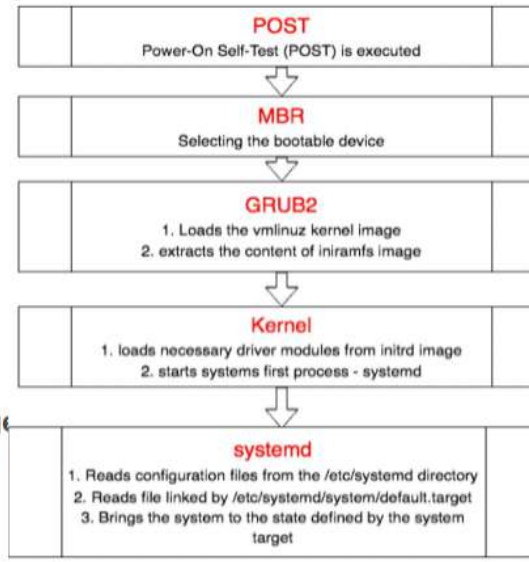
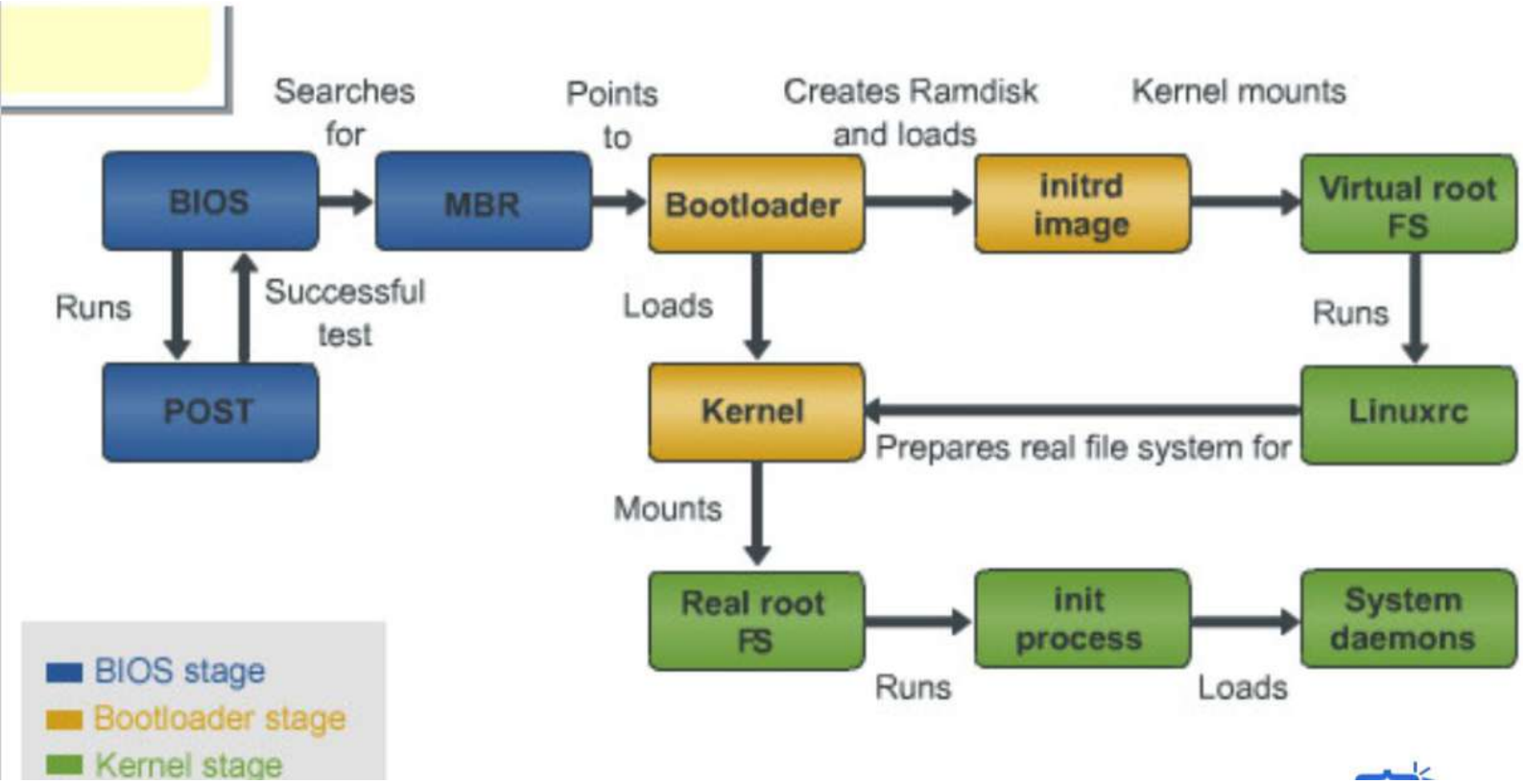
# LINUX BOOTING PROCESS

