scp (OpenSSH secure file copy) : scp copies files between hosts on a network. It uses ssh(1) for data transfer, and uses the same authentication and provides the same security as ssh(1). scp will ask for passwords or passphrases if they are needed for authentication.

The **source** and **target** may be specified as a local pathname, a remote host with optional path in the form [user@]host:[path], or a URI in the form scp://[user@]host[:port][/path]. Local file names can be made explicit using absolute or relative pathnames to avoid scp treating file names containing ':' as host specifiers.

When copying between two remote hosts, if the URI format is used, a port may only be specified on the target if the -3 option is used.

Synatx :

$ scp [-346BCpqrTv] [-c cipher] [-F ssh_config] [-i identity_file] [-J destination] [-l limit] [-o ssh_option] [-P port] [-S program] source ... target

Options :

- -3 : Copies between two remote hosts are transferred through the local host. Without this option the data is copied directly between the two remote hosts. Note that this option disables the progress meter.
- -4 : Forces scp to use IPv4 addresses only.
- -6 : Forces scp to use IPv6 addresses only.
- -B : Selects batch mode (prevents asking for passwords or passphrases).
- -C : Compression enable. Passes the -C flag to ssh(1) to enable compression.
- -c cipher : Selects the cipher to use for encrypting the data transfer. This option is directly passed to ssh(1).
- -F ssh_config : Specifies an alternative per-user configuration file for ssh. This option is directly passed to ssh(1).
- -i identity_file : Selects the file from which the identity (private key) for public key authentication is read. This option is directly passed to ssh(1).
- -J destination : Connect to the target host by first making an scp connection to the jump host described by destination and then establishing a TCP forwarding to the ultimate destination from there. Multiple jump hops may be specified separated by comma characters. This is a shortcut to specify a ProxyJump configuration directive. This option is directly passed to ssh(1).
- -l limit : Limits the used bandwidth, specified in Kbit/s.
- -o ssh_option : Can be used to pass options to ssh in the format used in ssh_config(5). This is useful for specifying options for which there is no separate scp command-line flag.
- -P port : Specifies the port to connect to on the remote host. Note that this option is written with a capital 'P', because -p is already reserved for preserving the times and modes of the file.

- -p : Preserves modification times, access times, and modes from the original file.
- -q : Quiet mode: disables the progress meter as well as warning and diagnostic messages from ssh(1).
- -r : Recursively copy entire directories. Note that scp follows symbolic links encountered in the tree traversal.
- -S program : Name of program to use for the encrypted connection. The program must understand ssh(1) options.
- -T : Disable strict filename checking. By default when copying files from a remote host to a local directory scp checks that the received filenames match those requested on the command-line to prevent the remote end from sending unexpected or unwanted files. Because of differences in how various operating systems and shells interpret filename wildcards, these checks may cause wanted files to be rejected. This option disables these checks at the expense of fully trusting that the server will not send unexpected filenames.
- -v : Verbose mode. Causes scp and ssh(1) to print debugging messages about their progress. This is helpful in debugging connection, authentication, and configuration problems.

Examples :

-p : Preserves modification times, access times, and modes from the original file.An estimated time and the connection speed will appear on the screen.

```
pungki@mint ~/Documents $ scp -p Label.pdf mrarianto@202.x.x.x:.
```

