ETHIOPIAN IT PARK
የኢትዮጵያ ኢንፎርሜሽን ቴክኖሎጂ ፓርክ

CITCOT

ZERGAW CLOUD

IT Infrastructure Administration

Day 5: Monitoring and Troubleshooting

Ephrem Teshale(PhD)
Tadios Abebe

July 25, 2025

Monitoring and Troubleshooting

Day 5
Training Outline

- Troubleshooting Methodology
- Common Troubleshooting Procedure
- Resolving common system problems
  - Hardware-related
  - Software-related
  - User interface-related
- Linux Troubleshooting
  - top, htop, sysstat and sar, netstat, free, df, du,
  - lsof, ping, traceroute, telnet, packet analysis,
  - w & uptime, who & whoami, history, logging, journald, journalctl
- Real-world system issues
  - user issues
  - disk issues
  - system issue
  - boot issue
  - permission issue
  - file issue
  - connectivity issue
  - package issues
  - service issue
  - remote access issue
- Windows Troubleshooting
  - System Information
  - Event Viewer
  - Performance Analysis Tools
- Knowledge Check
- Exercise

Troubleshooting Methodology

Troubleshooting involves monitoring that translates to observing log files and running performance utilities system to identify problems and their cause

Proactive Maintenance:

- Minimizing change of future problems
- Example: Perform a regular system backup
- Best for smaller or disconnected environments

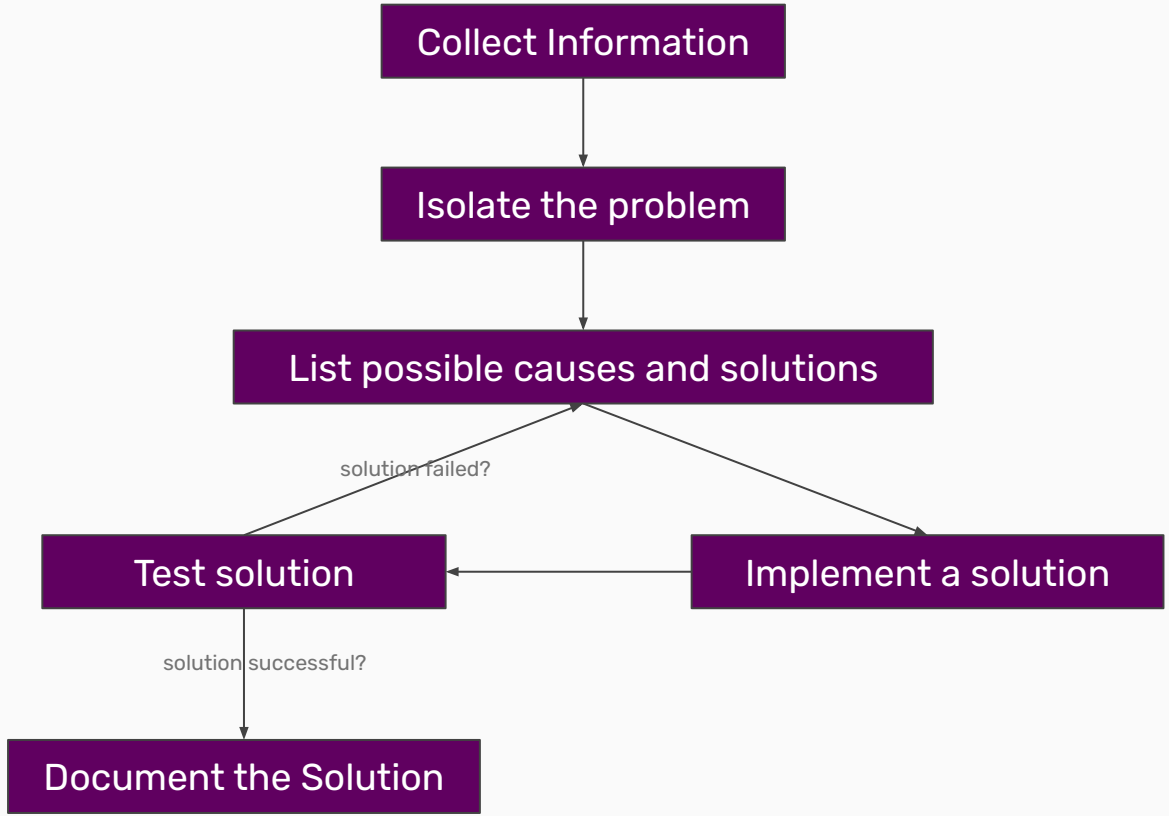
Reactive Maintenance:

- Correcting problems when they arise
- Example: Incident response

Documentation:

- System information stored in log book for future references
- All maintenance action should be documented

Common Troubleshooting Procedure



Common Troubleshooting Procedure con’t

Two troubleshooting golden rules:

Prioritize Problems:

- Prioritize problem according to severity
- Spend reasonable amount of time on each problem given its priority
- Ask for help if you can’t solve the problem

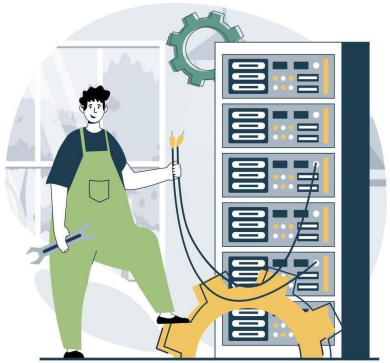
Solve the root of the problem:

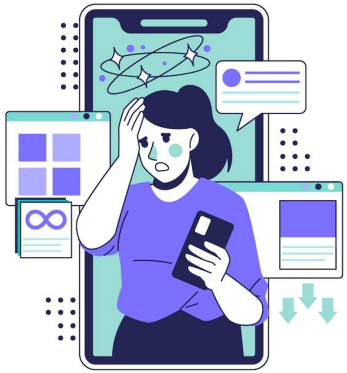
- Try to solve the root of the problem
- Avoid missing underlying cause
- Justify why a certain solution is successful

Resolving common system problems

Three main categories of problems:

1. Hardware-related
2. Software-related
3. User interface-related





Hardware Related Problems

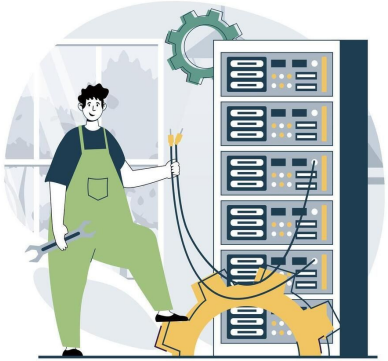
Often involve improper hardware or software configuration. Some example includes

- Failed DIMMs or Improper memory configuration
- Failed CPU or improper socket placement
- Failed PCI components
- Failed boot drives
- Loose hardware connection
- Hardware driver issue

On Linux hardware related issues are often logged on dmesg, /var/log/boot.log, /var/log/messages

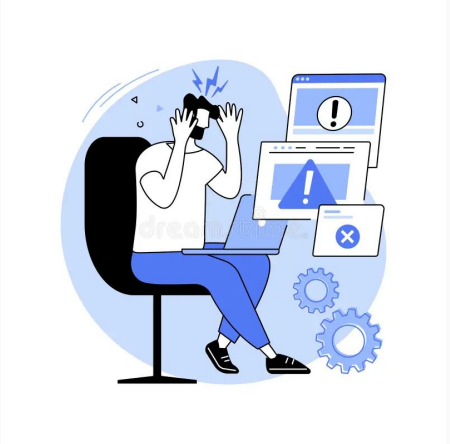
- dmesg: displays the hardware that is detected by the linux kernel
- lsusb: display a list of USB devices detected by the Linux kernel
- lspci: displays a list of PCI devices detected by the Linux kernel

You can compare outputs of commands to output of lsmod to determine if driver module is missing from kernel



