Linux @ the command line – Refresher Session

by AUB – IT

What we will cover?
Day 1

- **Shell**
- **File System & Management**
- **System Information**
- **Symbols**
- **Filters**
- **Hotkeys**
- **File Editors**

What we will cover?
Day 2

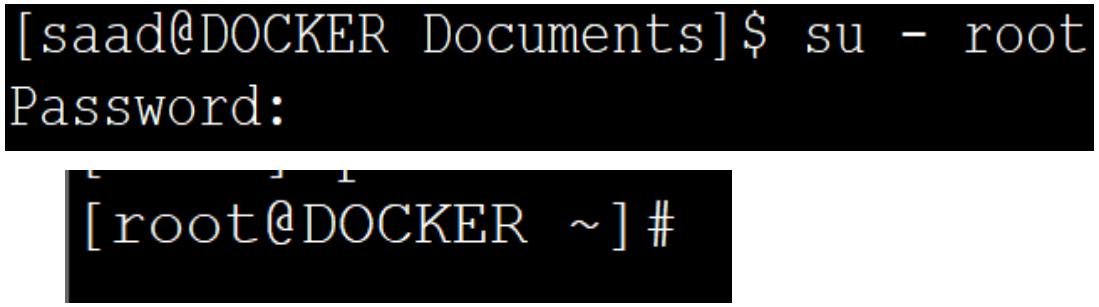
- **Network**
- **Process Management**
- **File Compression**
- **Package Management**
- **Services Management**
- **Firewall Management**
- **Date and Time Management**
- **Java, JDK and Certificate Tools**

File System & Management:

su,sudo,bash,ls,mkdir,cd, pwd,mv,cp,ln,touch,cat,mount,du,df,chmod,find, locate,hostnamectl,useradd,adduser

Bash Shell:

- Username
- System name
- Name of current directory
- ~ is special (home)
- $ normal user (non-admin)



- Command: su - root

Will switch user to root.

“root” user is like the “Administrator” user compared to Windows O/S and donated by the shell prompt “#”

File System & Management:

su,sudo,bash,ls,mkdir,cd,pwd,mv,cp,ln,touch,cat,mount,du,df,chmod,find, locate,hostnamectl,**useradd,adduser**

```
[saad@DOCKER ~]$ pwd
/home/saad
```

- Command: pwd (Print working directory)

```
[saad@DOCKER Documents]$ ls
file1.txt  file2.txt
```

- Command: ls (list directory)

To list hidden files and directories type the command: *ls -la*

```
[saad@DOCKER Documents]$ ls -lh
total 4.0K
-rw-rw-r-- 1 saad saad 29 Nov  6 19:05 file1.txt
-rw-rw-r-- 1 saad saad  0 Nov  6 18:57 file2.txt
```

- Command: ls -lh (can combine options -lh)

```
[saad@DOCKER ~]$ ls Videos/
video1.mpg
```

- Listing other directories

Try: ls --help will give you more info about available options

The Shell & Filesystem navigation

cd:

```
[saad@DOCKER ~]$ pwd
/home/saad
[saad@DOCKER ~]$ cd ..
[saad@DOCKER home]$ pwd
/home
[saad@DOCKER home]$
```

- ".." means "the directory above this one"

- $ cd ~

(Will take you to Home directory, as tilda ~ symbol represent home)

- Directories start with /<directory name>
- Example: /, /etc , /home
- $ cd .

(“.” means “this directory”)

- $ cd /usr

(will change directory to /usr)

- Tab completion: Press *Tab in order to complete the command and path without typing the whole command/path only if found.*

The Shell & Filesystem navigation
bash, ./:

- Bash: In order to run a program written in Bash (official shell for Linux) you may either run using the command *bash* or ./

Example: *bash* script.sh or ./script.sh

Bash and Shell scripts have the following lines on top of the file:

#!/bin/sh

#!/bin/bash

Bash and Shell scripts extension is “.sh”

File System & Management:

su,sudo,bash,ls,mkdir,cd,pwd,mv,cp,ln,touch,cat,mount,du,df,chmod,find, locate,hostnamectl,useradd,adduser

- Hostnamectl: *hostnamectl* will display information about system name, OS name, Machine ID, Kernel version and Architecture. Also it changes the name of the server.

Display info: *hostnamectl status*

Change server name (no reboot!): hostnamectl set-hostname name

- Link: *ln* is a command-line utility for creating links between files. By default, the ln command creates hard links. To create a symbolic link, use the -s (--symbolic) option.

