vagrant ssh-config > .vagrant\_ssh.conf

ssh -F .vagrant\_ssh.conf default # where default is the name of your box,

lsb\_release -a

root@ubuntu-xenial:~# vi /etc/apt/sources.list

whenever you make some configuration change in ubuntu we just restart the service and not the server

in ubunut the service is automatically enabled in ubuntu once you start it

gzip and gunzip

**surendra@linuxnix:~/test/test$ ls**

**2  3  4   dump.doc  file1  test.sh**

**surendra@linuxnix:~/test/test$ gzip dump.doc file1 test.sh**

**surendra@linuxnix:~/test/test$ ls**

**2  3  4  dump.doc.gz  file1.gz  test.sh.gz**

**Example5:** To know compression ratio use -l option

[root@node01 testing]# gzip -l \*.gz

compressed uncompressed ratio uncompressed\_name

101798 104857600 99.9% file10

101797 104857600 99.9% file1

101797 104857600 99.9% file2

101797 104857600 99.9% file3

101797 104857600 99.9% file4

101797 104857600 99.9% file5

101797 104857600 99.9% file6

101797 104857600 99.9% file7

101797 104857600 99.9% file8

101797 104857600 99.9% file9

203 10240 98.2% make.tar

1018372 1048586240 99.9% trunc.tar

2036546 2097172480 99.9% (totals)

you can do

mkdir /home/user1

usermod -s /bin/bash -d /home/user1 user1

root@ubuntu-xenial:~# cp /etc/skel/.bash\_logout /home/user1/

root@ubuntu-xenial:~# cp /etc/skel/.bashrc /home/user1/

root@ubuntu-xenial:~# cp /etc/skel/.profile /home/user1/

then copy the below files to

root@vagrant-ubuntu-trusty-64:~# declare -f | grep '^[a-z\_]'

sed 'p' /etc/passwd will print twice

root@vagrant-ubuntu-trusty-64:~# sed -n 'p' /etc/passwd

Printing lines 1-5

root@vagrant-ubuntu-trusty-64:~# sed -n '1,5 p' /etc/passwd

root@vagrant-ubuntu-trusty-64:~# ifconfig eth0|awk -F":" '/HWaddr/{ print $2 $3 $4 $5 %6 $7 }'|cut -d" " -f 4

root@vagrant-ubuntu-trusty-64:~# awk ' {print } ' /etc/ntp.conf

root@vagrant-ubuntu-trusty-64:~# awk 'BEGIN { print "vsftpd.conf" } { print } END { print NR } ' /etc/vsftpd.conf

root@vagrant-ubuntu-trusty-64:~# awk 'BEGIN { print "vsftpd.conf" } { print NR, $0 } END { print NR } ' /etc/vsftpd.conf

in the above $0 prints the whole line and NR will print the line number

root@server:~# ifconfig eth0 | awk -F":" '/HWaddr/{print $3 $4 $5 $6 $7}'

root@server:~# ifconfig eth0 | awk -F":" '/HWaddr/{print toupper($3 $4 $5 $6 $7)}'

root@server:~# ifconfig eth0 | grep -i Rx

root@server:~# grep -c sams /etc/passwd

after this running the below will means the same as running cat /etc/passwd

root@server:~# cat !$

#!/bin/bash

IFS=";"

while read student class marks

do

echo -e "\e[1;33m$student \

=========================\e[0m\n\

Class : \t $class \n\

Marks : \t $marks \n"

done < $1

-------------------------------------------

root@server:~# ./students.sh students | grep -A2 -i ankit

'^' start of string

'$' end of string

'^root' string starts with root

'4$' string ends with 4

root@server:~# grep -c ^server /etc/ntp.conf.bak

root@server:~# cat -vet !$

Ranges

any letter

'[A-Za-z]'

any digit

'[0-9]'

'[a-z\_]'

any lowercase letter or \_

'[349]'

matches 3, 4 or 9

root@server:~# declare -f | grep '^[a-z\_]'