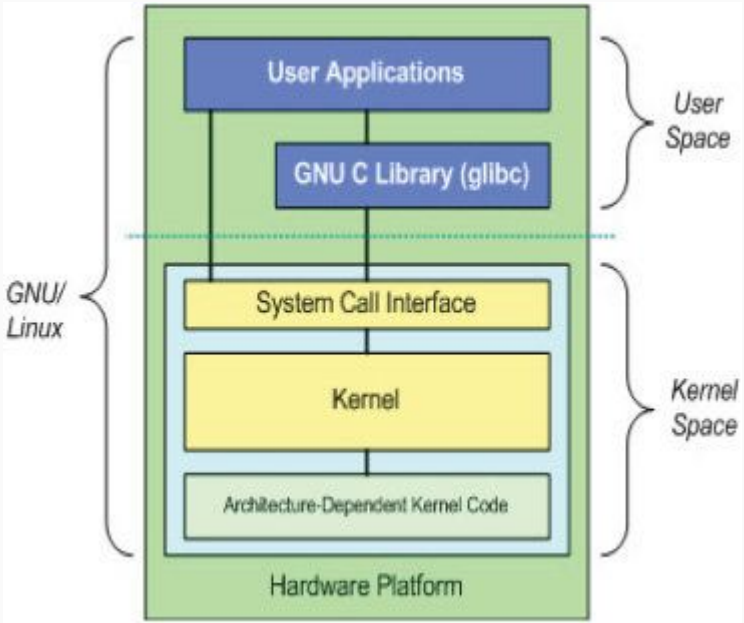
Software Related Problems

- Missing program libraries/files, process restrictions, or conflicting application dependencies
    - Prerequisite shared libraries or packages required for program execution
    - ldd: display shared libraries used by a program. .so files are shared objects similar to windows dll. i.e ldd /bin/bash
- To may running processes
    - Solve by killing parent process of zombie processes
- /var/log directory contains most system log files
- Grub problems is typically result of missing files in /boot directory
- Filesystem related issue should be checked with fsck utility. and running fsck is recommended on unmounted filesystem



Linux Troubleshooting

Linux Architecture Recap

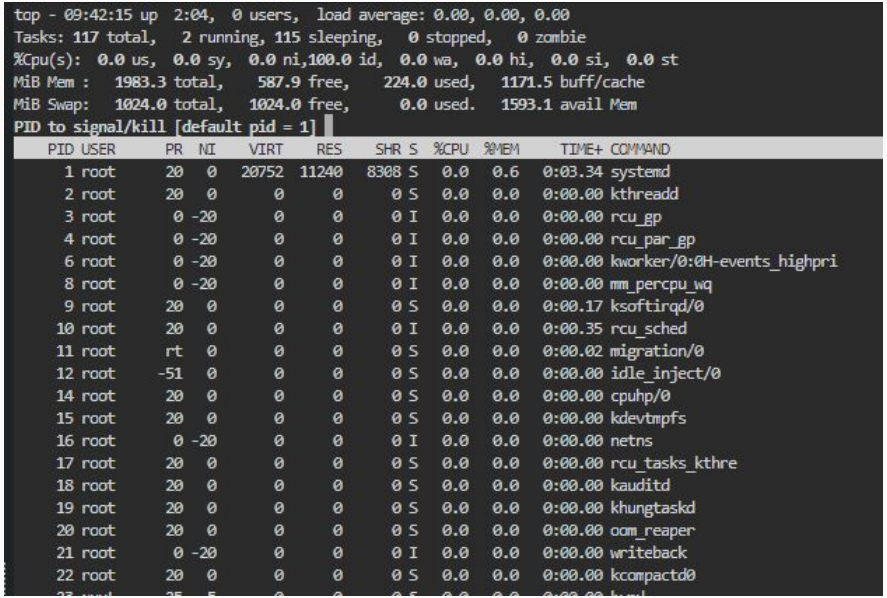


Important Tools

- top
- sysstat
- sar
- iostat
- vmstat
- free
- df
- du
- ps
- netstat
- history
- lsof
- ping
- telnet
- ifconfig
- w & uptime
- who
- whoami

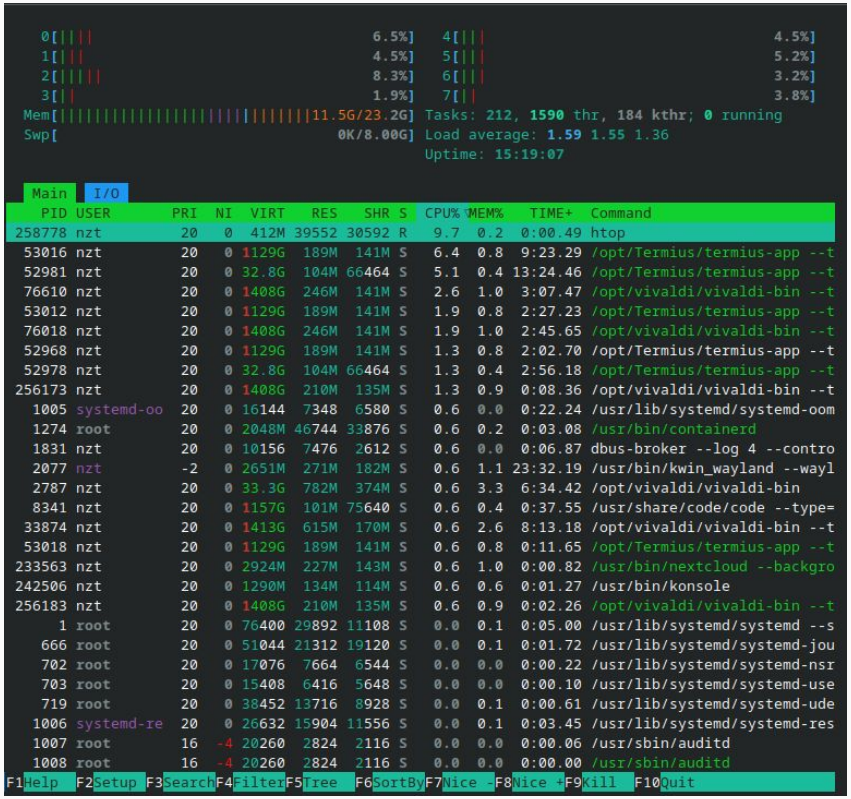
top

- Small tool, pre-installed in many unix systems.
- Display all running and active real time process in ordered list & updates it regularly.
- CPU usage, Memory usage, Swap memory, Cache size, Buffer size, Process pid, User, Commands and much more
- Show high memory and cpu utilization of a running process



htop

- The htop command is a handy interactive process viewer that's like an upgraded top version. You see a real-time table of running processes with their CPU and memory usage, similar to the top.
- Color coding so you can easily spot high-usage processes.
- Ability to scroll vertically and horizontally to see full commands, environment variables, etc.
- Easier process sorting and priority config.



sysstat and sar

- ❑ Powerful logging and monitoring tool for Linux/Unix systems.
- ❑ Contains utilities to monitor system performance and usage activity.
- ❑ Used to monitor system performance and troubleshooting problems.
- ❑ Sysstat is a go-to for power and can log and track pretty much everything going on within your linux box

Sar(System Activity Report)

- ❑ Sar is part of the sysstat package,
- ❑ Collect and display all system activities statistics
- ❑ Can monitor performance of various Linux subsystems (CPU, Memory, I/O) in real time.
- ❑ Also collect all performance data on an ongoing basis, store them, and do historical analysis to identify bottlenecks
- ❑ Collected information can be used with ksar to plot graphs.

The following is the list of utilities provided by the sysstat package

- mpstat: report individual or combined CPU related statistics
- iostat: reports CPU statistics and I/O statistics for devices, partitions, and the network filesystem.
- pidstat: Report statistics for Linux processes, including disk I/O, CPU, and memory usage
- tapestat: Reports statistics for tape drives connected to the system
- cifsiostat: Report statistics on shared file systems, printers, or serial ports over a network.
- sar: Collects, reports and saves system activity information (such as CPU, memory, disks and network interfaces usage statistics).

sar con’t

CPU Usage of ALL CPUs [**sar -u**]

- ❏ This gives the cumulative real-time CPU usage of all CPUs.
- ❏ **sar -u 1 3** Displays real time CPU usage every 1 second for three times
- ❏ **-P ALL** indicates that it should display statistics for all individual cores.

