

Control de Usuarios, Permisos y Grupos en Entornos Linux Multiusuario

1. Creación y organización de usuarios.

- **Crear los usuarios:** Ana, Carlos y Elena.
- **Crear los grupos:** webdev, infra y docs.
- **Asignar:** Ana al grupo webdev, Carlos a infra y Elena a docs.
- **Establecer contraseñas seguras para cada usuario.**

2. Estructura de directorios y control de accesos.

- **Crear las carpetas:** /grupos/web, /grupos/infra, /grupos/docs
- **Asignar propietario al grupo correspondiente y cambiar permisos:** Solo los miembros del grupo pueden leer y escribir en su carpeta y otros usuarios no deben tener acceso
- **Aplicar chmod y chown correctamente para:** Establecer permisos 770 y activar setgid para que los archivos nuevos hereden el grupo.

3. Configuración avanzada de permisos y restricciones.

- **Crear en /grupos/docs:** Un archivo llamado plan.txt
- **Permitir que solo Elena pueda modificarlo y que Ana y Carlos puedan leerlo sin editarlo.**
- **Configurar un grupo compartido llamado lectura e incluirlos a los tres.**
- **Usar ACLs para asignar permisos finos.**

4. Buenas prácticas y seguridad básica.

- **Establecer política de caducidad de contraseñas:** De 60 días para todos los usuarios.
 - **Bloquear el acceso SSH al usuario Elena.**
 - **Crear un alias de shell en /etc/skel.bashrc:** Para que los nuevos usuarios vean un mensaje de bienvenida personalizado al iniciar sesión.
-

1. Creación y organización de usuarios.

Para crear los usuarios Ana, Carlos y elena introducimos los siguientes comandos:

```
sudo adduser ana  
sudo adduser carlos  
sudo adduser elena
```

Para crear los grupos lo haremos con los siguientes comandos:

```
sudo groupadd webdev  
sudo groupadd infra  
sudo groupadd docs
```

Para asignar cada usuario al grupo correspondiente introducimos los siguientes comandos:

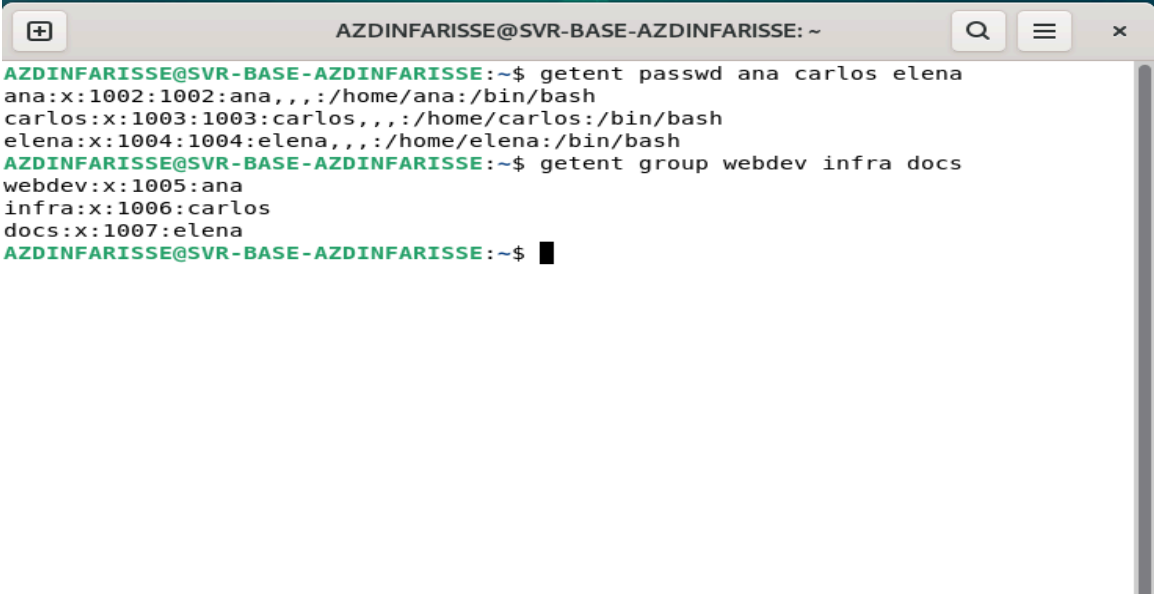
```
sudo usermod -aG webdev ana  
sudo usermod -aG infra carlos  
sudo usermod -aG docs elena
```

