

# Windows Server and Linux

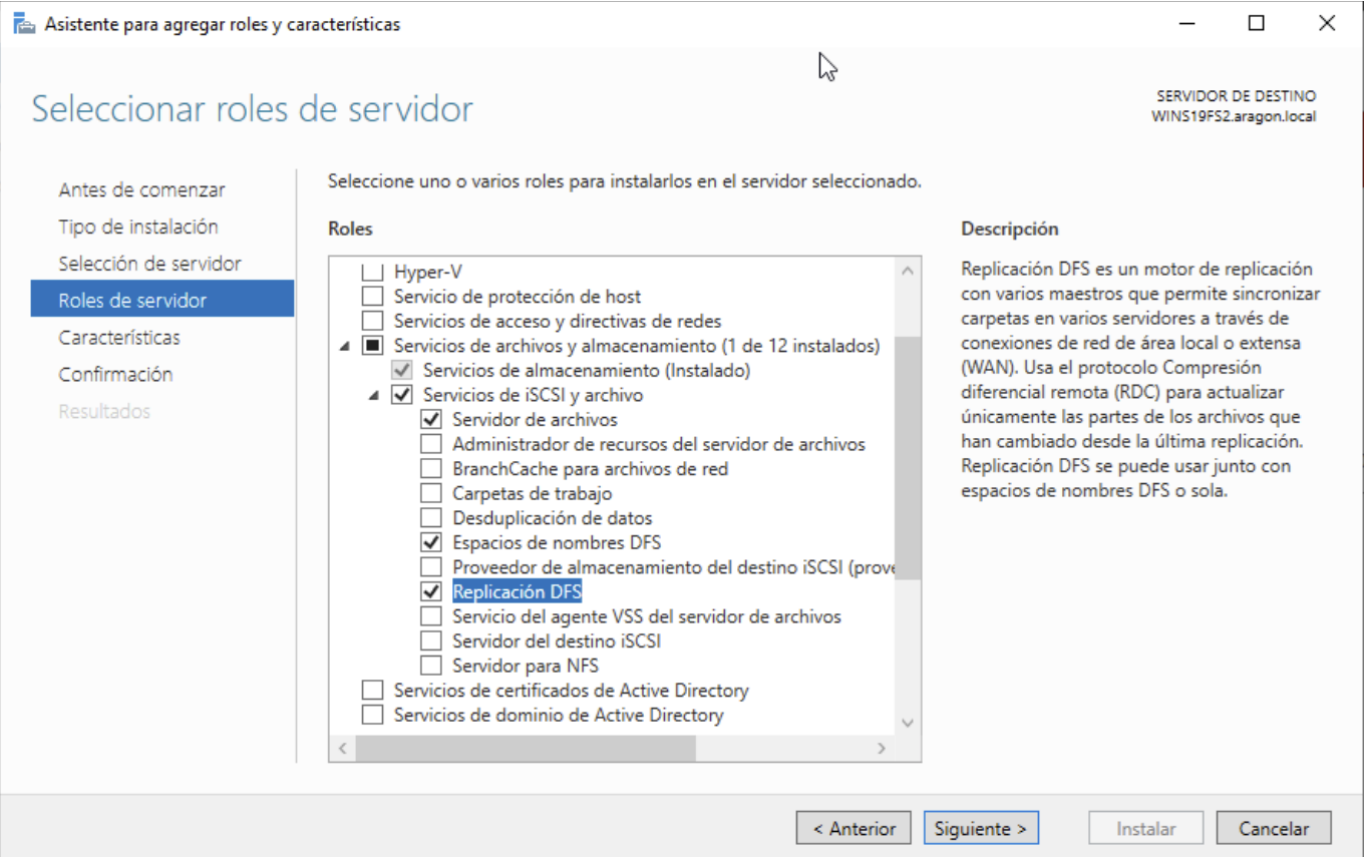
Documentación SpainSkills22  
Modalidad 39 Administración de sistemas en red

