LINUX

RAMPUP SCRIPTS :

1. **Log monitoring: Search single key word in apache/nginx or any application log file, and generate a new file with time stamp when the key work occurred in the file**

**Location : /var/log/httpd**

Vi log.sh

echo "THIS SCRIPT WILL PRINT THE LINE WHEREVER THE WORD ERROR OCCUR IN THE APACHE HTTPD ERROR.LOG FILE AND ALSO SAVE THE OUTPUT IN THE NEW FILE NAMED AS IMPETUS-I0086-DATE.LOG"

grep -i "error" /var/log/httpd/error\_log> Impetus-I0086-`date +"%d-%m-%Y"`.log

echo "output in Impetus-I0086-`date +"%d-%m-%Y"`.log"

cat Impetus-I0086-`date +"%d-%m-%Y"`.log > /var/log/httpd/log.txt

cat /var/log/httpd/log.txt

**2. Monitoring of process: Monitor memory utilization of a process periodically and generate log on screen. Inline Input process id and time interval in seconds**

echo "process monitoring"

top

top> mem.txt

read -p "Enter your process id : " pid

echo "pid is : $pid"

echo "memory utilized by the given pid is:"

grep " mem.txt >mem.logecho "Errors saved in mem.log file"

cat mem.logrep "$pid" mem.txt >mem.log

echo "Errors saved in mem.log file"

cat mem.log

**3. Monitoring of Port: Write a shell script to find the process (pid, pname) listening on a given port**

**Vi Port.sh**

echo "port monitoring"

sudo ss -lptn

sudo ss -lptn> port.txt

read -p "Enter your port number : " port

echo "port is : $port"

echo "processes listening to the given port are as follows:"

grep "$port" port.txt >port.log

echo "processes saved in port.log"

cat port.log

**4. Write a shell script for log rotate , i.e. if log of a file size go beyond a limit, create an archive file with time stamp and reduce the log size**

**Vi.rotate.sh**

echo "log rotate"

du -b /var/log/httpd/mem.txt

filename=./mem.txt

maxsize=300

filesize=$(stat -c%s "$filename")

echo "Size of $filename = $filesize bytes."

if (( filesize > maxsize )); then

echo "needs archival"

cp mem.txt rotate.txt

cat rotate.txt

tar cf rotate.tar rotate.txt

else

echo "no need to archive"

fi

**5.** **A shell script find 0 sized files and delete.**

**vi zero.sh**

echo "find all file with size zero and delete them"

cd /var/log

find /var/log -type f -size 0 -delete

ls -ltr

echo "files are deleted with size zero"

**6. A shell script to avoid Ctrl + C key input to abort shell script.**

**Controlc.sh**

#!/bin/bash

trap '' 2

echo "This is a test. Hit [Ctrl+C] to test it..."

sleep 20

trap 2

**Write a shell script to capture CPU utilization of entire system and a specific process. Inline parameters for scripts are process id and monitoring interval in seconds.**

**Vi utilization.sh**

**!/bin/bash**

echo "cpu and process utilization where Inline parameters for scripts are process id and monitoring interval in seconds."

echo "total cpu utilization is :"

top

top> utilize.txt

echo "user and cpu utilization in percentage is:"

iostat

echo "now enter the process id to get the exact utilization by a specific process: "

read pid

echo "process id is :$pid"

grep "pid" utilize.txt

cat utilize.txt

**Using cURL command write a shell script to verify pivot.impetus.co.in up and running.**

**Vi up.sh**

#!/bin/bash

curl –Is https://pivot.impetus.co.in/digite/Request?Key=framework\_main**&ContainedURL=inbox\_summary&**MenuSubmitted=true&MenuSelection=home | head -n 1

**Create a script which search files which are larger than X MB (inline input) and search in a path (inline input) recursively and prompt for deletion.**

**Vi mb.sh**

echo "script to delete files recursivly if the size of the file exceeds then the input given"

read -p "Enter path: " $path

ls -lh $path

read -p "enter size: "$size

find $path -size -20k -exec rm -i {} \;

**11. Write a shell script which copy file(s) from a directory SRC to DST and then validate both the destination are sync in then delete files from SRC folder. Also add this script in cron job for to run on 1300 hrs every day.**