Para establecer contraseñas seguras a cada usuario lo hacemos con los comandos:

```
sudo passwd ana  
sudo passwd carlos  
sudo passwd elena
```

Ahora para comprobar que los usuarios y los grupos se han creado correctamente lo haremos con los comandos:

```
getent passwd ana carlos elena  
getent group webdev infra docs
```

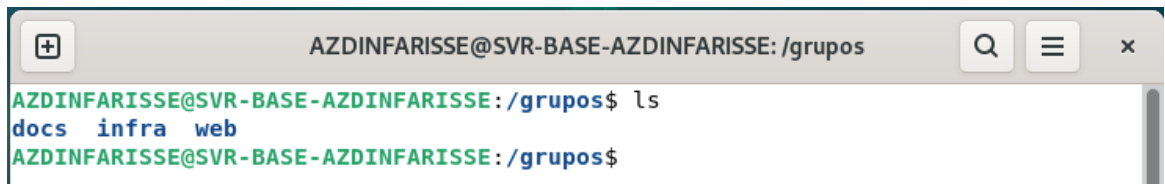


```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: ~  
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ getent passwd ana carlos elena  
ana:x:1002:1002:ana,,,:/home/ana:/bin/bash  
carlos:x:1003:1003:carlos,,,:/home/carlos:/bin/bash  
elena:x:1004:1004:elena,,,:/home/elena:/bin/bash  
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ getent group webdev infra docs  
webdev:x:1005:ana  
infra:x:1006:carlos  
docs:x:1007:elena  
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$
```

2. Estructura de directorios y control de acceso.

Para crear las carpetas lo hacemos con los comandos:

```
sudo mkdir -p /grupos/web
sudo mkdir -p /grupos/infra
sudo mkdir -p /grupos/docs
```

A terminal window titled 'AZDINFARISSE@SVR-BASE-AZDINFARISSE: /grupos'. The user runs 'ls' and the output shows 'docs infra web'.

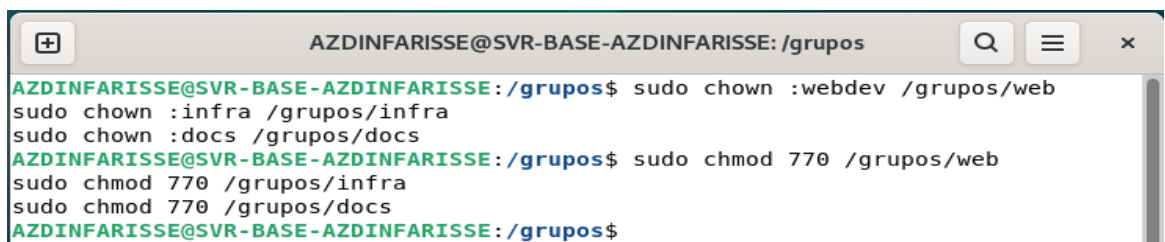
```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /grupos$ ls
docs  infra  web
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /grupos$
```

Para asignar propiedad del grupo a cada carpeta usamos:

```
sudo chown :webdev /grupos/web
sudo chown :infra /grupos/infra
sudo chown :docs /grupos/docs
```

Ahora para establecer los permisos 770 introducimos los comandos:

```
sudo chmod 770 /grupos/web
sudo chmod 770 /grupos/infra
sudo chmod 770 /grupos/docs
```

A terminal window titled 'AZDINFARISSE@SVR-BASE-AZDINFARISSE: /grupos'. The user runs 'sudo chown :webdev /grupos/web', 'sudo chown :infra /grupos/infra', and 'sudo chown :docs /grupos/docs'. Then they run 'sudo chmod 770 /grupos/web', 'sudo chmod 770 /grupos/infra', and 'sudo chmod 770 /grupos/docs'.