***Daniel Ubalde Laguia***

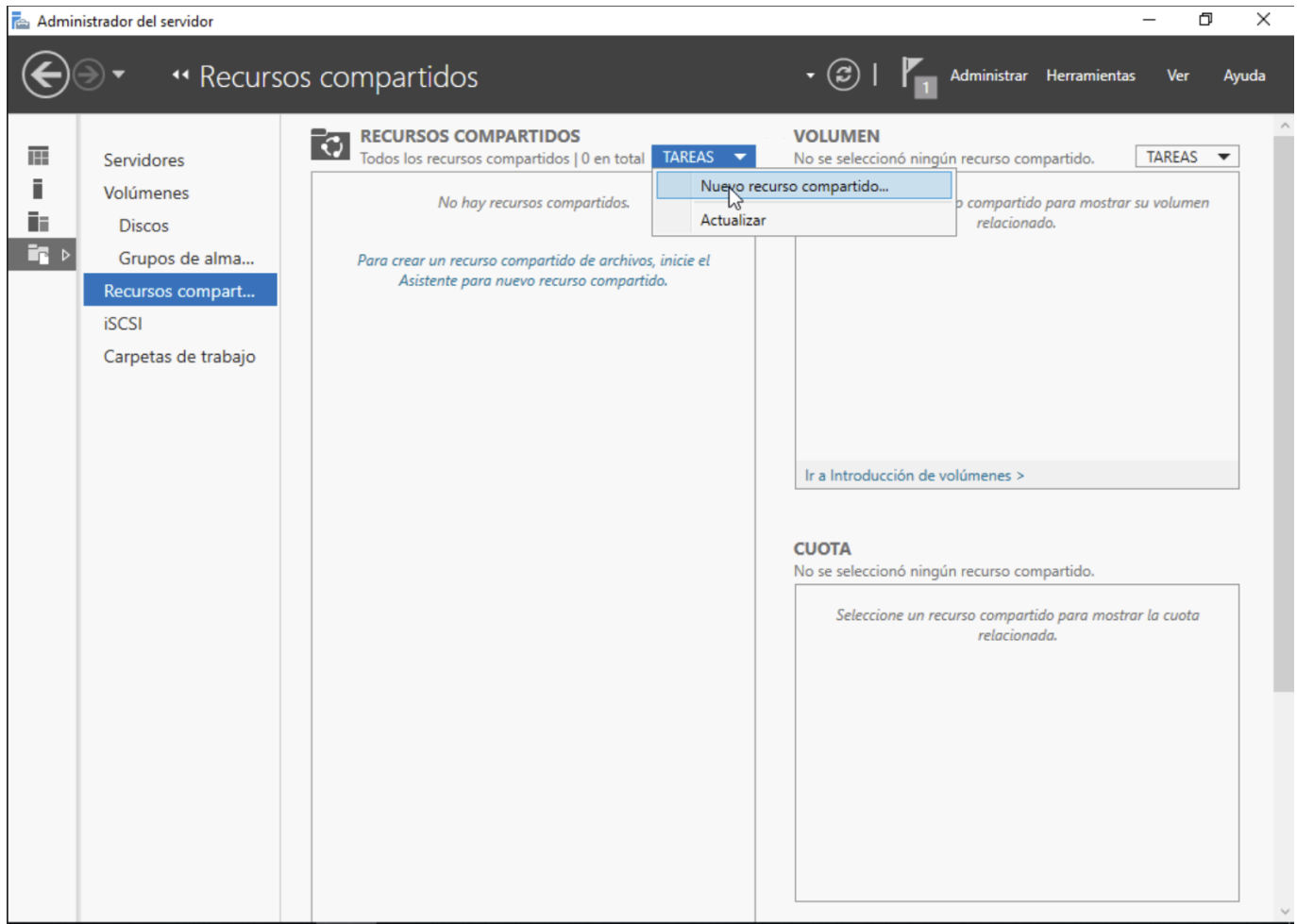
# DFS Windows Server

## Installation

First we have to install DFS rol



## Create shared folders



## New shared folder

- Select the first option (SMB Share - Quick)
- Select the path of the folder
- Choose share name
- Next enable access-based enumeration

## Permissions step

- Disable inheritance
- Remove users permissions
- Add the users we want to access to the shared folder

## Create Namespace

To create a namespace for sharing folders we need to go to the DFS admin

- Pick the name of the server

Asistente para crear nuevo espacio de nombres

**Servidor de espacio de nombres**

**Pasos:**

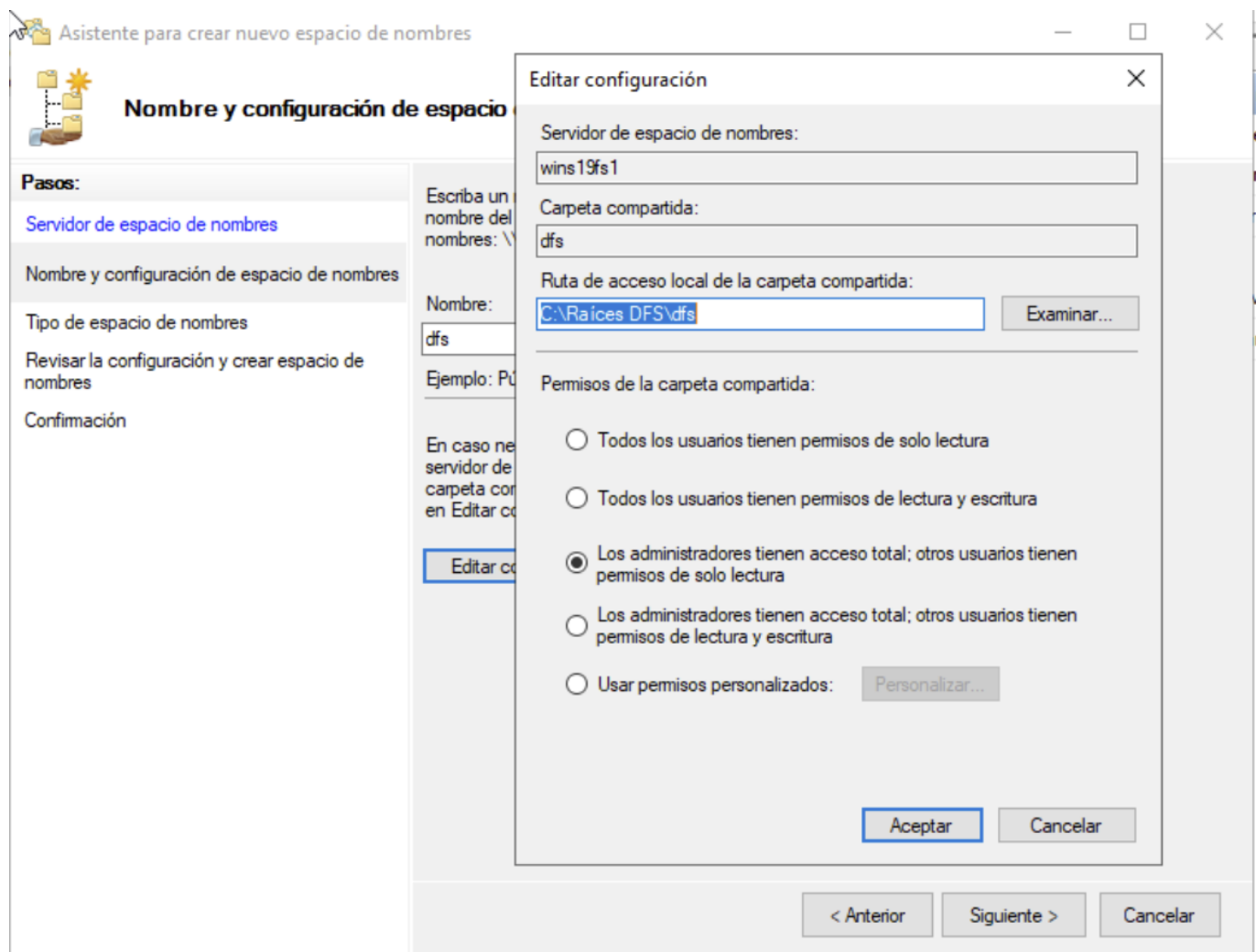
- Servidor de espacio de nombres
- Nombre y configuración de espacio de nombres
- Tipo de espacio de nombres
- Revisar la configuración y crear espacio de nombres
- Confirmación