Example: ln -s /usr/local/src/jetty-src/bin/jetty.sh jetty

Try: <command> --help or man <command> will give you more info & help about available options

File System & Management:

su,sudo,bash,ls,mkdir,cd,pwd,mv,cp,ln,touch,cat,mount,du,df,chmod,find, locate,hostnamectl,**useradd,adduser**

- Useradd: *useradd* will create a new user on the system.

Example to create a new user and add it to the same group:

useradd --system --home-dir /usr/local/src/jetty-src --user-group jetty

- *Adduser: adduser will create a new user similar to useradd with no additional options.*

Example: adduser jetty

**useradd is more granular and for manual specific requirement upon creating a new user.*

Try: <command> --help or man <command> will give you more info & help about available options

File System & Management:

su,sudo,bash,ls,mkdir,cd,pwd,mv,cp,ln,touch,cat,mount,du,df,chmod,find, locate,hostnamectl,useradd,adduser,export,clear,set

- Export: *export* is bash shell BUILTINS commands, which means it is part of the shell. It marks an environment variables to be exported to child-processes.

Example: *export JAVA_HOME=/usr/lib/jvm/java*

Try: <command> --help or man <command> will give you more info & help about available options

- Clear: *clear* command will flush the output of the terminal screen for a clean view.
- Set*: set* command will display the current loaded environment variables to the users from /etc/bashrc (system-wide), /etc/profile.d (system-wide), /home/<user>/.bashrc (local)

I need to note system-wide means loaded to all users.

***also .bashrc starts with a "." which means it's a hidden file, use command *ls -la* to list hiddent files.

File permissions chmod & chown:

```
[saad@DOCKER Documents]$ ls -lh file1.txt
-rwxrw-r-- 1 saad saad 29 Nov  6 19:05 file1.txt
```

- chmod: change mode file/directory permissions.

-rw-rw-r-- donates:

r	read
w	write
x	execute

- chmod u+rwx file1.txt will allow file1.txt to be read,write,executed by the user.
- file/directory permission is divided to 3 parts in order:

Users, groups, others.

Try: chmod --help will give you more info about available options

File permissions chmod & chown:

drwxrwxrwx

d = Directory
r = Read
w = Write
x = Execute

chmod 777

rwx|rwx|rwx
Owner|Group|Others

7	rwx
6	rw-
5	r-x
4	r--
3	-wx
2	-w-
1	--x
0	---

- *Chmod: "--- --- ---"filename.txt*

  To represent rwx triplet use 4+2+1=7

  To represent rw- triplet use 4+2+0=6

  To represent r– triplet use 4+0+0=4

  To represent r-x triplet use 4+0+1=5

To only give full permission to user, use it as follows:

chmod 0700 file.txt or chmod 700 file.txt

File permissions chmod & chown:

```
[saad@DOCKER Documents]$ ls -lh file1.txt
-rwxrw-r-- 1 saad saad 29 Nov  6 19:05 file1.txt
```

- chown: change ownership of file/directory.
- chown -R saad:saad <directory> will change the owner recursively to the directory and its sub-directories.
- chown saad:saad file1.txt will change file1.txt ownership/groupship to desired user/group.
- Need to be root (administrator on the system) to change to other owner.

Try: chown --help will give you more info about available options

File & Directory management cp,mv,rm,mkdir,touch,find,locate,cat:

- cp: will copy files & directories if you add option “-r” example: cp -r <dir1> <dir2>
- mv: will move files & directories and also it can rename!
- rm: will remove files only! If you add option “-rf” it will remove directories.
- mkdir: will make new directory.
- touch: will create new file

Try: <command> --help or man <command> will give you more info & help about available options

File & Directory management
cp, mv, rm, mkdir, touch, find, locate, cat :

- Find: search files and directories on the system.

Example: *find /etc -name hosts*

- Locate: need to be installed it uses database to search for files and directories fast!

Commands: *updatedb,locate <file>*

- Cat: prints output of a file.

Example: *cat <file1.txt>*

Try: <command> --help or man <command> will give you more info & help about available options

Symbols:
Pipe and redirections |, >, >>

- Pipe |: | is a command in Linux that lets you use two or more commands such that output of one command serves as input to the next. Example: cat file.txt | grep “hello”
- > : Redirect the input to another file.
- >>: Adds additional input appending the file.

Example 1 “>”: echo “This is a test” > newfile.txt

It will add the line: This is a test to the newfile.txt.

Example 2 “>>”: echo “This is a new line” >> newfile.txt

It will append a new line (This is a new line) to the newfile.txt

Try: <command> --help or man <command> will give you more info & help about available options

Filters:
grep, more, less, head, tail

- **Grep**: *grep* search for a string of characters in a specified file.

