Resume Topics

Tuesday, May 5, 2020 10:14 AM

Linux Administration
Virtualbox/Vmware
AWS/Azure/GC Cloud
Github,Jenkins,Ansible,Docker,Kubernetes
Shellscripting
Mysqldb & Postgres
Apache/Tomcat/PHP
Nagios
ServiceNow/Remedy
Changemgmt & Peoplemgmt
DHCP
DNS
NTP
NFS
FTP
SAMBA
APACHE
MAILSERVER
PXE
RSYNC
PAM
TCPRAPPERS
IPTABLES
SELINUX
USERADMINISTRATION
RAID
LVM
QUOTAS
ACL
CRON
KERNEL COMPILATION
NIS
KDUMP
YUM SERVER CONFIGURATIONS
NETWORK CONFIGURATIONS
OS UPGRADE FROM RHEL 6.2 to 7.0 MINOR
KICKSTART
PROXY SERVER CONFIGURATIONS [SQUID]
TROUBLESHOOTING RHEL 6 AND RHEL 7 ISSUES
GRUB ISSUES
REDHAT SATELLITE CONFIGURATIONS
YUM SERVER

https://www.tecmint.com/things-to-do-after-minimal-rhel-centos-7-installation/

https://www.tecmint.com/install-dhcp-server-in-centos-rhel-fedora/
https://www.tecmint.com/how-to-setup-nfs-server-in-linux/
https://www.tecmint.com/install-ntp-server-in-centos/
https://www.tecmint.com/install-ftp-server-in-centos-7/
https://www.tecmint.com/install-samba4-on-centos-7-for-file-sharing-on-windows/
https://www.tecmint.com/install-apache-on-centos-7/
https://www.tecmint.com/install-pxe-network-boot-server-in-centos-7/
https://www.tecmint.com/sync-two-apache-websites-using-rsync/
https://www.tecmint.com/configure-pam-in-centos-ubuntu-linux/
https://www.tecmint.com/use-pam_tally2-to-lock-and-unlock-ssh-failed-login-attempts/
https://www.tecmint.com/secure-linux-tcp-wrappers-hosts-allow-deny-restrict-access/
https://www.tecmint.com/linux-iptables-firewall-rules-examples-commands/
https://www.tecmint.com/mandatory-access-control-with-selinux-or-apparmor-linux/
https://www.tecmint.com/disable-selinux-in-centos-rhel-fedora/
https://www.tecmint.com/linux-server-hardening-security-tips/
https://www.tecmint.com/manage-users-and-groups-in-linux/
https://www.tecmint.com/understanding-raid-setup-in-linux/
https://www.tecmint.com/manage-software-raid-devices-in-linux-with-mdadm/
https://www.tecmint.com/create-lvm-storage-in-linux/
https://www.tecmint.com/set-access-control-lists-acls-and-disk-quotas-for-users-groups/
https://www.tecmint.com/11-cron-scheduling-task-examples-in-linux/
https://www.tecmint.com/create-and-manage-cron-jobs-on-linux/
https://www.tecmint.com/compile-linux-kernel-on-centos-7/
https://www.tecmint.com/setup-high-availability-clustering-in-centos-ubuntu/

https://www.tecmint.com/introduction-to-glusterfs-file-system-and-installation-on-rhelcentos-and-fedora/
https://www.tecmint.com/perform-self-heal-and-re-balance-operations-in-gluster-file-system/

https://www.tecmint.com/what-is-clustering-and-advantages-disadvantages-of-clustering-in-linux/
https://www.tecmint.com/cman-multi-node-cluster-setup-in-linux/
https://www.tecmint.com/fencing-and-adding-a-failover-to-clustering-server/
https://www.tecmint.com/sync-cluster-configuration-and-verify-failover-setup-in-nodes/

https://www.tecmint.com/install-haproxy-load-balancer-in-linux/

DHCP Server

DHCP (Dynamic Host Configuration Protocol) is a network protocol that enables a server to automatically assign an IP address and provide other related network configuration parameters to a client on a network, from a pre-defined IP pool.
This means that each time a client (connected to the network) boots up, it gets a “dynamic” IP address, as opposed to “static” IP address that never changes. The IP address assigned by a DHCP server to DHCP client is on a “lease”, the lease time can vary depending on how long a client is likely to require the connection or DHCP configuration.

Before we move any further, let’s briefly explain how DHCP works:

When a client computer (configured to use DHCP) and connected to a network is powered on, it forwards a DHCPDISCOVER message to the DHCP server.
And after the DHCP server receives the DHCPDISCOVER request message, it replies with a DHCPOFFER message.
Then the client receives the DHCPOFFER message, and it sends a DHCPREQUEST message to the server indicating, it is prepared to get the network configuration offered in the DHCPOFFER message.
Last but not least, the DHCP server receives the DHCPREQUEST message from the client, and sends the DHCPACK message showing that the client is now permitted to use the IP address assigned to it.

```
yum -y install dhcp
systemctl restart network

vi /etc/sysconfig/dhcpd
DHCPDARGS=eth0

cp /usr/share/doc/dhcp-4.2.5/dhcpd.conf.example /etc/dhcp/dhcpd.conf

vi /etc/dhcp/dhcpd.conf

option domain-name "tecmint.lan";
option domain-name-servers ns1.tecmint.lan, ns2.tecmint.lan;
default-lease-time 3600;
max-lease-time 7200;
authoritative;

subnet 192.168.56.0 netmask 255.255.255.0 {
      option routers                192.168.56.1;
      option subnet-mask            255.255.255.0;
      option domain-search          "tecmint.lan";
      option domain-name-servers    192.168.56.1;
      range   192.168.56.10   192.168.56.100;
      range   192.168.56.120  192.168.56.200;
}

host fedora-node {
           hardware  ethernet 00:4g:8h:13:8h:3a;
           fixed-address 192.168.56.110;
 }

systemctl start dhcpd
systemctl enable dhcpd
firewall-cmd --add-service=dhcp --permanent
firewall-cmd --reload

Other way
# iptables -A INPUT -p tcp -m state --state NEW --dport 67 -j ACCEPT
# service iptables save

cat /var/lib/dhcpd/dhcpd.leases
cat /var/lib/dhcpd/dhcpd.leases~
cat /var/lib/dhcpd/dhcpd.leases
journalctl -xe
```

DHCP Client

```
vi /etc/sysconfig/network-scripts/ifcfg-enp0s3
Note: BOOTPROTO=dhcpd
systemctl restart network
```

==========

NFS Server

NFS (Network File System) is basically developed for sharing of files and folders between Linux/Unix systems by Sun Microsystems in 1980. It allows you to mount your local file systems over a network and remote hosts to interact with them as they are mounted locally on the same system. With the help of NFS, we can set up file sharing between Unix to Linux system and Linux to Unix system.

Benefits of NFS

1. NFS allows local access to remote files.
2. It uses standard client/server architecture for file sharing between all *nix based machines.
3. With NFS it is not necessary that both machines run on the same OS.
4. With the help of NFS we can configure centralized storage solutions.
5. Users get their data irrespective of physical location.
6. No manual refresh needed for new files.
7. Newer version of NFS also supports acl, pseudo root mounts.

8. Can be secured with Firewalls and Kerberos.

NFS Services

Its a System V-launched service. The NFS server package includes three facilities, included in the portmap and nfs-utils packages.

1. portmap : It maps calls made from other machines to the correct RPC service (not required with NFSv4).
2. nfs: It translates remote file sharing requests into requests on the local file system.
3. rpc.mountd: This service is responsible for mounting and unmounting of file systems.