The below will only search for the files either ending with 'rotate 4' or 'rotate 6'

root@server:~# grep 'rotate [46]$' /etc/logrotate.d/\*

writing ^ inside the bracket will negate that character in the output like below:-

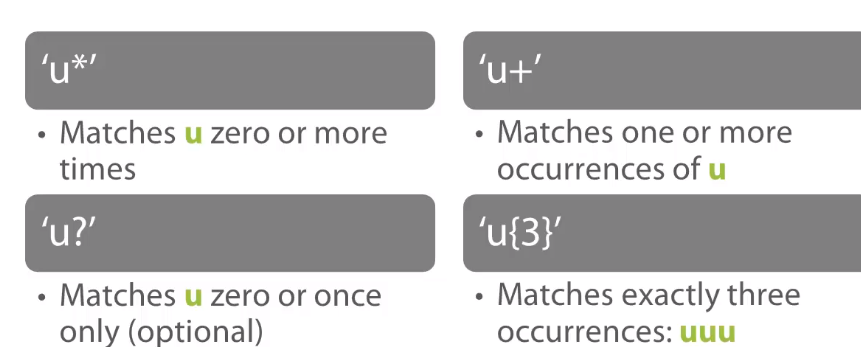
root@server:~# grep 'rotate [^4]$' /etc/logrotate.d/\*

root@server:~# grep '^[Ss]erver' server

Search and replace in a file

root@server:~# sed -i 's/shyam/ankit/g' testing

[root@localhost ~]# sed -i.bak '/^#/d;/^$/d' /etc/ntp.conf



[root@localhost ~]# grep -v '^\s\*#' test

[root@localhost ~]# cat test

#end

i am no good

#this is good

what are you doing

i am doing good.

#this is it.

#this is my goal

Vande mataram

jain shri ram

[root@localhost ~]# grep -vE '\b[0-9]{3}-[0-9]{2}-[0-9]{4}\b' employees

[root@localhost ~]# cat employees

Jones,Bob,232-78-3456

Jackeson,Jane,,

Federer,Jack,xxx-xx-xxxx

Maw,Michael,1879-0

Alexander,Sally,345-89-8095

Beder,Ioana,567-34-9802

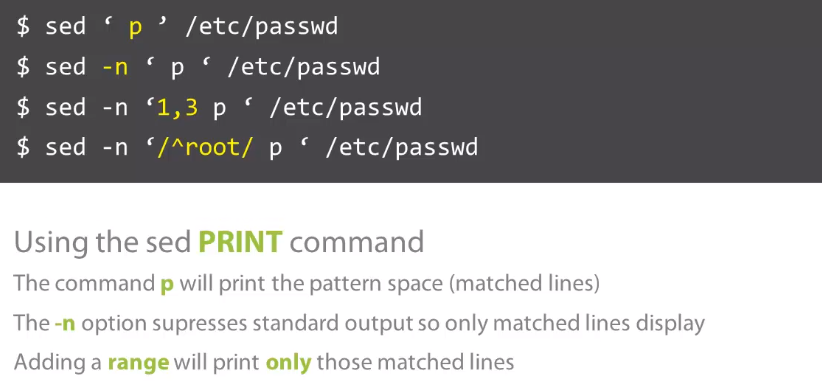
Staines,Brad,,

\s - whitespace

\b - word boundary

'\ssystem' - matches "file system"

'\bsystem' - matches "file system"



[root@localhost ~]# sed -n '/^root/ p ' /etc/passwd

.bash\_profile - is designed to execute whe you login, it configures your shell before you get the initial command prompt

when you are logged in already then you run .bashrc

# find /home -iname tecmint.txt

./tecmint.txt

./Tecmint.txt

4. Find Directories Using Name

Find all directories whose name is Tecmint in / directory.

# find / -type d -name Tecmint

/Tecmint

5. Find PHP Files Using Name

Find all php files whose name is tecmint.php in a current working directory.

# find . -type f -name tecmint.php

./tecmint.php

6. Find all PHP Files in Directory

Find all php files in a directory.