Example: grep “hello” <file1>

- **More**: *more* command is used to view the text files in the command prompt, displaying one screen at a time in case the file is large (For example log files).

- **Less:** *less* is most useful for viewing the content of large files or the results of commands that produce many lines of output.
- **Head**: *head* it prints the first 10 lines of the specified files.
- **Tail:** *tail* returns the last ten lines of each file that it is given. It may also be used to follow a file in real-time and watch as new lines are written to it.

Try: <command> --help or man <command> will give you more info & help about available options

- # Hot keys

- *CTRL + C*

When you have a process running in a terminal window, you can kill that process with the Ctrl + c combination.

- *CTRL+ALT+DELETE*

If all else fails and you just need to reboot, you can hit this combination to instantly start the reboot process.

- *CTRL + R*

When you hit Ctrl + r, you are prompted to enter a character (or string of characters). You'll have returned to you any previously issued command with that character or combination within.

File Editors
vim and nano:

- vim: Is a powerful text editor that most Linux admins & users use all the time.
- Basic usage, example: vim file1.txt

Movement: <Arrow Keys> or **<h>,<j>, <k>,<l>,<b>,<e>,<0>,<$>,<gg>,<G>**

Add text: press letter*: **i* (insert text/edit)**

Exit command line: **press ESC (exit to VIM command line)**

Change/Delete/Replace: **<x>,<cw>,<dw>,<dd>,<r>,<R>**

Copy/Paste: **<v>,<P>,<u>,<y>,<yy>**

Search/Replace: **</>,<n>,<N>,<:%s/'regex'/'replacement'/g>**

Save/Quit: **<:q>,<:w>,<:q!>**

Try: <command> --help or man <command> will give you more info & help about available options

File Editors
vim and nano:

- Nano: Nano text editor is pre-installed on most Linux distros.

Open files with nano type: nano <filename>

Edit: You can edit anywhere and press ctrl+x to save.

- User friendly text editor and easy to work on.

Try: <command> --help or man <command> will give you more info & help about available options

Process Management CLI:
ps,top,kill,bg,fg

- Ps: *ps* will print the list of processes – combine with options “-ef”, “aux” for listing all information regarding current running processes and by who on the system.
- Top: top same as ps but with interactive live output of the processes.
- Kill: *kill* command kills the PID (process ID) of the process. (kill –l) for listing kill signals (most popular is “-9” will terminate the process).
- Bg: *bg* (background) will send the process to run it background and return you to the shell, donated by “&” ex: vim &
- Fg: *fg* (foreground) brings the bg process back to the commandline.

Try: <command> --help or man <command> will give you more info & help about available options

Network Tools:
ssh,scp,ping,telnet,netstat,nslookup,wget,nc,curl

- ssh: *ssh is gateway to connect to the server, it uses tcp port 22 such as putty or native linux ssh tool.*
- scp: *scp* and winscp are tools to upload and download files to/from server.
- ping: *ping checks if the server is alive by sending request and receiving replies.*
- telnet: *telnet used to check if port is open on specific host.*
- *Netstat: displays the connection from/to and opened ports on the system.*
- Netcat: *nc* utility to check for open server ports with option “-vz”
- Nslookup : *nslookup will check server name to what ip address.*
- *Wget: wget* utility for downloading files from the Internet.
- Curl: *curl* is a tool to transfer data to or from a server, using any of the supported protocols (HTTP, FTP)

Try: <command> --help or man <command> will give you more info & help about available options

System Information: hostname,w,whoami, man,which,free,df,echo,date,uname

- Hostname: *hostname* shows the server name.
- W: *w* shows who is currently logging into the system with uptime.
- Whoami: *whoami* command will show your current logged in user.
- Man : *man* give descriptive help information about any given command.
- Which: *which* command identifies the location path of the executable binary that launches when you issue a command to the shell.
- Free: *free* utility to check RAM memory total/consumed.
- Df : *df utility to check the amount of free/consumed disk space.*
- *Echo: echo* is a command that outputs the strings it is being passed as arguments.
- *Date: date displays the current date & time on the system.*
- *uname: "uname –a" gives you information about the kernel version*

Try: <command> --help or man <command> will give you more info & help about available options

File compression:
tar, tar.gz, unzip, zip

- Tar: *tar* (Tape archive(r)) compress files and directories hence reduces file size. Extension can be .tar or .tar.gz files.

Example Create tar file: *tar -cvf* directry1.tar /home/user/directory1

Will compress all files inside directory1 to a create directory1.tar

Example Extract tar file: *tar -xvf* directory1.tar

Will extract the compressed files of directory1.tar to the current directory.