Important Files for NFS Configuration

1. /etc/exports : Its a main configuration file of NFS, all exported files and directories are defined in this file at the NFS Server end.
2. /etc/fstab : To mount a NFS directory on your system across the reboots, we need to make an entry in /etc/fstab.
3. /etc/sysconfig/nfs : Configuration file of NFS to control on which port rpc and other services are listening.

```
yum install nfs-utils
yum install portmap
systemctl start nfs
ps -ef |grep nfs
systemctl start rpcbind
ps -ef |grep rpc
mkdir /nfsshare
vi /etc/exports
cat /etc/exports
/nfsshare client.example.com(rw,sync,async,root_squash,no_root_squash)
exportfs  -s
exportfs -arv
ls -l /nfsshare/
firewall-cmd --permanent --add-service=nfs
firewall-cmd --permanent --add-service=rpc-bind
firewall-cmd --permanent --add-service=mountd
firewall-cmd --reload
systemctl restart nfs
systemctl restart rpcbind
systemctl restart firewalld
systemctl enable nfs rpcbind firewalld
vi /etc/exports
cd /nfsshare/
touch nfssuccess
```

NFS Client

```
yum install nfs-utils
yum install portmap
systemctl start nfs
systemctl  start rpcbind
showmount -e 192.168.31.91
mount -t nfs 192.168.31.91:/nfsshare /mnt/nfsshare
mount | grep nfs
cd /mnt/nfsshare/
mount

vi /etc/fstab
server.example.com:/nfsshare      /mnt/nfsshare nfs    defaults       0 0

mount -a
```

==========

NTP Server

Network Time Protocol – NTP- is a protocol which runs over port 123 UDP at Transport Layer and allows computers to synchronize time over networks for an accurate time. While time is passing by, computers internal clocks tend to drift which can lead to inconsistent time issues, especially on servers and clients logs files or if you want to replicate servers resources or databases.

```
yum install ntp

Vi /etc/ntp.conf

restrict 192.168.31.0 mask 255.255.255.0 nomodify notrap

firewall-cmd --add-service=ntp --permanent
firewall-cmd --reload
systemctl start ntpd
systemctl enable ntpd
systemctl status ntpd

ntpq -p
ntpdate -q  0.ro.pool.ntp.org  1.ro.pool.ntp.org

firewall-cmd --permanent --add-service=ntp
firewall-cmd --reload
```

NTP Client

```
Vi /etc/ntp.conf
server 192.168.31.91
ntpq -p

timedatectl
```

```
timedatectl list-timezones
timedatectl list-timezones | grep Kol
timedatectl set-timezone Asia/Kolkata
date -R
ntpq -p
timedatectl set-ntp 1
timedatectl set-ntp on
ntpdate -q  server.example.com
ntpstat
timedatectl set-timezone UTC
timedatectl set-time 15:58:30
timedatectl set-time 20151120
timedatectl set-time '2015-11-20 16:14:50'
timedatectl | grep local
timedatectl set-local-rtc 1
timedatectl set-local-rtc 0
timedatectl set-ntp true
timedatectl set-ntp false
timedatectl  status
timedatectl set-local-rtc 1
timedatectl set-local-rtc 0
```

==========

FTP Server

FTP (File Transfer Protocol) is a traditional and widely used standard tool for transferring files between a server and clients over a network, especially where no authentication is necessary (permits anonymous users to connect to a server). We must understand that FTP is unsecure by default, because it transmits user credentials and data without encryption.

```
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

Vi /etc/vsftpd/vsftpd.conf
anonymous_enable=NO
local_enable=YES
write_enable=YES
local_umask=022
chroot_local_user=NO
chroot_list_enable=YES
allow_writeable_chroot=YES
dirmessage_enable=YES
xferlog_enable=YES
connect_from_port_20=YES
xferlog_std_format=YES
listen=YES
pam_service_name=vsftpd
userlist_enable=YES
tcp_wrappers=YES
#local_root=/opt/test
#ssl_enable=YES
#ssl_tlsv1_2=YES
#ssl_sslv2=NO
#ssl_sslv3=NO
#rsa_cert_file=/etc/ssl/private/vsftpd.pem
#rsa_private_key_file=/etc/ssl/private/vsftpd.pem
#allow_anon_ssl=NO
#force_local_data_ssl=YES
#force_local_logins_ssl=YES
#require_ssl_reuse=NO
#ssl_ciphers=HIGH
#pasv_min_port=40000
#pasv_max_port=50000
#debug_ssl=YES

##############################PRESENT WORKING CONFIG#################################
/etc/vsftpd/vsftpd.conf
anonymous_enable=YES
no_anon_password=YES
local_enable=NO
write_enable=YES
local_umask=022
chroot_local_user=NO
chroot_list_enable=YES
allow_writeable_chroot=YES
dirmessage_enable=YES
xferlog_enable=YES
connect_from_port_20=YES
xferlog_std_format=YES
listen=YES
pam_service_name=vsftpd
userlist_enable=YES
tcp_wrappers=YES
```

anon_root=/var/ftp/pub1

Securing FTP Server with SELinux
setsebool -P ftp_home_dir on
semanage boolean -m ftpd_full_access --on

systemctl restart vsftpd

echo "centos" | tee -a /etc/vsftpd.userlist
cat /etc/vsftpd.userlist

SSL for vsftpd
openssl req -x509 -nodes -keyout /etc/ssl/private/vsftpd.pem -out /etc/ssl/private/vsftpd.pem -days 365 -newkey rsa:2048
firewall-cmd --zone=public --permanent --add-port=990/tcp
firewall-cmd --zone=public --permanent --add-port=40000-50000/tcp
firewall-cmd --reload

#ssl_enable=YES
#ssl_tlsv1_2=YES
#ssl_sslv2=NO
#ssl_sslv3=NO
#rsa_cert_file=/etc/ssl/private/vsftpd.pem
#rsa_private_key_file=/etc/ssl/private/vsftpd.pem
#allow_anon_ssl=NO
#force_local_data_ssl=YES
#force_local_logins_ssl=YES
#require_ssl_reuse=NO
#ssl_ciphers=HIGH
#pasv_min_port=40000
#pasv_max_port=50000
#debug_ssl=YES

Client Side
Yum install ftp -y
Ftp server.example.com

==========

Samba Server

Samba4 on CentOS 7 (also works on RHEL 7) for basic file sharing between other Linux systems and Windows machines.

yum install samba samba-client samba-common

firewall-cmd --permanent --zone=public --add-service=samba
firewall-cmd --reload

vi /etc/samba/smb.conf

Samba4 Anonymous File Sharing

mkdir -p /srv/samba/anonymous
chmod -R 0775 /srv/samba/anonymous
chown -R nobody:nobody /srv/samba/anonymous

Also, you need to change the SELinux security context for the samba shared directory as follows

chcon -t samba_share_t /srv/samba/anonymous

vi /etc/samba/smb.conf

[Anonymous]
    comment = Anonymous File Server Share
    path = /srv/samba/anonymous
    browsable =yes
    writable = yes
    guest ok = yes
    read only = no
    force user = nobody

testparm

systemctl enable smb.service
systemctl enable nmb.service
systemctl start smb.service
systemctl start nmb.service

Setup Samba4 Secure File Sharing
groupadd smbgrp
usermod tecmint -aG smbgrp
useradd tecmint
usermod tecmint -aG smbgrp
passwd tecmint
smbpasswd -a tecmint

```
mkdir -p /srv/samba/secure
chmod -R 0770 /srv/samba/secure
chown -R root:smbgrp /srv/samba/secure
chcon -t samba_share_t /srv/samba/secure

vi /etc/samba/smb.conf

[Secure]
        comment = Secure File Server Share
        path =  /srv/samba/secure
        valid users = @smbgrp
        guest ok = no
        writable = yes
        browsable = yes

Testparm
systemctl restart smb.service
systemctl restart nmb.service
```

