

# WIREGUARD LABORATORIO L2



## LABORATORIO L2

Instalado Debian, abro un terminal y ejecuto el comando ip a para averiguar que ip tiene mi host. En este caso se trata de la enp0s3 192.168.1.145/24

```
root@debianIvanaSanchez:/home/usuario# ip a
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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel
    state UP group default qlen 1000
        link/ether 08:00:27:96:3d:94 brd ff:ff:ff:ff:ff:ff
        inet 192.168.1.145/24 brd 192.168.1.255 scope global dynamic enp
    0s3
            valid_lft 86342sec preferred_lft 86342sec
            inet6 fe80::a00:27ff:fe96:3d94/64 scope link
                valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel
    state UP group default qlen 1000
        link/ether 08:00:27:36:58:06 brd ff:ff:ff:ff:ff:ff
        inet 192.168.100.2/24 brd 192.168.100.255 scope global enp0s8
            valid_lft forever preferred_lft forever
            inet6 fe80::a00:27ff:fe36:5806/64 scope link
                valid_lft forever preferred_lft forever
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noque
```

### PASO 1

Actualizamos con sudo **apt update && apt upgrade -y**

Seguidamente instalaremos las dependencias necesarias en nuestro Debian Host con: **sudo apt install lxc lxc-templates wireguard -y**

The screenshot shows a terminal window titled "DebianRegaloLDPA [Corriendo] - Oracle VM VirtualBox". The window has a dark theme with a light gray header bar. The header includes menu items: Archivo, Máquina, Ver, Entrada, Dispositivos, Ayuda. Below the header is a toolbar with tabs: Actividades, Terminal, and a date/time indicator: 5 de ago 12:36. To the right of the toolbar are icons for battery level (100%), volume, and network. The main area of the terminal is a dark gray scrollable window. It displays the following command and its output:

```
root@debianIvanaSanchez:/home/usuario# sudo apt install lxc lxc-temp  
lates wireguard  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias... Hecho  
Leyendo la información de estado... Hecho  
El paquete indicado a continuación se instaló de forma automática y  
ya no es necesario.  
  linux-image-6.1.0-29-amd64  
Utilice «sudo apt autoremove» para eliminarlo.  
Se instalarán los siguientes paquetes adicionales:  
  arch-test bridge-utils busybox-static cloud-image-utils  
  debootstrap distro-info fakechroot genisoimage ibverbs-providers  
  libaio1 libdistro-info-perl libfakechroot libfmt9 libgfapi0  
  libgfrpc0 libgfxdr0 libglusterfs0 libibverbs1 libiscsi7  
  liblxcommon liblxci1 libpam-cgfs librados2 librbd1 librdmacm1  
  libssh-4 libsubid4 liburing2 lxcfs mmdebstrap qemu-block-extra  
  qemu-utils rsync uidmap uuid-runtime wireguard-tools  
Paquetes sugeridos:  
  mtools squid-deb-proxy-client ubuntu-archive-keyring shunit2  
  wodim cdrkit-doc btrfs-progs lvm2 python3-lxc qemu-user-static  
  apt-transport-tor genext2fs perl-doc qemu-user squashfs-tools-ng  
  python3-braceexpand openresolv | resolvconf  
Los siguientes paquetes se ELIMINARÁN:  
  busybox
```

## PASO 2

Creamos los contenedores para cada nodo: A, B, C, D, E

***sudo lxc-create -t debian -n nodoA*** y así sucesivamente con cada uno de los contenedores.

The screenshot shows a terminal window titled "DebianRegaloLDPA [Corriendo] - Oracle VM VirtualBox". The window has a dark theme with white text. The title bar includes menu items: Archivo, Máquina, Ver, Entrada, Dispositivos, Ayuda. The top bar also shows the date and time: 5 de ago 12:39. A red box highlights the command being run:

```
root@debianIvanaSanchez:/home/usuario# sudo lxc-create -t debian -n nodoA
debootstrap is /usr/sbin/debootstrap
Checking cache download in /var/cache/lxc/debian/rootfs-stable-amd64
...
Downloading debian minimal ...
I: Target architecture can be executed
I: Retrieving InRelease
I: Checking Release signature
I: Valid Release signature (key id 4D64FEC119C2029067D6E791F8D2585B8
783D481)
I: Retrieving Packages
I: Validating Packages
I: Resolving dependencies of required packages...
I: Resolving dependencies of base packages...
I: Checking component main on http://deb.debian.org/debian...
I: Retrieving adduser 3.134
I: Validating adduser 3.134
I: Retrieving apt 2.6.1
I: Validating apt 2.6.1
I: Retrieving apt-transport-https 2.6.1
I: Validating apt-transport-https 2.6.1
I: Retrieving base-files 12.4+deb12u11
I: Validating base-files 12.4+deb12u11
```

### PASO 3

Arrancamos los contenedores con el comando ***sudo lxc-start -n nodoA -d*** y comprobamos que está corriendo con ***sudo lxc-ls -fancy***

The screenshot shows a terminal window titled "Terminal" with the command "usuario@debianIvanaSanchez: ~". The terminal displays the following commands and their outputs:

```
root@debianIvanaSanchez:/home/usuario# sudo lxc-start -n nodoA -d
lxc-start: nodoA: ./src/lxc/tools/lxc_start.c: main: 256 Container
is already running
root@debianIvanaSanchez:/home/usuario# sudo lxc-start -n nodoB -d
root@debianIvanaSanchez:/home/usuario# sudo lxc-start -n nodoC -d
root@debianIvanaSanchez:/home/usuario# sudo lxc-start -n nodoD -d
root@debianIvanaSanchez:/home/usuario# sudo lxc-start -n nodoE -d
root@debianIvanaSanchez:/home/usuario# sudo lxc-ls --fancy
```

NAME	STATE	AUTOSTART	GROUPS	IPV4	IPV6	UNPRIVILEGED
nodoA	RUNNING	0	-	10.0.3.200	-	false
nodoB	RUNNING	0	-	10.0.3.204	-	false
nodoC	RUNNING	0	-	10.0.3.237	-	false
nodoD	RUNNING	0	-	10.0.3.31	-	false
nodoE	RUNNING	0	-	10.0.3.117	-	false