**The following steps summarize how the boot procedure happens in RH**

1. **The computer's BIOS performs POST.**
2. **BIOS reads the MBR for the bootloader.**
3. **GRUB 2 bootloader loads the vmlinuz kernel image.**
4. **GRUB 2 extracts the contents of the initramfs image.**
5. **The kernel loads driver modules from initramfs.**
6. **Kernel starts the system's first process, systemd.**
7. **The systemd process takes over. It:**
    - **Reads configuration files from the /etc/systemd directory**
    - **Reads file linked by /etc/systemd/system/default.targe**
    - **Brings the system to the state defined by the system target**
    - **Executes /etc/rc.local**




Searches for
Points to
Creates Ramdisk and loads
Kernel mounts
BIOS
MBR
Bootloader
initrd image
Virtual root FS
Runs
Successful test
Loads
Runs
POST
Kernel
Linuxrc
Prepares real file system for
Mounts
Real root FS
init process
System daemons
Runs
Loads
BIOS stage
Bootloader stage
Kernel stage

# Linux Commands Cheatsheet

## Basic File Operations

| | |
|---|---|
| **ls -lh file** | Display file permissions, size, owner etc |
| **touch file** | Creates an empty file |
| **cp file1 file2** | Copy file1 to file2. File2 can be a directory |
| **mv file dir** | Move a file to a directory |
| **mv file1 file2** | Rename file1 to file2 |
| **rm file** | Delete a file |
| **ls -lah** | List all the contents in a directory |
| **mkdir data** | Creates a directory data |
| **cp -r dir1 dir2** | Copy dir1 and its contents to dir2 |
| **rm -rf dir** | Delete a directory and its contents |
| **pwd** | Print current working directory |
| **stat file** | Display attributes of files and directories |
| **wc file** | Count bytes, words, lines in a file or STDIN |
| **file file** | Identify (guess) the type of a file. |
| **type cd** | Find out whether cd binary is built-in , alias or external binary file |

## File Viewing

| | |
|---|---|
| **cat file.txt** | Print the contents of a text file |
| **tac file.txt** | Prints text in reverse |
| **more file.txt** | View large text files one page at a time |
| **less data.txt** | Same as more but with more features |
| **head -n 5** | View the first 5 line of a text file |
| **tail -n 5** | View the last 5 line of a text file |
| **nl file** | View text files with their lines numbered |
| **strings file** | Display text that's embedded in a binary file |

## Print Text

| | |
|---|---|
| **echo "Hello World"** | Print Hello World on the standard output |
| **printf "%5d\n" 42** | Print formatted text on standard output |
| **yes "Hello World"** | Print repeated text on the standard output |
| **seq 1 5** | Print a sequence of numbers from 1 to 5 |
| **clear** | Clear the terminal screen or window |

## File Search

| | |
|---|---|
| **locate file** | Searches for files and directories |
| **which cd** | Searches the location of the cd binary |
| **whereis ls** | Find ls' binary docs, and source files |
| **find /data -name hello.txt** | Searches for "hello.txt" in the /data directory |