Client side

```
yum install cifs-utils
```

Normal Mount

```
mount -t cifs -o user=tecmint //192.168.31.100/secure /mnt/samba
```

/etc/fstab

```
#//server.example.com/secure       /mnt/samba       cifs   credentials=/mnt/samba/.smbcredentials,defaults 0 0
//server.example.com/secure       /mnt/samba       cifs   user=tecmint,pass=tecmint,defaults 0 0
```

==========

Apache Server

Apache is a free, open source and popular HTTP Server that runs on Unix-like operating systems including Linux and also Windows OS. Since its release 20 years ago, it has been the most popular web server powering several sites on the Internet. It is easy to install and configure to host single or multiple websites on a same Linux or Windows server.

```
yum install httpd

 systemctl start httpd
 systemctl enable httpd
 systemctl status httpd

 firewall-cmd --zone=public --permanent --add-service=http
 firewall-cmd --zone=public --permanent --add-service=https
 firewall-cmd --reload
```

http://192.168.31.91/

Configure Name-based Virtual Hosts on CentOS 7

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

NameVirtualHost *:80
<VirtualHost *:80>
   ServerAdmin webmaster@mytecmint.com
   ServerName mytecmint.com
   ServerAlias www.mytecmint.com
   DocumentRoot /var/www/html/mytecmint.com/
   ErrorLog /var/log/httpd/mytecmint.com/error.log
   CustomLog /var/log/httpd/mytecmint.com/access.log combined
</VirtualHost>


 mkdir -p /var/www/html/mytecmint.com    [Document Root - Add Files]
 mkdir -p /var/log/httpd/mytecmint.com   [Log Directory]

 echo "Welcome to My TecMint Website" > /var/www/html/mytecmint.com/index.html

 systemctl restart httpd.service
```

==========

Apache Additional Info

Document root Directory: /var/www/html or /var/www
Main Configuration file: /etc/httpd/conf/httpd.conf (RHEL/CentOS/Fedora) and /etc/apache2/apache2.conf (Debian/Ubuntu).
Default HTTP Port: 80 TCP
Default HTTPS Port: 443 TCP
Test your Configuration file settings and syntax: httpd -t
Access Log files of Web Server: /var/log/httpd/access_log
Error Log files of Web Server: /var/log/httpd/error_log

How to hide Apache Version and OS Identity from Errors

ServerSignature Off

```
ServerTokens Prod
```

Disable Directory Listing

```
<Directory /var/www/html>
   Options -Indexes
</Directory>
```

Disable Unnecessary Modules

```
grep LoadModule /etc/httpd/conf/httpd.conf
```

Run Apache as separate User and Group

```
# groupadd http-web
# useradd -d /var/www/ -g http-web -s /bin/nologin http-web

User http-web
Group http-web
```

Use Allow and Deny to Restrict access to Directories

```
<Directory />
  Options None
  Order deny,allow
  Deny from all
</Directory>
```

Use mod_security and mod_evasive Modules to Secure Apache

```
# yum install mod_security
# /etc/init.d/httpd restart
```

Disable Apache's following of Symbolic Links

```
Options -FollowSymLinks

# Enable symbolic links
Options +FollowSymLinks
```

Turn off Server Side Includes and CGI Execution

```
Options -Includes
Options -ExecCGI

<Directory "/var/www/html/web1">
Options -Includes -ExecCGI
</Directory>
```

Limit Request Size

```
<Directory "/var/www/myweb1/user_uploads">
  LimitRequestBody 512000
</Directory>
```

Enable Apache Logging

```
<VirtualHost *:80>
DocumentRoot /var/www/html/example.com/
ServerName www.example.com
DirectoryIndex index.htm index.html index.php
ServerAlias example.com
ErrorDocument 404 /story.php
ErrorLog /var/log/httpd/example.com_error_log
CustomLog /var/log/httpd/example.com_access_log combined
</VirtualHost>
```

Securing Apache with SSL Certificates

```
# openssl genrsa -des3 -out example.com.key 1024
```

```
# openssl req -new -key example.com.key -out exmaple.csr
# openssl x509 -req -days 365 -in example.com.csr -signkey example.com.key -out example.com.crt

<VirtualHost 172.16.25.125:443>
        SSLEngine on
        SSLCertificateFile /etc/pki/tls/certs/example.com.crt
        SSLCertificateKeyFile /etc/pki/tls/certs/example.com.key
        SSLCertificateChainFile /etc/pki/tls/certs/sf_bundle.crt
        ServerAdmin ravi.saive@example.com
        ServerName example.com
        DocumentRoot /var/www/html/example/
        ErrorLog /var/log/httpd/example.com-error_log
        CustomLog /var/log/httpd/example.com-access_log common
</VirtualHost>
```

How to Password Protect Web Directories in Apache Using .htaccess File

```
vi /etc/httpd/conf/httpd.conf

<Directory /var/www/html>
Options Indexes Includes FollowSymLinks MultiViews
AllowOverride All
Require all granted
</Directory>


mkdir /home/tecmint

htpasswd -c /home/tecmint/webpass tecmint

chown apache: /home/tecmint/webpass
chmod 640 /home/tecmint/webpass

vi /var/www/html/.htaccess

AuthType Basic
AuthName "Restricted Access"
AuthUserFile /home/tecmint/webpass
Require user tecmint
```

==========

PXE Server

PXE Server – Preboot eXecution Environment – instructs a client computer to boot, run or install an operating system directly form a network interface, eliminating the need to burn a CD/DVD or use a physical medium, or, can ease the job of installing Linux distributions on your network infrastructure on multiple machines the same time.

```
yum install dnsmasq

Vi /etc/dnsmasq.conf

yum install syslinux

yum install tftp-server

mkdir /var/lib/tftpboot/pxelinux.cfg

Vi /var/lib/tftpboot/pxelinux.cfg/default

# yum install vsftpd
# cp -r /mnt/*  /var/ftp/pub/
# chmod -R 755 /var/ftp/pub

# firewall-cmd --add-service=ftp --permanent     ## Port 21
# firewall-cmd --add-service=dns --permanent    ## Port 53
# firewall-cmd --add-service=dhcp --permanent           ## Port 67
# firewall-cmd --add-port=69/udp --permanent   ## Port for TFTP
# firewall-cmd --add-port=4011/udp --permanent  ## Port for ProxyDHCP
# firewall-cmd --reload  ## Apply rules
```

==========

How to Sync Two Apache Web Servers/Websites Using Rsync

The purpose of creating a mirror of your Web Server with Rsync is if your main web server fails, your backup server can take over to reduce downtime of your website. This way of creating a web server backup is very good and effective for small and medium size web businesses.

Advantages of Syncing Web Servers

The main advantages of creating a web server backup with rsync are as follows:

1. Rsync syncs only those bytes and blocks of data that have changed.
2. Rsync has the ability to check and delete those files and directories at backup server that have been deleted from the main web server.
3. It takes care of permissions, ownerships and special attributes while copying data remotely.
4. It also supports SSH protocol to transfer data in an encrypted manner so that you will be assured that all data is safe.
5. Rsync uses compression and decompression method while transferring data which consumes less bandwidth.

