Linux Fundamentals Report

By Aditya Singh Rathaur

Introduction

Linux is a family of open-source, Unix-like operating systems based on the Linux kernel. Linux was made freely available for anyone to use, modify, and distribute. The community-driven development model has enabled Linux to evolve rapidly and gain widespread adoption across personal computers, servers, embedded systems, smartphones (via Android), and even supercomputers. The notable features of Linux are security, versatility stability and reliability.

Types of Linux Distribution:

- **Ubuntu**: User-friendly, popular for desktops and servers, with strong community support.
- **Kali Linux**: Specialized for security testing and ethical hacking.
- **Fedora**: Cutting-edge features, focuses on innovation and latest software.
- **Arch Linux**: Minimalist, highly customizable, rolling release model for experienced users.
- **CentOS / Rocky Linux / Alma Linux**: Enterprise-focused, Red Hat compatible, known for long-term stability in servers.

Linux is important in the computing world because:

- **Server Dominance**: Linux powers the majority of the world's web servers, cloud infrastructure, supercomputers, and many enterprise systems due to its reliability, performance, and cost-effectiveness.
- **Developer and Education Use**: Valued by developers and educational institutions for its openness, flexibility, and strong developer tools.
- **Security and Privacy**: The transparent, open-source nature enables users to verify code integrity, avoid vendor lock-in, and customize security to their requirements.
- **Innovation and Community**: Linux's collaborative environment accelerates the development of new technologies, encourages experimentation, and often serves as the foundation for technological innovations.

Architecture

Linux has a **layered architecture** comprising the following primary components:

- **Hardware:** The physical components—CPU, memory, storage, network interfaces—form the foundation.
- **Kernel:** The core of the system, responsible for managing hardware resources (CPU, memory, I/O devices), process and memory management, device drivers, file systems, and network operations. The Linux kernel adopts a monolithic design, meaning most OS services run within the kernel space, providing efficiency and direct hardware access.
- **Shell:** The interface between users and the kernel, allowing users to execute commands. Common shells are command-line (Bash, Zsh) and graphical shells (GNOME, KDE).
- **Utilities:** System tools and applications for system management (file operations, monitoring, installing software, networking).

