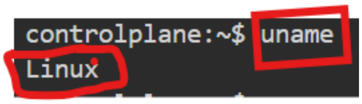
Linux Commands

System Commands:

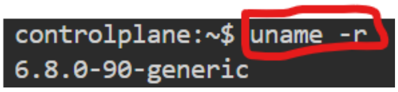
System commands are Linux commands used to check, monitor, and manage the system’s hardware, memory, CPU, disk, processes, and overall performance.

1.uname: used to get type of OS



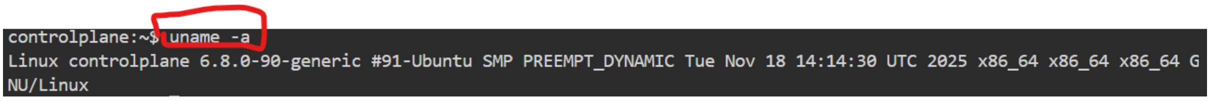
```
controlplane:~$ uname
Linux
```

2.uname -r : used to get kernel version of our OS



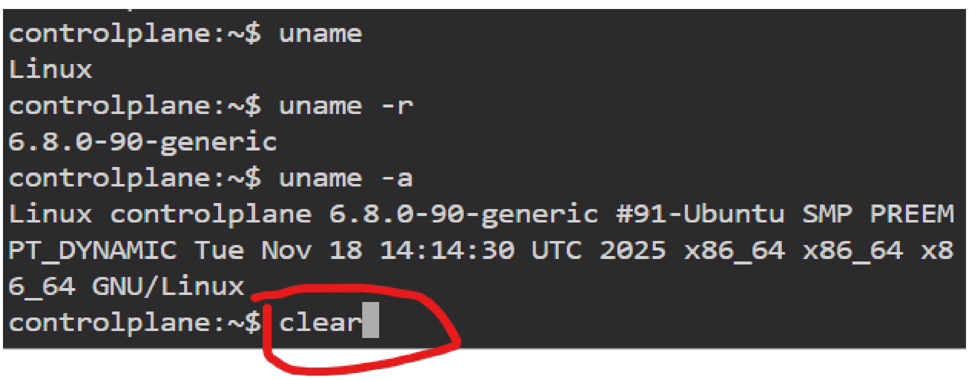
```
controlplane:~$ uname -r
6.8.0-90-generic
```

3.uname -a: used to get full info about OS

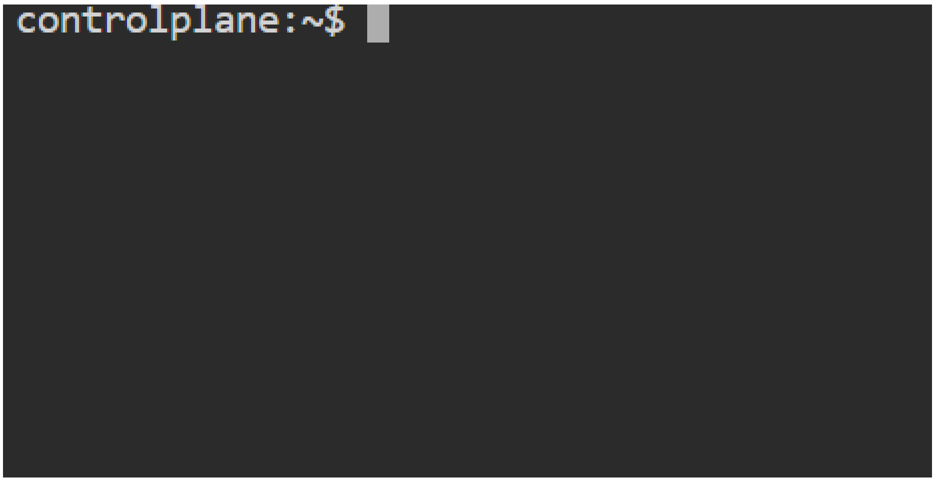


```
controlplane:~$ uname -a
Linux controlplane 6.8.0-90-generic #91-Ubuntu SMP PREEMPT_DYNAMIC Tue Nov 18 14:14:30 UTC 2025 x86_64 x86_64 x86_64 G
NU/Linux
```

4. clear: this command is used to clear the clear

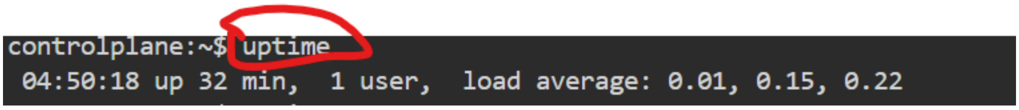


```
controlplane:~$ uname
Linux
controlplane:~$ uname -r
6.8.0-90-generic
controlplane:~$ uname -a
Linux controlplane 6.8.0-90-generic #91-Ubuntu SMP PREEM
PT_DYNAMIC Tue Nov 18 14:14:30 UTC 2025 x86_64 x86_64 x8
6_64 GNU/Linux
controlplane:~$ clear
```