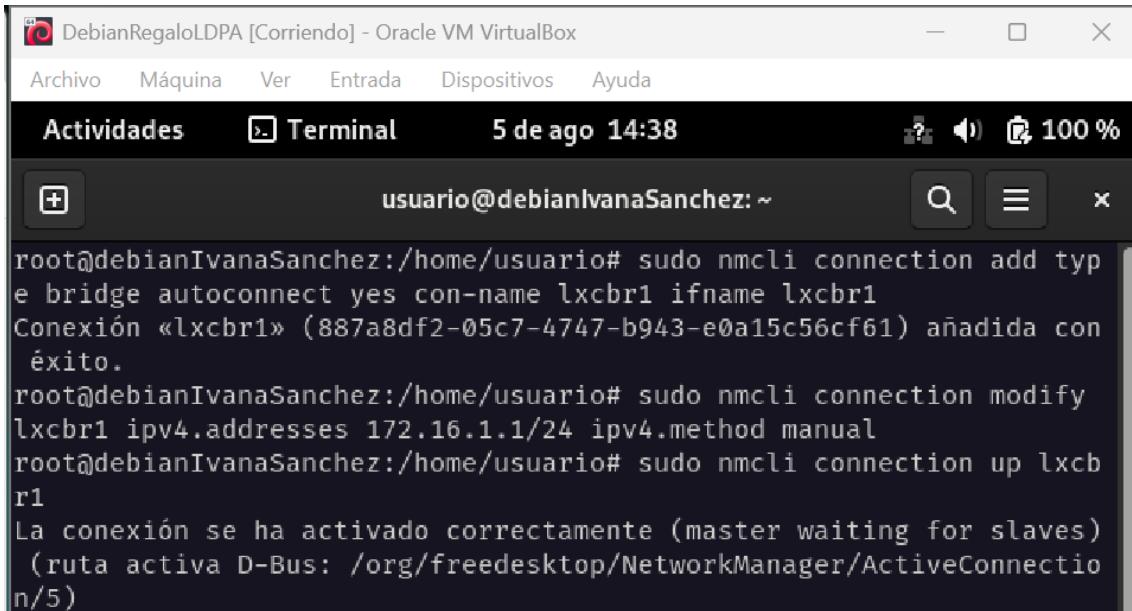
#### PASO 4

Asignamos un puente para cada nodo

**Creamos las interfaces de red de puente (lxcbr1, lxcbr2, etc.)** en tu máquina principal. Estas interfaces conectarán los contenedores entre sí, simulando los enlaces físicos del laboratorio.

- **Para la red A-B (172.16.1.0/24):**
  - sudo nmcli connection add type bridge autoconnect yes con-name lxcbr1 ifname lxcbr1
  - sudo nmcli connection modify lxcbr1 ipv4.addresses 172.16.1.1/24 ipv4.method manual
  - sudo nmcli connection up lxcbr1
- **Para la red B-C (172.16.2.0/24):**

- sudo nmcli connection add type bridge autoconnect yes con-name lxcbr2 ifname lxcbr2
  - sudo nmcli connection modify lxcbr2 ipv4.addresses 172.16.2.1/24 ipv4.method manual
  - sudo nmcli connection up lxcbr2
- **Para la red C-D (172.16.3.0/24):**
    - sudo nmcli connection add type bridge autoconnect yes con-name lxcbr3 ifname lxcbr3
    - sudo nmcli connection modify lxcbr3 ipv4.addresses 172.16.3.1/24 ipv4.method manual
    - sudo nmcli connection up lxcbr3
  - **Para la red D-E (172.16.4.0/24):**
    - sudo nmcli connection add type bridge autoconnect yes con-name lxcbr4 ifname lxcbr4
    - sudo nmcli connection modify lxcbr4 ipv4.addresses 172.16.4.1/24 ipv4.method manual
    - sudo nmcli connection up lxcbr4



The screenshot shows a terminal window titled "DebianRegaloLDPA [Corriendo] - Oracle VM VirtualBox". The window has a menu bar with "Archivo", "Máquina", "Ver", "Entrada", "Dispositivos", and "Ayuda". Below the menu is a toolbar with "Actividades", "Terminal", the date "5 de ago 14:38", and a battery icon at 100%. The terminal itself has a header bar with a plus sign icon, the user information "usuario@debianIvanaSanchez: ~", and icons for search, minimize, maximize, and close. The terminal content is as follows:

```
root@debianIvanaSanchez:/home/usuario# sudo nmcli connection add type bridge autoconnect yes con-name lxcbr1 ifname lxcbr1
Conexión «lxcbr1» (887a8df2-05c7-4747-b943-e0a15c56cf61) añadida con éxito.
root@debianIvanaSanchez:/home/usuario# sudo nmcli connection modify lxcbr1 ipv4.addresses 172.16.1.1/24 ipv4.method manual
root@debianIvanaSanchez:/home/usuario# sudo nmcli connection up lxcbr1
La conexión se ha activado correctamente (master waiting for slaves)
(ruta activa D-Bus: /org/freedesktop/NetworkManager/ActiveConnection/5)
```

## Crear los Archivos de Configuración de los Contenedores

Ahora necesitas entrar en la configuración de cada contenedor y conectarlos a las interfaces de puente que acabas de crear. Los archivos de configuración se encuentran en /var/lib/lxc/<nombre-del-nodo>/config.