# find . -type f -name "\*.php"

5. Find Files with 777 Permissions and Chmod to 644

Find all 777 permission files and use chmod command to set permissions to 644.

# find / -type f -perm 0777 -print -exec chmod 644 {} \;

16. Find Directories with 777 Permissions and Chmod to 755

Find all 777 permission directories and use chmod command to set permissions to 755.

# find / -type d -perm 777 -print -exec chmod 755 {} \;

17. Find and remove single File

To find a single file called tecmint.txt and remove it.

# find . -type f -name "tecmint.txt" -exec rm -f {} \;

18. Find and remove Multiple File

To find and remove multiple files such as .mp3 or .txt, then use.

# find . -type f -name "\*.txt" -exec rm -f {} \;

OR

# find . -type f -name "\*.mp3" -exec rm -f {} \;

To find all hidden files, use below command.

# find /tmp -type f -name ".\*"

# find /home -group developer

# find /home -user tecmint

# find /home -user tecmint -iname "\*.txt"

[root@tecmint /]# whereis date

date: /bin/date /usr/share/man/man1/date.1.gz

Move Current Working Directory

Move from current working directory to any level up by just providing the numerical value at the end of script while executing.

#! /bin/bash

LEVEL=$1

for ((i = 1; i <= LEVEL; i++))

do

CDIR=../$CDIR

done

cd $CDIR

echo "You are in: "$PWD

exec /bin/bash

[root@tecmint /]# chmod 755 up

[root@tecmint /]# ./up.sh 2

You are in: /

[root@tecmint /]# ./up.sh 4

You are in: /

[root@tecmint /]#

### Create a Random File or Folder

Create a random file (folder) with no chance of duplication.

#! /bin/bash

echo "Hello $USER";

echo "$(uptime)" >> "$(date)".txt

echo "Your File is being saved to $(pwd)"

This is a Simple script but it’s working is not that much simple.

1. ‘**echo**‘ : Prints everything written within the quotes.
2. ‘**$**‘ : Is a shell variable.
3. ‘**>>**‘ : The output is redirected to the output of **date** command followed by **txt** extension.

[root@localhost ~]# ping -c 5 [www.google.com](http://www.google.com)

&& Operator

For example, I want to visit website **tecmint.com** using [links command](https://www.tecmint.com/command-line-web-browsers/), in terminal but before that I need to check if the host is **live** or **not**.

root@localhost:/home/tecmint# ping -c3 www.tecmint.com && links www.tecmint.com

The **OR Operator** (**||**) is much like an ‘**else**‘ statement in programming. The above operator allow you to execute second command only if the execution of first command fails, i.e., the exit status of first command is ‘**1**‘.

tecmint@localhost:~$ apt-get update || links tecmint.com

Now delete all the files except ‘html‘ file all at once, in a smart way.

tecmint@localhost:~/tecmint$ rm -r !(\*.html)

**$ id tecmint**

**$ groups tecmint**

### 3. finger Command

**$ finger tecmint**

Login: tecmint Name: TecMint

Directory: /home/tecmint Shell: /bin/bash

On since Fri Sep 22 10:39 (IST) on tty8 from :0

2 hours 1 minute idle

No mail.

No Plan.

### 4. getent Command

**getent** is a command line utility for fetching entries from **Name Service Switch** (**NSS**) libraries from a specific system database.

To get a user’s account details, use the **passwd** database and the **username** as follows.

**$ getent passwd tecmint**

tecmint:x:1000:1000:TecMint,,,:/home/tecmint:/bin/bash

**$ grep -i tecmint /etc/passwd**

### 6. lslogins Command

**lslogins command** shows information about known users in the system, the **-u** flag only displays user accounts.

**$ lslogins -u**

UID USER PROC PWD-LOCK PWD-DENY LAST-LOGIN GECOS

0 root 144 root

1000 tecmint 70 10:39:07 TecMint,,,

1001 aaronkilik 0

1002 john 0 John Doo

### 7. users Command

**users command** shows the usernames of all users currently logged on the system like so.

**$ users**

tecmint

aaron

### 8. who Command

**who command** is used to display users who are logged on the system, including the terminals they are connecting from.

**$ who -u**

tecmint tty8 2017-09-22 10:39 02:09 2067 (:0)

