

Red Hat Certified System Administrator

FULL COURSE PDF

Table of Contents

Class -4	2
Class 5	17
Class 6	25
Class 7	32
Class 8	49
Class 9	66
Class 10	78
Day 11	89
Day 12	98
Day 13	108
Day 14	118
Day 15	126
Day 16	136
Day 17	152
Day 18	160
Day 19	184
Day 20 - RHEL 9 NTP Service and 'Cronjobs' Services - NTP Part	194
Day 20 - RHEL 9 Tasks Scheduler using 'Cronjobs'	198
Day 21 - Working with RHEL Container	206

Session :01 : Class -04

Working with Linux CLI :

=====

```
[student@node1 ~] $
```

~ = Home Directory

Linux User's Types:

লিনাক্স সিস্টেমে ৩ (তিন) ধরনের ইউজার একাউন্ট থাকে -

⌘ root user: Administrator (#) - সিস্টেমের অ্যাডমিন ইউজার।

⌘ system user: service (mail/ftp/ssh/daemons) - cannot login (সিস্টেমে বিল্ট-ইন থাকে)

⌘ regular user: student, guest, sakib (\$) - যে সকল ইউজার একাউন্ট 'root' কর্তৃক তৈরি করা হয়।

```
[root@node1 Desktop] #
```

```
1      2          3      4
```

1: user name (যে ইউজার দিয়ে সিস্টেমে লগইন করা হয়েছে)

2: hostname (কম্পিউটারের/সিস্টেমের নাম)

3: user's (ইউজারের বর্তমান অবস্থান)

4: user types (root: #, regular user: \$)

Working with Linux Shells & Terminal :

User -> Keyboard -> Terminal/Application -> Shell -> Kernel -> Hardware (CPU/MEM/DISK)

Screen <- Terminal/Application <- Shell <- Kernel <- Hardware

Linux Shell পরিচিতিঃ

লিনাক্স অপারেটিং সিস্টেমের একটি গুরুত্বপূর্ণ ও স্পেশাল প্রোগ্রাম হচ্ছে লিনাক্স শেল। লিনাক্সের শেল ইউজারের কাছ থেকে কমান্ড গ্রহণ করে এবং যদি ইউজার কর্তৃক প্রদত্ত

কমান্ড সঠিক হয়, তাহলে সিস্টেমের শেল এই কমান্ডকে কার্নেলের নিকট প্রেরণ করে অপারেটিং সিস্টেমের সঙ্গে তথ্য আদান-প্রদান করার জন্য লিনাক্স ইউজার ইন্টারফেস প্রদান করে। আবার প্রদত্ত কমান্ড যদি সঠিক না হয়, তাহলে 'Bash: Command not Found' দেখাবে। লিনাক্স শেল হচ্ছে একটি কমান্ড ল্যাঙ্গুয়েজ ইন্টারপ্রেটার, যেটা কিবোর্ড অথবা ফাইল হতে কমান্ড গ্রহণ করে এবং যেকোনো Programming Syntax বা কমান্ডকে সরাসরি execute করতে পারে।

Types of Linux Shell:

- ⌘ sh
- ⌘ Bash (commonly used)
- ⌘ cshell
- ⌘ tcshell
- ⌘ kshell
- ⌘ zsh

বর্তমানে কোন শেল ব্যবহার হচ্ছে এবং আমাদের সিস্টেমে এবং কি কি শেল ইন্সটল আছে সেটা জানার জন্য নিচের কমান্ডঃ

```
[student@node1 Desktop]$ echo $SHELL
[student@node1 Desktop]$ chsh -l
/bin/sh
/bin/bash
/usr/bin/sh
/usr/bin/bash
```

Linux Consoles:

⌘ Virtual (Physical) Consoles:

⌘ Ctrl + Alt + F1/F2: GUI

⌘ Ctrl + Alt + F3-F6: CMD (tty)

⌘ Pseudo Consoles (/dev/pts/x) - যে সকল টার্মিনাল আমরা গ্রাফিক্যালি রান করি, সেগুলোকে বলা হয়, 'Pseudo' কনসোল।

⌘ Virtual (Remote) Console (Telnet, SSH, RDP, VNC) - যে সকল টার্মিনাল আমরা রিমোট থেকে লগিন করি।

⌘ Web Console using https (cockpit) - ওয়েব ব্রাউজার দিয়ে লগিন করলে সেটাকে ওয়েব কনসোল বলে।

Classic Desktop এর ক্ষেত্রেঃ

Alt + Ctrl + F1 => GUI - seat0

Alt + Ctrl + F3 => new CMD terminal (F3-F6) - tty3-tty6

নোটঃ ফিজিক্যাল কনসোল দিয়ে লগইন করলে এবং টার্মিনাল থেকে 'who' কমান্ড দিলে সেখানে 'tty' দেখা যাবে।
উদাহরণ স্বরূপ কেউ যদি, Alt + Ctl + F3 দিয়ে লগইন করে, তাহলে 'who' কমান্ড দিলে 'tty3'। কিছু কিছু ক্ষেত্রে গ্রাফিক্যাল টার্মিনালের ক্ষেত্রে ':0' দেখাবে।

⌘ Press Ctl + Alt + F3

Login: student

pass: *****

[student@node1 Desktop]\$ su - ; রেগুলার ইউজার থেকে 'root' ইউজারে সুইচ করা হল।

Password: *****

[root@node1 ~]# exit ; 'root' ইউজার থেকে বের হওয়া।

[student@node1 ~]\$ exit ; 'student' ইউজার থেকে লগ আউট হওয়া।

Enable Web Console (Cockpit):

```
$ systemctl enable --now cockpit.socket (Root Authentication Required)
```

```
⌘ Open Firefox web browser
```

```
⌘ type: https://localhost:9090
```

```
username: root
```

```
password: redhat
```

লিনাক্সে বহুল ব্যবহৃত কিছু শর্টকাট কমান্ডঃ

```
$ ctrl + l - টার্মিনাল পরিষ্কার (clear) করার জন্য।
```

```
$ ctrl + shift + "t" - গ্রাফিক্যাল ইউজার ইন্টারফেস দিয়ে (GUI) নতুন ট্যাব খোলার জন্য।
```

```
$ Ctrl + d - কমান্ড কনসোলে লগ আউট করার জন্য ব্যবহার হয়।
```

```
$ Alt + F2 - নতুন কমান্ড রান করার জন্য। (Type 'firefox' and press Enter)
```

Linux Command Syntax/Pattern :

প্রতিটি কমান্ড লাইন ইন্টারফেস ভিত্তিক (CLI) অপারেটিং সিস্টেম বা ডিভাইসে কমান্ড লাইন ইন্টারফেস নিয়ে কাজ করতে চাইলে, প্রত্যেকটি কমান্ডের ০৩ (তিন) টা পার্ট থাকে। প্রথম টা হচ্ছে কমান্ড, যেটা দিতেই হবে এবং শেষের টা হচ্ছে আর্গুমেন্ট। কিছু কিছু কমান্ড আর্গুমেন্ট ছাড়াই কাজ করে আবার কিছু কিছু কমান্ড আর্গুমেন্ট ছাড়া কাজ করবেই না। আবার অনেক কমান্ডে আমরা কমান্ডের সাথে অপশন ব্যবহার করে থাকি। একটা কমান্ডের সাথে নানা রকম অপশন ব্যবহার করা যায় এবং একটি একটি অপশন ব্যবহার করে আলাদা আলাদা আউটপুট পাওয়া যায়। নিচের একটা কমান্ডের ০৩ (তিন) টা পার্ট দেখানো হয়েছেঃ

```
# command [options (-)] argument
```

example:

```
-----
```

```
# date
```

```
# cal
```

```
# ping
```

```
# ping 127.0.0.1
# ping -c 4 127.0.0.1
```

```
# ping --help
```

উপরে দেখতে পাচ্ছি এখানে একটি কমান্ডে কমান্ড, অপশন এবং আর্গুমেন্ট সবই ব্যবহার করা হয়েছে। এখানে 'ping' হচ্ছে কমান্ড, '-c4' হচ্ছে অপশন এবং '127.0.0.1' হচ্ছে আর্গুমেন্ট। এই কমান্ডে শুধু 'ping' লিখে এন্টার চাপলে কাজ করবে না। কারণ, 'ping' কোন হোস্ট/নোড কে করবেন, এইজন্য অবশ্যই আর্গুমেন্ট ব্যবহার করতে হবে।

একটা কমান্ডের সাথে অপশন ব্যবহার করে কিভাবে বিভিন্ন আউটপুট পাওয়া যায়, তার একটা উদাহরণ নিচে দেওয়া হয়েছে। এখানে 'ls' কমান্ডের সাথে '-l,-i,-a,-h' নানা অপশন ব্যবহার করে নানা ধরণের আউটপুট পাওয়া যাবে। পাশাপাশি আমরা চাইলে একসাথে সবগুলো অপশন ব্যবহার করতে পারি।

```
[student@node1 ~]$ ls ; বর্তমান ডিরেক্টরির মধ্যে সকল ফাইল/ডিরেক্টরি দেখা যাবে।
[student@node1 ~]$ ls -l ; লং লিস্ট আকারে (Properties) দেখা যাবে।
[student@node1 ~]$ ls -li ; ফাইল/ডিরেক্টরির আইনোড (inode) সহ দেখা যাবে।
[student@node1 ~]$ ls -la ; লুকানো (Hidden) ফাইল সহ দেখা যাবে।
[student@node1 ~]$ ls -lh ; Human Readable (K/M/G) তে দেখা যাবে।
[student@node1 ~]$ ls -laih ; সকল অপশন একসাথে ব্যবহার করা হয়েছে।
```

উপরে 'ls' কমান্ডের সাথে ব্যবহৃত অপশন সমূহের বিস্তারিত নিচে আলাপ করা হয়েছেঃ

-l = list (লম্বা লিস্ট বা বিস্তারিত দেখার জন্য এই অপশন টা ব্যবহার করা হয়)

-i = inode (প্রতিটি ফাইল/ডিরেক্টরির একটা ইন্ডেক্সিং নাম্বার থাকে যেটাকে বলা হয়ে আইনোড (inode), এটা মূলত সব ফাইলের আলাদা আলাদা।)

-a = all (এই অপশন ব্যবহার করলে বর্তমান ডিরেক্টরির সকল লুকায়িত (Hidden) ফাইল/ডিরেক্টরিসহ দেখা যাবে।)

-h = Human Readable (ফাইলের সাইজ সাধারণত বাইট আকারে দেখায়। এই অপশন ব্যবহার করলে Kilo (K), Mega (M), Giga (G) আকারে দেখাবে।)

** আর যদি সবগুলো অপশন একসাথে ব্যবহার করা হয়, তাহলে সবগুলো অপশনের আউটপুট একসাথে দেখাবে।

```
[student@node1 ~]$ pwd ; বর্তমানে কোন ডিরেক্টরির মধ্যে অবস্থান করছি, সেই আউটপুট দেখাবে।
```

"~" => Home dir (root/regular user)
"/" => root partition (/) (Computer)
"/root" => 'root' ইউজারের হোম (~) ডিরেক্টরি।
"/home" => সকল রেগুলার ইউজারদের হোম (~) ডিরেক্টরি।
i.e.: /home/student

[student@node1 ~]\$ => রেগুলার ইউজার 'student' এর home (~) ডিরেক্টরি।

কিছু বেসিক কমান্ড যেগুলো লিনাক্সের সকল ডিস্ট্রিবিউশনের মধ্যে কাজ করবেঃ

[student@node1 ~]\$ cal
[student@node1 ~]\$ cal 2024
[student@node1 ~]\$ date
[student@node1 ~]\$ who ; who are currently loggin (shorter)
[student@node1 ~]\$ w ; who are currently loggin (Details)

[student@node1 ~]\$ tty
[student@node1 ~]\$ hostname

[student@node1 ~]\$ uname -r ; kernel version
[student@node1 ~]\$ uname ; OS name
[student@node1 ~]\$ cat /etc/redhat-release ; redhat/centos version
[student@node1 ~]\$ top ; task manager (বের হওয়ার জন্য 'q' প্রেস করতে হবে)

[student@node1 ~]\$ lastlog ; login details
[student@node1 ~]\$ free -m ; RAM Info
[student@node1 ~]\$ uptime : system UPTiem info
[student@node1 ~]\$ ifconfig ; IP Address Info

[student@node1 ~]\$ history ; list of privious command
[student@node1 ~]\$!25 ; history থেকে '25' নাম্বার কমান্ডটি পুনরায় রান করার জন্য।

[student@node1 ~]\$ history -c ; history থেকে সকল কমান্ড মুছে দেওয়ার জন্য ব্যবহার করা হয়।

[student@node1 ~]\$ history

[student@node1 ~]\$ reboot or 'init 6' ; সিস্টেম রিবুট

[student@node1 ~]\$ poweroff or 'init 0' ; সিস্টেম শাটডাউন

Session :02

Linux Directory Structure:

[student@node1 ~]\$ cd / ; রুট পার্টিশনে (/) প্রবেশ করা হল।

[student@node1 /]\$ ls ; পার্টিশনের (/) মধ্যে কি কি আছে সেটা দেখা হয়েছে।

```
boot  etc  lib  media  opt  root  sbin  sys  usr
bin   dev  home  lib64  mnt  proc  run  srv  tmp  var
```

/bin - Binary Files (যে সকল কমান্ড গুলো রেগুলার ইউজার কর্তৃক রান করা হয়), যেমনঃ ls, cd, pwd, who, cal, date etc.

/boot - সিস্টেম বুট (boot) সম্পর্কিত ফাইল। যেমনঃ boot loader, Kernel, itramfilesystem এখানে থাকবে।

/dev - সিস্টেমের সকল ডিভাইস টাইপের ফাইল গুলো এখানে থাকে। যেমনঃ dvd, HDD, Printer, sound, Terminal, Serial/Parallel Devices.

/etc - সকল সিস্টেম এবং সার্ভার কনফিগারেশন ফাইল। ইউজার/গ্রুপ ডাটাবেজ, সিস্টেম ও সার্ভার কনফিগারেশন, প্রোফাইল, সিকিউরিটি সম্পর্কিত ফাইল।

/home - সকল রেগুলার ইউজারদের হোম ডিরেক্টরি। যেমনঃ student, tamim, sumon (FTP, Email, File সার্ভার ইউজার)।

`/lib` - সিস্টেমের সকল library ফাইল এবং কার্নেল মডিউল থাকে `/lib` এর ভিতরে। `/bin` & `/sbin` ডিরেক্টরিতে যত কমান্ড আছে, এই কমান্ড এক্সিকিউট করতে library দরকার হয়।

`/media` - এটা ডিফল্ট ভাবে ফাঁকা থাকে। মিডিয়া টাইপের ডিভাইস (DVD/ISO) বা এক্সটারনাল ডিভাইস এই লোকেশনে মাউন্ট করে এক্সেস করা হয়।

`/mnt` - mount point (DVD/HDD/USB/Network Share) - ইত্যাদি এই লোকেশন মাউন্ট করে এক্সেস করা যায়।

`/opt` - optional (empty) - এই ডিরেক্টরি সাধারণত ফাঁকা থাকে। বিভিন্ন ফাইল/আপ্লিকেশন/সার্ভিস টেস্ট করার জন্য এই ডিরেক্টরি ব্যবহার করা হয়।

`/proc` - Also called 'proFS' system process related info (CPU, RAM, Process, Running Apps, Driver, Modules and Kernel)

`/root` - লিনাক্সের অ্যাডমিন ইউজার (root) এর হোম ডিরেক্টরি। সাধারণ ইউজাররা (student,tarek,sadia) এখানে প্রবেশ করতে পারে না।

`/run` - service running data. Runtime data for processes started since the last boot.

`/sbin` - System Binary Files (যে সকল কমান্ড গুলো root ইউজার কতুক রান করা হয়), যেমনঃ useradd, groupadd, userdel, fdisk এই কমান্ড সমূহ `/sbin` ডিরেক্টরিতে পাওয়া যাবে।

`/srv` - Sort for Service. User's (`/home/*`) service related data. Like WWW, FTP, Email etc.

`/sys` - Sort for system. `/sys` directory as a virtual filesystem (sysfs) mounted under `/sys`. similar as proc.

`/tmp` - Temporary ফাইল সমূহ এখানে থাকে। সিস্টেম প্রতি ১০ দিন অন্তর অন্তর নিজে থেকেই মুছে (Delete) করে দিবে।

`/usr` - Thirdparty software install location + সিস্টেমের যত কমান্ড থাকে `/usr/bin`, `/usr/sbin` + Library File

/var - variable file (mail/log/hosting/ftpdata/dynamic data)

[student@node1 ~]\$ cd [enter] ; আমরা যে ডিরেক্টরিতেই থাকি না কেন, 'cd' কমান্ড দিলে সরাসরি হোম ডিরেক্টরিতে ফিরে আসা যাবে।

[student@node1 ~]\$ cd / ; 'cd' কমান্ডের সাথে '/' ব্যবহার করে এন্টার প্রেস করলে রুট পার্টিশনে (/) প্রবেশ করা যাবে।

[student@node1 /]\$ ls ; আমরা বর্তমানে রুট পার্টিশনের (/) মাঝে অবস্থান করছি, এখানকার সকল ফাইল/ডিরেক্টরি দেখার জন্য 'ls' কমান্ড দিতে হবে।

[student@node1 /]\$ cd

⌘ cd /dir1/dir2/dir3 ; Linux Directory Path Forward Slash (/) ব্যবহার করা হয়।

⌘ C:\dir1\dir2\dir3> ; Windows Directory Path Back Slash (\) ব্যবহার করা হয়।

[student@node1 ~]\$ cd /var/spool/mail

[student@node1 mail]\$ pwd ; বর্তমানে কোন অবস্থানে আছি, সেটা দেখা যাবে।

/var/spool/mail

[student@node1 mail]\$ ls

[student@node1 mail]\$ cd .. ; এক ধাপ পিছনের (Parent) ডিরেক্টরিতে যাওয়ার কমান্ড।

[student@node1 spool]\$ cd - ; পূর্বের (সর্বশেষ) ডিরেক্টরিতে ফিরে যাওয়ার জন্য।

[student@node1 mail]\$ cd

[student@node1 ~]\$ ls

[student@node1 ~]\$ cd Music

[student@node1 Music]\$ cd ../Videos

[student@node1 Videos]\$ ls

[student@node1 Videos]\$ pwd

[student@node1 Videos]\$ cd

[student@node1 ~]\$ mkdir linux21

[student@node1 ~]\$ ls

[student@node1 ~]\$ cd linux21

[student@node1 linux21]\$ ls

[student@node1 linux21]\$ pwd

```
[student@node1 linux21]$ mkdir day4
[student@node1 linux21]$ cd day4
[student@node1 day4]$ ls
[student@node1 day4]$ pwd
```

Test: Graphically (home/.....)

```
[student@node1 day4]$ touch file1 ; ফাইল তৈরির জন্য 'touch' ব্যবহার করা হয়।
[student@node1 day4]$ ls ; ফাইল দেখার জন্য 'ls' কমান্ড
file1
```

```
[student@node1 day4]$ touch index.html ; এক্সটেনশন সহ ফাইল তৈরির কমান্ড।
[student@node1 day4]$ touch file2 file3 file4 ; একাধিক ফাইল একসঙ্গে তৈরি করার কমান্ড।
[student@node1 day4]$ ls ; ফাইল গুলা দেখার জন্য 'ls' কমান্ড
[student@node1 day4]$ mkdir dir1 dir2 dir3 ; এক কমান্ডের মাধ্যমে একাধিক ডিরেক্টরি তৈরি করার
কমান্ড।
[student@node1 day4]$ ls ; আউপুট দেখার জন্য 'ls' কমান্ড
[student@node1 day4]$ mkdir -p dir1/linux/redhat ; ডিরেক্টরি পাথ তৈরির কমান্ড। (-p = parent)
[student@node1 day4]$ touch dir1/linux/rhel9 ; ডিরেক্টরি পাথে ফাইল তৈরি কমান্ড।
[student@node1 day4]$ cd dir1/linux/ ; ডিরেক্টরি পাথে প্রবেশ করার জন্য।
[student@node1 linux]$ ls ; বর্তমান ডিরেক্টরির মধ্যে কি আছে সেটা দেখার জন্য।
centos redhat ; এখানে 'centos' ফাইল এবং 'redhat'
[student@node1 linux]$ pwd ; আমরা কোথায় অবস্থান করছি সেটা দেখার কমান্ড।
/home/student/linux21/day4/dir1/linux
```

```
[student@node1 linux]$ cd /home/student/linux21/day4 ; রুট পার্টিশন (/) হয়ে 'day4' ডিরেক্টরিতে
প্রবেশ করার জন্য।
```

or

```
[student@node1 linux]$ cd ../.. ; দুই ধাপ পিছনে (Parent) ডিরেক্টরিতে যাওয়ার কমান্ড।
[student@node1 linux]$ ls
```

Session :03

Working with File/Directory Properties:

যখন উইন্ডোজ অপারেটিং সিস্টেমে কোনো ফাইল/ডিরেক্টরি সম্পর্কে বিস্তারিত জানার দরকার হয়, তখন ডিরেক্টরির উপরে মাউচের ডান বাটন প্রেস করে 'Properties' থেকে বিস্তারিত তথ্য পাওয়া যায়। যেমনঃ সিকিউরিটি, সাইজ, টাইপ, মোডিফাই/আপডেট তারিখ, ভার্শন ইত্যাদি।

```
[student@node1 day4]$ touch file1 ; 'touch' দিয়ে একটি ফাইল তৈরি করা হয়েছে।
```

```
[student@node1 day4]$ mkdir dir1 ; 'mkdir' দিয়ে একটি ডিরেক্টরি তৈরি করা হয়েছে।
```

```
[student@node1 day4]$ ll ; লিনাক্সে 'll' কমান্ড ব্যবহার করে ফাইল/ডিরেক্টরির বিস্তারিত তথ্য (Properties) দেখা যাবে।
```

```
d  rwx rwx r-x .  2  student student  6   Feb  4 18:06  dir1
-  rw- rw- r-- .  1  student student  0   Feb  4 18:06  file1
```

```
(1) 2a 2b 2c 2d 3 4 5 6 7 8
```

1: file/dir types (লিনাক্সে মোট সাত ধরনের (-, d, l, c, b, s, p) ফাইল/ডিরেক্টরি পাওয়া যাবে)

2: file/dir পার্মিশন ফিল্ড এবং এখানে ডট (.) সহ মোট ১০টি ক্যারেক্টার আছে। এই দশটি ক্যারেক্টার কে আবার ৪ টি আলাদা ফিল্ডে ভাগ করা হয়েছেঃ

2a: user permission,

2b: group permission,

2c: others permission

2d: ACL Permission (.)

3: file/dir link (Hard Link) - ফাইলের ক্ষেত্রে '1' আর ডিরেক্টরির ক্ষেত্রে '2' থাকবে, তবে ভিতরে যত ডিরেক্টরি থাকবে তার উপরে ভিত্তি করে সংখ্যা বাড়বে।

4: file/dir owner (যে ইউজার ফাইল/ডিরেক্টরিটি তৈরি করেছে তার নাম থাকবে)

5: file/dir group owner - (ইউজার যে গ্রুপের সদস্য সেই গ্রুপের নাম থাকবে)

6: file/dir size (byte) - ফাইল হলে '0' বাইট এবং ডিরেক্টরি হলে 6 বাইট (XFS) ফাইল সিস্টেমে।

- 7: file/dir modify date - তৈরির বা আপডেটের তারিখ এবং সময়।
8: file/dir name - ফাইল বা ডিরেক্টরির নাম।

File/dir Permission:

- r = read
- w = write
- x = execute
- = no permission
- . = ACL File Permission (+)

Linux File & Directory types:

- = Regular file : text/any file
- d = Directory : Regular directory (ফোল্ডার)
- l = Link file : \$ ls -l /dev/stdin
- b = Device CD/DVD/HDD : \$ ls -l /dev/sda
- c = Character device : \$ ls -l /dev/tty
- s = Socket : \$ ls -l /run/rpcbind.sock
- p = Pipe file : \$ ls -l /run/initctl

b - Block Devices (CD/DVD/ISO/HDD/USB)

c - printer, sound card, virtual terminal, serial port, parallel port

s - A socket file is used to pass information between applications/process for communication purpose

l - link file (softlink/symbolik link)

P - Pipes are unnamed objects created to allow two processes to communicate. One process reads and the other process writes to the pipe file. This unique type of file is also called a first-in-first-out (FIFO) file.

```
[student@node1 day4]$ cd /dev
```

```
[student@node1 dev]$ ll
```

```
[student@node1 dev]$ cd -
```

```
[student@node1 day4]$ cd /run
```

```
[student@node1 run]$ ll
[student@node1 run]$ cd -
[student@node1 day4]$
```

Copy/Paste/Remove/Rename/Delete

[student@node1 day4]\$ ls ; 'ls' কমান্ড দিয়ে দেখা যাচ্ছে 'file1' এবং 'dir1' নামে একটি করে ফাইল এবং ডিরেক্টরি আছে।

```
file1 dir1
```

[student@node1 day4]\$ cp file1 file2 ; ফাইল কপি করার জন্য 'cp' কমান্ড ব্যবহার করে 'file1' টি কপি করে 'file2' নামে আরেকটি করা হয়েছে।

```
[student@node1 day4]$ ls
```

[student@node1 day4]\$ cp file1 /home/student ; ফাইল কপি করে অন্য লোকেশনে '/home/student' একই নামে।

[student@node1 day4]\$ cp file1 /home/student/file3 ; ফাইল কপি করে অন্য লোকেশনে '/home/student' অন্য নামে।

[student@node1 day4]\$ cd /home/student ; '/home/student' লোকেশনে ঢুকে দেখা হয়েছে কপি হয়েছে কিনা।

```
[student@node1 ~]$ ls
```

```
file1
```

```
file3
```

```
[student@node1 ~]$ cd -
```

[student@node1 day4]\$ cp /etc/passwd . ; অন্য লোকেশন থেকে ফাইল কপি করে বর্তমান ডিরেক্টরির (.) মধ্যে নিয়ে আসার কমান্ড।

[student@node1 day4]\$ ls ; কপি হয়ে আসছে কিনা সেটা দেখার জন্য।

[student@node1 day4]\$ cp /etc/hostname /home/student ; বর্তমান ডিরেক্টরিতে থেকে অন্য ডিরেক্টরির ফাইল কপি করে অন্য লোকেশনে পাঠানোর কমান্ড।

```
[student@node1 day4]$ cd /home/student
```

```
[student@node1 ~]$ ls
```

```
[student@node1 ~]$ cd - ; পূর্বের ডিরেক্টরিতে ফিরে যাওয়ার কমান্ড।
```

[student@node1 day4]\$ cp dir1 dir99 ; wrong command

[student@node1 day4]\$ cp -r dir1 dir99 ; ডিরেক্টরি কপি করার জন্য অবশ্যই 'cp -r' ব্যবহার করতে হবে। (-r = recursively)

[student@node1 day4]\$ cp -r /etc . ; '/etc' ফুল ডিরেক্টরি কপি করে বর্তমান ডিরেক্টরির (.) মধ্যে নিয়ে আসার কমান্ড।

[student@node1 day4]\$ ls

[student@node1 day4]\$ mv file2 file4 ; 'mv' কমান্ড দিয়ে ফাইল 'file2' কে 'file4' নামে পরিবর্তন করা হয়েছে।

[student@node1 day4]\$ ls ; এখানে দেখা যাচ্ছে 'file2' নাম পরিবর্তন করে 'file4' নামে হয়েছে।

[student@node1 day4]\$ mv file4 /home/student ; ফাইল মুভ (cut) করে অন্য লোকেশনে নিয়ে যাওয়া।

[student@node1 day4]\$ ls ; বর্তমান ডিরেক্টরির মধ্যে 'file4' নামে কিছু পাওয়া যাবে না। কারণ এটা '/home/student' মধ্যে পাঠানো হয়েছে।

[student@node1 day4]\$ rm file1 ; ফাইল মুছে ফেলার জন্য 'rm' কমান্ড।

[student@node1 day4]\$ ls ; ফাইল টি নাই (deleted)

[student@node1 day4]\$ rm dir99 ; wrong command (শুধু 'rm' কমান্ড দিয়ে ডিরেক্টরি রিমুভ করা যাবে না)।

[student@node1 day4]\$ rm -r dir99 ; ডিরেক্টরি রিমুভ করার জন্য অবশ্যই '-r' ব্যবহার করতে হবে। (-r = recursively)

[student@node1 day4]\$ rm -r etc ; সম্পূর্ণ ডিরেক্টরি রিমুভ করার জন্য প্রতিবার 'y' অপশন প্রেস করে কনফার্ম করতে হবে।

C^

'Ctrl + C' প্রেস করে, ফাইল রিমুভ বিরত রাখতে পারি।

[student@node1 day4]\$ rm -rf etc ; প্রতিবার 'y' প্রেস করার পরিবর্তে '-rf' ব্যবহার করলে, একসঙ্গে সকল ফাইল/ডিরেক্টরি মুছবে।

(-f = forcefully, r=recursively)

[student@node1 day4]\$ ls ; 'etc' ডিরেক্টরি আছে কিনা দেখার জন্য।
[student@node1 day4]\$ rm -rf * ; বর্তমান ডিরেক্টরির সকল ফাইল/ডিরেক্টরি কনটেন্ট
রিমুভ করার জন্য (*) ব্যবহার করা হয়।
[student@node1 day4]\$ ls ; কিছুই পাওয়া যাবে না।

note: rm -rf * .* লুকানো ফাইল (Hidden) সহ ডিলিট করার কমান্ড।

- The End -

Class -05

Text Processing tools : (echo/cat/less/head/tail/Grep/wc/locate/find/man/help)

প্রথমে আমাদের Day05 এর working ডিরেক্টরি তৈরি করে নিতে হবে। এজন্য student ইউজার দিয়ে লগইন করে 'cd' কমান্ড দিয়ে 'home' ডিরেক্টরিতে প্রবেশ করতে হবে। 'Home' ডিরেক্টরিতে প্রবেশ করার পরে, 'mkdir -p' দিয়ে 'linux21/day5' নামে একটা ডিরেক্টরি পাথ তৈরি করতে হবে এবং 'cd' কমান্ড দিয়ে 'linux21/day5' -এ প্রবেশ করতে হবে। আজকের ল্যাব প্র্যাকটিস করার জন্য 'cp' কমান্ড দিয়ে '/etc/passwd' ফাইলটি কপি করে বর্তমান ডিরেক্টরির মধ্যে নিয়ে আসা হবে, পাশাপাশি 'touch' কমান্ড দিয়ে 'file1' নামে নতুন একটি ফাইল তৈরি হবে। পরিশেষে 'ls' কমান্ড দিয়ে সব কিছু ঠিকঠাক আছে কিনা দেখতে করতে হবে।

```
[student@node1~]$ cd
[student@node1~]$ mkdir -p linux21/day5
[student@node1~]$ cd linux21/day5
[student@node1 day5]$ ls
[student@node1 day5]$ cp /etc/passwd . ; অথবা #cp /etc/passwd
/home/student/linux21/day5
[student@node1 day5]$ ls
passwd
[student@node1 day5]$ touch file1
[student@node1 day5]$ ls
passwd file1
```

Working 'cat' command :

[student@node1 day5]\$ cat file1 ; ফাইলের কনটেন্ট দেখার জন্য 'cat' কমান্ড ব্যবহার করা হয়।
যেহেতু 'file1' নতুন ফাইল সুতরাং এটার ভিতরে কিছু দেখাবে না।

[student@node1 day5]\$ cat passwd ; 'passwd' ফাইলের সম্পূর্ণ কনটেন্ট দেখা যাবে।

[student@node1 day5]\$ cat -n passwd ; লাইন নাম্বার সহ ফাইল দেখার জন্য '-n' অপশন
ব্যবহার করতে হবে।

[student@node1 day5]\$ cat /etc/hostname ; 'cat' কমান্ড ব্যবহার করে অন্য কোনো ডিরেক্টরির ভিতরের ফাইল দেখার জন্য।

[student@node1 day5]\$ cat /etc/redhat-release ; RHEL কোন ভার্সন ব্যবহার করা হচ্ছে, সেটা জানার জন্য।

Working 'echo' and 'variables'

'echo' কমান্ড ব্যবহার করে টার্মিনালে যে কোনো টেক্সট প্রিন্ট করা যাবে। পাশাপাশি 'echo' কমান্ড ব্যবহার করে বিভিন্ন ধরনের সিস্টেম ভেরিয়েবল, ইউজার ডিফাইন ভেরিয়েবল এবং চাইলে সিস্টেমের যেকোনো কমান্ডকেও কল করা যাবে।

[student@node1 day5]\$ clear ; স্ক্রিন ক্লিয়ার করার কমান্ড।
[student@node1 day5]\$ Hello World ; আমরা যদি লিনাক্স টার্মিনালে 'Hello World' লিখে এন্টার প্রেস করলে 'Bash Error' আসবে।

[student@node1 day5]\$ echo "Hello World" ; কিন্তু আমরা যদি 'Hello World' কে 'echo' কমান্ডের মাধ্যমে কল করি, তাহলে সেটা দেখা যাবে।

[student@node1 day5]\$ echo \$SHELL ; লিনাক্স টার্মিনালে ব্যবহৃত শেল/ইন্টারপ্রিটার সম্পর্কে জানার জন্য। এখানে 'SHELL' বড় হাতের অক্ষরে হবে।

[student@node1 day5]\$ echo \$HOSTNAME ; লিনাক্স সিস্টেমে বিল্ট-ইন Environment Variable বড় হাতের (Block Letter) অক্ষরে প্রকাশ করা হয়।

[student@node1 day5]\$ a=100 ; user defined variable
[student@node1 day5]\$ echo \$a
[student@node1 day5]\$ os=linux
[student@node1 day5]\$ echo \$os
[student@node1 day5]\$ date
[student@node1 day5]\$ echo "Today is: \$(date)" ; all bash command

Working Input and Output Redirection (>, >>):

লিনাক্স কমান্ড লাইন ইন্টারফেসে তিন ধরনের ডেটা স্ট্রিম তৈরি ইনপুট/আউট

⌘ stdin - standard input (Value 0) - এক্ষেত্রে টেক্সট ইনপুট হিসেবে গ্রহণ করে। stdin ডিভাইস হচ্ছে - Keyboard

⌘ stdout - standard output (Value 1) - Stdout ডিভাইস হচ্ছে মনিটর

⌘ stderr - standard error (Value 2) - Stderr ডিভাইস error আউপুট জমা রাখে। কমান্ডে যখন কোনো এরর আসে, তখন এরর গুলা জমা করা হয়।

Output Redirection:

```
[student@node1 day5]$ echo "Welcome to RHEL 9"
Welcome to RHEL 9
[student@node1 day5]$ echo "Welcome to RHEL 9" > file1
[student@node1 day5]$ cat file1
[student@node1 day5]$ echo "Welcome to CentOS 9" > file1 ; Overwrite করবে।
[student@node1 day5]$ cat file1
[student@node1 day5]$ echo "Welcome to RHEL 9" >> file1 ; append
[student@node1 day5]$ cat file1
[student@node1 day5]$ history
[student@node1 day5]$ history > shell_history
[student@node1 day5]$ cat shell_history
[student@node1 day5]$ ping -c4 127.0.0.1 > ping
[student@node1 day5]$ ls
[student@node1 day5]$ cat ping
```

Working 'head, tail & less' command :

[student@node1 day5]\$ less passwd ; ফাইল উপর থেকে নিচ পর্যন্ত স্ক্রলিং করে দেখা যাবে। (Press 'q' for quit)

[student@node1 day5]\$ head passwd ; ফাইলের প্রথম ১০ লাইন দেখার জন্য।

[student@node1 day5]\$ tail passwd ; ফাইলের শেষের ১০ লাইন দেখার জন্য।

[student@node1 day5]\$ tail -15 passwd ; ফাইলের শেষের ১৫ লাইন দেখার জন্য।

[student@node1 day5]\$ head -5 passwd ; ফাইলের প্রথম ৫ লাইন দেখার জন্য।

Pipelines :

'|' (pipeline) = Pipeline মূলত ব্যবহার হয়, এক কমান্ডের আউটপুট ফিল্টার করে আরেকটি আউটপুট পাওয়ার জন্য। কমান্ডের সাথে একই সময়ে একাধিক বার '|' (Pipeline) ব্যবহার করা যাবে।

[student@node1 day5]\$ cat -n passwd | head

[student@node1 day5]\$ cat -n passwd | tail

[student@node1 day5]\$ cat -n passwd | less ; press 'q' for quit

Working with 'GREP' :

'grep' এর কাজ হচ্ছে ফাইলের ভিতর থেকে বা কোনো কমান্ডের আউটপুট থেকে কীওয়ার্ড খুঁজে বের করা। 'grep' ব্যবহার করে ফাইলের মধ্যে, লাইনের শুরুতে, লাইনের শেষে Keyword খুঁজতে পারি। 'grep' কে অন্য কমান্ড 'cat', 'tail', 'head', less এর সাথেও ব্যবহার করা যাবে।

[student@node1 day5]\$ grep -n root passwd

```
1 root:x:0:0:root:/root:/bin/bash
10 operator:x:11:0:operator:/root:/sbin/nologin
```

নোটঃ আউটপুটে দেখা যাচ্ছে 'root' কীওয়ার্ড আছে এবং '-n' অপশন ব্যবহার করা হয়েছে লাইন নম্বার দেখার জন্য। এখানে ১ এবং ১০ নম্বার লাইনে 'root' আছে।

```
[student@node1 day5]$ grep FTP passwd
```

```
[student@node1 day5]$ grep "FTP User" passwd ;একাধিক কীওয়ার্ড (String) একসাথে খুঁজতে চাইলে কোট ("") ব্যবহার করতে হবে।
```

```
[student@node1 day5]$ grep -i ftp passwd
```

```
[student@node1 day5]$ tail passwd | grep root ; আমরা চাইলে 'tail' কমান্ডের সাথেও 'grep' ব্যবহার করতে পারি।
```

```
[student@node1 day5]$ head passwd | grep root ; আমরা চাইলে 'head' কমান্ডের সাথেও 'grep' ব্যবহার করতে পারি।
```

```
[student@node1 day5]$ cat passwd | grep bash ; আমরা চাইলে 'cat' কমান্ডের সাথেও 'grep' ব্যবহার করতে পারি।
```

Working with 'wc':

কোনো ফাইলের মধ্যে কতগুলো লাইন, ওয়ার্ড, ক্যারেক্টার (Letter) আছে সেটা দেখাবে। কোনো ডিরেক্টরির মধ্যে কতগুলো ফাইল/ডিরেক্টরি আছে সেই লিস্টও পাওয়া যাবেঃ

```
[student@node1 day5]$ wc passwd
[student@node1 day5]$ ls | wc -l
[student@node1 day5]$ ls /etc | wc -l
```

Working with Linux 'locate command':

locate কমান্ডের মাধ্যমে লিনাক্স সিস্টেমের মধ্যে যে কোনো ফাইল/ডিরেক্টরি খুঁজে বের করা যাবে। locate কমান্ডগুলো প্রয়োগ করার জন্য প্রথমে 'root' ইউজারের প্রিভিলেজ নিতে হবে। এক্ষেত্রে নিচের 'su' (switch user) কমান্ড ব্যবহার করতে হবেঃ

```
[student@node1 day5]$ su
: *****
```

নোটঃ উপরে 'root' ইউজারের পাসওয়ার্ড দিতে হবে।

```
[root@node1 day5]# updatedb ; লিনাক্স সকল ফাইল/ডিরেক্টরি আপডেট (Cache) করার কমান্ড।
```

```
[root@node1 day5]# locate sshd_config ; যে কোন লোকেশনে 'sshd_config' ফাইলে খুঁজে বের করার জন্য।
```

```
[root@node1 day5]# locate -i .jpg ; (i=insensitive) word সাথে extension সহ খুঁজে বের করার জন্য।
```

```
[root@node1 day5]# locate Dhaka
```