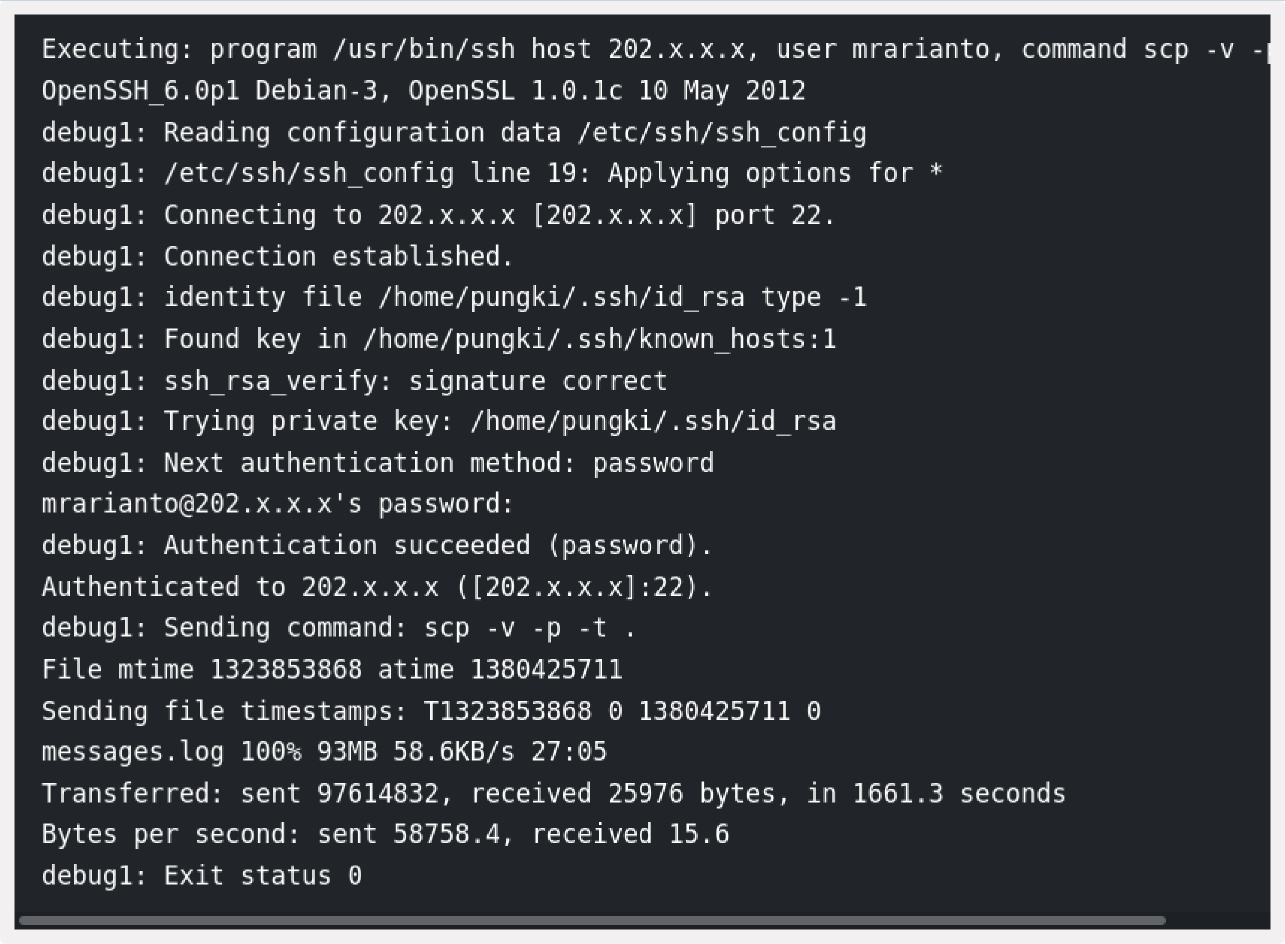
Output :

```
mrarianto@202.x.x.x's password:
Label.pdf 100% 3672KB 126.6KB/s 00:29
```

One of the parameters that can faster your file transfer is the **“-C”** parameter. The **“-C”** parameter will compress your files on the go. The unique thing is the compression-only happens in the network. When the file has arrived at the destination server, it will be returning to the original size as before the compression happen.

```
pungki@mint ~/Documents $ scp -Cpv messages.log mrarianto@202.x.x.x:.
```

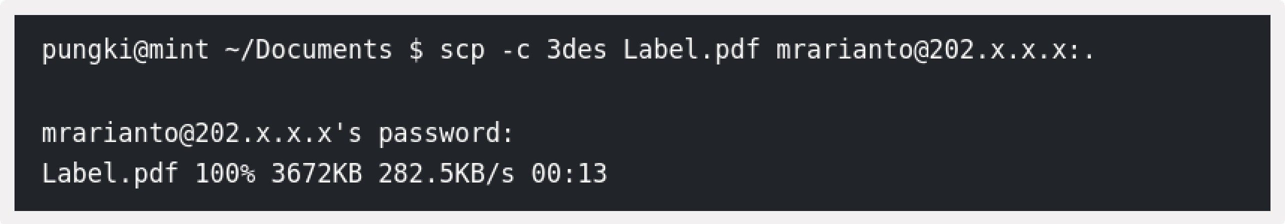
Output :



```
Executing: program /usr/bin/ssh host 202.x.x.x, user mrarianto, command scp -v -
OpenSSH_6.0p1 Debian-3, OpenSSL 1.0.1c 10 May 2012
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: /etc/ssh/ssh_config line 19: Applying options for *
debug1: Connecting to 202.x.x.x [202.x.x.x] port 22.
debug1: Connection established.
debug1: identity file /home/pungki/.ssh/id_rsa type -1
debug1: Found key in /home/pungki/.ssh/known_hosts:1
debug1: ssh_rsa_verify: signature correct
debug1: Trying private key: /home/pungki/.ssh/id_rsa
debug1: Next authentication method: password
mrarianto@202.x.x.x's password:
debug1: Authentication succeeded (password).
Authenticated to 202.x.x.x ([202.x.x.x]:22).
debug1: Sending command: scp -v -p -t .
File mtime 1323853868 atime 1380425711
Sending file timestamps: T1323853868 0 1380425711 0
messages.log 100% 93MB 58.6KB/s 27:05
Transferred: sent 97614832, received 25976 bytes, in 1661.3 seconds
Bytes per second: sent 58758.4, received 15.6
debug1: Exit status 0
```

Without -C it would take more time to transfer the file over network.