```
yum install rsync

useradd tecmint

passwd tecmint

rsync -avzhe ssh tecmint@webserver.example.com:/var/www/ /var/www

ssh-keygen -t rsa -b 2048

ssh-copy-id -i /root/.ssh/id_rsa.pub root@webserver.example.com

crontab –e

*/5     *     *     *     *  rsync -avzhe ssh root@webserver.example.com:/var/www/ /var/www/
```

==========

PAM

Linux-PAM (short for Pluggable Authentication Modules which evolved from the Unix-PAM architecture) is a powerful suite of shared libraries used to dynamically authenticate a user to applications (or services) in a Linux system.
It integrates multiple low-level authentication modules into a high-level API that provides dynamic authentication support for applications. This allows developers to write applications that require authentication, independently of the underlying authentication system.
Many modern Linux distributions support Linux-PAM (hereinafter referred to as "PAM") by default. In this article, we will explain how to configure advanced PAM in Ubuntu and CentOS systems.
Before we proceed any further, note that:

- As a system administrator, the most important thing is to master how PAM configuration file(s) define the connection between applications (services) and the pluggable authentication modules (PAMs) that perform the actual authentication tasks. You don't necessarily need to understand the internal working of PAM.
- PAM has the potential to seriously alter the security of your Linux system. Erroneous configuration can disable access to your system partially, or completely. For instance an accidental deletion of a configuration file(s) under /etc/pam.d/* and/or /etc/pam.conf can lock you out of your own system!

sudo ldd /usr/sbin/sshd | grep libpam.so

```
vim /etc/pam.d/sshd    &
/etc/pam.d/login


auth    required      pam_listfile.so \
      onerr=succeed  item=user  sense=deny  file=/etc/ssh/deniedusers


chmod 600 /etc/ssh/deniedusers
```

PAM_TALLY2

pam_tally2 module is used to lock user accounts after certain number of failed ssh login attempts made to the system. This module keeps the count of attempted accesses and too many failed attempts.
pam_tally2 module comes in two parts, one is pam_tally2.so and another is pam_tally2. It is based on PAM module and can be used to examine and manipulate the counter file. It can display user login attempts counts, set counts on individual basis, unlock all user counts.

```
/etc/pam.d/password-auth

auth       required      pam_tally2.so  file=/var/log/tallylog deny=3 even_deny_root unlock_time=1200

account    required      pam_tally2.so

pam_tally2 --user=tecmint

pam_tally2 --user=tecmint --reset
```

pam_tally2 --user=tecmint

==========

TCP Wrappers

When a network request reaches your server, TCP wrappers uses `hosts.allow` and `hosts.deny` (in that order) to determine if the client should be allowed to use a given service. By default, these files are empty, all commented out, or do not exist. Thus, everything is allowed through the TCP wrappers layer and your system is left to rely on the firewall for full protection. Since this is not desired, due to the reason we stated in the introduction,

Secure Network Services Using TCP Wrappers in Linux

```
ls -l /etc/hosts.allow /etc/hosts.deny

ldd /path/to/binary | grep libwrap

/etc/hosts.deny
sshd,vsftpd : ALL
ALL : ALL

/etc/hosts.allow
sshd,vsftpd : 192.168.0.102,LOCAL
```

ALL matches everything. Applies both to clients and services.
LOCAL matches hosts without a period in their FQDN, such as localhost.
KNOWN indicate a situation where the hostname, host address, or user are known.
UNKNOWN is the opposite of KNOWN.
PARANOID causes a connection to be dropped if reverse DNS lookups (first on IP address to determine host name, then on host name to obtain the IP addresses) return a different address in each case.

==========

IPTABLES

Iptables uses a set of tables which have chains that contain set of built-in or user defined rules. Thanks to them a system administrator can properly filter the network traffic of his system.
Per iptables manual, there are currently 3 types of tables:

```
FILTER
 INPUT
 FORWARD
 OUTPUT

NAT
 PREROUTING
 OUTPUT
 POSTROUTING

MANGLE
 PREROUTING
 OUTPUT
 INPUT
 POSTROUTING
 FORWARD

iptables -A INPUT -s xxx.xxx.xxx.xxx -j DROP

iptables -A INPUT -p tcp -s xxx.xxx.xxx.xxx -j DROP

iptables -D INPUT -s xxx.xxx.xxx.xxx -j DROP

 iptables -A OUTPUT -p tcp --dport xxx -j DROP

iptables -A INPUT -p tcp --dport xxx -j ACCEPT

# iptables -A INPUT  -p tcp -m multiport --dports 22,80,443 -j ACCEPT
# iptables -A OUTPUT -p tcp -m multiport --sports 22,80,443 -j ACCEPT

# iptables -A OUTPUT -p tcp -d 192.168.100.0/24 --dport 22 -j ACCEPT

# iptables -A OUTPUT -p tcp -d 66.220.144.0/20 -j DROP

# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 25 -j REDIRECT --to-port 2525
# iptables -A INPUT -p tcp --dport 80 -m limit --limit 100/minute --limit-burst 200 -j ACCEPT
```

```
# iptables -A INPUT -p icmp -i eth0 -j DROP
# iptables -A INPUT -i lo -j ACCEPT
# iptables -A OUTPUT -o lo -j ACCEPT

# iptables -A INPUT -i eth0 -j LOG --log-prefix "IPtables dropped packets:"
# grep "IPtables dropped packets:" /var/log/messages
# iptables -A INPUT -m mac --mac-source 00:00:00:00:00:00 -j DROP
# iptables -A INPUT -p tcp --syn --dport 22 -m connlimit --connlimit-above 3 -j REJECT
# iptables-save > ~/iptables.rules

# iptables-restore < ~/iptables.rules

# iptables -A INPUT -i eth0 -s xxx.xxx.xxx.xxx -j DROP
# iptables -A OUTPUT -p tcp --dports 25,465,587 -j REJECT

# firewall-cmd --state
# firewall-cmd --list-all
# firewall-cmd --list-interfaces
# firewall-cmd --get-service
# firewall-cmd --query-service service_name
```

==========

Selinux

SELinux is described as a mandatory access control (MAC) security structure executed in the kernel. SELinux offers a means of enforcing some security policies which would otherwise not be effectively implemented by a System Administrator.

setenforce 0

setenforce Permissive

vi /etc/sysconfig/selinux

SELINUX=disabled

Sestatus

Enforcing: SELinux denies access based on SELinux policy rules, a set of guidelines that control the security engine.
Permissive: SELinux does not deny access, but denials are logged for actions that would have been denied if running in enforcing mode.
Disabled (self-explanatory).

```
# getenforce
# setenforce 0
# getenforce
# setenforce 1
# getenforce
# cat /etc/selinux/config
```

==========

Managing Users & Groups, File Permissions & Attributes and Enabling sudo Access on Accounts

Adding User Accounts

```
# adduser [new_account]
# useradd [new_account]
```

.bash_logout
.bash_profile
.bashrc

Understanding /etc/passwd

[username]:[x]:[UID]:[GID]:[Comment]:[Home directory]:[Default shell]

Username: User login name used to login into system. It should be between 1 to 32 charcters long.
Password: User password (or x character) stored in /etc/shadow file in encrypted format.
User ID (UID): Every user must have a User ID (UID) User Identification Number. By default UID 0 is reserved for root user and UID's ranging from 1-99 are reserved for other predefined accounts. Further UID's ranging from 100-999 are reserved for system accounts and groups.
Group ID (GID): The primary Group ID (GID) Group Identification Number stored in /etc/group file.
User Info: This field is optional and allow you to define extra information about the user. For example, user full name. This field is filled by 'finger' command.
Home Directory: The absolute location of user's home directory.

Shell: The absolute location of a user's shell i.e. /bin/bash.

Understanding /etc/group

[Group name]:[Group password]:[GID]:[Group members]

```
# usermod [options] [username]
# usermod --expiredate 2014-10-30 tecmint
# usermod --append --groups root,users tecmint
# usermod --home /tmp tecmint
# usermod --shell /bin/sh tecmint
# groups tecmint
# id tecmint

