

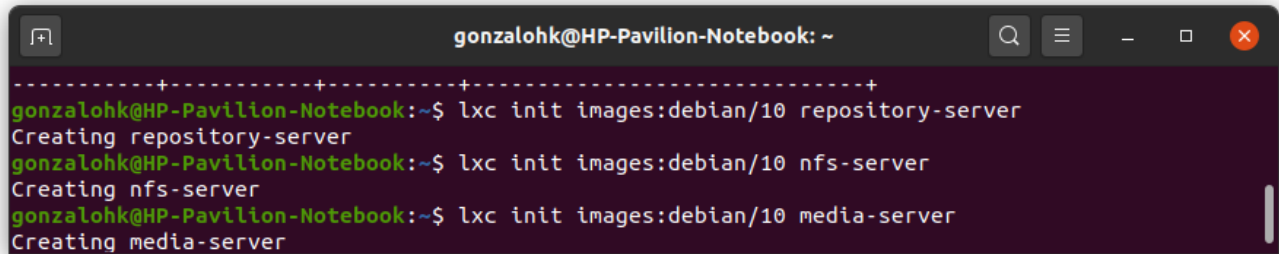
Actividad B1 – GONZALO OSCO HERNANDEZ

Realizar la creación de un contenedor en base a alguna imagen disponible en los repositorios

Para el ejemplo crearemos tres contenedores debian10 destinando a ser repositorios de código y de archivos.

```
lxc init images:debian/10 repository-server
lxc init images:debian/10 nfs-server
lxc init images:debian/10 media-server

lxc start media-server
lxc start nfs-server
lxc start repository-server
```



```
gonzalo@HP-Pavilion-Notebook: ~
-----+-----+-----+-----+
gonzalo@HP-Pavilion-Notebook:~$ lxc init images:debian/10 repository-server
Creating repository-server
gonzalo@HP-Pavilion-Notebook:~$ lxc init images:debian/10 nfs-server
Creating nfs-server
gonzalo@HP-Pavilion-Notebook:~$ lxc init images:debian/10 media-server
Creating media-server
```

Listamos los contenedores

```
lxc list
```



```
gonzalo@HP-Pavilion-Notebook: ~
-----+-----+-----+-----+
gonzalo@HP-Pavilion-Notebook:~$ lxc start media-server
gonzalo@HP-Pavilion-Notebook:~$ lxc start nfs-server
gonzalo@HP-Pavilion-Notebook:~$ lxc start repository-server
gonzalo@HP-Pavilion-Notebook:~$ lxc list
-----+-----+-----+-----+
| NAME           | STATE   | IPV4          | IPV6          | TYPE   | SNAPSHOTS |
-----+-----+-----+-----+
| media-server   | RUNNING | 10.19.238.102 (eth0) | fd42:79fd:5a0d:6889:216:3eff:fe19:44bb (eth0) | CONTAINER | 0          |
-----+-----+-----+-----+
| nfs-server     | RUNNING | 10.19.238.246 (eth0) | fd42:79fd:5a0d:6889:216:3eff:fe6a:62fe (eth0) | CONTAINER | 0          |
-----+-----+-----+-----+
| repository-server | RUNNING | 10.19.238.248 (eth0) | fd42:79fd:5a0d:6889:216:3eff:fe33:5c11 (eth0) | CONTAINER | 0          |
-----+-----+-----+-----+
gonzalo@HP-Pavilion-Notebook:~$
```

nfs-server

Ingresamos al contenedor *nfs-server*

```
lxc exec nfs-server bash
```

Instalamos **NFS** con los siguientes comandos.

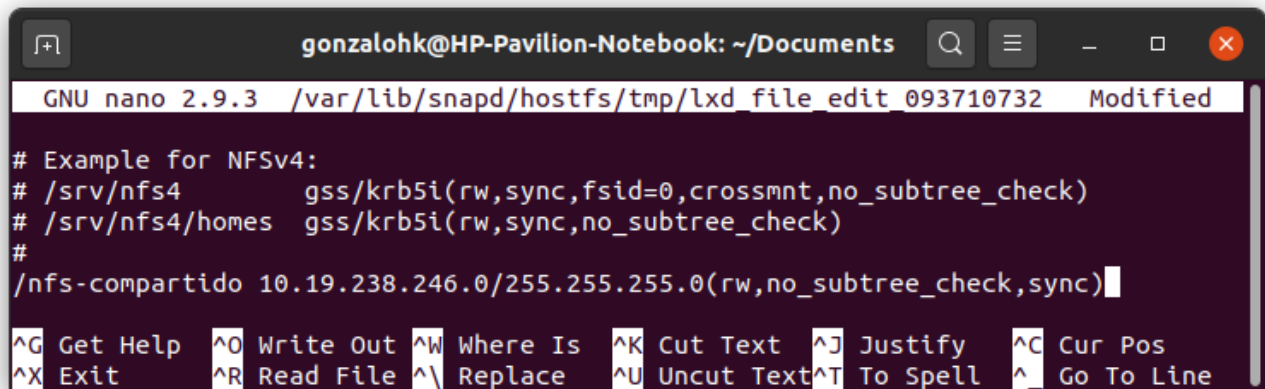
```
apt-get install nfs-kernel-server nfs-common
```