Working with Linux 'find command' :

```
[root@node1 day5]# find / -name mail ; mail নামে যে কোনো ফাইল/ফোল্ডার রুট (/) ফাইল সিস্টেমে খোঁজার জন্য।
```

```
[root@node1 day5]# find /var -name mail ; নির্দিষ্ট লোকেশনে '/var' মধ্যে 'mail' নামে কিছু খোঁজার জন্য।
```

```
[root@node1 day5]# find / -user student ; 'student' নামে Owner নামে যত ফাইল আছে, সব খুঁজে বের করার জন্য।
```

```
[root@node1 day5]# find / -type d -name ssh ; 'ssh' নামে ডিরেক্টরি খুঁজে বের করার জন্য।
```

```
[root@node1 day5]# find / -type f -name ssh ; 'ssh' নামে ফাইল খুঁজে বের করার জন্য।
```

```
[root@node1 day5]# find / -type b ; সিস্টেমের সকল ডিভাইস ফাইল (b) খুঁজে বের করার কমান্ড।  
[root@node1 day5]# find / -size +100M ; 100 MB এর উপরের সাইজের সকল ফাইল খুঁজে  
বের করার জন্য।
```

```
[root@node1 day5]# ls -sh /usr/lib64/firefox/libxul.so
```

Working with "Help and Manual" :

```
[root@node1 day5]# man  
[root@node1 day5]# man ping ; ping সম্পর্কিত সকল কমান্ডের তথ্য পাওয়া যাবে। বের হওয়ার  
জন্য 'q' প্রেস করতে হবে।
```

```
[root@node1 day5]# ping --help  
[root@node1 day5]# ping -?
```

```
[root@node1 day5]# mandb
```

```
[root@node1 day5]# whatis sshd_config ; সিস্টেমের নির্দিষ্ট একটি ফাইল সম্পর্কে জানার জন্য।  
[root@node1 day5]# whatis ping ; নির্দিষ্ট কমান্ড 'ping' সম্পর্কে জানার জন্য।  
[root@node1 day5]# whatis find ; নির্দিষ্ট কমান্ড 'find' সম্পর্কে জানার জন্য।
```

The Info Command :

```
[root@node1 day5]# info ping ; স্পেসবার ব্যবহার করে সামনের দিকে যাওয়া যাবে। বের হওয়ার জন্য  
'q' প্রেস করতে হবে।
```

Online Documentation :

docs.redhat.com -> RedHat Enterprise Linux -> System Administration -> PDF

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9

Home work :

** Find out and store number of log messages generated by your system on "July 24, 2023".

```
[student@node1 day5]$ su (switch to root)
,*****
```

Log location Hints:

```
[root@node1 day5]# less /var/log/messages
```

»» —Thank you — ««

Class -06

Linux Text Editor :

Windows Text Editor: notepad, notepad++, sublimetext, visual studio code, atom

Linux Text Editor: vi/vim, nano, gedit (GUI), emacs, pico

Most popular/Advanced Text Editor: Vi/Vim

=> vi - old, b&w, default (Default Editor for Minimal Install Method)

=> vim - advanced, GUI (Mode Only), Colorfull

[student@node1 ~]\$ cd /home ; 'cd' কমান্ড দিয়ে 'home' ডিরেক্টরিতে প্রবেশ করা হয়েছে।

[student@node1 ~]\$ mkdir -p linux21/day6 ; এখানে 'linux21' এর মধ্যে 'day6' নামে আরেকটি সাব-ডিরেক্টরি তৈরি করা হয়েছে।

[student@node1 ~]\$ cd linux21/day6 ; /home/student/linux21/day6 ডিরেক্টরের মাঝে প্রবেশ করা হল।

[student@node1 day6]\$ ls ; এই মুহূর্তে এখানে কোনও ফাইল/ডিরেক্টরি থাকার কথা না।

[student@node1 day6]\$ touch test1 ; 'test1' নামে নতুন একটি ফাইল তৈরি করা হল।

[student@node1 day6]\$ cp /etc/passwd . ; '/etc/passwd' ফাইলটি কপি করে বর্তমান ডিরেক্টরের মধ্যে নিয়ে আসা হয়েছে।

[student@node1 day6]\$ ls ; 'ls' কমান্ড দিয়ে দেখা হয়েছে ফাইল দুইটি ঠিকমত আছে কিনা।

passwd test1

Working with Basic 'vi/vim' editor:

```
[student@node1 day6]$ vim file1
```

; এটা নতুন ফাইল (New File) হিসেবে খুলবে। এখন কীবোর্ড থেকে 'i' বা 'insert' বাটন প্রেস করে লেখা যাবে। আমরা এখানে "welcome to linux vim editor" লাইনটি লিখবো এবং লেখার পরে কীবোর্ড থেকে 'Esc' বাটন প্রেস করবো।

```
:x
```

Note: লেখার জন্য "i" প্রেস করতে হবে। "i" প্রেস করলে নিচে দেখা যাবে, আমরা 'INSERT' মোডে আছি। 'INSERT' মোড ছাড়া মোটেও লেখা যাবে না। আবার যখন ফাইল থেকে বের হয়ে সেভ করার প্রয়োজন হবে, তখন 'INSERT' মোড থেকে 'Esc' বাটন প্রেস করতে হবে। 'Esc' বাটন প্রেস করার পরে 'INSERT' মোড থেকে বের হয়ে ফাইল সেভ করতেও পারি, আবার নাও করতে পারি। সেভ করার জন্য ':x' প্রেস করতে হবে, আর সেভ না করতে চাইলে, ':q!' প্রেস করতে হবে।

```
[student@node1 day6]$ ls
```

```
[student@node1 day6]$ cat file1
```

Working with Basic 'nano' editor:

```
[student@node1 day6]$ nano file2 ; 'nano' কমান্ড ব্যবহার করে সরাসরি ফাইলে লেখা যাবে।  
welcome to linux nano editor
```

এখানে 'vim/vi' এর মত লেখার জন্য কোনো বাটন প্রেস করতে হবে না। আমরা চাইলে সরাসরি লিখতে পারি। লেখার পরে, ফাইলের নিচের ইন্সট্রাকশন অনুযায়ী

'Ctrl+X' প্রেস (Exit) করতে হবে। তখন সেভ করা হবে কিনা (Y/N) এমন প্রশ্ন করবে, আমরা সেভ করতে চাইলে 'Y' প্রেস করে 'Enter' দিলে 'file2' নামে সেভ হবে।

```
[student@node1 day6]$ ls
```

```
[student@node1 day6]$ cat file2 ; 'nano' এডিটর দিয়ে ফাইলে লেখার পরে দেখার জন্য।
```

Working with Basic 'gedit' editor :

'gedit' ব্যবহার করতে হলে অবশ্যই গ্রাফিক্যাল ইউজার ইন্টারফেস (GUI) ব্যবহার করতে হবে। 'gedit' ব্যবহার করার সময় আমরা কীবোর্ড ছাড়াও মাউচ ব্যবহার করে ফাইল সেভ, ফাইলের কন্টেন্ট কপি (copy), পেস্ট (Paste), কাট (cut), মুছতে (delete) করতে পারি।

```
[student@node1 day6]$ gedit file3
```

```
welcome to graphical geditor
```

নোটঃ ফাইল সেভ করার জন্য 'Ctrl + s' প্রেস করে অথবা মাউচের বাউট উপরের ডান সাইডে 'save' বাটনে ক্লিক করে সেভ করা যাবে।

```
[student@node1 day6]$ cat file3
```

Working with vi/vim Mode :

⌘ Insert Mode : শুধুমাত্র লেখার জন্য। Insert মোডে ঢোকানোর জন্য "i" অথবা "insert" কী প্রেস করতে হয়।

⌘ Command Mode: ফাইলের মাঝে কার্সর নেভিগেশন, cut, copy, paste, undo, redo করতে ব্যবহার করা হয়।

⌘ Visual Mode: কীবোর্ড থেকে "ctrl + v" বা 'v' প্রেস করে 'VISUAL BLOCK' এর মাধ্যমে নির্দিষ্ট অংশ সিলেক্ট করতে পারেন এবং সেই অংশ ডিলিট,কপি (y) করতে পারেন। যেখানে নিয়ে পেস্ট করবেন সেখানে গিয়ে 'p' প্রেস করতে হবে।

⌘Exit Mode : যখন ফাইল হতে বের হওয়ার জন্য ':x' বা ':wq' প্রেস করি তখন সেটকে Exit মোড বলা হয়।

Working with 'vi/vim' command modes :

```
[student@node1 day6]$ ls
```

```
[student@node1 day6]$ vim file1 ; 'file1' নামের ফাইলটি পুনরায় ওপেন করা হয়েছে।
```

⌘ insert মোডে প্রবেশের জন্য কীবোর্ড থেকে "i" প্রেস করা হবে।

⌘ "welcome to linux training" এটা আমরা দ্বিতীয় লাইনে লিখবো।

⌘ লেখার পরে 'insert' মোড থেকে বের হওয়ার জন্য 'Esc' প্রেস করতে হবে।

⌘ শুধু সেভ করার জন্য ':w' লিখে এন্টার প্রেস করতে হবে, এক্ষেত্রে ফাইল থেকে বের হবে না, কিন্তু ফাইল সেভ হয়ে যাবে।

নোটঃ ':w' (write) প্রেস করার ফলে, আমাদের ২ টি লাইন সেভ হয়েছে।

⌘ পুনরায় 'insert' মোডে প্রবেশ করে তৃতীয় লাইনে 'goodbye' লেখার পরে, যদি তৃতীয় লাইন সেভ করতে না চাই, তাহলে পুনরায় 'Esc' বাটন প্রেস করে ':q!' লিখে এন্টার প্রেস করতে হবে।

নোটঃ এখানে ':q!' (quit) প্রেস করার পরে এডিটর থেকে সেভ ব্যাতিত বের হয়ে যাবে। এখানে (!) ব্যবহার করে Forcely quit করা হয়েছে।

```
[student@node1 day6]$ cat file1
```

```
welcome to linux vim editor
```

```
welcome to linux training
```

```
[student@node1 day6]$ cat passwd
```

```
[student@node1 day6]$ vim passwd ; 'passwd' ফাইল নামে যে ফাইলটি আমরা কপি করেছিলাম সেটা ওপেন করা হয়েছে।
```

```
:set nu
```

'esc' মোডে কীবোর্ড থেকে ':set nu' টাইপ করে এন্টার প্রেস করলে বাম পাশ দিয়ে ফাইলের লাইন নম্বার দেখাবে।

```
1 root:x:0:0:root:/root:/bin/bash
```

2 bin:x:1:1:bin:/bin:/sbin/nologin

3 daemon:x:2:2:daemon:/sbin:/sbin/nologin

Cursor Movement :

h j k l - বামে (h), নিচে (j), উপরে (k), ডানে (l)

^ - লাইনের শুরুতে যাওয়ার জন্য (Shift + 6) প্রেস করতে হবে।

\$ - লাইনের শেষে যাওয়ার জন্য (Shift + 4) প্রেস করতে হবে।

gg - ফাইলের প্রথম লাইনে যাওয়ার জন্য।

G - ফাইলের শেষের লাইনে যাওয়ার জন্য।

:n - ফাইলের নির্দিষ্ট লাইনে যাওয়ার জন্য। যদি ২৫ নাম্বার লাইনে যেতে চাই, তাহলে ':25' প্রেস করতে হবে।

Replacing Text :

r - কোনো অক্ষর (letter/character) পরিবর্তন করতে চাইলে, প্রথমে 'r' প্রেস করতে হবে, তার পরে কীবোর্ড থেকে যে অক্ষর (letter/character) প্রেস করা হবে, সেই অক্ষর (letter/character) দিয়ে পরিবর্তন (replace) হবে।

Copy/Paste :

yw - word কপি করার জন্য। যে word টি কপি করতে চাই, সেখানে কার্সর রেখে 'yw' প্রেস করলে কপি হবে। এর পরে যেখানে নিয়ে Paste করতে চাই, সেখানে গিয়ে 'P/p' প্রেস করতে হবে।

yy - Yank line (single Line) একটি লাইন কপি করতে চাইলে 'yy' প্রেস করতে হবে। যেখানে Paste করা হবে, সেখানে কার্সর রেখে 'p' (Paste) করতে হবে। লাইনের নিচে 'Paste' করার জন্য 'p' এবং লাইনের উপরে 'Paste' করার জন্য 'P' প্রেস করতে হবে।

nyy - কয়েক লাইন কপি (yank) করার জন্য। আমরা যদি ০৫ লাইন কপি করতে চাই, তাহলে '5yy' প্রেস করে কপি (yank) করতে পারি এবং যেখানে Paste করা হবে, সেখানে গিয়ে 'P/p' প্রেস করতে হবে।

p - লাইনের নিচে 'Paste' করার জন্য।

P - লাইনের উপরে 'Paste' করার জন্য।

Inserting Text :

- i - কমান্ড মোড থেকে Insert মোডে প্রবেশ করার জন্য 'i' বা 'Insert' বাটন প্রেস করতে হয়।
- I - লাইনের শেষে থেকে 'I' প্রেস করলে, লাইনের শুরুতে গিয়ে 'Insert' মোডে ঢোকা যাবে।
- a - কার্সরের পরে 'Insert' মোডে ঢোকানোর জন্য 'a' প্রেস করতে হবে।
- A - 'A' প্রেস করলে কার্সর লাইনের শেষে গিয়ে হতে 'Insert' মোডে প্রবেশ করা যাবে।
- o - 'o' প্রেস করলে বর্তমান লাইনের নিচে 'Insert' মোড আসবে বা লেখা যাবে।
- O - 'O' প্রেস করলে বর্তমান লাইনের উপরে 'Insert' মোড আসবে বা লেখা যাবে।

Delete/Cut Text :

- x - একটি করে অক্ষর (Character) কার্সরের সামনে থেকে ডিলিট (cut) হবে।
- dd - একটি লাইন ডিলিট বা কাট (Cut) করার জন্য।
- n dd - কয়েক লাইন (n সংখ্যক) ডিলিট করার জন্য।
- dw - delete (cut) word
- D - কার্সর যেখানে আছে (লাইনের মাঝে) সেখান থেকে লাইনের শেষ পর্যন্ত delete/cut করার জন্য।

Undo/Redo :

- u - Undo (পূর্বের অবস্থায় ফিরে যাওয়া যাবে)
- Ctrl+r - Redo

Searching :

- /pattern - Forward search for {pattern} উদাহরণ স্বরূপ '/var' তার পরে 'n' প্রেস করতে থাকুন।
- ?pattern - Reverse search for {pattern}
- n - পরবর্তী প্যাটার্নে জাম্প করবে।
- N - পূর্বের প্যাটার্নে জাম্প করবে।

Find and Replace :

- :%s/old_pattern/new_pattern/gc - 'gc' প্রেস করলে প্রতিটি প্যাটার্ন পরিবর্তন হওয়ার আগে কনফার্মেশন চাবে।
- :%s/old_pattern/new_pattern/g - 'g' প্রেস করলে সব প্যাটার্ন গুলা একসাথে পরিবর্তন হবে।

উদাহরণঃ :%s/root/admin/g

Module Summary :

:w - Write (শুধুমাত্র সেভ হবে।)

:wq or :x - সেভ হবে সেই সাথে ফাইল থেকে বের হয়ে যাবে।

:q - কোনো একটি ফাইলে ভুলক্রমে ঢুকে পড়েছেন, কিন্তু কোনো টেক্সট টাইপ করা হয়নি, এক্ষেত্রে বের হওয়ার জন্য 'q' প্রেস করলেই হবে।

:q! - কোনো একটি ফাইলে ঢুকে কিছু একটা লিখেছি, এক্ষেত্রে সেভ ছাড়া (Forcely) বের হওয়ার জন্য 'q!' প্রেস করলেই হবে।

:wq! - Forcely ফাইলে সেভ এবং বের হওয়ার জন্য।

\$ vimtutor - vi/vim সম্পর্কে আরোও বিস্তারিত জানার জন্য।

//////////////////// Thank you //////////////////////

Class -07 (Day 07) User and Group Administration

Module Objectives

- ⌘ লিনাক্স সিস্টেমে বিভিন্ন ধরনের ইউজার এবং গ্রুপ সম্পর্কে বিস্তারিত আলোচনা।
- ⌘ লিনাক্স সিস্টেমের ইউজার প্রোফাইল এবং গ্রুপ প্রোফাইল নিয়ে বিস্তারিত আলোচনা।
- ⌘ লিনাক্স সিস্টেমের ইউজার এবং গ্রুপ ম্যানেজমেন্টের উপরে বিস্তারিত আলোচনা।
- ⌘ লিনাক্সে 'Superuser' এবং 'SUDO' ইউজারের প্রিভিলেজ সম্পর্কে বিস্তারিত আলোচনা।
- ⌘ লিনাক্স সিস্টেমে ইউজারের পাসওয়ার্ড প্রোফাইল এবং পাসওয়ার্ড পলিসি নিয়ে বিস্তারিত আলোচনা।

আমাদের পূর্বের ক্লাস গুলোতে রেগুলার ইউজার 'student' দিয়ে লগইন করে সকল কাজ করেছি। আজ থেকে পরবর্তী সকল ক্লাস গুলো 'root' ইউজার দিয়ে লগইন করে কাজ করতে হবে। এক্ষেত্রে RHEL/CentOS অপারেটিং সিস্টেমের লগইন স্ক্রিন থেকে 'Not Listed' সিলেক্ট করে 'root' এর নাম দিতে হবে, তারপরে 'root' ইউজারের পাসওয়ার্ড চাবে। পাসওয়ার্ড দেওয়ার পরে এবং পাসওয়ার্ড সঠিক হলে লগইন করতে দিবে।

লিনাক্স সিস্টেমে সব ধরনের ইউজারদের ডাটাবেজ ফাইল হচ্ছে '/etc/passwd' নিচের কমান্ডের মাধ্যমে সিস্টেমের সকল ইউজার সম্পর্কিত তথ্য পাওয়া যাবেঃ

[root@node1 ~]# less /etc/passwd ; স্ক্রলিং করে সম্পূর্ণ ফাইল দেখা যাবে। বের হওয়ার জন্য কীবোর্ড থেকে 'q' প্রেস করতে হবে।

লিনাক্স সিস্টেমে তিন (০৩) ধরনের ইউজার আছে নিচের তার লিস্ট দেওয়া হলঃ

1. root : 0
2. system user: 1 - 999
3. regular user: 1000 +

এখানে root ইউজারের আইডি (UID) '0', সিস্টেম ইউজারের আইডি (UID) '1-999' এবং রেগুলার ইউজারের আইডি (UID) '1000' থেকে শুরু হয়। লিনাক্স সিস্টেম ডিফল্ট ভাবে 60,000 ইউজার সাপোর্ট করে। তবে বেশি ইউজার হলে লোকাল সিস্টেমের বদলে ডাটাবেজ ব্যবহার হয়। কারণ, বেশি ইউজার লোকাল সিস্টেম দিয়ে ম্যানেজ করা কঠিন হয়ে যায়।

নির্দিষ্ট কোনো ইউজারের ইউজার আইডি (UID) জানার জন্য নিচের কমান্ডঃ

```
[root@node1~]# id root ; root ইউজার সিস্টেমে আছে কিনা জানার জন্য।  
[root@node1~]# id mail ; mail ইউজার সিস্টেমে আছে কিনা জানার জন্য।  
[root@node1~]# id student ; student ইউজার সিস্টেমে আছে কিনা জানার জন্য।
```

```
[root@node1~]# id tarek ; tarek ইউজার সিস্টেমে আছে কিনা জানার জন্য।  
[root@node1~]# useradd tarek ; নতুন ইউজার হিসেবে 'tarek' ইউজারকে তৈরি করা হয়েছে।  
[root@node1~]# id tarek ; tarek ইউজার তৈরি হয়েছে কিনা সেটা দেখার জন্য।
```

```
[root@node1~]# tail /etc/passwd ; '/etc/passwd' ফাইলে 'tarek' ইউজারের প্রোফাইল যোগ হয়েছে  
কিনা সেটা দেখার জন্য।
```

```
[root@node1~]# grep tarek /etc/passwd ; '/etc/passwd' ফাইল থেকে শুধুমাত্র 'tarek' ইউজারের  
প্রোফাইল চেক করার জন্য।
```

```
tarek:x:1001:1001:~/home/tarek:/bin/bash
```

```
1 2 3 4 5 6 7
```

- 1 - ইউজারের লগইন নাম, যেটা ইউনিক হবে এবং নামের মাঝে কোনো স্পেস থাকবে না। যেমনঃ tarek, tarek.hossain, tarek123, tarek_123,
- 2 - ইউজারের পাসওয়ার্ড সম্পর্কিত তথ্য থাকে যেটা মূলত থাকে '/etc/shadow'
- 3 - ইউজার আইডি (UID) প্রত্যেকটি ইউজারের আলাদা আলাদা ইউজার আইডি (UID) থাকবে।
- 4 - প্রাইমারী গ্রুপ আইডি (GID), ইউজার আইডি (UID) এর পাশাপাশি প্রত্যেকটি ইউজারের GID থাকবে, এটাকে বলা হয়, প্রাইমারী গ্রুপ (GID) আইডি।
- 5 - ইউজারের পূর্ণ নাম বা বিস্তারিত তথ্য। যেমনঃ "Md. Tarek Hossain", "Database Admin", "System Admin"
- 6 - ইউজারের হোম ডিরেক্টরি। যখন কোনো রেগুলার ইউজার তৈরি করা হয়, তখন '/home' ডিরেক্টরির মধ্যে সেই ইউজারের নামে একটি ডিরেক্টরি তৈরি হয়। যার মধ্যে উক্ত ইউজারের সকল প্রোফাইল সম্পর্কিত তথ্য থাকে।

7 - ইউজারের ব্যবহৃত শেলের নাম (/bin/bash)। এটা শুধুমাত্র যে সকল ইউজার সিস্টেমে লগইন করবে তাদের ক্ষেত্রে থাকবে। যে সকল ইউজার সিস্টেমে লগইন করার পার্মিশন নাই তাদের ক্ষেত্রে '/sbin/nologin'। ইউজারের শেল এন্টিভ/এনাবেল না থাকলে সেই ইউজার সিস্টেমে লগইন করতে পারবে না।

; সিস্টেমের সকল ইউজারের পাসওয়ার্ড সম্পর্কিত তথ্য থাকে '/etc/shadow' ফাইলে। ইউজারের পাসওয়ার্ড সম্পর্কিত তথ্য জানার জন্য নিচের কমান্ড।

```
[root@node1~]# tail /etc/shadow
```

; /etc/shadow' ফাইল থেকে 'tarek' ইউজারের পাসওয়ার্ড সম্পর্কিত তথ্য জানার জন্য নিচের কমান্ড।

```
[root@node1~]# grep tarek /etc/shadow
```

```
[root@node1~]# useradd hasan ; 'hasan' নামে আরও একজন ইউজার তৈরি করা হল।
```

```
[root@node1~]# useradd mamun ; 'mamun' নামে আরও একজন ইউজার তৈরি করা হল।
```

```
[root@node1~]# passwd hasan ; 'passwd' কমান্ডের মাধ্যমে 'hasan' ইউজারের পাসওয়ার্ড সেট করা হয়েছে।
```

```
=====  
=== Changing password for user hasan. ===  
=== New password: ***** (redhat) ===  
=== BAD PASSWORD: The password is shorter than 8 characters ===  
=== Retype new password: ***** (redhat) ===  
=====
```

নোটঃ পাসওয়ার্ড দেওয়া এবং পরিবর্তন করার জন্য একই কমান্ড। পাসওয়ার্ড দেওয়ার সময় কিছু ওয়ার্নিং দিবে, পাসওয়ার্ড হিসেবে আমরা এখানে 'redhat' ব্যবহার করবো। পাসওয়ার্ড স্ক্রিনে দেখা যাবে না, সেই সাথে কোনো ইউজারের পুরাতন পাসওয়ার্ড দেখা যাবে না, বা জানা যাবে না। পাসওয়ার্ড এনক্রিপ্টেড (hash) ফরম্যাটে থাকে।

```
[root@node1~]# tail /etc/shadow ; /etc/shadow' ফাইলে 'hasan' ইউজারের পাসওয়ার্ড সেট হয়েছে কিনা সেটা জানার জন্য।
```

যেহেতু আমরা 'hasan' ইউজারের পাসওয়ার্ড সেট করেছি, সুতরাং আলাদা টার্মিনাল ব্যবহার করে লগইন করে টেস্ট করবো যে, 'hasan' ইউজার দিয়ে সিস্টেমে লগইন করা যায় কিনা। আমরা কীবোর্ড থেকে 'Alt + Ctl + F3' প্রেস

করলে নতুন 'Login Prompt' আসবে সেখানে 'Login Name' হিসেবে 'hasan' লিখে 'Enter' প্রেস করলে 'hasan' ইউজারের পাসওয়ার্ড দেওয়ার অপশন আসবে। এখন পাসওয়ার্ড হিসেবে 'redhat' টাইপ করে এন্টার প্রেস করলে লগইন হবে।

নোটঃ পাসওয়ার্ড দেখা যাবে না।

login: hasan ; এখানে লগইন নাম হিসেবে 'hasan' দিতে হবে।
pass: *** ; 'hasan' ইউজারের পাসওয়ার্ড দিতে হবে। উল্লেখ্য যে, পাসওয়ার্ড স্ক্রিনে দেখা যাবে না এবং numlock চেক করে নিতে হবে।

```
[hasan@node1 ~]$ exit
```

; 'hasan' ইউজার দিয়ে লগইন করলে, তার জন্য আলাদা 'shell' ওপেন হবে। এখান থেকে বের হওয়ার জন্য 'exit' প্রেস করতে হবে। পুনরায় গ্রাফিক্যাল ইউজার ইন্টারফেসে ফিরে যাওয়ার জন্য 'Alt+ctrl+F2' প্রেস করতে হবে।

Class Work :

Create an user 'rony' and set his password 'redhat' and login through command console 'tty5'.

Session 02: Linux Group Administration

লিনাক্স সিস্টেমে নতুন 'Group' তৈরি করার জন্য 'groupadd' কমান্ড ব্যবহার করা হয়।

```
[root@node1~]# groupadd trainer ; 'groupadd' কমান্ড ব্যবহার করে 'trainer' নামে একটি গ্রুপ তৈরি করা হয়েছে।
```

```
[root@node1~]# groupadd staff ; 'groupadd' কমান্ড ব্যবহার করে 'staff' নামে আরও একটি গ্রুপ তৈরি করা হয়েছে।
```

যখন কোনো Group তৈরি করা হয় তখন '/etc/group' ফাইলে জমা হয়। লিনাক্স সিস্টেমে Group সম্পর্কিত তথ্য জানার জন্য নিচের কমান্ডঃ

```
[root@node1~]# tail /etc/group
```

```
trainer :x: 1004:
```

```
1 2 3 4
```

- 1 - গ্রুপের নাম (প্রাইমারি/সেকেন্ডারি গ্রুপ)
- 2 - গ্রুপের পাসওয়ার্ড সম্পর্কিত তথ্য থাকে যেটা মূলত থাকে '/etc/gshadow'
- 3 - গ্রুপ আইডি (GID)
- 4 - গ্রুপের সদস্য (Members)

নোটঃ যখন কোনো ইউজার তৈরি করা হয়, তখন ইউজার একাউন্টের পাশাপাশি উক্ত ইউজারের '/etc/group' ফাইলে গ্রুপ একাউন্টও তৈরি হয়। এটাকে বলা হয় প্রাইমারী গ্রুপ এবং প্রাইমারী গ্রুপ কখনো ডিলিট করা যায় না।

```
[root@node1~]# grep trainer /etc/group ; 'grep' কমান্ড ব্যবহার করে সিস্টেমে 'trainer' Group আছে কিনা চেক করা হয়েছে।
```

```
trainer:x:1004:
```

```
[root@node1~]# useradd -G trainer pavel ; নতুন ইউজার হিসেবে 'pavel' ইউজারকে 'trainer' গ্রুপে যোগ করা হল।
```

```
[root@node1~]# usermod -G trainer tarek ; পুরাতন ইউজার হিসেবে 'tarek' ইউজারকে 'trainer' গ্রুপে যোগ করা হল।
```

```
[root@node1~]# grep trainer /etc/group ; 'grep' কমান্ড ব্যবহার করে সিস্টেমে 'trainer' গ্রুপের মেম্বার লিস্ট দেখা হয়েছে।
```

```
trainer:x:1004:tarek,pavel
```

```
[root@node1~]# useradd ikbal ; 'ikbal' নামে আরও একজন ইউজার তৈরি করা হল।
```

```
[root@node1~]# usermod -G trainer,staff ikbal
```

; উপরের কমান্ডে ikbal' ইউজারকে একাধিক গ্রুপে 'staff & trainer' ঢোকানো করা হল।

```
[root@node1~]# grep staff /etc/group ; 'staff' গ্রুপের তথ্য দেখার জন্য।  
[root@node1~]# grep trainer /etc/group ; 'trainer' গ্রুপের তথ্য দেখার জন্য।
```

```
[root@node1~]# id ikbal ;'ikbal' ইউজার কোন কোন গ্রুপের সদস্য সেটা দেখার জন্য  
ব্যবহার করা হয়েছে।
```

```
uid=1005(ikbal) gid=1005(ikbal) groups=1005(ikbal),1005(staff),1004(trainer)
```

```
[root@node1~]# gpasswd -d ikbal staff ; কোনো ইউজার কে নির্দিষ্ট কোনো গ্রুপ থেকে বাদ দেওয়ার  
জন্য 'gpasswd' কমান্ড।
```

```
[root@node1~]# grep staff /etc/group ; ইউজার 'ikbal' 'staff' গ্রুপ থেকে রিমুভ হয়েছে কিনা  
সেটা চেক করার জন্য।
```

Session 03: User Profile Modify

Login as root user

```
[root@node1~]# useradd -u 3000 rafat ; নির্দিষ্ট ইউজার আইডি (UID) ব্যবহার করে 'rafat' নামে  
ইউজার তৈরি করা হয়েছে।
```

```
[root@node1~]# id rafat ; 'rafat' ইউজার তৈরি হয়েছে কিনা সেটা জানার জন্য।
```

```
[root@node1~]# groupadd -g 4000 support ; নির্দিষ্ট গ্রুপ আইডি (GID) ব্যবহার করে 'support' নামে  
গ্রুপ তৈরি করা হয়েছে।
```

```
[root@node1~]# tail /etc/group ; গ্রুপ সম্পর্কিত তথ্য দেখার জন্য।
```

```
[root@node1~]# grep rafat /etc/passwd ; 'rafat' ইউজারের প্রোফাইল চেক করা হয়েছে।  
rafat:x:3000:3000::/home/rafat:/bin/bash
```

এখন 'rafat' ইউজারের প্রোফাইল পরিবর্তন করা হবে। এখানে আমরা 'rafat' ইউজারের পূর্ণ নাম, হোম ডিরেক্টরি এবং শেল পরিবর্তন করবো।

```
[root@node1~]# usermod -c "Mr. Rafat Hussain" rafat
```

নোটঃ 'rafat' ইউজারের পূর্ণ নাম যোগ করা হয়েছে। এখানে '-c' অপশন ব্যবহার করা হয়েছে যার অর্থ 'comment'।

```
[root@node1~]# grep rafat /etc/passwd
rafat:x:3002:3002:Mr. Rafat Hussain:/home/rafat:/bin/bash
```

; আমরা এখানে দেখতে পাচ্ছি যে, 'rafat' ইউজারের পূর্ণ নাম যোগ হয়েছে।

যখন কোনো ইউজার তৈরি করা হয়, তখন ডিফল্ট হিসেবে সকল ইউজার '/home' ডিরেক্টরির মধ্যে তৈরি হয়। তবে, আমরা চাইলে লোকেশন পরিবর্তন করতে পারি।

```
[root@node1~]# grep rafat /etc/passwd
rafat:x:3002:3002:Mr. Rafat Hussain:/home/rafat:/bin/bash
```

নোটঃ এখানে দেখতে পাচ্ছি 'rafat' ইউজারের হোম ডিরেক্টরি হচ্ছে '/home/rafat'

ইউজারের হোম ডিরেক্টরি পরিবর্তন করার জন্য, প্রথমে অন্য স্থানে আরেকটি ডিরেক্টরি তৈরি করে নিতে হবে। যেটা ইউজারের নতুন হোম ডিরেক্টরি হিসেবে কাজ করবে।

```
[root@node1~]# mkdir -p /newhome/rafat2 ; '/newhome/rafat2' একটি হোম ডিরেক্টরি তৈরি করা হয়েছে।
```

```
[root@node1~]# usermod -d /newhome/rafat2 rafat
```

নোটঃ উপরের কমান্ডে 'rafat' ইউজারের হোম ডিরেক্টরি পরিবর্তন করে '/newhome/rafat2' লোকেশনে দেওয়া হয়েছে।

```
[root@node1~]# grep rafat /etc/passwd ; এখানে দেখা যাচ্ছে 'rafat' ইউজারের হোম ডিরেক্টরি পরিবর্তন হয়েছে।
rafat:x:3002:3002: /newhome/rafat2:/bin/bash
```

Change User's Shell :

এখন 'rafat' ইউজারের শেল পরিবর্তন করা হবে। যখন কোনো ইউজার তৈরি করা হয়, তখন ডিফল্ট শেল হিসেবে '/bin/bash' থাকে তবে, ইউজার চাইলে তার ইচ্ছামত শেল পরিবর্তন করতে পারবে। লিনাক্স সিস্টেমে বিভিন্ন ধরনের শেল থাকে 'sh, bash, cshell, zshell, tcshell'। এক এক ধরনের শেল আলাদা আলাদা কমান্ড, কমান্ড সিনট্যাক্স, সাপোর্ট করে। তবে, অধিকাংশ ডিস্ট্রিবিউশন ব্যবহার করে '/bin/bash' কোনো ইউজার তার শেল ছাড়া সিস্টেমে লগইন করতে পারবে না, সেটা লোকাল কনসোল দিয়ে হউক অথবা রিমোট কনসোল দিয়ে। মূলত যারা সিস্টেম অ্যাডমিন তাদের শেল এনাবেল '/bin/bash' আর যারা রেগুলার সার্ভিস ব্যবহারকারী ইউজার (Mail, FTP, File) কিন্তু, সিস্টেম অ্যাডমিন না, তাদের ক্ষেত্রে '/sbin/nologin' থাকা উচিত। আর '/sbin/nologin' থাকলে কোনো ইউজার কোনো ভাবেই সিস্টেমে লগইন করতে পারবে না।

```
[root@node1~]# cat /etc/shells ; লিনাক্স সিস্টেমে যত গুনা 'shell' ইন্টল আছে  
সেই লিস্ট দেখা যাবে।
```

```
/bin/sh  
/bin/bash  
/usr/bin/sh  
/usr/bin/bash
```

```
[root@node1~]# grep rafat /etc/passwd ; 'rafat' ইউজারের বর্তমান 'shell' দেখার জন্য।  
rafat:x:3000:3000: :/newhome/rafat2:/bin/bash
```

; 'rafat' ইউজারের 'shell' হিসেবে '/bin/bash' সেট করা আছে, যার অর্থ 'rafat' ইউজার সিস্টেমে লগইন করতে পারবে।

এখন 'rafat' ইউজারের 'shell' পরিবর্তন করে '/sbin/nologin' করা হবে।

```
-----  
[root@node1~]# usermod -s /sbin/nologin rafat
```

; 'rafat' ইউজারের 'shell' পরিবর্তন করে '/sbin/nologin' করা হয়েছে।

```
[root@node1~]# grep rafat /etc/passwd  
rafat:x:3000:3000:Mr. Rafat Hussain:/newhome/rafat2:/sbin/nologin
```

; আউপুটে দেখা যাচ্ছে 'rafat' ইউজারের 'shell' পরিবর্তন হয়ে '/sbin/nologin' হয়েছে।

```
[root@node1~]# su rafat ; unable to switch rafat user
```

This account is currently not available.

```
[root@node1~]# usermod -s /bin/bash rafat ; পুনরায় 'rafat' ইউজারের লগইন 'shell'
'/bin/bash' সেট করা হয়েছে।
```

```
[root@node1~]# grep rafat /etc/passwd
rafat:x:3000:3000: :/newhome/rafat2:/bin/bash
```

```
[root@node1~]# su rafat ; আমরা চাইলে আবার চেক করে নিতে পারি।
```

User's Account Lock and unlock:

```
[root@node1~]# grep student /etc/shadow ; 'student' ইউজারের পাসওয়ার্ড সম্পর্কিত তথ্য
জানার জন্য।
```

```
student:$6$sfs8vX/U.....!yRg1::0:99999:7::: ; এখানে দেখতে পাচ্ছি 'student' ইউজারের পাসওয়ার্ড
দেওয়া আছে এবং শুরুতে '$6' মানে একটিভ আছে।
```

```
[root@node1~]# usermod -L student ; 'student' ইউজারের একাউন্ট লক করে দেওয়া হয়েছে।
এখানে '-L' মানে 'Lock'
```

```
[root@node1~]# grep student /etc/shadow
student: ! $6$sfs8vX/U.....!yRg1::0:99999:7:::
```

; এখানে দেখতে পাচ্ছি 'student' ইউজারের পাসওয়ার্ড দেওয়া আছে এবং কিন্তু শুরুতে '!\$6' মানে Locked করা।

- 1) ::: - কোনো পাসওয়ার্ড সেট করা হয়নি।
- 2) :\$6\$sfs8vX/U: => পাসওয়ার্ড সেট করা আছে এবং একটিভ (Unlocked)।
- 3) !:\$6\$sfs8vX/U: => পাসওয়ার্ড সেট করা আছে এবং ইন-একটিভ (Locked)।
- 4) : : => পাসওয়ার্ড রিমুভ করে দেওয়া হয়েছে। ইউজার পাসওয়ার্ড ছাড়াই লগইন করতে পারবে।

Check: Alt + Ctrl + F3 (use username password)

; 'Alt + Ctl + F3' প্রেস করে নতুন কনসোল দিয়ে লগইন করে দেখা যেতে পারে।

```
[root@node1~]# usermod -U student ; 'student' ইউজারের একাউন্ট পুনরায় Unlock করা হয়েছে।  
এখানে '-U' অর্থ 'Unlock'।
```

```
[root@node1~]# grep student /etc/shadow  
student:$6$sfs8vX/U..... !yRg1::0:99999:7:::
```

```
[root@node1~]# id ikbal ; 'ikbal' নামে কোনো ইউজার আছে কিনা চেক করা হয়েছে।
```

```
[root@node1~]# userdel -r ikbal
```

```
[root@node1~]# id ikbal
```

; ইউজারের সকল তথ্য (home/mail/data/log) সবকিছু সহ মুছতে চাইলে 'userdel -r' কমান্ড ব্যবহার করতে হবে।

```
[root@node1~]# cat /etc/passwd ; সিস্টেমে ইউজারের প্রোফাইল ডেটাবেজ চেক করলে 'rafat'
```

ইউজারের আর কোন তথ্য পাওয়া যাবে না।

```
[root@node1~]# groupdel support ; সিস্টেম থেকে গ্রুপ মুছে দেবার জন্য 'groupdel' কমান্ড  
ব্যবহার করা হয়।
```

```
[root@node1~]# tail /etc/group ; /etc/group ফাইল থেকে গ্রুপ রিমুভ হয়েছে কিনা সেটা দেখার  
জন্য।
```

Session 04: Linux SUDO Command

Working with Linux SUDO Privilege :

এখন লিনাক্স সিস্টেমে 'sudo' প্রিভিলেজ নিয়ে কাজ করা হবে, এইজন্য একজন নতুন ইউজার দরকার, সেজন্য 'mahedi' নামে একজন ইউজার তৈরি করা হবে এবং সেই সাথে 'mahedi' ইউজারের পাসওয়ার্ড সেট করা হবে।

```
[root@node1 ~]# useradd mahedi
```

```
[root@node1 ~]# passwd mahedi
```

```
[root@node1 ~]# su - mahedi
```

```
[mahedi@node1 ~]$ useradd rasel
useradd: Permission denied.
useradd: cannot lock /etc/passwd; try again later.
```

আমরা যদি 'mahedi' ইউজার দিয়ে লগইন করে 'rasel' নামে ইউজার তৈরি করতে চাই, তাহলে উপরের মত আউটপুট 'Permission denied' দিবে। কারণ, 'mahedi' ইউজার একজন রেগুলার ইউজার, সুতরাং 'mahedi' ইউজার অন্য কোনো ইউজারকে তৈরি করতে পারবে না। তবে, যদি কোনো রেগুলার ইউজারকে সুপার ইউজারের প্রিভিলেজ দেওয়া হয়, তখন উক্ত রেগুলার ইউজার সুপার ইউজারের পার্মিশনের উপরে ভিত্তি করে বিভিন্ন আডমিনিস্ট্রেশন কমান্ড রান করতে পারবে।

What is SUDO do?