# usermod --expiredate 2014-10-30 --append --groups root,users --home /tmp --shell /bin/sh tecmint
# usermod --lock tecmint
# usermod --unlock tecmint
# groupadd common_group # Add a new group
# chown :common_group common.txt # Change the group owner of common.txt to common_group
# usermod -aG common_group user1 # Add user1 to common_group
# usermod -aG common_group user2 # Add user2 to common_group
# usermod -aG common_group user3 # Add user3 to common_group

# groupdel [group_name]
# userdel --remove [username]
# chmod 660 common.txt
OR
# chmod u=rw,g=rw,o= common.txt [notice the space between the last equal sign and the file name]

# chmod g+s [filename]
# chmod 2755 [directory]

# chmod o+t [directory]
# chmod 1755 [directory]
# chattr +i file1
# chattr +a file2

# visudo
Defaults    secure_path="/usr/sbin:/usr/bin:/sbin"
root        ALL=(ALL) ALL
tecmint     ALL=/bin/yum update
gacanepa    ALL=NOPASSWD:/bin/updatedb
%admin      ALL=(ALL) ALL
```

==========

How to Set Access Control Lists (ACL's) and Disk Quotas for Users and Groups

Access Control Lists (also known as ACLs) are a feature of the Linux kernel that allows to define more fine-grained access rights for files and directories than those specified by regular ugo/rwx permissions.

Checking File System Compatibility with ACLs

```
# tune2fs -l /dev/sda1 | grep "Default mount options:"
# tune2fs -l /dev/xvda2 | grep "Default mount options:"
```

Introducing ACLs in Linux

```
# groupadd developers
# useradd walterwhite
# useradd saulgoodman
# usermod -a -G developers walterwhite
# usermod -a -G developers saulgoodman

 mkdir /mnt/test
# touch /mnt/test/acl.txtu
# chgrp -R developers /mnt/test
# chmod -R 770 /mnt/test

# su - walterwhite
```

```
# echo "My name is Walter White" > /mnt/test/acl.txt
# exit
# su - saulgoodman
# echo "My name is Saul Goodman" >> /mnt/test/acl.txt
# exit
```

Setting ACL's in Linux

```
# getfacl /mnt/test/acl.txt
# setfacl -m u:gacanepa:rw /mnt/test/acl.txt
# getfacl /mnt/test/acl.txt
# chmod +x /mnt/test
# echo "My name is Gabriel Cánepa" >> /mnt/test/acl.txt
# setfacl -m d:o:r /mnt/test
# getfacl /mnt/test/

# setfacl -x d:o /mnt/test
# setfacl -b /mnt/test
```

Set Linux Disk Quotas on Users and Filesystems

```
UUID=f6d1eba2-9aed-40ea-99ac-75f4be05c05a /home/projects ext4 defaults,grpquota 0 0
UUID=e1929239-5087-44b1-9396-53e09db6eb9e /home/backups ext4 defaults,usrquota 0 0

# umount /home/projects
# umount /home/backups
# mount -o remount /home/projects
# mount -o remount /home/backups

# mount | grep vg00
# quotacheck -avugc
# quotaon -vu /home/backups
# quotaon -vg /home/projects
```

Setting Linux Disk Quotas

```
# setfacl -m u:gacanepa:rwx /home/backups/
# edquota -u gacanepa
# dd if=/dev/zero of=/home/backups/test1 bs=2M count=1
# ls -lh /home/backups/test1

# setfacl -m g:developers:rwx /home/projects/
# edquota -g developers
# edquota -t
```

==========

Cron Scheduling

automate process like backup, schedule updates and synchronization of files and many more. Cron is a daemon to run schedule tasks. Cron wakes up every minute and checks schedule tasks in crontable. Crontab (CRON TABle) is a table where we can schedule such kind of repeated tasks.

Crontab file consists of command per line and have six fields actually and separated either of space or tab. The beginning five fields represent time to run tasks and last field is for command.

1. Minute (hold values between 0-59)
2. Hour (hold values between 0-23)
3. Day of Month (hold values between 1-31)
4. Month of the year (hold values between 1-12 or Jan-Dec, you can use first three letters of each month's name i.e Jan or Jun.)
5. Day of week (hold values between 0-6 or Sun-Sat, Here also you can use first three letters of each day's name i.e Sun or Wed.)
6. Command

```
# crontab -l
# crontab -e

# crontab -u tecmint -l
# crontab -r

# crontab -i -r
```

==========

How to Compile Linux Kernel on CentOS 7

Running a custom compiled Linux Kernel is always useful, specially when you are looking to enable or disable specific Kernel features, which are not available in default distribution-supplied kernels.

Install Required Packages for Kernel Compilation

```
# yum update
# yum install -y ncurses-devel make gcc bc bison flex elfutils-libelf-devel openssl-devel grub2
```

Compile and Install Kernel in CentOS 7

```
# cd /usr/src/
# wget  https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.6.13.tar.xz

# tar -xvf  linux-5.6.13.tar.xz

# cd  linux-5.6.13

#cp /boot/config-3.10.0-1127.el7.x86_64  .config

# cd /usr/src/linux-5.6.13
# make menuconfig

# make bzImage
# make modules
# make
# make install
# make modules_install
# uname -sr
```

==========

Introduction to RAID, Concepts of RAID and RAID Levels

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

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

Software RAID and Hardware RAID

Software RAID have low performance, because of consuming resource from hosts. Raid software need to load for read data from software raid volumes. Before loading raid software, OS need to get boot to load the raid software. No need of Physical hardware in software raids. Zero cost investment.

Hardware RAID have high performance. They are dedicated RAID Controller which is Physically built using PCI express cards. It won't use the host resource. They have NVRAM for cache to read and write. Stores cache while rebuild even if there is power-failure, it will store the cache using battery power backups. Very costly investments needed for a large scale.

Featured Concepts of RAID

1. Parity method in raid regenerate the lost content from parity saved information's. RAID 5, RAID 6 Based on Parity.
2. Stripe is sharing data randomly to multiple disk. This won't have full data in a single disk. If we use 3 disks half of our data will be in each disks.
3. Mirroring is used in RAID 1 and RAID 10. Mirroring is making a copy of same data. In RAID 1 it will save the same content to the other disk too.
4. Hot spare is just a spare drive in our server which can automatically replace the failed drives. If any one of the drive failed in our array this hot spare drive will be used and rebuild automatically.
5. Chunks are just a size of data which can be minimum from 4KB and more. By defining chunk size we can increase the I/O performance.

RAID's are in various Levels. Here we will see only the RAID Levels which is used mostly in real environment.

1. RAID0 = Striping
2. RAID1 = Mirroring
3. RAID5 = Single Disk Distributed Parity
4. RAID6 = Double Disk Distributed Parity
5. RAID10 = Combine of Mirror & Stripe. (Nested RAID)s

RAID are managed using mdadm package in most of the Linux distributions. Let us get a Brief look into each RAID Levels.

Creating Software RAID0 (Stripe) on 'Two Devices' Using 'mdadm' Tool in Linux

```
yum clean all && yum update
yum install mdadm -y
ls -l /dev | grep sd
fdisk  -l
mdadm --examine /dev/sd[b-c]
fdisk /dev/sdb
mdadm --examine /dev/sd[b-c]
fdisk /dev/sdc
fdisk  -l
mdadm --examine /dev/sd[b-c]
```

mdadm --create /dev/md0 --level=0 --raid-devices=2 /dev/sdb1 /dev/sdc1
mdadm --create /dev/md1 --level=1 --raid-devices=2 /dev/sdb1 /dev/sdc1
mdadm --create /dev/md5 --level=5 --raid-devices=3 /dev/sdb1 /dev/sdc1 /dev/sdd1
mdadm --create /dev/md6 --level=6 --raid-devices=4 /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1
mdadm --create /dev/md10 --level=10 --raid-devices=4 /dev/sd[b-e]1

mdadm --stop /dev/md0
mdadm --stop /dev/md1
mdadm --stop /dev/md5
mdadm --stop /dev/md6
mdadm --stop /dev/md10