## Directory Traversal

| | |
|---|---|
| **cd or cd ~** | Navigate to the user's home directory |
| **cd ..** | Navigate to the parent directory |
| **cd -** | Switch to the previous working directory |
| **cd /** | Navigate to the root directory |
| **cd /tmp** | Changes the current directory to /tmp |

## Disk Management

| | |
|---|---|
| **df -h** | Report file system disk space usage |
| **du -h /home** | Estimate file space usage in the home dir |
| **fdisk -l** | List available partitions on a disk |
| **cfdisk** | Create partitions |
| **lsblk** | List block devices |
| **mount /dev/sda /mnt** | Mount /dev/sda partion to /mnt directory |
| **umount /mnt** | Unmount the mounted partition in /mnt dir |
| **findmnt** | Displays if about all mounted filesystems |
| **fsck /dev/sda** | Check a disk partition for errors |

## Secure Shell (SSH)

| | |
|---|---|
| **ssh traw@10.1.3.1** | Remote login to 10.1.3.1 |
| **ssh-agent -t rsa** | Generate SSH rsa key pair |
| **ssh-copy-id** | Copy ssh public key to a remote host |
| **sshpass** | Non-interactive ssh password auth tool |

## File Permissions

| | |
|---|---|
| **chmod +x** | Set execute permissions to a file |
| **chmod u+s script.sh** | Set SUID permissions to a file |
| **chmod g+s dir** | Set SGID permissions to a directory |
| **chmod +t dir** | Set Sticky Bit permissions to a directory |
| **chgrp devops file.txt** | Changes file.txt group onwer to devops |
| **chmod 644 script.sh** | Set the file perms to be read/write for the owner, and read-only for group and others |
| **chown traw:sys file** | Changes file owner to traw and group owner to sys |
| **umask 022** | Sets the default perms for newly created files to 644 and for directories to 755 |

## History

| | |
|---|---|
| **sudo !!** | Execute the previous command with sudo |
| **^cat^tac** | Replace previous cat command with tac |
| **history** | Display command line history |
| **!$** | Last argument of the previous command |
| **!50** | Execute the 50th command in history |

## Process Management

| | |
|---|---|
| **ps** | Display a snapshot of running processes |
| **ps aux** | Display all processes of all users |
| **top** | Display real-time view of running processes |
| **pgrep firefox** | Find the process ID of firefox |
| **pidof firefox** | Find the process IDof firefox |
| **kill 6732** | Terminate a process with PID of 6732 |
| **killall proc** | Kills all processes named 'proc' |
| **pkill firefox** | Terminates the firefox process |
| **bg** | Resumes suspended jobs in the background |
| **fg** | Brings a suspended job to foreground |
| **jobs** | List active jobs in the current shell |
| **renice 12 PID** | Changes priority of process with given PID |
| **pstree** | Displays a tree of running processes |

## Networking

| | |
|---|---|
| **ping sysxplore.com** | Sends ICMP packets to sysxplore.com |
| **ip addr** | Displays all network interfaces information |
| **ifconfig** | Shows network interfaces configuration |
| **whois sysxplore.com** | Displays domain's registration information |
| **route** | Display the routing table |
| **ss** | Display information about network sockets |
| **netstat** | Displays network information and statistics |
| **dig sysxplore.com** | Queries DNS, provides domain's DNS info |
| **wget <url>** | Download file from the specified url |
| **curl sysxplore.com** | Retrieves sysxplore.com home page |

## Compression/Archives

| | |
|---|---|
| **tar -cf backup.tar /home** | Creates a tar archive of /home dir |
| **tar -xf backup.tar** | Extract files from "backup.tar" archive |
| **tar -zcvf data.tar.gz /home** | Creates compressed archive of /home |
| **gunzip data.gz** | Uncompress data.gz file |
| **zip -r data** | Zip the data directory |
| **unzip data.zip** | Unzip the data.zip file |
| **gzip data** | Compresses "data" into "data.gz", original is removed |