- **Edita el archivo del Nodo A:**

```
sudo nano /var/lib/lxc/nodoA/config
```

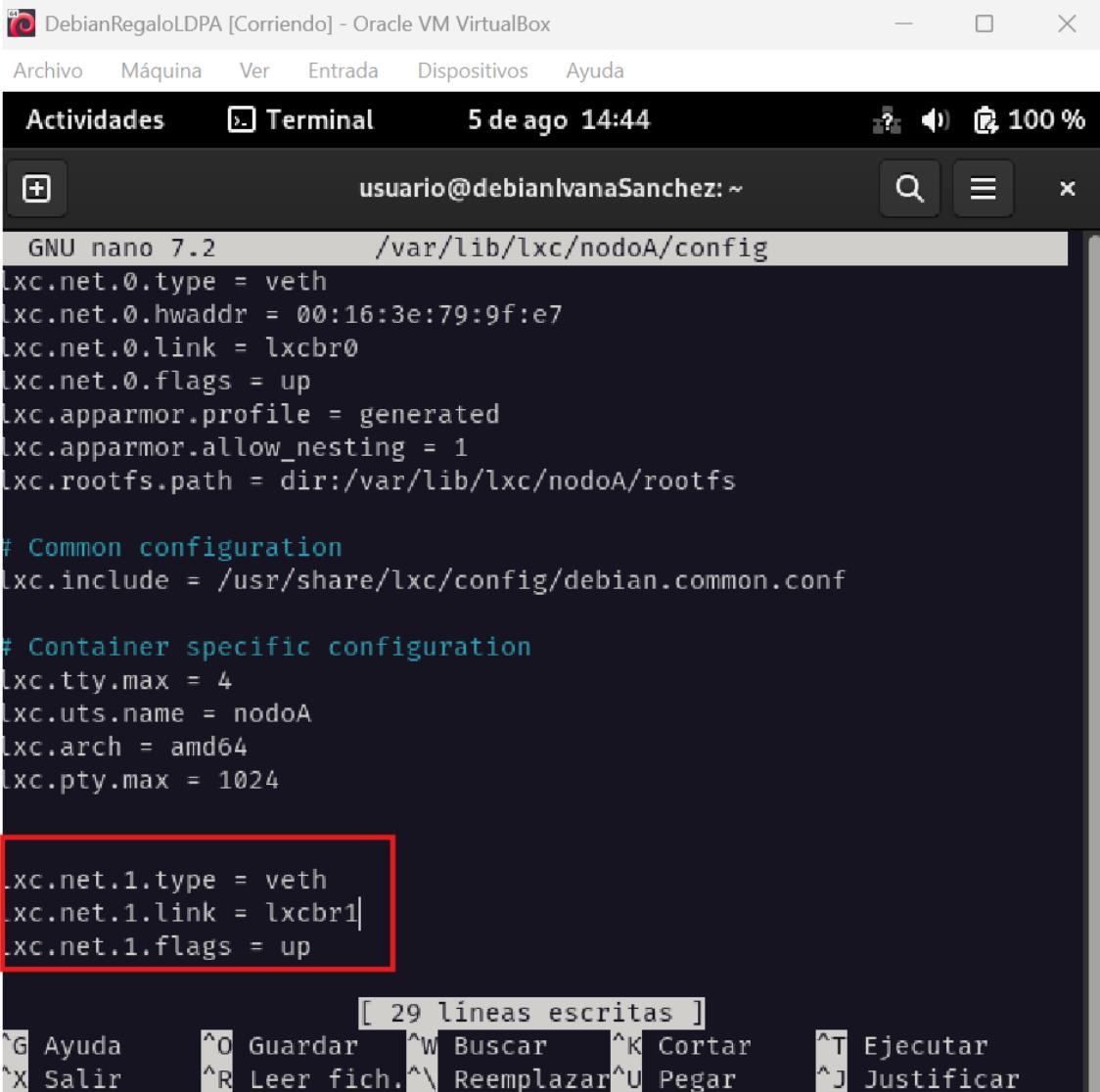
Añade estas líneas para conectarlo a lxcbr1:

```
lxc.net.1.type = veth
```

```
lxc.net.1.link = lxcbr1
```

```
lxc.net.1.flags = up
```

Guarda y cierra el archivo (Ctrl + X, luego S, luego Enter).



The screenshot shows a terminal window titled "DebianRegaloLDPA [Corriendo] - Oracle VM VirtualBox". The window has a dark theme. The title bar includes the window name, a minimize button, a maximize button, and a close button. Below the title bar is a menu bar with "Archivo", "Máquina", "Ver", "Entrada", "Dispositivos", and "Ayuda". The main area of the terminal shows the command "GNU nano 7.2" followed by the path "/var/lib/lxc/nodoA/config". The configuration file content is displayed in the terminal:

```
lxc.net.0.type = veth
lxc.net.0.hwaddr = 00:16:3e:79:9f:e7
lxc.net.0.link = lxcbr0
lxc.net.0.flags = up
lxc.apparmor.profile = generated
lxc.apparmor.allow_nesting = 1
lxc.rootfs.path = dir:/var/lib/lxc/nodoA/rootfs

# Common configuration
lxc.include = /usr/share/lxc/config/debian.common.conf

# Container specific configuration
lxc.tty.max = 4
lxc.uts.name = nodoA
lxc.arch = amd64
lxc.pty.max = 1024

lxc.net.1.type = veth
lxc.net.1.link = lxcbr1
lxc.net.1.flags = up
```

The last three lines of the configuration file are highlighted with a red rectangle. At the bottom of the terminal window, there is a status bar with the text "[ 29 líneas escritas ]" and a set of keyboard shortcuts:

^G Ayuda	^O Guardar	^W Buscar	^K Cortar	^T Ejecutar
^X Salir	^R Leer fich.	^V Reemplazar	^U Pegar	^J Justificar

- **Edita el archivo del Nodo B:**

```
sudo nano /var/lib/lxc/nodoB/config
```

Añade estas líneas para conectarlo a lxcbr1 y lxcbr2:

```
lxc.net.1.type = veth  
lxc.net.1.link = lxcbr1  
lxc.net.1.flags = up  
lxc.net.2.type = veth  
lxc.net.2.link = lxcbr2  
lxc.net.2.flags = up
```

Guarda y cierra el archivo.

- **Edita el archivo del Nodo C:**

```
sudo nano /var/lib/lxc/nodoC/config
```

Añade estas líneas para conectarlo a lxcbr2 y lxcbr3:

```
lxc.net.1.type = veth  
lxc.net.1.link = lxcbr2  
lxc.net.1.flags = up  
lxc.net.2.type = veth  
lxc.net.2.link = lxcbr3  
lxc.net.2.flags = up
```

Guarda y cierra el archivo.

- **Edita el archivo del Nodo D:**

```
sudo nano /var/lib/lxc/nodoD/config
```

Añade estas líneas para conectarlo a lxcbr3 y lxcbr4:

```
lxc.net.1.type = veth  
lxc.net.1.link = lxcbr3  
lxc.net.1.flags = up  
lxc.net.2.type = veth  
lxc.net.2.link = lxcbr4
```

```
lxc.net.2.flags = up
```

Guarda y cierra el archivo.

- **Edita el archivo del Nodo E:**

```
sudo nano /var/lib/lxc/nodoE/config
```

**Añade estas líneas** para conectarlo a lxcbr4:

```
lxc.net.1.type = veth
```

```
lxc.net.1.link = lxcbr4
```

```
lxc.net.1.flags = up
```

Guarda y cierra el archivo.

