

# General

Páginas que se aplican a todos los sistemas/equipos

- [Tunel SSH para equipos SGI](#)

# Tunel SSH para equipos SGI

## Linux

Instalamos el programa connect-proxy

**sudo apt install connect-proxy**

Si la clave pública y privada no ha sido creada hacemos:

1. Nos situamos en la carpeta Home del usuario con: `cd`
2. Una vez allí generamos la clave pulsando: `ssh-keygen`
3. Luego ingresando a `.ssh` con: `cd .ssh`
4. Copiamos la clave pública recientemente generada, es el archivo que termina con *id-rsa.pub*. Dicha clave se puede compartir para poder acceder a los servidores. Dicha clave dentro del servidor destino al cual nos queremos conectar se agrega dentro del archivo *authorized\_keys*
5. Crear archivo **config** en el directorio `/home/user/.ssh` con el siguiente bloque

```
Host jacinto.unc.edu.ar
  Hostname jacinto.unc.edu.ar
  User <usuario>
  ProxyCommand none

host *.psi.unc.edu.ar
  User root
  KexAlgorithms +diffie-hellman-group1-sha1
  HostKeyAlgorithms +ssh-dss
  PubkeyAcceptedKeyTypes +ssh-rsa
  ProxyCommand connect -S localhost %h %p
```

```
host *
ServerAliveInterval 120
HashKnownHosts no
ForwardAgent yes
StrictHostKeyChecking no
ConnectTimeout 30
Protocol 2,1
Compression yes
```

6. Dentro del mismo directorio (.ssh) creamos un archivo ejecutable llamado **proxy** con el siguiente bloque, y le damos permisos de ejecución con “**chmod +x proxy**”

para crearlo **vi ~/.ssh/proxy**

```
#!/bin/bash

if [ $(ssh-add -L|grep id_rsa|wc -l) -eq 0 ]; then
    ssh-add ~/.ssh/id_rsa
fi

if [ $(ps aux|grep "ssh -N -Dlocalhost:1080 jacinto.unc.edu.ar"|wc -l) -gt 1 ]; then
    kill -9 $(ps aux|grep 1080|grep "localhost"|sed 's/ */ /g'|cut -d" " -f2)
fi

ssh -N -Dlocalhost:1080 jacinto.unc.edu.ar &
pid=$!
echo "$pid" >~/proxy.pid.$HOSTNAME
```

7. Una vez realizado esto, para levantar el túnel ssh, estando en el directorio home, ejecutamos directamente el archivo proxy.sh

8. para montar carpetas remotas, crear un archivo unidades.sh, adaptando el siguiente bloque de código:

```
#!/bin/bash
```

#oidmap=user mapea los permisos del servidor a los del usuario especificado con las opciones uid y gid

#uid=1001,gid=1001 son los id del usuario actual. se puede verificar con el comando "id" en consola

```
sshfs -oidmap=user,uid=1001,gid=1001 root@<servidor>:<carpeta remota> /home/<usuario local>/<carpeta local>
```

#ejemplo: sshfs -oidmap=user,uid=1001,gid=1001 root@spgi-sanaviron-api-dev.psi.unc.edu.ar:/srv/desarrollo/  
/home/usuario/Sanaviron-psi

## Windows

<TODO>

## Linux - config ssh avanzado

```
HashKnownHosts no  
ForwardAgent yes  
StrictHostKeyChecking no  
ForwardX11 yes  
ServerAliveInterval 15  
ForwardX11Trusted yes  
GatewayPorts yes  
ConnectTimeout 15
```

### #Configuración para entrar a un/a servido/vm específico/a

```
Host <nombre-de-la-vm>  
  CheckHostIP no  
  HostName <nombre-de-la-vm-con-dominio>  
  User <nombre-de-usuario-que-quiero-usar-para-esta-vm>  
  ProxyCommand none
```

### #Ejemplo de configuración para entrar a un/a servido/vm específico/a

```
Host jacinto jacinto.psi.unc.edu.ar  
  CheckHostIP no
```

```
HostName jacinto.psi.unc.edu.ar
User gmunoz
ProxyCommand none
```

### #Configuración para entrar a un conjunto de servidores/vms sin usar proxy

```
Host *.srv.unc.edu.ar *.unc.edu.ar <nombre-de-una-vm-en-particular> <ip-de-un-docker-local> <ip-de-una-vm-
de-virtualbox-local-o-remota>
User <nombre-de-usuario-que-quiero-usar-todos-lo-que-agarre-el-filtro-de-arriba>
ProxyCommand none
```

### #Configuración para entrar a un conjunto de servidores/vms utilizando proxy

```
Host *.psi.unc.edu.ar
User <nombre-de-usuario-que-quiero-usar-todos-lo-que-agarre-el-filtro-de-arriba>
ProxyCommand connect -S localhost:1080 %h %p
```

### #Configuración para cuándo el host que me quiero conectar no están en ninguno de las configuraciones anteriores usando proxy

```
Host *
User <nombre-de-usuario-que-quiero-usar-todos-lo-que-no-haya-especificado-antes>
ProxyCommand connect -S localhost:1080 %h %p
```