## User Group Management

| | |
|---|---|
| **groups** | Print the group membership of a user |
| **groupadd devops** | Create a new group called devops |
| **groupdel devops** | Deletes the devops group |

## User Management

| | |
|---|---|
| **id** | Displays the user's UID, GID, and groups |
| **whoami** | Displays who is currently logged in |
| **finger traw** | Print information about user traw |
| **useradd -u 1002 traw** | Creates a new user traw with a specific UID |
| **userdel traw** | Deletes the user account named traw |
| **chfn traw** | Change a user's personal information |
| **usermod -aG sudo traw** | add user traw to the sudo group |
| **gpasswd -a traw sudo** | add user traw to the sudo group |
| **gpasswd -d traw sudo** | Remove user traw from the sudo group |
| **passwd** | Change user password |
| **passwd traw** | Change user traw's password |
| **chsh -s /bin/zsh** | Change user shell to zsh |
| **su james** | Switch to user james |

## Access Control Lists

| | |
|---|---|
| **getfacl file** | Display ACL permissions of a file or directory |
| **setfact -m u:traw:r-x file** | Set read/execute ACL perms for the user traw |
| **setfact -m g:sysops:r-w file** | Set read/write ACL perms for the group sysops |
| **setfact -x u:traw file** | Remove user traw ACL permissions |
| **setfact -x u:devops file** | Remove the group devops ACL perms |
| **setfact -b file** | Remove all ACL perms and keep default file permissions |

