

Configuración de NAT en NFTables.

Lo primero será comprobar que tenemos nftables instalado, para eso podemos usar el comando `nft -help` y si nos aparecen los comandos de ayuda, es porque efectivamente está instalado.

```
root@serverseguridad:/home/cesar# nft -help
Usage: nft [ options ] [ cmds ... ]

Options (general):
-h, --help                Show this help
-v, --version             Show version information
-V                        Show extended version information

Options (ruleset input handling):
-f, --file <filename>    Read input from <filename>
-D, --define <name=value> Define variable, e.g. --define foo=1.2.3.4
-i, --interactive        Read input from interactive CLI
-I, --includepath <directory> Add <directory> to the paths searched for include files. Default is: /etc
-c, --check              Check commands validity without actually applying the changes.
-o, --optimize           Optimize ruleset

Options (ruleset list formatting):
-a, --handle             Output rule handle.
-s, --stateless          Omit stateful information of ruleset.
-t, --terse              Omit contents of sets.
-S, --service            Translate ports to service names as described in /etc/services.
-N, --reversedns         Translate IP addresses to names.
-u, --guid               Print UID/GID as defined in /etc/passwd and /etc/group.
-n, --numeric            Print fully numerical output.
-y, --numeric-priority   Print chain priority numerically.
-p, --numeric-protocol   Print layer 4 protocols numerically.
-T, --numeric-time       Print time values numerically.

Options (command output formatting):
-e, --echo               Echo what has been added, inserted or replaced.
-j, --json               Format output in JSON
-d, --debug <level [,level ...]> Specify debugging level (scanner, parser, eval, netlink, mnl, proto-ctx, segtree, all)

root@serverseguridad:/home/cesar#
```

Habilitamos el servicio nftables y comprobamos si está operativo.

```
root@serverseguridad:/home/cesar# systemctl status nftables.service
● nftables.service - nftables
   Loaded: loaded (/lib/systemd/system/nftables.service; enabled; vendor preset: enabled)
   Active: active (exited) since Thu 2022-10-13 18:28:42 UTC; 5min ago
     Docs: man:nft(8)
           http://wiki.nftables.org
   Process: 2356 ExecStart=/usr/sbin/nft -f /etc/nftables.conf (code=exited, status=0/SUCCESS)
  Main PID: 2356 (code=exited, status=0/SUCCESS)
    CPU: 5ms

oct 13 18:28:42 serverseguridad systemd[1]: nftables.service: Deactivated successfully.
oct 13 18:28:42 serverseguridad systemd[1]: Stopped nftables.
oct 13 18:28:42 serverseguridad systemd[1]: Starting nftables ...
oct 13 18:28:42 serverseguridad systemd[1]: Finished nftables.
root@serverseguridad:/home/cesar# date
jue 13 oct 2022 18:34:18 UTC
root@serverseguridad:/home/cesar#
```

Ahora vamos a editar el fichero `nftables.conf` y añadir las tablas de filter y nat, con sus respectivas cadenas.

```
#!/usr/sbin/nft -f

flush ruleset

table ip filter {
    chain input {
        type filter hook input priority 0; policy drop;

        # permitir el loopback
        iifname lo accept;

        # permitir conexiones ya establecidas, como puede ser el ssh o el ping reply
        ct state established,related accept;

        # permitimos las futuras conexiones ssh desde la red local
        tcp dport ssh ip saddr 192.168.1.0/24 accept;

        # permitimos el echo reply desde los clientes
        ip protocol icmp icmp type 0 icmp code 0 ip saddr 192.168.10.100/32 counter accept;
        ip protocol icmp icmp type 0 icmp code 0 ip saddr 192.168.20.100/32 counter accept;

        # permitimos el echo request desde los clientes
        ip protocol icmp icmp type 8 icmp code 0 ip saddr 192.168.10.100/32 counter accept;
        ip protocol icmp icmp type 8 icmp code 0 ip saddr 192.168.20.100/32 counter accept;
    }
    chain forward {
        type filter hook forward priority 0; policy drop;

        # permitimos el ping y por lo tanto el forward entre clientes
        # de la .10.100 a la .20.100
        ip protocol icmp icmp type 8 icmp code 0 ip saddr 192.168.10.100/32 ip daddr 192.168.20.100/32 counter accept;
        ip protocol icmp icmp type 0 icmp code 0 ip daddr 192.168.10.100/32 ip saddr 192.168.20.100/32 counter accept;
        # de la .20.100 a la .10.100
        ip protocol icmp icmp type 8 icmp code 0 ip saddr 192.168.20.100/32 ip daddr 192.168.10.100/32 counter accept;
        ip protocol icmp icmp type 0 icmp code 0 ip daddr 192.168.20.100/32 ip saddr 192.168.10.100/32 counter accept;
    }
    chain output {
        type filter hook output priority 0; policy drop;

        # permitimos las futuras conexiones ssh hacia la red local
        tcp sport ssh ip daddr 192.168.1.0/24 accept;

        # permitimos el echo reply hacia los clientes
        ip protocol icmp icmp type 0 icmp code 0 ip daddr 192.168.10.100/32 counter accept;
        ip protocol icmp icmp type 0 icmp code 0 ip daddr 192.168.20.100/32 counter accept;

        # permitimos el echo request hacia los clientes
        ip protocol icmp icmp type 8 icmp code 0 ip daddr 192.168.10.100/32 counter accept;
        ip protocol icmp icmp type 8 icmp code 0 ip daddr 192.168.20.100/32 counter accept;
    }
}
table ip nat {
    chain postrouting {
    }
}
}
```

