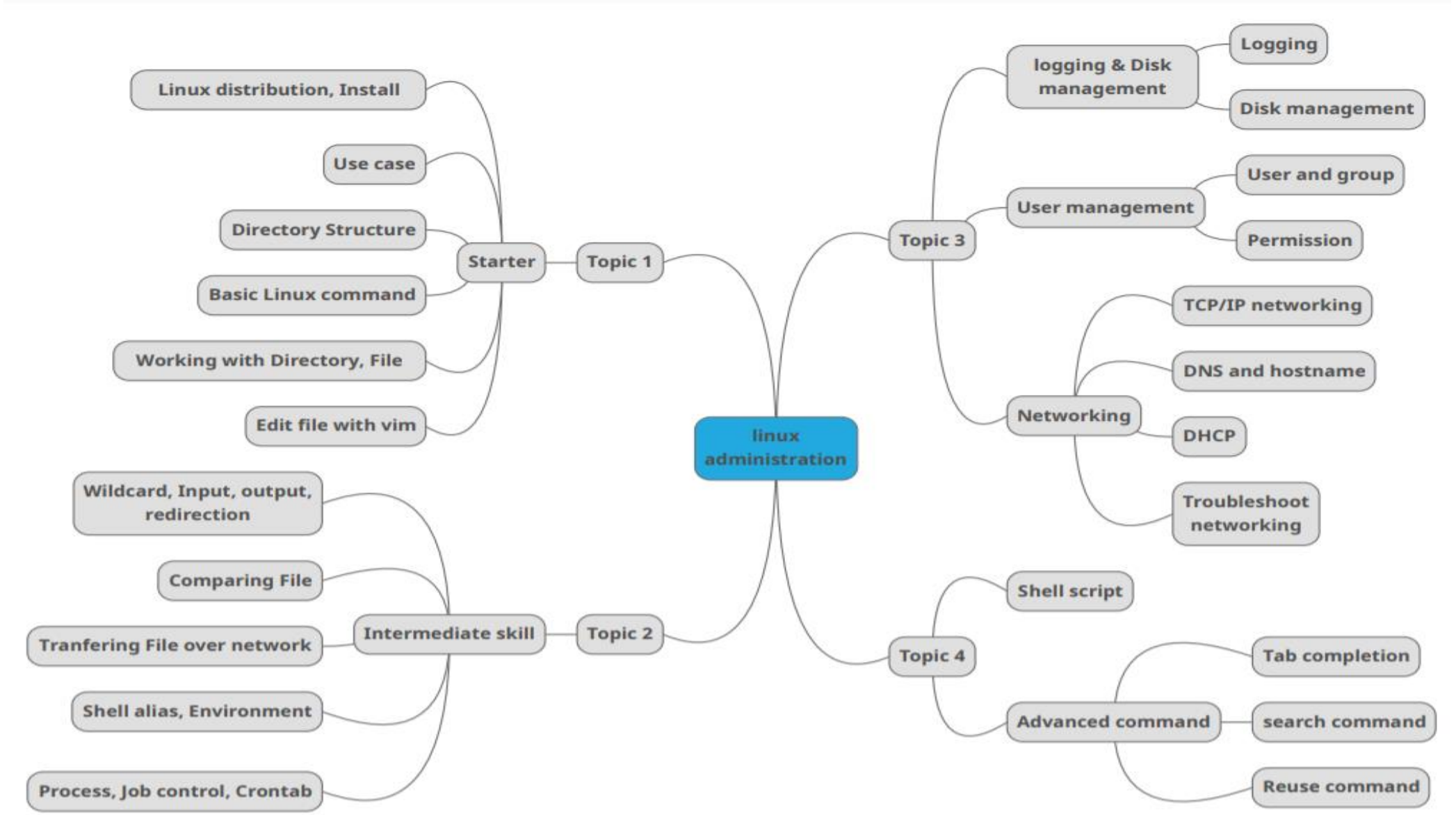
TechMaster
Học là có việc

Linux administration

Nguyễn Hàn Duy
duy@techmaster.vn

Nội dung


linux administration
Topic 1
Starter
Linux distribution, Install
Use case
Directory Structure
Basic Linux command
Working with Directory, File
Edit file with vim
Topic 2
Intermediate skill
Wildcard, Input, output, redirection
Comparing File
Tranfering File over network
Shell alias, Environment
Process, Job control, Crontab
Topic 3
logging & Disk management
Logging
Disk management
User management
User and group
Permission
Networking
TCP/IP networking
DNS and hostname
DHCP
Troubleshoot networking
Topic 4
Shell script
Advanced command
Tab completion
search command
Reuse command

Switching Users and Running Commands as Others

What You Will Learn

- How to switch to another account.
- How to run commands as others.