### 9. w Command

**w command** shows all users who are logged on the system and what they are doing.

**$ w**

12:46:54 up 2:10, 1 user, load average: 0.34, 0.44, 0.57

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT

tecmint tty8 :0 10:39 2:10m 4:43 0.46s cinnamon-sessio

### 10. last or lastb commands

**last/lastb commands** displays a list of last logged in users on the system.

**$ last**

OR

**$ last -a** #show hostname on the last column

### 11. lastlog Command

**lastlog command** is used to find the details of a recent login of all users or of a given user as follows.

**$ lastlog**

OR

**$ lastlog -u tecmint** #show lastlog records for specific user tecmint

**Check Disk Space:**

**MAX=95**

**EMAIL=server@127.0.0.1**

**PART=sda1**

**USE=`df -h |grep $PART | awk '{ print $5 }' | cut -d'%' -f1`**

**if [ $USE -gt $MAX ]; then**

**echo "Percent used: $USE" | mail -s "Running out of disk space" $EMAIL**

**fi**

**~**

**~**

**Colours**

#!/bin/bash

clear

echo -e "\033[1m Hello World"

# bold effect

echo -e "\033[5m Blink"

# blink effect

echo -e "\033[0m Hello World"

# back to noraml

echo -e "\033[31m Hello World"

# Red color

echo -e "\033[32m Hello World"

# Green color

echo -e "\033[33m Hello World"

# See remaing on screen

echo -e "\033[34m Hello World"

echo -e "\033[35m Hello World"

echo -e "\033[36m Hello World"

echo -e -n "\033[0m"

# back to noraml

echo -e "\033[41m Hello World"

echo -e "\033[42m Hello World"

echo -e "\033[43m Hello World"

echo -e "\033[44m Hello World"

echo -e "\033[45m Hello World"

echo -e "\033[46m Hello World"

echo -e "\033[0m Hello World"

**Encrypting a File:-**

#!/bin/bash

echo "Welcome, I am ready to encrypt a file/folder for you"

echo "currently I have a limitation, Place me to teh same folder, where a file to be encrypted is present"

echo "Enter the Exact File Name with extension"

read file;

gpg -c $file

echo "I have encrypted the file sucessfully..."

echo "Now I will be removing the original file"

rm -rf $file

~

~

~

Server Health:-

#!/bin/bash

date;

echo "uptime:"

uptime

echo "Currently connected:"

w

echo "--------------------"

echo "Last logins:"

last -a |head -3

echo "--------------------"

echo "Disk and memory usage:"

df -h | xargs | awk '{print "Free/total disk: " $11 " / " $9}'

free -m | xargs | awk '{print "Free/total memory: " $17 " / " $8 " MB"}'

echo "--------------------"

start\_log=`head -1 /var/log/messages |cut -c 1-12`

oom=`grep -ci kill /var/log/messages`

echo -n "OOM errors since $start\_log :" $oom

echo ""

echo "--------------------"

echo "Utilization and most expensive processes:"

top -b |head -3

echo

top -b |head -10 |tail -4

echo "--------------------"

echo "Open TCP ports:"

nmap -p- -T4 127.0.0.1

echo "--------------------"

echo "Current connections:"

ss -s

echo "--------------------"

echo "processes:"

ps auxf --width=200

echo "--------------------"

echo "vmstat:"

vmstat 1 5

~

~

~

Monitoring a Server:

[root@localhost ~]# cat tecmint\_monitor.sh

####################################################################################################

# Tecmint\_monitor.sh #

# Written for Tecmint.com for the post www.tecmint.com/linux-server-health-monitoring-script/ #

# If any bug, report us in the link below #

# Free to use/edit/distribute the code below by #

# giving proper credit to Tecmint.com and Author #

# #

####################################################################################################