%user% - % of CPU utilization that occurred while executing at user level

%nice% - at user level with nice priority.

%system - at system level

%iowait - % of time that CPU were idle during which s/m had an outstanding disk I/O request

%idle - idle and s/m did not have an outstanding disk i/o

%steal - % of time spent in involuntary wait by cpu/cpu's

```
root@tadios:~# sar -u 1 3
Linux 6.15.5-100.fc41.x86_64 (tadios.local)     07/17/2025      _x86_64_        (8 CPU)

08:33:32 PM     CPU     %user     %nice   %system   %iowait    %steal     %idle
08:33:33 PM     all      3.02      0.00      1.76      0.00      0.00     95.23
08:33:34 PM     all      3.02      0.00      2.39      0.00      0.00     94.59
08:33:35 PM     all      3.15      0.00      1.64      0.00      0.00     95.21
Average:        all      3.06      0.00      1.93      0.00      0.00     95.01
```

```
root@tadios:~# sar -u 1 1 -P ALL
Linux 6.15.5-100.fc41.x86_64 (tadios.local)     07/17/2025      _x86_64_        (8 CPU)

08:35:07 PM     CPU     %user     %nice   %system   %iowait    %steal     %idle
08:35:08 PM     all      3.15      0.00      2.02      0.00      0.00     94.83
08:35:08 PM       0      5.10      0.00      3.06      0.00      0.00     91.84
08:35:08 PM       1      5.10      0.00      1.02      0.00      0.00     93.88
08:35:08 PM       2      4.04      0.00      2.02      0.00      0.00     93.94
08:35:08 PM       3      2.00      0.00      0.00      0.00      0.00     98.00
08:35:08 PM       4      3.00      0.00      4.00      0.00      0.00     93.00
08:35:08 PM       5      2.02      0.00      3.03      0.00      0.00     94.95
08:35:08 PM       6      2.00      0.00      1.00      0.00      0.00     97.00
08:35:08 PM       7      2.02      0.00      2.02      0.00      0.00     95.96
```

sar con’t

Memory Free and Used (**sar -r**)

- This reports the memory statistics.
- `**1 3**` reports for every 1 seconds a total of 3 times.
- Focus on 'kbmemfree' and 'kbmemused' for free and used memory

	kbmemfree	kbavail	kbmemused	%memused	kbbuffers	kbcached	kbcommit	%commit	kbactive	kbinact	kbdirty
06:00:06 PM											
06:10:29 PM	12789884	17998724	4649348	19.15	20212	6323960	34749904	106.38	8047564	1754332	1224
06:20:49 PM	12640780	17887712	4621688	19.04	20212	6501920	34308436	105.03	8008612	1785532	840
06:40:00 PM	12491016	17746228	4773268	19.66	20212	6503676	34455504	105.48	8074884	1793752	364
06:50:49 PM	12048720	17777248	4559372	18.78	20212	7144068	31102384	95.22	8177900	2024016	1580
07:00:02 PM	8375880	15059988	7087976	29.20	20212	8197220	45282208	138.63	11083968	2614116	1768
07:10:28 PM	8185016	14924232	7079296	29.16	20212	8398292	44825948	137.23	11120872	2665104	1176
08:00:59 PM	8090084	14854000	7161212	29.50	20212	8412040	44921536	137.52	11137908	2689756	124
08:10:49 PM	8028396	14883156	7131448	29.38	20212	8497072	44799304	137.15	11123448	2778912	1144
08:20:49 PM	8045588	14930036	7133592	29.39	20212	8477348	44992924	137.74	11120164	2808684	1060
08:30:49 PM	7056012	13988480	7863608	32.39	20212	8728628	47689200	146.00	11878992	2850920	18748
Average:	9775138	16004980	6206081	25.56	20212	7718422	40712735	124.64	9977431	2376512	2803

sar con’t

Overall I/O Activities (**sar -b**)

- ❏ This report I/O statistics
- ❏ `**1 3**` reports for every 1 seconds a total of 3 times

tps - Transactions per second (this include both read and write)

rtps - Read transactions per second

wtps - Write transactions per second

bread/s - Bytes read per second

bwrtn/s - Bytes written per second

```
06:00:06 PM       tps      rtps      wtps      dtps   bread/s   bwrtn/s   bdscd/s
06:10:29 PM     67.70      3.17     58.78      5.75    267.58   1826.78    828.65
06:20:49 PM     12.74      0.12     11.10      1.52      6.26    321.64    350.13
06:40:00 PM      2.61      0.01      1.78      0.81      0.81     46.64    123.45
06:50:49 PM     34.21      7.60     25.04      1.57    908.77    761.20    752.59
07:00:02 PM     86.87     19.90     63.26      3.71   1439.35   1995.83   1009.40
07:10:28 PM     16.65      0.01     14.94      1.70      0.41    425.34    224.24
08:00:59 PM      2.79      0.00      2.53      0.26      0.02     81.82     20.34
08:10:49 PM     41.98      0.51     38.09      3.37     14.05   1019.17    399.63
08:20:49 PM     19.06      0.01     15.80      3.25      0.32    430.61    448.34
08:30:49 PM     25.76      0.57     22.17      3.02     21.24    600.50    364.77
Average:        21.44      2.06     17.59      1.78    174.59    522.38    314.95
```

```
root@tadios:~# sar -b 1 3
Linux 6.15.5-100.fc41.x86_64 (tadios.local)     07/17/2025      _x86_64_        (8 CPU)

08:40:33 PM       tps      rtps      wtps      dtps   bread/s   bwrtn/s   bdscd/s
08:40:34 PM      0.00      0.00      0.00      0.00      0.00      0.00      0.00
08:40:35 PM      0.00      0.00      0.00      0.00      0.00      0.00      0.00
08:40:36 PM      0.00      0.00      0.00      0.00      0.00      0.00      0.00
Average:         0.00      0.00      0.00      0.00      0.00      0.00      0.00
```

Sar Cheat Sheet

Command	Description
sar -o filename	Save SAR output to a file
sar --help	Show all available SAR options
sadf -d /var/log/sa/saXX	Export SAR data in CSV/JSON format
sar 1 5	Display real-time stats (1-second intervals 5 times)
sar -p	Pretty-print device names (for -d)

Command	Description
sar -s HH:MM:SS	Start reporting from a specific time
sar -e HH:MM:SS	End reporting at a specific time
sar -f /var/log/sa/saXX	Read from a specific SAR log file (replace XX with day
sar -i SEC	Set interval in seconds (e.g. sar -i 5)

Command	Description
sar	Display CPU usage by default (today's data)
sar -A	Display all system activity (CPU, memory, disk, network, etc.)
sar -u	Show CPU utilization (default)
sar -P ALL	Show CPU stats for all cores
sar -q	Display system load and run queue stats
sar -r	Show memory utilization (RAM & swap)
sar -S	Display swap space usage
sar -b	Show I/O and transfer rate stats
sar -d	Display disk activity (per device)
sar -n DEV	Show network interface stats
sar -n TCP	Display TCP statistics
sar -n EDEV	Show network error statistics
sar -w	Display context switches per second
sar -v	Show inode, file, and other kernel stats