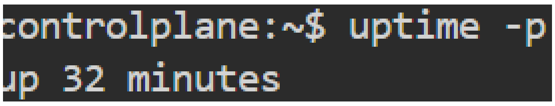
```
controlplane:~$
```

5. **uptime** : used to get since how long our system is in running state



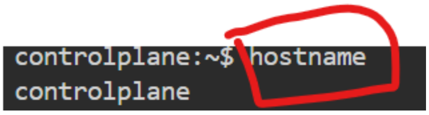
```
controlplane:~$ uptime
 04:50:18 up 32 min,  1 user,  load average: 0.01, 0.15, 0.22
```

6.uptime -p : this will give only time



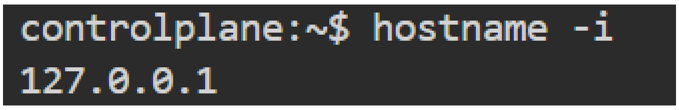
```
controlplane:~$ uptime -p
up 32 minutes
```

7.hostname: used to get private dns name of our system



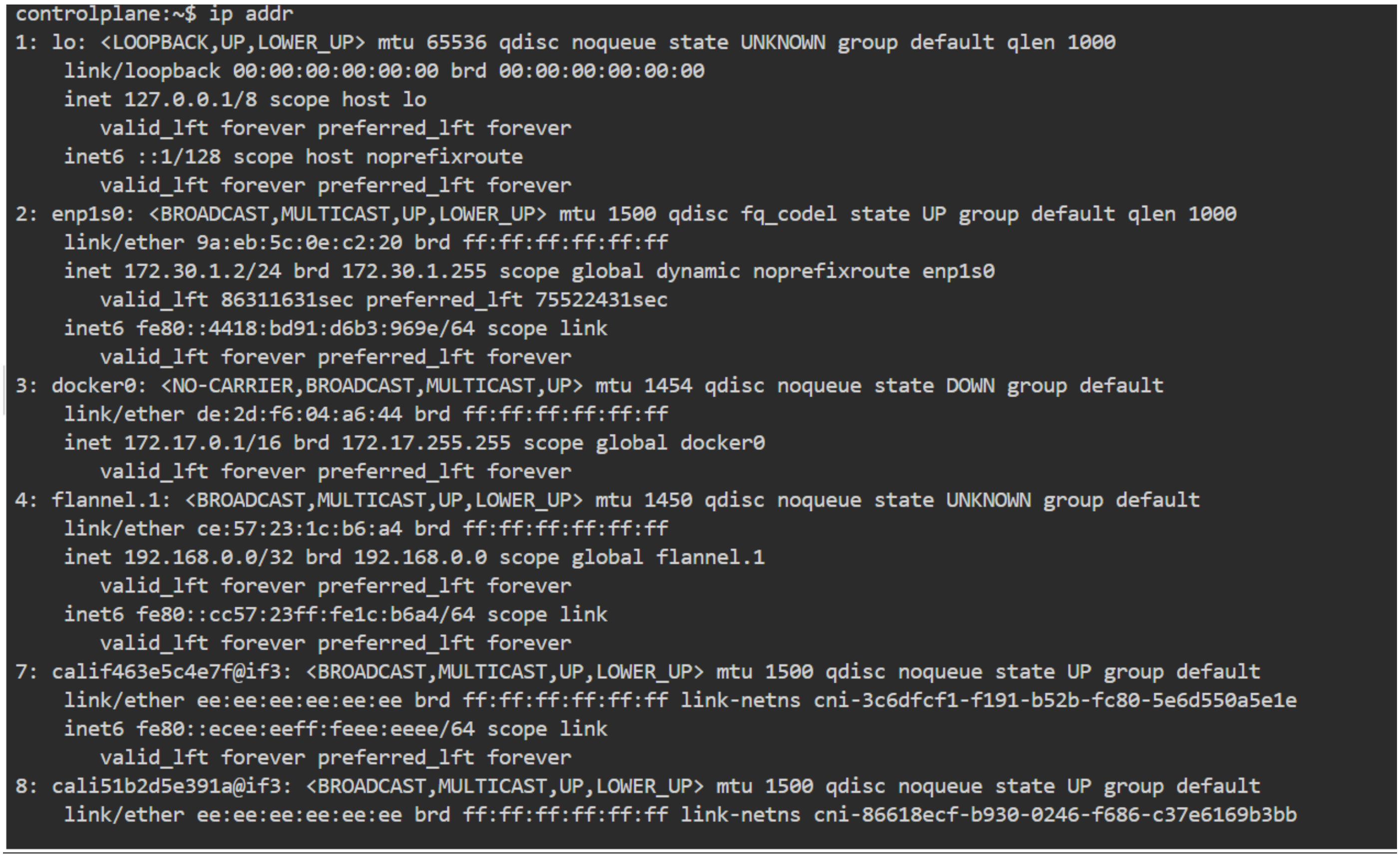
```
controlplane:~$ hostname
controlplane
```