```
cat /proc/mdstat
mdadm -E /dev/sd[b-c]1
mdadm --detail /dev/md0
mkfs.ext4 /dev/md0
mkdir /mnt/raid0
mount /dev/md0 /mnt/raid0/
df -h
cd /mnt/raid0/
touch /mnt/raid0/tecmint.txt
echo "Hi everyone how you doing ?" > /mnt/raid0/tecmint.txt
cat /mnt/raid0/tecmint.txt
ls -l /mnt/raid0/

vim /etc/fstab
/dev/md0                /mnt/raid0              ext4    defaults        0 0

mount -av
mdadm -E -s -v >> /etc/mdadm.conf
cat /etc/mdadm.conf
mdadm --detail --scan --verbose >> /etc/mdadm.conf
cat /etc/mdadm.conf
fdisk  -l
df -h
cd /mnt/raid0/


TO stop array
mdadm  --stop /dev/md0
```

==========

Setup Flexible Disk Storage with Logical Volume Management (LVM)

Logical Volume Management (LVM) makes it easier to manage disk space. If a file system needs more space, it can be added to its logical volumes from the free spaces in its volume group and the file system can be re-sized as we wish. If a disk starts to fail, replacement disk can be registered as a physical volume with the volume group and the logical volumes extents can be migrated to the new disk without data loss.

LVM Features

1. It is flexible to expand the space at any time.
2. Any file systems can be installed and handle.
3. Migration can be used to recover faulty disk.
4. Restore the file system using Snapshot features to earlier stage. etc…

Pvs
Vgs
Lvs
lsblk
partprobe

```
pvcreate /dev/sdb1
vgcreate test /dev/sdb1
lvcreate -n check -L 8G test
lsblk

lvremove  /dev/test/check
vgremove /dev/test
pvremove /dev/sdb1
lsblk
```

fdisk /dev/sdb
fdisk /dev/sdc

```
fdisk -l
pvs
pvcreate /dev/sdb1 /dev/sdc1
pvs
vgs
vgcreate -s 32M test_add_vg /dev/sdb1 /dev/sdc1
vgs
vgs -v
vgdisplay test_add_vg
lvs
pvs
vgdisplay  test_add_vg
lvcreate -L 18G -n test_lvm test_add_vg
lvs
mkfs.ext4 /dev/test_add_vg/test_lvm
mkdir /mnt/lvm
mount /dev/test_add_vg/test_lvm  /mnt/lvm/
df -h
cd /mnt/lvm/
```

How to Extend/Reduce LVM's

Logical Volume Extending

```
# pvs
# vgs
# lvs

# fdisk -l /dev/sda

# pvcreate /dev/sda1
# pvs

# vgextend test_add_vg /dev/sda1
# vgs

# pvscan

# vgdisplay

# lvextend -L 36G /dev/test_add_vg/test_lvm
# resize2fs /dev/test_add_vg/test_lvm

# lvdisplay

# vgdisplay
```

Reducing Logical Volume (LVM)

```
# lvs

# umount -v /mnt/lvm

# e2fsck -ff  /dev/test_add_vg/test_lvm

# resize2fs /dev/test_add_vg/test_lvm

# lvreduce -L 5G /dev/test_add_vg/test_lvm
```

lvdisplay test_add_vg

mount /dev/test_add_vg/test_lvm /mnt/lvm

lvdisplay test_add_vg

##TESTED STEPS##

pvcreate /dev/sdb1 /dev/sdc1 /dev/sdd1
pvs
vgcreate vg1 /dev/sdb1 /dev/sdc1 /dev/sdd1
vgs

CREATION
lvcreate -L +3G -n lv1 vg1
mkfs.ext4 /dev/vg1/lv1

EXTEND
lvextend -L 6G /dev/vg1/lv1
resize2fs /dev/vg1/lv1 +6G
lvdisplay /dev/vg1/lv1
lvs

REDUCING
Umount /dev/vg1/lv1 /firstlv
e2fsck -f /dev/vg1/lv1
resize2fs /dev/vg1/lv1 +3G
lvreduce -L 3G /dev/vg1/lv1

VG REDUCE
vgreduce vgs /dev/sdd1
vgs
pvs

PVREMOVE
pvremove /dev/sdd1

LVM SNAP
lvcreate -L 4G -s -n mysnap /dev/vg1/lv1
lvs

EXTENDING SNAP
lvextend -L+1G /dev/vg1/mysnap
lvdisplay /dev/vg1/mysnap

SNAP RESTORE
lvconvert --merge /dev/vg1/snap1

Deactivate and activate you volume:;;[][p# .[[p/*lvchange -a n /dev/volume_group/volume1
lvchange -a y /dev/volume_group/volume1

==========

Nagios

Server Side

yum install -y httpd httpd-tools php gcc glibc glibc-common gd gd-devel make net-snmp
useradd nagios
groupadd nagcmd
usermod -G nagcmd nagios
usermod -G nagcmd apache
mkdir /root/nagios
cd nagios/
wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.5.tar.gz
wget https://nagios-plugins.org/download/nagios-plugins-2.2.1.tar.gz
tar xvf nagios-4.4.5.tar.gz
cd nagios-4.4.5/
./configure --with-command-group=nagcmd
make all
make install
make install-init
make install-commandmode
make install-config
vi /usr/local/nagios/etc/objects/contacts.cfg
make install-webconf
htpasswd -s -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
systemctl start httpd.service
tar xvf nagios-plugins-2.2.1.tar.gz
cd nagios-plugins-2.2.1/

```
./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
make install
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
systemctl enable nagios
systemctl enable httpd
systemctl start nagios.service
systemctl status httpd
systemctl status nagios
```

==========

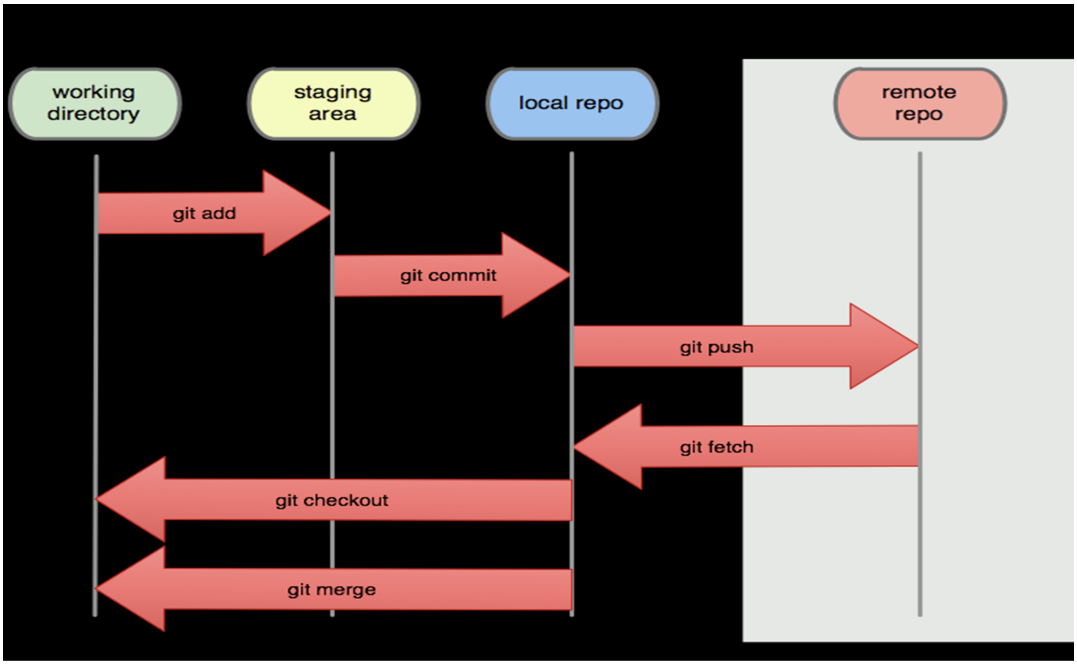
Git

Introduction to Git

- Git is an example of a Distributed Version Control System and source code management.
- It is a process of tracking and controlling changes in the software
- Free and open source
- Implicit backup
- Security(SHA1)
- No need of powerful hardware
- Local Repository
- Git doesn't works on client and server model.
- In SVN, for every commits it will create versions with files and additionally which increases the disk space this is the main disadvantage in SVN.
- Whereas in Git, instead of creating files git takes snapshots to reduce the file size.
- Git uses the checksums for every changes and commits with a commit id.
- Git contains three directories:
- Working directory
  - Staging Area
  - Git repository

• Three Main Components

- Working directory: This is the place where we can see the exact content of repository files physically from the server and users can write their source code and do the modifications we can call these files as a untracked files.
- Staging area: This area is called as Virtual area, like a snapshot will be taken after adding the data in to git. Every time you add the data, git do takes a snapshot at that particular instance instead of saving the complete file.
- Git repository: After committing, the data will be synchronized into this repository and same data contains in both working and repository directories. These files under this repository are called as Tracked files or unmodified files



CREATE NEW REPOSITORY ON THE COMMAND LINE

