1. Brief History of Unix + Linux
2. Installation of RHEL 7 (step by step)
3. First boot screen
4. some admin commands

   like

   - finger
   - whoami
   - w

UNIX

- UNiplexed Information & Computing Services
- Developed at AT & T's Bell labs in 1969.
- Linux is officially announced from Jan 1st, 1990.
- Later Linux was developed as "free open source" for personal computer on x86 architecture.
- Linux is an open source, free to use O.S used for hardware, software, game development.
- Unix is commonly used for internet servers, workstations & PC's in IBM, HP.

Architecture of UNIX

User (applications) -> [Shell] -> Unix commands -> [Kernel] -> Hardware.

Kernel

- is a part of unix OS, & is loaded when system is booted.
- Helps in
  - managing process
  - resource allocaions
  - between shell & hardware.

initial process that firstly activated inside HDD, name of kernel process is init.

Shell

- command interpreter, which interpretes the commands
- interface between user & kernel
- translate user defined language INTO machine language to kernel.
- variants of shell are avilable

  In UNIX

  - Korn shell ($ prompter)
  - Bourn shell ($ prompter)
  - c shell (% prompter)

In LINUX

- Bash shell (bourn again shell, default in linux)
- tcsh shell (Turbo C shell) // lil older

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

Run level numbers

- init 0 Halt the system.
- init 1 Single-user mode (for special administration).
- init 2 Local Multiuser with Networking but without network service (like NFS)
- init 3 Full Multiuser with Networking
- init 4 Not Used
- init 5 Full Multiuser with Networking and X Windows(GUI)
- init 6 Reboot.

RUNLEVEL

- A run level is a state of init and the whole system that defines what system services are operating.
- The first thing the kernel does is to execute init program.
- Init is the root/parent of all processes executing on Linux
- The first processes that init starts is a script /etc/rc.d/rc.sysinit. Based on the appropriate run-level, scripts are executed to start various processes to run the system and make it functional.

DAY 2

1. File mgmt
2. directory mgmt
3. disk mgmt
4. utilities

File Mgmt

To create file

- cat
- touch
- vi / vim

CAT

to see the content of the file

- cat <fileName>

to save a file

- ctrl + d

to append the data in a exsisting file

- cat >> fileName

TOUCH - to create an empty file (0byte size)

to create a file

- touch t1
- touch t1 t2
- touch {t1, t2, t3}
- touch test{1,2,3}

to delete file / remove file

- rm fileName
- rm log // ends with log
- rm test* // starts with test
- rm ?est // ? means any letter

MANAGING DIRECTORIES

to make dir

- mkdir <dirName>

to go inside a dir

- cd <dirName>

to remove non empty dir

- rm -r <dirName>

ls -a

. => Current Directory

.. => Parent Directory

MANAGING LOCAL USER & GROUPS

1. managing users (create users acc, passwd)

database files

a. /etc/passwd : it holds user account info

contains 7 columns, seperated by :

1. login name
2. password (encrypted)-> shadow file
3. user id
4. group id
5. comment
6. home directory
7. login shell

b. /etc/shadow: it holds user account passwd & passwd policies

contains 9 feilds

1. Login name

2. encrypted passwd (always of same length)

3. 16924 : passwd set/ modify from 1st Jan 1970

4. 0: mini passwd age (days)

5. 99999: max passwd age (days)

6. 7: warning to change the passwd(days)

7. : inactive days - ex: james wants to go for 30 days, we have to type (16924+30=16954)

8. : account expiry - 16924+(number of days for active)

9. : flag - not dveloped by UNIX yet

2. Basic commands

#useradd james

#passwd james

#usermod james modify user account details

#userdeljames delete user account

#finger user account info

#id to view 'uid' & 'gid'

to add user with specific values

#useradd -u 5000 -c 'system admin' mac

-u = user id

-c = comment

to delete any user

remove user acc w/o home dir

#userdel mac

remove user acc with home dir

#userdel -r mac

to list passwd policies

#chage -l mac

install rpm

#rpm -ivh --nodeps /run/media/root/RHEL7...../Packages/system-config-users-docs.....
.rpm

#

Group management

#groupadd adding a group

#groupmod modify a group

#groupdel del a group

#cat /etc/group

4 fields

1. group name
2. passwd
3. Gid
4. group members

Package Mgmt

Packages

- packages are having ".rpm" (Redhat Packet Manager)

Naming Convention

package-name:

- name-version-release-architecture.rpm

{ architecture

i686 (32 bits)

x86_64 (64 bits)

noarch (h/w independent) }

To manage Packages

can be managed by 2 ways

1. RPM (REDHAT PACKAGE MANAGER)
2. YUM (YELLODOG UPDATER MANAGER)

RPM vs YUM

- RPM is good for independent packages, as they didn't required for other packages
- YUM will detect all dependencies required for the packages and will install all supporting packages automatically.

MANAGING PACKAGES USING RPM COMMAND

1. to list all the installed packages

#rpm -qa // q=query, a=all

2. to check specific package installed or not

#rpm -q <httpd>

3. to install the package

#rpm -ivh <package name>

i=install

v=verbose mode, display summary during installation

h= hash

4. to upgrade the package

#rpm -Uvh <package name>

5. to uninstall the package

#rpm -e dhcp

e=erase

==========RPM over==========

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 config over==========

1. to list all installed packages

#yum list installed

2. to install the package

#yum install <package name>

if any error occurs, then type

#setenforce 0

#yum install dhcp

y=yes

n=no

d=download

3. to remove package

#yum remove dhcp

without user interaction

#yum remove dhcp -y

:- 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

Table of process

- #top

SELINUX - security built-ON kernal

#cat /etc/sysconfig/selinux

- Security Enhanced Linux
- Security of kernal of an o.s
- built-in, by-default enabled

Operates in 3 modes

1. Enforcing - (means SELinux is active)

2. Permissive - (means partially active - generates log, but wont deny if attacked)

3. Disabled - (means SELinux is inactive)

Check SELinux status

#getinforce

SELinux status can be changed by using by (temporarly)

#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, kernal 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.

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 changed 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 to check dns looup

1. nslookup <url>
2. hostname <url>
3. dig <url>

==========NETWORKING OVER==========

How to configure KICKSTART

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

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

Partitioning

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)

- total size = 512 bytes (size cannot be modified)
- 1st sector on hdd
- 446 bytes = GRUB (GRand Unified Boot) loader
- 64 bytes = Partition info (hdd geometry, 16 bytes per partition)
- 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

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