By default **SCP** using “**AES-128**” to encrypt files. If you want to change to another cipher to encrypt it, you can use the “**-c**” parameter. Take a look at this command.



```
pungki@mint ~/Documents $ scp -c 3des Label.pdf mrarianto@202.x.x.x:.

mrarianto@202.x.x.x's password:
Label.pdf 100% 3672KB 282.5KB/s 00:13
```

The above command tells SCP to **use the 3des algorithm** to encrypt the file. Please be careful that this parameter using “-c” not “-C”.

sftp (Openssh secure file transfer) : sftp is a file transfer program, similar to ftp(1), which performs all operations over an encrypted ssh(1) transport. It may also use many features of ssh, such as public key authentication and compression.

The destination may be specified either as [user@]host[:path] or as a URI in the form sftp://[user@]host[:port][/path].

If the destination includes a path and it is not a directory, sftp will retrieve files automatically if a non-interactive authentication method is used; otherwise it will do so after successful interactive authentication.

If no path is specified, or if the path is a directory, sftp will log in to the specified host and enter interactive command mode, changing to the remote directory if one was specified. An optional trailing slash can be used to force the path to be interpreted as a directory.

Since the destination formats use colon characters to delimit host names from path names or port numbers, IPv6 addresses must be enclosed in square brackets to avoid ambiguity.

Syntax :

$ sftp [-46aCfpqrv] [-B buffer_size] [-b batchfile] [-c cipher] [-D sftp_server_path] [-F ssh_config] [-i identity_file] [-J destination] [-l limit] [-o ssh_option] [-P port] [-R num_requests] [-S program] [-s subsystem | sftp_server] destination

Options :

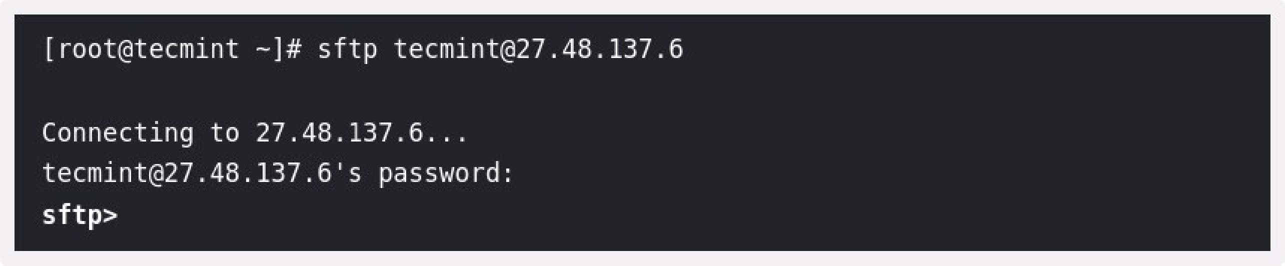
- -4 : Forces sftp to use IPv4 addresses only.
- -6 : Forces sftp to use IPv6 addresses only.
- -a : Attempt to continue interrupted transfers rather than overwriting existing partial or complete copies of files. If the partial contents differ from those being transferred, then the resultant file is likely to be corrupt.
- -B buffer_size : Specify the size of the buffer that sftp uses when transferring files. Larger buffers require fewer round trips at the cost of higher memory consumption. The default is 32768 bytes.
- -b batchfile : Batch mode reads a series of commands from an input batchfile instead of stdin. Since it lacks user interaction it should be used in conjunction with non-interactive authentication to obviate the need to enter a password at connection time (see sshd(8) and ssh-keygen(1) for details).

  A batchfile of ‘-’ may be used to indicate standard input. sftp will abort if any of the following commands fail: get, put, reget,reput,rename, ln, rm, mkdir, chdir, ls, lchdir, chmod, chown, chgrp, lpwd,df, symlink, and lmkdir.
- -C : Enables compression (via ssh's -C flag).
- -c cipher : Selects the cipher to use for encrypting the data transfers. This option is directly passed to ssh(1).
- -D sftp_server_path : Connect directly to a local sftp server (rather than via ssh(1)). This option may be useful in debugging the client and server.