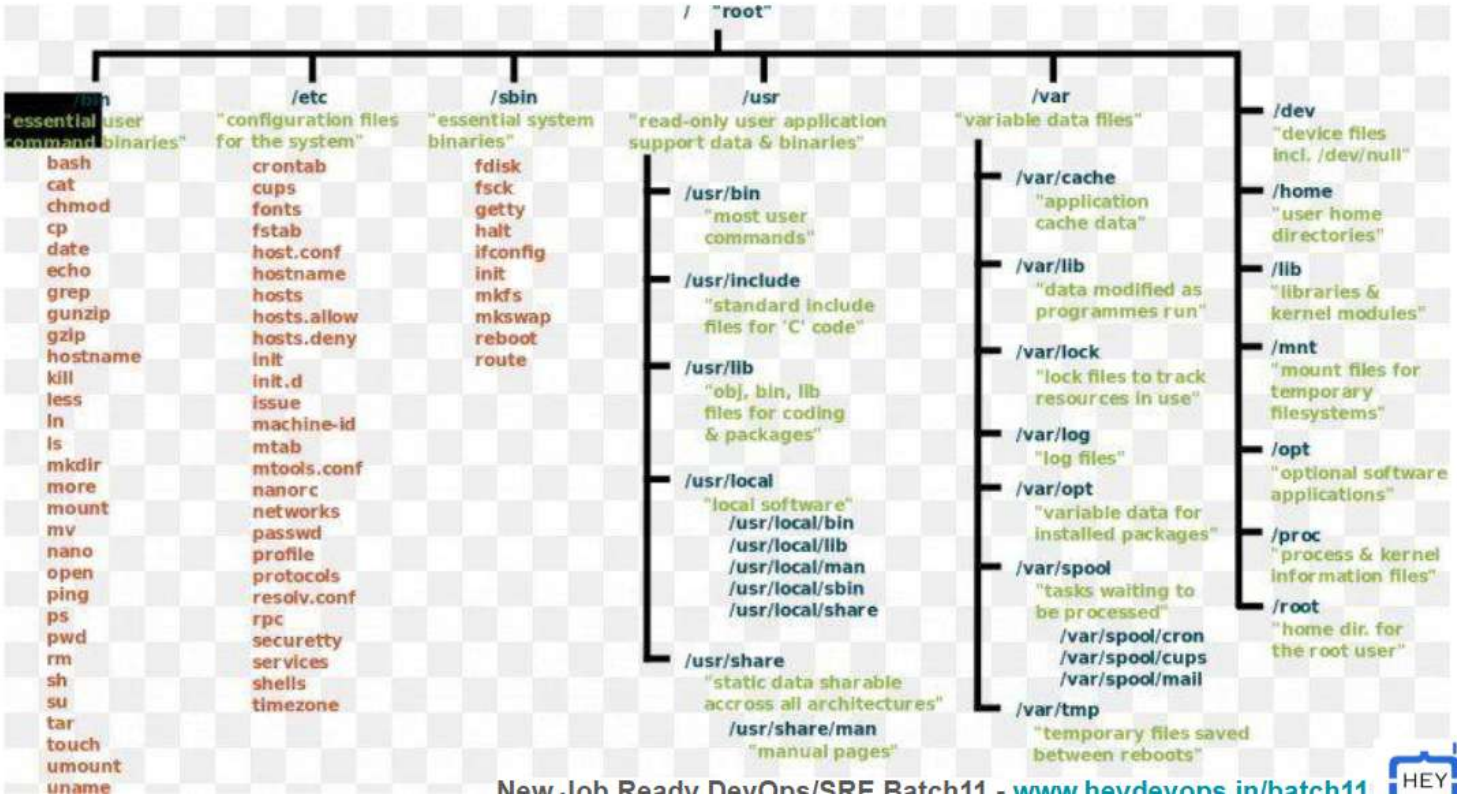
## File Transfer

| | |
|---|---|
| **scp file.txt user@host:/remote/dir** | Copies file.txt to remote host's specified directory |
| **rsync -a ~/ubuntu /backup/** | Synchronizes content from source directory to destination directory ,preserving attributes |
| **rsync -a /var/www/web/ user@rhost:/data/backup/** | Synchronizes local directory to remote, preserving attributes |

## Text Manipulation

| | |
|---|---|
| **grep "linux" file.txt** | Search for the word Linux in file.txt |
| **tr "a-z" "A-Z" <file** | Translate lowercase chars to uppercase |
| **rev < file.txt** | Print file.txt contents in reverse |
| **sort file.txt** | Sort lines of text by various criteria |
| **uniq file.txt** | Print only unique lines in file.txt |
| **vimdiff file1 file2** | Line-by-line comparison of two files in vim |
| **diff file1 file2** | Comparison of tw |
| **awk '{print $1}' file.txt** | Print the first co |
| **sed 's/cat/bat/g' file** | Substitute all cat |

# LINUX COMMANDS/FOLDERS AT ONE PLACE



# RHCSA Real Time Scenarios

## Yum Configuration Files

The main configuration file for yum is **/etc/yum.conf**. Configuration files that define repositories are in the **/etc/yum.repos.d** directory. An example of /etc/yum.conf follows here:

| | |
|---|---|
| vi /etc/yum.repos.d/rhcelab.repo | As we know repository configuration files are stored in ***/etc/yum.repos.d/*** directory with an extension .repo, So we executed this command to create the necessary configuration file for repository. |
| [rhcerepo] | This is the label of repository. Usually a repository file contains configuration for multiple repositories. In that case label is used as identifier of repository. |
| name=rhcerepo | This configuration value is used to set the name of repository. |
| baseurl=file:///rhcelab/repo | This configuration value defines the location of rpm files. |
| enabled=1 | This key defines the state of repository. If value is set to **1** then repository is enabled. If value is set to **0** then repository is disabled. |
| gpgcheck=0 | This key defines whether the integrity of package should be check or not. If value is set to **1**, integrity will be checked. If value is set to **0**, integrity will not be checked. |
| :wq | We used vi editor to create the file. In vi editor, the command: wq is used to save and quit from file. |

# Yum Repo

We create a .repo file within /etc/yum.repos.d using a text editor. In this example, we will create the repository file for MySQL 5.7

Step1 :
cd /etc/yum.repos.d/

Step2:
vim mysql57-community.repo
[mysql57-community]
name=MySQL 5.7 Community Server baseurl=http://repo.mysql.com/yum/mysql-5.7-community/el/7/$basearch/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

Step3:
yum-config-manager mysql57-community [ Validate the yum repository ]

Step4:
yum install mysql

# Disk Partitioning in Linux

We cannot create files and directories directly in the partition, before we use a partition for data storage we need to create a file system on it. File system is a logical container that is used to store the files and directories.