8.hostname -i : used to get private ip of our system



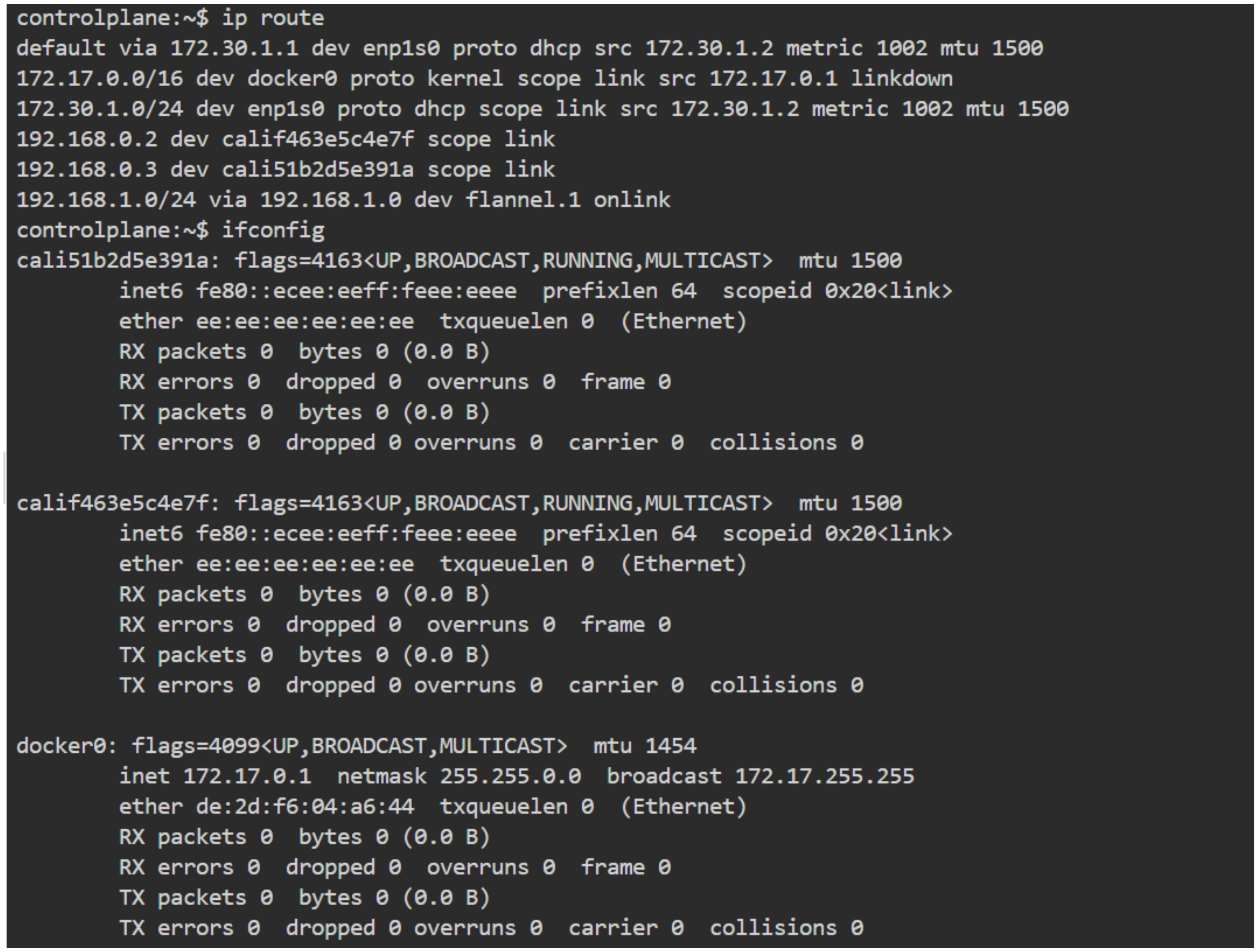
```
controlplane:~$ hostname -i
127.0.0.1
```