**Vi copy.sh**

echo "script to copy the contents from src to des , sync them , and then delete the files from source folder "

echo "contents in src and destination before copying "

cd /var/log/httpd/src

ls -ltr

cd

cd /var/log/httpd/des

ls -ltr

cp -r /var/log/httpd/src/. /var/log/httpd/des

echo "contents in src and destination after copying "

cd /var/log/httpd/src

ls -ltr

cd

cd /var/log/httpd/des

ls -ltr

echo "sync the contents of src and des "

rsync -avu /var/log/httpd/src/ /var/log/httpd/des

echo "after sync"

cd /var/log/httpd/src

ls -ltr

cd

cd /var/log/httpd/des

ls -ltr

echo "delete the contents of src"

find /var/log/httpd/src -size -1 -exec rm -i {} \;

**script to run the cron job on daily basis : under cd /etc/cron.daily**

**vi cron.sh**

#!/bin/bash

/var/log/httpd/proc.sh

/var/log/httpd/log.sh

/var/log/httpd/port.sh

/var/log/httpd/copy.sh

**./cron.sh**

**crontab -e**

**1 \* \* \* \* bash /var/log/httpd/copy.sh**

**1 \* \* \* \* bash /var/log/httpd/port.sh**

**1 \* \* \* \* bash /var/log/httpd/proc.sh**

**1 \* \* \* \* bash /var/log/httpd/log.sh**

* **Introduction** : architecture , Terminal process and host, shell types ,
* **User and groups** : whoami, who, tty, ps, sudo and su, uname, passwd, chmod, chown, chgrp, last, wc, more, less, cp, head, tail, diff. , tty ,. Ps , termcap terminfo , ,am , news , uname , logname , who , whodo , ps , passwd , clear , mesg , id , ,
* **Files** : cd , pwd , mkdir , rmdir , mv , mvdir , cp , chown , chgrp , chmod , ls , wc , more , pg , head , tail, cat , more , less , cut , sort , diff , vi , nano , Rsync , Link (in link , unlink ), gaining privledegs , tac
* **Sort and filter**: types , grep , sed , awk
* **I/O :** ilters , pipes , search sort and compare files , Redirection / indirection
* **Packgae** : RPM utility. , APT-GET , yum , System Services overview start, restart
* **Network** : ipfocnifg, , traceroute, ping, ping6, dig, nslookup, lsof , network files , IP allocation., Routes in Linux telnet, ssh, scp, rlogin, Ping, netstat, sftp, ftp , Established SSH login using public key.
* **Process and disk mgt** : kill,nice,jobs,fg,bg,nohup,df,du,bdf,mount/unmount, crontab which, cksum, history, file, shutdown, poweroff, reboot, mdfilesum
* **Monitoring** : top , vmstat , mpstat , psrstat , ps and its options , telnet ,Tcpdump, sar
* **Communication** : SSH , Password Less SSH Authentication , SFTP , SCP , TELNET
* **Group management** : Users and Group , Add/ delete User and group , configure , id, useradd, userdel, usermod, groupadd, groupdel, groupmod, chage
* **Scripts**

Introduction

* LINUX is a family of open source unix like os based on the Linux kernel  an operating system kernel first released on September 17, 1991 by Linus Torvalds
* Popular Linux distributions include Debian ,  Fedora, and Ubuntu
* Anamika@impetus-1250
* Anamika is username , impetus 1250 is hostname , $ end of cmd prompt
* An instance of a running command is known as a **process** Each process in Linux has a **process id** (**PID**
* To chek hostname : hostname , /proc/sys/kernel/hostname
* Architecture Of Linux
* Linux Operating System has primarily three components:
* **Kernel** − Kernel is the core part of Linux. It is responsible for all major activities of this operating system. It consists of various modules and it interacts directly with the underlying hardware. Kernel provides the required abstraction to hide low level hardware details to system or application programs.
* **System Library** − System libraries are special functions or programs using which application programs or system utilities accesses Kernel's features. These libraries implement most of the functionalities of the operating system and do not requires kernel module's code access rights.
* **System Utility** − System Utility programs are responsible to do specialized, individual level tasks.
* **Hardware layer** − Hardware consists of all peripheral devices (RAM/ HDD/ CPU etc).
* **Kernel** − It is the core component of Operating System, interacts directly with hardware, provides low level services to upper layer components.
* **Shell** − An interface to kernel, hiding complexity of kernel's functions from users. The shell takes commands from the user and executes kernel's functions.
* **Utilities** − Utility programs that provide the user most of the functionalities of an operating systems.