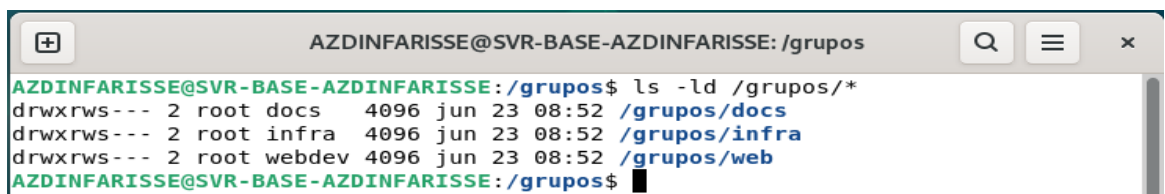
```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /grupos$ sudo chown :webdev /grupos/web
sudo chown :infra /grupos/infra
sudo chown :docs /grupos/docs
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /grupos$ sudo chmod 770 /grupos/web
sudo chmod 770 /grupos/infra
sudo chmod 770 /grupos/docs
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /grupos$
```

Para activar setgid utilizamos:

```
sudo chmod g+s /grupos/web
sudo chmod g+s /grupos/infra
sudo chmod g+s /grupos/docs
```

Y verificamos con el comando:

```
ls -ld /grupos/*
```

A terminal window titled 'AZDINFARISSE@SVR-BASE-AZDINFARISSE: /grupos'. The user runs 'ls -ld /grupos/*' and the output shows the permissions for each directory, with 'drwxrws---' indicating that setgid is active.

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /grupos$ ls -ld /grupos/*
drwxrws--- 2 root docs 4096 jun 23 08:52 /grupos/docs
drwxrws--- 2 root infra 4096 jun 23 08:52 /grupos/infra
drwxrws--- 2 root webdev 4096 jun 23 08:52 /grupos/web
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /grupos$
```

La s al final de rws nos indica que el setgid está activo.

3. Configuración avanzada de permisos y restricciones.

Para crear el archivo plan.txt en /grupos/docs, utilizamos el comando:

sudo touch /grupos/docs/plan.txt

Para crear un grupo compartido llamado lectura lo hacemos con el comando:

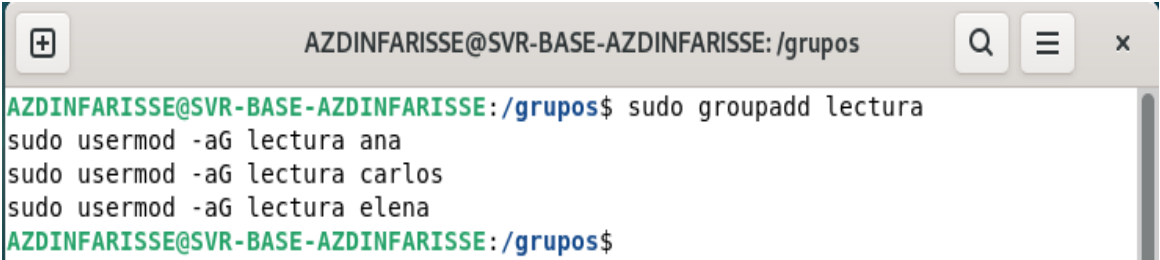
sudo groupadd lectura

Y para incluir a los tres usuarios lo hacemos con los comandos:

sudo usermod -aG lectura ana

sudo usermod -aG lectura carlos

sudo usermod -aG lectura elena