9.ip addr : used to get private IP



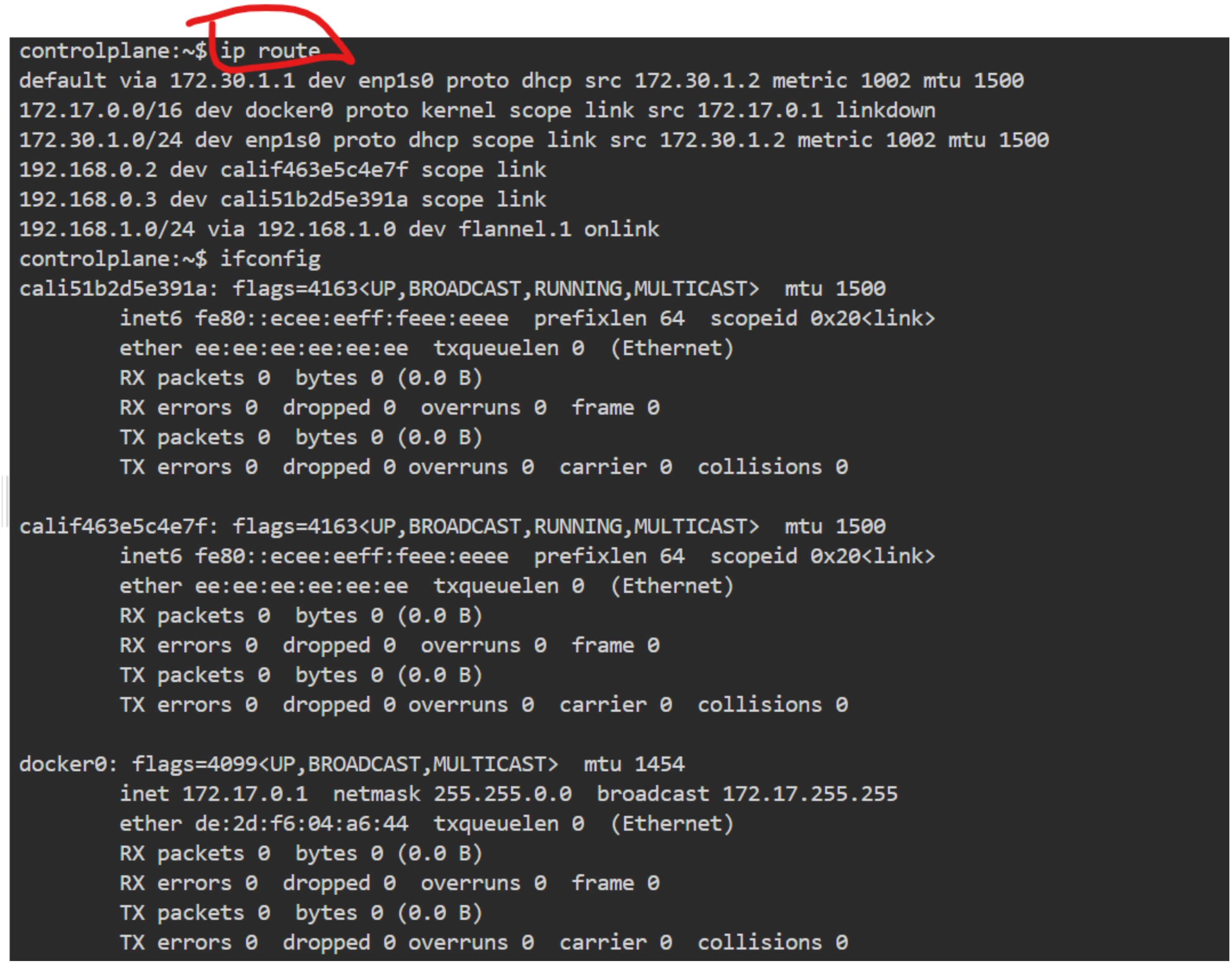
```
controlplane:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 9a:eb:5c:0e:c2:20 brd ff:ff:ff:ff:ff:ff
    inet 172.30.1.2/24 brd 172.30.1.255 scope global dynamic noprefixroute enp1s0
       valid_lft 86311631sec preferred_lft 75522431sec
    inet6 fe80::4418:bd91:d6b3:969e/64 scope link
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1454 qdisc noqueue state DOWN group default
    link/ether de:2d:f6:04:a6:44 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
4: flannel.1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc noqueue state UNKNOWN group default
    link/ether ce:57:23:1c:b6:a4 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.0/32 brd 192.168.0.0 scope global flannel.1
       valid_lft forever preferred_lft forever
    inet6 fe80::cc57:23ff:fe1c:b6a4/64 scope link
       valid_lft forever preferred_lft forever
7: calif463e5c4e7f@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether ee:ee:ee:ee:ee:ee brd ff:ff:ff:ff:ff:ff link-netns cni-3c6dfcf1-f191-b52b-fc80-5e6d550a5e1e
    inet6 fe80::ecee:eeff:feee:eeee/64 scope link
       valid_lft forever preferred_lft forever
8: cali51b2d5e391a@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether ee:ee:ee:ee:ee:ee brd ff:ff:ff:ff:ff:ff link-netns cni-86618ecf-b930-0246-f686-c37e6169b3bb
```