Shell and its types

* **Shell** is an interface to kernel, hiding complexity of kernel's functions from users. The shell takes commands from the user and executes kernel's functions.
* **Bash** : Bash stands for Bourne Again Shell and it is the default shell
* **TASK** : Command line editing , Job Control , Unlimited size command history , Shell Functions and Aliases ,Unlimited size Indexed arrays
* **Tcsh** is enhanced C shell, it can be used as a interactive login shell and shell script command processor.
* **TASK** : C like syntax , Command-line editor , Spelling correction ,Job control
* **Ksh**  is a complete, powerful, high-level programming language and also an interactive command language just like many other Unix/GNU Linux shells.
* **Zsh** is designed to be interactive and it incorporates many features of other Unix/GNU Linux shells such as bash, tcsh and ksh.
* **TASK** : Filename generation , Startup files , Login/Logout watching , Closing comments , Concept index , Variable index ,Functions index
* Users and groups
* "**who**" : Report which users are logged in to the system .
* options : -a , all users on your system
* SYNTAX : WHO , WHO -a
* **W** : add additional info related to user , syntax : w
* **Whoami**: It displays the username of the current user when this command is invoked.
* Syntax : whoami
* **tty**which displays information related to **terminal**. The **tty**command of terminal basically prints the file name of the terminal connected to standard input.
* Syntax : Sudo tty
* Option: -s , print nothing just exit status
* **Ps** : list down all the process which are currently running in a machine.
* Ps –ef , ps –aux : list processin full format
* -u displays the process that belongs to a specific username. ; ps –ef anamika
* -p/--ppid :  list of processes with a particular PPID. Ps –ef –ppid 9576
* **Sudo** : let you to execute the command as root, an user can also execute a command as any other user, if they have the permission to do it.
* sudo /sbin/service/httpd restart
* options : -k clear sudo cache credentials , -l list commands that a user can execute , -n no password prompt , -v update sudo cache credentials
* **'su**' is used to switch from one account to another.
* su - linux
* **Uname** : reports basic information about a computer software and hardware.
* Uname -a all , -s kernel name , -r kernel version , -m h/w name , -p cpu type , -o OS details
* sudoers.d file : location /etc/sudoers.d
* ALL(ALL:ALL) ALL -> HOST(USER:GROUP) CMDS -> vagrant ALL(ALL:ALL) NOPASSWD:ALL
* **chmod** : used to change the access mode of a file.
* chmod <refernce><operator><mode> file
* reference :  distinguish the users to whom the permissions apply u,g,o,a
* operator : specify how the modes of a file should be adjusted.  + , - , =
* mode : permissions are to be granted or removed from the specified classes. , r,w,e
* **Chown** : change the user and group ownership of a given file, directory or link.
* chown <options>user:group file -> chown –R ansible:rampup createuser.yml
* **Chgrp**:change the group of a file /SYMLINK
* chgrp <option> <newgroup> file , *sudo chgrp -h root symlink , option : -R*
* *Umask :* new file's permissions may be *restricted* in a specific way by applying a permissions "mask" called the **umask**. The **umask** command is used to set this mask, or to show you its current value.
* Umask <-s> [mask]
* **Umask u-x , g=r,o+w**

User and group management

* configure , id, groupadd, groupdel, groupmod,
* **Useradd** : create new accounts in Linux , **useradd <options>Anamika**
* **Usermod**  modify the existing accounts in linux , **usermod <options> anamika**
* **Userdel** :  delete local account in linux **, userdel <options> anamika**
* **Passwd** : assign password to local accounts or users. **, passwd Anamika**
* **Chage** : view & modify users password expiry information
* change password at first login by using command ‘**chage -d 0 <username>‘.**
* **chage -l <username>** command to view the user’s password expiry info.