SUDO প্রিভিলেজের মাধ্যমে একজন রেগুলার ইউজারকে (student, tarek, mahedi) সুপার ইউজারের প্রিভিলেজ বা এক্সেস দেওয়া হয়। এক্ষেত্রে উক্ত রেগুলার ইউজারকে নির্দিষ্ট কোনো কমান্ড (single command), বা একের অধিক কমান্ড (Group of commands), বা চাইলে সিস্টেমের সকল কমান্ডের (All Commands) প্রিভিলেজ (হুবুহু root ইউজারের মত) এক্সেস দেওয়া যেতে পারে।

লিনাক্স সিস্টেমে রুট (root) ইউজার কর্তৃক রেগুলার ইউজার তৈরি করা হয়। যেমনঃ student, tarek, hasan এবং রেগুলার ইউজার যে সকল কমান্ড গুলো রান করতে বা চালাতে পারে সেইগুলো '/bin' ডিরেক্টরিতে থাকে। কিছু কমান্ডের লিস্ট নিচে দেওয়া হয়েছেঃ

- ⌘ rm, cp, mv,
- ⌘ mkdir, touch
- ⌘ pwd, free -m
- ⌘ ping, df -HT
- ⌘ ip addr, tail

লিনাক্স সিস্টেমে অ্যাডমিন ইউজার হচ্ছে রুট (root) ইউজার। root ইউজার সিস্টেমের সকল কমান্ড রান করতে বা চালাতে পারে সেইগুলো '/bin' এবং '/sbin' ডিরেক্টরিতে থাকে। root ইউজার কর্তৃক রান করা যায়, এমন কিছু কমান্ডের লিস্ট নিচে দেওয়া হয়েছে -

- ⌘ useradd, userdel, groupadd
- ⌘ reboot, fdisk -l,
- ⌘ shutdown, poweroff, init

⌘ setenforce

Note: [root@node1~]# which useradd ; কোন কমান্ড কোন ডিরেক্টরি থেকে রান করে সেটা জানার জন্য 'which' কমান্ড -

```
[root@node1~]# which pwd
```

```
[root@node1~]# which touch
```

Editing sudo configuration File :

এখন যদি কোনো রেগুলার ইউজার (mahedi) কে সুপার ইউজারের (root) মত প্রিভিলেজ দিতে চাই, তাহলে '/etc/sudoers' ফাইলটিতে সেই ইউজারের নাম উল্লেখ করে প্রিভিলেজ দেওয়া যাবে। '/etc/sudoers' সরাসরি ওপেন করার জন্য নিচের কমান্ড 'visudo' ব্যবহার করতে হবে।

Rules 01: Permit Mahedi user for all commands

```
[root@node1~]# visudo
```

```
:set nu
```

```
100 root ALL=(ALL) ALL ; এটার অর্থ হচ্ছে 'root' ইউজার সকল কমান্ড (ALL) রান করতে পারে।  
101 mahedi ALL=(ALL) ALL ; শুধুমাত্র 'mahedi' ইউজারকে রুট (root) ইউজারের মত সম্পূর্ণ  
প্রিভিলেজ (ALL) দেওয়া হয়েছে।
```

```
:x (save and exit)
```

Switch to Mahedi User :

```
[root@node1 ~]# su - mahedi
```

```
[mahedi@node1 ~]$
```

; এখন যদি আমরা 'mahedi' ইউজার দিয়ে আবার লগইন করে ইউজার তৈরি করার চেষ্টা করি তাহলে একই 'permission denied' মেসেজ দিবে।

```
-----  
[mahedi@node1 ~]$ useradd rasel
```

```
[mahedi@node1 ~]$ sudo useradd rasel
```

; SUDO ইউজার হিসেবে সিস্টেম থেকে Authentication নিতে চাইলে, শুরুতে 'sudo' ব্যবহার করতে হবে। এবং কমান্ড চালানোর সময় sudo ইউজারের পাসওয়ার্ড দিতে হবে।

```
[sudo] password for mahedi: ****
```

```
[mahedi@node1 ~]$ tail /etc/passwd
```

```
[mahedi@node1 ~]$ exit
```

Rules 02: Permit 'trainer' group for Group of commands :

; লিনাক্স সিস্টেমের কোনো গ্রুপের (/etc/group) সকল সদস্যদের SUDO প্রিভিলেজ দিতে চাইলে, গ্রুপের শুরুতে '%' দিয়ে উল্লেখ করতে হবে। তাহলে, গ্রুপের সকল ইউজারের ক্ষেত্রে SUDO প্রিভিলেজ পেয়ে যাবে।

```
[root@node1 ~]# visudo
```

```
108 %wheel      ALL=(ALL)    ALL
```

```
109 %trainer ALL=(ALL)    ALL
```

Rules 03: Permit only some specific commands :

```
[root@node1~]# visudo
```

```
49 Cmnd_Alias NOC = /usr/sbin/useradd, /usr/sbin/usermod, /usr/sbin/groupadd
```

```
102 mahedi ALL=(ALL) NOC
```

; প্রথমে যে কমান্ড গুলো নির্দিষ্ট কোনো ইউজারের জন্য প্রিভিলেজ দিতে চাই, সেই কমান্ড গুলো '/etc/sudoers' ফাইলের উপরের দিকে (Line 49) 'Cmnd_Alias' অপশন দিয়ে উল্লেখ করতে হবে। তারপরে যে ইউজার কে উক্ত প্রিভিলেজ দেওয়া হবে, সেই ইউজারকে নিচের দিকে (Line 101) উল্লেখ করতে হবে।

Session 05: Password Policies

Working with /etc/shadow file :

```
[root@node1~]# useradd lucky
[root@node1~]# passwd lucky
```

```
: redhat
: redhat
```

```
[root@node1~]# chage -d 0 lucky ; ইউজার 'lucky' তার পরবর্তী লগইনের সময় পাসওয়ার্ড পরিবর্তন করে নিতে পারবে।
```

এখানে '-d' অর্থ 'day' এবং '0' দিন।

Testing: Alt + Ctl + F3 ব্যবহার করে চেক করা যাবে।

```
password: P@ssword123 (new password)
```

```
[root@node1~]# grep lucky /etc/shadow
```

```
lucky: $6 $ciiMifom $cPpqBIf2NOwan2byi.G6D0iM/g.tw7fcUyLDWIs.nbp0 :19570: 0 : 99999 : 7 :
: :
```

```
    ১ম      |                ২য়
|   ৩য়   | ৪   |   ৫ম   | ৬   | ৭   | ৮   | ৯
```

1st - ইউজারের নাম।

2nd - ইউজারের পাসওয়ার্ড (:!@, :!\$, :\$6..., : :)

3rd - সর্বশেষ কবে তার পাসওয়ার্ড পরিবর্তন করেছে। এখানে যে সংখ্যাটা দেখা যাচ্ছে সেটা দিন (19570) হিসেবে। আর এটা ধরা হয়, ০১-০১-১৯৭০ সাল থেকে আজ পর্যন্ত।

4th - পাসওয়ার্ডের মেয়াদ (Minimum)। এখানে ডিফল্ট ভাবে '0' থাকে, এর অর্থ হচ্ছে ইউজার চাইলে, যে কোনো সময় তার পাসওয়ার্ড পরিবর্তন করতে পারবে।

5th - পাসওয়ার্ডের মেয়াদ (Maximum)। ইউজার তার বর্তমান পাসওয়ার্ড (redhat) কত দিন ব্যবহার করতে পারবে।

6th - পাসওয়ার্ডের মেয়াদ উত্তীর্ণ হওয়ার কত দিন আগে নোটিফিকেশন দিবে। (ওয়ানিং প্রিয়ড)।

7th - পাসওয়ার্ডের মেয়াদ উত্তীর্ণ হওয়ার পর কত দিন পর্যন্ত একাউন্ট একটিভ থাকবে।

8th - একাউন্টের মেয়াদ উত্তীর্ণ হওয়ার তারিখ।

9th - Reserved for future use.

```
[root@node1~]# useradd sathi
```

```
[root@node1~]# chage -l sathi
```

```
[root@node1~]# chage -l sathi ; পাসওয়ার্ড সম্পর্কিত সকল তথ্য জানতে 'chage -l' কমান্ড ব্যবহার করা যাবে।
```

```
Last password change                : MM DD, YYYY
Password expires                     : never
Password inactive                    : never
Account expires                      : never
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
```

```
[root@node1~]# chage sathi
```

```
Minimum Password age [0]: 3
```

```
Maximum Password age [99999]: 30
```

```
Last Password Changed (YYYY-MM-DD): Press Enter (today)
```

```
Password Expiration Warning [7]: 5
```

```
Password Inactive [-1]: 5
```

```
Account Expiration Date (YYYY-MM-DD) [-1]: YYYY-MM-DD
```

note: If press Enter account never expire

```
[root@node1~]# chage -l sathi
```

```
[root@node1~]# tail /etc/shadow | grep sathi
```

```
[root@node1~]# date
```

```
[root@node1~]# date MMDDHHMMYY (MM = Month, DD = Day, HH= Hour, MM = Minute, YY=Year)
```

```
[root@node1~]# date
```

```
[root@node1~]# vim /etc/login.defs ; এই ফাইল থেকে ভ্যালু পরিবর্তন করে ইউজারের ডিফল্ট  
পাসওয়ার্ড সেটিংস পরিবর্তন করা যাবে।
```

```
131 PASS_MAX_DAYS 60
```

```
132 PASS_MIN_DAYS 0
```

```
133 PASS_MIN_LEN 6
```

```
134 PASS_WARN_AGE 5
```

```
[root@node1~]# useradd rakib
```

```
[root@node1~]# passwd rakib
```

Login regular user :

Send Key: Alt + Ctrl + F3 - নতুন কনসোল ওপেন করে লগইন করে চেক করা হবে পাসওয়ার্ড দেওয়ার অপশন আসে কিনা।

Send Key: Alt+Ctrl+F2 - পুনরায় গ্রাফিক্যাল ইউজার ইন্টারফেসে ফিরে যাওয়ার জন্য 'Alt+ctrl+F2' প্রেস করতে হবে।

Homework : - 1

1. create a group "rhcsa19"
2. user "oly, liza, rana" member of 'rhcsa19' group
3. all user's password set 'redhat'
4. lock liza user's account
5. Lock rafat user's shell
6. Create user robin using UID '5001'
7. Allow user 'oly' as a sudo user
8. User 'rana' must be reset his password on next login

//////////////////// Thank you //////////////////////

Class -08

Working with Linux File/Directory permission & ownership :

```
[root@node1 ~]# mkdir linux21
```

```
[root@node1 ~]# cd linux21
```

```
[root@node1 linux21]# ls
```

```
[root@node1 linux21]# mkdir day8
```

```
[root@node1 linux21]# cd day8
```

```
[root@node1 day8]# touch file1
```

; বর্তমান ডিরেক্টরি 'day8' এর মধ্যে 'file1' নামে একটা ফাইল তৈরি করা হল।

```
[root@node1 day8]# mkdir dir1
```

; বর্তমান ডিরেক্টরি 'day8' এর মধ্যে 'dir1' নামে একটা ডিরেক্টরি তৈরি করা হল।

```
[root@node1 day8]# cp /etc/passwd .
```

; '/etc/passwd' ফাইলটি কপি করে বর্তমান ডিরেক্টরি 'day8' মধ্যে নিয়ে আসা হয়েছে।

```
[root@node1 day8]# ll
```

```
d rwxr-xr-x. 2 root root 6 May 10 17:54 dir1
-rw-r--r--. 1 root root 2662 May 10 17:55 passwd
-rw-r--r--. 1 root root 0 May 10 17:54 file1
```

```
1 2 3 4 5 6 7 8
```

1: file/dir types (লিনাক্সে মোট সাত ধরনের (-, d, c, b, l, s, p) ফাইল/ডিরেক্টরি পাওয়া যাবে)

2: file/dir permission (User + Group + Others + ACL)

3: file/dir link (Hard Link) - ফাইলের ক্ষেত্রে '1' এবং ডিরেক্টরির ক্ষেত্রে '2' থাকবে, তবে ভিতরে যত ডিরেক্টরি থাকবে তার উপরে ভিত্তি করে সংখ্যা বাড়বে।

4: file/dir owner (যে ইউজার ফাইল/ডিরেক্টরিটি তৈরি করেছে তার নাম থাকবে)

5: file/dir group owner - (ইউজার যে গ্রুপের সদস্য সেই গ্রুপের নাম থাকবে)

6: file/dir size (byte) - ফাইল হলে '0' বাইট, এবং ডিরেক্টরি হলে '6' বাইট।

7: file/dir modify date

8: file/dir name

Session 01

Working with link file :

লিঙ্ক ফাইলের অর্থ, একটি ফাইলের সাথে আরেকটি ফাইলের লিঙ্ক থাকে। অর্থাৎ কোনো ফাইলে কোনো লেখা/টেক্সট মোডিফাই/আপডেট করলে অপর ফাইলটিও একই ভাবে আপডেট হবে। যেমনঃ উইন্ডোজ অপারেটিং সিস্টেমের ক্ষেত্রে শর্টকাট ফাইল। ফাইলের শর্টকাট গুলা থাকে আমাদের ডেস্কটপে এবং সেগুলার সোর্স (Origin) থাকে অন্য লোকেশনে। আমরা যদি ডেস্কটপ থেকে সেই সকল শর্টকাট ফাইলের উপরে ক্লিক করি, তাহলে মূল সোর্স (Origin) থেকে রান করবে।

লিনাক্স সিস্টেমে দুই ধরনের লিঙ্ক ফাইল আছেঃ

Type of link file:

1) Hard link - ফাইল গুলার আইনোড নাম্বার (inode) একই হবে। অর্থাৎ দুইটি ফাইলের মাঝে যদি হার্ডলিঙ্ক করা হয়, তাদের কনটেন্ট, সাইজ, পার্মিশন একই হবে, শুধু নাম আলাদা হবে। এক্ষেত্রে মূল ফাইলটি ডিলিট হলেও লিঙ্ক ফাইলের কোনো পরিবর্তন বা সমস্যা হবে না। হার্ডলিঙ্ক শুধুমাত্র ফাইলের ক্ষেত্রেই করা যাবে।

2) Soft link - ফাইল গুলার আইনোড নাম্বার (inode) ভিন্ন হবে। দুইটা ফাইল/ডিরেক্টরির মধ্যে যদি সফট লিঙ্ক করা হয়, তাদের কনটেন্ট একই হবে শুধু নাম আলাদা হবে। সফটলিঙ্ক বা symbolik লিঙ্ক ফাইলের শুরুতে 'l' থাকে। এক্ষেত্রে যদি মূল ফাইলটি ডিলিট করা হয়, তাহলে লিঙ্ক ফাইল কোনো কাজ করবে না। সফট লিঙ্ক ফাইল এবং ডিরেক্টরি উভয়ের ক্ষেত্রেই করা যায়।

```

[root@node1 day8]# ls -li
[root@node1 day8]# tail passwd
[root@node1 day8]# ln -s passwd passwd-soft ;softlink তৈরির ক্ষেত্রে '-s' ব্যবহার করতে হবে।

[root@node1 day8]# ln passwd passwd-hard ;hardlink তৈরির কমান্ড।
[root@node1 day8]# ll
[root@node1 day8]# tail passwd-hard
[root@node1 day8]# tail passwd-soft

[root@node1 day8]# ls -li

[root@node1 day8]# echo "The End" >> passwd
[root@node1 day8]# tail -3 passwd
[root@node1 day8]# tail -3 passwd-hard
[root@node1 day8]# tail -3 passwd-soft
[root@node1 day8]# rm -f passwd
[root@node1 day8]# ll
[root@node1 day8]# tail -3 passwd-hard
[root@node1 day8]# tail passwd-soft
[root@node1 day8]# ln -s /home/student/Documents studentdoc

```

Session 02

Field no: 2, 4 & 5 (Permission)

```

[root@node1 day8]# ll

- rw-r--r--. 1 root root 0 Sep 26 09:52 file1
d rwxr-xr-x. 2 root root 4096 Sep 26 09:33 dir1
(2) (4) (5)

```

subfield:

- rw- r-- r-- . = file1 - ডিফল্ট ভাবে লিনাক্সের ফাইলের কোনো এক্সিকিউট (x) পার্মিশন থাকে না।
d rwx r-x r-x . = dir1 - ডিফল্ট ভাবে লিনাক্সের ডিরেক্টরিতে ইউজার, গ্রুপ এবং অন্যান্য ইউজারদের জন্য এক্সিকিউট (x) পার্মিশন থাকে।

(u) (g) (o) (A)

u = user

g = group

o = others

A = ACL Permission (.)

r = read (4) - ফাইলের ক্ষেত্রে 'read' অর্থ ফাইলটি পড়তে পারবে এবং ডিরেক্টরির ক্ষেত্রে read অর্থ, ডিরেক্টরি টা কপি করা যাবে।

w = write (2) - ফাইলের ক্ষেত্রে 'write' অর্থ ফাইলটি এডিট এবং সেভ করা যাবে। ডিরেক্টরির ক্ষেত্রে write অর্থ, ডিরেক্টরির ভিতরে paste করা যাবে।

x = execute (1) - ফাইলের ক্ষেত্রে 'execute' পার্মিশন দরকার নাই। কিন্তু, ফাইলটি যদি, প্রোগ্রাম/স্ক্রিপ্ট/এপ্লিকেশন হয়, তাহলে অবশ্যই execute

পার্মিশন থাকতে হবে। ডিরেক্টরির ক্ষেত্রে 'execute' হচ্ছে ডিরেক্টরির মধ্যে প্রবেশ করার জন্য, অর্থাৎ যে ডিরেক্টরিতে 'execute' পার্মিশন নাই, সেখানে প্রবেশ করা যাবে না।

- = no permission (0) - এক্ষেত্রে 'read/write/execute' কোনো পার্মিশন থাকছে না।

. = ACL Permission (+) - ফাইল/ডিরেক্টরি উভয়ের ক্ষেত্রে এখানে dot '(.)' থাকবে। কিন্তু, যখন ACL (File Access Control) এপ্লাই করা হবে তখন, এখানে Dot (.) এর পরিবর্তে '+' সাইন চলে আসবে।

- rw- r-- r-- . = file1 (644)
d rwx r-x r-x . = dir1 (755)

$rw- = 4 + 2 + 0 = 6$ (U)	$rwx = 4 + 2 + 1 = 7$ (U)
$r-- = 4 + 0 + 0 = 4$ (G)	$r-x = 4 + 0 + 1 = 5$ (G)
$r-- = 4 + 0 + 0 = 4$ (O)	$r-x = 4 + 0 + 1 = 5$ (O)
-----	-----
644	= 755

** ডিফল্ট ফাইল পার্মিশন = 644 এবং ডিরেক্টরি পার্মিশন = 755 তবে, আমরা আমাদের প্রয়োজনে পার্মিশন লেভেল পরিবর্তন করতে পারি।

Maximum File Permission: 666 => (rw- rw- rw-)
Maximum Directory Permission: 777 => (rwx rwx rwx)

Class Work:

740 = rwx r-- ---
735 = rwx -wx r-x
642 = rw- r-- -w-
503 = r-x --- -wx
150 = --x r-x ---
326 = -wx -w- rw-
467 = r-- rw- rwx
631 = rw- -wx --x

Lab Demo: Linux Standard File Permission

এখানে 'file1' নিয়ে কাজ করা হবে। যেখানে একটা গ্রুপ থাকবে 'students' নামে এবং 'students' গ্রুপের মেম্বার হবে 'sadat, sumon, malek'। সেই সাথে একজন গেস্ট ইউজার (Others) থাকবে 'tamim' নামে।

```
Group:      Members                others
=====      =====                =====
students:   malek, sumon, sadat    all (except group members)
```

```
[root@node1 day8]# groupadd students
[root@node1 day8]# useradd -G students malek
[root@node1 day8]# useradd -G students sadat
[root@node1 day8]# useradd -G students sumon
```

```
[root@node1 day8]# useradd tamim
```

```
[root@node1 day8]# grep students /etc/group
students:x:5005:malek,sadat,sumon
```

এখন 'file1' কে 'Malek' ইউজারের Ownership এবং 'students' গ্রুপের Group Ownership দেওয়া হবে। 'malek' ইউজারের জন্য পূর্ণ (rwx) পার্মিশন দেওয়া হবে। পাশাপাশি 'students' গ্রুপের ইউজারদের শুধু read (r) পার্মিশন থাকবে এবং অন্য ইউজারের/গেস্ট (others) ক্ষেত্রে কোনো প্রকার পার্মিশন থাকবে না।

```
user: (malek) : full permission (rwx)
group: students: read (r--)
others: others : no  (---)
```

```
[root@node1 day8]# ls -l
-rw-r--r--. 1 root root 0 Jun 14 19:55 file1
```

```
[root@node1 day8]# chown malek file1 ; chown কমান্ড ব্যবহার করে Ownership
পরিবর্তন করা যায়।
```

```
[root@node1 day8]# ls -l
-rw-r--r--. 1 malek root 0 Jun 14 19:55 file1
```

```
[root@node1 day8]# chgrp students file1 ; chgrp কমান্ড ব্যবহার করে Group Ownership
পরিবর্তন করা যায়।
```

```
[root@node1 day8]# ls -l
-rw-r--r--. 1 malek students 0 Sep 26 09:52 file1
```

```
[root@node1 day8]# chmod 740 file1 ; chmod কমান্ড ব্যবহার করে পার্মিশন (rwx)
পরিবর্তন করা যায়।
```

```
[root@node1 day8]# ls -l
-rwxr-----. 1 malek students 0 Jun 14 19:55 file1
```

Testing:

(Ownership Test)

```
[root@node1 day8]# su malek
[malek@node1 day8]$ ls
[malek@node1 day8]$ echo "This is malek" > file1 ; malek can rw
[malek@node1 day8]$ cat file1
[malek@node1 day8]$ exit
```

(Groupowner Test)

```
[root@node1 day8]# su sumon
[sumon@node1 day8]$ ls

[sumon@node1 day8]$ cat file1
[sumon@node1 day8]$ echo "This is sumon" > file1 ; Permission Denied (only read)
[sumon@node1 day8]$ exit
```

(Others)

```
[root@node1X day8]# su tamim
[tamim@node1X day8]$ cat file1 ; access denied ; Permission Denied (no permission)
[tamim@node1X day8]$ echo "This is tamim" > file1 ; Permission Denied (no permission)
[tamim@node1X day8]$ exit
```

Linux File Access Control List (ACL):

- ⌘ যদি একটা ফাইল/ডিরেক্টরিতে একাধিক Ownership বা Group Ownership দরকার হয়।
- ⌘ একটা ফাইল/ডিরেক্টরিতে আলাদা আলাদা ইউজার এবং গ্রুপের আলাদা আলাদা পার্মিশন দরকার হলে।
- ⌘ নির্দিষ্ট কোনো গ্রুপ থেকে নির্দিষ্ট কোনো ইউজারের প্রিভিলেজ তুলে নিতে চাইলে বা পরিবর্তন করতে চাইলে।
- ⌘ আমরা যদি গেস্ট (others) ইউজার থেকে কাউকে স্পেশাল প্রিভিলেজ দিতে চাই।

তাহলে আমরা যদি উপরে উল্লেখিত ব্যাপারে চিন্তা করি, তাহলে আমাদের স্ট্যান্ডার্ড ফাইল/ডিরেক্টরি পার্মিশন ব্যবহার করে সেটা করা সম্ভব না। সেক্ষেত্রে ব্যবহার করতে হবে 'ACL' পার্মিশন। ACL পূর্ণ অর্থ (Access Control List), যেটা আমরা ফাইল এবং ডিরেক্টরি উভয়ের ক্ষেত্রে ব্যবহার করতে পারি। ACL সিকিউরিটি সার্ভিস লিনাক্স ফাইল সিস্টেমের (ext4/xfs) উপরে নির্ভর করে। যেহেতু আমাদের ফাইল সিস্টেম 'xfs' সুতরাং এটা ডিফল্ট ভাবে এনাবেল থাকে।

Working Directory:

```
[root@node1X day8]# ls
[root@node1X day8]# touch tutorial profile
[root@node1X day8]# ll

-rw-r--r--. 1 root root 0 Jun 28 17:00 tutorial
-rw-r--r--. 1 root root 0 Jun 28 17:00 profile
```

এই ল্যাব ডেমো প্র্যাকটিস করার জন্য 'jack', 'rose', 'tomy' নামে তিন জন ইউজার এবং 'support' নামে গ্রুপ একাউন্ট তৈরি করা হয়েছে।

```
[root@node1X day8]# useradd jack
[root@node1X day8]# useradd rose
```

```
[root@node1X day8]# useradd tomy
[root@node1X day8]# groupadd support
```

ACL Test:

```
[root@node1X day8]# getfacl profile ; ফাইল/ডিরেক্টরিতে ACL পার্মিশন দেখার কমান্ড।
# owner: root
# group: root
user::rw-
group::r--
other::r--
```

ACL Command Options :

- m = modify
- x = remove
- b = blank
- R = Recrusivly

User ACL :

Lab: আমরা নিচের উদাহরণে, 'profile' ফাইলে jack এর জন্য 'rwx', ইউজার 'rose' এর জন্য 'r--' এবং ইউজার 'tomy' এর ক্ষেত্রে কোনো পার্মিশন দেওয়া হবে না।

```
[root@serverX day8]# setfacl -m u:jack:rwx,rose:r--,tomy:--- profile ;ফাইল/ডিরেক্টরিতে ACL
পার্মিশন দেওয়ার কমান্ড।
```

```
[root@serverX day8]# ll
-rw-rw-r--+ 1 root root 0 Jun 28 17:00 profile
```

```
[root@serverX day8]# getfacl profile
# file: profile
# owner: root
```

```
# group: root
user::rw-
user:jack:rwx
user:rose:r--
user:tomy:---
group::r--
mask::rwx
other::r--
```

উপরের কমান্ড আউটপুটে দেখা যাচ্ছে, ইউজার 'jack' এর ক্ষেত্রে 'rwx' যোগ হয়েছে। পাশাপাশি ইউজার 'rose' এর ক্ষেত্রে 'r--' এবং other ইউজার 'tomy' এর ক্ষেত্রে পার্মিশন তুলে নেওয়া হয়েছে।

```
[root@serverX day8]# su jack
[jack@serverX day8]$ cat profile
[jack@serverX day8]$ echo this is Jack > profile
[jack@serverX day8]$ cat profile
[jack@serverX day8]$ exit
```

```
[root@serverX day8]# su rose
[rose@serverX day8]$ cat profile
[rose@serverX day8]$ echo this is Rose >> profile ; permission denied
[rose@serverX day8]$ exit
```

```
[root@serverX day8]# su tomy
[tomy@serverX day8]$ cat profile
[tomy@serverX day8]$ echo this is Tomy >> profile
[tomy@serverX day8]$ exit
```

***** Permission Denied

Group acl :

```
[root@serverX day8]# setfacl -m g:support:rw- tutorial
[root@serverX day8]# getfacl tutorial
```

Directory Permission:

```
[root@serverX day8]# mkdir akdir
[root@serverX day8]# touch akdir/file1
[root@serverX day8]# touch akdir/file2
[root@serverX day8]# ls -l akdir
[root@serverX day8]# setfacl -R -m u:rose:rxw akdir ; -R (recursively)
[root@serverX day8]# ls -l akdir
[root@serverX day8]# getfacl akdir
```

ACL Remove (user):

```
[root@serverX day8]# setfacl -x u:rose: profile
[root@serverX day8]# getfacl profile
```

ACL Remove (Group):

```
[root@serverX day8]# setfacl -x g:support: tutorial
[root@serverX day8]# getfacl tutorial
```

Remove ACL from File:

```
[root@serverX day8]# setfacl -b tutorial
```

Session 03

Linux SUID, SGID and Sticky Bit Concept:

লিনাক্সের স্ট্যান্ডার্ড পার্মিশনের পাশাপাশি কিছু স্পেশাল পার্মিশন ব্যবহার করা হয়।

⌘ স্ট্যান্ডার্ড পার্মিশন প্রকাশ করা হয় r=read, w=write, x=execute, -= no permission

⌘ স্পেশাল পার্মিশন প্রকাশ করা হয় ' S,S,T/s,s,t' দিয়ে।

Read=4 (r)

Write=2 (w)

Execute=1 (x)

no permission = 0 (-)

$rwX = 4+2+1 = 7$

উপরের 'Standard Permission' এর সাথে যোগ হবে লিনাক্স স্পেশাল (SUID/SGID/Sticky) পার্মিশন।

4 = SUID (s,S) - SUID পার্মিশন ব্যবহার হয় Owner ফিল্ডে (1st field) এবং এটা প্রকাশ করা হয় 'S' অথবা 's' দিয়ে।

2 = SGID (s,S) - SGID পার্মিশন ব্যবহার হয় Group ফিল্ডে (2nd field) এবং এটাও প্রকাশ করা হয় 'S' অথবা 's' দিয়ে।

1 = Sticky bit (t, T) - Sticky পার্মিশন ব্যবহার হয় Other ফিল্ডে (3rd field) এবং এটা প্রকাশ করা হয় 'T' অথবা 't' দিয়ে।

নোটঃ যদি ফাইল/ডিরেক্টরিতে আগে থেকেই এক্সিকিউট পার্মিশন থাকে, তাহলে স্পেশাল পার্মিশন সেট করার পরে ছোট হাতের 's,s,t' থাকবে, আর যদি কোনো এক্সিকিউট পার্মিশন (x) না থাকে সেক্ষেত্রে 'S,S,T'

উদাহরণ স্বরূপঃ

`rwX r-x r-x = rws r-s r-t` - 'x' বিট সহ স্পেশাল পার্মিশনের ব্যবহার করা হয়।

`rw- r-- r-- = rwS r-S r-T` - 'x' বিট বাদে স্পেশাল পার্মিশনের ব্যবহার করা হয়।

u g o

x755 (where x = Special Permission for directory)

x644 (where x = Special Permission for file)

একটা ডিরেক্টরির ক্ষেত্রে যদি চিন্তা করি,

4 + 755 = SUID (1st Field)

2 + 755 = SGID (2nd Field)

1 + 755 = Sticky bit (3rd Field)

Working with Test SUID:

SUID - যখন কোনো ফাইল/কমান্ডকে রেগুলার ইউজার কে সুপার ইউজারের (root) প্রিভিলেজে রান করার দরকার হয়, তখন সেই ফাইল বা কমান্ডের উপরে SUID সেট করা হয়। তার অর্থ উক্ত ফাইল/কমান্ডটি সুপার ইউজারের (root) (file Owner) পাশাপাশি যে কোনো রেগুলার ইউজার একই প্রিভিলেজে এক্সেস/রান করতে পারবে।

```
[root@node1 day8]# which passwd
[root@node1 day8]# ls -l /bin/passwd
-rwsr-xr-x. 1 root root 34512 Aug 12 2018 /bin/passwd
```

```
[root@node1 day8]# ls -l /usr/bin/ls
-rwxr-xr-x. 1 root root 166448 Jan 11 2018 /usr/bin/ls
```

```
[root@node1 day8]# su student
[student@node1 day8]$ ls /root
ls: cannot open directory '/root': Permission denied
```

```
[student@node1 day8]$ exit
[root@node1 day8]# ls -l /usr/bin/ls
-rwxr-xr-x. 1 root root 166448 Jan 11 2018 /usr/bin/ls
```

```
[root@node1 day8]# chmod 4755 /usr/bin/ls ; SUID Permission
```

```
[root@node1 day8]# ls -l /usr/bin/ls
-rwsr-xr-x. 1 root root 166448 Jan 11 2018 /usr/bin/ls
```

```
[root@node1 day8]# su student
[student@node1 day8]$ ls /root
[student@node1 day8]$ exit
```

Class work: Login as 'student' user and view log file '/var/log/messages' using 'tail'

Hints: [root@node1 day8]# tail -5 /var/log/messages

Hints: [student@node1 day8]\$ tail -5 /var/log/messages

Working with SGID:

SGID যখন ব্যবহার করা হবে, তখন গ্রুপের মেম্বারদের মধ্যে কোনো ইউজার যদি কোনো ফাইল/ডিরেক্টরি তৈরি করলে, তার প্রিভিলেজ আটোমেটিক্যালি গ্রুপের অন্যান্য মেম্বাররা পেয়ে যাবে। অর্থাৎ প্রতিটি ফাইলের গ্রুপ মেম্বারশিপ 'Inherit' হবে।

```
[root@node1 day8]# mkdir resource
```

```
[root@node1 day8]# ls -ld resource
```

```
drwxr-xr-x. 2 root root 6 Sep 18 22:13 resource
```

```
[root@node1 day8]# grep students /etc/group
```

```
students:x:1008:malek,sumon,sadat
```

```
[root@node1 day8]# chmod 770 resource
```

```
[root@node1 day8]# chgrp students resource
```

```
[root@node1 day8]# ls -ld resource
```

```
drwxrwx---. 2 root students 6 Sep 18 22:13 resource
```

```
[root@node1 day8]# su malek
```

```
[malek@node1 day8]$ touch resource/malek1
```

```
[malek@node1 day8]$ exit
```

```
[root@node1 day8]# su sadat
```

```
[sadat@node1 day8]$ touch resource/sadat1
```

```
[sadat@node1 day8]$ exit
```

```
[root@node1 day8]# ll resource
```

```
[root@node1 day8]# chmod 2770 resource ; SGID Permission
[root@node1 day8]# ls -ld resource
drwxrws---. 2 root students 6 Sep 18 22:13 resource
```

```
[root@node1 day8]# su sumon
[sumon@node1 day8]$ touch resource/sumon1
[root@node1 day8]# ll resource
```

Working with Sticky Bit :

লিনাক্সে ডিরেক্টরির ম্যাক্সিমাম পার্মিশন হচ্ছে '777' এবং এই ম্যাক্সিমাম পার্মিশন দেওয়ার পরে যে কোনো ইউজার চাইলে এই ডিরেক্টরির মধ্যকার ফাইল ডিলিট করে দিতে পারে। কিন্তু, আমরা যদি উক্ত ডিরেক্টরির উপরে Sticky bit পার্মিশন প্রয়োগ করি তাহলে, যে কোনো ইউজার ডিরেক্টরির মধ্যে ফাইল তৈরি করতে বা পেস্ট করতে পারবে কিন্তু ডিলিট করতে পারবে না।

```
[root@node1 day8]# ls -ld /tmp
```

```
[root@node1 day8]# mkdir mydir1
[root@node1 day8]# mkdir mydir2
```

```
[root@node1 day8]# ll
[root@node1 day8]# chmod 777 mydir1 ; standard permission
[root@node1 day8]# chmod 1777 mydir2 ; sticky bit
```

```
[root@node1 day8]# ls -ld mydir1 mydir2
drwxrwxrwx. 2 root root 6 Sep 18 21:53 mydir1
drwxrwxrwt. 2 root root 6 Sep 18 21:53 mydir2
```

```
[root@node1 day8]# touch mydir1/file1
```

```
[root@node1 day8]# touch mydir2/file2
```

```
[root@node1 day8]# su student
```

```
[student@node1 day8]$ rm mydir1/file1 ; এটার প্যারেন্ট ডিরেক্টরিতে ফুল পার্মিশন এটা ডিলিট হয়ে  
যাবে।
```

```
[student@node1 day8]$ rm mydir2/file2 ; এটার প্যারেন্ট ডিরেক্টরিতে যেহেতু sticky বিট সেট করা  
আছে সুতরাং এটা ডিলিট হবে না।
```

```
rm: cannot remove 'mydir2/file2': Operation not permitted
```

Home Work: 02

a) Copy the file '/etc/passwd' to /opt

=> The file '/etc/passwd' is owned by the 'root' user.

=> The file '/etc/passwd' belongs to the group 'root'

=> The file '/etc/passwd' should not be executable by anyone.

=> The user 'robin' is able to read and write '/opt/passwd'

=> The user 'rafat' can neither write nor read '/opt/passwd'

=> All other users (current or future) have the ability to read '/opt/passwd'

b) Create a collaborate directory "/shared/sysusers"

=> Group ownership of "/shared/sysusers" is 'sysusers'

=> Permission should be readable, writeable and access by 'sysusers' group but not to any other user.

=> Files created in "/shared/sysusers" automatically have group ownership set to the 'sysusers' Group.