#! /bin/bash

# unset any variable which system may be using

# clear the screen

clear

unset tecreset os architecture kernelrelease internalip externalip nameserver loadaverage

while getopts iv name

do

case $name in

i)iopt=1;;

v)vopt=1;;

\*)echo "Invalid arg";;

esac

done

if [[ ! -z $iopt ]]

then

{

wd=$(pwd)

basename "$(test -L "$0" && readlink "$0" || echo "$0")" > /tmp/scriptname

scriptname=$(echo -e -n $wd/ && cat /tmp/scriptname)

su -c "cp $scriptname /usr/bin/monitor" root && echo "Congratulations! Script Installed, now run monitor Command" || echo "Installation failed"

}

fi

if [[ ! -z $vopt ]]

then

{

echo -e "tecmint\_monitor version 0.1\nDesigned by Tecmint.com\nReleased Under Apache 2.0 License"

}

fi

if [[ $# -eq 0 ]]

then

{

# Define Variable tecreset

tecreset=$(tput sgr0)

# Check if connected to Internet or not

ping -c 1 google.com &> /dev/null && echo -e '\E[32m'"Internet: $tecreset Connected" || echo -e '\E[32m'"Internet: $tecreset Disconnected"

# Check OS Type

os=$(uname -o)

echo -e '\E[32m'"Operating System Type :" $tecreset $os

# Check OS Release Version and Name

cat /etc/os-release | grep 'NAME\|VERSION' | grep -v 'VERSION\_ID' | grep -v 'PRETTY\_NAME' > /tmp/osrelease

echo -n -e '\E[32m'"OS Name :" $tecreset && cat /tmp/osrelease | grep -v "VERSION" | cut -f2 -d\"

echo -n -e '\E[32m'"OS Version :" $tecreset && cat /tmp/osrelease | grep -v "NAME" | cut -f2 -d\"

# Check Architecture

architecture=$(uname -m)

echo -e '\E[32m'"Architecture :" $tecreset $architecture

# Check Kernel Release

kernelrelease=$(uname -r)

echo -e '\E[32m'"Kernel Release :" $tecreset $kernelrelease

# Check hostname

echo -e '\E[32m'"Hostname :" $tecreset $HOSTNAME

# Check Internal IP

internalip=$(hostname -I)

echo -e '\E[32m'"Internal IP :" $tecreset $internalip

# Check External IP

externalip=$(curl -s ipecho.net/plain;echo)

echo -e '\E[32m'"External IP : $tecreset "$externalip

# Check DNS

nameservers=$(cat /etc/resolv.conf | sed '1 d' | awk '{print $2}')

echo -e '\E[32m'"Name Servers :" $tecreset $nameservers

# Check Logged In Users

who>/tmp/who

echo -e '\E[32m'"Logged In users :" $tecreset && cat /tmp/who

# Check RAM and SWAP Usages

free -h | grep -v + > /tmp/ramcache

echo -e '\E[32m'"Ram Usages :" $tecreset

cat /tmp/ramcache | grep -v "Swap"

echo -e '\E[32m'"Swap Usages :" $tecreset

cat /tmp/ramcache | grep -v "Mem"

# Check Disk Usages

df -h| grep 'Filesystem\|/dev/sda\*' > /tmp/diskusage

echo -e '\E[32m'"Disk Usages :" $tecreset

cat /tmp/diskusage

# Check Load Average

loadaverage=$(top -n 1 -b | grep "load average:" | awk '{print $10 $11 $12}')

echo -e '\E[32m'"Load Average :" $tecreset $loadaverage

# Check System Uptime

tecuptime=$(uptime | awk '{print $3,$4}' | cut -f1 -d,)

echo -e '\E[32m'"System Uptime Days/(HH:MM) :" $tecreset $tecuptime

# Unset Variables

unset tecreset os architecture kernelrelease internalip externalip nameserver loadaverage

# Remove Temporary Files

rm /tmp/osrelease /tmp/who /tmp/ramcache /tmp/diskusage

}

fi

shift $(($OPTIND -1))