The `su` Command

`su [username]` Change user ID or become superuser

`su` Options

`-`	A hyphen is used to provide an environment similar to what the user would expect had the user logged in directly.
`-c command`	Specify a command to be executed.

`whoami` Example

```
$ whoami
jason
$ su oracle
Password:
$ whoami
oracle
```

Sudo - Super User Do

`sudo` Execute a command as another user, typically the superuser.

Using sudo

`sudo -l`	List available commands.
`sudo command`	Run command as root.
`sudo -u root command`	Same as above.
`sudo -u user command`	Run as user.

Using sudo

`sudo su`	Switch to the superuser account.
`sudo su -`	Switch to the superuser account with root's environment.
`sudo su - username`	Switch to the username account.

Shell History and Autocompletion

What You Will Learn

Shell History
Exclamation Mark Syntax
Autocompletion

Shell History

Executed commands are added to the history.
Shell history can be displayed and recalled.
Shell history is stored in memory and on disk.

- ~/.bash_history
- ~/.history
- ~/.histfile

`history` Command

`history` Displays the shell history.

`HISTSIZE` Controls the number of commands to retain in history.

```
export HISTSIZE=1000
```

! Syntax

`!N`	Repeat command line number N.
`!!`	Repeat the previous command line.
`!string`	Repeat the most recent command starting with "string."

! Syntax Examples

```
$ head files.txt sorted_files.txt notes.txt
<Output from head command here>
$ !!
head files.txt sorted_files.txt notes.txt
<Output from head command here>
$ vi !:2
vi sorted_files.txt
<vi editor starts>
```

Searching Shell History

`Ctrl-r`	Reverse shell history search
Enter	Execute the command
Arrows	Change the command
Ctrl-g	Cancel the search

Tab Completion

`Tab` autocompletion

- Commands
- Files, directories, paths
- Environment Variables
- Usernames (~)

System Logging

What You Will Learn

- The syslog standard
- Facilities and severities
- Syslog servers
- Logging rules
- Where logs are stored
- How to generate your own log messages
- Rotating log files

The Syslog Standard

- Aids in the processing of messages.
- Allows logging to be centrally controlled.
- Uses facilities and 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

Number	Keyword	Description
11	ftp	FTP daemon
12	-	NTP subsystem
13	-	log audit
14	-	log alert
15	cron	clock daemon
16	local0	local use 0 (local0)
16	local1	local use 0 (local1)
16	local2	local use 0 (local2)
16	local3	local use 0 (local3)
...		
23	local7	local use 7 (local7)

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

Syslog Servers

- Process syslog messages based on rules.
- syslogd
- rsyslog
- syslog-ng

Logging Rules

- Selector field
  - **FACILITY.SEVERITY**
  - **mail.***
  - **mail**
  - **FACILITY.none**
  - **FACILITY_1.SEVERITY; FACILITY_2.SEVERITY**
- Action field
  - Determines how a message is processed

Example Logging Rules

```
mail.info       -/var/log/mail.info
mail.warn       -/var/log/mail.warn
mail.err        /var/log/mail.err
```

Disk Management

Partitions

What You Will Learn

- Partitions
- MBR
- GPT
- Mount points
- fdisk

What You Will Learn

- Creating file systems
- Mounting file systems
- Unmount file systems
- How to prepare swap space for use
- File System Table
- Disk UUIDs and Labels

Partitions

- Disks can be divided into parts, called partitions.
- Partitions allow you to separate data.
- Partitioning schemes
  - 1) OS, 2) Application, 3) User, 4) Swap
  - 1) OS, 2) User home directories
  - As a system administrator, you decide.

Partitioning

- Can protect the overall system.
- Keep users from creating outages by using a home directory partition.

```
$ df -h
Filesystem  Size  Used  Avail Use%  Mounted on
/dev/sda2   100G   75G   25G   75%  /
/dev/sda1   488M  111M  342M   25%  /boot
/dev/sda3    10G   10G     0  100%  /home
```

MBR

- Master Boot Record
- Can only address 2 TB of disk space
- Being phased out by GPT
  - GPT= GUID Partition Table
- 4 Primary Partitions
- Extended partitions allow you to create logical partitions

GPT

- GPT = GUID Partition Table
- GUID = Global Unique Identifier
- Replacing the MBR partitioning scheme
- Part of UEFI
- UEFI = Unified Extensible Firmware Interface
- UEFI is replacing BIOS

fdisk

- Alternatives: gdisk, parted
- Earlier versions of fdisk did not support GPT

```
fdisk /path/to/device
```

mkfs

```
mkfs -t TYPE DEVICE

mkfs -t ext3 /dev/sdb2

mkfs -t ext4 /dev/sdb3
mkfs.ext4 /dev/sdb3
```

mkfs

```
# ls -1 /sbin/mkfs*
/sbin/mkfs
/sbin/mkfs.btrfs
/sbin/mkfs.cramfs
/sbin/mkfs.ext2
/sbin/mkfs.ext3
/sbin/mkfs.ext4
/sbin/mkfs.minix
/sbin/mkfs.xfs
```