Metric	Command	What to Check
CPU Usage	sar -u	High %idle is good %user + %system should not be consistently high
Memory Usage	sar -r	Check kbmemfree and %memused
Swap Usage	sar -S	High swap usage indicates memory pressure
Disk I/O	sar -d	High %util means disk is a bottleneck
Network Traffic	sar -n DEV	Check rxkB/s and txkB/s for bandwidth usage
Load Average	sar -q	ldavg-1 > CPU cores indicates high load

netstat

- Monitoring incoming and outgoing network packets statistics as well as interface statistics.
- Very useful tool for every system administrator to monitor network performance and troubleshoot network related problems

netstat -a - Show all listening and non-listening ports (TCP & UDP)

netstat -t - Display TCP connections only

netstat -u - Display UDP connections only

netstat -l - Show only listening ports

netstat -i - Display network interface statistics

netstat -rn - Display numeric routing table (no DNS resolution)

netstat -tulnp - Show all listening TCP/UDP ports with PIDs

netstat -ap - Show process name/PID using the port (requires sudo)

netstat -ano - Display all connections with PID & timers (Windows)

```
root@tadios:~# netstat -ltunpa
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:8829          0.0.0.0:*               LISTEN      8199/Code --standar
tcp        0      0 127.0.0.1:8828          0.0.0.0:*               LISTEN      8209/Code --standar
tcp        0      0 127.0.0.1:8830          0.0.0.0:*               LISTEN      8221/Code --standar
tcp        0      0 127.0.0.1:42655         0.0.0.0:*               LISTEN      8209/Code --standar
tcp        0      0 127.0.0.1:43657         0.0.0.0:*               LISTEN      8199/Code --standar
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      1006/systemd-resolv
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      1256/cupsd
tcp        0      0 127.0.0.54:53           0.0.0.0:*               LISTEN      1006/systemd-resolv
tcp        0      0 0.0.0.0:5355            0.0.0.0:*               LISTEN      1006/systemd-resolv
tcp        0      0 127.0.0.1:39371         0.0.0.0:*               LISTEN      8221/Code --standar
tcp        0      0 0.0.0.0:7070            0.0.0.0:*               LISTEN      1319/anydesk
tcp        0      0 192.168.210.170:47654   18.172.213.40:443       ESTABLISHED 52953/Termius --sta
tcp        0      0 192.168.210.170:53082   142.251.209.42:443      ESTABLISHED 2985/vivaldi-bin --
tcp        0      0 192.168.210.170:54166   172.217.170.170:443     ESTABLISHED 2985/vivaldi-bin --
tcp        0      0 192.168.210.170:36936   18.97.36.70:443         ESTABLISHED 2985/vivaldi-bin --
tcp        0      0 192.168.210.170:49450   142.250.180.182:443     ESTABLISHED 2282/plasmashell
tcp        0      0 192.168.210.170:52728   31.209.137.10:61613     ESTABLISHED 2985/vivaldi-bin --
tcp        0      0 192.168.210.170:60432   196.189.119.118:443     ESTABLISHED 2583/nextcloud
tcp        0      0 192.168.210.170:56674   108.177.127.188:443     ESTABLISHED 2985/vivaldi-bin --
tcp        0      0 192.168.210.170:41070   13.57.89.172:443        ESTABLISHED 52953/Termius --sta
tcp        0      0 192.168.210.170:50610   138.199.27.227:443      ESTABLISHED 1319/anydesk
tcp       32      0 192.168.210.170:35156   31.209.137.10:443       CLOSE_WAIT  2985/vivaldi-bin --
tcp        0      0 192.168.210.170:35168   31.209.137.10:443       CLOSE_WAIT  2985/vivaldi-bin --
tcp6       0      0 ::1:631                 :::*                    LISTEN      1256/cupsd
tcp6       0      0 :::1716                 :::*                    LISTEN      2498/kdeconnectd
tcp6       0      0 :::5355                 :::*                    LISTEN      1006/systemd-resolv
tcp6       0      0 :::7070                 :::*                    LISTEN      1319/anydesk
udp        0      0 127.0.0.54:53           0.0.0.0:*                           1006/systemd-resolv
udp        0      0 127.0.0.53:53           0.0.0.0:*                           1006/systemd-resolv
udp        0      0 192.168.210.170:68      192.168.210.167:67      ESTABLISHED 1172/NetworkManager
udp        0      0 127.0.0.1:323           0.0.0.0:*                           1058/chronyd
udp        0      0 0.0.0.0:50001           0.0.0.0:*                           1319/anydesk
udp        0      0 0.0.0.0:36065           0.0.0.0:*                           2498/kdeconnectd
udp        0      0 192.168.210.170:37270   172.217.170.206:443     ESTABLISHED 2985/vivaldi-bin --
```

free

- Built-in command that displays the total amount of free and used physical memory on your machine
- Also displays the buffers used by the kernel at that given moment

free - show memory and swap usage in byte

free -m - show memory and swap usage in MB

free -h - show memory and swap usage in human readable format

```
root@tadios:~# free
               total        used        free      shared  buff/cache   available
Mem:        24275880    11642824     5142548     2025124     9919480    12633056
Swap:        8388604           0     8388604
root@tadios:~# free -m
               total        used        free      shared  buff/cache   available
Mem:           23706       11344        5047        1959        9668       12362
Swap:           8191           0        8191
root@tadios:~# free -h
               total        used        free      shared  buff/cache   available
Mem:            23Gi        11Gi       5.0Gi       1.9Gi       9.4Gi        12Gi
Swap:          8.0Gi          0B       8.0Gi
root@tadios:~# free -g
               total        used        free      shared  buff/cache   available
Mem:               23          11           4           1           9          12
Swap:               7           0           7
```

df

- ❏ df is an abbreviation for disk free
- ❏ Pre-installed program in all unix systems used to display the amount of available disk space from filesystem which the user have access to

df - shows filesystem usage in bytes

df -h - shows filesystem usage in human readable format

```
root@tadios:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/nvme0n1p3  476G   75G  398G  16% /
devtmpfs         12G     0   12G   0% /dev
tmpfs            12G  173M   12G   2% /dev/shm
efivarfs        438K  197K  237K  46% /sys/firmware/efi/efivars
tmpfs           4.7G  2.3M  4.7G   1% /run
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-journald.service
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-network-generator.service
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-udev-load-credentials.service
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-sysctl.service
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-tmpfiles-setup-dev-early.service
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-tmpfiles-setup-dev.service
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-vconsole-setup.service
/dev/nvme0n1p3  476G   75G  398G  16% /home
tmpfs            12G  102M   12G   1% /tmp
/dev/nvme0n1p2  974M  376M  531M  42% /boot
```

du

- Linux du (disk usage) is standard Unix/Linux command
- Used to check the information of disk usage of files and directories on a machine
- Has many parameter option that can be used to get the result in many formats
- Also displays the file and directory sizes in a recursive manner

du - recursively show the size of file and folder starting from current directory

du -s - summarize the current directory size

du -h - display folder sizes in human readable formats

du -d 1 - display folder sizes under the current directory

```
root@tadios:~# du -sh /var/log
1.8G    /var/log
root@tadios:~# du -d 1 /var/log -h
3.3M    /var/log/anaconda
26M     /var/log/audit
0       /var/log/blivet-gui
0       /var/log/chrony
0       /var/log/cups
1.7G    /var/log/journal
0       /var/log/mariadb
0       /var/log/ppp
0       /var/log/private
0       /var/log/qemu-ga
0       /var/log/samba
0       /var/log/speech-dispatcher
36K     /var/log/sssd
3.0M    /var/log/tuned
12K     /var/log/passim
32K     /var/log/timeshift
17M     /var/log/sa
76K     /var/log/letsencrypt
0       /var/log/glusterfs
4.0K    /var/log/libvirt
0       /var/log/swtpm
64K     /var/log/ipp-usb
8.0K    /var/log/httpd
1.8G    /var/log
```

lsof

- lsof meaning 'list open files' is used to find out which files are opened by which process
- when a disk cannot be unmounted as it says the files are being used