```
echo "# TEST" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/SureshKumarPakalapati/TEST.git
git push -u origin master
```

PUSH AN EXISTING REPOSITORY FROM THE COMMAND LINE

```
git remote add origin https://github.com/SureshKumarPakalapati/TEST.git
git push -u origin master
```

Got 15 minutes and want to learn Git?

```
git init
git status
git add suresh.txt
git status
git commit -m "Add cute octocat story"
git add '*.txt'
git commit -m 'Add all the octocat txt files'
git log
git remote add origin https://github.com/try-git/try_git.git
git push -u origin master
git pull origin master
git diff HEAD
git add suresh/suresh.txt
git diff --staged
git reset suresh/suresh.txt
git checkout -- suresh.txt
git branch clean_up
git checkout clean_up
git rm '*.txt'
git commit -m "Remove all the cats"
git checkout master
git merge clean_up
git branch -d clean_up
git push
```

- **Create a branch and make a change**
- git branch –b test_branch
- git branch -a
- git checkout test_branch
- vi test.txt
- git add test.txt
- git commit test.txt -m 'making a change in a branch‘
- **Merge your branch**

- git status
- git checkout master
- git merge test_branch
- **To delete a branch**
- git branch -d test_branch

==========

Jenkins

Continuous Integration tool

Why Jenkins?

Jenkins is a free and open source web based application

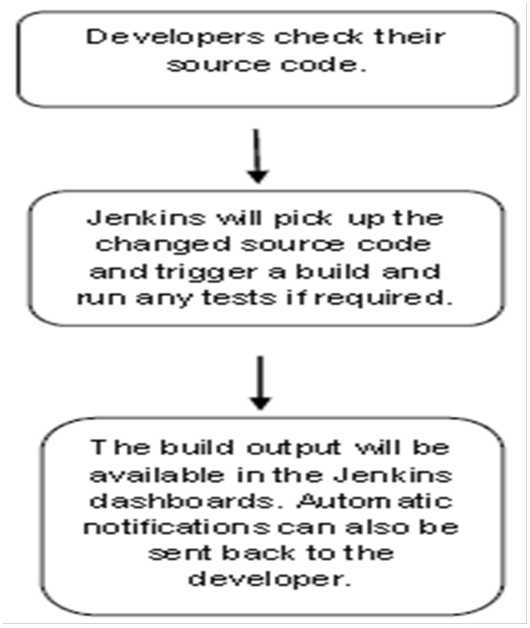
Continuous Integration:

- Continuous Integration is a development practice that requires developers to integrate code into a shared repository at regular intervals

Advantages:

- Should identify errors as very early in the process.
- Resulting artifacts are automatically created and tested.
- Quality and delivery of the product in a short time
- Perform a software build with Apache Maven.
- Archive the build result.
- Integration.

How Jenkins Works?



Jenkins Operations

- Downloading the source code from Version control system GIT into Jenkins
- Performing the build using build tools like ANT/MAVEN.
- Analyzing the source code using Sonarqube
- Pushing the Artifacts into various artifact repositories on Nexus.

- Deploying the artifacts into WebLogic server
- Update Jira Ticket
- Trigger Email notification on build status

Conclusion

- Continuous integration is a necessity on complex projects due to the benefits it provides regarding early detection of problems.
- Jenkins, a continuous build system, can be an integral part of any continuous integration system due to it's core feature set and extensibility through a plugin system.

- ==========

RHCSA

Red Hat Certified System Administrator (RHCSA) exam (EX200) that includes the topics described below:

1. Chapter 1: Understanding Essential Commands & System Documentation
2. Chapter 2: How to Perform File and Directory Management
3. Chapter 3: Text Editors and Regular Expressions
4. Chapter 4: Managing User Accounts
5. Chapter 5: Understanding the Boot Process and Process Management
6. Chapter 6: Operate and Manage Running Systems
7. Chapter 7: Setting Up and Configuring Local Storage
8. Chapter 8: File Systems Formats and Installing & Mounting Network Shares
9. Chapter 9: Network Operations (Manage Basic Networking)
10. Chapter 10: Package Management and System Logs
11. Chapter 11: Basics of FirewallD and IPtables
12. Chapter 12: Learn SELinux Basic Concepts and Operations
13. Chapter 13: How to Install Stratis to Manage Layered Local Storage
14. Chapter 14: How to Create Local HTTP Yum/DNF Repository
15. Chapter 15: How to Create a VDO Volume On a Storage Device
16. Chapter 16: How to Configure and Manage Basic Networking

RHCE

Red Hat Certified Engineer (RHCE) exam (EX294) that includes the topics described below:

1. Chapter 1: Reviewing Essential Commands & System Documentation
2. Chapter 2: How to Perform File and Directory Management
3. Chapter 3: Text Editors and Regular Expressions
4. Chapter 4: Managing Users and Groups
5. Chapter 5: Understanding the Boot Process and Process Management
6. Chapter 6: Operate and Manage Running Systems
7. Chapter 7: Setting Up and Configuring System Storage
8. Chapter 8: File Systems Formats and Installing & Mounting Network Shares
9. Chapter 9: Network Operations (Manage Basic Networking)
10. Chapter 10: Package Management and System Logs
11. Chapter 11: Basics of FirewallD and IPtables
12. Chapter 12: Learn SELinux to Manage Security
13. Chapter 13: How to Install Stratis to Manage Layered Local Storage
14. Chapter 14: How to Create Local HTTP Yum/DNF Repository
15. Chapter 15: How to Create a VDO Volume On a Storage Device
16. Chapter 16: Understand Core Components of Ansible
17. Chapter 17: How to Install and Configure an Ansible Control Node
18. Chapter 18: Configure Ansible Managed Nodes and Run ad-hoc Commands
19. Chapter 19: How to Use Static and Dynamic Inventories in Ansible
20. Chapter 20: How to Create Ansible Plays and Playbooks
21. Chapter 21: How to Use Ansible Modules for System Administration Tasks
22. Chapter 22: How to Create Templates in Ansible to Create Configurations On Managed Nodes
23. Chapter 23: How to Work with Ansible Variables and Facts
24. Chapter 24: How to Create and Download Roles on Ansible Galaxy and Use Them
25. Chapter 25: How to Use Ansible Vault in Playbooks to Protect Sensitive Data

==========

DNS

[root@server ~]# cat /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.
//
// See the BIND Administrator's Reference Manual (ARM) for details about the
// configuration located in /usr/share/doc/bind-{version}/Bv9ARM.html

options {
        listen-on port 53 { 127.0.0.1; any; };
#       listen-on-v6 port 53 { ::1; };
        directory       "/var/named";
        dump-file       "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
        recursing-file  "/var/named/data/named.recursing";
        secroots-file   "/var/named/data/named.secroots";
        allow-query     { localhost; any; };

        /*
         - If you are building an AUTHORITATIVE DNS server, do NOT enable recursion.
         - If you are building a RECURSIVE (caching) DNS server, you need to enable
           recursion.
         - If your recursive DNS server has a public IP address, you MUST enable access
           control to limit queries to your legitimate users. Failing to do so will
           cause your server to become part of large scale DNS amplification
           attacks. Implementing BCP38 within your network would greatly
           reduce such attack surface
        */
        recursion yes;

        dnssec-enable yes;
        dnssec-validation yes;

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

zone "." IN {
        type hint;
        file "named.ca";
};

zone "example.com" IN {
type master;
file "fwd.example.com.db";
allow-update { none; };
};

zone "31.168.192.in-addr.arpa" IN {
type master;
file "31.168.192.db";
allow-update { none; };
};

include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
```