//////////////////// Thank you //////////////////////

Class -09

Linux Boot up Process :

==> Multiuser (tty)

ON => BIOS/UEFI => MBR/GPT => GRUB2 => Linux Kernel => systemd => Target =>
(POST)

==> Graphical (pts)

==> Rescue/emergency

Change Default Boot Time :

আমরা যখন সিস্টেম চালু করি, তখন Firmware (MBR/GPT) স্টেজ পার হয়ে বুট লোডারে (GRUB) প্রবেশ করে, তখন সিস্টেমের বুট লোডার ডিফল্ট ভাবে ৫ সেকেন্ড অপেক্ষা করে। এই ৫ সেকেন্ডের মধ্যে কার্নেল সিলেক্ট না করলে ডিফল্ট কার্নেল লোড শুরু হবে। তবে, আমরা চাইলে বুট লোডারের (Grub) টাইম বাড়াতে পারি।

```
[root@node1 ~]# fdisk -l
```

Disklevel Type: dos (MBR)

Disklevel Type: gpt (EFI)

Working with Linux Kernel:

⌘ Device Driver

⌘ Kernel Modules (Filesystem, Security, Virtualization, Storage, Networking, Memory mgt)

⌘ Kernel-Subsystem

-> File System

-> Storage

⌘ Latest patch update

⌘ Software Bug Fixing

⌘ Security fixes

⌘ Updated Drivers

⌘ Kernel Modules update

⌘ New Kernel Features

⌘ Increased system speed

⌘ ip addr

আমরা দুইভাবে কান্নেল আপডেট করতে পারিঃ

1. Major update (Compile)
2. Minor update

```
[root@node1 ~]# uname -r
```

```
5.14.0-284.11.1.el9_2.x86_64
```

5.14.0 - Linux Kernel Built Version

70.13.1 - Redhat Package Version

el9_2 - Redhat Release number

x86_64 - Processor Arch (ppc, arm, x86)

```
[root@node1 ~]# ping www.centos.org
```

```
[root@node1 ~]# ping www.redhat.com
```

```
[root@node1 ~]# ping www.rocky.org
```

Note: নেটওয়ার্ক কানেকশন এন্টিভেট করার জন্য উপরের ডান পাশ থেকে (গ্রাফিক্যাল ইন্টারফেসের) পাওয়ার বাটনে ক্লিক করলে নেটওয়ার্কিং আইকন আসবে। সেখান থেকে 'connect' -এ ক্লিক করে এন্টিভেট করা যাবে।

```
[root@node1 ~]# yum list installed kernel-*
```

```
[root@node1 ~]# yum update kernel -y
```

```
[root@node1 ~]# reboot
```

```
[root@node1 ~]# yum list installed kernel
```

```
[root@node1 ~]# uname -r
```

systemd:

```
[root@node1 ~]# pstree
```

RHEL/CentOS 7.0 তাদের পূর্বের (RHL/RHEL 6) সার্ভিস ম্যানেজার System-V (sysv) থেকে সুইচ করে নতুন ভার্সনে systemd তে পরিবর্তন করেছে। সুতরাং, RHEL/CentOS 7 থেকে নতুন সার্ভিস ম্যানেজারের নাম হচ্ছে 'systemd'। যদিও systemd আগের ভার্সনের (init & SysV) সাথে সকল কমান্ড/সার্ভিসের সাথে সঙ্গতি রেখে

মেনে চলে। বর্তমানে RedHat, Ubuntu, Debian, Fedora, CentOS সবগুলো ডিস্ট্রিবিউশন 'systemd' ব্যবহার করে।

RHEL 9:

```
[root@node1 ~]# ls /lib/systemd/system/*.service ; সিস্টেমের সকল সার্ভিসের লিস্ট
[root@node1 ~]# ls /lib/systemd/system/*.service | wc -l ; মোট কতগুলো সার্ভিস আছে সেটার লিস্ট
[root@node1 ~]# systemctl -t service
[root@node1 ~]# systemctl --type=service --state=active
```

Common Services:

উপরের কমান্ডের আউটপুটে দেখা গেল অনেক সার্ভিস আমাদের সিস্টেমে চলে। কিন্তু, সব সার্ভিস আমাদের মনে রাখার দরকার নাই। এক্ষেত্রে নিচে কিছু কমন সার্ভিসের লিস্ট দেওয়া আছে, যেগুলো সচরাচর ব্যবহার করা হয়।

- ⌘ Network Service: NetworkManager.service
- ⌘ SSH Service: sshd.service
- ⌘ Web/WWW: httpd.service (Apache), nginx.service
- ⌘ NFS Service: nfs-server.service
- ⌘ Firewall: firewalld.service, iptables, fail2ban, ufw => Firewall: firewalld.service, iptables, fail2ban, ufw
- ⌘ Linux Scheduling: crond.service
- ⌘ Time Server: ntpd.service / chronyd.service
- ⌘ Mail (Outgoing): postfix.service (SMTP)
- ⌘ Mail (Incoming): dovecot.service (IMAP/POP3)
- ⌘ Virtualization: libvirtd.service
- ⌘ Samba Server: smb.service
- ⌘ NetBIOS: nmb.service
- ⌘ FTP Server: vsftpd.service
- ⌘ DNS Server: named.service (bind)
- ⌘ Database: mariadb, mysqld, postgresql
- ⌘ Printing: cupsd.service
- ⌘ DHCP Server: dhcpcd.service
- ⌘ Proxy Server: squid.service

Start/Stop/Restart/status Services with systemctl:

```
[root@node1 ~]# systemctl status crond.service
```

```
[root@node1 ~]# systemctl stop crond.service
```

```
[root@node1 ~]# systemctl status crond.service
```

```
[root@node1 ~]# systemctl restart crond.service
```

```
[root@node1 ~]# systemctl reload crond.service
```

Enable / Disable services to run at boot time :

```
[root@node1 ~]# systemctl enable crond.service
```

```
[root@node1 ~]# systemctl status crond.service
```

```
[root@node1 ~]# systemctl disable crond.service
```

```
[root@node1 ~]# systemctl status crond.service
```

Working with Runlevels/Targets :

‘systemd’ কর্তৃক সকল সার্ভিস লোড হওয়ার পরে সিস্টেম কোন মোডে (CLI, GUI, Rescue) রান করবে এটা নির্ভর করে ‘Systemd target’ সেটিংসের উপর।

systemd => 1) GUI (graphical.target)

2) CMD (multi-user.target)

3) Rescue.target

4) Emergency.target

নোটঃ সিস্টেম মেইনটেনেন্স, ট্রাবুলশ্টিং বা রিকভারি করার উদ্দেশ্যে ‘rescue.target’ সেট করা হয়। এই মোডে শুধু মাত্র ‘root’ (Single User) ইউজারের শেল প্রিভিলেজ থাকে এবং লিমিটেড কিছু সার্ভিস লোড হয়।

```
[root@node1 ~]# systemctl get-default  
graphical.target
```

```
[root@node1 ~]# free -m
```

```
[root@node1 ~]# systemctl set-default multi-user.target  
[root@node1 ~]# systemctl get-default  
multi-user.target
```

```
[root@node1 ~]# reboot
```

```
[root@node1 ~]# systemctl get-default  
multi-user.target
```

```
[root@node1 ~]# free -m
```

```
[root@node1 ~]# systemctl set-default graphical.target
```

```
[root@node1 ~]# reboot
```

Reset Root Password :

⌘ প্রথমে সিস্টেম কমান্ড মোডে 'Ctrl+Alt+Del' প্রেস করে অথবা গ্রাফিক্যাল মোডে 'restart' দিতে হবে।

⌘ সিস্টেম বুট হওয়ার সময় Grub বুট লোডার স্ক্রিন থেকে '0-rescue-xxx' ধরতে হবে।

⌘ কীবোর্ড থেকে 'e' প্রেস করে '0-rescue-xxx' কার্নেল এডিট করতে হবে।

⌘ কীবোর্ড থেকে Down Arrow প্রেস করে '4' নাম্বার লাইনে (linux দিয়ে শুরু যে লাইন) যেতে হবে।

⌘ এরপরে লাইনের শেষে (linux শুরু যেখানে) 'end' বাটন বা 'Right Arrow' প্রেস করে যেতে হবে।

⌘ লাইনের শেষে 'rd.break' লিখতে হবে। নোটঃ rd = rescue disk

⌘ এরপরে কীবোর্ড থেকে 'Ctrl+x' প্রেস করে সিস্টেম লোড করতে হবে।

```
switch_root:/# mount -o remount,rw /sysroot ; পুনরায় root ফাইল সিস্টেম rw মোডে মাউন্ট  
করা হল।
```

```
switch_root:/# chroot /sysroot ; শেল এক্সেস করার জন্য।
```

sh-5.1# passwd [press enter] ; নতুন পাসওয়ার্ড পরিবর্তন করা হল।

```
: ***** (123456) <----- redhat
: ***** (123456) <----- redhat
```

sh-5.1# touch /.autorelabel ; SELinux Policiy Relabel

sh-5.1# exit

switch_root:/# exit

⌘ Wait 2 Min

⌘ Login with new password

[root@desktopX ~]# passwd ; press Enter

```
: ***** (centos)
: ***** (centos)
```

Linux Process Management:

```
# ps
# ps aux
# ps -ef
# ps lax
```

লিনাক্স সিস্টেম অ্যাডমিনিস্ট্রেশন জবের একটি গুরুত্বপূর্ণ দায়িত্ব হচ্ছে সিস্টেমের পারফরমেন্স (Performance) মনিটরিং এবং টিউনিং। লিনাক্স সিস্টেমের সব ধরনের রিসোর্স, যেমনঃ প্রসেসর (CPU), মেমরি (RAM & Swap), স্টোরেজ, অ্যাপ্লিকেশন, প্রসেস, সার্ভিস সহ যাবতীয় মনিটরিং করা যায়।

[root@node1 ~]# ps ; বর্তমান টার্মিনালে কি কি প্রসেস রান করতেছে সেটা দেখার জন্য।

```
PID TTY TIME CMD
```

```
2521 pts/0 00:00:00 bash <----- টার্মিনালে পিছনে শেল রান করতেছে।
```

```
2622 pts/0 00:00:00 ps <----- টার্মিনালের সামনের যে কমান্ড 'ps' টি দেওয়া হয়েছে সেটা দেখা
```

যাচ্ছে।

[root@node1 ~]# ps -f ; বর্তমান টার্মিনালের সকল প্রসেস বিস্তারিত ভাবে (full description) দেখাবে।

```
-----  
UID      PID    PPID    C   STIME   TTY      TIME    CMD  
root    2521   2516    0   08:25   pts/0    00:00:00  bash  
root    2737   2521    0   08:35   pts/0    00:00:00  ps -f  
(1)      (2)    (3)    (4)  (5)    (6)      (7)     (8)
```

- 1) যে ইউজারের (root) মাধ্যমে প্রসেস চলছে।
- 2) প্রসেস (Child) আইডি (PID)
- 3) প্যারেন্ট (Parent) প্রসেস আইডি (PPID)
- 4) প্রসেসরের (CPU) ব্যবহার
- 5) প্রসেস স্টার্ট (Start) টাইম
- 6) প্রসেস কোন টার্মিনালে (TTY) চলছে।
- 7) প্রসেস টাইম (Duration)
- 8) প্রসেস দেখার জন্য যে কমান্ড (CMD) টি ব্যবহার করা হয়েছে।

[root@node1 ~]# ps -e ; সকল প্রসেস, সকল টার্মিনাল (pts0/pts1/tty3) থেকে দেখার জন্য।

[root@node1 ~]# ps -ef | less ; pid, ppid, time, cmd সহ বিস্তারিত স্ক্রলিং করে দেখার জন্য।

[root@node1 ~]# ps -ef | grep http

[root@node1 ~]# ps -e --forest ; সকল প্রসেস ট্রি (Tree) আকারে দেখাবে।

[root@node1 ~]# ps -auf

Type of Process :

1. User Defined Process : jobs & ps
2. System Defined Process : ps

[root@node1 ~]# ps -x ; সিস্টেমের সকল প্রসেস দেখার জন্য।

Lab demo:

এটা প্র্যাকটিস করার জন্য Alt+Ctrl+F3 প্রেস করে করে 'student' ইউজার দিয়ে লগইন করে, 'ping ::1' রান করে রেখে, পুনরায় 'Alt+Ctrl+F2' প্রেস করে GUI টার্মিনালে প্রবেশ করে নিচের কমান্ডের মাধ্যমে প্র্যাকটিস করা যাবে।

```
[root@node1 ~]# ps -u ; বর্তমান ইউজারের (root) সকল প্রসেস দেখার জন্য।
```

```
[root@node1 ~]# ps -au ; বর্তমান ইউজার সহ অন্যান্য টার্মিনালে যে সকল ইউজারের প্রসেস চলছে সব দেখাবে।
```

```
[root@node1 ~]# ps -au | less
```

```
-----  
student  3361  0.5  0.1 234492  5384 tty3    Ss  09:00  0:00 -bash  
root     3420  0.4  0.0 225792  1648 tty4    Ss+ 09:00  0:00 /sbin/agetty  
student  3437  0.0  0.0 242200  2280 tty3    S+  09:00  0:00 ping ::1  
root     3465  8.0  0.1 270524  4084 pts/0   R+  09:00  0:00 ps -ua  
-----
```

```
[root@node1 ~]# ps -u student ; নির্দিষ্ট ইউজারের প্রসেস দেখার জন্য এই কমান্ড ব্যবহার করতে হবে।
```

```
-----  
PID TTY          TIME CMD  
3137 ?            00:00:00 systemd  
3148 ?            00:00:00 (sd-pam)  
3161 ?            00:00:00 pulseaudio  
3162 tty5        00:00:00 bash  
3195 ?            00:00:00 dbus-daemon  
-----
```

```
[root@node1 ~]# top
```

top - * বর্তমান সিস্টেম টাইম।

* সিস্টেম কতক্ষণ ধরে চলছে।

* কত জন ইউজার লগইন করে আছে।

* সিপিউ (CPU) লোড এভারেজ (১ মিনিট, ৫ মিনিট এবং ১৫ মিনিট অন্তর অন্তর)

⌘ load average: Last 1 minute is 0.01

⌘ load average: Last 5 minutes is 0.02

⌘ load average: Last 15 minutes is 0.00

Tasks:

⌘ Total: যে সকল প্রসেস একটিভ মোডে আছে।

⌘ Running: বর্তমানে রানিং প্রসেসের সংখ্যা।

⌘ Sleeping: যে সকল প্রসেস স্লিপ মোডে আছে।

⌘ Stopped: যে সকল প্রসেস স্টপ (Stopped) অবস্থায় আছে।

⌘ Zombie: যে সকল প্রসেস Zombie State অবস্থায় আছে।

%Cpu(s) - সকল CPU Core এর তথ্য জানতে চাইলে '1' প্রেস করতে হবে।

PID - প্রসেস আইডি

USER - যে ইউজার কর্তৃক প্রসেস রান হচ্ছে।

PR (20) - Priority (PR=20+NI) - কার্নেল স্পেস

NI (0) - NICE Value - ইউজার স্পেস

VIRT - প্রসেস কর্তৃক ভার্চুয়াল মেমরির ব্যবহার।

RES - Residential Memory (Actual RAM)

SHR - Shared Memory

S - বর্তমান অবস্থা (status) - R= Running, S=Sleeping, I=Idle

%CPU - প্রসেস কর্তৃক CPU% ব্যবহার।

%MEM - প্রসেস কর্তৃক RAM% ব্যবহার।

TIME+ - কতক্ষন ধরে প্রসেস রানিং আছে।

COMMD - প্রসেসের নাম।

[root@node1 ~]# pidof systemd ; 'systemd' এর প্রসেস আইডি দেখার জন্য এই কমান্ড টি ব্যবহার করা হয়েছে।

```
3344 3334 3148 1900 1572 1565 1
```

[root@node1 ~]# pstree ; প্রসেস সমূহ Tree (শাখা-প্রশাখা) আকারে দেখাবে।

Process Dealing:

প্রসেস রান, সাসপেন্ড, স্টপ, প্রসেস প্রায়োরিটি বাড়ানো বা কমানোর কিছু কমান্ড ব্যবহার করতে হবে।

ps - ইউজার এবং সিস্টেম কর্তৃক যে সকল প্রসেস রান হয় সেটা জানার জন্য।

⌘ jobs - শুধু মাত্র ইউজার কর্তৃক যে সকল প্রসেস বর্তমান টার্মিনালে শুরু হয়েছে সেই গুনা দেখাবে।

⌘ kill - প্রসেস Kill (Terminate) করার জন্য ব্যবহার করা হয়।

⌘ bg - কোনো প্রসেস কে টার্মিনালের পিছনে (Background) পাঠানোর কমান্ড।

⌘ fg - কোনো প্রসেস কে টার্মিনালের ব্যাক গ্রাউন্ডে থেকে সামনে (Foreground) নিয়ে আসার কমান্ড।

⌘ nice - কোনো প্রসেস কে নির্ধারিত 'NICE' ভ্যালু দিয়ে রান করার জন্য ব্যবহার করা হয়।

⌘ renice - রানিং কোনো প্রসেসের 'priority' বাড়ানো বা কমানোর জন্য ব্যবহার করা হয়।

Foreground and Background Process :

```
[root@node1 ~]# firefox
```

ডিফল্ট ভাবে প্রসেস Foreground -এ রান করবে। এ অবস্থায় টার্মিনালে কোনো কাজ করা যাবে না। আমরা যদি কোনো প্রসেস টার্মিনালের ব্যাকগ্রাউন্ডে চালাতে চাই তাহলে কমান্ডের শেষে '&' যোগ করতে হবে।

```
[root@node1 ~]# firefox &
```

```
[root@node1 ~]# jobs
```

Process Stop and Terminate :

যে কোনো প্রসেস রানিং থেকে স্টপ করে দেওয়া যাবে এবং স্টপ করে পুনরায় রানিং করা যাবে। চাইলে স্থায়ী ভাবে প্রসেস টার্মিনেট (kill) করা যাবে। তবে, টার্মিনেট (kill) করলে সেটা আর পুনরায় রান করা যাবে না।

* 'Ctrl+Z' - কোনো প্রসেসকে স্টপ (suspend) করে ব্যাকগ্রাইন্ডে পাঠানোর কমান্ড।

* 'Ctrl+C' - প্রসেসকে টার্মিনেট (kill) করার জন্য।

```
[root@desktop ~]# ping ::1
```

```
64 bytes from ::1: icmp_seq=5 ttl=64 time=0.095 ms
64 bytes from ::1: icmp_seq=6 ttl=64 time=0.093 ms
^C
```

(Ctrl + C) - প্রেস করে প্রসেসটি Terminate (end) করা হয়েছে।

```
[root@desktop ~]# ping ::1
```

```
64 bytes from ::1: icmp_seq=1 ttl=64 time=0.091 ms
64 bytes from ::1: icmp_seq=2 ttl=64 time=0.079 ms
^Z
```

```
[4]+ Stopped          ping ::1
```

(Ctrl + Z) - প্রেস করে 'ping' প্রসেসটি Stop (Hold) করা হয়েছে।

```
[root@node1 ~]# top ; রিয়েল টাইম প্রসেস মনিটরিং করার জন্য 'top' ব্যবহার করা হয়।
```

'Ctrl + Z' বাটন প্রেস করে 'top' প্রসেসটি স্টপ (hold) করতে পারেন।

```
[root@node1 ~]# jobs - শুধু মাত্র রানিং প্রসেস গুলা দেখাবে
```

```
-----
[1]  Running          firefox &
[2]- Stopped         top
[3]+ Stopped         ping ::1
```

কোনো স্টপ (hold) প্রসেস কে রানিং অবস্থায় ফিরে নিয়ে আসতে 'fg' কমান্ড ব্যবহার করা হয়। এক্ষেত্রে 'fg' কমান্ডের সাথে '%' ব্যবহার করে প্রসেস সিরিয়াল নাম্বার দিতে হবে।

```
[root@node1 ~]# fg %3
```

Process Findout:

কোনো রানিং প্রসেসের প্রসেস আইডি (PID) বের করার জন্য নিচের কমান্ড।

```
[root@node1 ~]# firefox &
[root@node1 ~]# pidof firefox
3452 3373 3352 3317 3190 (main)
```

Process Kill & Terminate :

```
[root@node1 ~]# pkill <Process name>
[root@node1 ~]# kill <PID>      ; Normal kill
[root@node1 ~]# kill -9 <PID>   ; Forcely killed
                                ; "-9" killing frequency (highest)
```

Working wtih Nice Value:

```
[root@node1 ~]# top

(Low) 19..... 0 ..... -20 (High)
```

Nice Value:

```
[root@node1 ~]# ps axo pid,comm,nice --sort=-nice
[root@node1 ~]# firefox &
[root@node1 ~]# ps axo comm,nice | grep firefox
firefox  0
```

New Process Priority:

```
[root@node1 ~]# nice -n 15 firefox &
[root@node1 ~]# ps axo comm,nice | grep firefox
firefox  15
```

Existing Process Priority:

```
[root@node1 ~]# pidof firefox
[root@node1 ~]# renice -n -10 [pid]
[root@node1 ~]# ps axo comm,nice | grep firefox
```

===== The End =====

Class -10

Linux Partition Management :

যখন RHEL 9 অপারেটিং সিস্টেম ইন্সটল করা হয়, তখন ০৪ (চার) টা পার্টিশন করা হয় এবং সেগুলার সাইজ যথাক্রমে নিচে উল্লেখ করা হয়েছেঃ

- 1) /boot/efi - 200 MiB (GPT ভিত্তিক পার্টিশনের ক্ষেত্রে লাগবে)
- 2) /boot - 1024 MiB
- 3) / - 20 GiB
- 4) swap - 2 GiB

[root@desktop ~]# lsblk ; এই কমান্ড দিয়ে সিস্টেমের সকল পার্টিশন লিস্ট দেখা যাবে।

```
sda      8:0    0   30G  0 disk
├─sda1   8:1    0   200M  0 part /boot/efi
├─sda2   8:2    0    1G   0 part /boot
├─sda3   8:3    0   20G  0 part /
└─sda4   8:4    0    2G   0 part [SWAP]
```

Partition Table Structure :

লিনাক্স/উইন্ডোজ সিস্টেমে দুইভাবে পার্টিশন টেবিল তৈরি হয়, 1) MBR 2) GPT

* BIOS/Legacy Firmware ==> MBR (Master Boot Record) - লিনাক্সে দেখা যায় 'dos' নামে।

* UEFI Firmware ==> GPT(GUID Partition Table) + MBR - লিনাক্সে দেখা যায় 'gpt' নামে।

```
=====
== [root@server ~]# fdisk -l                                     ===
== Disk /dev/sda: 30 GiB, 32212254720 bytes, 62914560 sectors    ===
== Units: sectors of 1 * 512 = 512 bytes                        ===
== Sector size (logical/physical): 512 bytes / 512 bytes      ===
== I/O size (minimum/optimal): 512 bytes / 512 bytes         ===
== Disklabel type: dos                                         ===
== Disk identifier: 0x7231a122                                  ===
```

```

=====
== Device      Boot    Start      End  Sectors  Size Id Type           ==
== /dev/sda1   *        2048  1026047  1024000  500M 83 Linux           ==
== /dev/sda2           1026048 42969087 41943040   20G 83 Linux           ==
== /dev/sda3           42969088 47163391  4194304    2G 82 Linux swap/Solaris ==
=====

```

```

=====
== [root@server ~]# fdisk -l                               ==
== Disk /dev/sda: 30 GiB, 32212254720 bytes, 62914560 sectors ==
== Units: sectors of 1 * 512 = 512 bytes                    ==
== Sector size (logical/physical): 512 bytes / 512 bytes   ==
== I/O size (minimum/optimal): 512 bytes / 512 bytes       ==
== Disklabel type: gpt                                       ==
== Disk identifier: 0x7231a122                               ==
==                                                         ==
== Device          Start      End  Sectors  Size Type           ==
== /dev/sda1       2048  1026047  1024000  500M EFI System    ==
== /dev/sda2       1026048 42969087 41943040   20G Linux filesystem ==
== /dev/sda3       42969088 47163391  4194304    2G Linux swap     ==
=====

```

MBR = 32 bit, $2^{32} \times 512 = 2\text{TB}$

GPT = 64bit , $2^{64} \times 512 = 9.4 \text{ ZB}$

ডিস্ক পার্টিশন কি ?

আমাদের সিস্টেমের (Windows/Linux) অপারেটিং সিস্টেম এবং ইউজার/এপ্লিকেশন ডেটা আলাদা পার্টিশনে বা ড্রাইভে রাখার জন্য ডিস্ক পার্টিশন করা হয়।

ফলে আমাদের সিস্টেম কোনো একটি পার্টিশন ত্রুটি (Crash) করলে বা পার্টিশন ডিলিট হয়ে গেলেও অন্য পার্টিশনের ডাটা অক্ষত থাকবে।

পার্টিশন ফরম্যাট বা ফাইল সিস্টেম কি ?

অপারেটিং সিস্টেম ভেদে এক একটি পার্টিশনের ফরম্যাট আলাদা আলাদা পদ্ধতিতে হয়ে থাকে। যেমনঃ উইন্ডোজ অপারেটিং সিস্টেমের ক্ষেত্রে NTFS এবং FAT32

পার্টিশন ফরম্যাট ব্যবহার করা হয়ে থাকে। NTFS এর ক্ষেত্রে Security, encryption, File Size, Partition Size, Block Size, Compression

ইত্যাদি ফিচার থাকে, যেটা FAT32 তে পাওয়া যাবে না। যেমনঃ FAT32 তে ম্যাক্সিমাম ডাটা ক্যাপাসিটি ৩২ গিগা বাইট, অপরদিকে NTFS 2TB পর্যন্ত সাপোর্ট করে, আবার NTFS ফাইল সিস্টেম ডাটা Encryption করতে পারে, কিন্তু FAT32 পারে না।

Window File System: FAT32, NTFS

Linux File system: ext2, ext3, ext4, XFS (current), vfat, swap, GFS2, ZFS (BSD)

যেমনঃ EXT4 ফাইল সিস্টেম 50 TB, এবং XFS ফাইল সিস্টেম 1 PB পর্যন্ত সাপোর্ট করে।

All device files location: /dev/*

* SSD, HDD, DVD, USB, Serial, Keyboard, Sound Card, Virtual Terminal (tty)

MBR ==> 512-byte sectors, 2TB Max Partition Size (Single)

GPT ==> 512-byte sectors, 9.4 Zettabytes Max Partition Size

Total Partition: MBR - BIOS (dos)

Linux Partition = 15 (4 Primary + 11 Logical)

Total Partition: GPT - UEFI

Total Partition: 128 (Windows/Linux)

Disk Name:

IDE/SATA/SAS/SCSI HDD: sda, sdb, sdc

NVMe = nvme

Linux Virtual Machine: vda, vdb (Exam)

USB: sda1, sdb1

DVD: sr0

[root@node1~]# fdisk -l ; সিস্টেমের পার্টিশন টেবিল দেখা যাবে।

sda = 1st disk

sdb = 2nd disk
sdc = 3rd disk

vda = 1st virtual disk
vdb = 2nd virtual disk

Linux partition ID: (MBR)

NTFS - 7
Extended - 5
ext3/ext4/xfs - 83
swap - 82
LVM - 8e
vfat - f

Create New Partition :

```
[root@node1 ~]# fdisk -l  
[root@node1 ~]# cfdisk  
[root@node1 ~]# parted  
(parted) print free
```

নোটঃ parted থেকে বের হওয়ার জন্য 'q' প্রেস করতে হবে।

```
[root@node1 ~]# fdisk /dev/sda ; MBR/GPT based পার্টিশন করার ক্ষেত্রে ব্যবহার হয়।
```

নোটঃ GPT Based পার্টিশন করার জন্য 'gdisk /dev/sda' ব্যবহার করা যেতে পারে।

Command (m for help): m 'm' বাটন প্রেস করার মাধ্যমে বিভিন্ন হেল্প অপশন পাওয়া যাবে।

- d পার্টিশন ডিলিট করার জন্য ব্যবহার হয়।
- l কি কি ধরনের পার্টিশন ID আছে সেটা জানার জন্য।
- m প্রেস করে বিভিন্ন হেল্প অপশন পাওয়া যাবে।
- n নতুন পার্টিশন তৈরি করার জন্য।
- p কি কি পার্টিশন তৈরি করা আছে সেটা জানার জন্য।
- q হেল্প অপশন থেকে বের হওয়ার জন্য।
- t পার্টিশন আইডি পরিবর্তন করার জন্য।

w পার্টিশন তৈরি/আপডেট করার পরে সেভ করার জন্য 'w' (Write) প্রেস করতে হবে।

Command (m for help): p

Command (m for help): n

Partition number (5-128, default 5): 5 =====> (পার্টিশন নাম্বার)

First sector (49260544-62914526, default 49260544): {প্রেস এন্টার}

Last sector, +sectors or +size{K,M,G,T,P} (49260544-62914526, default 62914526): +300M ===>

৩০০ মেগাবাইটের পার্টিশন।

Created a new partition 5 of type 'Linux filesystem' and of size 300 MiB.

Command (m for help): p

Device	Start	End	Sectors	Size	Type
/dev/sda1	2048	1026047	1024000	500M	EFI System
/dev/sda2	1026048	42969087	41943040	20G	Linux filesystem
/dev/sda3	42969088	45066239	2097152	1G	Linux filesystem
/dev/sda4	45066240	49260543	4194304	2G	Linux swap
/dev/sda5	49260544	49874943	614400	300M	Linux filesystem

নোটঃ উপরে দেখতে পাচ্ছি '/dev/sda5' নামে '300M' সাইজের একটি পার্টিশন তৈরি হয়েছে।

Command (m for help): w

The partition table has been altered.

Syncing disks.

Create Extended and Logical Partition (MBR Method) :

```
[root@node1~]# fdisk /dev/sda
```

Partition type:

p primary (3 primary, 0 extended, 1 free)

e extended

Select (default e): e

Selected partition 4

First sector (12314624-16777215, default 12314624): {প্রেস এন্টার}

Using default value 12314624

Last sector, +sectors or +size{K,M,G} (12314624-16777215, default 16777215): {প্রেস এন্টার}

Using default value 16777215

Partition 4 of type Extended and of size 4.1 GiB is set

নোটঃ Extended পার্টিশনের ক্ষেত্রে 'First Sector' এবং 'Last Sector' উভয়ের ক্ষেত্রেই এন্টার প্রেস করতে হবে। তাহলে সম্পূর্ণ ফ্রি স্পেস থেকে লজিক্যাল পার্টিশন তৈরি করা যাবে।

Command (m for help): p

Command (m for help): w

নোটঃ একটা বিষয় খেয়াল রাখতে হবে, যখন extended পার্টিশন করা হয় তখন দুই বার এন্টার প্রেস করতে হবে।

Create New Logical Partition :

```
[root@node1~]# fdisk /dev/sda
```

Command (m for help): n

All primary partitions are in use

Adding logical partition 5

First sector (12316672-16777215, default 12316672): {প্রেস এন্টার}

Last .. +sectors or +size{K,M,G}. default 16777215): +300M ==> ৩০০ মেগাবাইটের পার্টিশন।

Partition 5 of type Linux and of size 300 MiB is set

Command (m for help): p

Command (m for help): w

নোটঃ লজিক্যাল পার্টিশন করার সময় প্রথমবার এন্টার এবং পরেরবার সাইজ উল্লেখ করতে হবে।

```
[root@serverX ~]# reboot (মাঝে মাঝে রিবুট করার দরকার হতে পারে)
```

Class Work :

* 500 MiB Another Partition (GPT)

Partition Format Command :

এখন দুইটি পার্টিশন (/dev/sda5, /dev/sda6) আলাদা আলাদা ভাবে (xfs, ext4) ফরম্যাট করা হবে।

```
[root@node1~]# mkfs.xfs /dev/sda5  
[root@node1~]# mkfs.ext4 /dev/sda6
```

Partition Mount :

ফরম্যাট করার পরে পার্টিশন মাউন্ট করতে হবে। যে ডিরেক্টরির আন্ডারে মাউন্ট করা হবে, আগে সেই ডিরেক্টরি তৈরি করে নিতে হবে।

```
[root@node1~]# mkdir /data1  
[root@node1~]# mkdir /data2  
[root@node1~]# lsblk  
  
[root@node1~]# mount /dev/sda5 /data1  
[root@node1~]# mount /dev/sda6 /data2
```

```
[root@node1~]# lsblk -f
```

```
[root@node1~]# reboot
```

```
[root@node1~]# lsblk -f
```

নোটঃ সিস্টেম রিবুট দেওয়ার পরে মাউন্ট পয়েন্ট চলে যাবে। কারণ, এটা অস্থায়ী মাউন্ট।

Permanent Mount :

```
[root@node1~]# lsblk -f
```

```
[root@node1~]# vim /etc/fstab
```

```
:set nu
```

```
UUID=1b42c7df-717a-420d-b054-81d5a594b5 /dataX xfs defaults 0 0
```

or

```
##### Add the following lines #####
```

```
/dev/sda5          /data1  xfs     defaults  0 0
/dev/sda6          /data2  ext4    defaults  0 0
```

```
(1)                2      3      4      5 6
```

- 1 - পার্টিশন নাম বা পার্টিশন ব্লক আইডি (blkid)
- 2 - মাউন্ট পয়েন্ট (পার্টিশন যে ডিরেক্টরির সাথে ম্যাপিং করা)।
- 3 - ফাইল সিস্টেম টাইপ (xfs, ext4, swap, nfs)
- 4 - Defaults সাথে Options হিসেবে quota, acl, ro, luks (এনক্রিপশন) ব্যবহার করা হয়।
- 5 - ডাম্প প্রোগামের জন্য।

*** ৫ নাম্বার ফিল্ডটি ব্যবহার করার হয় Dump প্রোগামের জন্য। যদি ভ্যালু '1' সেট করা হয়, তাহলে Dump ইউটিলিটি পার্টিশন ব্যাকআপের জন্য কাজ করবে। আর যদি ভ্যালু '0' সেট করা হয়, তাহলে Dump ইউটিলিটি পার্টিশন ব্যাকআপের জন্য কোনো কাজ করবে না।

Backup operation – the fifth field contains a 1 if the dump utility should back up a partition or a 0 if it shouldn't. If you never use the dump backup program, you can ignore this option.

- 6 - ফাইল সিস্টেম চেক অপশন।

*** ৬ নাম্বার ফিল্ডটি ব্যবহার করার হয় 'fsck' (File System Checking) প্রোগামের জন্য। সিস্টেম বুটিং এর সময় ডিভাইস/ পার্টিশনের Error চেক করে। এখানে '0' উল্লেখ করলে 'fsck' কাজ করবে না। আর যদি রুট পার্টিশনের ক্ষেত্রে '1' এবং অন্য পার্টিশনের ক্ষেত্রে '2' সেট করা হয়, তাহলে সিস্টেম বুটিং এর সময় ফাইল সিস্টেম চেক করবে।

File system check order – the sixth field specifies the order in which fsck checks the device/partition for errors at boot time. A 0 means that fsck should not check a file system.

Higher numbers represent the check order. The root partition should have a value of 1 , and all others that need to be checked should have a value of 2.

```
[root@node1 ~]# mount -a ;সিনটেক্স চেক এবং আপডেট করার জন্য।
```

```
[root@node1 ~]# mount
```

```
[root@node1 ~]# reboot
```

```
[root@node1 ~]# lsblk -f
```

Partition delete

*** Warning ****

=> Remove fstab entry (vim /etc/fstab)

=> Unmount

=> Then delete

```
[root@node1~]# vim /etc/fstab
```

'/etc/fstab' ফাইল ওপেন করার পরে, /dev/sda5 & /dev/sda6 পার্টিশনের এন্ট্রি টা '#' দিয়ে কমেন্ট করে দিতে হবে।

```
[root@node1~]# lsblk
```

```
[root@node1~]# umount /data1
```

```
[root@node1~]# umount /data2
```

```
[root@node1~]# lsblk
```

```
[root@node1~]# fdisk /dev/sda
```

Command (m for help): d
Partition number (1-6): 6

Partition 6 has been deleted.

Command (m for help): p
Command (m for help): d
Partition number (1-5):5

Partition 5 has been deleted.

Command (m for help):p
Command (m for help):w

Note: Before delete, you should unmount partition and delete fstab entry.

```
[root@node1~]# fdisk -l  
/dev/sda1  
/dev/sda2  
/dev/sda3  
/dev/sda4
```

Mount DVD/ISO :

```
[root@node1 ~]# mount /dev/sr0 /media  
[root@node1 ~]# cd /media  
[root@node1 media]# ls  
[root@node1 media]# cd Packages  
[root@node1 Packages]# ls  
[root@node1 Packages]# cd  
[root@node1 ~]# umount /media  
[root@node1 ~]#
```

Mount USB pendrive :

```
[root@node1 ~]# fdisk -l
```

Disk /dev/sdb: 32.2 GB, 32176472064 bytes

```
[root@node1 ~]# mount /dev/sdb1 /mnt
```

```
[root@node1 ~]# lsblk
```

```
[root@node1 ~]# umount /mnt
```

```
[root@node1 ~]# lsblk
```

```
** [root@node1 ~]# mount -t ntfs /dev/sdb1 /mnt ;(যদি NTFS ফাইল সিস্টেম হয়)
```

Homework Work :

** mount your NTFS based Pendrive under '/mnt' directory.

** For NTFS Pendrive you have to install 'ntfs-3g' application.

```
[root@node1 ~]# yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm
```

```
[root@node1 ~]# yum install ntfs-3g -y
```

Watch this video: <https://www.youtube.com/watch?v=bbbmM6CuEnE>

----- x -----

Homework

1. The node1 machine has unused space. On the unused space, create a 400 MiB GPT partition.

2. Format the 400 MiB partition with an XFS file system and persistently mount it to the /backup directory.

3. To verify your work, reboot the Node1 machine. Confirm that the system automatically mounts the partition onto the /backup directory.

Class - 11

LVM Introduction :

যখন লিনাক্স (CentOS/Red Hat) ইন্সটল করা হয় তখন '/boot/efi', /boot, '/' এবং 'Swap' এই চারটি পার্টিশন করা হয়। উক্ত পার্টিশন সমূহ 'Standard' পার্টিশন পদ্ধতি ব্যবহার করে তৈরি করা হয়েছিল। কিন্তু 'Standard' পার্টিশন পদ্ধতিতে পার্টিশন তৈরি করলে, ভবিষ্যতে পার্টিশনে কোনো স্পেস বাড়ানো বা কমানো যায় না। যেটা প্রডাকশন সার্ভারে বড় একটা সমস্যা হয়ে যায়। কারণ, ভবিষ্যতে সার্ভারের স্টোরেজ ফুরিয়ে গেলে স্টোরেজ বাড়ানো দরকার হতে পারে।

RHEL 9 Install Method :

- 1) Standard (এই পদ্ধতিতে ভবিষ্যতে কোনো পার্টিশনের স্পেস বাড়ানো বা কমানো যাবে না)
- 2) LVM Method (ভবিষ্যতে ইচ্ছামত স্পেস বাড়ানো বা কমানো যাবে। পাশাপাশি অতিরিক্ত Hard disk অ্যাড বা রিমুভ করাও যাবে)

```
/boot/efi - 200MiB
/boot     - 1024 MiB
/         - 20 GiB
Swap     - 2048 MiB
```

নোটঃ '/boot' বা '/boot/efi' পার্টিশন LVM হবে না।

```
[root@localhost ~]# lsblk
[root@localhost ~]# fdisk -l
```

এই ল্যাব ডেমোটি প্র্যাকটিস করার জন্য '/dev/sda5' '/dev/sda6' '/dev/sda7' নামে 250MB সাইজের তিনটি পার্টিশন তৈরি করতে হবে এবং সেগুলার পার্টিশন

আইডি পরিবর্তন করে Linux থেকে Linux LVM করতে হবে। নিচে এই ব্যাপারে বিস্তারিত কমান্ড সহ দেখানো হয়েছেঃ

```
[root@localhost ~]# fdisk /dev/sda
```

Command (m for help): n

Selected partition 5

First sector (12314624-16777215, default 12314624): {Press Enter}

Using default value 12314624

Last sector, +sectors or +size{K,M,G} (12314624-16777215, default 16777215): {Press Enter}

Using default value 16777215

Partition 4 of type Extended

Command (m for help): p

Command (m for help): n

First sector (902815744-976771071, default 902815744): [press Enter]

Last ... +size{K,M,G} (902815744-976771071, default 976771071): +250M

Command (m for help): p

Command (m for help): t

Command (m for help): Enter your partition no

Command (m for help): l

Command (m for help): 8e (MBR), 31/30 (GPT)

Command (m for help): p

Command (m for help): w

Note: Do 2 more partition for LVM

```
[root@localhost ~]# lsblk
```

Physical Volume Create :

```
[root@localhost ~]# pvcreate /dev/sda5
```

```
[root@localhost ~]# pvcreate /dev/sda6
```

```
[root@localhost ~]# pvcreate /dev/sda7
```

```
[root@localhost ~]# pvdisplay
```

```
[root@localhost ~]# pvscan
```

```
# yum install lvm2      (যদি কারো 'pvcreate' কমান্ড কাজ না করে)
```

Group Volume Create :

সবগুলো ফিজিক্যাল ভলিউম (PV) মিলে একটা গ্রুপ ভলিউম (VG) তৈরি করা হবে।

```
[root@localhost ~]# vgcreate vg1 /dev/sda5 /dev/sda6 /dev/sda7
[root@localhost ~]# vgdisplay
[root@localhost ~]# vgscan
```

ডিফল্ট ফিজিক্যাল এক্সটেন্ট (PE) সাইজ '4 MiB' এবং ভলিউম গ্রুপের সাইজ '744 MiB' অর্থাৎ এখানে মোট ১৮৬ টি '4 MiB' ফিজিক্যাল এক্সটেন্ট (PE) হয়েছে।

উল্লেখ্য যে, RHCSA ভেন্ডর এক্সামে ডিফল্ট ফিজিক্যাল এক্সটেন্ট (PE) সাইজ 4MiB পরিবর্তে '8MiB' করতে বলবে।

Note for Exam: `vgcreate -s 8M vg1 /dev/vda5 /dev/vda6 /dev/vda7`

Logical Volume Create :

```
[root@localhost ~]# lvcreate -n lv1 -L 400M vg1
[root@localhost ~]# lvcreate -n lv2 -L 200M vg1
```

```
[root@localhost ~]# lvscan
```

```
[root@localhost ~]# lvdisplay
```

```
[root@localhost ~]# mkfs.xfs /dev/vg1/lv1
[root@localhost ~]# mkdir /lvdata1
[root@localhost ~]# mount /dev/vg1/lv1 /lvdata1
[root@localhost ~]# df -HT
[root@localhost ~]#
```

```
[root@localhost ~]# mkfs.ext4 /dev/vg1/lv2
[root@localhost ~]# mkdir /lvdata2
[root@localhost ~]# mount /dev/vg1/lv2 /lvdata2
[root@localhost ~]# df -HT
[root@localhost ~]#
```

Parmanent Mount:

উপরের মাউন্টের পরে সিস্টেম রিবুট দিলে মাউন্ট পয়েন্ট চলে যাবে। এইজন্য স্থায়ী ভাবে মাউন্ট করার জন্য নিচের '/etc/fstab' ফাইলে এন্ট্রি দিতে হবেঃ

```
[root@localhost ~]# vim /etc/fstab
:set nu
```

```
/dev/vg1/lv1 /lvdata1 xfs defaults 0 0
/dev/vg1/lv2 /lvdata2 ext4 defaults 0 0
```

```
[root@localhost ~]# mount -a ; ফাইল সিনট্যাক্স চেক করার জন্য।
[root@localhost ~]# mount ; মাউন্ট হয়েছে কিনা দেখার জন্য।
[root@localhost ~]# reboot
```

VG Extended :

এই ল্যাবটি প্র্যাকটিস করার জন্য প্রথমে ভার্চুয়াল মেশিনে (Redhat/CentOS) -এ 5GiB সাইজের আরেকটি ডিস্ক অ্যাড করতে হবে। অ্যাড করার পরে নিচের কমান্ড দিয়ে দেখা যাবে, ডিস্কটি '/dev/sdb' নামে পেয়েছে। VG এক্সটেন্ড করার জন্য প্রথমে নতুন ডিস্কটি ফিজিক্যাল ভলিউমে (PV) তে কনভার্ট করতে হবে।

```
[root@localhost ~]# fdisk -l
```

```
=> fdisk /dev/sdb (5GB)
```

```
[root@localhost ~]# pvcreate /dev/sdb
[root@localhost ~]# pvscan
```

=> vgextend

```
[root@localhost ~]# vgextend vg1 /dev/sdb
```

```
[root@localhost ~]# vgdisplay
```

=> lvextend (xfs file system)

```
[root@localhost ~]# lvextend -L +100M /dev/vg1/lv1
```

```
[root@localhost ~]# df -HT | grep lvdata
```

```
[root@localhost ~]# lvscan
```

```
[root@localhost ~]# xfs_growfs /lvdata1
```

```
[root@localhost ~]# df -HT | grep lvdata
```

=> lvextend (ext4 file system)

```
[root@localhost ~]# lvextend -L +100M /dev/vg1/lv2
```

```
[root@localhost ~]# df -HT | grep lvdata2
```

```
[root@localhost ~]# lvscan
```

```
[root@localhost ~]# resize2fs /dev/vg1/lv2
```

```
[root@localhost ~]# df -HT | grep lvdata
```

Lv remove :

```
[root@localhost ~]# vim /etc/fstab ;remove fstab entery
```

```
[root@localhost ~]# umount /lvdata1
```

```
[root@localhost ~]# umount /lvdata2
```

```
[root@localhost ~]# lsblk
```

```
[root@localhost ~]# lvremove /dev/vg1/lv1
```

```
[root@localhost ~]# lvremove /dev/vg1/lv2
```

VG Remove :

```
[root@localhost ~]# vgremove vg1
```

PV Remove :

```
[root@localhost ~]# pvremove /dev/vda5
```

```
[root@localhost ~]# pvremove /dev/vda6
```

```
[root@localhost ~]# pvremove /dev/vda7
```

```
[root@localhost ~]# pvremove /dev/vda8
```

```
[root@localhost ~]# fdisk /dev/vda
```

```
Command (m for help): d
```

```
Partition number (1-8, default 8): 4
```

```
Command (m for help): w
```

```
[root@localhost ~]# fdisk -l
```

```
/dev/vda1
```

```
/dev/vda2
```

```
/dev/vda3
```

```
[root@localhost ~]# reboot
```

Working with Linux Swap Partition :

যখন আমরা RHEL 9 ইন্সটল করি তখন চারটি পার্টিশন তৈরি করি, '/boot/efi', '/boot', '/', এবং 'swap' পার্টিশন। এর মধ্যে '/boot', '/boot/efi' এবং '/' হচ্ছে ডাটা বা সিস্টেম পার্টিশন, যেখানে ডাটা বা অপারেটিং সিস্টেমের বিভিন্ন কম্পোনেন্ট জমা থাকে আর 'swap' পার্টিশন টা তৈরি করা হয়, সেকেন্ডারি মেমোরি হিসেবে। swap পার্টিশনে swap সিগনেচার থাকে বিধায় এটা মেমরির (RAM) মত ব্যবহার করা হয়। যখন CentOS/RedHat ইন্সটল করা হয়েছিল, তখন আমাদের প্রয়োজন অনুযায়ী 2048 MB (2 GiB) swap দিয়ে উল্লেখ করেছিলাম। এখন পরবর্তীতে যদি অতিরিক্ত swap স্পেস যোগ করার দরকার হয়, তাহলে আমরা নিচের পদ্ধতিতে যোগ করতে পারবো।

[root@serverX ~]# free -m ; মেমোরি (RAM) এবং swap মেমোরির তথ্য পাওয়া যাবে।

Minimum required Swap Memory :

RAM	Recommended Swap Space
2GB or Less	Twice of RAM
2GB to 8GB	Equal to RAM
8GB to 256GB	Min 4GB

Note: 4:1

```
[root@serverX ~]# fdisk -l
/dev/sda2 1026048 5220351 4194304 2G Linux swap
```

```
[root@serverX ~]# lsblk
```

```
[root@serverX ~]# fdisk /dev/sda
```

```
Command (m for help): p
```

```
Command (m for help): n
```

```
Partition number (5-128, default 5): 5
```

```
First sector (12314624-16777215, default 12314624): {Press Enter}
```

```
Using default value 12314624
```

```
Last sector, +sectors or +size{K,M,G} (12314624-16777215, default 16777215): +1G
```

```
Using default value 16777215
```

```
Partition 4 of type Extended and of size 1 GiB is set
```

```
Command (m for help): p
```

```
Command (m for help): t
```

```
Partition number (1-5, default 5): 5
```

```
de (type L to list all codes): L
```

Hex code (type L to list all codes): 82 (MBR), 19 (GPT)

Changed type of partition 'Linux filesystem' to 'Linux swap'.

