

WORD DOCUMENT

<Process Title>

<Apex/Group/Operational> Layer

Document Version / Details: Ver. 2.0/ 14-Nov-2022

History:

Bash stores your commands automatically in `~/.bash_history`. If you want to have a look at the history, either print the output of this file using one of

```
cat ~/.bash_history
less ~/.bash_history
```

we can use bash's builtin command:

```
history
```

options:

You can truncate the output to some lines (where 5 is the number of lines):

```
history 5
```

If do you want to view only commands containing a string (i.e. `mv`), you can do this:

```
history | grep mv
```

You can recall a command by typing `!` followed by the entry number.

Let's say that I have a history like this:

```
1 ls -la
2 mkdir foo
3 mv bar.txt foo
```

- To run `mkdir foo`, you can type `!2`.
- To run the last command, you can use `!-1` or `!!`
- To run the penultimate, you can use `!-2`

output:

```
1 passwd labuser
2 systemctl stop iptable
3 systemctl stop firewalld
4 ipconfig.io
5 apt-get install net-tools
6 df -h
7 passwd labuser
8 sudo visudo
9 passwd ubuntu
10 passwd labuser etc.,
```

ifconfig - configure a network interface

`ifconfig` command is used to configure the kernel-resident network interfaces. It is used at boot time to set up interfaces as necessary. After that, it is usually needed to

query and manage network parameters. It is also required during debugging or when system tuning is needed.

If no arguments are given, `ifconfig` command displays the status of the currently active interfaces. If a single interface argument is given, it displays the status of the given interface only; in case of `-a` option, it displays the status of all interfaces, even those that are down. Otherwise, it configures an interface.

Output:

```
ens5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
    inet 172.31.20.131 netmask 255.255.240.0 broadcast 172.31.31.255
    inet6 fe80::49e:c9ff:fe07:8cfa prefixlen 64 scopeid 0x20<link>
    ether 06:9e:c9:07:8c:fa txqueuelen 1000 (Ethernet)
    RX packets 16891 bytes 22442909 (22.4 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4357 bytes 1352935 (1.3 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 365 bytes 28632 (28.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 365 bytes 28632 (28.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Options:

- a display all interfaces which are currently available, even if down
- s display a short list (like `netstat -i`)
- v be more verbose for some error conditions

"Java - The Technology for everything'!!?"

"Java - The Technology for everything'!!?"

Write a script that will check if a given argument is the string "Java"

If it's the string "Java" print "Java - The Technology for everything'!!?"

If it's not the string "Java" print the actual argument & also print "Wondering how is it relate to Java!!"

Write a script to accept 2 numbers as arguments viz., no1 and no2.

If no1 is not equal to no2

Display the following output.

1. The sum of the two numbers
2. The difference of the two numbers
3. The product of two numbers
4. The factor of two numbers
5. The percentage of no1 to no2
6. The reminder of the 2 numbers

Run the `top` command and observe for utilization parameters such as CPU, Memory. Based on your understanding, write a script to capture them and display along with date. Sample output is as below

Current Timestamp: 10-10-2023 14:42

Total Memory: 4MB

Memory Consumed: 3 MB

Memory Free: 1 MB

CPU%: 3.7

Total # of Tasks running: 303

1. Reliability, availability, scalability
2. Git branching strategies available
3. MTBF, MTTR, Error budget, toiling in SRE(toil care off)
4. SLA, SLI, SLO

Sre is metric driven

SRE key terminologies

Toil – repetitive works

Anscripts <--Jenkins

Devops – abstract while SRE is concrete

**Let's get to the
future, faster.
Together.**

