
success
[root@svr1 ~]# firewall-cmd --zone=public --permanent --add-service=https
success
[root@svr1 ~]# firewall-cmd --reload
success
[root@svr1 ~]#
```

Step 7: Verify Nginx Install

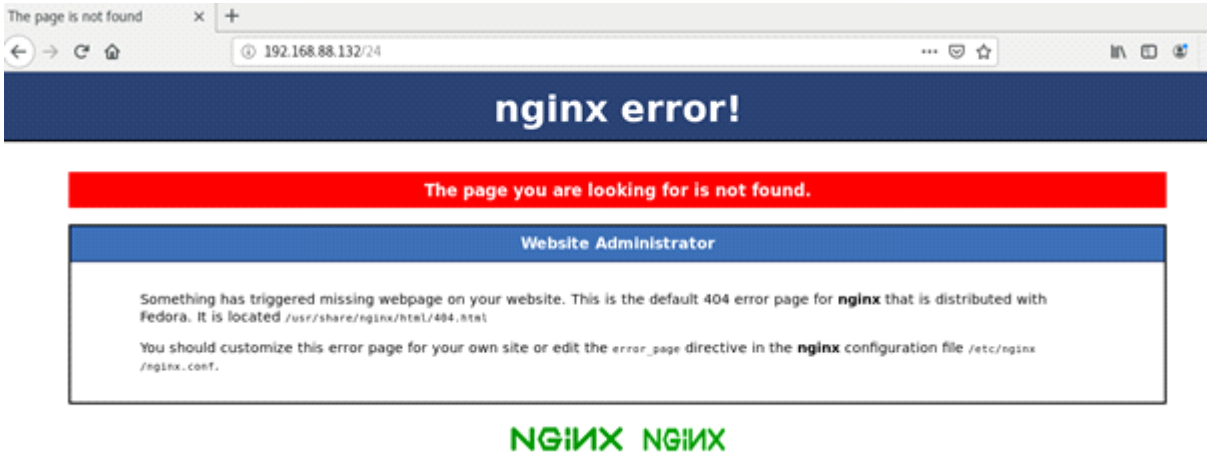
ip a

```
[root@svr1 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:66:43:97 brd ff:ff:ff:ff:ff:ff
    inet 192.168.88.132/24 brd 192.168.88.255 scope global noprefixroute dynamic ens33
       valid_lft 996sec preferred_lft 996sec
    inet6 fe80::b018:2fcb:f1ec:e8f5/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 52:54:00:f1:8b:5e brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
       valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 state DOWN group default qlen 1000
    link/ether 52:54:00:f1:8b:5e brd ff:ff:ff:ff:ff:ff
[root@svr1 ~]#
```

Access your web browser with this IP address:



Or, you might also receive below page.



In order to customize your website:

Switch to “/usr/share/nginx/html”

```
[root@svr2 html]# pwd
/usr/share/nginx/html
[root@svr2 html]# ls
404.html  50x.html  en-US  icons  img  index.html  nginx-logo.png  poweredby.png
[root@svr2 html]#
```

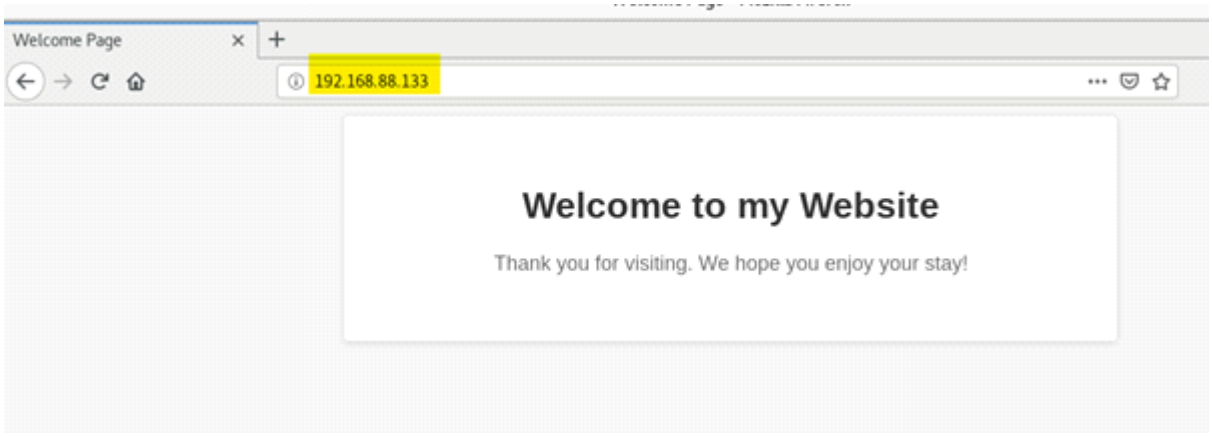
Move/delete the existing “index.html” file to /tmp & create a new index.html file.