Escriba el nombre del servidor que hospedará el espacio de nombres. El servidor que especifique se conocerá como el servidor de espacio de nombres.

Servidor:

< Anterior    Siguiente >    Cancelar

- Assign a name to the dfs namespace and click on edit configuration and select administrators full access and other users read-only



- Pick Domain-based namespace

## Add a shared folder to the namespace

Nueva carpeta

Nombre:  
fotos

Vista previa del espacio de nombres:  
\\aragon.local\dfs\fotos

Destinos de carpeta:

Agregar...

Editar...

Quitar

Agregar destino de carpeta

Ruta de acceso de destino de carpeta:  
\\WINS19FS1\fotos

Examinar...

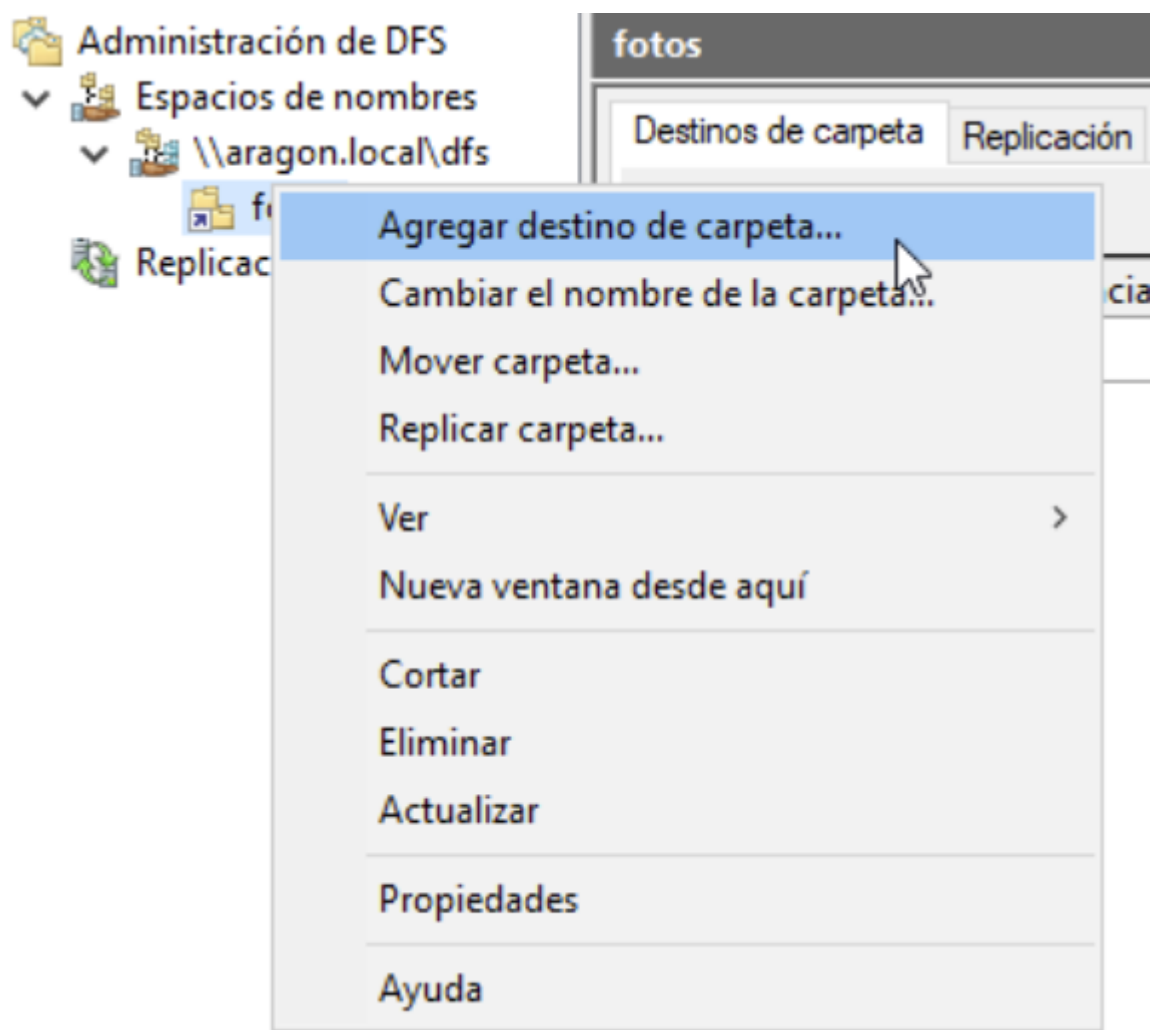
Ejemplo: \\Servidor\Carpeta compartida\Carpeta