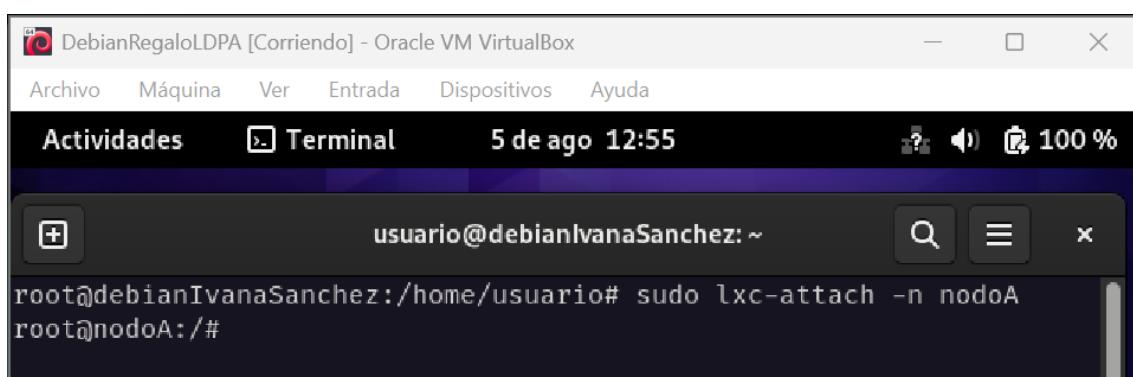
Volvemos a levantar todos los nodos-

```
root@debianIvanaSanchez:/home/usuario# sudo lxc-start -n nodoA -d
root@debianIvanaSanchez:/home/usuario# sudo lxc-start -n nodoB -d
root@debianIvanaSanchez:/home/usuario# sudo lxc-start -n nodoC -d
root@debianIvanaSanchez:/home/usuario# sudo lxc-start -n nodoD -d
root@debianIvanaSanchez:/home/usuario# sudo lxc-start -n nodoE -d
root@debianIvanaSanchez:/home/usuario# |
```