Command (m for help): p

Command (m for help): w

Check Current Swap Status :

```
[root@serverX ~]# swapon --show ; পূর্বের swap পার্টিশন দেখার কমান্ড।
```

```
[root@serverX ~]# mkswap /dev/sda5 ; swap পার্টিশন ফরম্যাট করার কমান্ড।  
Setting up swapspace version 1, size = 511996 KiB  
no label, UUID=7906ac34-4e2a-4b06-b57d-79dd6c66399a
```

```
[root@serverX ~]# swapon /dev/sda5
```

```
[root@serverX ~]# swapon --show
```

```
[root@serverX ~]# free -m
```

total	used	free	shared	buffers	cached
Swap:	3072	0	1211		

```
[root@serverX ~]# swapon -s
```

Filename	Type	Size	Used	Priority
/dev/vda5	partition	716796	0	-1
/dev/vda3	partition	524284	0	-2

Permanent Mount :

```
[root@serverX ~]# lsblk -f
```

```
[root@serverX ~]# vim /etc/fstab
```

```
:set nu
```

```
/dev/sda5          none      swap defaults 0 0
```

```
[root@serverX ~]# mount -a  
[root@serverX ~]# swapoff -a  
[root@serverX ~]# swapon -a  
[root@serverX ~]# free -m
```

or

```
[root@serverX ~]# reboot  
[root@serverX ~]# free -m
```

===== The End =====

Home Work : -1

Install CentOS 9 Stream using following Parameters :

Storage Size: 30 GiB

Partition Type: MBR

Installation Type: LVM

Volume Group Name: rhcsa

MBR Partition - Standard (1024MB) - /boot

root Partition (/) - 20GiB (LVM)

Swap - - 2GiB (LVM)

Free Space - 7 GB

Homework 2: (LVM Based OS)

1. Extend your Volume group (rhcsa) with additional 5G Hdd
2. Extend your '/' partition 1G from current size.
3. Extend your 'swap' partition 1G from current size

Class - 12

Networking Address :

নেটওয়ার্কিং -এ আমরা দুই ধরনের অ্যাড্রেস ব্যবহার করে থাকিঃ

1) Logical Address - L3 Address, IP Address, Network Address

1) IPv4(32 Bits)

2) IPv6 (128 Bits)

2) Physical Address - L2 Address, MAC Address, Ethernet Address

IPv4 Address :

⌘ 32 Bits (4 Bytes) - 11010101010101010101010000101010111

⌘ 4 Parts 4 Octets (1st.2nd.3rd.4th) - 11010101.01010101.01010000.10101111

⌘ Separated by dot (.)

⌘ Decimal Format (0-9)

⌘ Example: 203.112.194.243

⌘ Total Address: IPv4 (32bits), $2^{32}=2^8 \cdot 2^8 \cdot 2^8 \cdot 2^8$ (256x256x256x256) $\Rightarrow 2^{32} = 4,294,967,296$

⌘ 8 bit এর মেক্সিমাম ডেসিম্যাল ভ্যালু = $2^8 = 256$ (0-255)

IPv4 Classification :

1) Class A - 0-127 (শুধুমাত্র প্রথম ঘর (1st Octet) এর জন্য প্রযোজ্য)

2) Class B - 128-191 ||

3) Class C - 192-223 ||

4) Class D - 224-239 <----- Multicast Application

5) Class E - 240-255 <----- Reserved/Experimental

বিঃ দ্রঃ কোন আইপি কোন ক্লাসে পড়ছে এটা আমরা বের করতে পারবো 1st Octet দেখে।

⌘ 203.112.194.243 - C

⌘ 224.0.0.10 - D

⌘ 191.224.242.20 - B

⌘ 135.253.104.25 - B

IPv4 (TCP/IP Communication) - Unicast & Broadcast IP

Class A: 0.0.0.0 - 127.255.255.255

Class B: 128.0.0.0 - 191.255.255.255

Class C: 192.0.0.0 - 223.255.255.255

Types of IPv4 Address :

- 1) Public IP (Globally Routable & Paid, lease, Real)
- 2) Private IP (Non Routable & Free)
- 3) Localhost IP - 127.0.0.0/8 - লুপব্যাক আইপি (IPv4) অ্যাড্রেস। প্রতিটি সিস্টেমে এটা ডিফল্ট থাকে।
- 4) Reserved IP - 169.254.0.1-169.254.255.254 (APIPA) - যখন স্ট্যাটিক কনফিগার করা থাকে না, এবং DHCP থাকে না তখন এটা সেট হয়।
- 5) Documentation - 198.51.100.0/24

Private Address Range :

Class A: 10.0.0.0 - 10.255.255.255

Class B: 172.16.0.0 - 172.31.255.255 <---- RHCSA Exam Block

Class C: 192.168.0.0 - 192.168.255.255

Linux Network Management :

উইন্ডোজ অপারেটিং সিস্টেমে যখন আইপি বসানো হয়, তখন নিচের নামের মত বিভিন্ন ইন্টারফেস দেখতে পাই।

Control Panel > Network and Internet > View Network Connections

* Windows NIC1: Ethernet1

* Windows NIC2: Ethernet2

```
[root@node1 ~]# ip link
```

* Interface Name = lo,en,wl,vir (lo=loopback, en=ethernet, wl=wireless, vir-br=Virtual Bridge)

* Status = State UP

* Link Speed = 1000

* MAC Address = Ether

* MTU = 1500 (সর্বোচ্চ কত সাইজের ফ্রেম সেন্ড/রিসিভ করতে পারে)

```
[root@node1 ~]# ip addr show ; CMD or GUI
```

inet = IPv4 Address

inet6 = IPv6 Address

brd = broadcast Address

```
[root@hostX ~]# ifconfig ; (active)
```

```
[root@hostX ~]# ifconfig ens160 ; sepecific LAN (br0, eth0, ens33, enp2s0, ens192)
```

mtu = maximum transmission unit (1500)

inet = IPv4 Address

inet6 = IPv6 Address (Link Local Address)

netmask = Subnet mask

broadcast = Broadcast Address

ether = MAC Address

Speed = 1000

Note: `ifconfig` কমান্ডের ক্ষেত্রে 'net-tools' নামের প্যাকেজ (RPM) ইন্সটল থাকতে হবে।

Linux (RHEL/CentOS-9)

Ethernet : enp, eno, ens (start with 'en')

Legacy: eth0

Virtual/sub Interface: enp2s0:1 or ens160:1, ens33:1, eth0:1

loopback: lo

Bridge : br0, vbri0

Wireless: wl

Check Physical Connectivity :

```
[root@localhost ~]# ip link  
state "UP" or State "Down"
```

```
[root@localhost ~]# ethtool enxxx (interface name)
```

Host Name Configure (Desktop Machine)

```
[root@localhost ~]# hostname  
localhost.localdomain
```

```
[root@localhost ~]# hostnamectl status
```

```
[root@localhost ~]# hostnamectl set-hostname node1.example.com
```

```
[root@localhost ~]# exec bash
```

```
[root@node2 ~]# cat /etc/hostname
node1.example.com
```

```
[root@node2 ~]# hostname
```

Network Card Enable and Disable :

```
[root@node2 ~]# ip link                               ; প্রথমে ইন্টারফেসের নাম দেখে নিতে হবে।
[root@node2 ~]# ip link set ens160 down               ; নেটওয়ার্ক কার্ড ডিজবেল করা হয়েছে।
[root@node2 ~]# ip link                               (Showing DOWN)
[root@node2 ~]# ip link set ens160 up                 ; নেটওয়ার্ক কার্ড এনাবেল করা হয়েছে।
```

গেটওয়ে দেখার জন্য নিচের কমান্ডঃ

```
[root@node1 ~]# route -n
Destination Gateway Genmask Flags Metric Ref Use Iface
0.0.0.0 192.168.43.1 0.0.0.0 UG 100 0 0 ens160
192.168.43.0 0.0.0.0 255.255.255.0 U 100 0 0 ens160
```

নোটঃ যদি কোথাও 'route -n' কমান্ড কাজ না করে তাহলে, 'ip route' কমান্ড ব্যবহার করতে হবে।

```
[root@node1 ~]# ping 172.25.11.1
64 bytes from 172.25.11.1 : icmp_seq=1 ttl=64 time=0.451 ms
64 bytes from 172.25.11.1 : icmp_seq=2 ttl=64 time=0.317 ms
C^
```

নোটঃ 'Ctrl + C' প্রেস করে 'ping' বন্ধ করা যাবে।

নির্দিষ্ট নাম্বার 'ping' দিতে চাইলে নিচের কমান্ডের মাধ্যমে দিতে হবেঃ

```
[root@node1 ~]# ping -c 4 172.25.11.1
```

Ping to external Network :

```
[root@node1 ~]# ping 8.8.8.8
```

DNS দেখার জন্য নিচের কমান্ডঃ

```
[root@node1 ~]# cat /etc/resolv.conf
nameserver 172.25.11.254
nameserver 4.2.2.2
```

```
[root@node1 ~]# ping www.google.com
```

New Default GW Add (Optional) :

```
[root@node1 ~]# route add default gw 172.25.11.254
[root@node1 ~]# route -n
[root@node1 ~]# ping 8.8.8.8
```

```
[root@node1 ~]# route del default gw 172.25.11.254
[root@node1 ~]# route -n
[root@node1 ~]# ping 8.8.8.8
```

Local Host Entry :

```
[root@node2 ~]# vim /etc/hosts
```

IP Client Configure

আমরা দুইভাবে আইপি অ্যাড্রেস কনফিগার করতে পারিঃ

- ⌘ Static (Manually)
- ⌘ dhcp : Autmatically ip configure

Static IP Configure :

বিভিন্ন ইউটিলিটি (Tools) ব্যবহার করে স্ট্যাটিক আইপি কনফিগার করা যাবে।

- 1) nmtui - NetworkManager Text user interface
- 2) nmcli - NetworkManager Command line Interface
- 3) GUI - Graphical User Interface
- 4) vim - Using Editor Tools
- 5) yml - using YML File

লিনাক্সে নেটওয়ার্ক সম্পর্কিত সার্ভিস 'NetworkManager' ডেমন (Daemon) দিয়ে ম্যানেজ করা হয়। সুতরাং, আইপি বসানোর আগে 'NetworkManager' সার্ভিসটি ঠিকমত রান করতেছে কিনা সেটা দেখে নিতে হবে। রান না করলে নিচের কমান্ড দিয়ে এটাকে স্টার্ট (start) এবং এনাবেল (enable) করে নিতে হবে।

```
[root@node1 ~]# systemctl status NetworkManager.service
```

```
[root@node1 ~]# systemctl restart NetworkManager.service
```

```
[root@node1 ~]# systemctl enable NetworkManager.service
```

```
[root@node1 ~]# nmtui ; {Press Enter}
=> edit connection
=> select 'Interface' => 'Edit'
=> Change Interface 'Automatic' to 'Manual' => Change 'Show'
=> Add
=> Addressess: 172.25.11.X/24
    Gateway: - 172.25.11.1
    DNS1 - 8.8.8.8
    DNS2 - 9.9.9.9
```

[x] Automatically Connect

```
=> Now Press 'ok'
=> Back
=> Activate Connection (Press two times, for deactivate and activate)
```

```
[root@node1 ~]# ifconfig
```

Working with NMCLI:

```
[root@node1 ~]# nmcli connection show
```

NAME	UUID	TYPE	DEVICE
ens160	5fb06bd0-0bb0-7ffb	802-3-ethernet	ens160

```
[root@node1 ~]# nmcli con modify 'ens160' ipv4.addresses 172.25.11.X/24
```

```
[root@node1 ~]# nmcli con modify 'ens160' ipv4.gateway 172.25.11.1
```

```
[root@node1 ~]# nmcli connection modify ens192 ipv4.dns 8.8.8.8
```

```
[root@node1 ~]# nmcli con modify 'ens160' ipv4.method manual
```

```
[root@node1 ~]# nmcli con modify ens160 autoconnect yes
```

```
[root@node1 ~]# nmcli con down 'ens160'
```

```
[root@node1 ~]# nmcli con up 'ens160'
```

```
NAME = 'ens160'
```

```
[root@node1 ~]# ifconfig 'ens160'
```

Static IPv6 Configuration:

Description	IPv4	IPv6
-----	----	----
No bits	32	128
Part	4	8
Each Part	Octet	Hextet
Format	Dot (.)	(:)
Number	Decimal (0-9)	Hexa (0-9, A-F)
Class	5 (A,B,C,D,E)	N/A
Example	192.168.11.10	2001:db8::abcd:12c:0:1

Types of IPv4

- ⌘ Loopback (127.0.0.0/8) - প্রতিটি হোস্টের নিজস্ব আইপি
- ⌘ Private IP (10.0.0.0/8, 172.16.0.0 - 172.31.255.255, 192.168.0.0-192.168.255.255)
- ⌘ Public IP (except private, loopback, reserved block)
- ⌘ Reserved (169.254.) - APIPA
- ⌘ Multicast IP (224.0.0.0-239.255.255.255)
- ⌘ All IP - 0.0.0.0

Types of IPv6 :

- ⌘ ::1/128 - loopback (Similar as 127.0.0.1)
- ⌘ fd00 - Unique local address (same as IPv4 Private)
- ⌘ 2000 - Global unicast address (same as public IP). For APNIC 2400:
- ⌘ fe80 - Link local address (similar as 169.254....)
- ⌘ ff00 - Multicast (Similar as 224.0.0.0)
- ⌘ :: - all IP (similar as 0.0.0.0)

```
# ping 127.0.0.1
# ping ::1
```

Configure IPv6 to ens160 Interface :

```
[root@node1 ~]# nmcli connection show
```

NAME	UUID	TYPE	DEVICE
ens160	5fb06bd0-7ffb-45f1-65f3e03	802-3-ethernet	ens160

```
[root@node1 ~]# nmcli con modify "ens160" ipv6.addresses fd00:db08:abcd::1234:X/64
[root@node1 ~]# nmcli con modify "ens160" ipv6.gateway fd00:db08:abcd::1234:1
[root@node1 ~]# nmcli con modify "ens160" ipv6.dns fd00:db08:abcd::1234:254
[root@node1 ~]# nmcli con modify "ens160" ipv6.method manual
[root@node1 ~]# nmcli con modify ens160 autoconnect yes
[root@node1 ~]# nmcli con up "ens160"
```

```
[root@node1 ~]# ifconfig
[root@node1 ~]# ping fd00:db08:abcd::1234:5
```

===== Thank you =====

Class - 13

RHEL/CentOS 9 Package Management :

Windows : .exe, .bin, .msi
Apple (MacOS) : .pkg, .dmg
Android : .apk
RHEL/Fedora/CentOS/Oracle : .rpm
Ubuntu,Mint,Debian,Kali : .deb
Script/Common Distribution : .tar.gz, .tar.bz2, .tar

RPM Source :

আমরা যে সকল সোর্স থেকে প্যাকেজ বা RPM সংগ্রহ করতে পারি সেই সকল লোকেশন গুলো নিচে দেওয়া হয়েছেঃ

- ⌘ ISO/DVD
- ⌘ Internet (Public/Private Repo)
- ⌘ Third Party Website (Like Teamviewer)

Note : <https://www.centos.org/centos-stream>

Linux OS Kernel Info:

[root@node1 ~]# uname -r ; কার্নেল ভার্সন দেখার জন্য।
[root@node1 ~]# cat /etc/redhat-release ; অপারেটিং সিস্টেম ভার্সন দেখার জন্য।

Lab RPM Location : (HomeLab)

- ⌘ Insert DVD or Attach ISO file
- ⌘ Explore DVD or ISO
- ⌘ AppStream and BaseOS (RPM Container) নামে ২টি আলাদা আলাদা ডিরেক্টরি থাকে।
যার মধ্যে সকল RPM পাওয়া যাবে।

Class Room RPM Location :

- ⌘ Browse <http://172.25.11.254/rpm>
- ⌘ Download all RPMs
- ⌘ Explore download location

Mount ISO file from Desktop/CentOS.....

```
[root@node1 ~]# cd Desktop
[root@node1 Desktop]# mount Cen.....(tab) /media
[root@node1 Desktop]# lsblk
[root@node1 Desktop]# cd /media
[root@node1 media]# ls
```

Mount DVD/ISO on /media Directory: (DVD থেকে RPM ব্যবহার করার জন্য)

```
[root@node1 ~]# lsblk
sr0    11:0    1 9.4G 0 rom  /run/media/root/RHEL-9-42-0-BaseOS-x86_64
```

নোটঃ DVD/ISO আমাদের সিস্টেমে 'sr0' নামে পাবে। উপরের 'lsblk' কমান্ড দিয়ে আমরা দেখতে পারি।

```
[root@node1 ~]# mount /dev/sr0 /media
mount: /media: WARNING: device write-protected, mounted read-only.
```

```
[root@node1 ~]# lsblk
[root@node1 ~]# cd /media
sr0    11:0    1 9.4G 0 rom  /media
```

```
[root@node1 media]# ls
AppStream BaseOS images ...
```

```
[root@node1 media]# cd AppStream
```

```
[root@node1 AppStream]# ls
[root@node1 AppStream]# cd Packages
[root@node1 Packages]# ls
```

```
vsftpd-3.0.2-9.el8.x86_64.rpm
zsh.....
ftp .....
gnote-.....
wireshark-.....
```

RPM Format:

```
vsftpd-3.0.3-28.el9.x86_64.rpm
 1    2    3    4    5
```

- 1 - RPM name - এটা ইউনিক হবে। অন্য কোনো RPM এর নামের সাথে মিলবে না।
- 2 - RPM version - এটা কিছুদিন পর পর নতুন রিলিজ আসবে, তখন নতুন ভার্সন দেখা যাবে।
- 3 - Linux platform - 28.el9 = Enterprise Linux 9, el8 = Enterprise Linux 8
- 4 - Processor Architecture - 32bit = i686, 64 bits = x86_64, যে কোনো আর্কিটেকচার = noarch
- 5 - .rpm (extention) - RPM এক্সটেনশন। .rpm শুধুমাত্র RedHat, CentOS, Oracle, Fedora, Rocky লিনাক্সে কাজ করবে।

Common Linux RPM for Linux Server :

যখন লিনাক্স (CentOS/RHEL) ইন্সটল করা হয়, তখন প্রায় ১৪০০+ প্যাকেজ ইন্সটল হয়। কিন্তু একটা CentOS/RHEL ISO/DVD তে প্রায় ০৭ (সাত) হাজারের উপরে RPM থাকে। আমরা চাইলে পরবর্তীতে প্রয়োজন অনুযায়ী RPM ইন্সটল করতে পারি। আর যদি মিনিমাল মোডে ইন্সটল করা হয় তখন ৩০০-৪০০ প্যাকেজ বা RPM ইন্সটল হয়।

- 1) SSH Service - openssh-server, openssh-client, openssh
- 2) Telnet - telnet, telnet-server
- 3) CronJobs - crontab
- 4) Proxy Server - squid
- 5) DHCP Server - dhcp-server, dhcp-client

- 6) FTP Server - vsftpd (server), ftp (client)
- 7) Mail Server - postfix (MTA), dovecot(imap/pop3), roundcube (webmail)
- 8) NFS Server - nfs, nfs-utils
- 9) Samba File Server - samba, smbclient, cifs-utils
- 10) Web Server - httpd, mod_ssl, nginx
- 11) DNS Server - bind, bind-utils, unbound

- 12) Monitoring - cacti, snmp, mrtg, nagios, zabbix, librenms, opennms, wireshark, nmap
- 13) Time server - chrony, ntp
- 14) Virtualization - libvirt, qemu-kvm, virt-manager
- 15) Database - mysql, mariadb, postgresql, mongodb
- 16) Container - docker, podman
- 17) Programming - php, ruby, python3

Other: firefox, thunderbird, tigervnc, libreoffice, nmap, firewallld, tcpdump, wireshark

RPM Install/Remove :

- ⌘ Manually (RPM) - এক্ষেত্রে উইন্ডোজ অপারেটিং সিস্টেমের মত 'rpm' সংগ্রহ করে ইন্সটল করা যাবে।
- ⌘ Automatically (YUM/DNF) - সিস্টেম নিজে থেকেই লোকাল/প্রাইভেট/পাবলিক রিপোজিটরি থেকে 'rpm' সংগ্রহ করে ইন্সটল করবে।

Working with RPM (Manually)

- 1) Query : -qa (Query All) - RPM ইন্সটল আছে কিনা চেক করার জন্য।
- 2) Package Info: -qpi - RPM এর বিস্তারিত তথ্য (size, version, signature, license) জানার জন্য।
- 3) Install: -ivh - RPM ইন্সটল করার জন্য 'ivh' ব্যবহার করা হয়।
- 4) Upgrade: -U (upgrade) - পুরাতন ভার্সন থেকে নতুন ভার্সনে আপগ্রেড করার জন্য।
- 5) RPM info: -qf - কমান্ড থেকে RPM এর নাম জানার জন্য ব্যবহার করা হয়।
- 6) Config file: -qc (Query Configuration) - কনফিগারেশন টাইপের ফাইল খুঁজে বের করার জন্য।
- 7) Remove: -e (erase) - RPM uninstall (রিমুভ) করার জন্য ব্যবহার হয়।

RPM Query :

[root@node1 Packages]# rpm -qa সিস্টেমে যত RPMs (প্যাকেজ) ইন্সটল করা আছে সেই লিস্ট দেখা যাবে।

[root@node1 Packages]# rpm -qa | wc -l ; মোট কতগুলো প্যাকেজ ইন্সটল আছে, সেটা জানা যাবে।

[root@node1 Packages]# rpm -qa | grep cups ; 'cups' নামে কোনো rpm আছে কিনা, জানার জন্য।

[root@node1 Packages]# rpm -qa | grep vsftpd ; 'vsftpd' নামে কোনো rpm আছে কিনা, জানার জন্য।

[root@node1 Packages]# rpm -qa | grep firefox ; 'firefox' নামে কোনো rpm আছে কিনা, জানার জন্য।

Package Information :

কোনো RPM বা প্যাকেজ ইন্সটলের আগে সেটার বিস্তারিত তথ্য (size, version, signature, license) চেক করে নেওয়া যেতে পারেঃ

```
[root@node1 Packages]# rpm -qpi vsftpd.....x86_64.rpm (press tab)
```

RPM Install (Manual) :

প্যাকেজ (RPM) ইন্সটল করার জন্য নিচের কমান্ড ব্যবহার করতে হবে। উল্লেখ্য এখানে 'RPM' এর পূর্ণ নাম উল্লেখ করতে হবে।

```
[root@node1 Packages]# rpm -qa | grep vsftpd
```

```
[root@node1 Packages]# ls
```

```
[root@node1 Packages]# rpm -ivh vsftpd-3.0.2-9.el8.x86_64.rpm ; সম্পূর্ণ rpm এর নাম ট্যাব প্রেস করে নেওয়া যাবে।
```

i - install

v - verbosely (Installation Process)

h - hash (#####)

```
[root@node1 Packages]# rpm -qa | grep vsftpd
```

Class Work-1: Install 'zsh' RPM package.

RPM Check for specific command/application:

```
[root@node1 Packages]# ifconfig
```

```
[root@node1 Packages]# which ifconfig
```

```
/sbin/ifconfig
```

```
[root@node1 Packages]# rpm -qf /sbin/ifconfig
```

```
net-tools-2.0-0.51.20160904git.el8.x86_64
```

Class Work-2: Find out 'useradd' command's RPM name.

Configuration files for specific RPM :

কোনো প্যাকেজের 'RPMs' এর কনফিগারেশন সম্পর্কিত ফাইলের তথ্য বের করতে চাইলে নিচের কমান্ডঃ

```
[root@node1 Packages]# rpm -qc vsftpd
```

```
[root@node1 Packages]# rpm -qc openssh-server
```

RPM Remove :

প্যাকেজ (RPM) রিমুভ (uninstall) করার জন্য নিচের কমান্ড ব্যবহার করতে হবে। উল্লেখ্য এখানে 'RPM' এর পূর্ণ নাম উল্লেখ করার দরকার নাই।

```
[root@node1 ~]# rpm -qa | grep vsftpd
```

```
[root@node1 ~]# rpm -e vsftpd
```

```
[root@node1 ~]# rpm -qa | grep vsftpd
```

Dependency Issue :

কিছু কিছু প্যাকেজ আছে যেগুলো একটি আরেকটির উপরে নির্ভরশীল । অর্থাৎ উক্ত প্যাকেজের সকল ডিপেন্ডেন্সি (Dependencis) ছাড়া ইন্সটল করা সম্ভব না। যেমনঃ

```
[root@node1 Packages]# rpm -ivh wireshark-3 ... press Tab
```

error: Failed dependencies:

উপরের 'wireshark' প্যাকেজ ইন্সটল করার সময় 'error' দিবে, যে এটার প্রয়োজনীয় ডিপেন্ডেন্সি (Dependencies) ইন্সটল নাই। এক্ষেত্রে আমরা ম্যানুয়াল ইন্সটলেশনের বদলে অটোম্যাটিক ইন্সটল পদ্ধতি (YUM/DNF) ব্যবহার করবো। এক্ষেত্রে সিস্টেম অবশ্যই লোকাল/প্রাইভেট/পাবলিক রিপোজিটরির সাথে যুক্ত থাকতে হবে।

Connectivity Public YUM/DNF: (Use CentOS)

```
[root@node1 ~]# ping www.centos.org
```

```
[root@node1 ~]# yum repolist
```

```
[root@node1 ~]# yum info vsftpd
```

```
[root@node1 ~]# yum install vsftpd
```

```
[root@node1 ~]# yum install php
```

```
[root@node1 ~]# yum install wget
```

```
[root@node1 ~]# yum install ftp telnet mysql zsh
```

```
[root@node1 ~]# rpm -qa | grep php
[root@node1 ~]# rpm -qa | grep vsftpd
```

```
[root@node1 ~]# yum remove vsftpd
[root@node1 ~]# yum remove php
```

```
[root@node1 ~]# rpm -qa | grep php
[root@node1 ~]# rpm -qa | grep vsftpd
```

EPEL (Extra Package for Enterprise Linux) :

যে সকল প্যাকেজ সমূহ (RPMs) CentOS/RedHat/Fedora এর ডিফল্ট রিপোজিটরি পাওয়া যায় না এবং কিন্তু Fedora ভিত্তিক ডিস্ট্রিবিউশনে ব্যবহার করা হয়, সেই সকল প্যাকেজ সমূহ EPEL (Extra Package for Enterprise Linux) রিপোজিটরিতে থেকে ইন্সটল করা যাবে।

```
⌘ librenms
⌘ cacti
⌘ nagios
⌘ filezilla
```

```
[root@node1 ~]# yum install epel-release -y
[root@node1 ~]# yum info cacti
```

Create YUM Local repository :

```
[root@node1 ~] # lsblk
[root@node1 ~] # cd /media
[root@node1 media]# ls
```

AppStream BaseOS

নোটঃ আমরা চাইলে '/media' বা অন্য কোনো ডিরেক্টরির ভিতরে মাউন্ট করতে পারি।

Configure YUM Local Repo (DVD) :

```
[root@node1 mnt]# cd /etc/yum.repos.d/
[root@node1 yum.repos.d]# ls
[root@node1 yum.repos.d]# mkdir backup
[root@node1 yum.repos.d]# mv *.repo backup
[root@node1 yum.repos.d]# ls
backup
[root@node1 yum.repos.d]# vim local.repo
```

```
[BaseOS]
name=dvd local Base OS repo
baseurl=file:///media/BaseOS
enabled=1
gpgcheck=0
```

```
[AppStream]
name=dvd local AppStream repo
baseurl=file:///media/AppStream
enabled=1
gpgcheck=0
```

Exam Tips :

```
baseurl=http://192.168.10.254/pub/BaseOS
```

baseurl=http://instructor.example.com/pub/BaseOS

baseurl=ftp://192.168.10.254/pub/BaseOS

[root@node1 ~]# yum clean all

[root@node1 ~]# yum repolist

[root@node1 ~]# yum info wireshark

[root@node1 ~]# yum install wireshark

[root@node1 ~]# yum remove wireshark

Homework: Install 'VLC' and 'google-chrome' using CentOS Stream

===== Thank You =====

Class - 14

What is OpenSSH ?

'SSH' এর পূর্ণ অর্থ হচ্ছে 'Secure Shell (SSH)', 'SSH' ক্লায়েন্ট-সার্ভার আর্কিটেকচার পদ্ধতিতে কাজ করে। 'SSH' অ্যাপ্লিকেশন বা প্রটোকল ব্যবহার করে আমরা রিমোট ডিভাইস রাউটার, ফায়ারওয়াল, সার্ভারে লগইন করতে পারি। এটা 'Telnet' অ্যাপ্লিকেশন/প্রটোকলের মত রিমোট অ্যাডমিনিস্ট্রেশনের জন্য ব্যবহার হয়, কিন্তু 'Telnet' সেশন সিকিউর হয় না, অর্থাৎ 'Telnet' প্রটোকল 'Plain Text' পদ্ধতিতে সেশন তৈরি করে, যার ফলে রিমোট ডিভাইসের (Server/Router/Firewall) লগইন ইউজার নাম এবং পাসওয়ার্ড সহজে হ্যাক হতে পারে। অপরদিকে 'SSH' প্রটোকল সার্ভার এবং ক্লায়েন্টের মধ্যে সিকিউর (Encrypted) সেশন তৈরি করে, ফলে রিমোট সার্ভার/ডিভাইস এবং ক্লায়েন্টের মাঝে ইউজার নাম, পাসওয়ার্ড এবং ডাটা ট্রান্সফার এনক্রিপ্টেড (Cipher Text) পদ্ধতিতে হয়ে থাকে। 'SSH' প্রটোকলের ডিফল্ট পোর্ট নাম্বার '22' এবং এটা একটি 'TCP Protocol' ভিত্তিক সার্ভিস, তবে এটা পোর্ট নাম্বার পরিবর্তন করে কাস্টম (Custom) পোর্ট ব্যবহার করা যায়।

RHEL/CentOS/Rocky 9 অপারেটিং সিস্টেমে ডিফল্ট ভাবে 'SSH' এনাবেল থাকে, কিন্তু 'root' ইউজার এক্সেস থাকে না। সিকিউরিটি ইস্যুর কারণে ডিফল্ট কনফিগারেশন (Default Port, Root Login, Password-based authentication) পরিবর্তন করে নিতে হবে। যার ফলে নির্দিষ্ট ইউজার বা হোস্ট ছাড়া অন্য কেউ লগইন করতে পারবে না।

Module Objectives :

- ⌘ 'RHEL/CentOS Stream 9' -এ 'SSH' প্রটোকল/অ্যাপ্লিকেশনের ব্যবহার।
- ⌘ 'SSH' এর মাধ্যমে লিনাক্স এবং উইন্ডোজ ক্লায়েন্ট থেকে লিনাক্স (RHEL/CentOS 9) সার্ভারে লগইন করা।
- ⌘ 'SSH' এর 'password-based' এবং 'key-based' অথেনটিকেশন সম্পর্কে বিস্তারিত আলোচনা করা।
- ⌘ 'SSH' এর Private এবং Public key সম্পর্কে বিস্তারিত আলোচনা।
- ⌘ 'SSH key pair' তৈরি করা এবং রিমোট সার্ভারে পাঠানো।
- ⌘ 'SSH' দিয়ে লিনাক্স সার্ভারে 'root' ইউজারের সরাসরি এক্সেস বন্ধ করা।
- ⌘ 'Password-based' অথেনটিকেশন বন্ধ করা এবং শুধুমাত্র 'key-based' অথেনটিকেশন চালু রাখা।
- ⌘ 'SSH' এর ডিফল্ট পোর্ট (22) পরিবর্তন করে কাস্টম পোর্ট ব্যবহার করে সিকিউরিটি প্রদান করা।

Reference Tables :

- ⌘ Port no: 22 (default), কিন্তু পরিবর্তন করা যাবে।
- ⌘ Protocol: TCP
- ⌘ Package (RPMs): openssh-server (node2), openssh-client(node1)
- ⌘ Configuration file: /etc/ssh/sshd_config (node2)
- ⌘ Daemon/Service: sshd

Step by step configuration :

Step 01 : Host Name Change :

```
[root@localhost ~]# hostnamectl set-hostname node1.example.com
[root@localhost ~]# bash
[root@nodeX ~]#
```

Step 02 : Check IP Address Configure :

```
[root@nodeX ~]# ifconfig
```

Step 03 : RPM Query, install (node2)

```
[root@node1 ~]# rpm -qa | grep openssh-server
openssh-server-8.7p1-28.el9.x86_64
```

Step 04 : Service Restart

```
[root@node2 ~]# systemctl start sshd.service
[root@node2 ~]# systemctl enable sshd.service
[root@node2 ~]# systemctl status sshd.service
[root@node2 ~]# netstat -ntlp | grep ssh
```

Step 05 : Create SSH Login User and Set Password :

```
[root@node2 ~]# useradd admin1
[root@node2 ~]# useradd user1
[root@node2 ~]# passwd admin1
[root@node2 ~]# passwd user1
```

SSH Client :

- ⌘ Linux Client (default installed)
- ⌘ Windows Client (putty, ssh client)
 - > putty > ssh > ip > port (22)
- ⌘ Move to linux client machine
- ⌘ ping 172.25.11.254 (ssh server)
- ⌘ Command syntax: SSH <remote host IP/Name>

SSH Login with Root User (SSH Linux Client) :

```
[root@node1 ~]# ssh admin1@172.25.11.X
```

```
Are you sure you want to continue connecting (yes/no)? yes
admin1@172.25.11.X's password: ***** (remote PC)
```

```
[admin1@node2 ~]$ who
[admin1@node2 ~]$ su -
Password: ****
```

```
[root@node2 ~]#
```

SSH Authentication :

ডিফল্ট ভাবে 'RHEL 9' -এ 'root' ইউজারের লগিন এক্সেস থাকে না। তবে চাইলে নির্দিষ্ট কিছু ইউজারকে 'ssh' ব্যবহার করার প্রিবিলেজ দিতে পারি।

```
[root@node2 ~]# vim /etc/ssh/sshd_config
```

```
:set nu
```

```
37 # Authentication:
```

```
38 PermitRootLogin no (yes) <----- যদি 'root' ইউজারের এক্সেস দরকার হয়।
```

```
39 AllowUsers admin1
```

```
[root@node2 ~]# systemctl reload sshd
```

```
[root@node2 ~]# exit
```

এখন host/node2 মেশিন থেকে টেস্ট করা হবে।

```
[root@hostX ~]# ssh root@172.25.11.node1 - (Permission denied) - যদি 'PermitRootLogin no'  
থাকে
```

```
[root@hostX ~]# ssh user1@172.25.11.node1 - (Permission denied)
```

```
[root@hostX ~]# ssh admin1@172.25.11.node1 - allowed
```

Type of SSH Authentication :

⌘ Password authentication

⌘ Public key authentication (main authentication method)

⌘ 2FA Authentication

Password Authentication পদ্ধতিতে কিছু সিকিউরিটি ইস্যু আছে। যেমন বিভিন্ন ভাবে পাসওয়ার্ড ব্র্যাক হতে পারে। আরো কিছু সমস্যা নিচে উল্লেখ করা হয়েছেঃ

⌘ Dictionary Attack

⌘ Brutforce attack

⌘ MiM Attack

⌘ Password lost

⌘ Password Forget

⌘ Password Manage (10+ server)

Key Based Authentication :

- ⌘ Public key: প্রত্যেক রিমোট সিস্টেমের কাছে থাকবে এটা।
- ⌘ Private key: প্রত্যেক লোকাল সিস্টেম (ssh client) এর কাছে থাকবে এটা।

KeyPair তৈরি করার জন্য যে সকল এলগরিদম সমূহ ব্যবহার করা হয়ঃ

- ⌘ RSA
- ⌘ DSA (deprecated)
- ⌘ ECDSA
- ⌘ ed25519

*** এর মধ্যে RSA (3072 bit) অধিক সিকিউওর এবং বেশি ব্যবহার করা হয়।

Password Less ssh login :

```
[root@node1 ~]# cd
[root@node1 ~]# ls
[root@node1 ~]# ls -a
[root@node1 ~]# cd .ssh ; SSH সম্পর্কিত
[root@node1 .ssh]# ls
known_host (list of known hosts)

[root@node1 .ssh]# ssh-keygen ; (Enter 3 Times)
id_rsa.pub id_rsa

[root@node1 .ssh]# cat id_rsa
[root@node1.ssh]# cat id_rsa.pub

[root@node1 .ssh]# ssh-copy-id admin1@172.25.11.X (nod2)
[root@node1 .ssh]# ssh admin1@172.25.11.X
```

```
[admin1@node2 ~]$ cd .ssh
```

```
[admin1@node2 .ssh]$ ls  
authorized_keys
```

```
[admin1@node1 .ssh]$ cat authorized_keys ; same as public key of hostX
```

```
[admin1@node1 .ssh]$ exit
```

Change Default Port :

SSH এর ডিফল্ট পোর্ট TCP '22' তবে, সিকিউরিটি ইস্যুর কারণে, ডিফল্ট পোর্ট পরিবর্তন করে কাস্টম পোর্ট ব্যবহার করা উচিত।

```
[root@node1 ~]# netstat -ntlp | grep ssh
```

```
[root@node1 ~]# ss -ltn
```

```
[root@node1 ~]# vim /etc/ssh/sshd_config
```

```
:set nu
```

```
21 #Port 22 ; old
```

```
21 Port 2222 ; remove '#'
```

```
[root@node1 ~]# systemctl restart sshd.service
```

```
[root@node1 ~]# setenforce 0 --> SELinux Permissive mode
```

```
[root@node1 ~]# systemctl restart sshd.service
```

Verify current SSH port :

```
[root@node1 ~]# netstat -ntlp | grep ssh
```

```
[root@node1 ~]# ss -ltn
```

Check Firewall Status :

```
[root@node1 ~]# systemctl status firewalld
```

```
[root@node1 ~]# systemctl stop firewalld
```

==> Move to node1

```
[root@host1 ~]# ssh admin1@172.25.11.X (default port) - Connection Refused
```

```
[root@host1 ~]# ssh -p 2222 admin1@172.25.11.X ; যদি ইউজার 'admin1' এবং নতুন পোর্ট  
'2222' দিয়ে লগিন করতে চাই।
```

```
[admin1@node1 ~]$
```

Homework :

- ⌘ allow user 'admin2' to login through ssh
- ⌘ Generate key pair on windows os

===== Thank you =====

Class -15

What is Firewall ?