```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /grupos
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /grupos$ sudo groupadd lectura
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /grupos$ sudo usermod -aG lectura ana
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /grupos$ sudo usermod -aG lectura carlos
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /grupos$ sudo usermod -aG lectura elena
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /grupos$
```

Ahora para asignar los permisos finos sobre el archivo plan.txt dando permisos a elena para que pueda modificarlo, pero dejando que ana y carlos solo puedan leerlo, lo haremos de la siguiente manera:

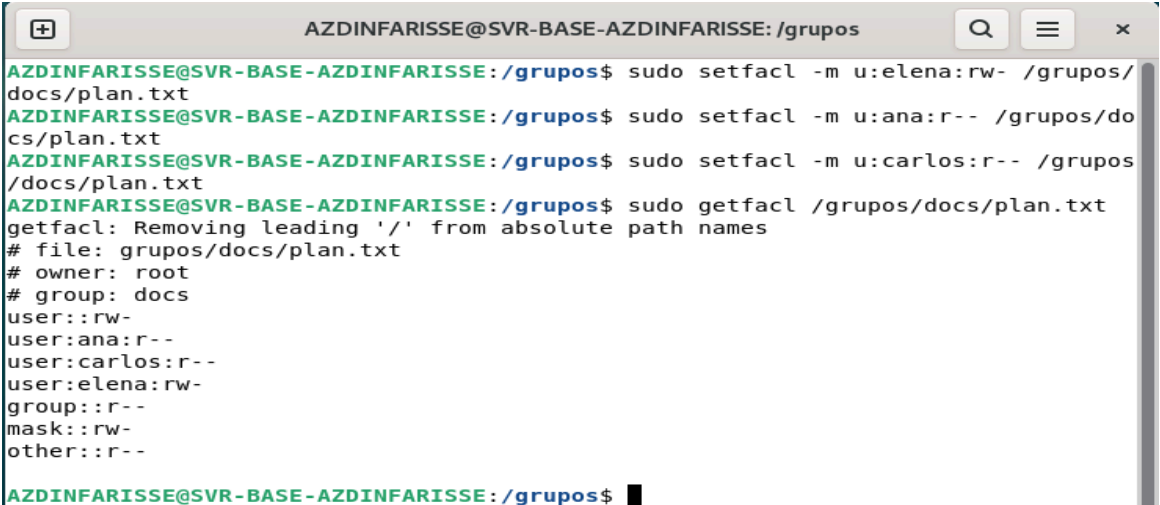
Para elena: **sudo setfacl -m u:elena:rw- /grupos/docs/plan.txt**

Para ana y carlos: **sudo setfacl -m u:ana:r-- /grupos/docs/plan.txt**

sudo setfacl -m u:carlos:r-- /grupos/docs/plan.txt

Y para verificar los permisos lo haremos con el comando:

sudo getfacl /grupos/docs/plan.txt



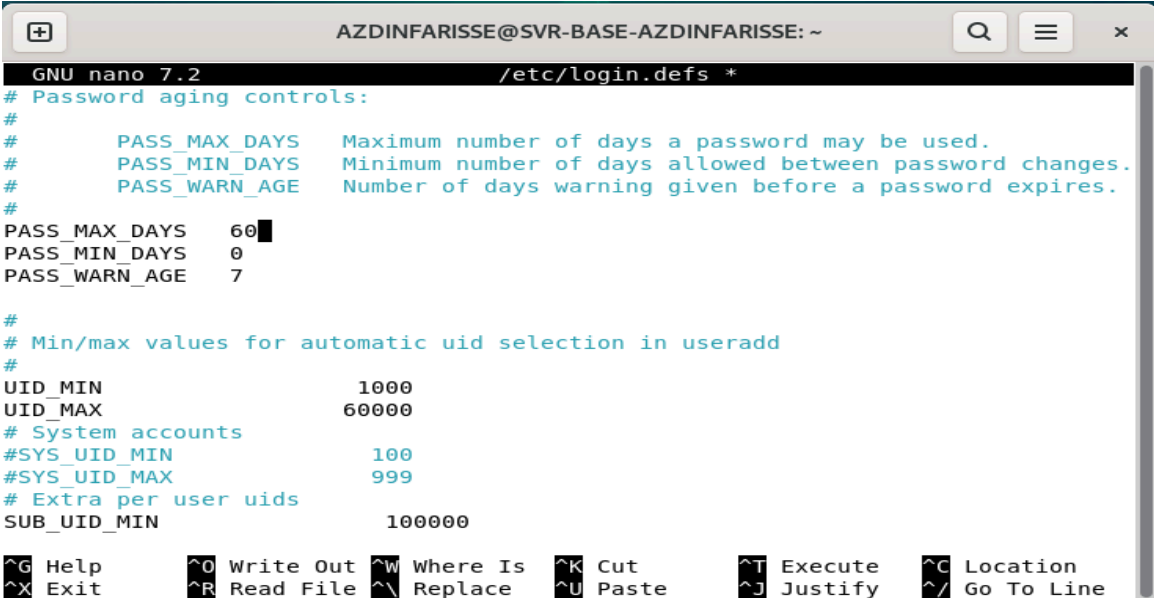
```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /grupos
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /grupos$ sudo setfacl -m u:elena:rw- /grupos/docs/plan.txt
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /grupos$ sudo setfacl -m u:ana:r-- /grupos/docs/plan.txt
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /grupos$ sudo setfacl -m u:carlos:r-- /grupos/docs/plan.txt
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /grupos$ sudo getfacl /grupos/docs/plan.txt
getfacl: Removing leading '/' from absolute path names
# file: grupos/docs/plan.txt
# owner: root
# group: docs
user::rw-
user:ana:r--
user:carlos:r--
user:elena:rw-
group::r--
mask::rw-
other::r--
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /grupos$
```

4. Buenas prácticas y seguridad básica.

Para establecer una política de caducidad de contraseñas de 60 días para todos los usuarios hay que editar el archivo **login.defs** con el comando:

sudo nano /etc/login.defs

Una vez abierto el archivo editamos la línea **PASS_MAX_DAYS** a 60.