lsof -u user : list user specific opened files

kill -9 `lsof -t -u user' : kill all activities of a particular user

lsof -p 1 : search by pid

```
root@tadios:~# lsof | head -n 10
lsof: WARNING: can't stat() fuse.portal file system /run/user/1000/doc
      Output information may be incomplete.
COMMAND      PID    TID TASKCMD              USER   FD      TYPE             DEVICE    SIZE/OFF       NODE NAME
systemd        1                            root  cwd       DIR              0,44         212        256 /
systemd        1                            root  rtd       DIR              0,44         212        256 /
systemd        1                            root  txt       REG              0,44      115016    2643440 /usr/lib/systemd/systemd
systemd        1                            root  mem       REG              0,40                2643440 /usr/lib/systemd/systemd (path dev=0,44)
systemd        1                            root  mem       REG              0,40                 683421 /usr/lib64/libzstd.so.1.5.7 (path dev=0,44)
systemd        1                            root  mem       REG              0,40                 262437 /usr/lib64/libbpf.so.1.4.7 (path dev=0,44)
systemd        1                            root  mem       REG              0,40                2706389 /etc/selinux/targeted/contexts/files/file_contexts.bin (path dev=0,44)
systemd        1                            root  mem       REG              0,40                  40031 /usr/lib64/libpcre2-8.so.0.13.0 (path dev=0,44)
systemd        1                            root  mem       REG              0,40                 522044 /usr/lib64/libcrypto.so.3.2.4 (path dev=0,44)
```

ping

- Used to find out whether the peer host/gateway is reachable
- How much time it takes for that data to be exchanged
- Default ping waits for 1 second before sending the next packet. This can be changed using the -i flag

ping 8.8.8.8

ping -i 5 8.8.8.8

ping -c 3 8.8.8.8

```
root@tadios:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=107 time=194 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=107 time=217 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=107 time=171 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=107 time=160 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 160.140/185.579/217.133/21.860 ms
```

traceroute

- The traceroute command is used to see how packets are getting routed. It works by sending packets with increasing TTL values, starting with 1. So the first router gets the packet, and it decrements the TTL value by one, thus dropping the packet.
- The router sends back an ICMP Time Exceeded message back to us. And then the next packet gets a TTL of 2, so it makes it past the first router, but when it gets to the second router the TTL is 0 and it returns another ICMP Time Exceeded message.
- Traceroute works this way because as it sends and drops packets it is build a list of routers that the packets traverse, until it finally gets to its destination and gets an ICMP Echo Reply message.

```
nzt@tadios:~$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  _gateway (192.168.210.167)  3.707 ms  4.152 ms  4.287 ms
 2  * * *
 3  10.224.231.1 (10.224.231.1)  37.653 ms  47.116 ms  37.623 ms
 4  10.255.254.5 (10.255.254.5)  47.361 ms  48.713 ms  48.697 ms
 5  10.184.10.19 (10.184.10.19)  71.315 ms  61.982 ms  47.552 ms
 6  10.1.41.6 (10.1.41.6)  47.317 ms  42.606 ms  48.997 ms
 7  10.1.41.5 (10.1.41.5)  42.267 ms  35.377 ms  35.308 ms
 8  * 41.79.199.136 (41.79.199.136)  49.335 ms  51.793 ms
 9  41.79.199.138 (41.79.199.138)  110.381 ms  111.098 ms  120.223 ms
10  192.178.105.75 (192.178.105.75)  140.806 ms 41.189.225.170 (41.189.225.170)  108.575 ms 192.178.105.157 (192.178
.105.157)  141.932 ms
11  * 192.178.105.75 (192.178.105.75)  125.825 ms 173.194.123.48 (173.194.123.48)  102.405 ms
12  108.170.233.243 (108.170.233.243)  134.791 ms dns.google (8.8.8.8)  107.372 ms  106.575 ms
nzt@tadios:~$
```

Telnet and nc

- Both telnet and nc can be used to troubleshoot network and service connectivity
- Telnet is a remote terminal connection tool
- Netcat is a general purpose tcp/upd tool

telnet localhost 80

nc -zv localhost 80

nc -ul localhost 514 (send udp packet)

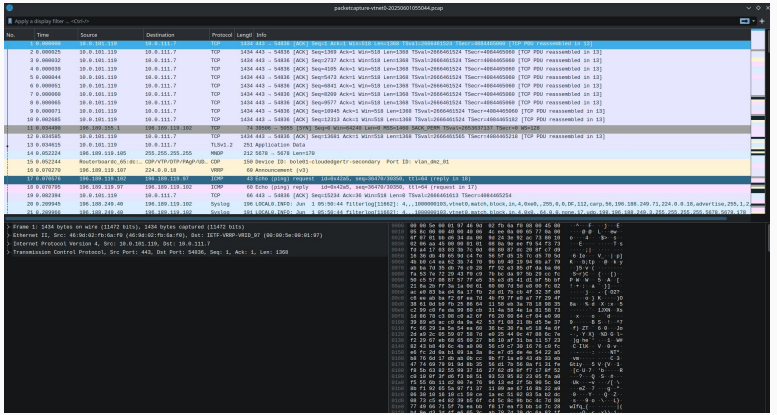
```
nzt@tadios:~$ telnet localhost 7070
Trying ::1...
Connected to localhost.
Escape character is '^]'.
```

```
nzt@tadios:~$ nc -zv localhost 7070
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Connected to ::1:7070.
Ncat: 0 bytes sent, 0 bytes received in 0.01 seconds.
nzt@tadios:~$
```

packet analysis

There are two extremely popular packet analyzers, Wireshark and tcpdump. These tools scan your network interfaces, capture the packet activity, parse the packages and output the information for us to see.

```
nzt@tadios:~$ sudo tcpdump
[sudo] password for nzt:
dropped privs to tcpdump
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on wlp0s20f3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
23:35:12.623398 IP tadios.local.39038 > lb-140-82-113-22-iad.github.com.https: Flags [S], seq 3696725201, win 64240,
 options [mss 1460,sackOK,TS val 3187073483 ecr 0,nop,wscale 7], length 0
23:35:12.689990 IP tadios.local.36808 > one.one.one.one.domain: 17396+ [1au] PTR? 22.113.82.140.in-addr.arpa. (55)
23:35:12.997832 IP mba01s08-in-f10.1e100.net.https > tadios.local.36372: Flags [P.], seq 3613775291:3613775376, ack
791685522, win 1109, options [nop,nop,TS val 1032939939 ecr 3836582936], length 85
23:35:13.000175 IP tadios.local.36372 > mba01s08-in-f10.1e100.net.https: Flags [P.], seq 1:36, ack 85, win 481, opti
ons [nop,nop,TS val 3836587441 ecr 1032939939], length 35
23:35:13.000556 IP tadios.local.36372 > mba01s08-in-f10.1e100.net.https: Flags [P.], seq 36:71, ack 85, win 481, opt
ions [nop,nop,TS val 3836587442 ecr 1032939939], length 35
23:35:13.202671 IP mba01s08-in-f10.1e100.net.https > tadios.local.36372: Flags [.], ack 36, win 1109, options [nop,n
op,TS val 1032940211 ecr 3836587441], length 0
23:35:13.202672 IP mba01s08-in-f10.1e100.net.https > tadios.local.36372: Flags [.], ack 71, win 1109, options [nop,n
op,TS val 1032940211 ecr 3836587442], length 0
```



ETHIOPIAN IT PARK

CITCOT

ZERGAW CLOUD

w & uptime

w - displays information about the users currently on the machine, and their processes

uptime - tell how long the system has been running

```
nzt@tadios:~$ w
 22:39:04 up  4:44,  4 users,  load average: 0.77, 0.98, 1.56
USER     TTY        LOGIN@   IDLE   JCPU   PCPU WHAT
nzt      tty2      17:55    4:44m  0.13s  0.13s /usr/bin/startplasma-wayland
root     pts/1     20:23    7:22   3.35s  3.29s sudo su
root               20:23    6:29   0.00s  0.30s /usr/lib/systemd/systemd --user
nzt                17:55    6:29   0.00s  1.63s /usr/lib/systemd/systemd --user
nzt@tadios:~$
```

```
nzt@tadios:~$ uptime
 22:39:33 up  4:44,  4 users,  load average: 0.52, 0.90, 1.52
nzt@tadios:~$
```

who & whoami

who - prints information about all users who are currently logged in

Displays the username line and time of all currently logged-in sessions

whoami - prints the username associated with the current effective user ID

```
nzt@tadios:~$ who
nzt      tty2         2025-07-17 17:55
nzt      pts/0        2025-07-17 17:55 (:0)
nzt      pts/1        2025-07-17 18:07 (:0)
nzt      pts/4        2025-07-17 18:55 (:0)
nzt      pts/5        2025-07-17 20:23 (:0)
nzt@tadios:~$
```

```
nzt@tadios:~$ whoami
nzt
nzt@tadios:~$
```