Files

* **Ls** : list files
* OPTIONS : -r in reverse , -l with attributes and permission , -lh in readable , -a hidden ,
* -R recursively , -F /in the end , -n uid and gid display ,
* **Cat** : show file content , create file .
* OPTIONS : -n for line no , -e for eof ,
* EX: Cat test , cat > test , cat test.txt|more ,
* OPERATORS : test>test1 overwrite , test>> test1 append
* **Touch** : create a empty file
* OPTIONS : - a change the access time , -c if the file does not exist, do not create , -d, update the access and modification times , -m, change modification time ,-r use the access and modification times , -t, creates file using a specified time
* **Rm** : remove a file
* OPTIONS : -I prompt , -d empty , -r recursively , -ri files and sub directories recursively , -f forcefully , ./\ start with – hyphen.
* **Mkdir** create a directory
* OPTIONS : -m file mode , -p create parent if dsnt exst (/home/abc) ,
* **Rmdir**: remove a directory
* -p all component removal in pathname if empty , --ignore-fail-on-non-empty
* **Mvdir** : move a directory also rename directories in file system , for large files
* Mvdir file1 file2
* if file 2 exist it moves file 1 to file 2 , but if not it will rename file1 to file2
* **Cd** : change directory
* Cd .. Parent dir , cd – last working loc , ../../ 2 directory up , ~ move to home
* **Mv** : move or rename the files but it will delete the original file while moving.
* -- backup create backup of s and d , - f overwrite , -I prompt , -n never overwrite , -t provide destination for files
* **Cp** : copy and wont delete the file
* -b backup , -d preserve links , -l link files instead of copying , -R recursively , -s make symbolic links instead of copying , -I prompt before overwrite , -f remove existing destinations .
* **Man** : default user manual in linux
* **Pwd** : **pwd** stands for **P**rint **W**orking **D**irectory. It prints the path of the working directory, starting from the root.
* ***Options :pwd -L****: Prints the symbolic path.* , ***pwd -P****: Prints the actual path.*
* **wc** : find out number of newline count, word count, byte and characters count
* wc <option > file , -l no of lines , -w no of words , -c count of bytes , -m count of charcters , -L length of longest line
* **more** : view the text files in the command prompt, displaying one screen at a time in case the file is large
* ***more [-options] [-num] [+/pattern] [+linenum] [file\_name]***
* ***Options : -****d navigate , -f don’t wrap long lines , -p clear screen and display text , -s squeeze blank lines , -u omit underlines*
* ***Num :****number of lines that you want to display per screen.*
* ***Pattern : r****eplace pattern with any string that you want to find in the text file. +/anamika*
* ***linenum :*** *use the line number from where you want to start displaying the text content. +/30*
* **Less** : read contents of text file one page(one screen) per time.
* -f open non regular file , -p pattern , -s squeeze , -N numbers too .
* **pg :** displays a text file, pausing after each "page“
* **Pg <+number><-p string> <+line><+/pattern/> file**
* **+number : start , -number no of lines per page . -e show EOF , -p string instead of : string is displayed**
* **head :** by default, prints the first 10 lines of each FILE to standard output.
* Head <o> file : **-n lines , -c 20 bytes ,**
* **tail, :** prints the last 10 lines of the specified files.
* **Ex : cat state.txt |head –n 20| tail –n 20 > list.txt**
* **LAST :** show who has recently used the server and logged in and out date/time
* **Options : Last <user>/<option> -R hide hostnames , -F complete time , -w fqdn show , -x reboot last reboot , -t date**
* **diff :** display the differences in the files by comparing the files line by line.
* **Diff <o>a.txt b.txt , -c context with + - , -u no redundant info , -I case sensitive ,**
* **tac : reverse version of cat**
* **Tac /var/log/auth.log or tail /var/log/auth.log | tac**
* **news :** writes system news items to standard output , **/var/news** directory.
* **News , -a all news , -n report current news items without displaying contents , -s report news items without names or contents**
* **logname print the name of the current user , with root too – sudo logname - anamika**
* **Clear : clear screen and i**gnores any command-line parameters that may be present.
* **mesg :** Allow /disallow other users to send you messages. Also display current write status of terminal
* Mesg<y/n>