root@serverseguridad:/home/cesar# date
jue 13 oct 2022 18:34:54 UTC
root@serverseguridad:/home/cesar#

De momento hemos configurado los pings desde el servidor a los clientes y viceversa.

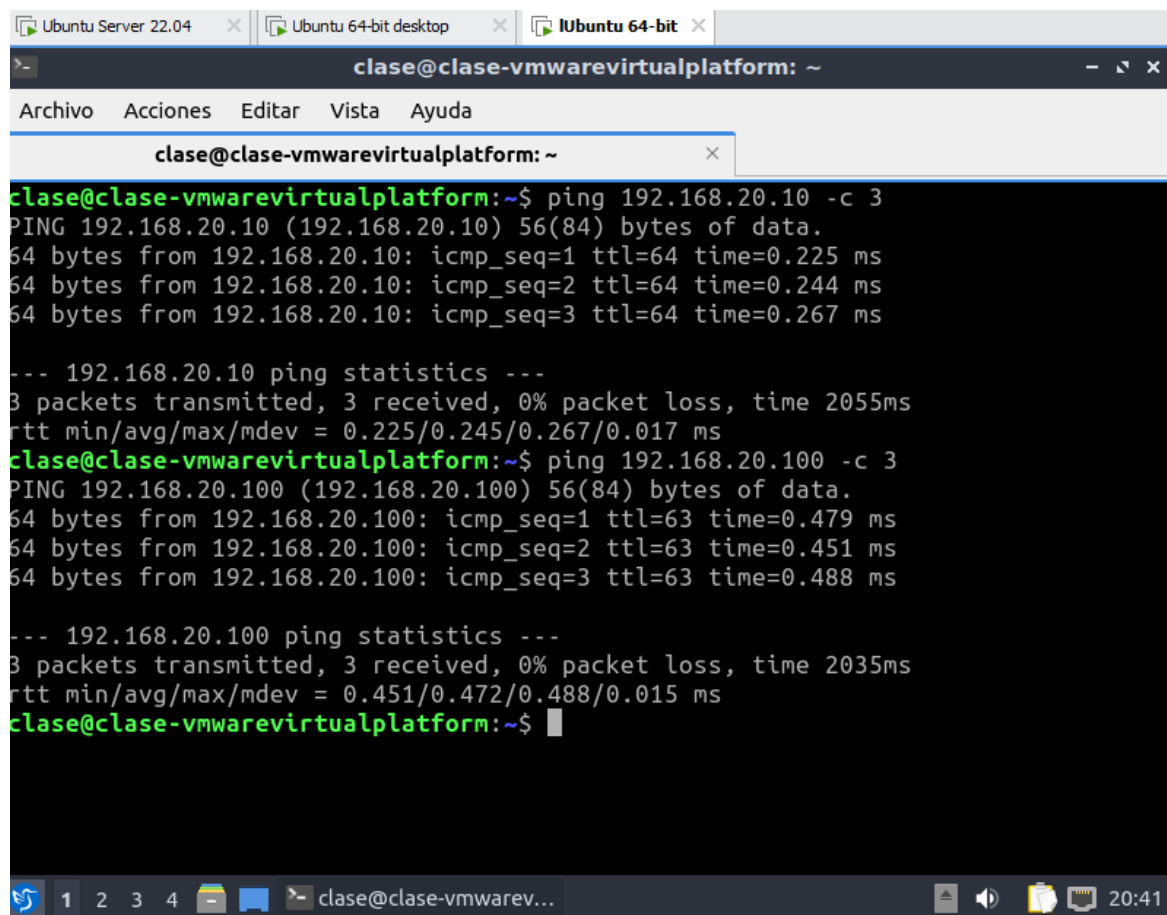
Ping desde el server a los clientes.

```
cesar@serverseguridad:~$ ping 192.168.10.100 -c 3
PING 192.168.10.100 (192.168.10.100) 56(84) bytes of data.
64 bytes from 192.168.10.100: icmp_seq=1 ttl=64 time=0.273 ms
64 bytes from 192.168.10.100: icmp_seq=2 ttl=64 time=0.258 ms
64 bytes from 192.168.10.100: icmp_seq=3 ttl=64 time=0.257 ms

--- 192.168.10.100 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2047ms
rtt min/avg/max/mdev = 0.257/0.262/0.273/0.007 ms
cesar@serverseguridad:~$ ping 192.168.20.100 -c 3
PING 192.168.20.100 (192.168.20.100) 56(84) bytes of data.
64 bytes from 192.168.20.100: icmp_seq=1 ttl=64 time=0.273 ms
64 bytes from 192.168.20.100: icmp_seq=2 ttl=64 time=0.325 ms
64 bytes from 192.168.20.100: icmp_seq=3 ttl=64 time=0.220 ms

--- 192.168.20.100 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2055ms
rtt min/avg/max/mdev = 0.220/0.272/0.325/0.042 ms
cesar@serverseguridad:~$ date
jue 13 oct 2022 18:35:23 UTC
cesar@serverseguridad:~$
```