History

The history command can be used to list bash's log of the commands you have typed:

The history command performs one of several operations related to recently-executed commands recorded in a history

```
nzt@tadios:~$ history | tail -n 15
 1003  nc zergaw.com 443
 1004  telnet -u
 1005  netstat -u
 1006  nc -u localhost 32829
 1007  ss -ltunp
 1008  ss -lunp
 1009  ss -ltnp
 1010  telnet localhost 7070
 1011  nc -zv localhost 7070
 1012  w
 1013  uptime
 1014  who
 1015  whoami
 1016  history | tail
 1017  history | tail -n 15
nzt@tadios:~$
```

Logging

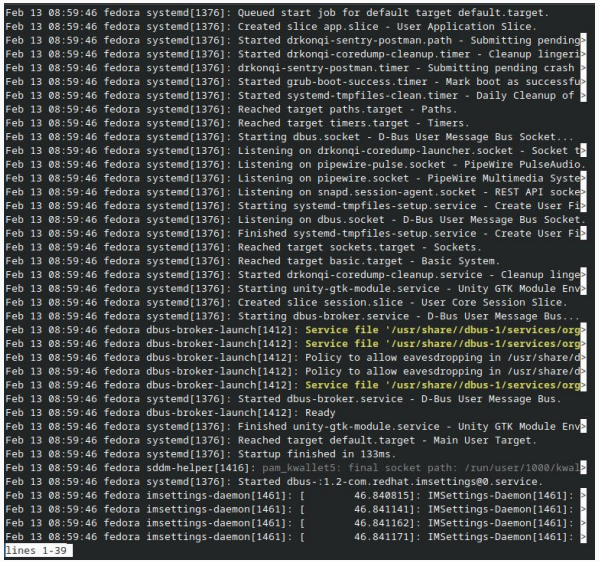
- All Linux systems create and store information log files for boot processes, applications, and other events. These files are a helpful resource for troubleshooting system issues.
- Most Linux log files are stored in plain text files (ASCII format) in the /var/log directory and subdirectories. Logs are generated by the Linux system daemon log, syslogd, or rsyslogd. Properly managing these logs ensures essential data is readily available for analysis and auditing.
- System log files in Linux contain information about the core operating system activities, including boot processes, kernel messages, and hardware events. Some example of log files inside the /var/log directory include
  - /var/log/syslog
  - /var/log/kern.log
  - /var/log/dmesg
  - /var/log/boot.log
  - /var/log/auth.log
  - /var/log/cron.log

```
nzt@tadios:~$ ls /var/log
anaconda           btmp-20250702  dnf.librepo.log  maillog           private           spooler-20250619
anydesk.trace      chrony         dnf.log          maillog-20250611  qemu-ga           spooler-20250626
audit              cron           dnf.rpm.log      maillog-20250619  README            spooler-20250703
blivet-gui         cron-20250611  dpkg.log         maillog-20250626  sa                sssd
boot.log           cron-20250619  firewalld        maillog-20250703  samba             swtpm
boot.log-20250617  cron-20250626  fsck_hfs.log     mariadb           secure            timeshift
boot.log-20250619  cron-20250703  glusterfs        messages          secure-20250611   tuned
boot.log-20250625  cups           httpd            messages-20250611 secure-20250619   wtmp
boot.log-20250628  dnf5.log       ipp-usb          messages-20250619 secure-20250626
boot.log-20250702  dnf5.log.1     journal          messages-20250626 secure-20250703
boot.log-20250703  dnf5.log.2     lastlog          messages-20250703 speech-dispatcher
boot.log-20250709  dnf5.log.3     letsencrypt      passim            spooler
btmp               dnf5.log.4     libvirt          ppp               spooler-20250611
nzt@tadios:~$
```

Essential commands the are needed to interact with log files include cat, less, more, tail, head, grep

Journald

- Systemd logs all Linux messages from the kernel and system processes. One of the most powerful systemd functionalities is the logging features. Systemd provides a centralized solution for logging all kernel and user processes through logs known as journals.
- The journalctl command queries and manipulates the journal data collected by the journald daemon.
- Without any parameters, the journalctl command outputs the entire journal contents starting from the oldest entry.
- To jump to the pager end and display the most recent entries, use the -e option.
- To control how many lines display in the output, use the -n option followed by the number of lines.
- To limit the logs to the current boot, use the -b tag without any parameters
- Jump to a specific boot by adding an offset parameter. For example, journalctl -b 1



Journalctl

- An alternative way to see a specific boot is to use a boot ID. Fetch the boot IDs using --list-boots with:

  journalctl --list-boots

  journalctl -b cc07702b00884ec59312ece62604cac8

- Filter the journal by specifying a time limit. The two options for limiting since or until a specified time are:

  journalctl -S 2022-04-02 -U 2022-04-22

  journalctl -S "50 minutes ago"

- Filter the logs by the specific systemd unit using the -u tag and providing the unit name. For example, to filter only the Jenkins service unit records, run:

  journalctl -u apache2

- To display only the kernel journal log messages, use the -k option:

  journalctl -k

- Use the -f or --follow tag to print the most recent logs continuously:

  journalctl -f

Real-World System Issues and Troubleshooting (Password Reset)

Using Recovery Mode

For ubuntu/debian systems

- Reboot and hold shift during startup to access grub menu, and enter edit mode by clicking on e
- find the line that starts with linux or linux16. This line specifies the boot parameters. At this line, locate 'ro quiet', replace 'ro' to 'rw' then add the word single or init=/bin/bash, depending on your distribution and setup. and click ctrl + x of F10
- Here, remount the root filesystem at writable and reset the root password

  mount -o remount,rw /

  passwd root
- reboot the system

For centos/rhel/fedora

- Reboot and press e at the GRUB menu and find the line starting with linux and add rd.break at the end
- Next, mount the root filesystem and reset the root password

  mount -o remount,rw /sysroot

  chroot /sysroot

  passwd root
- Finally, exit and reboot

Using Live OS

- Boot into he live environment
- Mount the root partition of the system

  mount /dev/sda1 /mnt
- Switch as a root user for the mounted system

  sudo chroot /mnt
- Verify the existence of the user you are trying to change the password for

  cat /mnt/etc/passwd
- Change the password for the user

  passwd username
- Enter the password twice
- Exit the mounted filesystem
- Unmount the file system
- Reboot

Prevention Tips:

- Create a passwordless sudo user to avoid being locked out of root access.
- Use SSH keys instead of passwords for authentication.

Real-World System Issues and Troubleshooting (Disk Full)

Your system displays a “No space left on device” error, preventing software updates, logging, and normal operations.

Step 1: Check Disk Usage

The solution is, first you need to check how much space is used on each partition on your system using the df command.

```
df -h
```

Step 2: Find and Delete Large Files

```
du -ah / | sort -rh | head -10
```

Step 3: Remove Unnecessary Logs

```
sudo journalctl --vacuum-time=2d
sudo apt autoclean
```

Step 4: Remove Old Kernels (Ubuntu/Debian)

```
sudo apt autoremove --purge
```

Prevention Tips:

- Set Up Log Rotation: Use logrotate to automatically manage log file sizes and retention periods.
- Monitor Disk Usage: Install tools like ncdu to track disk usage and identify space hogs.
- Regular Cleanups: Schedule periodic cleanups to remove temporary files, caches, and unused packages.
- ncdu is a good visual utility to monitor you disk space usage

Real-World System Issues and Troubleshooting (Unresponsive System)

You are managing a Linux server, and suddenly, it stops responding and you try connecting via SSH, but the connection times out or refuses to establish.

Step 1: Access the Server Locally or via TTY

If SSH isn't working, try accessing the server directly or through a TTY session:

Step 2: Check System Load

check the system's load and resource usage(uptime), which will show the system's load averages over 1, 5, and 15 minutes. A load value higher than the number of CPU cores indicates high demand.