Aceptar

Cancelar

## Replication

We must have the same folder shared with the same permissions on each server to do the replication



Select the other server and search for the other folder and done.

# Group Policies

---

## Commands

---

Update GPOs on host

```
gpupdate /force
```

Check GPOs applied

```
gpresult /R
```

## 1. Set Maximum Password Age to Lower Limits

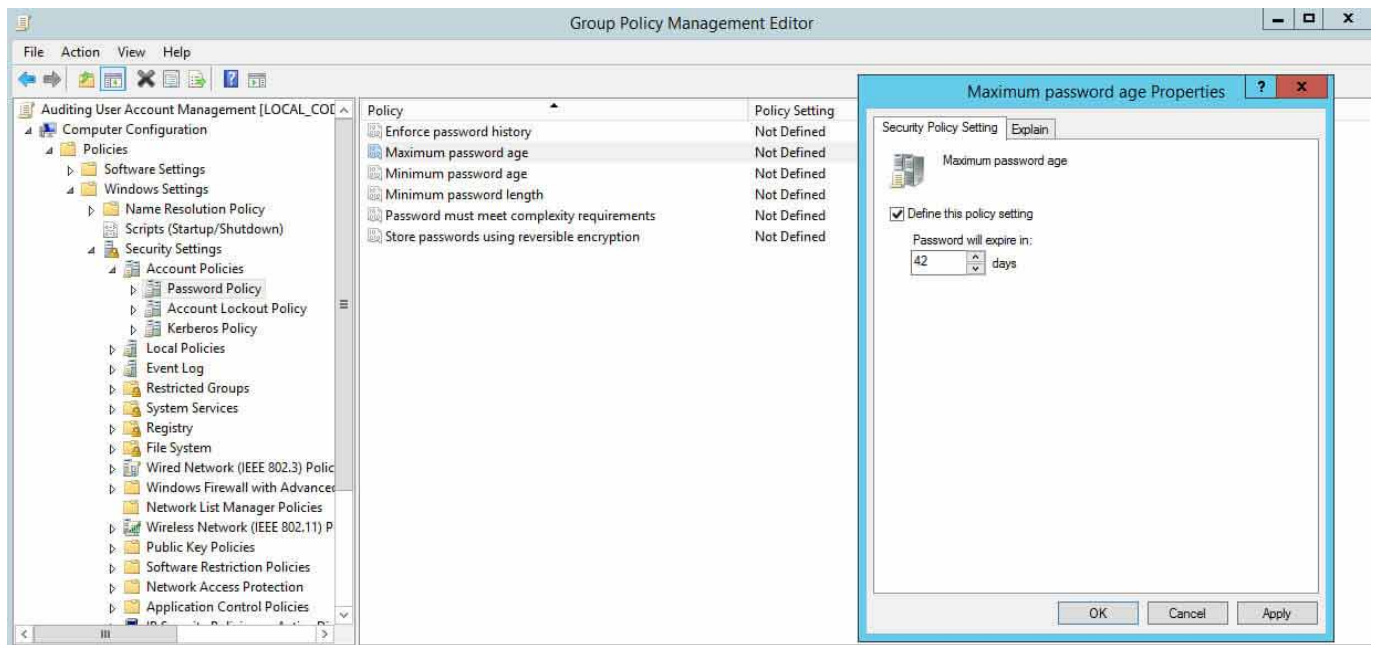
---

If you set the password expiration age to a lengthy period of time, users will not have to change it very frequently, which means it's more likely a password could get stolen. Shorter password expiration periods are always preferred.

Windows' default maximum password age is set to 42 days. The following screenshot shows the policy setting used for configuring "Maximum Password Age". Perform the following steps:

1. In Group Policy Management Editor window (opened for a custom GPO), go to "Computer Configuration" "Windows Settings" "Security Settings" "Account Policies" "Password Policy".
2. In the right pane, double-click "Maximum password age" policy.
3. Select "Define this policy setting" checkbox and specify a value.
4. Click "Apply" and "OK".