The df command

```
# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2       198G  1.7G  196G   1% /
devtmpfs        489M     0  489M   0% /dev
tmpfs           497M     0  497M   0% /dev/shm
tmpfs           497M  6.5M  491M   2% /run
tmpfs           497M     0  497M   0% /sys/fs/cgroup
/dev/sdb3       484G   73M  459G   1% /opt
```

Managing Users and Groups

What You Will Learn

- How to manage users and groups.
- Where user and group information lives.
- How to add, delete, and change users and groups.

Accounts have a:

- Username (or login ID).
- UID (user ID). This is a unique number.
- Default group.
- Comments.
- Shell.
- Home directory location.

useradd

```
useradd -c "Grant Stewart" -m
-s /bin/bash grant
```

Create a password using `passwd`

```
# passwd grant
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated
successfully
```

More `useradd` options

-g GROUP	Specify the default group.
-G GROUP1,GROUPN	Additional groups.

userdel [-r] username

```
# ls /home
eharris grant
# userdel eharris
# ls /home
eharris grant
# userdel -r grant
# ls /home
```

usermod

```
usermod [ options ] username
```

-c "COMMENT"	Comments account.
-g GROUP	Specify the default group.
-G GROUP1,GROUPN	Additional groups.
-s /shell/path	Path to the user's shell.

Networking

TCP/IP

What You Will Learn

- TCP/IP
- Classful networks
- Subnet masks
- Broadcast addresses
- CIDR
- Private address space

TCP/IP

- TCP/IP
  - Used for network communications
  - TCP = Transmission Control Protocol
  - IP = Internet Protocol
- TCP - controls data exchange
- IP - sends data from one device to another
- Hosts
  - devices on a network that have an IP address

IP Networking

- IP address
  - Example: 199.83.131.186
- subnet mask
  - Example: 255.255.255.0
- broadcast address
  - Example: 199.83.131.255
- octet.octet.octet.octet
  - octet values can be from 0 to 255

Determining Your IP Address

- `ip address`
  - `ip addr`
  - `ip a`
  - `ip address show` **or** `ip a s`

```
# ip address
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: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP qlen 1000
    link/ether 08:00:27:43:f5:18 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.122/24 brd 192.168.1.255 scope global dynamic eth0
       valid_lft 84249sec preferred_lft 84249sec
    inet6 fe80::a00:27ff:fe43:f518/64 scope link
       valid_lft forever preferred_lft forever
```

```
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.122  netmask 255.255.255.0  broadcast
192.168.1.255
        inet6 fe80::a00:27ff:fe43:f518  prefixlen 64  scopeid
0x20<link>
        ether 08:00:27:43:f5:18  txqueuelen 1000  (Ethernet)
        RX packets 82371  bytes 95773879 (91.3 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 32907  bytes 3386585 (3.2 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
```

hostnames

- human-readable name for an IP address
  - webprod01 = 10.109.155.174

DNS hostnames

- FQDN = fully qualified domain name
  - webprod01.mycompany.com
- TLD
  - .com, .net, .org, etc.
- Domains
  - below (to the left of) TLD
- sub-domain
  - below (to the left of) the domain

Displaying the hostname

```
$ hostname
webprod01
$ uname -n
webprod01
$ hostname -f
webprod01.mycompany.com
```

Setting the hostname

```
# hostname webprod01

# echo 'webprod01' > /etc/hostname

# vi /etc/sysconfig/network
  HOSTNAME=webprod01
```

Resolving DNS Names

- host
- dig

```
$ host www.mycompany.com
webprod01.mycompany.com has address 1.2.1.6
$ host 1.2.1.6
6.1.2.1.in-addr.arpa domain name pointer
www.mycompany.com.
$
```

Sample /etc/hosts file

```
127.0.0.1       localhost
1.2.1.6         webprod01.mycompany.com webprod01
10.11.12.14     webprod02.mycompany.com webprod02
10.11.12.15     webprod03.mycompany.com webprod03
10.11.13.7      dbcluster
```

DHCP

- Dynamic Host Configuration Protocol
- DHCP servers assign IP address to DHCP clients
  - IP Address
  - netmask
  - gateway
  - DNS servers

DHCP

- Each IP is "leased" from the pool of IP addresses the DHCP server manages.
  - The lease expiration time is configurable on the DHCP server. (1hr, 1day, 1 week, etc.)
  - The client must renew the lease if it wants to keep using the IP address. If no renewal is received, the IP is available to other DHCP clients.