Step 3: Identify and Kill Runaway Processes

To identify the most resource-intensive processes, run: ps aux --sort=-%cpu | head

Step 4: Check System Logs

sudo tail -f /var/log/syslog

Or

sudo dmesg | tail

Step 5: Reboot

Identify Hardware Issues (Advanced): In extreme cases, hardware problems like overheating or failing components might cause crashes.

Real-World System Issues and Troubleshooting (Boot failure)

Symptoms: Kernel panic, GRUB errors, initramfs prompt.

- ❏ Use a live USB to chroot (chroot /mnt/sysimage).
- ❏ Reinstall GRUB: **grub-install** /dev/sda + update-grub.
- ❏ Update initramfs: **update-initramfs**
- ❏ Check logs: **journalctl -b -1** (previous boot), **journalctl -xb**, /var/log/boot.log or /var/log/syslog.
- ❏ Repair filesystems: **fsck /dev/sdX**.

Prevention:

- ❏ Keep system updated
- ❏ Avoid manual changes to bootloader configs
- ❏ Use LTS kernels in production
- ❏ Regular backups of /boot, /etc, and critical configs.
- ❏ Test kernel updates in staging.
- ❏ Use UUIDs in /etc/fstab instead of device names.

Real-World System Issues and Troubleshooting (Permission)

These errors indicate that you lack the necessary authorization to perform an action. For example, attempting to edit a system file without root privileges might result in a permission denied error.

- **Verify Ownership:** Use the ls -l command to view file ownership and permissions. Ensure you have the appropriate permissions (read, write, execute) for the intended action.
- **Utilize sudo:** If necessary, use the sudo command to temporarily gain root privileges. However, use sudo with caution, as it grants elevated access to the system.
- **Change File Ownership (Advanced):** In specific scenarios, you might need to adjust file ownership using commands like chown or chgrp. Consult the man pages for proper usage.

Prevention:

- Use ACLs for complex permissions
- Enforce user/group standards

Real-World System Issues and Troubleshooting (File Not Found)

These errors indicate that the file you're trying to access doesn't exist or is located in a different directory.

- ❏ Double-Check File Path: Ensure the file path you're using is accurate, including case sensitivity in Linux.
- ❏ Utilize Tab Completion: Leverage the Tab key for autocompletion to avoid typos in file paths.
- ❏ Search for the File: Use the find command to search for the file by name across directories.

Real-World System Issues and Troubleshooting (Connectivity)

Symptoms: Can’t ping gateway, DNS failures.

- ❏ Check interfaces: **ip addr** (missing IP? Run dhclient eth0).
- ❏ Validate DNS: **dig example.com** (check /etc/resolv.conf).
- ❏ Inspect routes: **ip route** (default gateway missing?).
- ❏ Test firewall: **sudo ufw status** (temporarily disable: ufw disable).
- ❏ Review network configuration: verify you network configuration files /etc/network/intefaces, /etc/netplan/
- ❏ Flush DNS cache: Clear the DNS cache with systemd-resolve --flush-caches, which can resolve some conflicts.
- ❏ Restart network services: systemctl restart network, systemctl restart networking, systemctl restart NetworkManager

Prevention:

- ❏ Use static IP for servers
- ❏ Document network configs
- ❏ Script network config backups.
- ❏ Use configuration management (e.g., Ansible).
- ❏ Enable NetworkManager for dynamic control.

Real-World System Issues and Troubleshooting (Package Issues)

Symptoms: apt/yum errors during installs/updates.

- ❏ Refresh package list: ensure that you have latest package list(sudo apt update)
- ❏ Fix broken packages:

  Debian: **apt --fix-broken install.**

  RHEL: **yum-complete-transaction**.

  Clean cache: **apt clean or yum clean all**.
- ❏ Use **dpkg --configure -a** (Debian) if interrupted.

Prevention:

- ❏ Test updates in a sandbox.
- ❏ Use version-locked repos (e.g., yum versionlock).
- ❏ Prefer LTS releases for stability.

Real-World System Issues and Troubleshooting (Service Issues)

Symptoms: Service crashes, fails to start.

- ❏ Check status: **systemctl status <service>**.
- ❏ View logs: **journalctl -u <service> -e --since "5 min ago"**.
- ❏ Test configs: **sshd -t (for SSH), nginx -t**.
- ❏ Check ports: **ss -tulpn | grep :<port>**.

Prevention:

- ❏ Validate configs after changes
- ❏ Use systemd-analyze to review boot issues
- ❏ Use systemd restart limits (StartLimitInterval).
- ❏ Isolate services in containers (Docker/LXC).
- ❏ Monitor resource usage (CPU, RAM, file handles).

Real-World System Issues and Troubleshooting (Remote Access)

- ❑ Check firewall configs
- ❑ Check ssh service configs /etc/ssh/sshd_config
- ❑ Check fail2ban jail
- ❑ Use console access if locked out

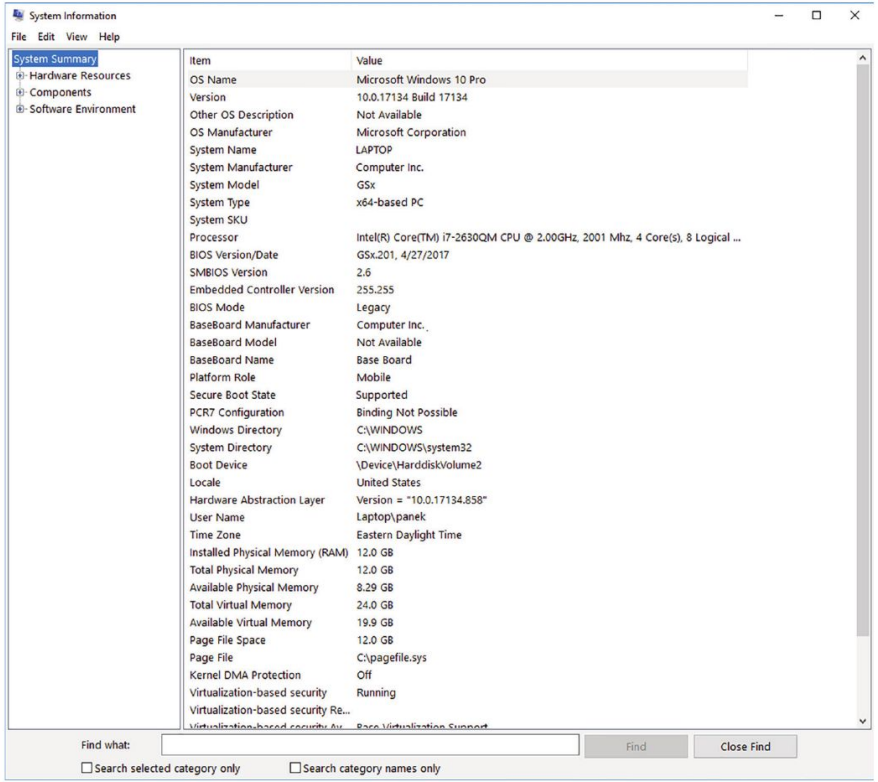
Prevention:

- ❑ Setup Key based login
- ❑ Use AllowUsers and fail2ban protection

Windows Server Troubleshooting

System Information

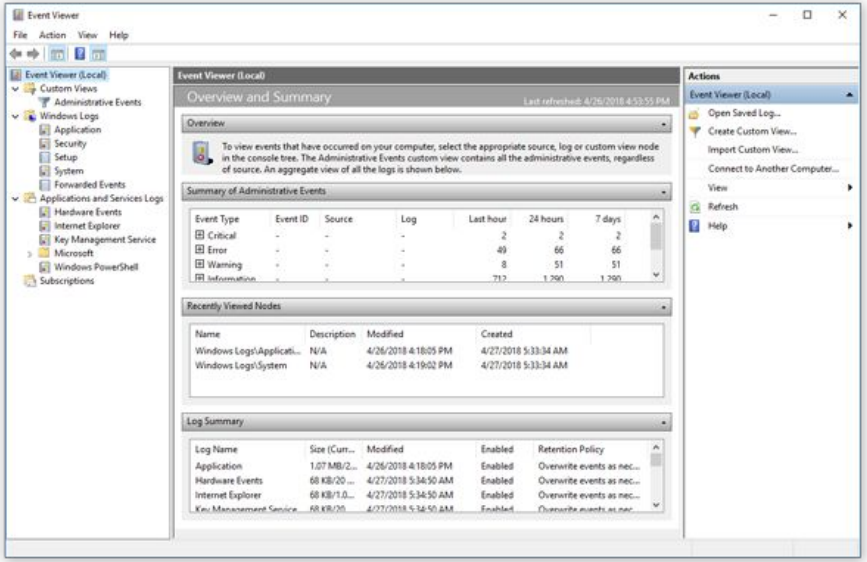
- When you first start troubleshooting a server, you need to know what is in the server such as the type and number of processors and the amount of RAM. You will also need to know what programs and services are running.
- System Information shows details about your computer's hardware configuration, computer components, and software, including drivers.