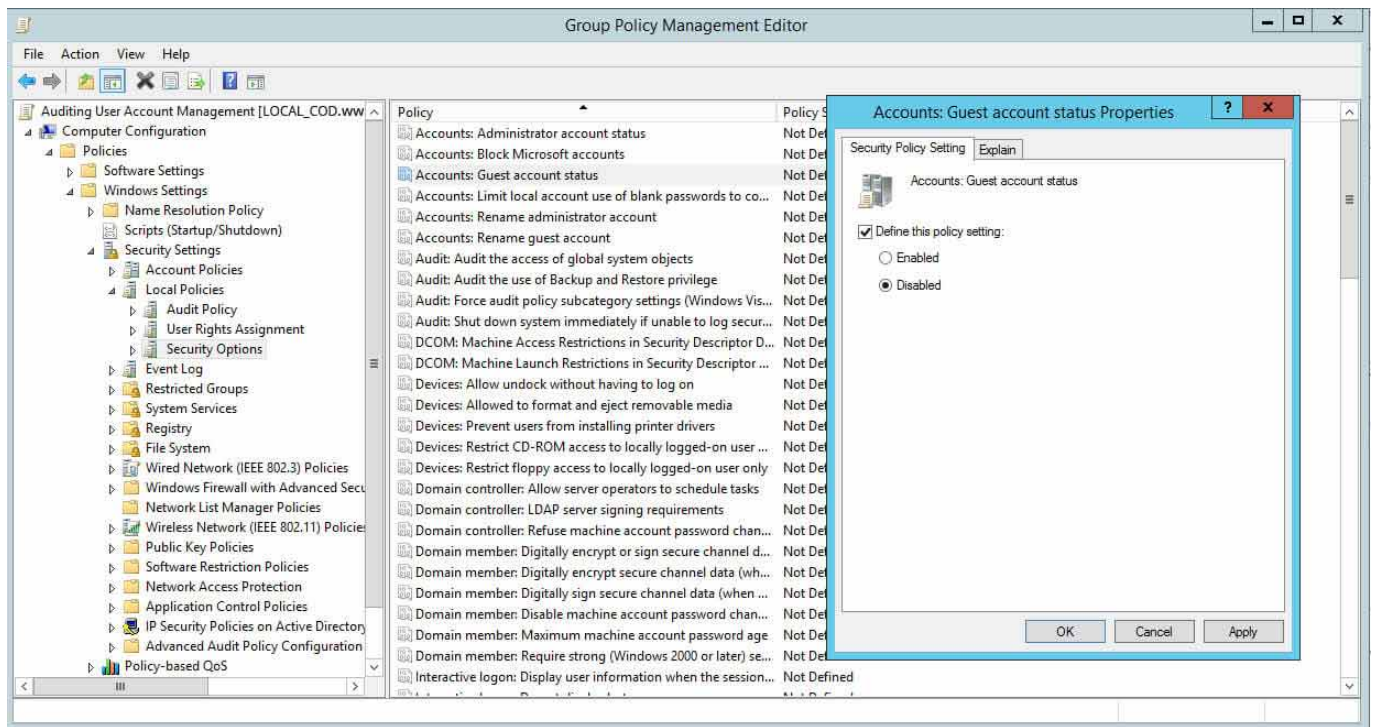


## 2. Disable Guest Account

Through a Guest Account, users can get access to sensitive data. Such accounts grant access to a Windows computer and do not require a password. Enabling this account means anyone can misuse and abuse access to your systems.

Thankfully, these accounts are disabled by default. It's best to check that this is the case in your IT environment as, if this account is enabled in your domain, disabling it will prevent people from abusing access:

1. In Group Policy Management Editor (opened for a custom GPO), go to "Computer Configuration" "Windows Settings" "Security Settings" "Local Policies" "Security Options".
2. In the right pane, double-click "Accounts: Guest Account Status" policy.
3. Select "Define this policy setting" checkbox and click "Disabled".
4. Click "Apply" and "OK".