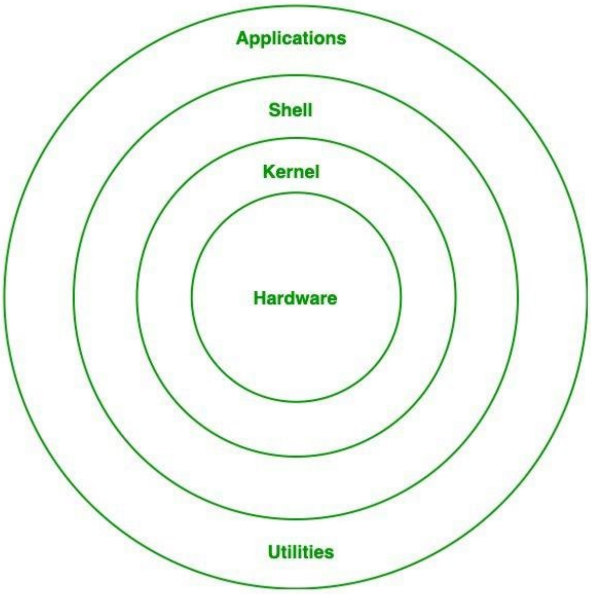
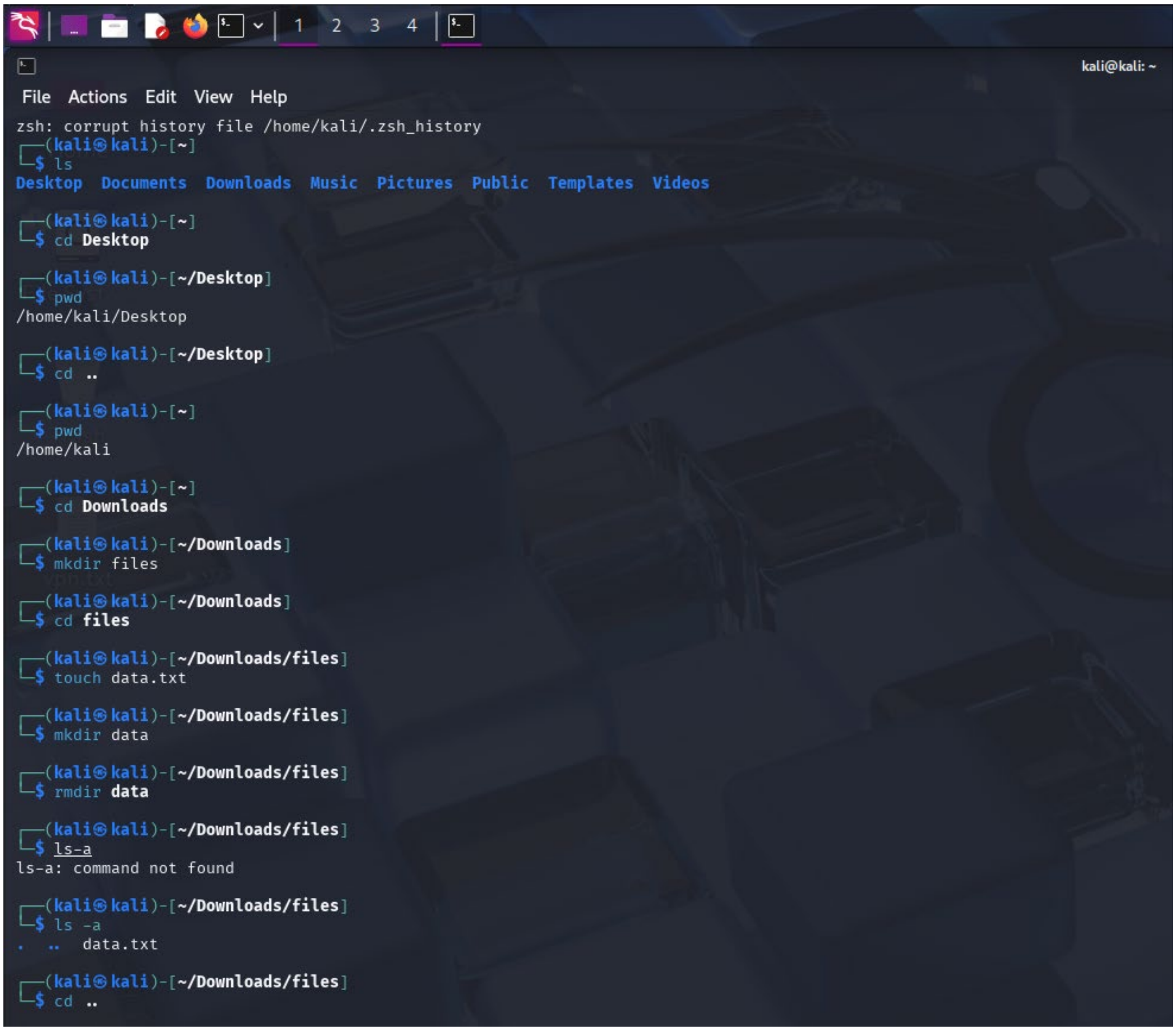


Figure 1: Architecture of Linux

Basic Linux Commands

- **ls:** List files and directories in the current directory
    - **ls -l:** Lists in long format showing permissions, ownership, size, etc.
    - **ls -a:** Shows all files including hidden files
- **cd:** change directory (Example: cd /home/user/Downloads)
    - **cd ..:** Move back one directory
    - **cd ~:** Move to the home directory
- **pwd:** Print current working directory
- **find:** Search a file (Example: find -name “10.22.10.255”)