**Why do we need it?**

- To upgrade Hard Disk (to incorporate a new Hard Disk into the system)
- Dual Booting (Multiple Operating Systems on the same system)
- Efficient disk management
- Ensure backup and security.
- Work with different File Systems using the same system.

**SWAP Space**

Swap space is the special space in hard disk that is used as a temporary memory. This space can be allocated as a separate swap partition, LVM partition or as a file (*file is used only to extend the available swap space*). Swap space is used only if a shortage of physical memory occurs. In shortage situation system moves recently unused data from memory to swap space. When requires, system moves back this data from swap to memory. This is the convenient way to improve kernel memory usage.

**LVM (Logical Volume Manager)**

Classical partition scheme is fixed in nature. It means, once created partition size cannot be changed later. We are not allowed to add additional space in a partition which is filled up with data. Same way we cannot shrink a partition which has a lot of unused free space. LVM not only solves this issue but also provides several other advantages over the classical approach. LVM is flexible in nature. We can shrink or grow a partition as per requirement.

# SWAP Partition HandsOn

Create a swap partition

fdisk /dev/xvda
Press n [ N for new ]
+512 M
t –> For type of partition 19 number is for SWAP
w [quit]
partprobe /dev/xvda3 [ To let the kernel know about partition ]

mkswap /dev/vda3 [ To use the swap partition ]

mount –a [To check for errors]
vi /etc/fstab [For permanent mount]
/dev/vda3 swap swap

swapon -s

## ifconfig

Used to find network details, nitialize an interface, assign IP address, enable or disable an interface.

## ip

Latest and more powerful version of ifconfig. The utility is used for displaying and manipulating routing, network devices, interfaces.

## traceroute

Network troubleshooting utility for tracing the full path/route of packet from your local system to another network system.

## ping

It is used to check the connectivity between two hosts/nodes on a Local Area Network or Wide Area Network. It makes use of the ICMPs to make communicate with end nodes.

## netstat

Netstat command stands for Network statistics . It displays information about different interface statistics, including open sockets, routing tables, and connection information.

## ss

The ss command is a replacement for netstat command. This command gives more information in comparison to the netstat. It is also faster than netstat as it gets all info from kernel userspace.

## dig

Dig stands for domain internet gropper is ar simple DNS lookup utility, that is used to query DNS related info such as A Record, CNAME, MX Record etc. It mainly deals with debug DNS related problems.

## route

Used to displays and manipulate IP routing table for your system.

## nslookup

This is also another command-line utility to query DNS servers both interactively and non-interactively. It is used to query DNS resource records (RR).

## host

The host command displays domain name for given IP address or vice-versa. It also performs DNS lookups related to the DNS query.

## arp

The command arp stands for Address Resolution Protocol. It allows us to view or add content into kernel's ARP table.

## iwconfig

Similar to ifconfig, but is dedicated to the wireless interfaces. The command iwconfig configures a wireless network interface. You can view and set basic wi-fi details like SSID and encryption.

## hostname

The hostname command allows us to set and view /show system's hostname. A hostname is the name of any computer that is connected to a network that is uniquely identified over a network.

## whois

The whois command displays information about a website's record. You may get all the information about a website regarding its registration and owner's information.

## tracepath

It is similar to traceroute command, but it doesn't require root privileges. By default, it is installed in Ubuntu. If it's not found in your system you have to install it using your system package manager.

## curl

The curl (Client URL) command is mostly used to transfer data over the network and supports various protocols including HTTP, FTP, IMAP, and many others.

## wget

It is used to download files using HTTP, HTTPS, FTP Protocols. It provides the ability to download multiple files, resume downloads, download in the background, etc.