10. ip route : used to get private IP



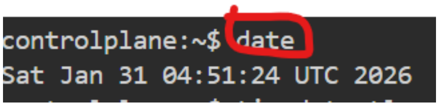
```
controlplane:~$ ip route
default via 172.30.1.1 dev enp1s0 proto dhcp src 172.30.1.2 metric 1002 mtu 1500
172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1 linkdown
172.30.1.0/24 dev enp1s0 proto dhcp scope link src 172.30.1.2 metric 1002 mtu 1500
192.168.0.2 dev calif463e5c4e7f scope link
192.168.0.3 dev cali51b2d5e391a scope link
192.168.1.0/24 via 192.168.1.0 dev flannel.1 onlink
controlplane:~$ ifconfig
cali51b2d5e391a: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::ecee:eeff:feee:eeee  prefixlen 64  scopeid 0x20<link>
        ether ee:ee:ee:ee:ee:ee  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

calif463e5c4e7f: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::ecee:eeff:feee:eeee  prefixlen 64  scopeid 0x20<link>
        ether ee:ee:ee:ee:ee:ee  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1454
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether de:2d:f6:04:a6:44  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

11.ifconfig : used to get private IP



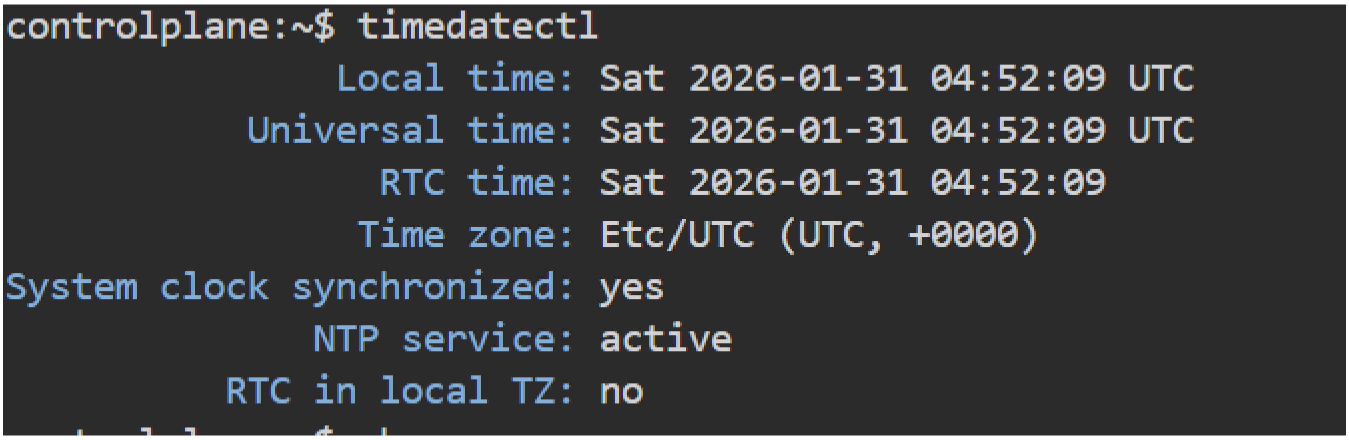
```
controlplane:~$ ip route
default via 172.30.1.1 dev enp1s0 proto dhcp src 172.30.1.2 metric 1002 mtu 1500
172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1 linkdown
172.30.1.0/24 dev enp1s0 proto dhcp scope link src 172.30.1.2 metric 1002 mtu 1500
192.168.0.2 dev calif463e5c4e7f scope link
192.168.0.3 dev cali51b2d5e391a scope link
192.168.1.0/24 via 192.168.1.0 dev flannel.1 onlink
controlplane:~$ ifconfig
cali51b2d5e391a: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::ecee:eeff:feee:eeee  prefixlen 64  scopeid 0x20<link>
        ether ee:ee:ee:ee:ee:ee  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

calif463e5c4e7f: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::ecee:eeff:feee:eeee  prefixlen 64  scopeid 0x20<link>
        ether ee:ee:ee:ee:ee:ee  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1454
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether de:2d:f6:04:a6:44  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

12.date : to get todays date



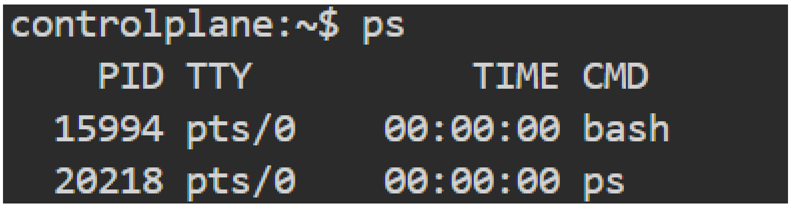
```
controlplane:~$ date
Sat Jan 31 04:51:24 UTC 2026
```

13.timedatectl : used to get timezones



```
controlplane:~$ timedatectl
               Local time: Sat 2026-01-31 04:52:09 UTC
           Universal time: Sat 2026-01-31 04:52:09 UTC
                 RTC time: Sat 2026-01-31 04:52:09
                Time zone: Etc/UTC (UTC, +0000)
System clock synchronized: yes
              NTP service: active
          RTC in local TZ: no
```

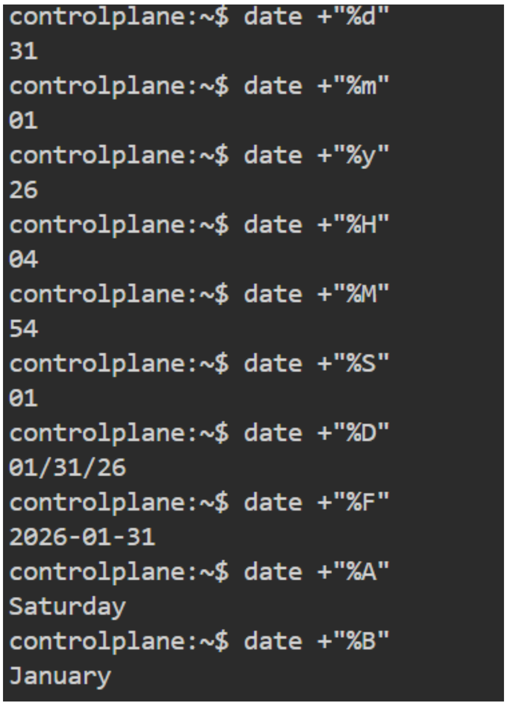
14. ps : used to see the running processors in system



```
controlplane:~$ ps
    PID TTY          TIME CMD
  15994 pts/0    00:00:00 bash
  20218 pts/0    00:00:00 ps
```