Ping desde lubuntu a los otros dos.

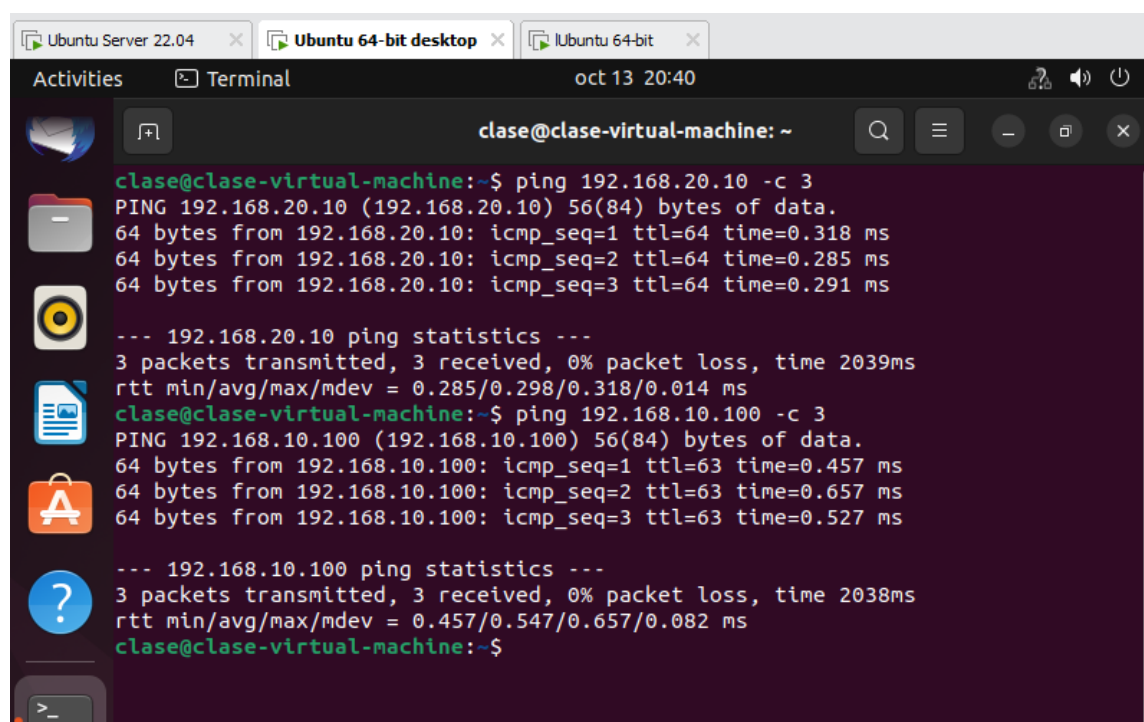


```
clase@clase-vmwarevirtualplatform: ~$ ping 192.168.20.10 -c 3
PING 192.168.20.10 (192.168.20.10) 56(84) bytes of data.
64 bytes from 192.168.20.10: icmp_seq=1 ttl=64 time=0.225 ms
64 bytes from 192.168.20.10: icmp_seq=2 ttl=64 time=0.244 ms
64 bytes from 192.168.20.10: icmp_seq=3 ttl=64 time=0.267 ms

--- 192.168.20.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2055ms
rtt min/avg/max/mdev = 0.225/0.245/0.267/0.017 ms
clase@clase-vmwarevirtualplatform:~$ ping 192.168.20.100 -c 3
PING 192.168.20.100 (192.168.20.100) 56(84) bytes of data.
64 bytes from 192.168.20.100: icmp_seq=1 ttl=63 time=0.479 ms
64 bytes from 192.168.20.100: icmp_seq=2 ttl=63 time=0.451 ms
64 bytes from 192.168.20.100: icmp_seq=3 ttl=63 time=0.488 ms

--- 192.168.20.100 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2035ms
rtt min/avg/max/mdev = 0.451/0.472/0.488/0.015 ms
clase@clase-vmwarevirtualplatform:~$
```