- -F ssh_config : Specifies an alternative per-user configuration file for ssh(1). This option is directly passed to ssh(1).
- -f : Requests that files be flushed to disk immediately after transfer. When uploading files, this feature is only enabled if the server implements the "fsync@openssh.com" extension.
- -i identity_file : Selects the file from which the identity (private key) for public key authentication is read. This option is directly passed to ssh(1).
- -J destination : Connect to the target host by first making an sftp connection to the jump host described by destination and then establishing a TCP forwarding to the ultimate destination from there. Multiple jump hops may be specified separated by comma characters. This is a shortcut to specify a ProxyJump configuration directive. This option is directly passed to ssh(1).
- -l limit : Limits the used bandwidth, specified in Kbit/s.
- -o ssh_option : Can be used to pass options to ssh in the format used in ssh_config(5). This is useful for specifying options for which there is no separate sftp command-line flag. For example, to specify an alternate port use: sftp -oPort=24.
- -P port : Specifies the port to connect to on the remote host.
- -p : Preserves modification times, access times, and modes from the original files transferred.
- -q Quiet mode : disables the progress meter as well as warning and diagnostic messages from ssh(1).
- -R num_requests : Specify how many requests may be outstanding at any one time. Increasing this may slightly improve file transfer speed but will increase memory usage. The default is 64 outstanding requests.
- -r : Recursively copy entire directories when uploading and downloading. Note that sftp does not follow symbolic links encountered in the tree traversal.
- -S program : Name of the program to use for the encrypted connection. The program must understand ssh(1) options.
- -s : subsystem | sftp_server - Specifies the SSH2 subsystem or the path for an sftp server on the remote host. A path is useful when the remote sshd(8) does not have an sftp subsystem configured.
- -v : Raise logging level. This option is also passed to ssh.

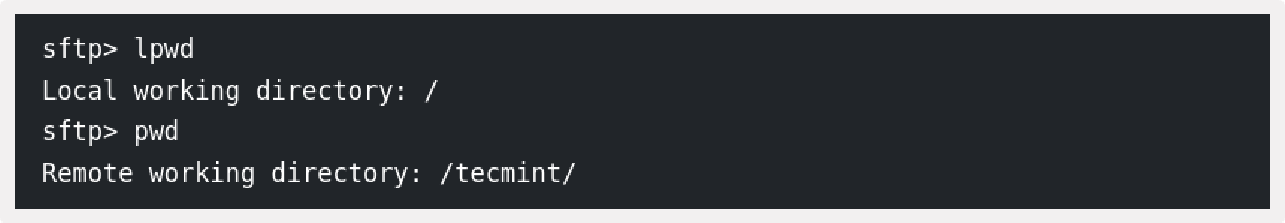
Examples :

To start an SFTP session, enter the username and remote hostname or IP address at the command prompt. Once authentication is successful, you will see a shell with an sftp> prompt.



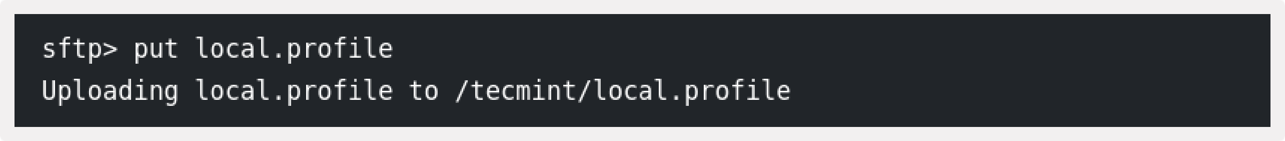
```
[root@tecmint ~]# sftp tecmint@27.48.137.6

Connecting to 27.48.137.6...
tecmint@27.48.137.6's password:
sftp>
```

The command ‘lpwd‘ is used to check the Local present working directory, whereas the pwd command is used to check the Remote working directory.



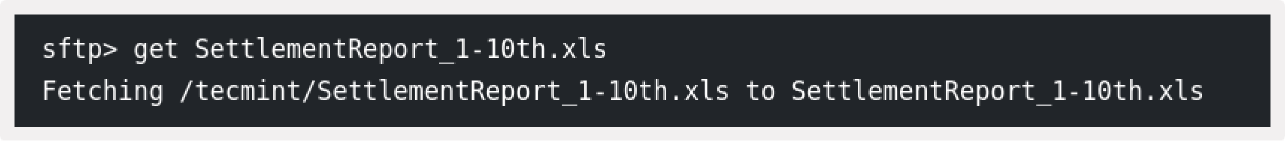
```
sftp> lpwd
Local working directory: /
sftp> pwd
Remote working directory: /tecmint/
```

Put single or multiple files in remote system ftp server.