Example Extract tar.gz file: tar -xzf file.tar.gz

Will extract the compressed files of file.tar.gz to the current directory.

- Zip: *zip* with “-r” will create a zip file.
- Unzip: *unzip* a zipped file.

Try: <command> --help or man <command> will give you more info & help about available options

Package Management: yum,apt,dnf,rpm,dpkg

- Yum (CENTOS,RHEL,SUSE): *yum* is the primary tool for getting, installing, deleting, querying, and managing rpm packages.

Example install wget utility: yum install wget

Example update/upgrade: yum update;

- APT (Debian, Ubuntu): *apt* is a command-line utility for installing, updating, removing, and otherwise managing deb packages on Ubuntu, Debian.

Example install wget: *apt install wget* or *apt-get install wget*

Example update/upgrade: apt-get update or apt upgrade

- RPM*: rpm is used to install/remove .rpm files.*
- *Dpkg: dpkg is used install/remove .deb files.*

****dnf new package management is the future of yum, same options as yum applies.*

Try: <command> --help or man <command> will give you more info & help about available options

Services Management:
systemctl and service

Systemctl: *systemctl* command is a utility which is responsible for examining and controlling the systemd system and service manager. It is a collection of system management libraries, utilities and daemons which function as a successor to the System V init daemon.

Example: To enable on system startup apache webserver:

systemctl enable httpd

Example: To start apache webserver:

systemctl start httpd

Example: To check the status of apache webserver:

systemctl status httpd

Note that you can also use the "stop" switch to stop the webserver and "restart" switch to restart the webserver.

Try: <command> --help or man <command> will give you more info & help about available options

Services Management:
systemctl and service

Service: *service* command is used to run a System V init script. Usually all system V init scripts are stored in /etc/init.d directory and service command can be used to start, stop, and restart the daemons and other services under Linux.

Chkconfig: *chkconfig* command is used to list all available services and view or update their run level settings for startup/shutdown.

Example: To Add and Enable on system startup Jetty webserver:

chkconfig --add jetty; chkconfig jetty on

Example: To start Jetty webserver:

service jetty start

Example: To check the status of Jetty webserver:

service jetty status

Note that you can also use the "stop" switch to stop the webserver and "restart" switch to restart the webserver.

Try: <command> --help or man <command> will give you more info & help about available options

Firewall Management:
iptables, firewalld

- **Iptables:** *iptables* is a command line interface used to set up and maintain tables for the Netfilter firewall for IPv4, included in the Linux kernel. The firewall matches packets with rules defined in these tables and then takes the specified action on a possible match.

Basic commands for iptables:

List applied rules: *iptables -L*

Allow incoming web traffic on port 8080*: iptables -A INPUT -p tcp --dport 8080 -j ACCEPT*

Block incoming web traffic on port 8080*: iptables -A INPUT -p tcp --dport 8080 -j DROP*

Clear iptables rules*: iptables -F*

Try: <command> --help or man <command> will give you more info & help about available options

Firewall Management:
iptables, firewalld

- **Firewalld:** *firewalld is a firewall management solution available for many Linux distributions which acts as a frontend for the iptables packet filtering system provided by the Linux kernel.*

Basic commands:

Allow incoming https service: firewall-cmd -permanent --add-service=https

Block incoming https service: firewall-cmd -permanent --remove-service=https

Viewing firewall rules: firewall-cmd --list-all

*Note that you must enable/start firewalld service before you start using it.

Enable Firewalld: systemctl enable firewalld

Start Firewalld: systemctl start firewalld

Stop Firewalld: systemctl stop firewalld

Status Firewalld: systemctl status firewalld

Try: <command> --help or man <command> will give you more info & help about available options

Date and Time Management: chronyd, ntpd, date, timedatectl

- **Chrony:** chronyc is a flexible implementation of the Network Time Protocol (NTP). It is used to synchronize the system clock from different NTP servers, reference clocks or via manual input.

Chrony comes with two programs:

chronyc – command line interface for chrony

chronyd – daemon that can be started at boot time

Configuration file is found at /etc/chrony.conf

Enable Chrony: systemctl enable chronyd

Start Chronyd: systemctl start chronyd

Status Chronyd: systemctl status chronyd

Check Chrony time sources: chronyc sources -v

Try: <command> --help or man <command> will give you more info & help about available options

Date and Time Management: chronyd or ntpd, date, timedatectl

- **NTPD: *ntpd, ntp***

The Network Time Protocol (NTP) is a protocol used to synchronize computer system clock automatically over a networks. The machine can have the system clock use Coordinated Universal Time (UTC) rather than local time, uses udp port (123).