15. date, time, month, year, day

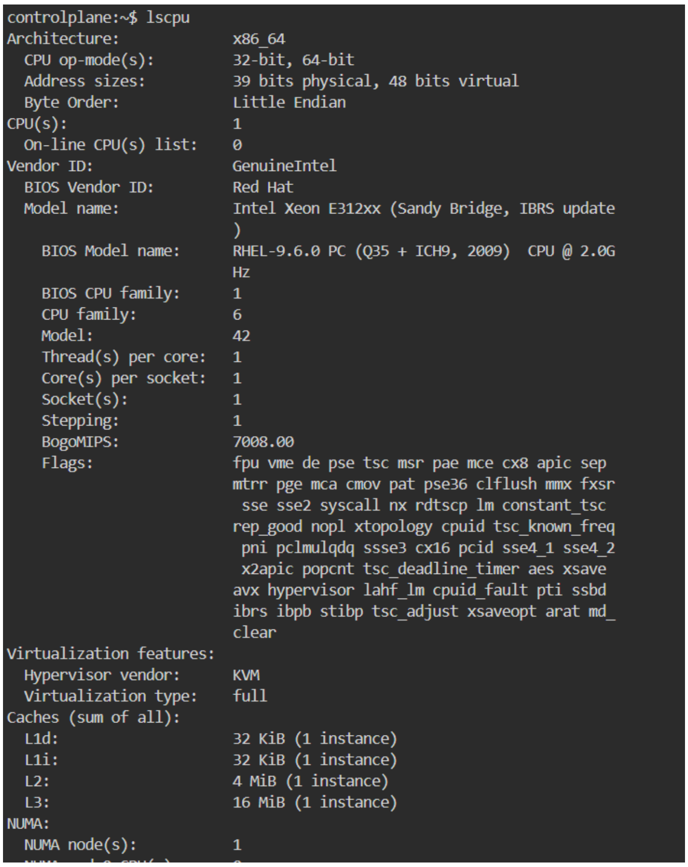
- **date +”%d”** : Prints day of the month (01-31)
- **date +”%m”** : Prints the month of the year 01-12
- **date +”%y”** : Prints only the last two digits of Year
- **date +”%H”** : Prints the hour 00-23
- **date +”%M”** : Prints the Minute of the hour 00-59
- **date +”%S”** : Prints the current seconds count in the minute (00-60)
- **date +"%D"** : Prints Date in MM/DD/YY
- **date +”%F”** : Prints only the Full date as YYYY-MM-DD
- **date +”%A”** : Prints the Day of the Week Saturday-Sunday
- **date +”%B”** : Prints the month between January-December



```
controlplane:~$ date +"%d"
31
controlplane:~$ date +"%m"
01
controlplane:~$ date +"%y"
26
controlplane:~$ date +"%H"
04
controlplane:~$ date +"%M"
54
controlplane:~$ date +"%S"
01
controlplane:~$ date +"%D"
01/31/26
controlplane:~$ date +"%F"
2026-01-31
controlplane:~$ date +"%A"
Saturday
controlplane:~$ date +"%B"
January
```

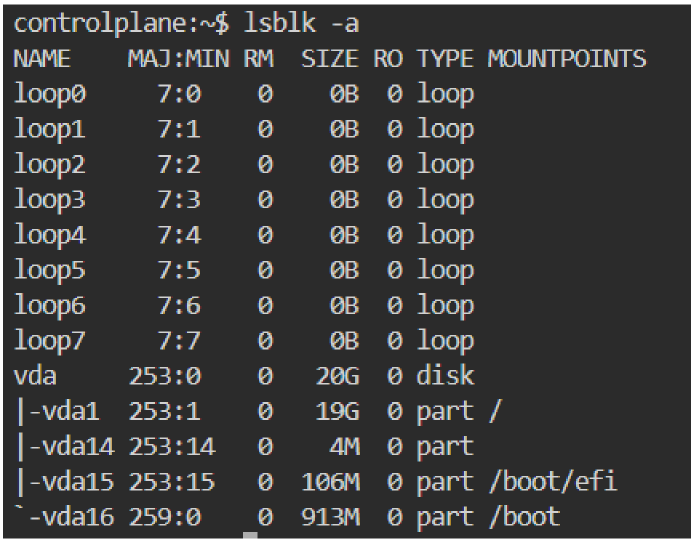
HARDWARE COMMANDS:

- **lscpu** (or) **cat /proc/cpuinfo**: Displays information about the CPU architecture