[root@server ~]# cat /var/named/fwd.example.com.db

```
$TTL    604800

@       IN      SOA     server.example.com. root.example.com. (
                              6             ; Serial
                         604820             ; Refresh
                          86600             ; Retry
                        2419600             ; Expire
                         604600 )           ; Negative Cache TTL

;Name Server Information
@       IN      NS      server.example.com.
@       IN      NS      client.example.com.
;IP address of Your Domain Name Server(DNS)
server IN       A       192.168.31.100
```

```
client IN      A      192.168.31.101

[root@server ~]# cat /var/named/31.168.192.db
$TTL    604800
@       IN      SOA     server.example.com. root.example.com. (
                              21            ; Serial
                          604820        ; Refresh
                           864500        ; Retry
                          2419270       ; Expire
                           604880 )      ; Negative Cache TTL

;Your Name Server Info
@       IN      NS      server.example.com.
@       IN      NS      client.example.com.
;Reverse Lookup for Your DNS Server
100     IN      PTR     server.example.com.
101     IN      PTR     client.example.com.

[root@server ~]# cat /etc/hostname
server.example.com
[root@server ~]#


[root@server ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.31.100 server server.example.com

[root@server ~]# cat /etc/resolv.conf
# Generated by NetworkManager
search server.example.com
nameserver 192.168.31.100
nameserver 8.8.8.8
nameserver 8.8.4.4
[root@server ~]#

[root@client ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.31.101 client client.example.com
[root@client ~]#


[root@client ~]# cat /etc/hostname
client.example.com
[root@client ~]#


[root@client ~]# cat /etc/resolv.conf
# Generated by NetworkManager
search server.example.com
nameserver 192.168.31.100
[root@client ~]#
```

==========

ITIL Service Operation

ITIL Service Operation involves managing the smooth delivery of IT services with the ultimate goal of delivering value to the business. Service Operation must be aware of the changing needs within business based on advancing technology, such as cloud computing and cloud security needs. Service operation is made up of five processes: Incident Management, Event Management, Access Management, Request Fulfillment, Problem Management.

1. **Incident Management** is the process of taking action to rapidly restore interruptions in service due to incidents. Incidents may include, password resets, printer failure, or an error message. (For more information, including Incident Management benefits, roles and responsibilities, process flow, KPIs, and implementation best practices, read the Essential Guide to ITIL Incident Management.)
2. **Problem Management** works to pinpoint and prevent the recurrence problems and incidents. (For more in depth coverage of Problem Management, read the Essential Guide to ITIL Problem Management.)
3. **Event Management** examines and analyzes all service events that may arise from applications, monitoring solutions, and other systems so that action, if needed, can be taken to ensure service continuity.
4. **Access Management** controls who has access to the systems by preventing unauthorized attempts to access the system while allowing access for legitimate users.
5. **Request Fulfillment** process includes receiving, logging, prioritizing, and resolving service requests received by the service desk.

From <https://www.cherwell.com/library/essential-guides/essential-guide-to-itil-framework-and-processes/>

2) Common Standard Ports Used :

ANS: =

= 21/20 ftp

= 22 ssh

= 23 telnet

= 25 smtp

= 53 DNS (tcp/udp)

= 68 DHCP

= 69 TFTP

= 80/443 http/https (tcp)

= 88/464 Kerberos (tcp/udp)

= 110 pop3

= 123 NTP(udp)

= 137 nmbd

= 138,139,445 smbd

= 143 IMAP

= 161 SNMP

= 389/636 LDAP/LDAPS (tcp)

= 514 (udp) syslogd

= 2049 NFS

From <https://codingcompiler.com/linux-administrator-interview-questions-answers/>

Job Description for Unix L4/ Unix Engineering in Novartis

Job Title:	**Engineering Expert – Unix**
Work Experience:	**12 – 15 Years**
Work Location:	**Hyderabad (for now WFH)**
Good to have Skills:	**Understanding of Public Cloud (Azure/AWS)**
Reports to:	**Engineering Manager – Unix**

Job Requirements

Key Responsibilities:

- Ready for 24X7 Support in rotational manner
- DevOps inclination, would be required to work on Unix related tools
- Performance Tuning and architecture design
- Analyze how new technology could benefit their organization
- Engineer roadmaps to integrate that technology into existing systems
- Draft support documents and updating operations manuals, standard operating procedures (SOPs), life-cycle documentation
- Research emerging technology trends
- Provide subject matter expertise to maintain overall system health and stability
- Collaborate with Operations team to help them with Critical issues/outages
- Validate Root Cause Analysis for the critical outages
- Collaborate with vendors for smooth introduction of new solutions/services for operations
- Performance tuning, High Availability design, Security hardening, Release management
- Good understanding of networking and security principles
- Knowledge of virtualization and cloud concepts
- Knowledge of Agile methodology
- Knowledge of ITIL processes

Technical Expertise:

- Strong Hands-on experiencein Linux OS (RedHat Linux 6,7 & 8, Ubuntu, SuSE) & Aix 7.x
- Good Knowledge on scripting (Shell, python, bash)

- Strong Hands-on experience in OS internals, system calls, advanced system administration
- Additional Skills – Ansible, Ansible Tower, Redhat Satellite, RedHat Open Shift
- Redhat Clusters
- Should have supported Oracle RAC clusters
- Strong grasp of Unix-based operating system processes, such as paging and swapping, file system concepts, inter-process communication, and variations between virtual and physical systems
- Strong understanding of IT Infrastructure setup (Server, Storage (NAS/SAN), VMWare, Networking)

Professional Attributes:

- Able to prioritize and execute tasks in a high-pressure environment
- Experience working in a team-oriented, collaborative environment
- Should be able to initial level of trouble shootings
- Flexible to work in 24/7 On-Call (Rotational Basis)
- Collaboration and strong interpersonal communication skills
- Performance Tuning and architecture design
- Strong Troubleshooting
- Solution designing
- Problem solving
- Collaboration
- Innovation
- Value generation

Additional Information:

- Engineering graduate
- Minimum 12 years of experience in Linux 6, 7 & 8 & Aix 7.x
- Experience of working in a GxP/SOX qualified environment
- Experience of working with cross-functional teams
- Experience of working with global teams including off-shore teams
- Experience of leading technical projects