Ping de ubuntu a los otros dos.



```
clase@clase-virtual-machine:~$ ping 192.168.20.10 -c 3
PING 192.168.20.10 (192.168.20.10) 56(84) bytes of data.
64 bytes from 192.168.20.10: icmp_seq=1 ttl=64 time=0.318 ms
64 bytes from 192.168.20.10: icmp_seq=2 ttl=64 time=0.285 ms
64 bytes from 192.168.20.10: icmp_seq=3 ttl=64 time=0.291 ms

--- 192.168.20.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2039ms
rtt min/avg/max/mdev = 0.285/0.298/0.318/0.014 ms
clase@clase-virtual-machine:~$ ping 192.168.10.100 -c 3
PING 192.168.10.100 (192.168.10.100) 56(84) bytes of data.
64 bytes from 192.168.10.100: icmp_seq=1 ttl=63 time=0.457 ms
64 bytes from 192.168.10.100: icmp_seq=2 ttl=63 time=0.657 ms
64 bytes from 192.168.10.100: icmp_seq=3 ttl=63 time=0.527 ms

--- 192.168.10.100 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2038ms
rtt min/avg/max/mdev = 0.457/0.547/0.657/0.082 ms
clase@clase-virtual-machine:~$
```

Ahora para que funcione el ping a internet desde el servidor, basta con adaptar el input y el output.

```
chain output {
    type filter hook output priority 0; policy drop;

    # permitimos las futuras conexiones ssh hacia la red local
    tcp sport ssh ip daddr 192.168.1.0/24 accept;

    # permitimos el echo reply hacia los clientes
    ip protocol icmp icmp type 0 icmp code 0 ip daddr 192.168.10.100/32 counter accept;
    ip protocol icmp icmp type 0 icmp code 0 ip daddr 192.168.20.100/32 counter accept;

    # permitimos el echo request hacia los clientes
    ip protocol icmp icmp type 8 icmp code 0 ip daddr 192.168.10.100/32 counter accept;
    ip protocol icmp icmp type 8 icmp code 0 ip daddr 192.168.20.100/32 counter accept;

    # permitimos el echo request hacia internet (es decir todas las redes, por eso pongo la 0.0.0.0/0)
    ip protocol icmp icmp type 8 icmp code 0 ip daddr 0.0.0.0/0 counter accept;
}
```

```
table ip filter {
    chain input {
        type filter hook input priority 0; policy drop;

        # permitir el loopback
        iifname lo accept;

        # permitir conexiones ya establecidas, como puede ser el ssh o el ping reply
        ct state established,related accept;

        # permitimos las futuras conexiones ssh desde la red local
        tcp dport ssh ip saddr 192.168.1.0/24 accept;

        # permitimos el echo reply desde los clientes
        ip protocol icmp icmp type 0 icmp code 0 ip saddr 192.168.10.100/32 counter accept;
        ip protocol icmp icmp type 0 icmp code 0 ip saddr 192.168.20.100/32 counter accept;

        # permitimos el echo request desde los clientes
        ip protocol icmp icmp type 8 icmp code 0 ip saddr 192.168.10.100/32 counter accept;
        ip protocol icmp icmp type 8 icmp code 0 ip saddr 192.168.20.100/32 counter accept;

        # permitimos el echo request desde internet (es decir todas las redes, por eso pongo la 0.0.0.0/0)
        ip protocol icmp icmp type 8 icmp code 0 ip saddr 0.0.0.0/0 counter accept;
    }
}
```