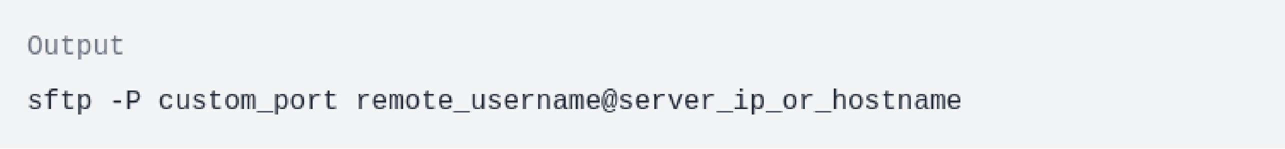
```
sftp> put local.profile
Uploading local.profile to /tecmint/local.profile
```

Getting single or multiple files in a local system.



```
sftp> get SettlementReport_1-10th.xls
Fetching /tecmint/SettlementReport_1-10th.xls to SettlementReport_1-10th.xls
```

If the remote SSH server is not listening on the default port 22, use the -P option to specify the SFTP port:



```
Output

sftp -P custom_port remote_username@server_ip_or_hostname
```

ifconfig(interface configuration) : ifconfig 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 only needed when debugging or when system tuning is needed.

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

Syntax :

```
$ ifconfig [-v] [-a] [-s] [interface]
$ ifconfig [-v] interface [aftype] options | address ...
```

Options :

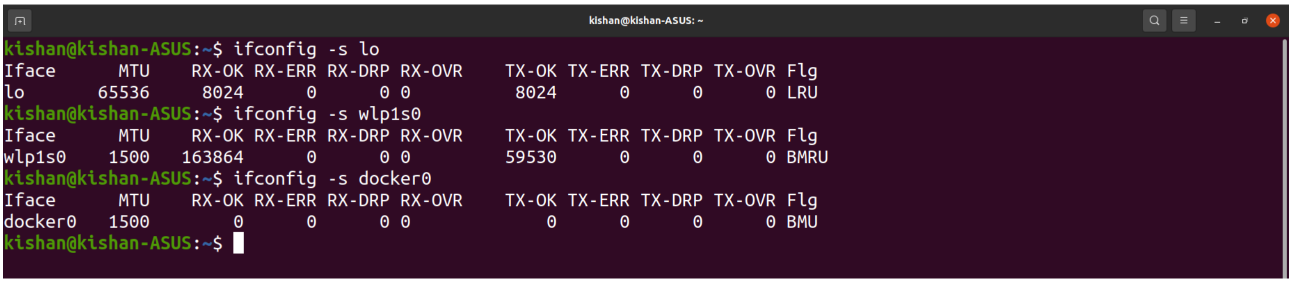
- -a : display all interfaces which are currently available, even if down
- -s : display a short list (like netstat -i)
- -v : be more verbose for some error conditions
- interface : The name of the interface. This is usually a driver name followed by a unit number, for example eth0 for the first Ethernet interface. If your kernel supports alias interfaces, you can specify them with syntax like eth0:0 for the first alias of eth0. You can use them to assign more addresses. To delete an alias interface use ifconfig eth0:0 down. Note: for every scope (i.e. same net with address/netmask combination) all aliases are deleted, if you delete the first (primary).
-
- up : This flag causes the interface to be activated. It is implicitly specified if an address is assigned to the interface; you can suppress this behavior when using an alias interface by appending an - to the alias (e.g. eth0:0-). It is also suppressed when using the IPv4 0.0.0.0 address as the kernel will use this to implicitly delete alias interfaces.
- down : This flag causes the driver for this interface to be shut down.
- [-]arp : Enable or disable the use of the ARP protocol on this interface.
- [-]promisc : Enable or disable the promiscuous mode of the interface. If selected, all packets on the network will be received by the interface.
- [-]allmulti : Enable or disable all-multicast mode. If selected, all multicast packets on the network will be received by the interface.
- mtu N : This parameter sets the Maximum Transfer Unit (MTU) of an interface.
- dstaddr addr : Set the remote IP address for a point-to-point link (such as PPP). This keyword is now obsolete; use the pointopoint keyword instead.
- Netmaskaddr : Set the IP network mask for this interface. This value defaults to the usual class A, B or C network mask (as derived from the interface IP address), but it can be set to any value.
- add addr/prefixlen : Add an IPv6 address to an interface.
- del addr/prefixlen : Remove an IPv6 address from an interface.
- tunnel ::aa.bb.cc.dd : Create a new SIT (IPv6-in-IPv4) device, tunnelling to the given destination.