Linux Firewall Services :

- ⌘ firewalld - সাধারণ এবং বেসিক ফায়ারওয়াল সার্ভিসের ক্ষেত্রে ব্যবহার করা হয়।
- ⌘ nftables
- ⌘ iptables
- ⌘ ufw
- ⌘ fail2ban

Firewalld হচ্ছে এক ধরনের 'hostbase' & Network ফায়ারওয়াল যেটা 'Fedora/CentOS/RHEL' টাইপের অপারেটিং সিস্টেমের সাথে ডিফল্ট ভাবে থাকে।

Filtering :

- ⌘ layer3: IP/Subnet/ICMP(ping)
- ⌘ layer4: TCP/UDP port or 21,22,25,53,80,443,161,110,123
- ⌘ Layer7: dns,ftp,http,smtp,pop3,squid,imap,snmp,nfs,tftp,ntp,ssh,https

Step 01 : Change System Host Name: (Firewall Node)

```
[root@localhost ~]# hostnamectl set-hostname node2F.example.com  
[root@localhost ~]# exec bash
```

Step 02 : Check IP Address:

```
[root@node2F ~]# ifconfig
```

নোটঃ আইপি অ্যাড্রেস সেট করা না থাকলে বসিয়ে নিতে হবে ।

Starting and Enable Firewalld Services :

```
[root@node2F ~]# systemctl status firewalld
[root@node2F ~]# systemctl start firewalld
[root@node2F ~]# systemctl enable firewalld
```

Working with Firewalld :

- ⌘ By using Graphical Desktop Application (Sundry)
- ⌘ By using the web Console (https) Browser based (Cockpit)
- ⌘ By directly editing (vim/nano) configuration files in '/etc/firewalld/'.
- ⌘ By using 'firewall-cmd' from the command line

Firewalld GUI Interface :

```
[root@node2F ~]# yum install firewall-config -y
[root@node2F ~]# firewall-config &
```

Firewalld Web Interface (Cockpit) :

```
[root@node2F ~]# yum install cockpit -y
[root@node1F ~]# systemctl restart cockpit.socket
[root@node1F ~]# systemctl restart cockpit.service
```

Open Browser: <https://x.x.x.x:9090> -> Proceed

Firewalld Pre-defined (Allowed) Services :

যে কোনো ফায়ারওয়াল ডিফল্ট ভাবে সকল সার্ভিস ব্লক করবে। Firewalld তে কিছু সার্ভিস আছে। পূর্বনির্ধারিত

- ⌘ SSH - Local SSH Service (port TCP 22)
- ⌘ ipp-client - Local IPP Printing (Port UDP 631)
- ⌘ mdns- Multicast DNS (mDNS) for local-link name resolution (224.0.0.251 & port 5353/UDP)
- ⌘ samba-client - Local Windows File and Printer client (Port UDP 137 & 138)
- ⌘ dhcpv6-client - Local DHCPv6 client (Port 546/UDP)
- ⌘ Cockpit - Web based (https) interface (Port TCP 9090)

Firewall Configure Using 'vim/vi' editor :

```
[root@node1-F ~]# cd /etc/firewalld/
[root@node1-F firewalld]# ls
[root@node1-F firewalld]# cd zones
[root@node1-F zones]# ls
[root@node1-F zones]# cd public
[root@node1-F public]# vim public.xml
```

Check the firewall configure in CMD Mode :

```
[root@node1F ~]# firewall-cmd --state ; ফায়ারওয়ালের বর্তমান অবস্থা জানা যাবে।
[root@node1F ~]# firewall-cmd --get-zones ; জোন সমূহের লিস্ট দেখা যাবে।
[root@node1F ~]# firewall-cmd --get-services ; ফায়ারওয়াল যেসকল সার্ভিস নিয়ে কাজ করতে পারবে সেই লিস্ট দেখা যাবে।
```

'public' জোনে যে সকল সার্ভিস/পোর্ট/প্রটোকল এলাও করা আছে, সেই লিস্ট দেখা যাবেঃ

```
[root@node1F ~]# firewall-cmd --permanent --list-all
```

```
interfaces:
```

```
sources:
```

services: dhcpv6-client ssh

```
ports:
```

```
masquerade: no
```

```
forward-ports:
```

```
icmp-blocks:
```

```
rich rules:
```

Firewalld Pre-defined Zones :

⌘ public (default)

⌘ internal

⌘ external

⌘ trusted

⌘ home

⌘ DMZ

⌘ block

⌘ work

⌘ drop

⌘ libvirt

```
[root@node1F ~]# firewall-cmd --get-active-zones ; যে সকল জোন সমূহ একটিভ আছে সেই লিস্ট দেখা যাবে।
```

```
public
```

```
  interfaces: ens160 ens192
```

```
[root@node1F ~]# firewall-cmd --get-default-zone ; ডিফল্ট ভাবে কোন জোন সেট করা আছে সেটা দেখা যাবে।
```

```
public (output)
```

Add new Zone (DMZ) :

```
[root@node1F ~]# firewall-cmd --permanent --zone=dmz --list-all
```

```
[root@node1F ~]# firewall-cmd --permanent --zone=dmz --change-interface=ens192
```

```
success
```

```
[root@node1F ~]# firewall-cmd --reload
```

```
[root@node1F ~]# firewall-cmd --get-active-zones
```

```
dmz
```

```
  interfaces: ens192
```

```
public
```

```
  interfaces: ens160
```

Set the zone, if no zone defined :

```
[root@node1F ~]# firewall-cmd --set-default-zone=home
```

```
[root@node1F ~]# firewall-cmd --permanent --list-all
```

Configure Firewalld :

firewalld সার্ভিস প্র্যাকটিস করার জন্য Apache 'httpd' এবং 'mod_ssl' প্যাকেজটি ইন্সটল করে নিতে হবে।

Firewalld Testing with HTTPs service :

```
[root@node1F ~]# yum install httpd mod_ssl -y
```

```
[root@node1F ~]# cd /var/www/html/
```

```
[root@node1F html]# vim index.html
```

```
<h1> ##### welcome to example.com ### </h1>
```

```
[root@node1F ~]# systemctl start httpd.service
```

```
[root@node1F ~]# systemctl enable httpd.service
```

```
[root@node1F ~]# systemctl status httpd.service
```

Testing :

⌘ একটি ব্রাউজার নিয়ে (node2) Firewall এর আইপি দিয়ে হিট করতে হবে

⌘ http://172.25.11.X

⌘ Result: This site can't be reached

Allow port through firewall-cmd :

```
[root@node1F ~]# firewall-cmd --permanent --add-port=443/tcp
```

```
success
```

```
[root@node1F ~]# firewall-cmd --reload
```

```
success
```

or

Allow service through firewall-cmd :

```
[root@node1F ~]# firewall-cmd --permanent --add-service=https
success
[root@firewallX ~]# firewall-cmd --reload
success
```

```
[root@firewallX ~]# firewall-cmd --zone=public --list-all
```

Remove port through firewall-cmd: (Optional)

```
[root@serverX ~]# firewall-cmd --permanent --zone=public --remove-port=443/tcp
success
[root@serverX ~]# firewall-cmd --reload
success
```

Remove service through firewall-cmd: (Optional)

```
[root@serverX ~]# firewall-cmd --permanent --remove-service=https
success

[root@serverX ~]# firewall-cmd --reload
success

[root@firewallX ~]# firewall-cmd --zone=public --list-all
```

***** Move to Desktop

⇒ https://172.25.11.200+X or http://serverX.example.com (should succeed)

Commonly use Port & services :

1. SMTP = 25/tcp
2. FTP = 21/tcp
3. DNS = 53/tcp & 53/udp
4. POP3 = 110/tcp
5. SMB = 445/tcp
6. DHCPD = 67/udp & 68/udp
7. SSHD = 22/tcp
8. NFS = 2049/tcp
9. IMAP = 143/tcp
10. HTTP/HTTPS = 80/443 - tcp
11. iSCSI = 3260/tcp
12. MySQL = 3306/tcp

Example 01: All Service Blocked from Particular IP:

```
[root@node2F ~]# firewall-cmd --permanent --add-rich-rule="rule family='ipv4' source address='172.25.11.X' reject"
```

নোটঃ যদি নির্দিষ্ট নেটওয়ার্ক করতে চাই, তাহলে নেটওয়ার্ক আইডি দিতে হবেঃ 172.25.11.0/24

```
[root@node2F ~]# firewall-cmd --reload  
[root@node2F ~]# firewall-cmd --permanent --list-all
```

Testing :

```
host> ping 172.25.11.201 (Unreacheable)
```

```
node1> ssh 172.25.11.254 (no response/connection timeout)
```

Example 02: SSH Blocked from Particular IP :

```
[root@node2F ~]# firewall-cmd --permanent --add-rich-rule="rule family='ipv4' source  
address='172.25.11.126' service name='ssh' reject"
```

```
PC> ping 172.25.11.254 (ok)
```

```
PC> ssh 172.25.11.254 (no response/connection timeout)
```

»»»»»»»»»»»»»»»» Thank You ««««««««««««««««

Homework

1. Set your your 3x Network interfaces 1 = External, 2=Internal 3= DMZ
2. Again Set all your 3x Interfaces to Default zone (public)
3. Disallow all hosts/network to your Firewall except only single Host (ssh client) - take help from google
4. Disallow Ping from all Hosts in the network using Firewall
5. Change ssh port 2222 and allow 2222 port on firewall
6. Allow https & http and deny 'ssh' from particular IP

Class 16

- What is SELinux ?
- SELinux Modes
- Manage the SELinux Policy
- Activate & deactivate SELinux Policy Rules
- Use SELinux Log Analysis
- Change the SELinux Enforcement Mode
- The SELinux context

সিকিউরিটি মডেলঃ

অধিকাংশ লিনাক্স সিস্টেম অ্যাডমিন লিনাক্সের স্ট্যান্ডার্ড ফাইল পার্মিশনের (User, Group, Others) সাথে সুপরিচিত, যেটা সিকিউরিটির ভাষায় Discretionary Access Control (DAC) এর অন্তর্ভুক্ত। DAC ভিত্তিক সিকিউরিটি ব্যবহার করা সহজ (Flexible), এটা সহজেই এপ্লাই করা যায়, এবং এই ধরনের পার্মিশন ইউজার কর্তৃক ম্যানেজ করা হয়। অর্থাৎ, কে ফাইল read করতে পারবে, কে write করতে পারবে, কে Delete করতে পারবে, সেই ভাবে ফাইল ওনার সিকিউরিটি সেট করতে পারে। তবে, DAC ভিত্তিক সিকিউরিটি সিস্টেম কম সিকিউর প্রদান করে।

MAC (Mandatory Access Control) ভিত্তিক সিকিউরিটি মডেলে ইউজারের কোনো ক্ষমতা থাকে না। এক্ষেত্রে সিস্টেম কর্তৃক সকল রিসোর্স এক্সেস কন্ট্রোল, পার্মিশন এবং ম্যানেজ করা হয়। এক্সেস পার্মিশন সমূহ সিস্টেমের পূর্ব নির্ধারিত বিভিন্ন পলিসির মাধ্যমে ম্যানেজ করা হয়। যেহেতু পলিসি সমূহ সিস্টেম কর্তৃক (Centrally) ম্যানেজ করা হয়, এক্ষেত্রে ইউজারের কোনো কিছু পরিবর্তন করার সুযোগ থাকে না। এইজন্য MAC (Mandatory Access Control) ভিত্তিক সিকিউরিটি মডেল Discretionary Access Control (DAC) এর চেয়ে অধিক সিকিউর। AppArmor, SELinux ইত্যাদি MAC ভিত্তিক সিকিউরিটি সিস্টেম।

SELinux কি ?

SELinux এর পূর্ণ অর্থ 'Security-Enhanced Linux' যেটা Mandatory Access Control (MAC) সিস্টেম সমর্থন করে। SELinux সিকিউরিটি লিনাক্স কার্নেলের সিকিউরিটি সাব-সিস্টেমের অন্তর্ভুক্ত। SELinux সিকিউরিটি

NSA (National Security Agency) কর্তৃক ডেভলপ করা হয় এবং RHEL 4 থেকে অর্থাৎ লিনাক্স কার্নেল 2.6 ভার্সনের সাথে প্রথম SELinux এর ব্যবহার শুরু হয়।

SELinux সিকিউরিটি লিনাক্স সিস্টেমের মিলিটারি গ্রেড সিকিউরিটির অন্তর্ভুক্ত। SELinux পলিসি বা বুলিয়ান (Boolean) এর মাধ্যমে সিস্টেমের বিভিন্ন ধরনের রিসোর্স, যেমনঃ ফাইল, অ্যাপ্লিকেশন, সার্ভিস, প্রসেস, পোর্ট সমূহ রুট লেভেল থেকে নিয়ন্ত্রণ করা যায়।

SELinux Status :

- * Enable (Enforcing & Permissive)
- * Disable

SELinux Enable Modes :

- ⇒ Enforcing Mode (1)
- ⇒ Permissive Mode (0)

⌘ Enforcing: এটা SELinux এর ডিফল্ট মোড, Enforcing মোডের মাধ্যমে পলিসি সমূহ প্রয়োগ (Enforce) করা হয়। SELinux সিকিউরিটি পলিসি কর্তৃক অনুমোদিত ব্যাতিত কোনো ইউজার বা প্রোগাম কাজ করবে না।

⌘ Permissive: এই মোডে SELinux পলিসি সমূহ সিস্টেমে লোডিং অবস্থায় এবং এন্টিভ থাকে, কিন্তু কোনো পলিসি প্রয়োগ (Enforce) হয় না। তবে, SELinux এর সম্পর্কিত (Security Violations) সকল লগ ফাইল সিস্টেমে জমা থাকে। ফলে, সিস্টেম টেস্টিং, অডিটিং, প্রটেকশন এবং ট্রাবলশুটিং করতে সুবিধা হয়।

⌘ Disabled: কোনো SELinux সিকিউরিটি পলিসি কার্নেল কর্তৃক লোড হয় না। Disable অর্থ SELinux সার্ভিস সম্পূর্ণ বন্ধ থাকে।

SELinux Information :

```
[root@node1 ~]# getenforce      ; SELinux এর বর্তমান অবস্থা দেখার জন্য।  
Enforcing
```

নোটঃ Disable থাকলে এই কমান্ড কাজ করবে না।

Set SELinux Temporary Enforcing & Permissive :

```
[root@node1 ~]# setenforce      ; এই কমান্ডের মাধ্যমে SELinux এর মোড 'Enforcing' বা  
'Permissive' করা যাবে।  
[root@node1 ~]# setenforce 0 | 1  
[root@node1 ~]# getenforce
```

নোটঃ এখানে '0' অর্থ Permissive এবং '1' এর অর্থ Enforcing. আর 'getenforce' কমান্ডের মাধ্যমে SELinux এর বর্তমান অবস্থা জানা যাবে।

Current SELinux Status :

```
[root@node1 ~]# sestatus      ; SELinux সম্পর্কিত সকল তথ্য পাওয়া যাবে।
```

SELinux status: enabled - SELinux এর বর্তমান কি অবস্থায় (Disabled or Enabled) আছে।

SELinuxfs mount: /sys/fs/selinux - SELinux এর ফাইল সিস্টেম যে লোকেশনে মাউন্ট করা আছে।

SELinux root directory: /etc/selinux - যে লোকেশনে SELinux সম্পর্কিত কনফিগারেশন ফাইল আছে।

Loaded policy name: targeted - দুই ধরনের 'Policy' আছে (Targetd & Strict).
নির্দিষ্ট ডেমন বা আপ্লিকেশন 'targeted' এবং সব
ধরনের ডেমনের ক্ষেত্রে 'strict'

Current mode: enforcing - SELinux এর বর্তমান মোড।

Mode from config file: enforcing - SELinux কনফিগারেশন ফাইলে বর্তমানে যে মোডে আছে।

Policy MLS status: enabled

Policy deny_unknown status: allowed

Max kernel policy version: 33 - SELinux পলিসির যে ভার্সনটি বর্তমানে ব্যবহৃত হচ্ছে।

Change SELinux Mode :

স্থায়ী ভাবে SELinux মোড পরিবর্তন করার জন্য নিচের ফাইলটি এডিট করে প্রয়োজন মত SELinux মোড 'Enforcing', 'Permissive' এবং 'Disabled' সেট করে, ফাইল সেভ করে রিবুট দিতে হবে। সিস্টেম রিবুটের সময় সিস্টেমের সকল ফাইল/ডিরেক্টরি/সার্ভিস/পোর্ট/প্রসেস SELinux এর নতুন মোড কর্তৃক পুনরায় লেভেল (relabel) হবে। এইজন্য মোড পরিবর্তন করার পরে অবশ্যই রিবুট দিতে হবে।

```
[root@node1 ~]# vim /etc/selinux/config  
SELINUX=enforcing
```

```
[root@node1 ~]# reboot
```

The SELinux Context :

SELinux এর ব্যবহার লিনাক্স/ইউনিক্স সিস্টেমে ব্যবহৃত প্রচলিত সিকিউরিটি সিস্টেম থেকে সম্পূর্ণ আলাদা। SELinux সিকিউরিটি বিভিন্ন ধরনের সিকিউরিটি 'Context' কর্তৃক সিস্টেমের সকল রিসোর্স (Processes, Files, Service, Ports) লেভেল করা থাকে।

নিচের কমান্ডের মাধ্যমে ফাইল/ডিরেক্টরির SELinux সিকিউরিটি কন্টেক্সট সম্পর্কে জানা যাবেঃ

```
[root@node1 ~]# ls -Z  
system_u:object_r:admin_home_t:s0 anaconda-ks.cfg
```

```
[root@mynms ~]# ls -Z /
```

```
[root@node1 ~]# ls -dlZ /tmp/
drwxrwxrwt. root root system_u:object_r:tmp_t:s0 /tmp/
```

```
[root@node1 ~]# ls -lZ /home
drwx----- student 1000 unconfined_u:object_r:user_home_dir_t:s0 student
drwx-----. 5001 5001 unconfined_u:object_r:user_home_dir_t:s0 tarek
```

নোটঃ উপরের কমান্ড দেওয়ার পরে, প্রতিটি ফাইল/ডিরেক্টরির গায়ে নতুন একটি লেভেল দেখা যাচ্ছে, এক একটি পার্টকে লেভেল বলা হয়। সকল লেভেলকে এক সঙ্গে SELinux Context বলা হয়।

SELinux Label :

⇒ system_u:object_r:syslog_t:s0 (MLS)

- SELinux User - (_u)
- SELinux Role - (_r)
- SELinux Type - (_t)
- SELinux Sensitivity - (s0-s15)

RHEL 9 provides the following packages for working with SELinux :

policies: selinux-policy-targeted, selinux-policy-mls

tools: polycoreutils, polycoreutils-gui, libselinux-utils, polycoreutils-python-utils, setools-console, checkpolicy

Related Package Install :

```
[root@node1 ~]# yum install setools-console -y  
[root@node1 ~]# yum install policycoreutils* -y
```

উপরের প্যাকেজ ইন্সটলের পরে 'seinfo' কমান্ড ব্যবহার করা যাবে। 'seinfo' কমান্ডের মাধ্যমে SELinux সম্পর্কিত বিভিন্ন কুয়েরী যেমনঃ Users, Roles, Types, Attributes, Booleans, ইত্যাদি বিষয়ের তথ্য পাওয়া যাবে। নিচে 'seinfo' কমান্ড ব্যবহার করে SELinux সম্পর্কিত বিভিন্ন তথ্য বের করার পদ্ধতি দেখানো হয়েছেঃ

SELinux User :

SELinux কনটেক্সট (Context) এর প্রথম অংশ হচ্ছে SELinux ইউজার পার্ট। অর্থাৎ, প্রথম অংশ দেখে ইউজার SELinux ইউজার আইডেন্টিফাই করা যাবে। RHEL সিস্টেমে আট (০৮) ধরনের SELinux ইউজার থাকে। একজন SELinux ইউজার একাধিক SELinux রোল (Role) ম্যানেজ করতে পারে। নিচের কমান্ড ব্যবহার করে সিস্টেমে কি কি SELinux ইউজার আছে, এই সম্পর্কে জানা যাবে।

```
[root@node1 ~]# seinfo -u
```

Users: 8

```
sysadm_u  
system_u  
xguest_u  
root  
guest_u  
staff_u
```

user_u
unconfined_u

Role :

SELinux কনটেক্সট (Context) এর দ্বিতীয় অংশ হচ্ছে 'Role' এবং 'Role' সমূহ '_r' দিয়ে উল্লেখ করা হয়। একজন SELinux ইউজার একাধিক 'Role' ম্যানেজ করতে পারে। SELinux ইউজার কর্তৃক যে সকল 'Role' ম্যানেজ করা হয়, সেটা জানার জন্য নিচের কমান্ড।

```
[root@node1 ~]# seinfo -r
```

Roles: 14

auditadm_r
dbadm_r
guest_r
logadm_r
nx_server_r
object_r

secadm_r
staff_r
sysadm_r
system_r
unconfined_r
user_r
webadm_r
xguest_r

SELinux Contexts: (-t)

সিস্টেমে কি কি security context (_t) এবং মোট কতগুলো সিকিউরিটি Context আছে সেটা জানার জন্যঃ

```
[root@node1 ~]# seinfo -t
```

```
[root@node1 ~]# seinfo -t | wc -l
```

4960

```
[root@node1 ~]# seinfo -t | grep sshd
```

```
[root@node1 ~]# yum install httpd -y
```

```
[root@node1 ~]# systemctl restart httpd
```

```
[root@node1 ~]# systemctl enable httpd
```

```
[root@node1 ~]# systemctl status httpd
```

```
[root@node1 ~]# firewall-cmd --permanent --add-port=80/tcp
```

```
[root@node1 ~]# firewall-cmd --reload
```

```
[root@node1 ~]# iptables -F
```

Create a webpage :

```
[root@node1 ~]# echo "Hello World" >> /var/www/html/index.html
```

Visit web page: <http://172.25.11.X>

SELinux Testing: Example: 01

```
[root@node1 ~]# cd
[root@node1 ~]# ls -Zd ~
dr-xr-x---. root root system_u:object_r:admin_home_t:s0 /root
```

```
[root@node1 ~]# touch redhat
```

```
[root@node1 ~]# ls -Z redhat
-rw-r--r--. root root unconfined_u:object_r:admin_home_t:s0 redhat
```

```
[root@node1 ~]# cp redhat /var/www/html/centos
[root@node1 ~]# mv redhat /var/www/html/
[root@node1 ~]# ls -Zd /var/www/html/
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html/
```

```
[root@node1 ~]# ls -Z /var/www/html/*
-rw-r--r--. root root unconfined_u:object_r:admin_home_t:s0 redhat
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 centos
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 index.html
```

Change SELinux Security Context: (Method-1)

```
[root@node1 ~]# restorecon -v /var/www/html/redhat ; প্যারেন্ট ডিরেক্টরি অনুযায়ী ফাইলটি
'relabel' হবে।
```

```
[root@node1 ~]# ls -Z /var/www/html/redhat
```

```
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 redhat
```

Method 02: (Context Change)

```
[root@node1 ~]# chcon -t httpd_sys_content_t /var/www/html/redhat  
[root@node1 ~]# ls -lZ /var/www/html/redhat
```

```
[root@node1 ~]# semanage fcontext -l  
[root@node1 ~]# semanage fcontext -l | grep "/var/www"
```

```
Note: # semanage fcontext -a -t httpd_sys_content_t '/var/www/html(/.*)?'  
      # restorecon -Rv
```

SELinux Testing: Example: 02

```
[root@node1 ~]# mkdir /webhosting
```

```
[root@node1 ~]# ls -ld /webhosting  
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /webhosting
```

```
[root@node1 ~]# touch /webhosting/index.html  
[root@node1 ~]# echo "Hello SELinux World" >> /webhosting/index.html
```

```
[root@node1 ~]# ls -lZ /webhosting  
-rw-r--r--. root root unconfined_u:object_r:default_t:s0 index.html
```

```
[root@node1 ~]# vim /etc/httpd/conf/httpd.conf
```

```
124 #DocumentRoot "/var/www/html"  
123 DocumentRoot "/webhosting"  
134 <Directory "/webhosting">
```

```
[root@node1 ~]# systemctl restart httpd
```

⇒ Switch to DesktopX

⇒ Open Firefox Browser

⇒ <http://172.25.11.254>

এখন সাইট ভিজিট করলে ডিফল্ট হোম পেজ আসবে, কারণ SELinux এখন Enforcing মোডে আছে এবং '/webhosting' ডিরেক্টরির সিকিউরিটি কন্টেক্সট (context) 'http' সার্ভিসের সাথে মিল নাই। তবে আমরা যদি এটাকে 'setenforce 0' কমান্ড দিয়ে 'Permissive' মোড সেট করে দিই, তাহলে নতুন লোকেশন থেকে পেজ রান হবে।

⇒ <http://172.25.11.254/index.html>

এমতাবস্থায় SELinux 'Permissive' মোডে রাখলে ওয়েব পেজ আসবে।

```
[root@node1 ~]# setenforce 0
```

```
[root@node1 ~]# getenforce
```

⇒ <http://172.25.11.254/index.html>

```
[root@node1 ~]# setenforce 1
```

```
[root@node1 ~]# getenforce
```

⇒ <http://172.25.11.254/index.html>

কিন্তু, আমাদেরকে সব সময় SELinux মোড 'Enforcing' মোডে রেখে কাজ করতে হবে। নিচের কমান্ডের মাধ্যমে 'httpd_sys_content_t' কন্টেক্সট (context) দিয়ে '/webhosting' ডিরেক্টরি এবং এটির মধ্যে সকল সাব-ডিরেক্টরি, ফাইল সমূহের লেভেল পরিবর্তন করা হয়েছে।

```
[root@node1 ~]# semanage fcontext -a -t httpd_sys_content_t '/webhosting(/.*)?'  
[root@node1 ~]# restorecon -Rv /webhosting  
[root@node1 ~]# getenforce
```

এখন যদি পুনরায় সাইট ভিজিট করা হয়, তাহলে ওয়েব সাইট দেখা যাবে।

⇒ <http://172.25.11.254/index.html>

SELinux Port Labeling :

SELinux ফাইল এবং প্রসেস লেভেলে সিকিউরিটি (SELinux Policy) দেওয়ার পাশাপাশি নেটওয়ার্ক ট্রাফিকেও খুব শক্তভাবে মনিটরিং এবং সিকিউরিটি প্রদান করে। যেমন, বিভিন্ন ধরনের নেটওয়ার্ক সার্ভিস সমূহ (TCP/UDP) SELinux সিকিউরিটি কর্তৃক নিয়ন্ত্রণ করা হয়। উদাহরণ স্বরূপ SSH সার্ভিসের ডিফল্ট পোর্ট হিসেবে 22/TCP, HTTP সার্ভিসের জন্য '80' এবং HTTPS সার্ভিসের জন্য ডিফল্ট পোর্ট '443' ইত্যাদি SELinux কর্তৃক লেভেল (bind) করা থাকে। যেমনঃ

* ssh_port_t = 22/TCP

* http_port_t = 80/TCP & 443/TCP

যখন সিস্টেমে বিভিন্ন সার্ভিসের ডিফল্ট পোর্ট পরিবর্তন করে যদি অন্য পোর্ট ব্যবহার করা হয়, তখন SELinux সিকিউরিটি কর্তৃক সেই সার্ভিস বন্ধ করে দেওয়া হয়।

[root@node2 ~]# netstat -ntlp | grep ssh ; বর্তমানে 'SSH' সার্ভিস কোন পোর্টে রান করছে সেটা দেখার জন্য।

[root@node2 ~]# semanage port -l ; SELinux কর্তৃক লেভেলকৃত পোর্টে সমূহের লিস্ট দেখা যাবে।

[root@node2 ~]# semanage port -l | grep ssh ;SELinux কর্তৃক লেভেলকৃত 'SSH' সার্ভিসের পোর্ট নাম্বার দেখার জন্য।

```
ssh_port_t          tcp      22
```

[root@node2 ~]# semanage port -l | grep http

[root@node2 ~]# semanage port -a -t ssh_port_t -p tcp 2222 ;'SSH' এর পোর্ট '22' এর পাশাপাশি '2222' পোর্টেও লেভেল (bind) করা হল।

[root@node2 ~]# semanage port -l | grep ssh

```
ssh_port_t          tcp      2222, 22
```

** Change SSH Default port '2222'

SELinux Boolean :

SELinux Booleans এক ধরনের কনফিগারেশন পদ্ধতি যেটা ব্যবহার করে অ্যাডমিনিস্ট্রেটর খুব সহজে SELinux পলিসি পরিবর্তন ছাড়াই ব্যবহার করতে পারবে।

[root@node2 ~]# getsebool -a ; সিস্টেমে যত 'Boolean' আছে সকল লিস্ট দেখার জন্য।

[root@node2 ~]# semanage boolean -l

[root@node2 ~]# semanage boolean -l | grep ftp ; service must be enabled

[root@node2 ~]# semanage boolean -l | grep httpd ; service must be enabled

[root@node2 ~]# semanage boolean -l -C

যে কোনো Boolean এনাবেল করার জন্য 'on' বা ডিজাবেল করার জন্য 'off' করতে হবে।

Boolean Enable করার জন্য নিচের কমান্ডঃ

```
[root@node2 ~]# setsebool -P < boolean_name > on
```

Lab Demo :

এখানে 'student' ইউজারের 'home' ডিরেক্টরিতে একটি webpage হোস্ট করা হবে। উক্ত ওয়েব পেজটির হোস্টিং ডিরেক্টরি 'SELinux Boolean' দিয়ে enable করতে হবে।

```
[root@node2 ~]# vim /etc/httpd/conf.d/userdir.conf
```

```
11 <IfModule mod_userdir.c>
17 #   UserDir disabled
24   UserDir public_html
25 </IfModule>
```

```
[root@node2 ~]# systemctl restart httpd
```

```
[root@node2 ~]# su - student
```

```
[student@node2 ~]$ mkdir ~/public_html
```

```
[student@node2 ~]$ echo 'This is student content on Node1' > ~/public_html/index.html
```

```
[student@node2 ~]$ chmod 711 ~
```

Student ইউজারের হোস্টিং এক্সেস করার জন্য নিচের URL অনুযায়ী চেক করতে হবেঃ

<http://172.25.11.254/~student/index.html> ভিজিট করতে হবে।

Output: An error message states that you do not have permission to access the file.

```
[root@node2 ~]# semanage boolean -l | grep httpd_enable_homedirs
httpd_enable_homedirs      (off , off) Allow httpd to read home directories
```

Enable Boolean 'httpd_enable_homedirs' for user home directory :

```
[root@node2 ~]# setsebool -P httpd_enable_homedirs on
```

```
[root@node2 ~]# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> on
```

Now Visit: <http://172.25.11.254/~student/index.html> ভিজিট করতে হবে।

===== Thank you =====

Homework

Your company has decided to run a new web app. This application listens on ports 80/TCP and 8090/TCP. Port 2222/TCP for ssh access must also be available. All changes you make should persist across a reboot.

```
# sealert -a /var/log/audit/audit.log
```

Class -17

NFS (Network File System) :

NFS, এর পূর্ণ অর্থ Network File System, NFS ক্লায়েন্ট এবং সার্ভার আর্কিটেকচারে কাজ করে। NFS নেটওয়ার্কে ফাইল সার্ভার বা স্টোরেজ হিসেবে কাজ করে। NFS ফাইল ভিত্তিক স্টোরেজ সলিউশন দেয় এবং NFS মূলত লিনাক্স-লিনাক্স, লিনাক্স-ইউনিক্স ফাইল সিস্টেম ব্যবহার করে। এছাড়া উইন্ডোজ অপারেটিং সিস্টেমে NFS এর ক্লায়েন্ট অ্যাপ্লিকেশন ইন্সটল করে NFS সার্ভার/শেয়ার উইন্ডোজ অপারেটিং সিস্টেম থেকে ব্যবহার করা যায়।

ডাটা স্টোরেজ হিসেবে আমরা তিন (০৩) ধরনের পদ্ধতি ব্যবহার করিঃ

1. DAS Storage - DAS (Direct Attached Storage) হচ্ছে আমাদের সিস্টেমের সাথে সংযুক্ত লোকাল স্টোরেজ। শুধুমাত্র লোকাল সিস্টেমের ইউজারের ডাটা এখানে স্টোর করা হয়। DAS এর সবচেয়ে বড় অসুবিধা হচ্ছে ইচ্ছামত স্টোরেজ বাড়ানো যায় না এবং অন্য ইউজারের মাঝে শেয়ার দিয়ে কন্ট্রোল করা যায় না।

2. NAS Storage - Network Attached Storage (NAS) এক ধরনের ফাইল স্টোরেজ এবং এটা এক প্রকার শেয়ারড স্টোরেজ। অর্থাৎ NAS স্টোরেজ ব্যবহার করে আমরা নেটওয়ার্কে বিভিন্ন ইউজারদের মাঝে স্টোরেজ স্পেস শেয়ার করতে পারি। NAS হচ্ছে এক প্রকার ফাইল স্টোরেজ অর্থাৎ NAS এর মধ্যে যা কিছু রাখি না কেন, সব ফাইল হিসেবে গন্য হবে। NAS সার্ভার থেকে যে ফাইল সিস্টেমে ফরম্যাট করবে এবং যতটুকু স্পেস শেয়ার করা হবে, ক্লায়েন্ট ততটুকু স্পেস পাবে। ক্লায়েন্ট কোনো ভাবেই স্পেস বাড়াতে বা ফাইল সিস্টেম ফরম্যাট করতে বা পরিবর্তন করতে পারবে না। একই NAS স্পেস নেটওয়ার্কে অনেক ইউজারের মাঝে শেয়ার করা যায়।

দুই ধরনের NAS স্টোরেজ হয়ে থাকেঃ

1. NFS - Network File System (লিনাক্স/ইউনিক্স ফাইল সিস্টেম (XFS/EXT4) ফরম্যাট করে NAS কনফিগার করা হয়)

2. CIFS - Common Internet File System (উইন্ডোজ/লিনাক্স/ইউনিক্স/ম্যাক সকল অপারেটিং সিস্টেমের ফাইল সিস্টেম সাপোর্ট করে)

3. SAN Storage - Storage Area Network (SAN) হচ্ছে এন্টারপ্রাইজ গ্রেডের স্টোরেজ সলিউশন, SAN মূলত ব্লক (Block) টাইপ স্টোরেজ। অর্থাৎ SAN এর মাধ্যমে ইউজারদের মাঝে কোনো স্পেস শেয়ার করলে, ইউজার চাইলে নিজের মত ফাইল সিস্টেম (Ext4/XFS/NTFS) ফরম্যাটে পরিবর্তন করে ব্যবহার করতে পারে। SAN মূলত অনেক ব্যয়বহুল স্টোরেজ সলিউশন এবং reliable স্টোরেজ সলিউশন প্রোভাইড করে।

Benifits of NFS :

- ⌘ Supported on Linux, Unix, macOS, and Windows
- ⌘ Multiple NFS Clients can access the same share
- ⌘ Seamless permission and access control on in Unix/Linux platform.
- ⌘ Configuration is fairly straightforward
- ⌘ TCP Oriented file sharing (Old UDP)
- ⌘ Support Kerberos-based Security (Authentication)

Reference Table :

Package: nfs-utils

Daemon: nfs-server

Protocol: TCP

Port: 2049 (TCP)

NFS Server: node1: - 172.25.11.X/24

NFS Client: node2: - 172.25.11.Y/24

Change Host name :

```
[root@localhost ~]# hostnamectl set-hostname node2-NFS.example.com
```

```
[root@localhost ~]# bash
```

```
[root@node2-NFS ~]# ip addr
```

```
* # nmcli connection up <int name>
```

Step 01: Query and Package Install

```
[root@node2-NFS ~]# rpm -qa | grep nfs-utils
```

```
[root@node2-NFS ~]# yum install nfs-utils -y ; যদি প্যাকেজ (RPMs) ইন্সটল না থাকে।
```

Step 02: Create a shared directory

```
[root@node2-NFS ~]# mkdir /nfsshare
```

```
[root@node2-NFS ~]# cd /nfsshare
```

```
[root@node2-NFS nfsshare]# ls
```

```
[root@node2-NFS nfsshare]# mkdir private public
```

```
[root@node2-NFS nfsshare]# ls
```

```
private public
```

```
[root@node2-NFS nfsshare]# touch private/file{1..10}
```

```
[root@node2-NFS nfsshare]# touch public/test{1..10}
```

Step 03: Export shared directory on NFS node1 :

```
[root@node2-NFS ~]# vim /etc/exports
```

```
/nfsshare/private 172.25.11.0/24(ro,sync,no_root_squash)
```

```
/nfsshare/public 172.25.11.0/24(rw,sync,no_root_squash)
```

Note :

⇒ /nfshare/* - shared directory

⇒ 172.25.11.0/24 - IP address range of clients

⇒ rw - Writable permission to shared folder

⇒ ro - Read only permission to shared folder

⇒ sync - Synchronize shared directory

⇒ no_root_squash - Allow root privilege

⇒ no_all_squash - All users, including non-root users, have restricted access to the NFS server.