File System and Process Management

* File system : A file system is a logical collection of files on a partition or disk. A partition is a container for information and can span an entire hard drive if desired.
* **file hierarchy :**
* **/ :** root directory , contain only the directories needed at the top level of the file structure
* **/bin :** executable files , available to all users
* **/dev :** These are device drivers
* **/etc:** Supervisor directory commands, configuration files, disk configuration files, valid user lists, groups, ethernet, hosts, where to send critical messages
* **/lib :** Contains shared library files and sometimes other kernel-related files
* **/boot :** Contains files for booting the system
* **/home :**home directory for users and other accounts
* **/mnt:** mount other temporary file systems, such as **cdrom**
* **/proc: :** Contains all processes marked as a file by **process number**
* **/tmp :** temporary files used between system boots
* **/usr:** for miscellaneous purposes,Includes administrative commands, shared files, library files, and others
* **/var:** ontains variable-length files such as log and print files
* **/sbin:** Contains binary (executable) files, usually for system administration.
* **/kernel :** Contains kernel files
* **Links** : Softlink and Hard link
* connection between a file name and the actual data on the disk.
* two types : "hard" links, and "soft" or symbolic links.
* **Hard links** are low-level links which the system uses to create elements of the file system itself, such as files and directories.
* **Soft links** : symbolic link is a special file that points to another file or directory, which is called the **target**.
* Ln –s target linkname
* **Rsync** : Remote Sync) is a most commonly used command for copying and synchronizing files and directories remotely as well as locally
* rsync <o> src des
* -r copy data recursively , -a copy recursively also prevnt symbolic links , -z compress file data , -h human readable ,
* ‘–delete‘ option to delete files that are not there in source directory. , --max-size=‘200k’

File types in Linux

* File types : Regular , directory and special (block , character , pipe , symbolic link , socket)
* **Regular** : vi and touch : ls –l | grep ^-
* **Directory** : mkdir : ls –l | grep ^d
* **Block** : fdisk to create partition : **ls -l | grep ^b**
* **Character : provides a stream of i/o : ls -l | grep ^c**
* **Pipe : FIFO files : ls -l | grep ^p , use mkfifo to create**
* **Symbolic link : ln command ,** linked files to other files.: **ls -l | grep ^l**
* **Socket files :** pass information between applications for communication purpose
* **ls -l | grep ^s**
* Sort and Filter Command
* Filter Command : filters
* Grep : file pattern searcher
* Option : -i ignore case , -v invert the output , -n display the line number in the file , -w earches for the entire pattern that is in the string , -r search recursively:
* Iex: fconfig | grep –w “RUN”
* Awk : simple command line filtering tool.
* Ex: awk '/[0-9]/{print}' /etc/hosts = print all the lines having 0-9 in hosts
* Ex: awk '/^fe/{print}' /etc/hosts- starts with ff
* Ex: awk '/^ff/{print}' /etc/hosts -starts with ff
* Sed : view certain sections, replace words, and filter content from those files
* Ex :sed -n '5,10p' myfile.txt : return lines 5 through 10 from file
* Ex: sed -n -e '5,7p' -e '10,13p' myfile.txt : lines 5-7 and 10-13 from file
* -e print lines for a range , G blank lines

Controlling I/0 Commands and Combining Commands

* Redirection ,
* Write : cat anamika > abc.txt
* Append : echo line2 >> users
* pgm > file : Output of pgm is redirected to file
* pgm < file :Program pgm reads its input from file
* pgm >> file : Output of pgm is appended to file
* n > file : Output from stream with descriptor n redirected to file
* n >> file : Output from stream with descriptor n appended to file
* n >& m : Merges output from stream n with stream m
* n <& m : Merges input from stream n with stream m
* << tag : Standard input comes from here through next tag at the start of line
* Pipeline : pipes
* the output of each process directly as input to the next one like a pipeline. The symbol '|' denotes a pipe.
* cat filename | less : normally screen zoom , but with less scroll down possible

Searching Commands,

Sorting commands.