## PASO 5

Entramos en cada nodo y configuramos WireGuard

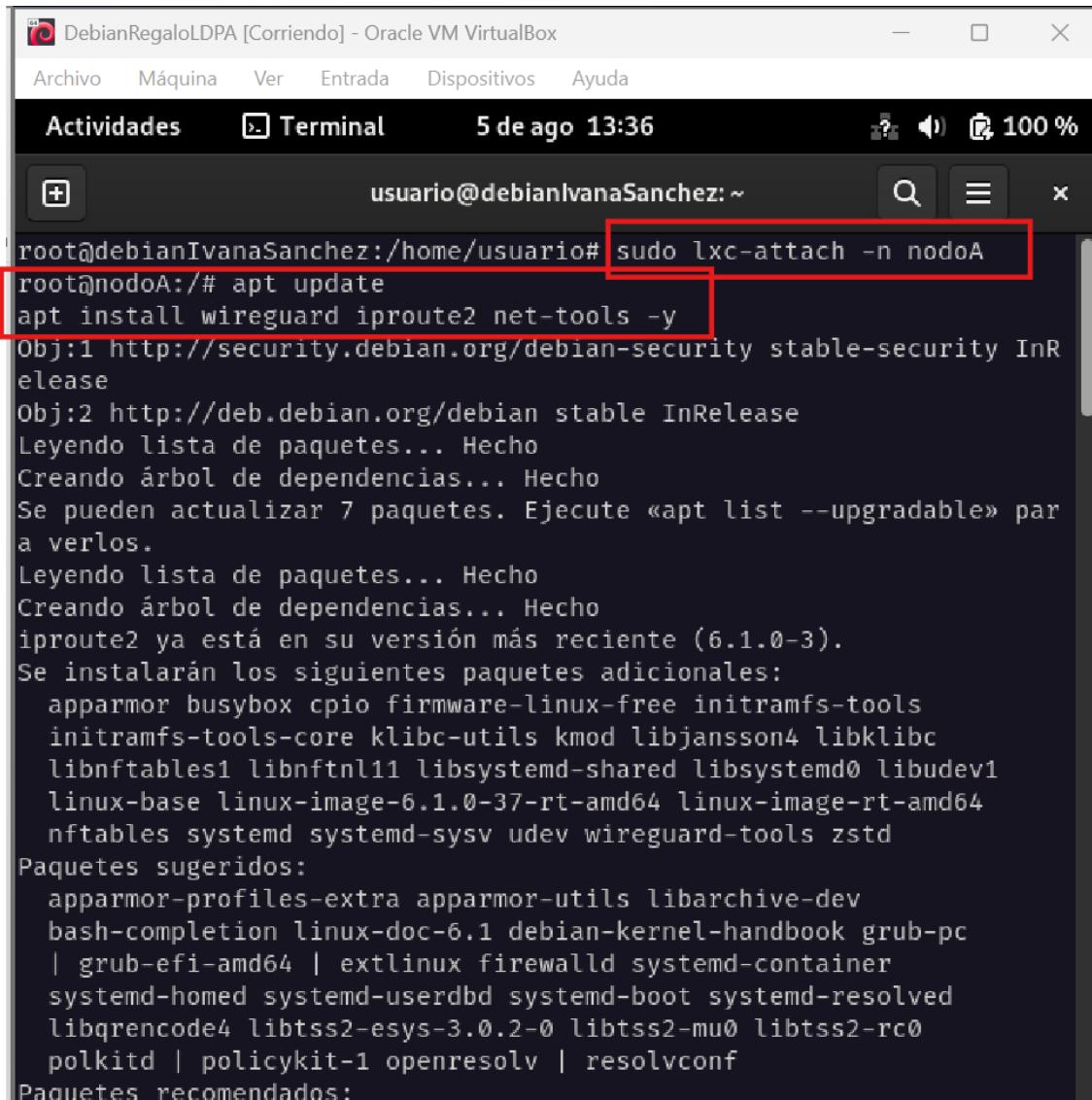
```
sudo lxc-attach -n nodoA
```



Y ya dentro del contendor

**apt update**

**apt install wireguard iproute2 net-tools -y**



The screenshot shows a terminal window titled "DebianRegaloLDPA [Corriendo] - Oracle VM VirtualBox". The window has tabs for "Actividades" and "Terminal", with "Terminal" selected. The date and time "5 de ago 13:36" are displayed. The terminal session starts with "root@debianIvanaSanchez:~". The user runs "sudo lxc-attach -n nodoA", then "apt update", and finally "apt install wireguard iproute2 net-tools -y". The output of the update command shows package versions and dependencies. The "apt install" command shows the installation of several packages including apparmor, busybox, and various kernel modules. A red box highlights the "apt update" and "apt install" lines.

```
root@debianIvanaSanchez:/home/usuario# sudo lxc-attach -n nodoA
root@nodoA:/# apt update
apt install wireguard iproute2 net-tools -y
Obj:1 http://security.debian.org/debian-security stable-security InRelease
Obj:2 http://deb.debian.org/debian stable InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Se pueden actualizar 7 paquetes. Ejecute «apt list --upgradable» para verlos.
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
iproute2 ya está en su versión más reciente (6.1.0-3).
Se instalarán los siguientes paquetes adicionales:
  apparmor busybox cpio firmware-linux-free initramfs-tools
  initramfs-tools-core klibc-utils kmod libjansson4 libklibc
  libnftables1 libnftnl11 libsystemd-shared libsystemd0 libudev1
  linux-base linux-image-6.1.0-37-rt-amd64 linux-image-rt-amd64
  nftables systemd systemd-sysv udev wireguard-tools zstd
Paquetes sugeridos:
  apparmor-profiles-extra apparmor-utils libarchive-dev
  bash-completion linux-doc-6.1 debian-kernel-handbook grub-pc
  | grub-efi-amd64 | extlinux firewalld systemd-container
  systemd-homed systemd-userdbd systemd-boot systemd-resolved
  libqrencode4 libtss2-esys-3.0.2-0 libtss2-mu0 libtss2-rc0
  polkitd | policykit-1 openresolv | resolvconf
Paquetes recomendados:
```

## PASO 6

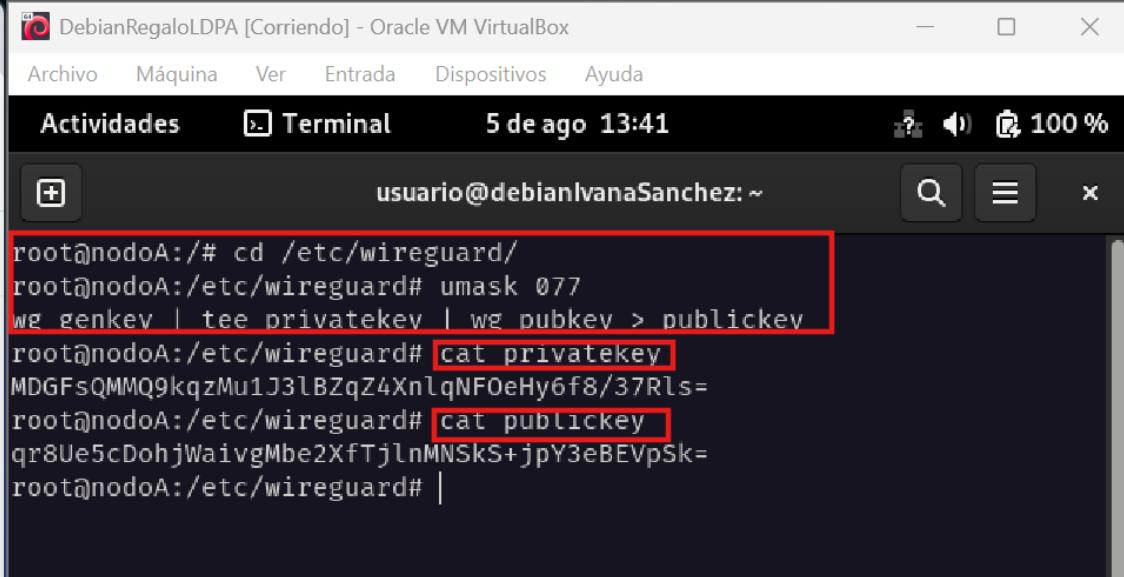
Generamos las claves WireGuard y las guardamos por ejemplo en el bloc de notas

**cd /etc/wireguard**

**umask 077**

**wg genkey | tee privatekey | wg pubkey > publickey**

Y para ver las claves generadas, simplemente lo hacemos con el comando con el comando **cat**



The screenshot shows a terminal window titled "DebianRegaloLDPA [Corriendo] - Oracle VM VirtualBox". The terminal is running under the user "usuario" on the host "debianIvanaSanchez". The window includes a menu bar with "Archivo", "Máquina", "Ver", "Entrada", "Dispositivos", and "Ayuda". The title bar also displays the date and time: "5 de ago 13:41". The terminal interface has tabs for "Actividades" and "Terminal", and a status bar showing battery level at 100%.

```
root@nodoA:/# cd /etc/wireguard/
root@nodoA:/etc/wireguard# umask 077
wg genkey | tee privatekey | wg pubkey > publickey
root@nodoA:/etc/wireguard# cat privatekey
MDGF5QMMQ9kqzMu1J3lBZqZ4XnlqNFOeHy6f8/37Rls=
root@nodoA:/etc/wireguard# cat publickey
qr8Ue5cDohjWaivgMbe2XfTjlnMNSkS+jpY3eBEVpSk=
root@nodoA:/etc/wireguard# |
```

```
NODOA
root@nodoA:/etc/wireguard# cat privatekey
MDGFsQMMQ9kqzMu1J3lBZqZ4XnlqNFOeHy6f8/37Rls=
root@nodoA:/etc/wireguard# cat publickey
qr8Ue5cDohjWaivgMbe2XfTjlnMNSkS+jpY3eBEVpSk=


NODOB
root@nodoB:/etc/wireguard# cat privatekey
QEx82m9tB4Hc/CiDf/w4o79DN+AeBfoinmzHaqTYQ1k=
root@nodoB:/etc/wireguard# cat publickey
5Zohcd04/dKTAKXvD0LKLjhJgU5YUezgxbZ54vEODxI=


NODOC
root@nodoC:/etc/wireguard# cat publickey
1RX53hxXLQ+/JBY0P157RKN4BGgVjRa4RrymBglFcCM=
root@nodoC:/etc/wireguard# cat privatekey
IMvXSsnX/ZuBx41u+zvzRPe1wonsBYe3tFkRmZZCw/Vc=


NODOD
root@nodoD:/etc/wireguard# cat publickey
o3z73XcKmzlmi3NV/GA9trSH7dl+Li7kdLkgjNTRhc=
root@nodoD:/etc/wireguard# cat privatekey
aNlcmtDT1250btUo2EZtWFalZ1+yXSU6WdczzcRKnKlc=


NODOE
root@nodoE:/etc/wireguard# cat publickey
DTqpPSmrS9R1og9AoHkz41EpwZ0TtQ77qwLAwLm4UFk=
root@nodoE:/etc/wireguard# cat privatekey
wD4cCMaSvVMqoS40n921grtjar4p2jb9hMTKPJwgzw4=
```