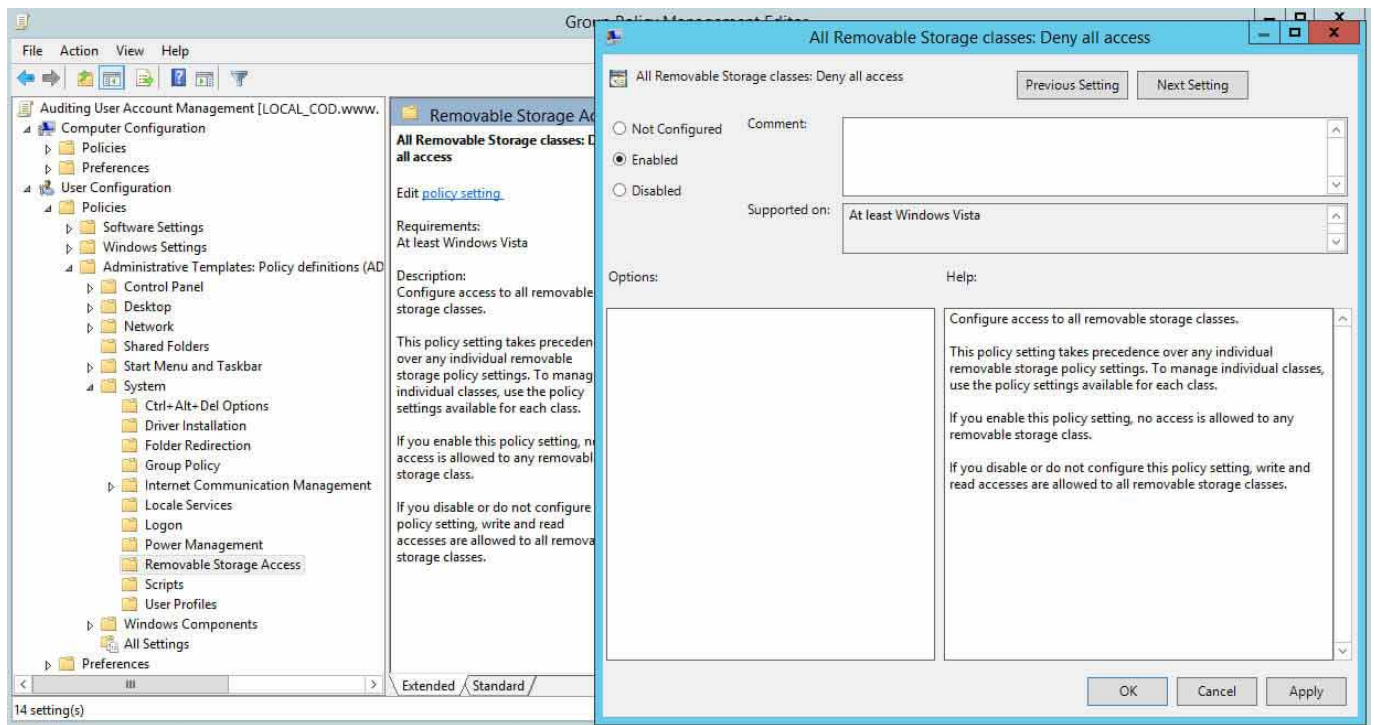


### 3. Disallow Removable Media Drives, DVDs, CDs, and Floppy Drives

Removable media drives are very prone to infection, and they may also contain a virus or malware. If a user plugs an infected drive to a network computer, it can affect the entire network. Similarly, DVDs, CDs and Floppy Drives are prone to infection.

It is therefore best to disable all these drives entirely. Perform the following steps to do so:

1. In Group Policy Management Editor window (opened for a custom GPO), go to “User Configuration” “Policies” “Administrative Templates” “System” “Removable Storage Access”.
2. In the right pane, double-click “All removable storage classes: Deny all accesses” policy
3. Click “Enabled” to enable the policy.
4. Click “Apply” and “OK”.

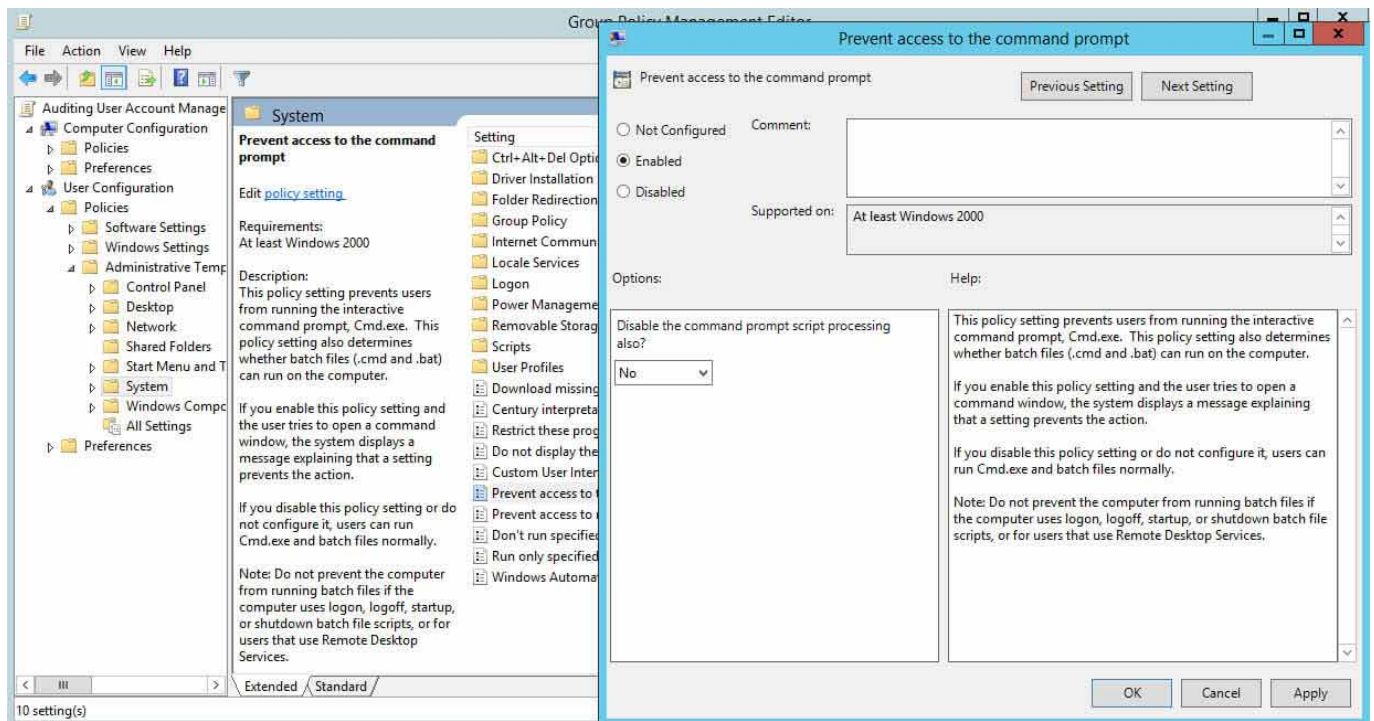


## 4. Control Access to Command Prompt

Command Prompts can be used to run commands that give high-level access to users and evade other restrictions on the system. So, to ensure system resources' security, it's wise to disable Command Prompt.