NTPD comes with two programs:

ntpq – command line interface for chrony

ntpd – daemon that can be started at boot time

Configuration file is found at /etc/ntpd.conf

Enable NTPD: systemctl enable ntpd

Start NTPD: systemctl start ntpd

Status NTPD: systemctl status ntpd

Check NTPD time sources: ntpq –p or systemctl status ntpd

Try: <command> --help or man <command> will give you more info & help about available options

Date and Time Management: chronyd or ntpd, date, timedatectl

- **NTPD: *ntpd, ntp***

The Network Time Protocol (NTP) is a protocol used to synchronize computer system clock automatically over a networks. The machine can have the system clock use Coordinated Universal Time (UTC) rather than local time, uses udp port (123).

NTPD comes with two programs:

ntpq – command line interface for chrony

ntpd – daemon that can be started at boot time

Configuration file is found at /etc/ntpd.conf

Enable NTPD: systemctl enable ntpd

Start NTPD: systemctl start ntpd

Status NTPD: systemctl status ntpd

Check NTPD time sources: ntpq –p or systemctl status ntpd

Try: <command> --help or man <command> will give you more info & help about available options

Date and Time Management: chronyd or ntpd, date, timedatectl

- **Date:** *date* displays the current date & time on the system.
- **Timedatectl***: timedatectl* command is a new utility which comes as a part of the systemd system and service manager, a replacement for old traditional date command used in sysvinit daemon based Linux distributions.

Check status of date and timezone: *timedatectl status*

List time zones on the system*: timedatectl list-timezones*

Change time zone of the system*: timedatectl set-timezone "Asia/Beirut"*

To Set time only: *timedatectl set-time 15:58:30*

To Set date only the format of date in YY:MM:DD (Year, Month, Day)*: timedatectl set-time 20151120*

Try: <command> --help or man <command> will give you more info & help about available options

Javas: Oracle JDK, OpenJDK, Amazon Corretto, and Certificate Tools:

- **Oracle JDK**: is a non-free Java Development Kit for enterprise systems, it provides updates and support for Enterprise Applications.
- **OpenJDK:** The biggest difference between OpenJDK and Oracle JDK is licensing. OpenJDK is completely open source Java with a GNU General Public License. Oracle JDK requires a commercial license under Oracle Binary Code License Agreement. But there are many other differences within support and cost, too.
- **Amazon Corretto:** is a reliable build of OpenJDK with the assurance of long-term support provided at no cost to you. Corretto fixes or mitigates a problem we found running OpenJDK. Amazon also plans to apply urgent fixes (including security) when they are available and ready to use, outside of the regular quarterly cycle. Amazon runs Corretto internally on thousands of production services.

Basic Java commands:

Check version: java –version; java

Try: java -help or man <command> will give you more info & help about available options

Javas: Oracle JDK, OpenJDK, Amazon Corretto, and Certificate Tools:

- **Openssl:** Is a cryptography software library or toolkit that makes communication over computer networks more secure. The OpenSSL program is a command-line tool for using the various cryptography functions of OpenSSL's crypto library from the shell. It is generally used for Transport Layer Security(TLS) or Secure Socket Layer(SSL) protocols and creates keys and csr for requesting SSL/TLS Certificates.
- Basic used commands:

Display version: *openssl version*

Create RSA Private Key: *openssl genrsa -out private.key 2048*

Create new Private Key and CSR*: openssl req -nodes -newkey rsa:2048 -keyout custom.key -out custom.csr*

It will ask for the details like country code, state and locality name, Organization name, your name, email address, etc. And after entering all the details it will generate 2 files one with the CSR extension and the other with key extension representing CSR and private key respectively.

Try: openssl -help or man <command> will give you more info & help about available options

Javas: Oracle JDK, OpenJDK, Amazon Corretto, and Certificate Tools:

- **Openssl cont'd:**

Send the CSR to Certificate Authority to obtain Certificate and import along along with the key and Intermediate/Root Certificate to webservers example: Apache, Apache Tomcat, Jitty, Nginx.

Check validity of the certificate (prints out sha1 key): *openssl x509 -in federation-cert.pem -fingerprint -sha1 –noout*

Read crt/cer certificate file: *openssl x509 -noout -text -in WebCertificate.crt (or .cer)*

Read private key file*: openssl rsa -noout -text -in private.key*

Check certificate expiry date: *openssl x509 -noout -in certificate.cer -dates*

Try: openssl --help or man <command> will give you more info & help about available options