Windows Server Troubleshooting

Event Viewer

- One of the most useful troubleshooting tools is the Event Viewer MMC snap-in, which essentially is a log viewer.
- Any time you have problems, you should look in the Event Viewer to see any errors or warning, which may reveal what a problem is.

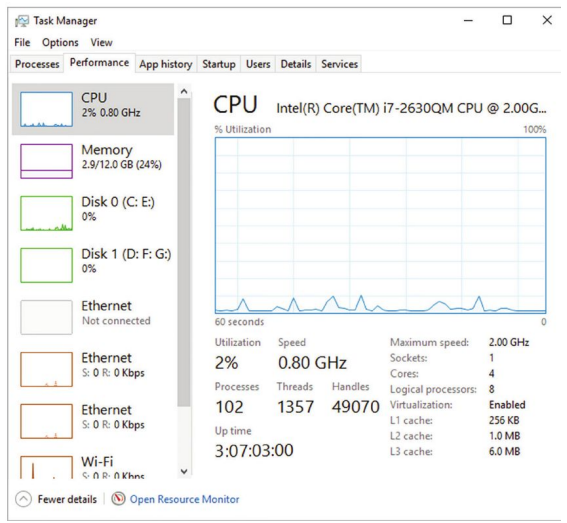


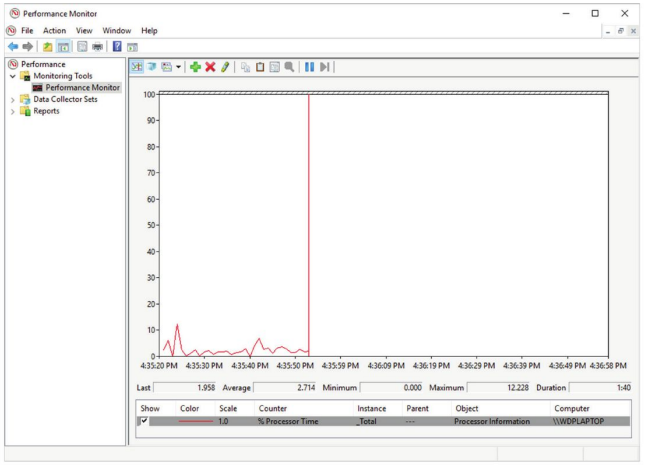
Windows Server Troubleshooting

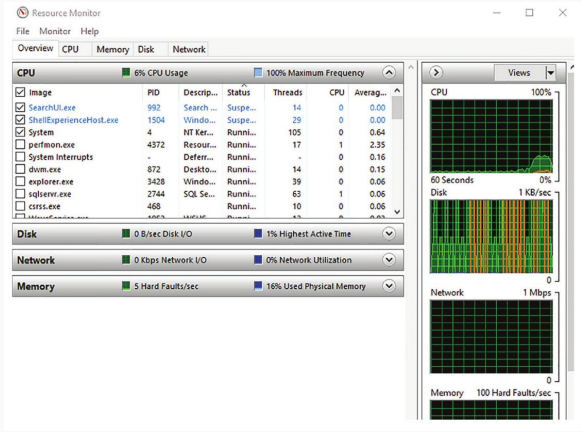
Performance

There are several tools available with Windows for you to analyze performance. They include:

- Task Manager
- Performance Monitor
- Resource Monitor







Knowledge Check

Which of the following is the first step in the standard troubleshooting methodology?

A. Implement a solution

B. Identify the problem

C. Verify full system functionality

D. Document the issue

What is the purpose of establishing a theory of probable cause during troubleshooting?

A. To confirm the issue is fixed

B. To document the solution

C. To narrow down potential root causes

D. To escalate the issue

Which of the following tools can you use on Linux to view real-time CPU usage per process?

A. df

B. ping

C. htop

D. du

A Linux

Knowledge Check

system is experiencing slow performance. Which command would help assess memory usage?

A. du

B. df -h

C. free -m

D. ls

Which tool would help you analyze disk usage directory-wise on a Linux system?

A. df

B. du

C. lsof

D. top

Which command shows all open files by processes on a Linux system?

A. top

B. lsof

C. uptime

D. sar

Knowledge Check

A user is unable to SSH into a Linux server. Which tool would best help verify if the port is reachable?

A. free

B. telnet

C. df

D. du

A Windows server shows slow response times. Which built-in tool helps monitor real-time resource usage?

A. Device Manager

B. Event Viewer

C. Performance Monitor

D. Services Console

Which Linux command provides historical system performance statistics?

A. uptime

B. htop

C. sar

D. ping

Knowledge Check

In Linux, which command lets you check which users are currently logged in?

A. who

B. ps

C. top

D. uptime

What Linux tool would you use to check system logs managed by journald?

A. logger

B. journalctl

C. sysstat

D. sar

A file is missing permission for a user. What Linux command helps adjust this?

A. chmod

B. ping

C. uptime

D. du

Knowledge Check

Which tool in Windows is best for viewing system error logs?

A. Task Manager

B. Disk Management

C. Event Viewer

D. Device Manager

If a service fails to start in Linux, which command gives the most helpful output?

A. du

B. free

C. systemctl status <service>

D. df

Which command helps determine if a file system is nearly full in Linux?

A. ls

B. du

C. df -h

D. uptime

Exercise 1

Scenario:
A production website hosted on an Apache web server has suddenly become inaccessible. You are provided with the server's credentials and IP address.
Task:
Investigate the root cause of the issue and restore website availability.

Exercise 2

Scenario:
A developer has deployed a test website using the NGINX web server, but cannot access the page from a browser. You have been given the server's login credentials and IP address.
Task:
Diagnose the problem and resolve it. Afterward, provide the developer with the correct method to access the test site.

Exercise 3

Scenario:
An IT administrator is unable to connect to the company's internal DNS server while trying to modify its configuration files. You are granted direct console access to the DNS server.

Task:
Identify and fix the connectivity issue preventing access to the DNS server.

Exercise 4

Scenario:
You were in the process of setting a static IP address and DNS server on a Linux system. During the configuration, you lost connectivity to the server. Console access is now available.

Task:
Troubleshoot the networking configuration and restore proper connectivity to the server.

Exercise 5

Scenario:
A client application is unable to connect to a remote MySQL database server. You are provided with the database server's IP and SSH credentials.

Task:
Investigate the database connection failure and take necessary actions to restore access.

Exercise 6

Scenario:
A server is performing very poorly—it is consistently slow and experiences frequent disconnections, despite running no active services.

Task:
Assess the health of the server resolve any issues.

Exercise 7

Scenario:
You are experiencing very slow file transfers from a server. Even small files take a long time to move across the network.
Task:
Identify and resolve the root cause of the network performance degradation.

Exercise 8

Scenario:
Investigate what the server http://zergaw.com:9876 is relying at a network level using packet capture
Task:
Use tcpdump to capture client request to http://zergaw.com:9876 and analyze it in wireshark

Thank You

- +2519770355111
- info@citcot.com
- citcot.com
  nunaethiopia.com