## mtr

It is a combination of ping and traceroute utilities and is mainly used for network diagnostics and gives live look at network response and connectivity.

## iftop

The iftop (Interface TOP) is often used by system admins to monitor stats related to bandwidth and can also be used as a diagnostic tool when you're having issues with the network.

## tcpdump

The tcpdump is a packet sniffing and analyzing utility used to capture, analyze and filter network traffic.

## iperf

The iperf is an open-source utility written in C allowing users to perform network performance measurement and tuning.

## ethtool

ethtool is a command-line utility for querying and modifying network interface controller parameters and device drivers.

## scp and sftp

SCP and SFTP are both file transfer protocols, but they have different functionalities. SCP only allows file transfer, while SFTP allows file access, transfer, and management.

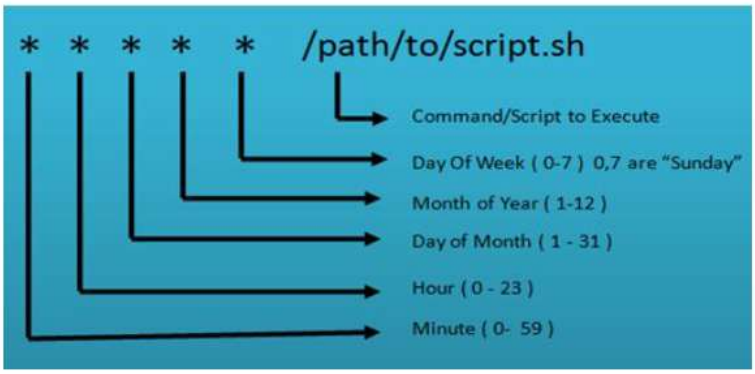
## rsync

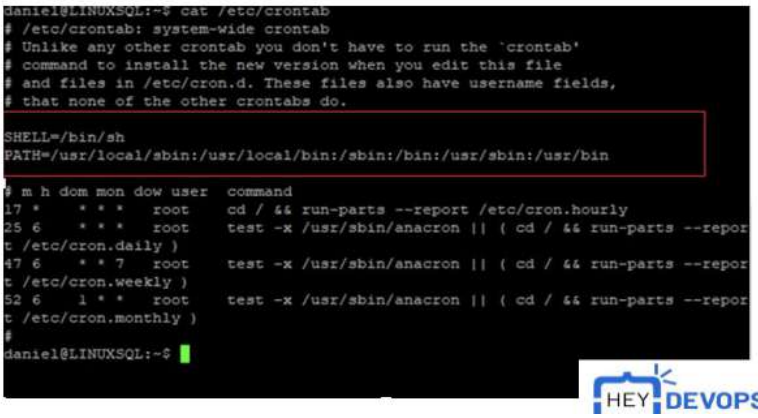
rsync is a fast and versatile command-line utility for synchronizing files and directories between two host over an ssh tunnel.

# Linux Crontab format

crontab -e -> To edit the file
crontab -l -> To list down the user cron tasks

| Description | Command |
| --- | --- |
| Cron command to do the various scheduling jobs.<br>Below given command execute at 7 AM and 5 PM daily. | `0 7,17 * * * /scripts/script.sh` |
| Command to execute a cron after every 5 minutes. | `*/5* * * * *  /scripts/script.sh` |
| Cron scheduler command helps you to execute the task on every Monday at 5 AM. This command is helpful for doing weekly tasks like system clean-up. | `0 5 * * mon  /scripts/script.sh` |
| Command run your script on 3 minutes interval. | `*/3 * * * * /scripts/monitor.sh` |
| Command to schedule a cron to which executes for a specific month.<br>This command to run tasks run in Feb, June and September months. Sometimes we need to schedule a task to execute a select monthly task. | `* * * feb,jun,sep * /script/script.sh` |
| Command to execute on selected days. This example will run each Monday and Wednesday at 5 PM. | `0 17 * * mon,wed /script/script.sh` |




HEY DEVOPS