* **cut :** cutting out the sections from each line of files and writing the result to standard output.
* **Ex: cut –d “ ” file.txt - > -b -3 bytes , -c -5 column , -d “delimiter “**
* **sort :**  sort a file, arranging the records in a particular order.
* **Sort <o> file : -o output to a new file , -r reverse order , -n numerically , -k column number , -u remove duplicates , -M by month , -c check sorted or not .**

Editor in linux

* File editors : Text editors can be used for writing code, editing text files such as configuration files, creating user instruction files and many more.
* Ex: VI , NANO , gedit
* Vi editor:
* Start : vi anamika.yml
* I: insert mode (left), a: insert mode (right),
* :q to quit , : wq save and quit , ESC Terminate insert mode
* A : Write at the end of line (goes into insert mode) ,
* u :Undo last change , U:Undo all changes to the entire line
* O: Open a new line (goes into insert mode)
* Dd: Delete line , 3dd: Delete 3 lines, D: Delete contents of line after the cursor ,
* C: Delete contents of a line after the cursor and insert new text.
* Dw : Delete word , 4dw : Delete 4 words .
* Cw: Change word , x : Delete character at the cursor
* R : Replace character , R : Overwrite characters from cursor onward
* S: Substitute one character , S: Substitute entire line and begin to insert at the beginning of the line
* ~ Change case of individual character

Package and Service Management

* Package Utilities
* A software repository contains software packages.
* package” refers to a compressed file archive containing all of the files that come with a particular application.
* Debian and Ubuntu Package ManagementPermalink
* The Debian package management system, based on a tool called dpkg with the very popular apt system

**Advanced Packaging Tool (APT)**

* a command which uses the advanced packaging tool to interact with the operating system’s package system.
* apt-get install package-name(s) - Installs the package(s) specified, along with any dependencies.
* apt-get remove package-name(s) - Removes the package(s) specified, but does not remove dependencies.
* apt-get autoremove - Removes any orphaned dependencies, meaning those that remain installed but are no longer required.
* apt-get clean - Removes downloaded package files (.deb) for software that is already installed.
* apt-get purge package-name(s) - Combines the functions of remove and clean for a specific package, as well as configuration files.
* apt-get update - Reads the /etc/apt/sources.list file and updates the system’s database of packages available for installation. Run this after changing sources.list.
* apt-get upgrade - Upgrades all packages if there are updates available. Run this after running apt-get update
* dpkg -i package-file-name.deb - Installs a .deb file.
* dpkg --list search-pattern - Lists packages currently installed on the system.
* dpkg --configure package-name(s) - Runs a configuration interface to set up a package.
* dpkg-reconfigure package-name(s) - Runs a configuration interface on an already installed package.

Installation utilities.: RPM and YUM

* YUM (Yellowdog Updater Modified) is an open source command-line as well as graphical based package management tool for RPM (RedHat Package Manager) based Linux systems. It allows users and system administrator to easily install, update, remove or search software packages on a systems.
* Yum –y install firefox : install a package
* Yum remove firefox : remove a package completely with their all dependencies
* Yum update mysql :update it to the latest stable version.
* Yum list openssh “: search for the specific package with name
* Yum search vsftpd : search all the available packages to match the name of the package
* Yum info firefox : information of a package
* RPM : package management utility for Red Hat based systems like (RHEL, CentOS and Fedora). The tool allows system administrators and users to install, update, uninstall, query, verify and manage system software packages in Unix/Linux operating systems. The RPM formerly known as .rpm file, that includes compiled software programs and libraries needed by the packages.
* Install : It is used to install any RPM package.
* Remove : It is used to erase, remove or un-install any RPM package.
* Upgrade : It is used to update the existing RPM package.
* Verify : It is used to verify an RPM packages.
* Query : It is used query any RPM package.
* Rpm –ivh piggin-2.7.9-5.el6.2.i686.rpm
* I install , -v nicer display , -h print hash marks , -q query a package , -p list capabilities provided , -R list dependent capabilities .
* **Creation of Local Repo**

Yum install createrepo : install createrepo

Mkdir /repo1 : create dir

Wget <rpmfile.rpm> : put rpm files into dir

Createrepo / repo1 “ create metadata for repo1

/etc/yum.repos.d/custom.repo :create yum configuration file

[customrepo]