- irq addr : Set the interrupt line used by this device. Not all devices can dynamically change their IRQ setting.
- io_addr addr : Set the start address in I/O space for this device.
- mem_start addr : Set the start address for shared memory used by this device. Only a few devices need this.
- media type : Set the physical port or medium type to be used by the device. Not all devices can change this setting, and those that can vary in what values they support. Typical values for type are 10base2 (thin Ethernet), 10baseT (twisted-pair 10Mbps Ethernet), AUI (external transceiver) and so on. The special medium type of auto can be used to tell the driver to autosense the media. Again, not all drivers can do this.
- [-]broadcast [addr] : If the address argument is given, set the protocol broadcast address for this interface. Otherwise, set (or clear) the IFF_BROADCAST flag for the interface.
- [-]pointopoint [addr] : This keyword enables the point-to-point mode of an interface, meaning that it is a direct link between two machines with nobody else listening on it.If the address argument is also given, set the protocol address of the other side of the link, just like the obsolete dstaddr keyword does. Otherwise, set or clear the **IFF_POINTOPOINT** flag for the interface.
- hw class address : Set the hardware address of this interface, if the device driver supports this operation. The keyword must be followed by the name of the hardware class and the printable ASCII equivalent of the hardware address. Hardware classes currently supported include ether (Ethernet), ax25 (AMPR AX.25), ARCnet and netrom (AMPR NET/ROM).
- Multicast : Set the multicast flag on the interface. This should not normally be needed as the drivers set the flag correctly themselves.
- address : The IP address to be assigned to this interface.
- txqueuelen length : Set the length of the transmit queue of the device. It is useful to set this to small values for slower devices with a high latency (modem links, ISDN) to prevent fast bulk transfers from disturbing interactive traffic like telnet too much.

Examples :

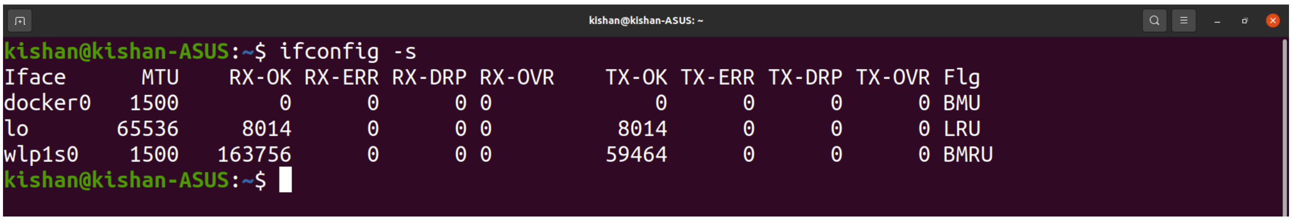
ifconfig -s <interface_name> : This option displays shorthand information about the mentioned interface.

Output :



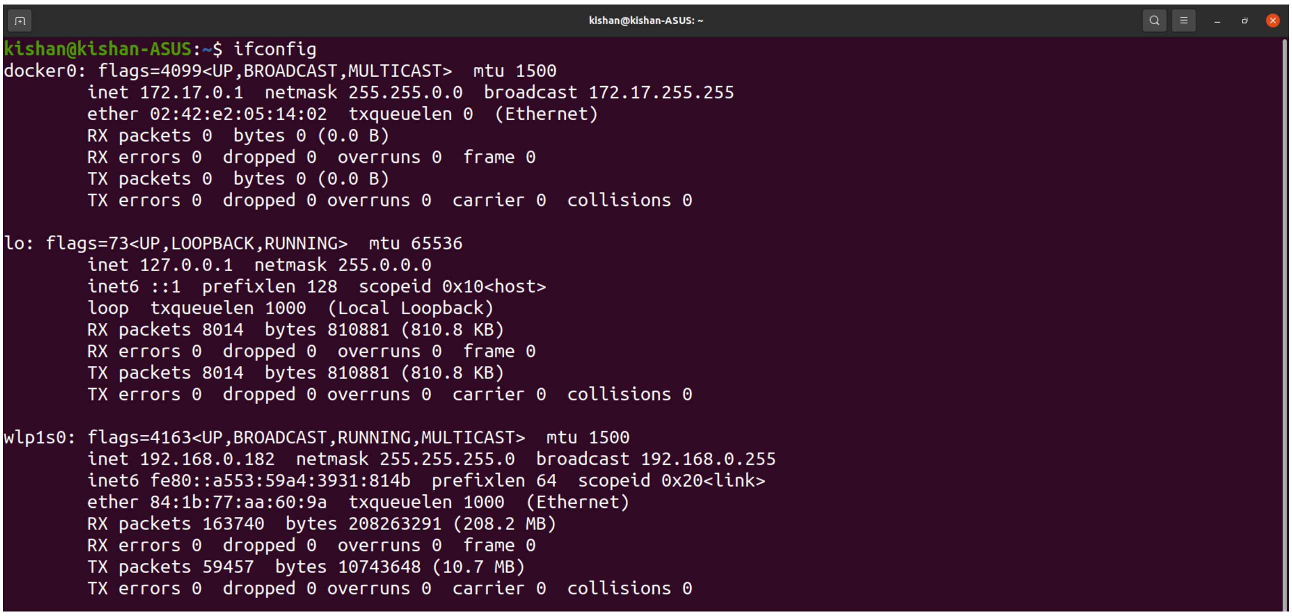
```
kishan@kishan-ASUS:~$ ifconfig -s lo
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
lo       65536     8024      0      0 0          8024      0      0      0 LRU
kishan@kishan-ASUS:~$ ifconfig -s wlp1s0
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
wlp1s0    1500   163864      0      0 0         59530      0      0      0 BMRU
kishan@kishan-ASUS:~$ ifconfig -s docker0
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
docker0   1500        0      0      0 0             0      0      0      0 BMU
kishan@kishan-ASUS:~$ 
```