```
GNU nano 7.2 /etc/login.defs *
# Password aging controls:
#
#      PASS_MAX_DAYS   Maximum number of days a password may be used.
#      PASS_MIN_DAYS   Minimum number of days allowed between password changes.
#      PASS_WARN_AGE   Number of days warning given before a password expires.
#
PASS_MAX_DAYS   60
PASS_MIN_DAYS   0
PASS_WARN_AGE   7

#
# Min/max values for automatic uid selection in useradd
#
UID_MIN          1000
UID_MAX          60000
# System accounts
#SYS_UID_MIN      100
#SYS_UID_MAX      999
# Extra per user uids
SUB_UID_MIN      100000

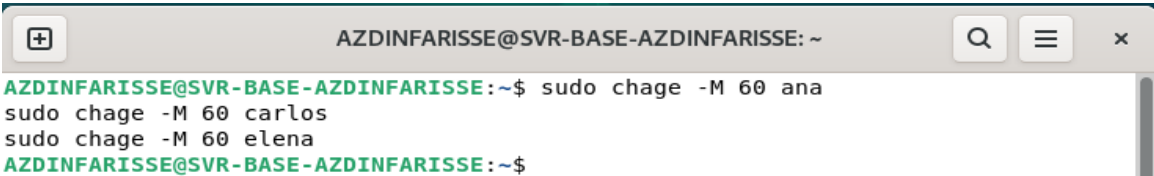
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^N Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

Ahora hay que editar los usuarios existentes para que su contraseña también caduque a los 60 días, ya que el archivo anterior es para los usuarios que se creen nuevos. Para modificar la política de estos usuarios existentes introducimos los comandos:

sudo chage -M 60 ana

sudo chage -M 60 carlos

sudo chage -M 60 elena



```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: ~$ sudo chage -M 60 ana
sudo chage -M 60 carlos
sudo chage -M 60 elena
AZDINFARISSE@SVR-BASE-AZDINFARISSE: ~$
```

Lo siguiente es bloquear el acceso SSH al usuario elena. Para ello editamos el archivo de configuración SSH con el comando:

sudo nano /etc/ssh/sshd_config

Y le introducimos la siguiente línea:

DenyUsers elena

Ahora reiniciamos el servicio SSH para que el cambio haga efecto con el comando:

sudo systemctl restart ssh

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: ~
GNU nano 7.2 /etc/ssh/sshd_config

#VersionAddendum none

# no default banner path
#Banner none

# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

# override default of no subsystems
Subsystem sftp /usr/lib/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#    X11Forwarding no
#    AllowTcpForwarding no
#    PermitTTY no
#    ForceCommand cvs server
DenyUsers elena

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

Y comprobamos desde otra máquina para verificar que el usuario elena no tiene permisos SSH con el comando:

ssh -p 2222 elena@192.168.1.20

```
azdiin@azdiin-UBUNTU: ~
azdiin@azdiin-UBUNTU:~$ ssh -p 2222 elena@192.168.1.20
elena@192.168.1.20's password:
Permission denied, please try again.
elena@192.168.1.20's password:
```

Ahora para crear un alias de shell en **/etc/skel/.bashrc** con un mensaje de bienvenida para los nuevos usuarios al iniciar sesión, editamos el archivo **.bashrc** con el con el comando:

sudo nano /etc/skel/.bashrc

Y le añadimos al final del archivo la línea

echo "¡Bienvenido al servidor de Azdin Farisse! Esto son pruebas para las prácticas en CodeArts"

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: ~
GNU nano 7.2 /etc/skel/.bashrc *
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi
echo "¡Bienvenido al servidor de Azdin Farisse! Esto son pruebas para las prácticas en CodeArts"
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location  M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line M-E Redo
```