Name =custom repo

Baserurl = file:///repo1/

Enabled=1

Gpgcheck=0

------ vi /etc/yum.repos.d/example.repo

Managing Services :System services overview

* Services are programs or processes that run on your server at all times, usually from the time a server boots up.
* init system is the first process that starts outside the kernel, and is generally the backend service that controls when and how services are started.
* systemd, services are included in the more general description of 'units'. Units can also be used to perform one off tasks on boot, and a bunch of other things.
* Systemctl : Service units end with the  .service file extension and serve a similar purpose as init scripts. To view, start, stop, restart, enable, or disable system services, use the systemctl command
* Systemctl is the main utility to control daemons/ services in system , while service command is traditional utility in sysvinit .
* Systemctl is ,ore powerful version of service

|  |  |  |
| --- | --- | --- |
| service | systemctl | Description |
| service name start | systemctl start name.service | Starts a service. |
| service name stop | systemctl stop name.service | Stops a service. |
| service name restart | systemctl restart name.service | Restarts a service. |
| service namecondrestart | systemctl try-restart name.service | Restarts a service only if it is running. |
| service name reload | systemctl reload name.service | Reloads configuration. |
| service name status | systemctl status name.service systemctl is-active name.serviceUser and group management  configure , id, groupadd, groupdel, groupmod,  **Useradd** : create new accounts in Linux , **useradd <options>Anamika**  **Usermod**  modify the existing accounts in linux , **usermod <options> anamika**  **Userdel** :  delete local account in linux **, userdel <options> anamika**  **Passwd** : assign password to local accounts or users. **, passwd Anamika**  **Chage** : view & modify users password expiry information  change password at first login by using command ‘**chage -d 0 <username>‘.**  **chage -l <username>** command to view the user’s password expiry info. | Checks if a service is running. |
| service --status-all | systemctl list-units --type service --all | Displays the status of all services. |

|  |  |  |
| --- | --- | --- |
| chkconfig | systemctl | Description |
| chkconfig name on | systemctl enable name.service | Enables a service. |
| chkconfig name off | systemctl disable name.service | Disables a service. |
| chkconfig --list name | systemctl status name.service systemctl is-enabled name.service | Checks if a service is enabled. |
| chkconfig --list | systemctl list-unit-files --type service | Lists all services and checks if they are enabled. |
| chkconfig --list | systemctl list-dependencies --after | Lists services that are ordered to start before the specified unit. |
| chkconfig --list | systemctl list-dependencies --before | Lists services that are ordered to start after the specified unit. |

Network Administration

* If config : display all the active interfaces details. The ifconfig command also used to check the assigned IP address of an server.
* Eth0 : specific network intf , enable : up , disable down , assign ip : eth0 <ip> , assign netmask : eth0 <ip>
* **Ping** :  testing if a host is reachable on an Internet Protocol (IP) network.
* Ping <ip>
* Options : -I set interval ,
* **Ping6** : **To ping the node (network computer) which has an IPv6 address i.e. to ping or check the connectivity under IPv6 network,**
* Ping6 <hostname>
* **Dig** : Domain Information Groper) for querying Domain Name System (DNS) name servers. It is useful for verifying and troubleshooting DNS problems and also to perform DNS lookups and displays the answers that are returned from the name server that were queried
* Dig yahoo.com,
* +short : cut o/p ,
* **Nslookup** : testing and troubleshooting DNS servers
* Nslookup yahoo.com
* -query=ns , MX ,any , -type=soa , -debug
* Lsof:  long listing of open files
* -u user specific , -I specific port , 4 ipv4 , 6 ipv6 , -p specific pid
* route:  show / manipulate the IP routing table
* host : DNS lookup utility host <ip>
* Loopback : The **loopback** IP address is the address used to access itself. The IPv4 designated 127.0.0.1 as the **loopback** address with the 255.0.0.0 subnet mask . A **loopback**interface is also known as a virtual IP,
* Traceroute :  map the journey that a packet of information undertakes from its source to its destination.
* Traceroute<hostaddress >
* kill terminate processes manually. , kill pid
* scp : copy file(s) between servers in secure way.
* scp source\_file userame@dest\_host: destfolder
* -P port bind , -r copy recursively ,
* Netstat :  monitoring network connections both incoming and outgoing as well as viewing routing tables, interface statistics etc.
* -a for tcp and udp , -at tcp , -au udp , -l all , -lt all active tcp , -lu all active udp
* Iptables :  Iptables is a Linux command line firewall that allows system administrators to manage incoming and outgoing traffic via a set of configurable table rules.
* Filter : default table, which contains the built in chains for:INPUT  – packages destined for local sockets
* FORWARD – packets routed through the system
* OUTPUT – packets generated locally
* NAT : a table that is consulted when a packet tries to create a new connection. It has the following built-in:PREROUTING – used for altering a packet as soon as it’s received
* OUTPUT – used for altering locally generated packets
* POSTROUTING – used for altering packets as they are about to go out
* MANGLE : used for packet altering.