Configuring a DHCP Client - RHEL

`ifconfig -a` or `ip link`

/etc/sysconfig/network-scripts/ifcfg-DEVICE
/etc/sysconfig/network-scripts/ifcfg-eth0
/etc/sysconfig/network-scripts/ifcfg-enp5s2
BOOTPROTO=dhcp

Assigning a Static IP Address - RHEL

/etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE=eth0
BOOTPROTO=static
IPADDR=10.109.155.174
NETMASK=255.255.255.0
NETWORK=10.109.155.0
BROADCAST=10.109.155.255
GATEWAY=10.109.155.1
ONBOOT=yes
```

Manually Assigning an IP Address

Format:

```
ip address add IP[/NETMASK] dev NETWORK_DEVICE
```

```
ip address add 10.11.12.13 dev eth0
```

```
ip address add 10.11.12.13/255.255.225.0 dev eth0
```

```
ip link set eth0 up
```

Network Troubleshooting

What You Will Learn

- ping
- traceroute / tracepath
- netstat
- tcpdump
- telnet

Testing Connectivity with Ping

Format:

```
ping HOST
ping -c COUNT HOST
```

Example:

```
ping -c 3 google.com
```

```
$ ping -c 3 google.com
PING google.com (216.58.2.7) 56 bytes of data.
64 bytes from 216.58.2.7: icmp_seq=1 ttl=53 time=20.1 ms
64 bytes from 216.58.2.7: icmp_seq=2 ttl=53 time=20.2 ms
64 bytes from 216.58.2.7: icmp_seq=3 ttl=53 time=23.9 ms

--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time
2004ms
rtt min/avg/max/mdev = 21.489/22.924/24.154/1.111 ms
```

```
# traceroute -n google.com
traceroute to google.com (216.58.2.7), 30 hops
max, 60 byte packets
Diagnosing Network Connections 413
 1  10.0.2.2  0.296 ms  0.178 ms  0.220 ms
 2  192.168.1.1  2.529 ms  2.713 ms  2.630 ms
 3  72.14.237.231  23.750 ms  22.087 ms
12.122.132.137  22.701 ms
 4  216.58.216.78  20.549 ms 12.250.16.30  22.904
ms 216.58.216.78  20.724 ms
```

The `netstat` Command

-n	Display numerical addresses and ports.
-i	Displays a list of network interfaces.
-r	Displays the route table. (netstat -rn)
-p	Display the PID and program used.
-l	Display listening sockets. (netstat -nlp)
-t	Limit the output to TCP (netstat -ntlp)
-u	Limit the output to UDP (netstat -nulp)

```
[jason@linuxsvr ~]$ netstat -i
Kernel Interface table
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0      1500     3975      0      0 0          2627      0      0      0 BMRU
lo       65536        8      0      0 0             8      0      0      0 LRU
[jason@linuxsvr ~]$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         10.0.2.2        0.0.0.0         UG        0 0          0 eth0
10.0.2.0        0.0.0.0         255.255.255.0   U         0 0          0 eth0
```

Packet sniffing with `tcpdump`

tcpdump

-n Display numerical addresses and ports.
-A Display ASCII (text) output.
-v Verbose mode. Produce more output.
-vvv Even more verbose output.

```
$ sudo tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
19:25:49.639495 IP linuxsvr.ssh > 10.0.2.2.64440: Flags [P.], seq 3312803324:3312803408, ack
2443835, win 40880, length 84
19:25:49.639586 IP linuxsvr.ssh > 10.0.2.2.64440: Flags [P.], seq 84:120, ack 1, win 40880, length 36
19:25:49.639750 IP 10.0.2.2.64440 > linuxsvr.ssh: Flags [.], ack 84, win 65535, length 0
19:25:49.639763 IP 10.0.2.2.64440 > linuxsvr.ssh: Flags [.], ack 120, win 65535, length 0
$ sudo tcpdump -Anvvv
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
19:44:27.067530 IP (tos 0x10, ttl 64, id 5120, offset 0, flags [DF], proto TCP (6), length 64)
    10.0.2.44.37534 > 10.0.2.15.80: Flags [P.], cksum 0xfe34 (incorrect -> 0xce40), seq 1:13, ack 1, win
683, options [nop,nop,TS val 1585227 ecr 1584441], length 12
E..@..@.@.(.....P..>.:.....4.....
..0K..-9GET /about
```

```
        telnet HOST_OR_IP PORT_NUMBER
$ telnet google.com 80
Trying 216.58.2.7...
Connected to google.com.
Escape character is '^]'.
GET /
HTTP/1.0 200 OK
^]
telnet> quit
closed.
```