```
[root@svr2 html]# cat index.html
<!DOCTYPE html>
<html>
<head>
    <title>Welcome Page</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f2f2f2;
        }

        .container {
            max-width: 600px;
            margin: 0 auto;
            padding: 40px;
            background-color: #ffffff;
            border: 1px solid #dddddd;
            border-radius: 5px;
            box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
        }

        h1 {
            color: #333333;
            text-align: center;
        }

        p {
            color: #666666;
            text-align: center;
            font-size: 18px;
            margin-top: 20px;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Welcome to my Website</h1>
        <p>Thank you for visiting. We hope you enjoy your stay!</p>
    </div>
</body>
</html>

[root@svr2 html]#
```

Restart & check the status of Nginx:

```
[root@svr2 html]# systemctl restart nginx
[root@svr2 html]#
[root@svr2 html]# systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; vendor preset: disabled)
   Active: active (running) since Tue 2023-07-18 12:28:27 IST; 6s ago
  Process: 68001 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
  Process: 67999 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
  Process: 67998 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
 Main PID: 68003 (nginx)
    Tasks: 2
   CGroup: /system.slice/nginx.service
           ├─68003 nginx: master process /usr/sbin/nginx
           └─68008 nginx: worker process

Jul 18 12:28:27 svr2.alpha.corp systemd[1]: Stopped The nginx HTTP and reverse proxy server.
Jul 18 12:28:27 svr2.alpha.corp systemd[1]: Starting The nginx HTTP and reverse proxy server...
Jul 18 12:28:27 svr2.alpha.corp nginx[67999]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Jul 18 12:28:27 svr2.alpha.corp nginx[67999]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Jul 18 12:28:27 svr2.alpha.corp systemd[1]: Started The nginx HTTP and reverse proxy server.
[root@svr2 html]#
```

Verify this on web browser:



From <https://d.docs.live.net/4b07f4230dd85219/Documents/Nginx%20document.docx>

LDAP Config2 - tested

Saturday, July 22, 2023 11:56 AM

Step 1: Install LDAP Packages
yum install -y openldap openldap-clients openldap-servers

Step 2: Configure LDAP Server
cp /usr/share/openldap-servers/DB_CONFIG.example /var/lib/ldap/DB_CONFIG
chown ldap:ldap /var/lib/ldap/*
systemctl start slapd
systemctl enable slapd
slappasswd

Create a file named alpha-corp.ldif using your preferred text editor:
vim alpha-corp.ldif

Add the following content to the file:

Replace "your_password_hash" with the hashed password generated in Step 4
Replace "dc=alpha,dc=corp" with your desired domain components
For example, if you want "alpha.corp," use "dc=alpha,dc=corp"
Adjust the organization (o) and domain (dc) names as needed.

dn: olcDatabase={2}hdb,cn=config
changetype: modify
replace: olcSuffix
olcSuffix: dc=alpha,dc=corp

dn: olcDatabase={2}hdb,cn=config
changetype: modify
replace: olcRootDN
olcRootDN: cn=admin,dc=alpha,dc=corp

dn: olcDatabase={2}hdb,cn=config
changetype: modify
replace: olcRootPW
olcRootPW: {SSHA}your_password_hash

Apply the LDIF file to the LDAP configuration:
ldapmodify -Y EXTERNAL -H ldapi:/// -f alpha-corp.ldif

Create a file named base.ldif using your preferred text editor:
vim base.ldif

Add the following content to the file:
Replace "dc=alpha,dc=corp" with the domain components chosen in Step 5
Adjust the organization (o) and domain (dc) names as needed.

dn: dc=alpha,dc=corp

objectClass: top
objectClass: dcObject
objectClass: organization
o: Alpha Corporation
dc: alpha

dn: cn=admin,dc=alpha,dc=corp
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator

Step 7: Test LDAP Configuration
ldapsearch -x -b dc=alpha,dc=corp

Step 8: Install and Configure LDAP Client (Optional)
yum install -y openldap openldap-clients

NGINX (working)

Sunday, July 23, 2023 11:46 PM

Install and Configure Nginx on CentOS
yum install -y epel-release
yum install nginx -y

enable & start nginx service
systemctl start nginx
systemctl enable nginx
systemctl status nginx

create Nginx config backup
cp /etc/nginx/nginx.conf /etc/nginx/nginx.conf-backup

creating server config file
mkdir /var/www
mkdir /var/www/mysites/
vim alpha.corp
```
server {
   listen 80;
   listen [::]:80;
   root /var/www/mysites/alpha.corp;
   index index.html;
   server_name alpha.corp;
   location / {
      try_files $uri $uri/ =404;
   }
}
```
:wq!

creating additional directory
mkdir /etc/nginx/sites-enabled

creating soft link
ln -s /etc/nginx/mysites/alpha.corp /etc/nginx/sites-enabled/

changing the ownership & group ownership
chown -R nginx:nginx /var/www/mysites/alpha.corp

creating index.html file
vim /var/www/alpha.corp/index.html
```
<html>
<head>
<title>Nginx demo site</title>
</head>
<body>
<h1 align='center'>Welcome to LTI Batch 11</h1>
</body>
</html>
```

:wq!

verifying the configuration:
nginx -t

restarting & checking the status of nginx
systemctl restart nginx
systemctl status nginx

checking IP address
ip a

enter site IP address to host file
vim /etc/hosts
<IP> alpha.corp www.alpha.corp

access web browser
http://<server-ip-address>/ OR
http://alpha.corp