- **mkdir:** Create a new directory (Example: mkdir new_folder)
- **rmdir:** Remove an empty directory (Example: rmdir new_folder)
- **cp:** Copies file or directories (Example: cp file1.txt file2.txt)
- **mv:** Moves or renames files/directories (Example mv old.txt new.txt)
- **rm:** Removes files or directories (Example: rm file.txt)
- **touch:** Creates an empty file (Example: touch file.txt)
- **nano/vim:** Edit files in the terminal
- **env:** Display environment variables
- **sudo:** Execute commands as host
- **cat:** Concatenates and displays file content (Example: cat file.txt)
- **Shell Operators:**
  - **&:** Run commands in background
  - **&&:** Used to run a list of commands but the 2nd command runs only after the first command
  - **>:** Redirect the output from a command and direct it elsewhere
  - **>>:** Same as (>) but it appends rather than replace



```
zsh: corrupt history file /home/kali/.zsh_history
┌──(kali㉿kali)-[~]
└─$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos

┌──(kali㉿kali)-[~]
└─$ cd Desktop

┌──(kali㉿kali)-[~/Desktop]
└─$ pwd
/home/kali/Desktop

┌──(kali㉿kali)-[~/Desktop]
└─$ cd ..

┌──(kali㉿kali)-[~]
└─$ pwd
/home/kali

┌──(kali㉿kali)-[~]
└─$ cd Downloads

┌──(kali㉿kali)-[~/Downloads]
└─$ mkdir files

┌──(kali㉿kali)-[~/Downloads]
└─$ cd files

┌──(kali㉿kali)-[~/Downloads/files]
└─$ touch data.txt

┌──(kali㉿kali)-[~/Downloads/files]
└─$ mkdir data

┌──(kali㉿kali)-[~/Downloads/files]
└─$ rmdir data

┌──(kali㉿kali)-[~/Downloads/files]
└─$ ls-a
ls-a: command not found

┌──(kali㉿kali)-[~/Downloads/files]
└─$ ls -a
.  ..  data.txt

┌──(kali㉿kali)-[~/Downloads/files]
└─$ cd ..
```

Figure 2: Basic Linux Commands

Permissions Format:

A permission string contains 10 characters. The first character indicates the file type such as regular file(-), directory(d), symbolic link(l) etc. and the next 9 characters are grouped into three sets of three as given below:

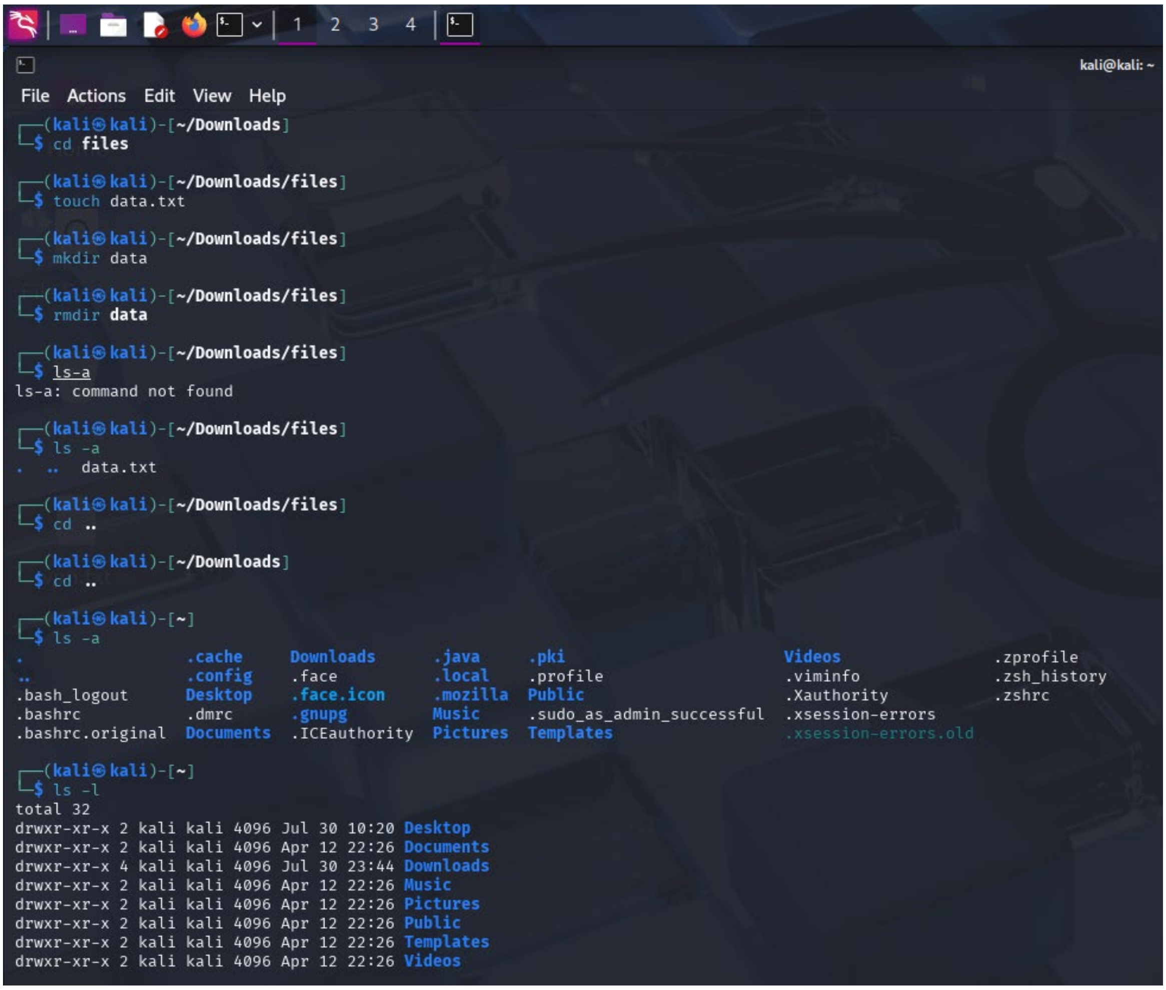
Position	Permission Group	Meaning
2-4	User (Owner)	Owner permission
5-7	Group	Group permission
8-10	Others	Permission for others

Each group has three characters representing:

- **r:** read permission
- **w:** write permission
- **x:** execute permission
- **-:** no permission for that type

Commands to modify permission:

- **chmod**: Changes file or directory permissions.
- **chown**: Changes the owner of a file or directory.
- **chgrp**: Changes the group ownership.