After you have disabled Command Prompt and someone tries to open a command window, the system will display a message stating that some settings are preventing this action. Perform the following steps:

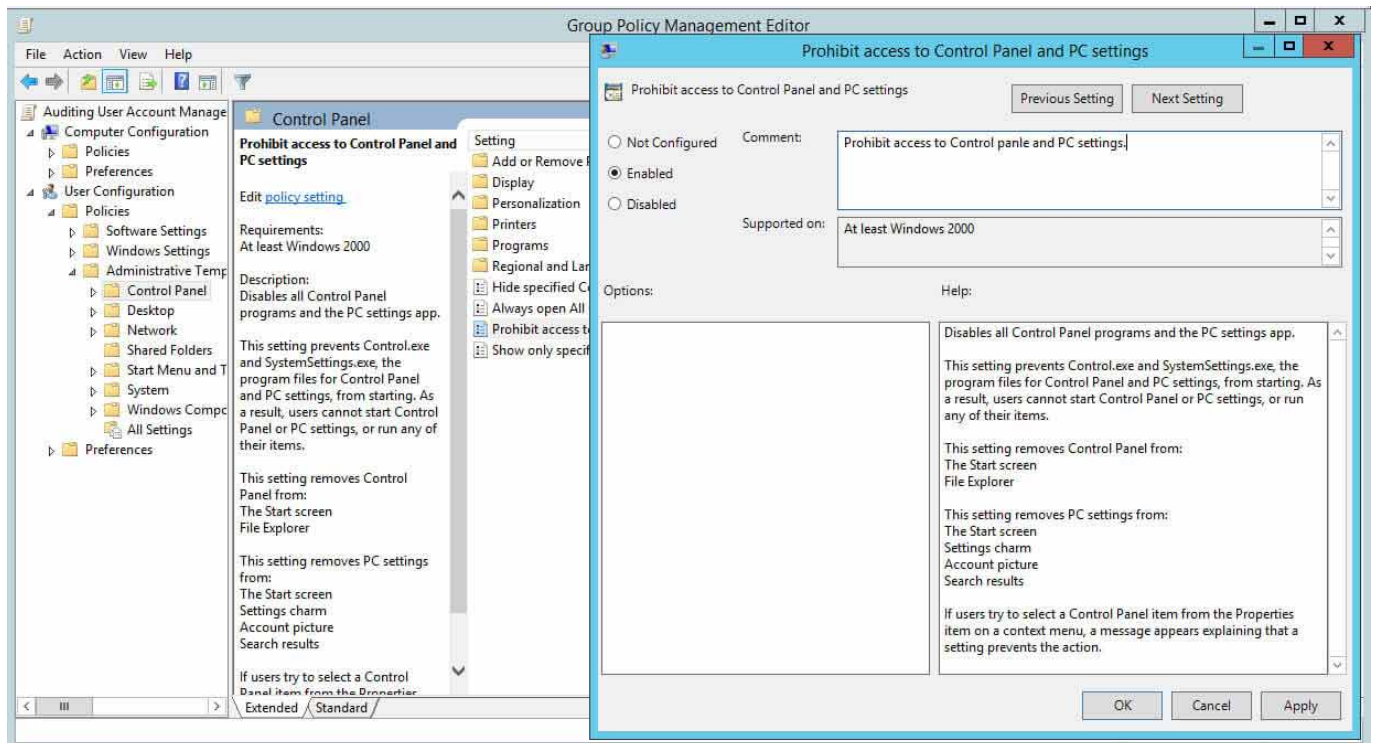
1. In the window of Group Policy Management Editor (opened for a custom GPO), go to "User Configuration" "Windows Settings" "Policies" "Administrative Templates" "System".
2. In the right pane, double-click "Prevent access to the command prompt" policy.
3. Click "Enabled" to apply the policy.
4. Click "Apply" and "OK".



## 5. Moderating Access to Control Panel

Setting limits on a computers' Control Panel creates a safer business environment. Through Control Panel, you can control all aspects of your computer. So, by moderating who has access to the computer, you can keep data and other resources safe. Perform the following steps:

1. In Group Policy Management Editor (opened for a user-created GPO), navigate to "User Configuration" "Administrative Templates" "Control Panel".
2. In the right pane, double-click "Prohibit access to Control Panel and PC settings" policy in to open its properties.
3. Select "Enabled" from the three options.
4. Click "Apply" and "OK".