Creamos una carpeta compartida en la raíz con los permisos pertinentes.

```
mkdir /nfs-compartido
```

Volvemos a la máquina anfitrión, para editar el archivo de configuración del servidor nfs de forma directa.

```
export EDITOR=nano  
lxc file edit nfs-server/etc/export
```



```
gonzalohk@HP-Pavilion-Notebook: ~/Documents  
GNU nano 2.9.3 /var/lib/snapd/hostfs/tmp/lxd file edit_093710732 Modified  
# Example for NFSv4:  
# /srv/nfs4      gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)  
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)  
#  
/nfs-compartido 10.19.238.246.0/255.255.255.0(rw,no_subtree_check,sync)  
  
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos  
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

Adicionamos la siguiente instrucción para que se puede acceder al archivo desde cualquier equipo en mi red.

```
/nfs-compartido 10.19.238.246.0/255.255.255.0(rw,no_subtree_check,sync)
```

Ya desde una maquina cliente podemos montar la carpeta compartida, pero antes instalamos

```
apt-get install nfs-common
```

Creamos y montamos la carpeta compartida en un carpeta de mi maquina cliente

```
mkdir nfc-cliente
sudo mount 10.19.238.246:/nfs-compartido /home/gonzalohk/Documents/nfc-cliente
```

repository-server

Ingresamos al contenedor *repository-server*

```
lxc exec repository-server bash
```

Instalamos **GIT** con los siguientes comandos.

```
apt update
apt install git
```

Creamos el usuario GIT

```
adduser --disabled-password git
```

Creamos la carpeta *.ssh* al igual que el archivo *authorized_keys* y adicionamos permisos

```
cd /home/git/
mkdir ~/.ssh && chmod 700 ~/.ssh
touch ~/.ssh/authorized_keys && chmod 0600 ~/.ssh/authorized_keys
```

Ahora tenemos que adicionar a los usuarios que tendran acceso a mi repositorio.

Para ello copiamos la llave public del cliente a mi servidor repositorio.

```
lxc file push id_rsa.pub repository-server/home/git/
```

Ahora añadimos dicha llave a mi archivo *authorized_keys*

```
cat id_rsa.pub >> $HOME/.ssh/authorized_keys
```

Creamos un proyecto de prueba, que será en un inicio un proyecto vacío.

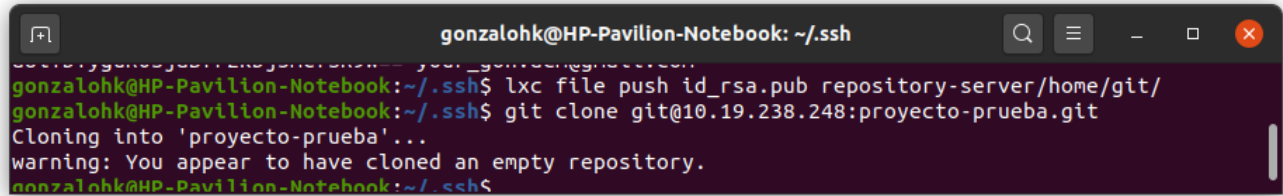
```
mkdir proyecto-prueba.git
cd proyecto-prueba.git && git init --bare && cd ..
chown -R git.git proyecto-prueba.git/
```

Ahora para probar, clonaremos el proyecto pero antes se deben establecer el nombre y correo del cliente.

```
git config --global user.name "gonzalo"
git config --global user.email "goh.acm@gmail.com"
```

Clonamos

```
git clone git@10.19.238.248:proyecto-prueba.git
```

A terminal window titled 'gonzalohek@HP-Pavilion-Notebook: ~/.ssh' showing the execution of a git clone command. The prompt is 'gonzalohek@HP-Pavilion-Notebook:~/.ssh\$'. The command entered is 'git clone git@10.19.238.248:proyecto-prueba.git'. The output shows 'Cloning into 'proyecto-prueba'...' followed by a warning: 'warning: You appear to have cloned an empty repository.' The prompt returns to 'gonzalohek@HP-Pavilion-Notebook:~/.ssh\$'.

Adicionalmente instalaremos **SVN**

Inicialmente instalamos apache.

```
apt-get install -y apache2 apache2-utils
```

Luego, SVN considerando las dependencias necesarias.

```
apt-get install -y subversion subversion-tools libapache2-mod-svn
```

Creamos el repositorio llamado svn-repo-test.

```
svnadmin create /var/lib/svn/svn-repo-test  
chown -R www-data:www-data /var/lib/svn/svn-repo-test
```

También editamos la configuración del apache.

```
nano /etc/apache2/mods-enabled/dav_svn.conf
```

Adicionando a dav_svn.conf

```
<Location /svn>  
  DAV svn  
  SVNParentPath /var/lib/svn  
  AuthType Basic  
  AuthName "Subversion Repository"  
  AuthUserFile /etc/apache2/dav_svn.passwd  
  Require valid-user  
</Location>
```

Finalmente creamos un usuario con su respectivo password y reiniciamos

```
htpasswd -cm /etc/apache2/dav_svn.passwd gonzalohek  
systemctl restart apache2
```

Ahora podemos probar si tenemos acceso desde cualquier equipo de nuestra red.