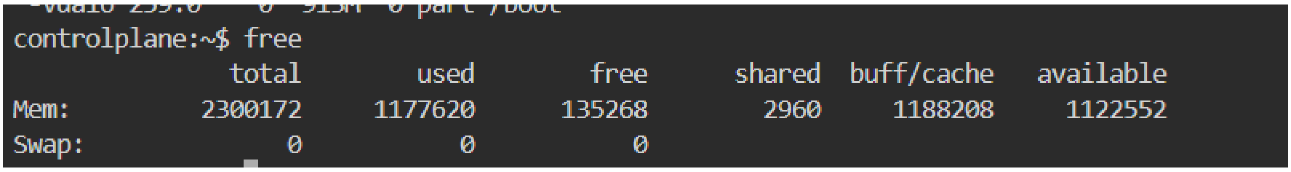
```
controlplane:~$ lscpu
Architecture:             x86_64
  CPU op-mode(s):         32-bit, 64-bit
  Address sizes:          39 bits physical, 48 bits virtual
  Byte Order:             Little Endian
CPU(s):                   1
  On-line CPU(s) list:    0
Vendor ID:                GenuineIntel
  BIOS Vendor ID:         Red Hat
  Model name:             Intel Xeon E312xx (Sandy Bridge, IBRS update
                          )
    BIOS Model name:      RHEL-9.6.0 PC (Q35 + ICH9, 2009)  CPU @ 2.0G
                          Hz
    BIOS CPU family:      1
    CPU family:           6
    Model:                42
    Thread(s) per core:   1
    Core(s) per socket:   1
    Socket(s):            1
    Stepping:             1
    BogoMIPS:             7008.00
    Flags:                fpu vme de pse tsc msr pae mce cx8 apic sep
                          mtrr pge mca cmov pat pse36 clflush mmx fxsr
                           sse sse2 syscall nx rdtscp lm constant_tsc
                          rep_good nopl xtopology cpuid tsc_known_freq
                           pni pclmulqdq ssse3 cx16 pcid sse4_1 sse4_2
                           x2apic popcnt tsc_deadline_timer aes xsave
                          avx hypervisor lahf_lm cpuid_fault pti ssbd
                          ibrs ibpb stibp tsc_adjust xsaveopt arat md_
                          clear
Virtualization features:
  Hypervisor vendor:      KVM
  Virtualization type:    full
Caches (sum of all):
  L1d:                    32 KiB (1 instance)
  L1i:                    32 KiB (1 instance)
  L2:                     4 MiB (1 instance)
  L3:                     16 MiB (1 instance)
NUMA:
  NUMA node(s):           1
```

- **lsblk -a**: Lists the information about all the block devices attached to the system



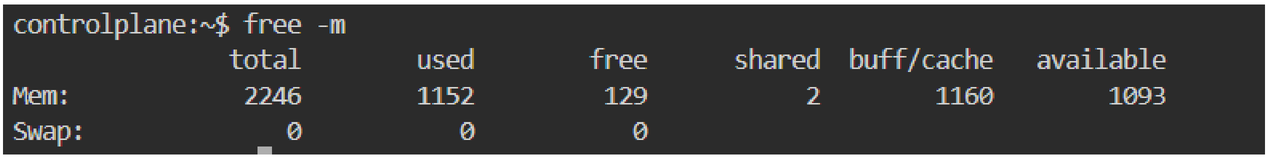
```
controlplane:~$ lsblk -a
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0     7:0    0    0B  0 loop
loop1     7:1    0    0B  0 loop
loop2     7:2    0    0B  0 loop
loop3     7:3    0    0B  0 loop
loop4     7:4    0    0B  0 loop
loop5     7:5    0    0B  0 loop
loop6     7:6    0    0B  0 loop
loop7     7:7    0    0B  0 loop
vda     253:0    0   20G  0 disk
|-vda1  253:1    0   19G  0 part /
|-vda14 253:14   0    4M  0 part
|-vda15 253:15   0  106M  0 part /boot/efi
`-vda16 259:0    0  913M  0 part /boot
```

- **free** (or) **cat /proc/meminfo**: Displays system memory (RAM) details in KB



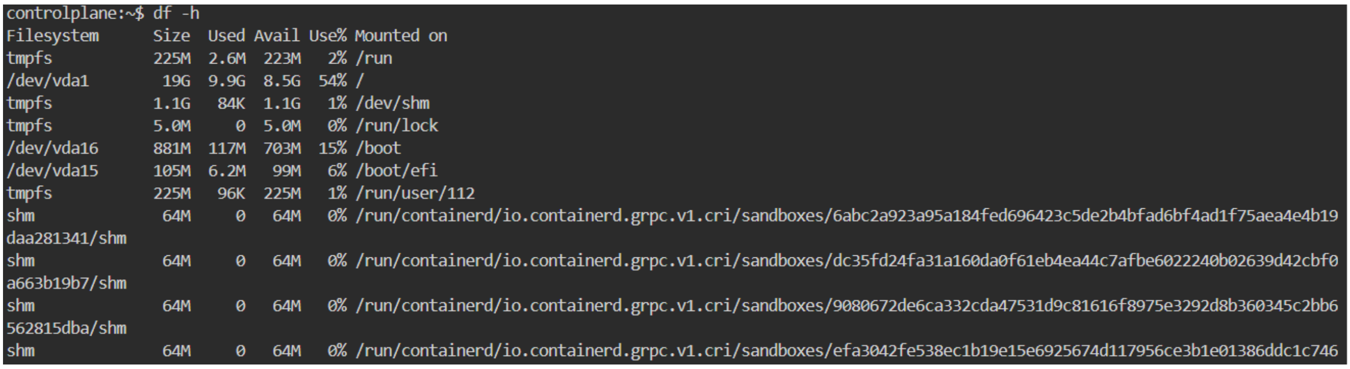
```
`-vda16 259:0    0  913M  0 part /boot
controlplane:~$ free
               total        used        free      shared  buff/cache   available
Mem:         2300172     1177620      135268        2960     1188208     1122552
Swap:              0           0           0
```