## PASO 7

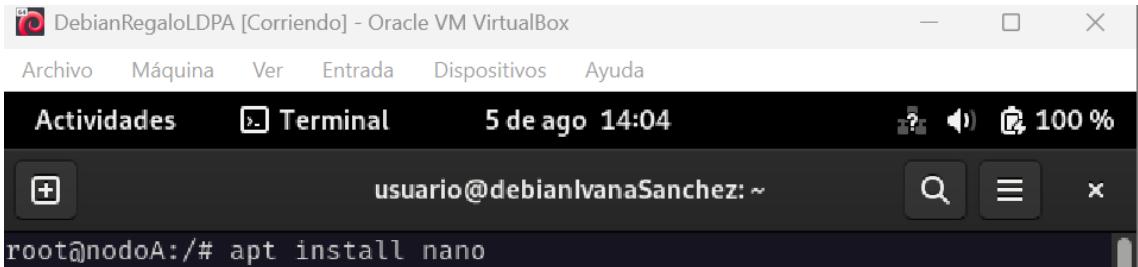
Creamos un archivo de configuración wg.conf en cada nodo

Para ello usamos las IPs y claves exactas del laboratorio del PDF

Entramos en el nodo correspondiente con:

***sudo lxc-attach -n nodoA***

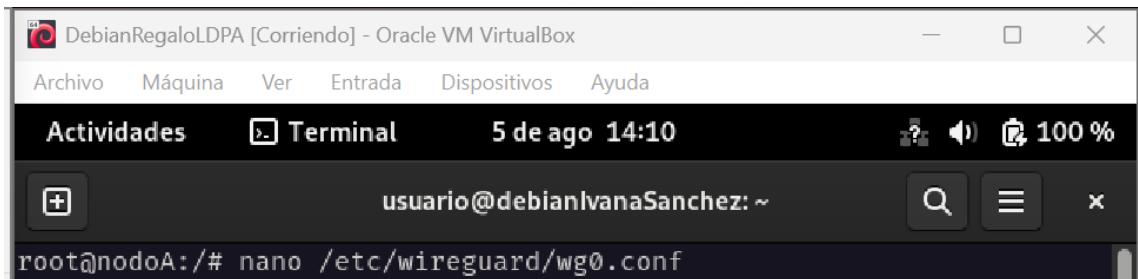
instalamos el editor de texto que deseemos. En mi caso voy a instalar nano.



```
root@nodoA:/# apt install nano
```

Y ejecutamos el comando:

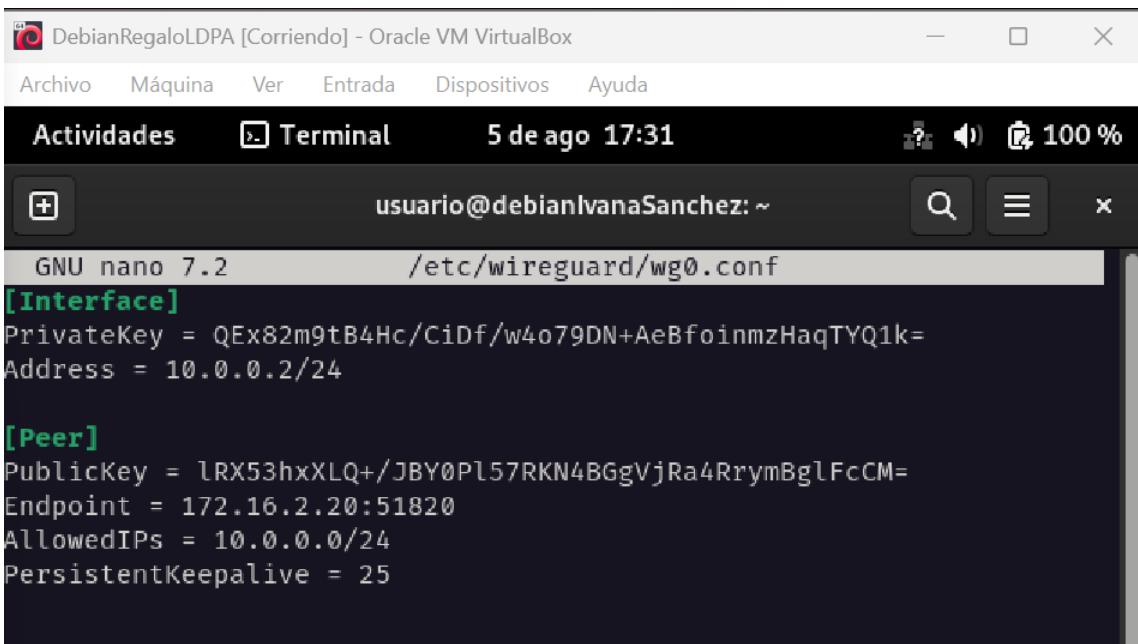
para configurar nuestro archivo según el PDF dado. Haremos lo mismo con todos los nodos.



```
root@nodoA:/# nano /etc/wireguard/wg0.conf
```

NODO A CON EL PEER C

NODO B con el PEER C



```
GNU nano 7.2          /etc/wireguard/wg0.conf
[Interface]
PrivateKey = QEx82m9tB4Hc/CiDf/w4o79DN+AeBfoinmzHaqTYQ1k=
Address = 10.0.0.2/24

[Peer]
PublicKey = lRX53hxXLQ+/JBY0Pl57RKN4BGgVjRa4RrymBglFcCM=
Endpoint = 172.16.2.20:51820
AllowedIPs = 10.0.0.0/24
PersistentKeepalive = 25
```

NODO C CON TODOS LOS PEER

The screenshot shows a terminal window titled "DebianRegaloLDPA [Corriendo] - Oracle VM VirtualBox". The terminal is displaying the contents of the file "/etc/wireguard/wg0.conf". The configuration includes sections for the interface, peer A, peer B, peer D, and peer E, each defining private keys, addresses, and allowed IP ranges.

```
GNU nano 7.2          /etc/wireguard/wg0.conf
[Interface]
PrivateKey = IMvXSnX/ZuBx41u+zvzRPe1wonsBYe3tFkRmzzCw/Vc=
Address = 10.0.0.3/24
ListenPort = 51820

[Peer] # peer A
PublicKey = qr8Ue5cDohjWaivgMbe2XfTjlnMNSkS+jpY3eBEVpSk=
AllowedIPs = 10.0.0.1/32

[Peer] # peer B
PublicKey = 5Zohcd04/dKTAKXvD0LKLjhJgU5YUezgxbZ54vEODxI=
AllowedIPs = 10.0.0.2/32

[Peer] # peer D
PublicKey = o3z73XcKmzlm2i3NV/GA9trSH7dL+Li7kdLkgjNTRhc=
AllowedIPs = 10.0.0.4/32

[Peer] # peer E
PublicKey = DTqP5SmrS9Rlog9AoHkz41EpwZ0TtQ77qwLAvLm4UFk=
AllowedIPs = 10.0.0.5/32
```