Process and disk management

* nohup,df,du,bdf,mount/unmount, cksum, history, file, shutdown, poweroff, reboot, mdfilesum , fstab
* Job control commands enable you to place jobs in the foreground or background, and to start or stop jobs. The table describes the job control commands.
* Jobs
* : list all jobs
* BG%N : PLACE THE JOB IN BAGROUNG , N IS JOB ID
* fg %N : PLACE THE JOB IN goreground, N IS JOB ID
* Nohup : run any program after log out or exit from Linux operating system then you have to use nohup command.
* nohup bash sleep1.sh
* Du: du command (short for disk usage) is useful command which is used to find disk usage for files & directories.
* **du /home**
* **-h : human readable , -s size ,**
* Df : **play information of device name, total blocks, total disk space, used disk space, available disk space and mount points on a file system.**
* Bdf : identify the list of filesystems available in the server, where it is mounted and corresponding disk space details in kbytes.
* Crontabs Cron is a daemon to run schedule tasks. Cron wakes up every minute and checks schedule tasks in crontable. Crontab (CRON TABle) is a table where we can schedule such kind of repeated tasks.
* Location L /etc/cron.d , /etc/cron.daily , /etc/cron.hourly , /etc/cron.monthly , /etc/cron.weekly
* Option : -l list or manage tak , -e edit crontab entry , -u userspecific , -r remove scheduled jobs , -I prompt before deleting
* Character : - define range , / 10 min , , separate items .
* What is SELinux and how to configure it 
* set of security policies/modules which are going to apply on the machine to improve the overall security of the machine.
* Sestatus to check status
* Disable : edit /etc/selinux/config …. Disbaled/enforcing

Monitoring commands

* ,Tcpdump, Java heap dump and thread dump
* Top :  processor activity of your Linux box and also displays tasks managed by kernel in real-time. It’ll show processor and memory are being used and other information like running processes. ( tasks, memory, cpu)
* -O sorting , -u user specific , z highlight
* Vmstat : sysstat package install, (vmstat, sarand iostat) system monitoring tools.
* Summary information of Memory, Processes, Paging etc.
* 2 6 after 2 sec and stop after 6 intervals , -t timestamps , -s memory stats , -d disk stats
* Iostat : Central Processing Unit (CPU) statistics and input/output statistics for devices and partitions
* generates CPU, I/O statistics.
* Sar : monitor performance of various Linux subsystems (CPU, Memory, I/O..) in real time.
* collects and displays ALL system activities statistics.
* Mpstat :collects and displays information about CPU utilization and performance statistics. displays CPU statistics.
* Communication in Linux
* SSH : securely log onto remote systems.
* ssh-keygen : gen ssh key
* Ssh copy-id : copy key on hosts
* ssh remote\_host
* ssh remote\_username@remote\_host
* Start “: service ssh start , systemctl start ssh
* SFTP , **sftp** performs all operations over an encrypted **ssh** session. It uses many of the features of **ssh**, such as public key authentication and data compression
* TELNET : The **telnet** command is used for interactive communication with another host using the TELNET protocol.
* **telnet** [-**8EFKLacdfrx**] [-**X***authtype*] [-**b***hostalias*] [-**e***escapechar*] [-**k***realm*] [-**l***user*] [-**n***tracefile*] [ *host* [*port*] ]