1. ping: Checking Network Connectivity

- **Scenario**: Your web server can't reach a database server.
- **Example**:

```
Sh: ping database.example.com
```

- **Output**:

```
PING database.example.com (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=64 time=0.045 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=64 time=0.047 ms
```

- **Explanation**: If you receive replies, the network connection is working. If not, there might be a network issue.

2. traceroute: Diagnosing Network Path Issues

- **Scenario**: You are experiencing slow connection to a remote server.
- **Example**:

```
Sh: traceroute google.com
```

- **Output**:

```
traceroute to google.com (172.217.4.110), 30 hops max, 60 byte
packets
192.168.0.1 (192.168.0.1)  1.098 ms  1.004 ms  0.933 ms
* * *
10.0.0.1 (10.0.0.1)  10.533 ms  10.489 ms  10.441 ms
...
```

- **Explanation**: You can see the path packets take and identify where delays or failures occur.

3. netstat: Checking Network Connections

- **Scenario**: Your web server is slow, and you suspect too many connections.
- **Example**:

```
Sh: netstat -tuln
```

- **Output**:

```
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address
State
tcp        0      0 0.0.0.0:80              0.0.0.0:*
LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*
LISTEN
```

- **Explanation**: Shows all listening ports. You can see if there are too many connections or unexpected services running.

4. ifconfig / ip: Checking Network Interface Configuration

- **Scenario**: You need to verify the IP address of your network interface.
- **Example**:

  ```
  Sh: ifconfig
  ```

- **Output**:

  ```
  eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
  inet 192.168.0.101  netmask 255.255.255.0  broadcast 192.168.0.255
  ```

- **Explanation**: Shows the network configuration of your interfaces.

5. iwconfig: Checking Wireless Network Status

- **Scenario**: Troubleshooting a slow Wi-Fi connection.
- **Example**:

  ```
  Sh: iwconfig
  ```

- **Output**:

  ```
  wlan0     IEEE 802.11  ESSID:"MyNetwork"
  Mode:Managed  Frequency:2.437 GHz  Access Point: 00:14:22:01:23:45
  Bit Rate=54 Mb/s   Tx-Power=20 dBm
  Link Quality=70/70  Signal level=-40 dBm
  ```

- **Explanation**: You can check the signal quality and strength to diagnose connectivity issues.

6. ps: Checking Running Processes

- **Scenario**: Your server is slow, and you suspect a runaway process.
- **Example**:

  ```
  Sh: ps aux | grep apache2
  ```

- **Output**:

  ```
  www-data  1234  0.1  2.3 235124 23900 ? S  12:00 0:00
  /usr/sbin/apache2 -k start
  ```

- **Explanation**: Shows all running processes related to `apache2`. You can see the CPU and memory usage.

7. top: Real-Time System Monitoring

- **Scenario**: Monitoring server performance in real-time.
- **Example**:

```
Sh: top
```

- **Output**:

```
top - 12:05:01 up 1 day,  2:34,  2 users,  load average: 0.25, 0.15, 0.10
Tasks: 112 total,   1 running, 111 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.0 us,  0.2 sy,  0.0 ni, 98.8 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   2048.0 total,    512.0 free,   1024.0 used,    512.0 buff/cache
```

- **Explanation**: Provides a dynamic view of system performance, including CPU and memory usage.

8. htop: Enhanced Real-Time System Monitoring

- **Scenario**: You need a more user-friendly interface for real-time monitoring.
- **Example**:

```
Sh: htop
```

- **Explanation**: Similar to `top` but with a more interactive and visually appealing interface.

9. df: Checking Disk Space Usage

- **Scenario**: Your application is failing due to insufficient disk space.
- **Example**:

```
Sh: df -h
```

- **Output**:

```
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       50G   25G   23G  53% /
```

- **Explanation**: Shows available and used disk space on all mounted filesystems.

10. du: Checking Directory Disk Usage

- **Scenario**: Finding out which directories are using the most space.
- **Example**:

```
Sh: du -sh /var/log/*
```

- **Output**:

```
10M     /var/log/apache2
100M    /var/log/mysql
```

- **Explanation**: Shows disk usage for each directory, helping you find large directories.

11. free: Checking Memory Usage

- **Scenario**: Your system is running out of memory.
- **Example**:

```
Sh: free -h
```

- **Output**:

```
total        used        free       shared  buff/cache   available
Mem:           2.0G        1.0G        512M        128M        512M
1.0G
Swap:          1.0G        256M        768M
```

- **Explanation**: Displays the amount of free and used memory in the system.

12. journalctl: Viewing System Logs

- **Scenario**: Investigating a systemd service that failed to start.
- **Example**:

```
Sh: journalctl -u apache2.service
```

- **Output**:

```
Jun 10 12:00:00 hostname systemd[1]: Starting The Apache HTTP
Server...
Jun 10 12:00:00 hostname apache2[1234]: AH00558: apache2: Could not
reliably determine the server's fully qualified domain name, using
127.0.0.1. Set the 'ServerName' directive globally to suppress this
message
Jun 10 12:00:00 hostname systemd[1]: Started The Apache HTTP Server.
```

- **Explanation**: Shows logs specific to the `apache2` service, helping diagnose why it failed.

13. dmesg: Viewing Kernel Messages

- **Scenario**: Checking for hardware-related errors.
- **Example**:

  ```
  Sh: dmesg | grep error
  ```

- **Output**:

  ```
  [0.123456] ACPI Error: Method parse/execution failed
  ```

- **Explanation**: Displays messages from the kernel ring buffer, useful for diagnosing hardware and driver issues.

14. ls: Listing Files and Directories

- **Scenario**: Checking the contents of a directory.
- **Example**:

  ```
  Sh: ls -l /var/www/html
  ```

- **Output**:

  ```
  total 4
  -rw-r--r-- 1 root root 4096 Jun 10 12:00 index.html
  ```

- **Explanation**: Lists files and directories with detailed information such as permissions, owner, and size.

15. grep: Searching Through Files

- **Scenario**: Searching log files for a specific error message.
- **Example**:

  ```
  Sh: grep "error" /var/log/syslog
  ```

- **Output**:

  ```
  Jun 10 12:00:00 hostname kernel: [123456.789] error: Disk read failed
  ```

- **Explanation**: Finds lines containing "error" in the specified log file, useful for quickly identifying issues.

These examples demonstrate how these commands are used in real-world scenarios to troubleshoot and diagnose problems in a Linux environment.