- **free -m**: Displays system memory (RAM) details in MB



```
controlplane:~$ free -m
               total        used        free      shared  buff/cache   available
Mem:            2246        1152         129           2        1160        1093
Swap:              0           0           0
```

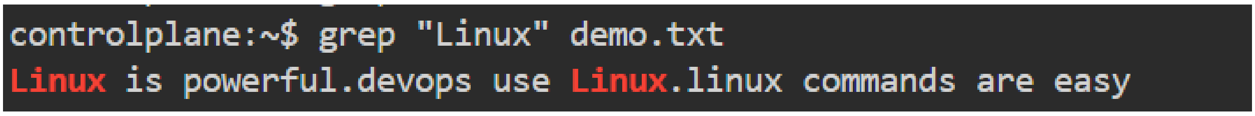
- **df -h**: Report file system disk space usage in human readable languages



```
controlplane:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           225M  2.6M  223M   2% /run
/dev/vda1        19G  9.9G  8.5G  54% /
tmpfs           1.1G   84K  1.1G   1% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
/dev/vda16      881M  117M  703M  15% /boot
/dev/vda15      105M  6.2M   99M   6% /boot/efi
tmpfs           225M   96K  225M   1% /run/user/112
shm              64M     0   64M   0% /run/containerd/io.containerd.grpc.v1.cri/sandboxes/6abc2a923a95a184fed696423c5de2b4bfad6bf4ad1f75aea4e4b19
daa281341/shm
shm              64M     0   64M   0% /run/containerd/io.containerd.grpc.v1.cri/sandboxes/dc35fd24fa31a160da0f61eb4ea44c7afbe6022240b02639d42cbf0
a663b19b7/shm
shm              64M     0   64M   0% /run/containerd/io.containerd.grpc.v1.cri/sandboxes/9080672de6ca332cda47531d9c81616f8975e3292d8b360345c2bb6
562815dba/shm
shm              64M     0   64M   0% /run/containerd/io.containerd.grpc.v1.cri/sandboxes/efa3042fe538ec1b19e15e6925674d117956ce3b1e01386ddc1c746
```

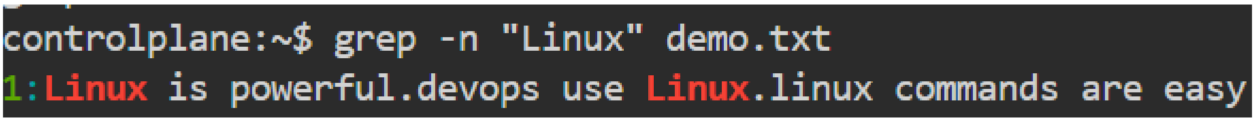
SYNTAX: grep “word” filename

1.grep “word” filename : used to search for a word in a file



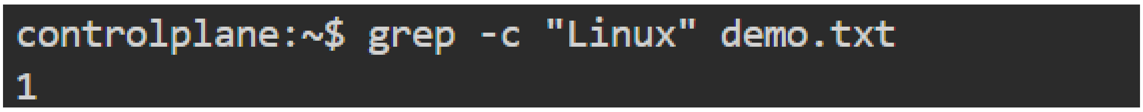
```
controlplane:~$ grep "Linux" demo.txt
Linux is powerful.devops use Linux.linux commands are easy
```

2.grep -n “word” filename : it prints the data along with line numbers



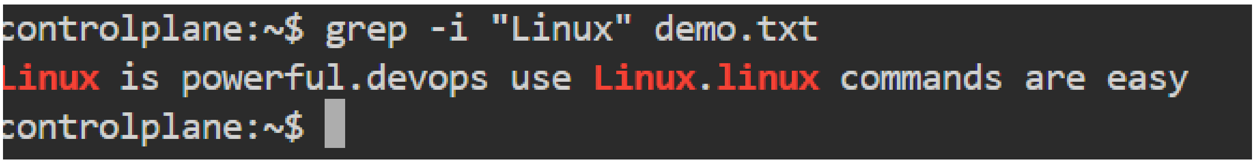
```
controlplane:~$ grep -n "Linux" demo.txt
1:Linux is powerful.devops use Linux.linux commands are easy
```

3.grep -c “word” filename: it prints no of occurances of a word



```
controlplane:~$ grep -c "Linux" demo.txt
1
```

4.grep -i “word” filename: used to search for a case-sensitive



```
controlplane:~$ grep -i "Linux" demo.txt
Linux is powerful.devops use Linux.linux commands are easy
controlplane:~$
```