Probamos a hacer ping a internet desde el server.

```
cesar@serverseguridad:~$ ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=64 time=0.656 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=64 time=0.548 ms
^C
--- 1.1.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1028ms
rtt min/avg/max/mdev = 0.548/0.602/0.656/0.054 ms
cesar@serverseguridad:~$
```

Para los clientes, hay que adaptar el forward para que permita las conexiones entre internet y los clientes. Y la cadena de postrouting de nat, para que haga enmascaramiento o source nat de las ips privadas a la pública.

```

}
chain forward {
    type filter hook forward priority 0; policy drop;

    #permitimos el ping y por lo tanto el forward entre clientes
    # de la .10.100 a la .20.100
    ip protocol icmp icmp type 8 icmp code 0 ip saddr 192.168.10.100/32 ip daddr 192.168.20.100/32 counter accept;
    ip protocol icmp icmp type 0 icmp code 0 ip daddr 192.168.10.100/32 ip saddr 192.168.20.100/32 counter accept;
    # de la .20.100 a la .10.100
    ip protocol icmp icmp type 8 icmp code 0 ip saddr 192.168.20.100/32 ip daddr 192.168.10.100/32 counter accept;
    ip protocol icmp icmp type 0 icmp code 0 ip daddr 192.168.20.100/32 ip saddr 192.168.10.100/32 counter accept;

    #permitimos el ping de los clientes a internet
    #desde la .10.100
    ip protocol icmp icmp type 8 icmp code 0 ip saddr 192.168.10.100/32 ip daddr 0.0.0.0/0 counter accept;
    ip protocol icmp icmp type 0 icmp code 0 ip daddr 192.168.10.100/32 ip saddr 0.0.0.0/0 counter accept;
    #desde la .20.100
    ip protocol icmp icmp type 8 icmp code 0 ip saddr 192.168.20.100/32 ip daddr 0.0.0.0/0 counter accept;
    ip protocol icmp icmp type 0 icmp code 0 ip daddr 192.168.20.100/32 ip saddr 0.0.0.0/0 counter accept;

table ip nat {
    chain postrouting {
        type nat hook postrouting priority 0; policy accept;

        #permitimos el enmascaramiento de los clientes hacia internet.

        #de la .10.100
        oifname "ens33" ip saddr 192.168.10.100/32 counter masquerade

        #de la .20.100
        oifname "ens33" ip saddr 192.168.20.100/32 counter masquerade
    }
}

```

Probamos a hacer los correspondientes pings.