Step 04: Restart the NFS service and verify :

```
[root@node2-NFS ~]# systemctl restart nfs-server.service
```

```
[root@node2-NFS ~]# systemctl enable nfs-server.service
```

```
[root@node2-NFS ~]# exportfs -ra ; যদি কোনো ভুল কনফিগারেশন থাকে, তাহলে ধরবে।
```

```
[root@node2-NFS ~]# showmount -e localhost
```

Step 05: SELinux Policies: (If not applied)

```
[root@node2-NFS ~]# getenforce ; SELinux এর বর্তমান অবস্থা জানার জন্য।
```

```
[root@node2-NFS ~]# setenforce 0 ; SELinux Permissive মোডে সেট করা হয়েছে।
```

```
[root@node2-NFS ~]# getenforce
```

Step 06: Allow port & service through Firewall :

```
[root@node2-NFS ~]# systemctl status firewalld.service
```

```
[root@node2-NFS ~]# firewall-cmd --permanent --add-service=nfs
```

```
[root@node2-NFS ~]# firewall-cmd --permanent --add-service=mountd
```

```
[root@node2-NFS ~]# firewall-cmd --permanent --add-service=rpc-bind
```

```
[root@node2-NFS ~]# firewall-cmd --reload
```

```
[root@node2-NFS ~]# firewall-cmd --list-all
```

Client Side Configuration :

Step 07: Install NFS packages in your client system

```
[root@node1 ~]# yum install nfs-utils ; যদি প্যাকেজ (RPMs) ইন্সটল না থাকে।
```

Step 08: View NFS Share on NFS node1

```
[root@node1 ~]# showmount -e 172.25.11.201 ; your NFS Server (node1) IP
```

Export list for 172.25.11.201:

```
/nfsshare/private 172.25.11.0/24
```

```
/nfsshare/public 172.25.11.0/24
```

Step 09: Mount NFS shares On client

```
[root@node1 ~]# df -HT
```

```
[root@node1 /]# mkdir /nfs_public
```

```
[root@node1 /]# mkdir /nfs_private
```

```
[root@node1 ~]# mount -t nfs 172.25.11.201:/nfsshare/public /nfs_public
```

```
[root@node1 ~]# df -HT
```

```
[root@node1 ~]# cd /nfs_public
```

```
[root@node1 nfs_public]# ls ; এখানে শেয়ার দেওয়া সকল ফাইল দেখা যাবে।
```

```
[root@node1 nfs_public]# touch file.tar ;নতুন ফাইল তৈরি হয়েছে।
```

```
[root@node1 nfs_public]# cd
```

```
[root@node1 ~]#
```

*** Lab Task: Mount 'private' share under '/nfs_private' directory. Create a file under '/nfs_private'

Unmount করার জন্য নিচের কমান্ডঃ

```
[root@node1 ~]# umount /nfs_public  
[root@node1 ~]# umount /nfs_private
```

Working with AutoFS :

আমরা যখন রিমোট শেয়ার এক্সেস করার দরকার হয়, তখন শেয়ার ম্যানুয়ালি লোকাল মাউন্টে পয়েন্ট মাউন্ট করে নিয় এবং এটা অস্থায়ী পদ্ধতি। আমরা যদি রিমোট শেয়ার, বা কোনো এক্সটারনাল ড্রাইভ (HDD/DVD/USB) সিস্টেমে অটোমেটিক মাউন্ট করতে চাই, তাহলে আমাদের AutoFS কনফিগার করতে হবে।

⌘ NFS Shared directories

Install Required Package: (Node1)

```
[root@node1 ~]# yum install autofs -y
```

Enable AutoMount NFS Share: (node-1)

```
[root@node1 ~]# vim /etc/auto.master  
:set nu
```

8 /nfsmount /etc/automount.nfs --timeout=30 ---> ৩০ সেকেন্ড পরে নিজে থেকেই ডিস্কানেট হয়ে যাবে।

Note: /<Mount-Point> <Map-file> <Timeout-Value>

নোটঃ Map ফাইল হচ্ছে একটি কাস্টম ফাইল। আমরা চাইলে ইচ্ছেমত নাম দিয়ে তৈরি করতে পারি।

In map file (/etc/auto.misc or /etc/auto.xxxx) also we have three different fields:

[root@node1 ~]# vim /etc/automount.nfs ; এটি একটি নতুন ফাইল, যেটা উপরের '/etc/auto.master' ফাইলে উল্লেখ করা হয়েছে।

Hints: <mount point> <-mount options> <server-location>

private	-ro,sync,nfs4	172.25.11.201:/nfsshare/private
public	-rw,sync,nfs4	172.25.11.201:/nfsshare/public

```
[root@node1 ~]# systemctl enable autofs --now
```

```
[root@node1 ~]# df -HT
```

```
[root@node1 ~]# mount -a
```

```
[root@node1 ~]# cd /
```

```
[root@node1 ~]# ls
```

```
[root@node1 ~]# cd /nfsmount/
```

```
[root@node1 nfsmount]# ls
```

```
[root@node1 nfsmount]# cd private
```

```
[root@node1 nfsmount]# df -HT
```

* Reboot Required

নোটঃ '/nfsmount' নামে আলাদা কোনো ডিরেক্টরি তৈরি করতে হবে না। যেহেতু '/etc/auto.master' ফাইলের মধ্যে এই নামে লোকাল মাউন্ট পয়েন্ট উল্লেখ করা হয়েছে, সুতরাং 'autofs' সার্ভিস রিস্টার্ট দিলে এই নামে একটা ডিরেক্টরি তৈরি হয়ে যাবে। '/nfsmount' এর মধ্যে 'private' এবং 'public' নামে শেয়ার ডিরেক্টরি তৈরি হবে। তবে, '/nfsmount' এর মধ্যে গেলে কিছুই পাওয়া যাবে না। যখন cd কমান্ড দিয়ে 'private' বা 'public' ডিরেক্টরিতে ঢোকার চেষ্টা করা হবে। তখন শুধু আসবে। তবে, সেটাও নির্দিষ্ট টাইমের পরে চলে যাবে।

===== Thank You =====

Class -18

Module Objectives :

- ⌘ Introducing Linux Shells
- ⌘ What is shell scripts?
- ⌘ Why need Shell Programming?
- ⌘ Writing a shell script
- ⌘ Executing a shell script
- ⌘ Using Shell Variables
- ⌘ Arithmetic Operators
- ⌘ Working with Conditionals Statements
- ⌘ Working with Loops

Linux Shell পরিচিতিঃ

লিনাক্স কার্নেল ভিত্তিক অপারেটিং সিস্টেমের একটি গুরুত্বপূর্ণ ও স্পেশাল প্রোগ্রাম হচ্ছে লিনাক্স শেল (Shell)। লিনাক্সের শেল (Shell) ইউজারের কাছ থেকে কমান্ড গ্রহণ করে এবং ইউজার কর্তৃক প্রদত্ত কমান্ড যদি সঠিক হয়, তাহলে শেল (Shell) এই কমান্ডকে কার্নেলের নিকট প্রেরণ করে অপারেটিং সিস্টেমের সঙ্গে তথ্য আদান-প্রদান (Interact) করার সুবিধা প্রদান করে। লিনাক্স শেল (Shell) হচ্ছে একটি কমান্ড ল্যাঙ্গুয়েজ ইন্টারপ্রেটার, যেটা কীবোর্ড অথবা ফাইল হতে কমান্ড গ্রহণ করে এবং কমান্ডকে সরাসরি এক্সিকিউট করতে পারে।

- ⇒ It's like a container
- ⇒ Interfaces between users and Kernel/OS
- ⇒ input from keyboard/Application/files via users
- ⇒ output to monitor/terminal from kernel
- ⇒ Linux sh, bash, zshell, tcshell are shell
- ⇒ its a interpreted Programming interface

Linux -এ ব্যবহৃত Shell সমূহঃ

1. sh - Bourne Shell
2. bash - Bourne-again shell (commonly used)
3. zsh - zshell (with macOS)
4. csh - C cshell (c programming based)
5. ksh - Korn shell (Bell Labs)

Find your Shell :

নিচের কমান্ডের মাধ্যমে লিনাক্স সিস্টেমে কি কি Shell ইন্সটল করা আছে সেটা দেখা যাবে। পাশাপাশি বর্তমানে কোন Shell ব্যবহার হচ্ছে সেটাও দেখা যাবে।

```
[root@node1 ~]# echo $SHELL
[root@node1 ~]# cat /etc/shells
/bin/sh
/bin/bash
/usr/bin/sh
/usr/bin/bash
```

যদি নতুন শেল ইন্সটল করার দরকার হয়, তাহলে আমরা নিচের কমান্ডের মাধ্যমে ইন্সটল করতে পারিঃ

```
[root@node1 ~]# yum install zsh -y
[root@node1 ~]# yum install tcsh -y

[root@node1 ~]# cat /etc/shells
```

What is Shell Scripts ?

Shell Script হচ্ছে একটি সাধারণ টেক্সট ফাইল যার মধ্যে কিছু কমান্ড Sequentially সংরক্ষণ করা হয়। যখন Script টি execute করা হয়, তখন script -এ সংরক্ষিত কমান্ডগুলো Sequentially execute করে এবং নির্ধারিত কাজ সম্পন্ন করে। Shell Script কে MS-Dos batch ফাইলের সঙ্গে তুলনা করা যায়, তবে Shell Script, MS-Dos batch ফাইল অপেক্ষা অধিকতর শক্তিশালী এবং কার্যক্ষম। এক কথায় বলা যেতে পারে, আমরা ম্যানুয়ালি যত কমান্ড রান করি, সেই কমান্ড গুলো কোনো ফাইলের মধ্যে রেখে ফাইল/প্রোগ্রামটি execute করতে পারলে, সেটাকে Shell Script বলা যাবে। উল্লখ্য যে, আমরা চাইলে স্ক্রিপ্টের মাঝে বিভিন্ন কমান্ড, কন্ডিশনাল স্টেটমেন্ট (if, else, while, until), লুপ, ফাংশন এবং ভ্যারিয়েবল কল করতে পারি।

উদাহরণ হিসেবে, আমরা যদি লিনাক্স সিস্টেমে একজন ইউজার তৈরি এবং পাসওয়ার্ড দিতে চাই, এই জন্য আমাদের অনেক গুলো কমান্ড চালাতে হবে। আবার উক্ত ইউজার সিস্টেমে আগে থেকেই আছে কিনা সেটাও জানা দরকার। এখন এই কাজ টি আমরা যদি স্ক্রিপ্টের মাধ্যমে করতে চাই, তাহলে প্রথমে আমাদেরকে সিস্টেমে ইউজার আছে কিনা চেক করতে হবে, তারপর ইউজার তৈরি এবং ইউজারের পাসওয়ার্ড দিতে হবে। এখন এই কমান্ড গুলো যদি কোনো ফাইলের মধ্যে রেখে সেটাকে স্ক্রিপ্ট হিসেবে রান করলে আমাদের প্রয়োজনীয় আউটপুট পাওয়া যাবে।

```
# id <username>
# useradd <username>
# passwd <username>
  <2 times>
# id <username>
```

Shell Programming এর প্রয়োজনীয়তাঃ

- ⌘ শেল স্ক্রিপ্ট ইউজারের কাছ থেকে কীবোর্ডের মাধ্যমে বা ফাইল হতে ডাটা গ্রহণ করতে পারে এবং ফাইলে বা স্ক্রিনে ডাটা আউটপুট দিতে পারে।
- ⌘ নিজের মত করে ব্যবহার উপযোগী কমান্ড তৈরি করা যায়।
- ⌘ লিনাক্স সিস্টেম অ্যাডমিনিস্ট্রেশন সম্পর্কিত সকল কাজ অটোমেটিক ভাবে সম্পাদন করা যায়।
- ⌘ শেল স্ক্রিপ্টের মাধ্যমে সিস্টেম অপারেট করলে অতি দ্রুত সব কাজ সম্পাদন করা যায়।

- ⌘ একাধিক সার্ভারের মধ্যে আপডেট, প্যাচ, সফটওয়্যার ইন্সটল করার ক্ষেত্রে।
- ⌘ সিস্টেমের বিভিন্ন টাস্ক/রিসোর্স মনিটরিং করার ক্ষেত্রে।
- ⌘ সিস্টেম থেকে কোনো এলার্ম বা নোটিফিকেশন পাঠানোর ক্ষেত্রে।
- ⌘ বিভিন্ন ধরনের ট্রাবুলশুটিং এবং অডিটিং করার ক্ষেত্রে।

Shell Script Extension:

যে কোনো প্রোগ্রামিং ল্যাঙ্গুয়েজ লিখলে সেই ফাইলের একটি এক্সটেনশন থাকে, যেমনঃ সি (C) প্রোগ্রামিং এর ক্ষেত্রে '.c', পাইথন (Python) প্রোগ্রামিং এর ক্ষেত্রে '.py', পার্ল (Perl) এর ক্ষেত্রে '.pl' দিতে হয়। সুতরাং আমরা যখন শেল (Bash) স্ক্রিপ্ট লিখবো, তখন এটারও একটা এক্সটেনশন দিতে হবে। এক্সটেনশন হিসেবে '.sh' বা '.bash' ব্যবহার করা হয়। এক্সটেনশন বাদেও স্ক্রিপ্ট এক্সিকিউট করা যায়, সেক্ষেত্রে ম্যানুয়ালি স্ক্রিপ্ট রান করতে হয়।

- ⇒ test
- ⇒ test.sh
- ⇒ test.bash

How to write a Bash Script :

প্রথমে 'day18' এর মাঝে 'scripting' নামে একটা ডিরেক্টরি তৈরি করে নেওয়া যাক, যার মধ্যে শেল স্ক্রিপ্ট সম্পর্কিত সকল ল্যাবগুলো প্যাকটিস করা হবে। আমরা এখানে 'touch' কমান্ড দিয়ে এক্সটেনশনসহ (.sh or .bash) অথবা 'vim' কমান্ড দিয়ে এক্সটেনশনসহ (.sh or .bash) নতুন ফাইল তৈরি করতে পারি। যেহেতু এটা আমাদের প্রথম স্ক্রিপ্ট এবং টেস্ট স্ক্রিপ্ট হবে, সুতরাং এটার নাম দেওয়া যাক 'myscript1.sh'

```
[root@node1 ~]# mkdir day18/scripting -p
[root@node1 ~]# cd day18/scripting
[root@node1 scripting]# ls
[root@node1 scripting]# vim myscript1.sh
```

```
#!/bin/bash
# Date: 01-01-2023
# This is my first bash script
```

```
echo "Welcome to Bash Shell Script"
date
```

```
pwd
whoami
```

- ⌘ Line1: প্রথম লাইন কে বলা হয়, Shebang যেটা '#!' দিয়ে শুরু হয়। '#' এর পরে interpreter এর পাথ উল্লেখ থাকে। এখানে 'interpreter' হিসেবে 'bash' ব্যবহার হচ্ছে।
- ⌘ Line2: '#' দিয়ে কোনো লাইন শুরু হলে সেটাকে কমেন্ট বলা হয়। '#' দেওয়া লাইন এক্সিকিউট হবে না।
- ⌘ Line5: 'echo' কমান্ড দিয়ে স্ট্রিং প্রিন্ট করা হয়েছে।
- ⌘ Line6: 'date' কমান্ড কে কল করা হয়েছে। ফলে, আজকের তারিখ এবং সময় দেখাবে।
- ⌘ Line7: এখানে 'pwd' কমান্ড ব্যবহার করা হয়েছে। অর্থাৎ বর্তমান ডিরেক্টরির পাথ দেখাবে।
- ⌘ Line8: whoami কমান্ড ব্যবহার করা হয়েছে। আমি কোন ইউজার হিসেবে লগইন করেছি সেটা দেখাবে।

উপরের কমান্ড সমূহ লেখার পরে ':x' প্রেস করে ফাইল সেভ করে বের হয়ে আসতে হবে। এখন স্ক্রিপ্ট ফাইলটি (myscript1.sh) এক্সিকিউট (Execute) করলে ফাইলের মধ্যে থাকা কমান্ড সমূহ ধারাবাহিক ভাবে আউটপুট দেখাবে।

Setting shell permission :

আমরা জানি লিনাক্সের টেক্সট ফাইল সমূহে কোনো এক্সিকিউট পার্মিশন থাকে না। সুতরাং কোনো স্ক্রিপ্ট ফাইলকে এক্সিকিউট করতে চাইলে, অবশ্যই সেটাকে এক্সিকিউট পার্মিশন (x) যোগ করতে হবে। নিচের কমান্ডের মাধ্যমে 'myscript1.sh' ফাইলকে ইউজার এক্সিকিউট (Owner) পার্মিশন দেওয়া হয়েছে।

```
[root@node1 scripting]# ll
-rw-r--r--. 1 root root 90 Jul 27 10:46 myscript1.sh
```

```
[root@node1 scripting]# chmod u+x myscript.sh
```

```
[root@node1 scripting]# ll
-rwxr--r--. 1 root root 90 Jul 27 10:46 myscript1.sh
```

Script Execute/Run :

স্ক্রিপ্ট আমরা দুইভাবে এক্সিকিউট করতে পারি। 1) Path ভ্যারিয়েবল ব্যবহার করে এবং 2) 'sh' বা 'bash' কমান্ড দিয়ে।

```
[root@node1 scripting]# ./myscript1.sh ; এক্ষেত্রে এক্সিকিউট পার্মিশন (x) থাকতে হবে।
```

or

```
[root@node1 scripting]# sh myscript1.sh ; এক্ষেত্রে এক্সিকিউট পার্মিশন (x) না থাকলেও স্ক্রিপ্ট রান করবে।
```

```
#####  
# Welcome to bash shell scripting !!  
# Mon Aug 21 09:59:10 PM +06 2023  
# /root/day18/scripting  
# root  
#####
```

Quotes in Bash :

যে কোনো প্রোগ্রামিং ল্যাঙ্গুয়েজের ক্ষেত্রে 'Quote' একটি গুরুত্বপূর্ণ বিষয়। Linux Scripting -এ আমরা বিভিন্ন command, Texts, এবং string এর সাথে 'Quote' ব্যবহার করে থাকি। আমরা এখানে সিঙ্গেল 'Quote', ডাবল "Quote" এবং ব্যাক `Quote` এর ব্যবহার শিখবো। সাধারণ টেক্সট ফাইলের ক্ষেত্রে সিঙ্গেল 'Quote' এবং ডাবল "Quote" মধ্যে তেমন কোন পার্থক্য নাই, কিন্তু, যখন কোনো Scripting এর মাঝে কোনো variable বা command কল করা হবে তখন, আমাদের সিঙ্গেল 'Quote' এবং ডাবল "Quote" এর ব্যবহার ঠিকমত করতে হবে।

```
[root@node1 scripting]# vim myscript2.sh
```

```
#!/bin/bash  
# Using Quote in Bash
```

```
echo 'String in single quote'  
echo "String in double quote"  
  
mkdir '/day18'  
touch "/day18/file1"  
  
echo `cal`  
echo "Today date is:" `date`  
echo 'My system name is:' `hostname`  
  
[root@node1 scripting]# sh myscript2.sh
```

Linux -এ Variable এর ব্যবহারঃ

সিস্টেমের ডেটা/তথ্য কে প্রসেস করার জন্য সিস্টেমের মেমোরি (RAM) ব্যবহার করা হয়। RAM ডেটাকে ভাগ করে ছোট ছোট লোকেশনে রাখে এবং প্রতিটি লোকেশনের একটি ইউনিক নাম্বার আছে, যেটাকে বলা হয় Memory Location Address.

আমরা মেমোরি লোকেশনকে ইচ্ছামত নাম দিতে পারি। যাকে আমরা বলি variable, variable হচ্ছে memory storage যেখানে আমরা আমাদের ডাটা/ভ্যালু স্টোর করে রাখতে পারি এবং পরে ইচ্ছামত কল বা ব্যবহার করতে পারি। যেহেতু Variable মেমরিতে অর্থাৎ RAM -এ ডাটা স্টোর করে থাকে এবং RAM যেহেতু Volatile মেমোরি সুতরাং রিবুটের পরে এটা Variable এবং Variable এর ভ্যালু বা ডাটা মুছে যাবে।

লিনাক্সে দুই ধরনের Variable ব্যবহার করা যায়ঃ

- ⇒ Environment Variables (Block Letter)
- ⇒ User Defined Variables (UDV) (Small and Block)

Environment Variables :

Environment ভ্যারিয়েবল লিনাক্স সিস্টেমের নিজেস্ব ভ্যারিয়েবল যা লিনাক্স সিস্টেমের শেলের মধ্যে সংরক্ষণ (Built-in) থাকে। Environment ভ্যারিয়েবল সমূহ Capital Letter দ্বারা প্রকাশ করা হয়। শেলের সেশন এবং ওয়ার্কিং এনভায়রনমেন্টের উপরে ভিত্তি করে Environment ভ্যারিয়েবল তথ্য বা ডাটা সেটার করে। উল্লেখ্য যে, Environment ভ্যারিয়েবলের ভ্যালু স্থায়ী ভাবে সংরক্ষিত থাকে। তবে, যখন Bash সেশন রান করা হয়, তখন লিনাক্স সিস্টেম নতুন করে Environment ভ্যারিয়েবল সেট করে। Bash শেলে দুই ধরনের Environment ভ্যারিয়েবল ব্যবহার হয়ঃ

⌘ Global Variables

⌘ Local Variables

[root@node1 ~]# printenv ; সকল Global Environmental ভ্যারিয়েবল দেখার কমান্ড।

[root@node1 ~]# set ; সকল local + Global ভ্যারিয়েবল দেখার কমান্ড।

Environment Variable ব্যবহার করার নিয়মঃ

যে কোনো ভ্যারিয়েবল (User Defined & Environment Defined) ব্যবহার করার সময় ভ্যারিয়েবলের পূর্বে '\$' সাইন ব্যবহার করতে হবে। নিচে কিছু Environment ভ্যারিয়েবল ব্যবহার করার পদ্ধতি দেখানো হয়েছেঃ

[root@node1 ~]# echo \$LOGNAME - সিস্টেমে বর্তমানে কে লগইন করে আছে সেই নাম দেখাবে।

[root@node1 ~]# echo \$PWD - বর্তমান ডিরেক্টরির (Working) অবস্থান দেখাবে।

[root@node1 ~]# echo \$SHELL - বর্তমানে ব্যবহৃত শেলের নাম জানা যাবে।

[root@node1 ~]# echo \$HOSTNAME - সিস্টেমের হোস্ট নাম জানা যাবে।

[root@node1 ~]# echo \$OSTYPE - অপারেটিং সিস্টেম সম্পর্কে জানা যাবে।

শুধু 'HOSTNAME' বা '\$HOSTNAME' ব্যবহার করে এন্টার প্রেস করলে ভুল কমান্ড দেখাবে। এই জন্য কমান্ডের শুরুতে 'echo' ব্যবহার করতে হবে। আবার 'echo LOGNAME' ব্যবহার করলে শুধু LOGNAME স্ক্রিনে প্রিন্ট হবে। LOGNAME ভ্যারিয়েবলকে কল করতে চাইলে '\$' দিয়েই করতে হবে

User Defined Variable :

User defined variable ইউজার নিজে তৈরি করবে এবং ভ্যালু (ডেটা) সেট করবে। User define variable গুলো small/block latter দুই ভাবেই ব্যবহার করা হয়। মনে রাখবেন, Linux একটি case sensitive OS, অর্থাৎ 'abc' এবং 'ABC' যদিও একই শব্দ (string) কিন্তু, Linux -এ দুইটি ভিন্ন ভিন্ন শব্দ (string) বলে বিবেচিত হয়। User define variable ব্যবহার করার জন্য নিচের কমান্ড সিনট্যাক্স অনুসরণ করা হয়ঃ

Variable name=value

নিচে কিছু উদাহরণ দেখানো হয়েছেঃ

```
⇒ a=10
⇒ B=20
⇒ name=ikbal
⇒ id=1001
```

মনে রাখতে হবে যে, 'variable', '=' এবং 'value' এই তিনটির মাঝে যেন কোনো space না থাকে। 'a = 10', 'a= 10', 'a =10' এইগুলো ভুল ব্যবহার। একইভাবে, '10 = a' এই ভাবেও Variable ব্যবহার করা যাবে না, 'Variable name' সর্বদা '=' Sign এর বাম পাশে থাকবে এবং ভ্যালু (Data) ডান পাশে থাকবে। আপনারা যে কোনো Script -এ যে কোনো Variable ব্যবহার করতে পারেন। এমনকি Shell prompt থেকেও যে কোনো Variable ব্যবহার বা কল করা যায়। Variable কে কল বা ব্যবহার করার জন্য পূর্বের মত '\$' চিহ্ন ব্যবহার করতে হবে।

নিচে কিছু উদাহরণ দেখানো হয়েছেঃ

```
[root@node1 day18]# a=10
[root@node1 day18]# echo $a
10
```

```
[root@node1 day18]# B=20
[root@node1 day18]# echo $B
20
```

```
[root@node1 day18]# name=ikbal
[root@node1 day18]# echo $name
ikbal
```

```
[root@node1 day18]# id=1000
[root@node1 day18]# echo $id
1000
```

```
[root@node1 day18]# _user=student
[root@node1 day18]# echo $_user
student
```

```
[root@node1 day18]# _2023=year
[root@node1 day18]# echo $_2023
year
```

User Input In Bash using 'read' :

লিনাক্সে 'read' কমান্ড ব্যবহার করে সরাসরি interactive মোডে ব্যাশ স্ক্রিপ্টে ইউজার ইনপুট বা ইউজারের কাছ থেকে ডেটা বা তথ্য নেওয়া যায়। নিচের কমান্ডের মাধ্যমে ইউজার ইনপুট নেওয়ার পদ্ধতি দেখানো হয়েছেঃ

উদাহরণ স্বরূপঃ

```
[root@node1 ~]# read a
```

উপরের কমান্ডটি ইউজারের কাছ থেকে একটি ভ্যালু নিয়ে (read) সেটা ভ্যারিয়েবল 'a' তে সংরক্ষণ করবে। পরে যদি 'a' ভ্যারিয়েবল কে কল করা হয়, তাহলে 'a' ভ্যারিয়েবলের আউটপুট দেখাবে। যেমনঃ

```
[root@node1 scripting]# read a
10
[root@node1 scripting]# read b
20
[root@node1 scripting]# echo `expr $a + $b`
```

Example 03 : Take Value from Terminal/user

নিচের উদাহরণে লগইন ইউজারের কাছ থেকে দুইটি ভ্যালু নেওয়া হবে এবং শেষে 'expr' (expression) কমান্ড ব্যবহার করে দুইটি ভ্যালুর ক্যালকুলেশন আউটপুট দেখানো হবে।

```
[root@node1 scripting]# vim myscript3.sh
#!/bin/bash
# this script testing for user input from terminal

echo "Hello Mr. $LOGNAME"

echo -n "Give me value-1: "
read a

echo -n "Give me value-2: "
read b

echo "your output is: `expr $a + $b`"

:x
[root@node1 scripting]# sh myscript3.sh
```

নোটঃ উপরের স্ক্রিপ্টটিতে 'root' ইউজারকে কল করে a, b ভ্যারিয়েবলের ডাটা নেওয়া হয়েছে এবং ক্যালকুলেশন শেষে আউটপুট দেখানো হয়েছে।

Bash Mathematical Operators :

দুইটি ভ্যালুর মধ্যে তুলনা করার জন্য বিভিন্ন ধরনের অপারেটর ব্যবহার করা হয়। তবে, String এবং integer ভ্যালুর মধ্যে আলাদা আলাদা অপারেটর ব্যবহার হয়।

Integer comparison :

----- List of Operator: -----	Example -----
-eq = equal to	10=10
-ne = not equal to	10!=9
-lt = less than	9<10
-le = less than equal to	9<=10
-gt = greater than	10>9
-ge = greater than equal to	10>=9

String Comparison :

'==' = equal to
'!==' = not equal to

Conditional Statements in Bash Scripts :

ডিসিশন ভিত্তিক কোনো প্রোগ্রাম রান করার ব্যাশ (Bash) স্ক্রিপ্টে ক্ষেত্রে বিভিন্ন ধরনের 'Conditional Statement' ব্যবহার করা হয়। Bash স্ক্রিপ্টের মধ্যে বিভিন্ন ধরনের Conditional Statment ব্যবহার করা হয়। যেমনঃ 'if', 'elif' (Short for else if), else, case ইত্যাদি। নিচে ব্যাশ (Bash) স্ক্রিপ্টের মধ্যে 'if', 'elif', 'else' এর ব্যবহার দেখানো হয়েছে।

Working with Conditional Staments (if) :

নির্দিষ্ট কোনো কন্ডিশন চেকিং এর ক্ষেত্রে ব্যবহার হয়। এক্ষেত্রে কন্ডিশন 'True' হলে আউটপুট দেখাবে, আর না হলে অন্য কোনো আউটপুট দেখাবে না।

```
# First form
```

```
if [condition];
```

```
then
```

```
<commands> ; (যদি কন্ডিশন True হয় তাহলে কোড এক্সিকিউট হবে)
```

fi

Second form (if-else)

কন্ডিশন 'True' হলে এক ধরনের আউটপুট দেখাবে, আর 'True' না হলে অন্য আউটপুট দেখাবে।

```
if [condition];
```

```
then
```

```
<commands> ; (যদি কন্ডিশন True হয় তাহলে কোড এক্সিকিউট হবে)
```

```
else
```

```
<commands> ; (যদি কন্ডিশন True না হয় তাহলে কোড এক্সিকিউট হবে)
```

```
fi
```

Third form (if-elif-else)

একাধিক কন্ডিশন (series of conditions) চেকিং এর সময় নিচের মত ব্যবহার করতে হবে।

```
if [condition];
```

```
then
```

```
<commands> ; (যদি প্রথম কন্ডিশন True হয় তাহলে কোড এক্সিকিউট হবে)
```

```
elif [condition];
```

```
then
```

```
<commands> ; (যদি দ্বিতীয় কন্ডিশন True হয় তাহলে কোড এক্সিকিউট হবে)
```

```
else
```

```
<commands> ; (যদি প্রথম এবং দ্বিতীয় কোনো কন্ডিশনই True না হয় তাহলে কোড এক্সিকিউট হবে)
```

```
fi
```

Example 04: Compare Positive/Negative using 'if-then-else' Statement :

নিচের উদাহরণে ইউজার কর্তৃক প্রদত্ত ভ্যালু 'Positive' না 'Negative' সেটা চেক করবে। প্রথমে 'read' কমান্ডের মাধ্যমে ইউজার কাছ থেকে ভ্যালু নিয়ে 'number' ভ্যারিয়েবলে রাখবে এবং পরবর্তীতে '0' এর সাথে তুলনা করবে। যদি ইউজার কর্তৃক প্রদত্ত ভ্যালু '0' থেকে বড় হয়, তাহলে প্রোগ্রামের আউটপুট 'The number is Positive' দেখাবে অথবা ইউজার কর্তৃক প্রদত্ত ভ্যালু '0' থেকে ছোট হলে 'The number is Negative' দেখাবে।

```
[root@node1 scripting]# vim mysrcript4.sh
#!/bin/bash
# this script for if-then-else statement check
```

```
echo -n "Enter the number: "
read number
if [ $number -ge 0 ];
then
echo "The number is positive"
else
echo "The number is negative"
fi
```

```
[root@node1 scripting]# sh mysrcript4.sh
Enter the value: 10
The number is positive
```

```
[root@node1 scripting]# sh mysrcript4.sh
Enter the value: -10
The number is negative
```

Example 05: Compare Equal & Not Equal Operator using 'if' Statement :

নিচের উদাহরণে 'read' কমান্ড ব্যবহার করে ইউজারের কাছ থেকে একটি ভ্যালু নেওয়া হবে এবং সেটা ভ্যারিয়েবল 'num' এর মধ্যে রাখবে। এরপর ইউজার কর্তৃক প্রদত্ত ভ্যালু '10' সাথে তুলনা করবে, প্রদত্ত ভ্যালু যদি '10' থেকে বড় হয়, তাহলে আউটপুট আসবে 'The Number is greater than 10' এবং প্রদত্ত ভ্যালু যদি '10' থেকে ছোট হয়, তাহলে আউটপুট 'The Number is Less than 10' দেখাবে। আর যদি কোনো Conditions না মিলে অর্থাৎ প্রদত্ত ভ্যালু স্ক্রিপ্টে উল্লেখিত ভ্যালু '10' এর সাথে মিলে যায়, তাহলে আউটপুট 'The Number equal to 10' দেখাবে।

```
[root@node1 scripting]# vim example5.sh
```

```
#!/bin/bash
```

```
# Get user input
```

```
echo -n "Enter a number: "
```

```
read num
```

```
# Check if the number is greater than 10
```

```
if [ $num -gt 10 ];
```

```
then
```

```
    echo "The number is greater than 10"
```

```
elif [ $num -lt 10 ];
```

```
then
```

```
    echo "The number is less than 10"
```

```
else
```

```
    echo "The number is equal to 10"
```

```
fi
```

```
[root@node1 scripting]# sh example5.sh
```

Example 06: UID verify from Linux System using Script :

নিচের উদাহরণে মাধ্যমে Superuser, System User এবং Regular User দের UID যাচাই করা হবে। Superuser 'root' এর ইউজারের UID '0' সুতরাং, ইউজার কর্তৃক প্রদত্ত UID '0' হলে, এট আউটপুট দিবে 'This is Root user UID', আইডি রেঞ্জ 1-999 এর মধ্যে হলে দেখাবে 'This is System User UID' এবং UID রেঞ্জ 1000+ হলে দেখাবে 'This is Regular User UID' ইউজার।

```
[root@node1 scripting]# vim example6.sh
```

```
#!/bin/bash
```

```
# If the UID is 0, if UID is 1-999 then system users, otherwise regular user
```

```
echo -n "Enter the UID: "
```

```
read uid
```

```
if [ $uid -eq 0 ]
```

```
then
```

```
    echo "This is Root user UID"
```

```
elif [ $uid -gt 0 ] && [ $uid -le 999 ]
```

```
then
```

```
    echo "This is System User UID"
```

```
elif [ $uid -gt 999 ]
```

```
then
```

```
    echo "This is Regular user UID"
```

```
fi
```

```
[root@node1 scripting]# sh example6.sh
```

Example 07: Working with String Comparison :

নিচের স্ক্রিপ্টের মাধ্যমে লিনাক্স সিস্টেমে নির্দিষ্ট কোনো ইউজার আছে কিনা, সেটা চেক করা যাবে। থাকলে আউটপুট দিবে 'User already exist' না থাকলে বলবে 'User does not exist'

```
[root@node1 scripting]# vim example7.sh
```

```
#!/bin/bash
```

```
# This script for string comparison
```

```
echo -n "Enter the username:"
```

```
read user
```

```
if [ `grep ^$user /etc/passwd` ]
```

```
then
```

```
echo "User $user already exist"
```

```
else
```

```
echo "User $user does not exist"
```

```
fi
```

```
[root@node1 scripting]# sh example7.sh
```

Class Work :

1. Create a username and set password with shell script and user must change password on next login.

```
#!/bin/bash
#Find user name if not found create new user
```

```
echo -n "Enter the user name: "
```

```
read user
```

```
if [ `grep ^$user /etc/passwd` ]
```

```
then
```

```
    echo "user already exists"
```

```
    else
```

```
        useradd $user
```

```
    sleep 2
```

```
echo "123456" | passwd --stdin $user
```

```
chage -d 0 $user
```

```
echo "New User Has been Added"
```

```
id $user
```

```
fi
```

```
[root@node1 scripting]# sh example7.sh
```

Loop এর ব্যবহারঃ

loop বলতে আমরা বুঝি, কিছু কমান্ডের সমষ্টি বা list of commands যা ক্রমাগত এক্সিকিউট হতে থাকবে ততক্ষণ পর্যন্ত যতক্ষণ পর্যন্ত Loop এর কন্ডিশন শেষ (True) না হয়। বিভিন্ন ধরনের loop এর নাম উল্লেখ করা হয়েছেঃ

- 1) for loops
- 2) While loops
- 3) Until loops

The basic syntax of a for loop is :

```
$ for <variable name> in <a list of items>; do <some command> ${variable name}; done
```

Example :

```
$ for user in jony liza boby;do echo $user;done
```

Loop Testing in Linux Terminal :

```
[root@node1 ~]# for user in rony sumon jony rana liza
> do
> echo "This is User $user"
> done
[root@node1 ~]#
```

নোটঃ উপরের কমান্ডের আউটপুটে এক এক করে প্রতিটি ইউজার প্রিন্ট করবে যতক্ষণ পর্যন্ত শেষ না হয়।

Example 08: For Loop Example :

```
[root@node1 ~]# vim example8.sh

#!/bin/bash
#for loop test

for user in rony sumon sathi liza tomal
do
    echo "My name is:" $user
done

[root@node1 ~]# sh example8.sh
```

Example 09: Print User Name using 'for loop' :

Instruction: Put some username into 'user.txt' file

```
[root@node1 ~]# sh example9.sh

#!/bin/bash
#Print User list

for user in $(cat user.txt);
do
    echo "This is" $user
done

[root@node1 ~]# sh example9.sh
```

=====

```
#!/bin/bash
#Create User using for loop
```

```
for user in $(cat user.txt)
do
    useradd $user
done
```

```
[root@node1 ~]# sh example9.sh
```

===== Thank You =====

Home work :

- ⌘ Write a shell script view your system Kernel, Memory, Disk, CPU, uptime Information.
- ⌘ Write a shell script to create a group 'sysadmin' and user 'rony' add to 'sysadmin' group
- ⌘ Create a file using 'touch command' with 5 users (user1, user2, user3, user4, user5) and create user account based on file' user list.

- 1) for loops - Series of command চালানোর ক্ষেত্রে ব্যবহার করা হয়।
- 2) While loops
- 3) Until loops - যতক্ষণ না পর্যন্ত কন্ডিশন True না হবে ততক্ষণ পর্যন্ত কোড রিপিট হতে থাকবে।

The main difference is that while loops are designed to run while a condition is satisfied and then terminate once that condition returns false. On the other hand, until loops are designed to run while the condition returns false and only terminate when the condition returns true.