When not using any interface names it will display shorthand info of all the interfaces available.



```
kishan@kishan-ASUS:~$ ifconfig -s
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
docker0   1500        0      0      0 0             0      0      0      0 BMU
lo       65536     8014      0      0 0          8014      0      0      0 LRU
wlp1s0    1500   163756      0      0 0         59464      0      0      0 BMRU
kishan@kishan-ASUS:~$ 
```

We can set ipaddress using **ifconfig** command.



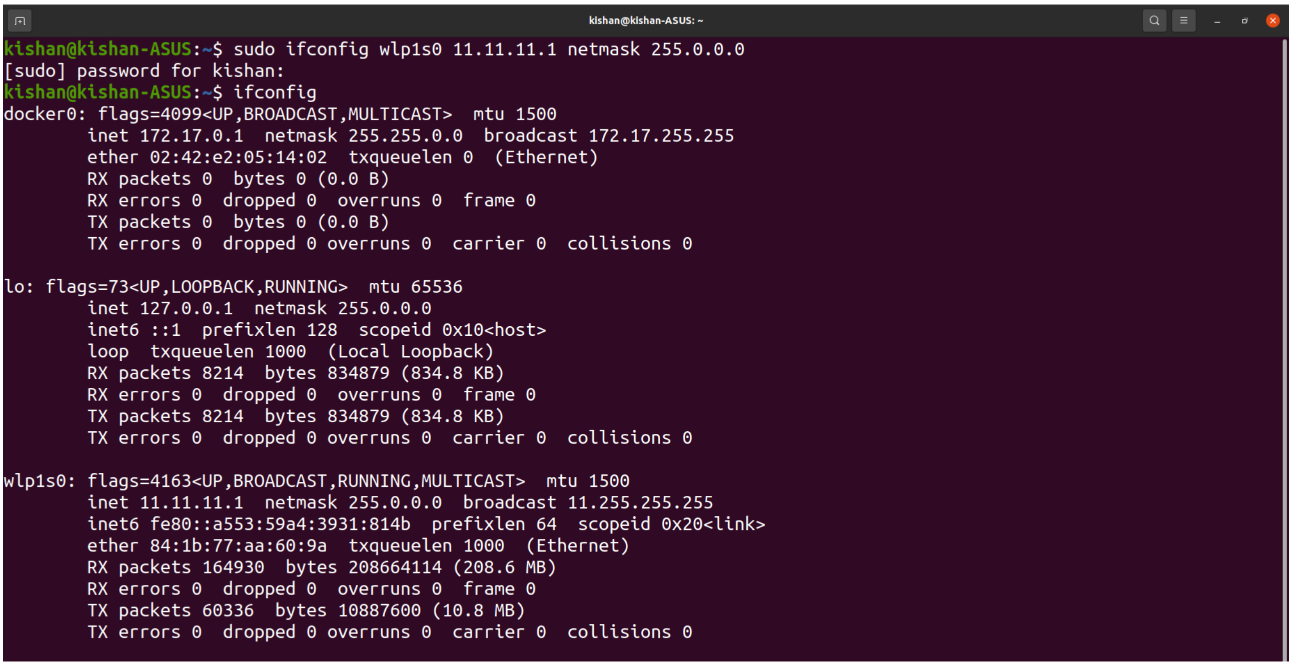
```
kishan@kishan-ASUS:~$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:e2:05:14:02  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

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 8014  bytes 810881 (810.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8014  bytes 810881 (810.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.182  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a553:59a4:3931:814b  prefixlen 64  scopeid 0x20<link>
        ether 84:1b:77:aa:60:9a  txqueuelen 1000  (Ethernet)
        RX packets 163740  bytes 208263291 (208.2 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 59457  bytes 10743648 (10.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Checking ipAddress by ifconfig command.

Now changing the ipaddress by following command and then verifying the same by ifconfig command.

$sudo ifconfig <interface_name> <New_IP_Address> netmask <Subnetmask>

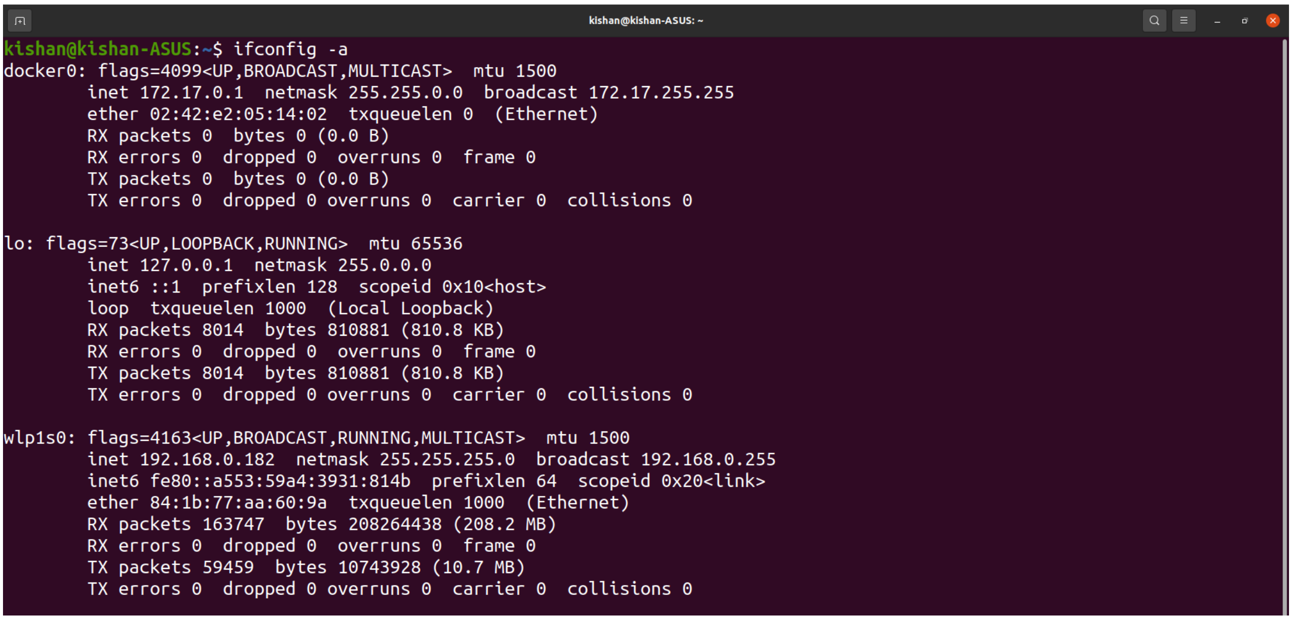
Output :



```
kishan@kishan-ASUS:~$ sudo ifconfig wlp1s0 11.11.11.1 netmask 255.0.0.0
[sudo] password for kishan:
kishan@kishan-ASUS:~$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:e2:05:14:02  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

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 8214  bytes 834879 (834.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8214  bytes 834879 (834.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 11.11.11.1  netmask 255.0.0.0  broadcast 11.255.255.255
        inet6 fe80::a553:59a4:3931:814b  prefixlen 64  scopeid 0x20<link>
        ether 84:1b:77:aa:60:9a  txqueuelen 1000  (Ethernet)
        RX packets 164930  bytes 208664114 (208.6 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 60336  bytes 10887600 (10.8 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Changed ip address (to class A) and set netmask.Verified by ifconfig

ipconfig -a : Displays all the active and nonactive network interfaces available on the system.



```
kishan@kishan-ASUS:~$ ifconfig -a
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:e2:05:14:02  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

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 8014  bytes 810881 (810.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8014  bytes 810881 (810.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.182  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a553:59a4:3931:814b  prefixlen 64  scopeid 0x20<link>
        ether 84:1b:77:aa:60:9a  txqueuelen 1000  (Ethernet)
        RX packets 163747  bytes 208264438 (208.2 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 59459  bytes 10743928 (10.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```