Y NODOS D Y E CON EL PEER C, TAMBIÉN

## PASO 8

Levantamos la interfaz de WireGuard. Para ello dentro de cada nodo ejecutaremos el comando

***wg-quick up wg0***

Comprobaremos con

***wg show***

The screenshot shows a terminal window titled "DebianRegaloLDPA [Corriendo] - Oracle VM VirtualBox". The terminal is running as root on the host machine. The user has run the command `sudo lxc-attach -n nodoA` to enter the container environment. Inside the container, they have run `wg-quick up wg0` to bring up the interface. They then ran `wg show` to view the configuration of the interface. The output shows the public key, private key (hidden), and listening port (42949). The terminal window has a red box highlighting the command `wg-quick up wg0` and the output of `wg show`.

```
root@debianIvanaSanchez:/home/usuario# sudo lxc-attach -n nodoA
root@nodoA:/# wg-quick up wg0
[#] ip link add wg0 type wireguard
[#] wg setconf wg0 /dev/fd/63
[#] ip -4 address add 10.0.0.1/24 dev wg0
[#] ip link set mtu 1420 up dev wg0
root@nodoA:/# wg show
interface: wg0
  public key: qr8Ue5cDohjWaivgMbe2XFTjlnMNSkS+jpY3eBEVpSk=
  private key: (hidden)
  listening port: 42949
root@nodoA:/# |
```

## PASO 9

**GENERAMOS LAS CLAVES PARA EL TUNEL WG1.** Para este segundo túnel, necesitas generar un nuevo par de claves (pública y privada) en cada nodo que participará: nodoA, nodoB, nodoD y nodoE.

Entramos en el contenedor: **sudo lxc-attach -n nodoA**

Y generamos la claves; **cd /etc/wireguard && wg genkey | tee wg1.key | wg pubkey > wg1.pub**

Hacemos un **cat** para ver las claves y guardarlas en nuestro bloc de notas

The screenshot shows a terminal window titled "DebianRegaloLDPA [Corriendo] - Oracle VM VirtualBox". The terminal is running as root on the host machine. The user has run the command `sudo lxc-attach -n nodoA` to enter the container environment. Inside the container, they have run the command `cd /etc/wireguard && wg genkey | tee wg1.key | wg pubkey > wg1.pub` to generate the keys. They then ran `ls` to list the files in the directory, which showed `privatekey`, `publickey`, `wg0.conf`, `wg1.key`, and `wg1.pub`. The terminal window has a red box highlighting the command `cd /etc/wireguard && wg genkey | tee wg1.key | wg pubkey > wg1.pub` and the output of `ls`.

```
root@debianIvanaSanchez:/home/usuario# sudo lxc-attach -n nodoA
root@nodoA:/# cd /etc/wireguard && wg genkey | tee wg1.key | wg pubkey > wg1.pub
root@nodoA:/etc/wireguard# ls
privatekey  publickey  wg0.conf  wg1.key  wg1.pub
root@nodoA:/etc/wireguard# cat wg1.key
oB1MSXu/u5S68bNHRix1Q6Tvx4sFtZGL6+Nc7uo8XVw=
root@nodoA:/etc/wireguard# cat wg1.pub
SeYEqIYPT1gvsYUrOyhD+1ZVCa5/vP4vI1YwE11RTAg=
root@nodoA:/etc/wireguard# |
```

```

CLAVES WG1

NODOA
root@nodoA:/etc/wireguard# cat wg1.key
oB1MSXu/u5S68bNHRix1Q6Tvx4sFtZGL6+Nc7uo8XVw=
root@nodoA:/etc/wireguard# cat wg1.pub
SeYEqIYPT1gvsYUrOyhD+1ZVCa5/vP4vI1YwE11RTAg=


NODOB
root@nodoB:/etc/wireguard# cat wg1.pub
x3PNVJadbY4tXVS09SgGEDwUyZx3AlM5Jm4ZgB+I+Eo=
root@nodoB:/etc/wireguard# cat wg1.key
WB1GAQ+OAXmyXKAhL4AwR9B6BpgxDzPBKOVu5317Hmg=


NODOD
root@nodoD:/etc/wireguard# cat wg1.pub
veYDrQB/05bslB22yBzz5cWCBFH90xzvdDRlnvHt2xM=
root@nodoD:/etc/wireguard# cat wg1.key
SG27T04bnSp9kiJFZtFYhb+cKRTVjLb8Cs/zHZCLFHM=


NODOE
root@nodoE:/etc/wireguard# cat wg1.pub
koIVNs7UkYAdOUKiNq0m+70ghdC4HinboupNqH46P1Q=
root@nodoE:/etc/wireguard# cat wg1.key
YINTug2isjnGyQwMUslgSM9Sk8x6GHnWspFMLidrgHo=

```

## PASO 10

Transferimos las claves públicas entre los contenedores.

- Entramos en nodoA: **sudo lxc-attach -n nodoA**
- Creamos el archivo de configuración wg1.conf: nano **/etc/wireguard/wg1.conf**

```

[Interface]
PrivateKey = [La clave privada de wg1 que generaste en el nodo A]
Address = 20.0.0.1/24
ListenPort = 51900

[Peer] # peer B
PublicKey = [La clave pública de wg1 del nodo B]
AllowedIPs = 20.0.0.2/32

[Peer] # peer D
PublicKey = [La clave pública de wg1 del nodo D]
AllowedIPs = 20.0.0.4/32

```

DebianRegaloLDPA [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Actividades Terminal 5 de ago 15:45

usuario@debianIvanaSanchez: ~

```
GNU nano 7.2 /etc/wireguard/wg1.conf *
```

[Interface]

```
PrivateKey = oB1MSXu/u5S68bNHRix1Q6Tvx4sFtZGL6+Nc7uo8XVw=
Address = 20.0.0.1/24
ListenPort = 51900
```

[Peer] # peer B

```
PublicKey = x3PNVJadbY4tXVSo9SgGEDwuYzx3AlM5Jm4ZgB+I+Eo=
AllowedIPs = 20.0.0.2/32
```