⌘ while - condition should be true to execute the body

⌘ until - Condition should be false to execute the body

Numeric ranges for syntax is as follows :

```
for VARIABLE in 1 2 3 4 5 .. N
do
    command1
    command2
    commandN
done
```

```
#!/bin/bash
# For Loop Testing
```

```
for ,condition]
do
    ### code
done
```

RnD Sections

```
echo "Enter 1 To Show IP address"
echo "Enter 2 To Show MAC"

read -p "Select Your Option : " option
echo "You have select Option" $option

if [ $option == 1 ];
then
    echo "Your IP Address is:" $(hostname -I)
elif [ $option == 2 ];
then
    echo "Your MAC Address is: " $(ip addr show | grep ether | awk '{print $2}')
else
echo "Invalid Option!"
fi

#!/bin/bash
# IP address findout
for url in $(cat site.txt);
do
echo "$url IP Address is:" $(ping -c $url | grep "64 bytes from" | awk 'NR == 1'){print $5}'
done

#!/bin/ksh
# Tested with ksh version JM 93t+ 2010-03-05
for i in {1..100}
do
# your-unix-command-here
echo $i
done
```

Thank You

Class -19

Linux Backup, Archive & Compression Utilities

Why Need Archive & Compression :

- ১। ফাইল/ডিরেক্টরির সাইজ ছোট করার জন্য। (Compression)
- ২। অনেকগুলো ফাইল/ডিরেক্টরিকে একত্রে বা একটি ফাইলে রূপান্তর করার জন্য।
- ৩। ফাইল সিকিউরিটি দেওয়ার জন্য (Password, Malware attack, missing)
- ৪। ডেটা ব্যাকআপ (Archive) বা অনেক ডেটা এক জায়গা থেকে অন্য জায়গায় পাঠানোর জন্য।
- ৫। ডেটা এনক্রিপশন এবং পাসওয়ার্ড প্রটেক্টেড করার জন্য।
- ৬। ইমেইল এটাস্টমেন্ট হিসেবে ব্যবহার করার জন্য।

Archive Utility :

ডেটা আর্কাইভ করলে কোনো স্পেস কমবে না। তবে, বিভিন্ন মেটা ডাটার (inode, name, security, block) জন্য কিছু স্পেস কমতে পারে। আর্কাইভের এক্সটেনশন সাধারণত '.tar' ফরম্যাটে হয়ে থাকে অর্থাৎ ফাইলের নামের শেষে '.tar' থাকবে। উদাহরণঃ

⌘ Archive.tar

Compression/Archive Utilities :

- ⌘ Windows: zip, rar, 7zip
- ⌘ linux: .gz, .bz2, .xz optional (.zip, .rar)

Archive + Compression :

- ⌘ Backup.tar.gz
- ⌘ Backup.tar.bz2
- ⌘ Backup.tar.xz

Compression Ratio :

- ⌘ Main File: 30MB
- ⌘ Archive(tar): 30MB
- ⌘ Compression: .gz(7MB), .bz2(6MB), .xz(4MB)

Working Directory :

```
[root@node1 ~]# mkdir /linux21/day19 -p
[root@node1 ~]# cd /linux21/day19
[root@node1 day19]# cp -rv /etc . ; /etc ডিরেক্টরিকে কপি করে বর্তমান ডিরেক্টরি
'~/linux21/day19' এর মধ্যে নিয়ে আসা হয়েছে।
[root@node1 day19]# ls
[root@node1 day19]# du -sh etc ; 'etc' ডিরেক্টরির সাইজ দেখা যাচ্ছে 30MB
30M etc
```

Archive (tar):

আমরা 'tar' কমান্ড ব্যবহার করে আর্কাইভ এবং কম্প্রেশন উভয় কাজই করতে পারি এবং এটার প্যাকেজ (RPM) 'tar' সিস্টেমে ডিফল্ট ভাবে ইন্সটল থাকে।

'tar' কমান্ডের সাথে যে সকল অপশন ব্যবহার করা যাবেঃ

- c = Create
- x = extract
- t = list
- v = verbose
- f = files
- z = gzip
- j = bzip2
- J = xz
- C = change

```
[root@node1 day9]# tar -cvf etcarchive.tar etc
[root@node1 day9]# ls
[root@node1 day9]# du -sh *
30M  etc
27M  etcarchive.tar
```

নোটঃ আর্কাইভ করার পরে আমরা 'ls' কমান্ড দিয়ে দেখলাম 'etcarchive.tar' নামে লাল রঙের একটি (tar) ফাইল তৈরি হয়েছে। যেটার সাইজ '27 MB' অর্থাৎ মূল ফাইল 'etc' ডিরেক্টরি থেকে মেটাডাটার (inode, name, security, block) জন্য কিছু স্পেস কম হয়েছে।

Multiple File Archive to single File :

```
[root@node1 day19]# touch file1 file2 file3
[root@node1 day19]# tar -cvf files.tar file1 file2 file3 ; একাধিক ফাইলকে আর্কাইভ করে একটি
ফাইলে রূপান্তর করার জন্য।
[root@node1 day19]# tar -tvf files.tar ; আর্কাইভের মধ্যে কি আছে সেট দেখার জন্য
'-t' অপশন।
```

Archive extract :

আর্কাইভ থেকে ফাইল এক্সট্রাক্ট (extract) করার জন্য, প্রথমে মূল ডিরেক্টরি '/etc' রিমুভ করে নেওয়া হবে।

```
[root@node1 day19]# rm -rf etc
[root@node1 day19]# ls

[root@node1 day19]# tar -xvf etcarchive.tar
[root@node1 day19]# ll
```

Archive + Compress :

আমরা তিন (০৩) পদ্ধতিতে আর্কাইভ সাথে কম্প্রেসড করতে পারি।

⇒ Gzip (backup.tar.gz) - এক্ষেত্রে 'tar' কমান্ডের সাথে '-z' অপশন যোগ করতে হবে।

⇒ Bzip2 (backup.tar.bz2) - এক্ষেত্রে 'tar' কমান্ডের সাথে '-j' অপশন যোগ করতে হবে।

⇒ XZ (backup.tar.xz) - এক্ষেত্রে 'tar' কমান্ডের সাথে '-J' অপশন যোগ করতে হবে।

Archive + Gzip compress :

```
[root@node1 day19]# tar -czvf backup.tar.gz etc
```

```
[root@node1 day19]# ll
```

Archive + Bzip2 compress :

```
[root@node1 day19]# tar -cjvf backup.tar.bz2 etc
```

```
[root@node1 day19]# ll
```

Archive + XZ compress :

```
[root@node1 day19]# tar -cJvf backup.tar.xz etc
```

```
[root@node1 day19]# ll
```

```
[root@node1 day19]# du -sh *
```

```
29M  etc
```

```
27M  etcarchive.tar
```

```
4.6M  etcbackup.tar.bz2
```

```
6.0M  etcbackup.tar.gz
```

```
3.8M  etcbackup.tar.xz
```

Extract :

```
[root@node1 day19]# rm -rf etc
[root@node1 day19]# tar -xzvf etcbackup.tar.gz
[root@node1 day19]# ll
```

extract: (.bz2)

```
[root@node1 day19]# rm -rf etc
[root@node1 day19]# tar -xjvf etcbackup.tar.bz2
[root@node1 day19]# ls
```

extract: (.xz)

```
[root@node1 day19]# rm -rf etc
[root@node1 day19]# tar -xJvf etcbackup.tar.xz
[root@node1 day19]# ls
```

File Backup Local to Remote System :

- ⇒ FTP Server
- ⇒ NFS Server
- ⇒ Samba (CIFS) Server
- ⇒ SCP (Secure Copy) - SFTP
- ⇒ Rsync

Secure Copy (scp) from server: (From Node1)

আমরা চাইলে লিনাক্স সিস্টেম থেকে আরেক লিনাক্স সিস্টেমে ডেটা ট্রান্সফার করতে পারি। এক্ষেত্রে 'scp' কমান্ড ব্যবহার করতে হবে। ডেটা ট্রান্সফারের ক্ষেত্রে সিকিউর ফাইল ট্রান্সফার প্রটোকল SFTP (port 22) ব্যবহার করা হয়।

; লোকাল সিস্টেম (node1) থেকে ফাইল রিমোট সিস্টেমে (node2) পাঠানোর জন্য নিচের কমান্ডঃ

Secure Copy (scp) to node2 :

```
[root@node1 day19]# scp backup.tar.gz student@172.25.11.Y:/tmp ;(node2 সার্ভারে পাঠানো হয়েছে)
```

; রিমোট সিস্টেম (node2) থেকে ফাইল কপি করে লোকাল সিস্টেমে (node1) নিয়ে আসা হয়েছে।

```
[root@node1 day19]# scp student@172.25.11.Y:/etc/passwd /linux21/day19 ;(node1)
[root@node1 day19]# ls
```

File Copy using Rsync :

Here's a shortlist of features rsync offers.

- ⇒ Directory copying
- ⇒ Easy backup configuration
- ⇒ Can work over SSH
- ⇒ Can run as daemon/server
- ⇒ File permission retention

Rsync stands for the term RemoteSync. Despite the name, it can handle file synchronization remotely and locally. The term rsync is also used to refer to the rsync protocol that rsync uses for syncing.

Rsync Command structure :

```
# rsync <options> <SRC> <DST>
```

Rsync করার জন্য 'node2' সিস্টেম (CentOS 9) ব্যবহার করা হবে।

```
[root@node2 ~]# yum install rsync -y
```

Rsync within Local System Directories :

```
[root@node2 ~]# mkdir /day19/backup -p
```

```
[root@node2 ~]# cd /day19/backup
```

```
[root@node2 backup]# mkdir centos redhat
```

```
[root@node2 backup]# touch centos/file{1..3}
```

```
[root@node2 backup]# ls centos
```

```
[root@node2 backup]# rsync -av centos redhat ; centos এর সাথে 'redhat' এর sync করা হল।
```

```
[root@node2 backup]# ls redhat/centos
```

```
[root@node2 backup]# touch centos/file4
```

```
[root@node2 backup]# rsync -av centos redhat
```

```
[root@node2 backup]# ls redhat/centos
```

Rsync within Remote System Directories :

```
[root@node2 backup]# rsync -arv centos student@172.25.11.X:/tmp
```

Optional (যদি আলাদা পোর্ট নাম্বার দিয়ে পাঠানো হয়)

```
[root@node2 backup]# rsync -av --rsh='ssh -p 2222' centos student@172.25.11.X:/tmp
```

a = archive

v = verbose

r = recursively

Move to Remote Server :

```
[root@node1 ~]# cd /tmp
[root@node1 tmp]# ls
```

Working with Linux Log Files :

বিভিন্ন ধরনের লগ ফাইল (Events & Messages) সিস্টেম, অ্যাপ্লিকেশন, ইউজার কর্তৃক তৈরি হয়ঃ

- ⌘ Kernel & Boot Log
- ⌘ Application Log (http, mail, samba, crond)
- ⌘ User login info (ssh, user login)
- ⌘ Audit Log

লগ ফাইল সমূহ 'cat, tail, less, grep, head' কমান্ড দিয়ে দেখা যায়। কিছু কিছু লগ ফাইল বাইনারী অবস্থায় থাকে, এক্ষেত্রে ফাইলে এনক্রিপ্টেড অবস্থায় থাকে। ফলে, ভিউ করা যায় না। এক্ষেত্রে কমান্ড ব্যবহার করে ভিউ করতে হয়।

- * /var/log/messages: সিস্টেমের বুট এবং কার্নেল মেসেজ সম্পর্কিত লগ ফাইল জমা হয় এখানে।
- * /var/log/secure: 'SSH' সার্ভিসের সকল লগ জমা হয় এই ফাইলে।
- * /var/log/maillog: Mail সার্ভার সম্পর্কিত সকল লগ (incoming/outgoing) এখানে পাওয়া যায়।
- * /var/log/dmesg: ডিভাইস ড্রাইভার সম্পর্কিত সকল লগ এখানে জমা হয়।
- * /var/log/cron: Cronjobs সম্পর্কিত সকল লগ এখানে জমা হয়।
- * /var/log/dnf.log: লিনাক্স প্যাকেজ ম্যানেজমেন্ট সিস্টেম (YUM) এর লগ এখানে জমা হয়।
- * /var/log/boot.log: সিস্টেম বুটিং সম্পর্কিত সকল লগ জমা হয় এখানে।

সিস্টেমের সকল ডিভাইস ড্রাইভার সম্পর্কিত সকল লগ ফাইল ফাইল নিচের কমান্ড দিয়ে দেখা যাবেঃ

```
[root@node1 ~]# dmesg
[root@node1 ~]# dmesg | grep scsi
```

To show all event messages, use :

Systemd-journald saves the events and messages in a binary format that cannot be read with a text editor.

```
[root@node1 ~]# journalctl
[root@node1 ~]# journalctl -u sshd
```

System Kernel and Boot Log :

```
[root@node1 ~]# tail -f /var/log/messages
```

SSH Login Users Info :

```
[root@node1 ~]# cat /var/log/secure
```

Using Tuned :

To maximize the end-to-end performance of services, applications and databases on a server, system administrators usually carry out custom performance tuning, using various tools, both generic operating system tools as well as third-party tools. One of the most useful performance tuning tools on CentOS/RHEL/Fedora Linux is Tuned.

```
[root@node2 ~]# yum install tuned -y ; (CentOS Node)
```

After the installation, you will find following important tuned configuration files.

- * /etc/tuned – tuned configuration directory.
- * /etc/tuned/tuned-main.conf- tuned main configuration file.
- * /usr/lib/tuned/ – stores a sub-directory for all tuning profiles.

Now you can start or manage the tuned service using following commands.

```
[root@node2 ~]# systemctl start tuned
[root@node2 ~]# systemctl enable tuned
[root@node2 ~]# systemctl status tuned
```

এখন 'tunde-adm' কমান্ড ব্যবহার করে 'tuned' প্রোফাইল ম্যানেজ করা যাবে।

পূর্বে উল্লেখিত (Pre-defined) প্রোফাইল সমূহের লিস্ট দেখার জন্য নিচের কমান্ডঃ

```
[root@node2 ~]# tuned-adm list
```

বর্তমানে একটি প্রোফাইল দেখার জন্য নিচের কমান্ডঃ

```
[root@node2 ~]# tuned-adm active
Current active profile: virtual-guest
```

প্রোফাইল পরিবর্তন করার জন্য নিচের কমান্ডঃ

```
[root@node2 ~]# tuned-adm profile virtual-host
[root@node2 ~]# tuned-adm active
```

ডিফল্ট প্রোফাইল লোড করার জন্য নিচের কমান্ডঃ

```
[root@node2 ~]# tuned-adm recommend
```

প্রোফাইল বন্ধ রাখার জন্য নিচের কমান্ডঃ

```
[root@node2 ~]# tuned-adm off
[root@node2 ~]# tuned-adm active
```

===== Thank you =====

Class -20

Configure NTP Service on Linux System (CentOS)

Reference Table :

Package Name: Chrony
Daemon Name: chronyd
Port Number: 123/UDP

Step 01: Package Install :

```
[root@node2 ~]# yum install chrony -y
```

Step 02: Configure NTP Server :

⌘ NTP এর কনফিগারেশন ফাইল এডিট করে পূর্বের থাকা পাবলিক NTP সার্ভার সমূহ '#' দিয়ে কमेंট করে দিতে হবে।

```
[root@node2 ~]# vim /etc/chrony.conf
```

```
3 # pool 2.rhel.pool.ntp.org iburst -----> Red hat
```

```
8 #server 0.asia.pool.ntp.org
```

```
9 #server 1.asia.pool.ntp.org
```

```
10 #server 2.asia.pool.ntp.org
```

```
11 #server 3.asia.pool.ntp.org
```

⌘যে নেটওয়ার্কের সিস্টেম/ডিভাইস সমূহ NTP ব্যবহার করবে সেই নেটওয়ার্ক আইডি এখানে উল্লেখ করতে হবেঃ

```
25 # Allow NTP client access from local network.
```

```
26 allow 172.25.11.0/24
```

⌘ আমাদের লিনাক্স সিস্টেম টাইম সোর্স হিসেবে কাজ করবে। এইজন্য 'local startum' লাইন থেকে '#' উঠিয়ে দিয়ে আন-কমেন্ট করতে হবে। এখানে ভ্যালু 1-15 সেট করা যাবে।

```
29 local stratum 3
```

** এরপরে সেভ করে বের হতে হবে।

```
[root@node1 ~]# timedatectl
Local time: Wed 2023-06-14 22:56:09 +06
Universal time: Wed 2023-06-14 16:56:09 UTC
RTC time: Wed 2023-06-14 16:56:08
Time zone: Asia/Dhaka (+06, +0600)
System clock synchronized: no
NTP service: inactive
RTC in local TZ: no
```

Step 03: Start Chrony Service and Enable it :

```
[root@node1 ~]# systemctl enable chronyd --now
[root@node1 ~]# systemctl status chronyd
```

```
[root@node1 ~]# timedatectl
Local time: Wed 2023-06-14 22:57:03 +06
Universal time: Wed 2023-06-14 16:57:03 UTC
RTC time: Wed 2023-06-14 16:57:02
Time zone: Asia/Dhaka (+06, +0600)
System clock synchronized: no
NTP service: active
RTC in local TZ: no
```

Step 04: Configuring the firewall for chronyd :

```
[root@node1 ~]# firewall-cmd --permanent --add-service=ntp --add-service=ptp
[root@node1 ~]# firewall-cmd --reload
```

```
[root@node1 ~]# firewall-cmd --list-all
```

Configure NTP on RHEL9: Client Setup :

Step 01: Package Install :

```
[root@node2 ~]# yum install chrony
```

Step 02: Configure NTP Client :

⌘ NTP এর কনফিগারেশন ফাইল এডিট করে পূর্বের থাকা পাবলিক NTP সার্ভার সমূহ '#' দিয়ে কমেন্ট করে দিতে হবে এবং আমাদের লোকাল NTP এর সার্ভারের আইপি অথবা হোস্টনাম উল্লেখ করতে হবে।

```
[root@node2 ~]# vim /etc/chrony.conf
```

```
# pool 2.centos.pool.ntp.org iburst
```

```
4 server 172.25.11.254 iburst
```

Step 03: Start Chrony Service and Enable it :

```
[root@node2 ~]# systemctl enable chronyd --now
```

```
[root@node2 ~]# systemctl status chronyd
```

```
[root@node2 ~]# timedatectl
```

Step 04: Check Server and Peers :

```
[root@node2 ~]# chronyc activity
```

```
200 OK
```

```
1 sources online
```

```
[root@node2 ~]# chronyc sourcestats
```

Step 05: Synchronized Server Info :

```
[root@node2 ~]# chronyc tracking
```

Step 06: Check Current System Time :

```
[root@node2 ~]# timedatectl
Local time: Wed 2022-11-30 17:35:49 +06
Universal time: Wed 2022-11-30 11:35:49 UTC
RTC time: Wed 2022-11-30 11:35:49
Time zone: Asia/Dhaka (+06, +0600)
System clock synchronized: yes
NTP service: active
RTC in local TZ: no
```

```
[root@node2 ~]# chronyc sources
```

```
MS Name/IP address      Stratum Poll Reach LastRx Last sample
=====
^? 172.25.11.254        0  8  0  -  +0ns[ +0ns] +/-  0ns
```

নোটঃ উপরের আউটপুটের মত '?' চিহ্ন থাকলে , বোঝা যাবে ক্লায়েন্ট সার্ভারের সাথে যোগাযোগ করতে পারতেছে না।

```
[root@node2 ~]# chronyc sources
```

```
MS Name/IP address      Stratum Poll Reach LastRx Last sample
=====
^* 172.25.11.254        3  6  37  27  -18ns[ +449us] +/- 271us
```

===== Thank You =====

Day 20 - (RHEL 9 Tasks Scheduler using 'Cronjobs')

Module Objectives :

- ⌘ লিনাক্সে সিডিউল টাস্কের প্রয়োজনীয়তা।
- ⌘ সিডিউল টাস্ক হিসেবে Crontab এর ব্যবহার।
- ⌘ Crontab ফাইল ফরম্যাট সম্পর্কে ধারণা
- ⌘ Crontab প্রয়োজনীয় প্যাকেজ (RPMs) ইন্সটল করা।
- ⌘ বিভিন্ন ধরনের cronjob সিডিউল কনফিগার করা।

What is Cronjobs ?

সিস্টেম অ্যাডমিনের অনুপস্থিতিতে সিস্টেম কর্তৃক যে সকল কাজ গুলো সম্পাদন করা হবে, সেটা জব সিডিউল বা সিডিউল টাস্কের মাধ্যমে করতে হবে। আর এক্ষেত্রে লিনাক্সে যে পদ্ধতি ব্যবহার করা হয় তাকে বলা হয়, cronjob পদ্ধতি। cronjob পদ্ধতি ব্যবহার করে, মিনিট, ঘন্টা, প্রতিদিন, সাপ্তাহিক, মাসিক ভিত্তিতে বিভিন্ন টাস্ক (Jobs) সম্পাদন করা হয়।

Why Need Job Scheduling ?

- * সিস্টেম এবং সার্ভার কনফিগারেশন ফাইল ব্যাকআপ রাখার জন্য। যেমনঃ '/etc, '/usr' '/var'।
- * নির্দিষ্ট টাইমে ডাটা ব্যাকআপ নেয়ার জন্য। যেমনঃ '/home', /var, /backup ইত্যাদি।
- * নির্দিষ্ট টাইমে সিস্টেম শাটডাউন (Poweroff) এবং রিবুট (Reboot) করার জন্য।
- * নির্দিষ্ট টাইমে সিস্টেম (OS), কার্নেল এবং অ্যাপ্লিকেশন আপডেট করার জন্য।
- * নেটওয়ার্ক এবং সিস্টেম মনিটরিং করার জন্য।
- * নির্দিষ্ট টাইমে সিস্টেম থেকে টেম্পোরারি (Tmp) ফাইল মুছে ফেলার জন্য।
- * সিস্টেমে জমাকৃত বিভিন্ন লগ ফাইল মুছে ফেলার জন্য।
- * নির্দিষ্ট টাইমে ইউজারদের/ক্লায়েন্টের কে ই-মেইল (Mass email) নোটিফিকেশন পাঠানোর জন্য।

Reference Tables :

- ⌘ Packages (RPMs) - cronie
- ⌘ Daemon (Service) - crond
- ⌘ Application - Linux Schedule Task or Job Scheduling.

Understanding How Cron Works :

```
* * * * * <cmd>
```

```
1 2 3 4 5
```

1 = Minutes (0-59)

2 = Hour (0-23)

3 = Day of month (1-31)

4 = Months (1-12)

5 = Day of Week (0-6) here 0 or 7 is sunday

<cmd> = command to be execute

Examples of setting cron jobs :

নিচে কিছু 'crontab' এর উদাহরণ দেওয়া হয়েছে।

Example 01: Schedule a cron job every hour at Mins 30:

```
30 * * * * <cmd-to-execute>
```

Example 02: Schedule a cron job 5 minute interval :

```
*/5 * * * * <cmd-to-execute> - ৫ মিনিট অন্তর অন্তর
```

```
5 * * * * <cmd-to-execute> - ৫ম মিনিটে
```

Example 03: Schedule a cron job Everyday 03:30 PM :

```
30 15 * * * <cmd-to-execute>
```

Example 04: Schedule a cron job every hour (at minute 0) :

```
00 */1 * * * <cmd-to-execute>
```

```
00 * * * * <cmd-to-execute>
```

Example 05: Schedule a cron job Specific hours in a day :

```
00 0,9,12,15,18 * * * <cmd-to-execute>
```

Example 06: Schedule a cron job on 9AM – 5PM :

```
00 09-17 * * * <cmd-to-execute>
```

Example 07: Schedule a cron job for Twice (1st & 15th) of every month :

```
00 00 1,15 * * <cmd-to-execute>
```

Example 08: Schedule a cron job on December 31st Night 23:30:

```
30 23 31 12 * <cmd-to-execute>
```

Example 09: Schedule a cron job Friday Mid Night:

```
00 00 * * FRI <cmd-to-execute>
```

or

```
00 00 * * 5 <cmd-to-execute>
```

Class Work :

1. Daily 9.30 PM run a schedule task: ----->
2. Every monday midnight: ----->
3. Every FRI & SAT 5.00 PM ----->
4. Jan 01 & July 01 Every Year ----->
5. Every 6 hrs weekly MON-FRI days: ----->
6. Run Cron Job on December 31st – Midnight ---->
7. At 4 am on Tuesday and Thursday ----->

Cron Directories :

এছাড়া 'root' ইউজার চাইলে তার 'Cron Jobs' সমূহ '/etc/cron.d' ডিরেক্টরিতে রাখতে পারে।

There are four cron directories :

- 1) /etc/cron.hourly - Run all scripts once an hour
- 2) /etc/cron.daily - Run once a day
- 3) /etc/cron.weekly - Run once a week
- 4) /etc/cron.monthly - Run once a month

Some Cronjobs with Examples :

Step 01: Package Install :

```
[root@node1 ~]# rpm -qa | grep cronic ; প্যাকেজ (RPM) ডিফল্ট ভাবে ইন্সটল থাকে।  
[root@node1 ~]# yum install cronic -y ; যদি ইন্সটল না থাকে। 'yum/dnf' কমান্ড দিয়ে ইন্সটল করা  
যাবে।
```

Step 02: Cron Service Restart + Enable :

```
[root@node1 ~]# systemctl status crond.service ; ডিফল্ট ভাবে সার্ভিস Active এবং  
Running থাকে।  
[root@node1 ~]# systemctl restart crond.service  
[root@node1 ~]# systemctl enable crond.service
```

Crontab Command Syntax :

```
[root@node1 ~]# crontab -e - ইউজারের জন্য 'job' সেট করা।  
[root@node1 ~]# crontab -l - বর্তমান ইউজারের 'job' লিস্ট দেখার জন্য।  
[root@node1 ~]# crontab -e -u student - নির্দিষ্ট ইউজারের জন্য 'job' সেট করা।  
[root@node1 ~]# crontab -l -u student - নির্দিষ্ট ইউজারের 'job' লিস্ট দেখার জন্য।
```

```
[root@node1 ~]# crontab -r          - রিমুভ করার করার জন্য।
```

Example 01: Schedule a job every night 11.59 PM to shutdown the system :

```
[root@node1 ~]# crontab -e
59 23 * * * /usr/sbin/poweroff
```

```
[root@node1 ~]# crontab -l
```

Set the date 23.58 for job testing:

অস্থায়ী ভাবে টাইম সেট করার জন্য নিচের কমান্ডঃ

```
[root@node1 ~]# date
[root@node1 ~]# date -s "23:58:00" ; অস্থায়ী ভাবে টাইম এগিয়ে সেট করা হয়েছে।
[root@node1 ~]# date
```

```
[root@node1 ~]# watch date
```

Example 02: Take '/etc' backup in '.tar' format under '/backup' directory on every Friday at midnight:

'/etc' ডিরেক্টরি প্রতি শুক্রবার মধ্যে রাতে '/backup' ডিরেক্টরির মধ্যে backup নেওয়া হবে এবং ফাইলের নামের শেষে এক্সটেনশন হিসেবে ব্যাকআপের দিন-তারিখ উল্লেখ থাকবে।

```
[root@node1 ~]# date +%Y-%m-%d
```

```
[root@node1 ~]# mkdir /backup
```

```
[root@node1 ~]# crontab -e
```

```
00 00 * * 5 tar -cvf /backup/etc_backup_$(date +%Y-%m-%d).tar /etc
```

```
[root@node1 ~]# date
[root@node1 ~]# cal           ; verify recent Thursday
```

```
[root@node1 ~]# date MMDD2358YY
[root@node1 ~]# date
```

```
[root@node1 ~]# systemctl restart crond.service
```

```
[root@node1 ~]# watch date
```

```
[root@node1 ~]# cd /backup
[root@node1 ~]# ls
```

Example 03: Schedule a job to run every six hours in a day :

```
[root@node1 ~]# crontab -e

00 0,6,12,18 * * * 'curl https://example.com'
```

Example 04: Run a script at 01:00 am each weekday [Monday - Friday] :

```
00 01 * * 1-5 /bin/sh /backup/monitor.sh
```

Example 05: Run a cronjob December 31 at Mid Night (23:59) every year :

```
59 23 31 12 * /bin/sh /backup/backup.sh
```

এখানে '/bin/sh' হচ্ছে কমান্ড, এবং '/backup/backup.sh' হচ্ছে স্ক্রিপ্ট লোকেশন।

Restricting Access :

কোনো ইউজারকে 'crontab' এক্সেস থেকে বিরত রাখতে চাইলে, সেই ইউজারের নাম '/etc/cron.deny' ফাইলের ভিতরে দিয়ে সেভ করলে, উক্ত ইউজার আর কোনো 'crontab' কমান্ড এক্সিকিউট করতে পারবে না।

```
[root@node1 ~]# vim /etc/cron.deny
```

```
student
```

```
guest
```

নোটঃ উপরের উদাহরণে 'student' এবং 'guest' ইউজারকে 'crontab' ব্যবহার থেকে বিরত রাখা হয়েছে।

Check Cron log :

```
[root@desktop ~]# tail /var/log/cron
```

Home Work

1. Create a shell script for '/etc' backup as '.rar' format and send it '/backup' directory with 'date' extension.
2. Generate a cronjob for 'student' user that he will get a 'Greeting Text' on his Desktop at mid night.

===== Thank you =====

Day 21 - Working with RHEL Container

Manage containers :

- ⇒ Find and retrieve container images from a remote registry
- ⇒ Inspect container images
- ⇒ Perform container management using commands such as podman and skopeo
- ⇒ Build a container from a Container file
- ⇒ Perform basic container management such as running, starting, stopping, and listing running containers
- ⇒ Run a service inside a container
- ⇒ Configure a container to start automatically as a systemd service
- ⇒ Attach persistent storage to a container

Podman একটি কন্টেইনার টুলস যেটা Daemonless (Unlike Docker) পদ্ধতিতে কাজ করে।

Building a Container Lab Environment :

Container সম্পর্কিত ল্যাব ডেমো প্র্যাকটিস করতে হলে আমাদের নিচের লিস্ট অনুযায়ী সিস্টেম রেডি করতে হবেঃ

- ⌘ CPU: 2 Core (64 Bit) Intel/AMD/ARM
- ⌘ Memory: 2 GB
- ⌘ Storage: 20 GB
- ⌘ OS: CentOS/RedHat/Rocky
- ⌘ IP Address: 172.25.11.254/24

Required Packages for Container :

- ⌘ Buildah - কন্টেইনার ইমেজ (OCI or Docker Format) বিল্ড (build) করার জন্য ব্যবহার করা হয়।
- ⌘ Podman - কন্টেইনার ইমেজ রান করার জন্য ব্যবহার হয়। Podman লোকাল সিস্টেমে হতে অথবা রেজিস্ট্রি হতে ইমেজ নামাতে (pull) পারে।

⌘ Skopeo - রেজিস্ট্রি থেকে ইমেজ নামানো (Pulling) এবং পাঠানোর (Pushing) জন্য ব্যবহার করা হয়।
এছাড়াও ইমেজ ইন্সপেকশন, ইমেজ ট্রান্সফার (Registry to Registry), ইমেজের তথ্য
সংগ্রহ করার জন্য কাজ করে।

Install Container tools with Podman :

```
[root@node1 ~]# yum install container-tools -y  
[root@node2 ~]# yum info container-tools
```

Check Podman Version :

```
[root@node1 ~]# podman version  
[root@node1 ~]# podman -v  
[root@node1 ~]# podman help  
[root@node1 ~]# podman info
```

Podman এর জন্য প্রয়োজনীয় কমান্ডঃ

```
[root@node1 ~]# podman build  
[root@node1 ~]# podman run  
[root@node1 ~]# podman pull  
[root@node1 ~]# podman start  
[root@node1 ~]# podman stop  
[root@node1 ~]# podman ps  
[root@node1 ~]# podman rmi  
[root@node1 ~]# podman rm  
[root@node1 ~]# podman restart  
[root@node1 ~]# podman images  
[root@node1 ~]# podman search
```

☞ নিচের ফাইলটি এডিট করে রেজিস্ট্রি লোকেশন পরিবর্তন করা যাবেঃ

* Registry Location: /etc/containers/registries.conf

- ⇒ Docker Hub
- ⇒ RedHat Registry
- ⇒ CentOS Registry
- ⇒ Fedora Registry

registries:

search:

- registry.access.redhat.com
- registry.redhat.io
- docker.io

Listing Available Images :

নিচের কমান্ডের মাধ্যমে লোকাল সিস্টেমে কি কি ইমেজ নামানো (Pull) আছে সেটা জানা যাবে।

```
[root@node1 ~]# podman images
REPOSITORY TAG          IMAGE ID   CREATED   SIZE
```

রেজিস্ট্রিতে ইমেজ খোঁজার (Search) জন্য নিচের কমান্ডঃ

```
[root@node1 ~]# podman search httpd
```

```
[root@node1 ~]# podman search nginx
```

```
[root@node1 ~]# podman search ubuntu
```

```
[root@node1 ~]# podman search centos
```

```
[root@node1 ~]# podman search centos | grep 'centos7'
```

docker.io/publici/httpd

httpd:latest

```
docker.io/manasip/httpd
docker.io/centos/httpd
docker.io/centos/httpd-24-centos7:latest
```

Image inspect to Docker Registry using skopeo:

```
[root@node1 ~]# skopeo inspect docker://registry.access.redhat.com/rhel7
[root@node1 ~]# skopeo inspect docker://docker.io/library/ubuntu
```

Pull an Image to local :

```
[root@node1 ~]# podman pull docker.io/library/nginx
[root@node1 ~]# podman pull docker.io/library/ubuntu
[root@node1 ~]# podman pull docker.io/library/httpd
[root@node1 ~]# podman pull docker.io/centos/httpd-24-centos7
```

Available pulled images :

সকল লোকাল (Pulled) কন্টেইনার ইমেজ লিস্ট দেখার জন্য নিচের কমান্ডঃ

```
[root@node1 ~]# podman images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
docker.io/library/httpd	latest	1132a4fc88fa	9 days ago	148 MB
docker.io/library/ubuntu	latest	ba6accedd29	2 weeks ago	75.2 MB
docker.io/library/nginx	latest	87a94228f133	2 weeks ago	138 MB
docker.io/centos/httpd-24-centos7	latest	5ae4c76f6a3e	3 months ago	361 MB

Image ID ব্যবহার করে ইমেজ সম্পর্কে বিস্তারিত জানার জন্য নিচের কমান্ডঃ

```
[root@node1 ~]# podman inspect <image id>
```

লোকাল Image রিমুভ করার জন্য নিচের কমান্ডঃ

```
[root@node1 ~]# podman rmi <image id>
```

```
[root@node1 ~]# podman rmi -a
```

রানিং কন্টেইনারের (Container) সেই লিস্ট দেখার জন্য নিচের কমান্ডঃ

```
[root@node1 ~]# podman ps          (শুধুমাত্র রানিং কমান্ডের লিস্ট দেখা যাবে)
```

```
[root@node1 ~]# podman ps -a      (Running + Exited)
```

লোকাল ইমেজ ব্যবহার করে কন্টেইনার রান করার জন্য নিচের কমান্ডঃ

```
[root@node2 ~]# podman run -d <image id or image name>
```

```
[root@node2 ~]# podman run -d httpd    (httpd কন্টেইনার ব্রাকগ্রাউন্ডে রান করবে)
```

```
[root@node2 ~]# podman run -it ubuntu /bin/bash    (Ubuntu run in interactive Shell)
```

-d: detached (Run container in background and print container ID)

-t: tty (Allocate a Pseudo-TTY for container)

-i: interactive

-p: --publish (define port number to the host)

```
root@0430d193aa3d:/# ls
```

```
root@0430d193aa3d:/# cat /etc/os-release
```

```
NAME="Ubuntu"
```

```
root@0430d193aa3d:/# uname -r
```

```
root@0430d193aa3d:/# cat /etc/resolv.conf
```

```
root@0430d193aa3d:/# free -m
```

```
root@0430d193aa3d:/# cat /proc/cpuinfo
root@0430d193aa3d:/# cat /etc/passwd
root@0430d193aa3d:/# top
root@0430d193aa3d:/# useradd robin
root@0430d193aa3d:/# passwd robin
root@0430d193aa3d:/# id robin
root@0430d193aa3d:/# cat /etc/passwd
root@0430d193aa3d:/# exit
```

Install packages :

```
root@0430d193aa3d:/# apt-get -y update
root@0430d193aa3d:/# apt-get -y install vi
root@0430d193aa3d:/# apt-get -y install net-tools
```

```
[root@node2 ~]# podman ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
302398173a6a	docker.io/library/httpd:latest	httpd-foreground	23 minutes ago	Exited (0) 10 seconds ago

⇒ কন্টেইনারের মধ্যে নতুন করে প্রসেস টেবিল (PID) তৈরি হবে, যেটা হোস্ট সিস্টেম থেকে দেখা যাবে না।

⇒ কন্টেইনার মধ্যে আলাদা নেটওয়ার্ক ইন্টারফেস তৈরি হবে এবং DHCP এর মাধ্যমে আইপি (Private) অ্যাড্রেস পাবে।

⇒ কন্টেইনার কর্তৃক নতুন ফাইল সিস্টেম তৈরি হবে।

সরাসরি রেজিস্ট্রি থেকে ইমেজ রান করা যায়। এক্ষেত্রে ইমেজ লোকাল সিস্টেমে সংরক্ষিত থাকে, এবং পরবর্তীতে কন্টেইনার রান করা যায়।

```
[root@node1 ~]# podman run -d nginx (এখানে '-d=detach' ব্যাকগ্রাউন্ডে রান করার জন্য +
কন্টেইনার আইডি প্রিন্ট করবে)
```

```
[root@node1 ~]# podman images
```

```
[root@node1 ~]# podman ps -a
```

ব্যাকগ্রাউন্ডে রানিং থাকা অবস্থায় কোনো কন্টেইনারের মধ্যে প্রবেশ/প্রসেস রান জন্য নিচের কমান্ডঃ

```
[root@node2 ~]# podman exec -it <Conatiner ID> /bin/bash
```

```
[root@node2 ~]# podman exec -it 2e83a5cff7ea /bin/bash
```

```
root@85b7824a0534:/usr/local/apache2#
```

কন্টেইনার Stop, Start এবং Remove করার জন্য নিচের কমান্ডঃ

```
[root@node1 ~]# podman stop <container id>
```

```
[root@node1 ~]# podman ps -a
```

```
[root@node1 ~]# podman start <container id> (Exited Conatiner also Tested)
```

```
[root@node1 ~]# podman rm <container id>
```

```
[root@node1 ~]# podman ps -a
```

নোটঃ সরাসরি ইমেজ থেকে কন্টেইনার স্টার্ট করা যাবে না। শুধু মাত্র স্টপ (Exited) কন্টেইনার থেকেই প্রসেস স্টার্ট করা যাবে। কন্টেইনার রিমুভ করার পূর্বে এটাকে স্টপ করে নিতে হবে।

Project: Run a service inside a container (Building website using Apache httpd)

```
[root@node1 ~]# mkdir /opt/web1
```

```
[root@node1 ~]# mkdir /opt/web2
```

```
[root@node1 ~]# mkdir /opt/web3
```

```
[root@node1 ~]# podman run -dt --name web1_info -p 8081:80 -v /opt/web1:/usr/local/apache2/htdocs/ httpd
```

```
[root@node1 ~]# podman run -dt --name web2_net -p 8082:80 -v /opt/web2:/usr/local/apache2/htdocs/ httpd
```

```
[root@node1 ~]# podman run -dt --name web3_com -p 8083:80 -v /opt/web3:/usr/local/apache2/htdocs/ httpd
```

```
[root@node1 ~]# podman ps
```

Create Sample WebPage :

```
[root@node1 ~]# cd /opt/web1
```

```
[root@node1 web1]# vim index.html
```

```
<html>
<head>
<body bgcolor="#f25dfd">
<h1 align="center"> ##### welcome to example.info  ### </h1>
</body>
</head>
</html>
```

নোটঃ উপরের মত অন্য দুইটি ডিরেক্টরি 'web2 & web3' এর মধ্যে আলাদা আলাদা দুইটি 'index.html' ফাইল তৈরি করতে হবে।

Try Test Pages :

```
http://172.25.11.254:8081
```

```
http://172.25.11.254:8082
```

http://172.25.11.254:8083

```
[root@node1 ~]# reboot
[root@node1 ~]# podman ps
[root@node1 ~]# podman ps -a
[root@node1 ~]# podman start <servie Name>
```

Start a container on boot with Podman and Systemd :

```
[root@node1 ~]# podman generate systemd --new --name web1_info
[root@node1 ~]# podman generate systemd --new --name web1_info >
/etc/systemd/system/web1.service

[root@node1 ~]# ls -al /etc/systemd/system/web1.service
-rw-r--r--. 1 root root 821 Nov 30 07:49 /etc/systemd/system/web1.service

[root@node1 ~]# systemctl start web1.service
[root@node1 ~]# systemctl enable web1.service
[root@node1 ~]# systemctl status web1.service

[root@node1 ~]# reboot
[root@node1 ~]# podman ps
```

অপারেটিং সিস্টেমের ইমেজ সমূহ, যেমনঃ ubuntu, centos7 রান করলে, ইমেজ 'exited' মোডে থাকবে। নিচের কমান্ডের মাধ্যমে 'ubuntu' ইমেজকে সরাসরি (রেজিস্ট্রি হতে) রান করে শেলে ঢোকা হয়েছেঃ

RnD Section

```
[root@node2 /]# cat /etc/os-release  
[root@node2 /]# yum install httpd -y  
[root@node2 /]# cd /var/www/html/  
[root@node2 /]# vi index.html
```

HELlo World !!!

```
[root@node2 /]# /usr/sbin/httpd
```

```
[root@node1 ~]# podman run -it --network host centos:7  
--network - Connect a container to a network
```

===== Thank you =====