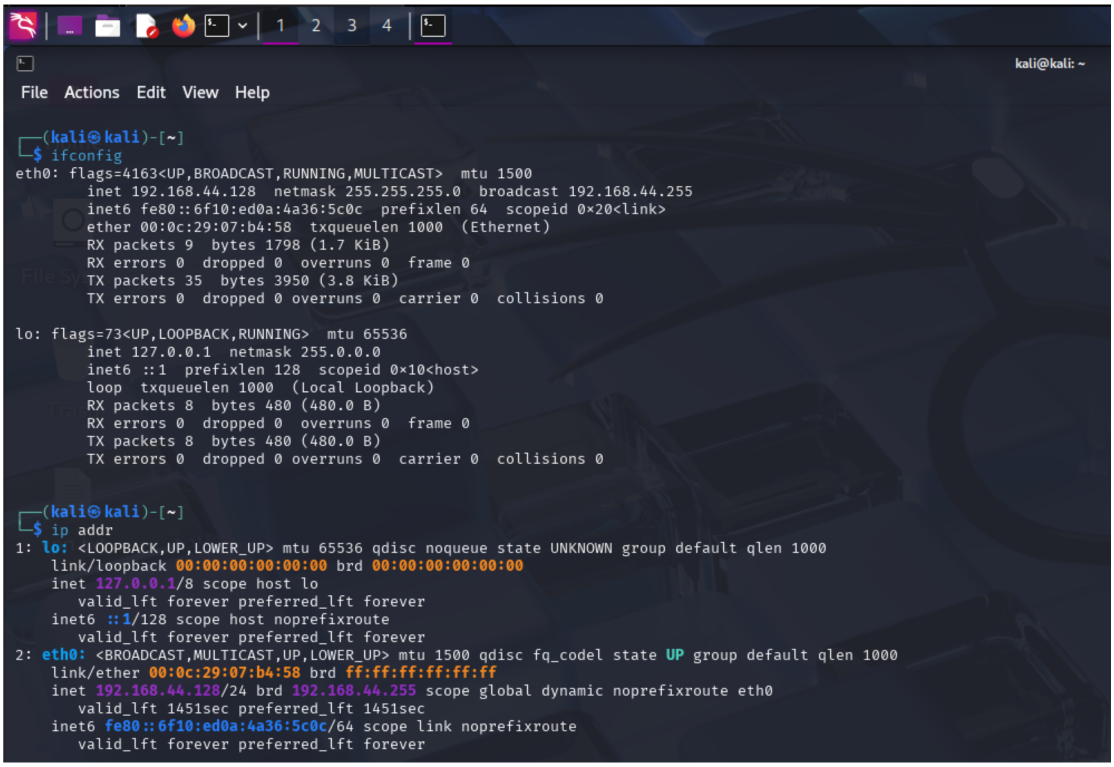
```
┌──(kali㉿kali)-[~/Downloads]
└─$ cd files

┌──(kali㉿kali)-[~/Downloads/files]
└─$ touch data.txt

┌──(kali㉿kali)-[~/Downloads/files]
└─$ mkdir data

┌──(kali㉿kali)-[~/Downloads/files]
└─$ rmdir data

┌──(kali㉿kali)-[~/Downloads/files]
└─$ ls-a
ls-a: command not found

┌──(kali㉿kali)-[~/Downloads/files]
└─$ ls -a
.  ..  data.txt

┌──(kali㉿kali)-[~/Downloads/files]
└─$ cd ..

┌──(kali㉿kali)-[~/Downloads]
└─$ cd ..

┌──(kali㉿kali)-[~]
└─$ ls -a
.                 .cache     Downloads      .java     .pki                       Videos                .zprofile
..                .config    .face          .local    .profile                   .viminfo              .zsh_history
.bash_logout      Desktop    .face.icon     .mozilla  Public                     .Xauthority           .zshrc
.bashrc           .dmrc      .gnupg         Music     .sudo_as_admin_successful  .xsession-errors
.bashrc.original  Documents  .ICEauthority  Pictures  Templates                  .xsession-errors.old

┌──(kali㉿kali)-[~]
└─$ ls -l
total 32
drwxr-xr-x 2 kali kali 4096 Jul 30 10:20 Desktop
drwxr-xr-x 2 kali kali 4096 Apr 12 22:26 Documents
drwxr-xr-x 4 kali kali 4096 Jul 30 23:44 Downloads
drwxr-xr-x 2 kali kali 4096 Apr 12 22:26 Music
drwxr-xr-x 2 kali kali 4096 Apr 12 22:26 Pictures
drwxr-xr-x 2 kali kali 4096 Apr 12 22:26 Public
drwxr-xr-x 2 kali kali 4096 Apr 12 22:26 Templates
drwxr-xr-x 2 kali kali 4096 Apr 12 22:26 Videos
```

Figure 3: Permission Format

Basic Networking in Linux:

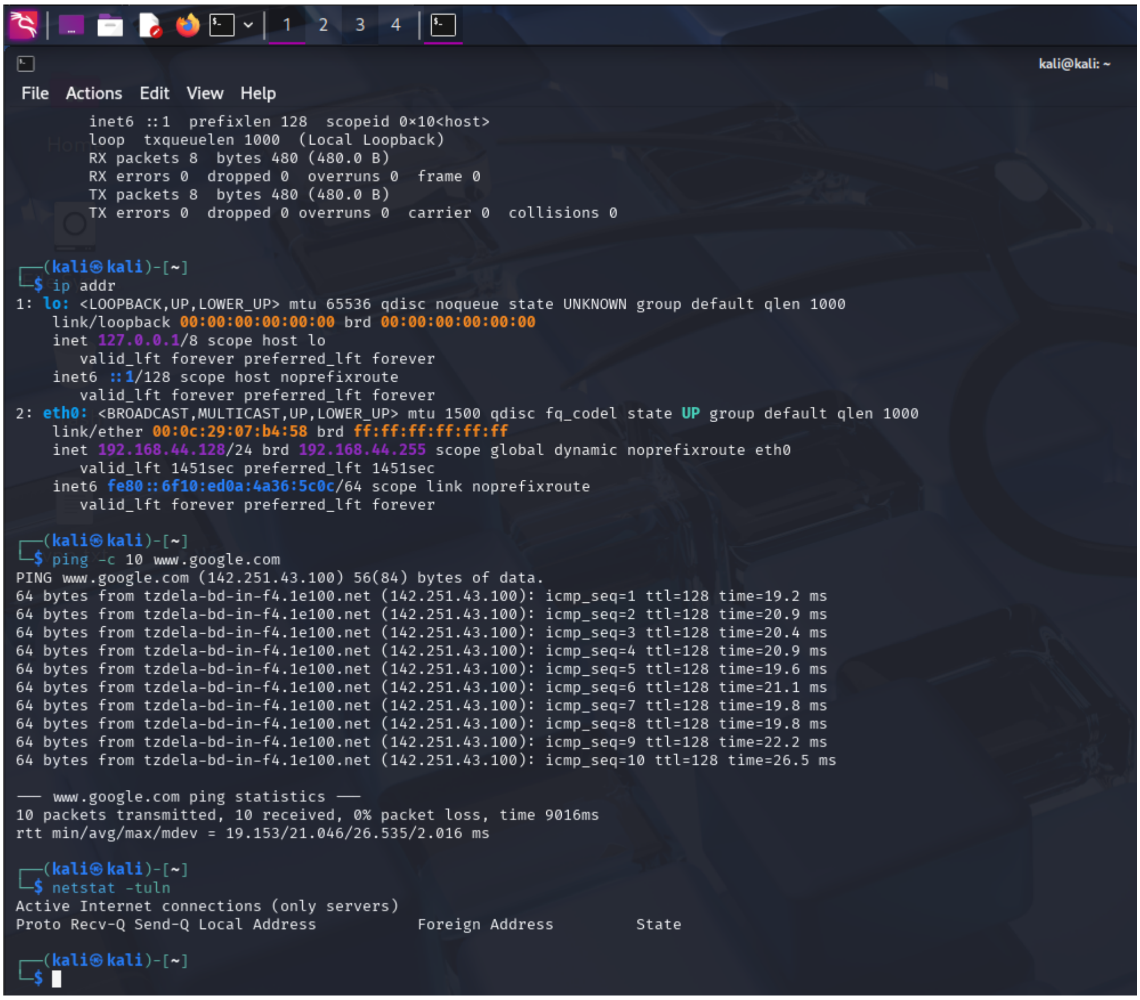
Linux provides powerful tools and commands for configuring, managing, and troubleshooting network connections. Some of them are:

- **ifconfig** (Interface Configuration)
    - Displays and configures network interface parameters
    - Assign ip address: sudo ifconfig eth0 192.168.1.10 netmask 255.255.255.0
    - **ifconfig** is now considered old and relaced by **ip** in many modern distributions
- **ip** (Recommended replacement for ifconfig)
    - Offers comprehensive modern interface and routing management
    - Lists interfaces: ip addr
    - Bring interface up/down: sudo ip link set eth0 up or sudo ip link set eth0 down
    - Assign IP: sudo ip addr add 192.168.1.10/24 dev eth0
    - Shows routing table: ip route show
- **ping**
    - It tests if a host is reachable and measures response by sending ICMP echo packets.
    - Syntax: ping <hostname/IP>
    - Example of sending 10 packets: ping -c 10 www.google.com
- **netstat** (Network Statistics)
    - Displays network connections, interface statistics etc.
    - List connections: netstat -tuln



```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.44.128  netmask 255.255.255.0  broadcast 192.168.44.255
        inet6 fe80::6f10:ed0a:4a36:5c0c  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:07:b4:58  txqueuelen 1000  (Ethernet)
        RX packets 9  bytes 1798 (1.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 35  bytes 3950 (3.8 KiB)
        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 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 480 (480.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 480 (480.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0


┌──(kali㉿kali)-[~]
└─$ 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: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:07:b4:58 brd ff:ff:ff:ff:ff:ff
    inet 192.168.44.128/24 brd 192.168.44.255 scope global dynamic noprefixroute eth0
       valid_lft 1451sec preferred_lft 1451sec
    inet6 fe80::6f10:ed0a:4a36:5c0c/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Figure 4: ifconfig and ip command



```
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 480 (480.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 480 (480.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0


┌──(kali㉿kali)-[~]
└─$ 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: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:07:b4:58 brd ff:ff:ff:ff:ff:ff
    inet 192.168.44.128/24 brd 192.168.44.255 scope global dynamic noprefixroute eth0
       valid_lft 1451sec preferred_lft 1451sec
    inet6 fe80::6f10:ed0a:4a36:5c0c/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──(kali㉿kali)-[~]
└─$ ping -c 10 www.google.com
PING www.google.com (142.251.43.100) 56(84) bytes of data.
64 bytes from tzdela-bd-in-f4.1e100.net (142.251.43.100): icmp_seq=1 ttl=128 time=19.2 ms
64 bytes from tzdela-bd-in-f4.1e100.net (142.251.43.100): icmp_seq=2 ttl=128 time=20.9 ms
64 bytes from tzdela-bd-in-f4.1e100.net (142.251.43.100): icmp_seq=3 ttl=128 time=20.4 ms
64 bytes from tzdela-bd-in-f4.1e100.net (142.251.43.100): icmp_seq=4 ttl=128 time=20.9 ms
64 bytes from tzdela-bd-in-f4.1e100.net (142.251.43.100): icmp_seq=5 ttl=128 time=19.6 ms
64 bytes from tzdela-bd-in-f4.1e100.net (142.251.43.100): icmp_seq=6 ttl=128 time=21.1 ms
64 bytes from tzdela-bd-in-f4.1e100.net (142.251.43.100): icmp_seq=7 ttl=128 time=19.8 ms
64 bytes from tzdela-bd-in-f4.1e100.net (142.251.43.100): icmp_seq=8 ttl=128 time=19.8 ms
64 bytes from tzdela-bd-in-f4.1e100.net (142.251.43.100): icmp_seq=9 ttl=128 time=22.2 ms
64 bytes from tzdela-bd-in-f4.1e100.net (142.251.43.100): icmp_seq=10 ttl=128 time=26.5 ms

── www.google.com ping statistics ──
10 packets transmitted, 10 received, 0% packet loss, time 9016ms
rtt min/avg/max/mdev = 19.153/21.046/26.535/2.016 ms

┌──(kali㉿kali)-[~]
└─$ netstat -tuln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State

┌──(kali㉿kali)-[~]
└─$ 
```

Figure 5: ping and netstat command