Lubuntu

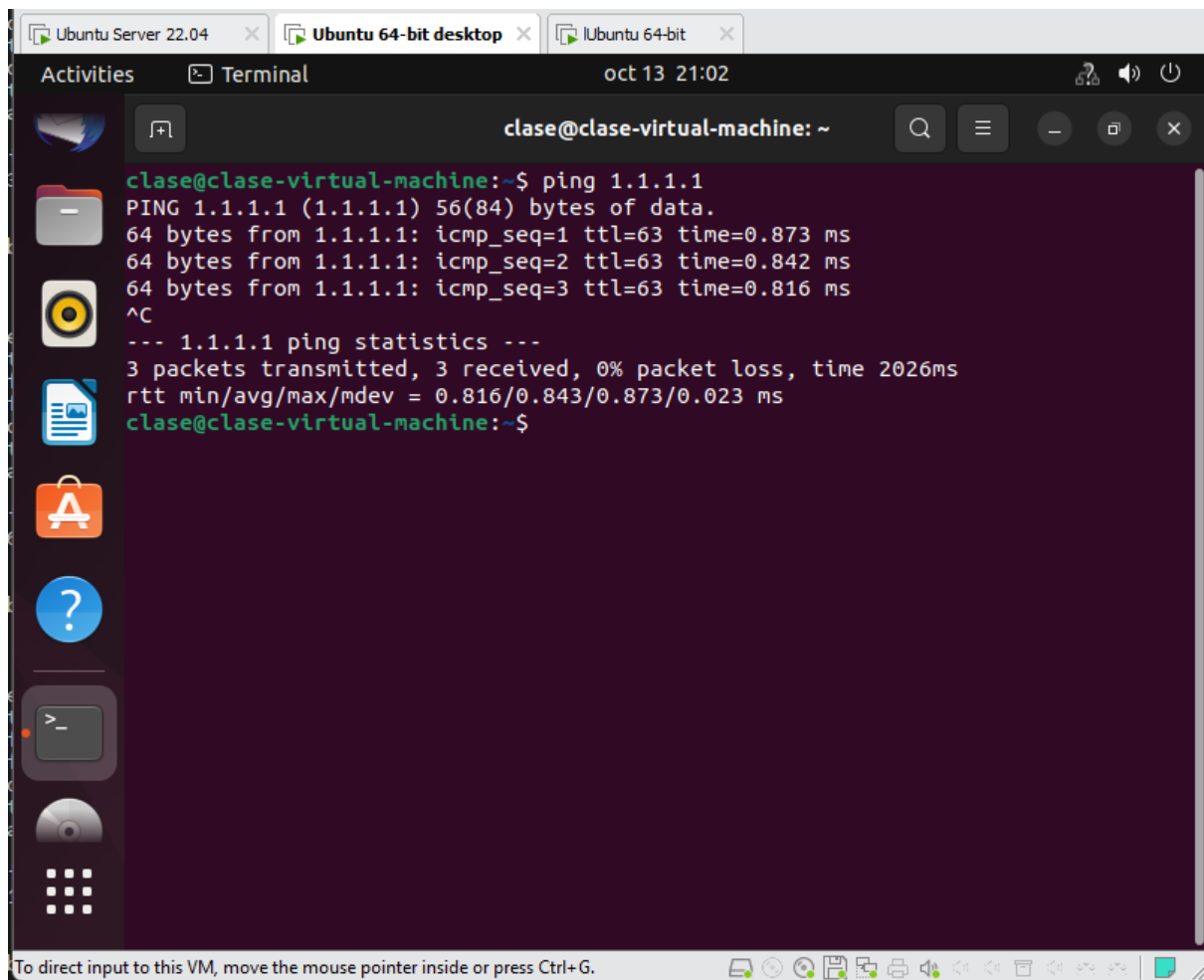
The screenshot shows a terminal window titled 'clase@clase-vmwarevirtualplatform: ~'. The user has executed the command 'ping 1.1.1.1'. The output shows three successful pings with varying times (0.915 ms, 0.826 ms, 0.866 ms). Below the pings, the statistics are displayed: '--- 1.1.1.1 ping statistics ---', '3 packets transmitted, 3 received, 0% packet loss, time 2026ms', and 'rtt min/avg/max/mdev = 0.826/0.869/0.915/0.036 ms'. The terminal window is part of a desktop environment with other windows visible in the background.

```

clase@clase-vmwarevirtualplatform: ~
Archivo Acciones Editar Vista Ayuda
clase@clase-vmwarevirtualplatform: ~
clase@clase-vmwarevirtualplatform:~$ ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data:
64 bytes from 1.1.1.1: icmp_seq=1 ttl=63 time=0.915 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=63 time=0.826 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=63 time=0.866 ms
^C
--- 1.1.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2026ms
rtt min/avg/max/mdev = 0.826/0.869/0.915/0.036 ms
clase@clase-vmwarevirtualplatform:~$

```

Ubuntu



The screenshot displays an Ubuntu desktop environment. At the top, there are three window tabs: 'Ubuntu Server 22.04', 'Ubuntu 64-bit desktop', and 'Ubuntu 64-bit'. The main window is titled 'Terminal' and shows the command prompt 'clase@clase-virtual-machine: ~'. The user has executed the command 'ping 1.1.1.1', which has returned the following output:

```
clase@clase-virtual-machine:~$ ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=63 time=0.873 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=63 time=0.842 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=63 time=0.816 ms
^C
--- 1.1.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2026ms
rtt min/avg/max/mdev = 0.816/0.843/0.873/0.023 ms
clase@clase-virtual-machine:~$
```

The desktop environment includes a sidebar with various application icons (Files, Music, Videos, Documents, Applications, Help, Settings, and a Dash icon) and a top bar with system status indicators (network, volume, and power). The bottom of the screen features a status bar with the text 'To direct input to this VM, move the mouse pointer inside or press Ctrl+G.' and a series of icons for window management and system utilities.