[Peer] # peer D

```
PublicKey = veYDrQB/05bslB22yBzz5cWCBFH9OxzvdDRlnvHt2xM=
AllowedIPs = 20.0.0.4/32
```

[Peer] # peer E

```
PublicKey = koIVNs7UkYAd0uKiNq0m+70ghdC4HinboupNqH46P1Q=
AllowedIPs = 20.0.0.5/32
```

^G Ayuda ^O Guardar ^W Buscar ^K Cortar ^T Ejecutar  
^X Salir ^R Leer fich.^V Reemplazar ^U Pegar ^J Justificar

NODOB

```
[Interface]
```

PrivateKey = [La clave privada de wg1 que generaste en el nodo B]  
Address = 20.0.0.2/24

[Peer] # peer A

```
PublicKey = [La clave pública de wg1 del nodo A]
Endpoint = 10.0.0.1:51900
AllowedIPs = 20.0.0.0/24
PersistentKeepalive = 25
```

## NODO D

```
[Interface]
PrivateKey = [La clave privada de wg1 que generaste en el nodo D]
Address = 20.0.0.4/24

[Peer] # peer A
PublicKey = [La clave pública de wg1 del nodo A]
Endpoint = 10.0.0.1:51900
AllowedIPs = 20.0.0.0/24
PersistentKeepalive = 25
```

## NODEE

```
[Interface]
PrivateKey = [La clave privada de wg1 que generaste en el nodo E]
Address = 20.0.0.5/24

[Peer] # peer A
PublicKey = [La clave pública de wg1 del nodo A]
Endpoint = 10.0.0.1:51900
AllowedIPs = 20.0.0.0/24
PersistentKeepalive = 25
```

## PASO 11

Una vez todos los archivos wg1.conf estén correctamente configurados, procederemos a levantar las interfaces de WireGuard.

Por cada uno de los nodos A, B, D y E ejecutaremos →

**wg-quick up wg1**

y verificaremos con

**wg show.**

```
root@debianIvanaSanchez:/home/usuario# sudo lxc-attach -n nodoB
root@nodoB:/# wg-quick up wg1
[#] ip link add wg1 type wireguard
[#] wg setconf wg1 /dev/fd/63
[#] ip -4 address add 20.0.0.2/24 dev wg1
[#] ip link set mtu 1420 up dev wg1
root@nodoB:/# wg show
interface: wg1
  public key: x3PNVJadbY4tXVSo9SgGEDwuYzx3AlM5Jm4ZgB+I+Eo=
  private key: (hidden)
  listening port: 60730

peer: SeYEqIYPT1gvsYUrOyhD+1ZVCa5/vP4vI1YwE11RTAg=
  endpoint: 10.0.0.1:51900
  allowed ips: 20.0.0.0/24
  transfer: 0 B received, 296 B sent
  persistent keepalive: every 25 seconds
root@nodoB:/# |
```

## PASO 12

### VERIFICACIONES FINALES

Teniendo levantado tanto wg0 como wg1, Entramos en cada nodo con → **sudo lxc-attach -n nodoA**.

Instalamos traceroute → **sudo apt install traceroute -y**

Ejecutamos

**sudo lxc-attach -n nodoA -- traceroute -i wg1 20.0.0.4**

**sudo lxc-attach -n nodoA -- traceroute -i wg0 10.0.0.5**

```

root@debianIvanaSanchez:/home/usuario# sudo lxc-attach -n nodoA -- w
g show wg0
interface: wg0
  public key: qr8Ue5cDohjWaivgMbe2XfTjlnMNSkS+jpY3eBEVpSk=
  private key: (hidden)
  listening port: 57515

peer: lRX53hxXLQ+/JBY0Pl57RKN4BGgVjRa4RrymBglFcCM=
  endpoint: 172.16.2.20:51820
  allowed ips: 10.0.0.0/24
  latest handshake: 2 minutes, 8 seconds ago
  transfer: 3.62 KiB received, 2.93 KiB sent
  persistent keepalive: every 25 seconds
root@debianIvanaSanchez:/home/usuario# sudo lxc-attach -n nodoA -- t
traceroute -i wg1 20.0.0.4
traceroute to 20.0.0.4 (20.0.0.4), 30 hops max, 60 byte packets
  1  20.0.0.4 (20.0.0.4)  0.937 ms  0.869 ms  0.834 ms
root@debianIvanaSanchez:/home/usuario# sudo lxc-attach -n nodoA -- t
traceroute -i wg0 10.0.0.5
traceroute to 10.0.0.5 (10.0.0.5), 30 hops max, 60 byte packets
  1  10.0.0.3 (10.0.0.3)  1.372 ms  0.931 ms  0.266 ms
  2  10.0.0.3 (10.0.0.3)  0.282 ms !H  0.146 ms !H  0.138 ms !H
root@debianIvanaSanchez:/home/usuario# |

```

La salida de `wg show wg0` ahora muestra la línea `latest handshake` (**`sudo lxc-attach -n nodoA -- wg show wg0`**), lo que significa que la conexión entre el nodoA y el nodoC a través del túnel wg0 está activa y funcionando.

La prueba de traceroute confirma que la topología de red funciona como se espera:

1. **traceroute -i wg1 20.0.0.4:** La salida muestra un solo salto directamente al nodoD (20.0.0.4). Esto prueba que el tráfico viaja directamente a través del túnel wg1, sin pasar por el nodoC, lo que garantiza la confidencialidad.
2. **traceroute -i wg0 10.0.0.5:** La salida muestra dos saltos. El primer salto es al nodoC (10.0.0.3), y luego el tráfico se enruta desde allí hasta el destino final. Esto demuestra que el tráfico en el túnel wg0 pasa por el nodo central, tal como se describe en el laboratorio.

**NOTA→ Si tiene problemas, apagaremos y levantaremos de nuevo todos los nodos tanto tan wg0 y wg1**

Baja y vuelve a subir el túnel wg0 en todos los nodos que lo tengan.

- `sudo lxc-attach -n nodoA -- wg-quick down wg0`

- `sudo lxc-attach -n nodoA -- wg-quick up wg0`
- Repite para los nodos B, C, D y E.

Baja y vuelve a subir el túnel wg1 en todos los nodos que lo tengan.

- `sudo lxc-attach -n nodoA -- wg-quick down wg1`
- `sudo lxc-attach -n nodoA -- wg-quick up wg1`
- Repite para los nodos B, D y E.