# Structure creation

---



```

$dominioLDAP="DC=dom,DC=as21"
$domain_name="dom.as21"
$userPass=(convertto-securestring "IVSZ2e12" -AsPlainText -Force)

# Creacion de las unidades organizativas
New-ADOrganizationalUnit -DisplayName "USUARIOS" -Name "USUARIOS" -
ProtectedFromAccidentalDeletion $False -path $dominioLDAP

New-ADOrganizationalUnit -DisplayName "Gestores" -Name "Gestores" -
ProtectedFromAccidentalDeletion $False -path "OU=USUARIOS,$dominioLDAP"
New-ADOrganizationalUnit -DisplayName "Tutores" -Name "Tutores" -
ProtectedFromAccidentalDeletion $False -path "OU=USUARIOS,$dominioLDAP"
New-ADOrganizationalUnit -DisplayName "Competidores" -Name "Competidores" -
ProtectedFromAccidentalDeletion $False -path "OU=USUARIOS,$dominioLDAP"

# Creacion de los grupos de usuarios

New-ADGroup -DisplayName "Gestores" -Name "Gestores" -GroupScope Global -GroupCategory Security
-Path "OU=Gestores,OU=USUARIOS,$dominioLDAP"
New-ADGroup -DisplayName "Tutores" -Name "Tutores" -GroupScope Global -GroupCategory Security -
Path "OU=Tutores,OU=USUARIOS,$dominioLDAP"
New-ADGroup -DisplayName "Competidores" -Name "Competidores" -GroupScope Global -GroupCategory
Security -Path "OU=Competidores,OU=USUARIOS,$dominioLDAP"

for (( $i = 1); $i -lt 11 ; $i++)
{
    $name="Gestor$i"
    New-ADUser -Name $name -UserPrincipalName "$name@$domain_name" -PasswordNeverExpires $True
-CannotChangePassword $True -Enabled $True -AccountPassword $userPass -Path
"OU=Gestores,OU=USUARIOS,$dominioLDAP"
    Add-ADGroupMember "Gestores" $name
}

for (( $i = 1); $i -lt 11 ; $i++)
{
    $name="Tutor$i"
    New-ADUser -Name $name -UserPrincipalName "$name@$domain_name" -PasswordNeverExpires $True
-CannotChangePassword $True -Enabled $True -AccountPassword $userPass -Path
"OU=Tutores,OU=USUARIOS,$dominioLDAP"
    Add-ADGroupMember "Tutores" $name
}

for (( $i = 1); $i -lt 11 ; $i++)
{

```

```
$name="Competidor$i"
New-ADUser -Name $name -UserPrincipalName "$name@$domain_name" -PasswordNeverExpires $True
-CannotChangePassword $True -Enabled $True -AccountPassword $userPass -Path
"OU=Competidores,OU=USUARIOS,$dominioLDAP"
Add-ADGroupMember "Competidores" $name
}
```



# Debian command utils

---

## Prepare machine

---

### Change hostname

```
hostname <new hostname>
```

### Edit `/etc/hosts`

```
127.0.0.1 localhost
127.0.0.1 <new hostname>
```

### Config `/etc/resolv.conf`

---

```
nameserver 1.1.1.1
nameserver 8.8.8.8
nameserver 8.8.4.4
domain aragon.local
```

## Basic interface config

---

Always start with interface up

```
auto eth0

# Static
iface eth0 inet static
    address 192.168.1.5
    netmask 255.255.255.0
    gateway 192.168.1.254
    dns-nameservers 192.168.1.250
```

## DHCP

```
auto eth0  
iface eth0 inet dhcp
```

## NSLOOKUP

---

```
apt install dnsutils
```

# Instalar Bind9 DNS

Instalación y configuración del servicio DNS Bind9

## 1. Requisitos

Verificamos la configuración de fichero `/etc/resolv.conf` Tiene que quedar algo como esto:

```
nameserver 127.0.0.1          // DNS con el que el host local va a intentar
resolver
search aragon.local          // El dominio donde va a buscar el nombre primero
```

Note: Al ejecutar un comando como por ejemplo: `ping www` lo primero que hará será buscar [www.aragon.local](http://www.aragon.local), si no se encuentra, probará resolver el nombre `www`.

## 2. Instalación

Instalamos el paquete `bind9` con `apt install bind9` Verificamos que el servicio está en ejecución con el comando `systemctl status bind9`

```
• named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset:
   enabled)
   Active: active (running) since Mon 2021-12-06 10:31:03 CET; 1h 19min ago
     Docs: man:named(8)
  Main PID: 402 (named)
    Tasks: 5 (limit: 2330)
   Memory: 22.9M
      CPU: 482ms
   CGroup: /system.slice/named.service
           └─402 /usr/sbin/named -f -u bind
```

## 3. Configuración

En el archivo de configuración `/etc/bind/named.conf.local` configuramos las siguientes zonas:

### 3.1. Zona directa

```
zone "aragon.local" {
    type master;
    file "/etc/bind/db.aragon.local";
    notify yes;
};
```

## 3.2. Zona inversa

```
zone "30.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.30.168.192";
    notify yes;
};
```

## 3.3. Configurar archivos de las zonas directa e inversa

```
cp /etc/bind/db.local      /etc/bind/db.aragon.local
cp /etc/bind/db.local      /etc/bind/db.30.168.192
```

Configuramos el archivo de la zona directa `db.aragon.local`

```
;
; BIND data file for zona directa aragon.local
;
$TTL      604800
@         IN      SOA      dns.aragon.local. root.aragon.local. (
                        2      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS       dns.aragon.local.
dns       IN      A        192.168.30.5
dhcp      IN      CNAME    dns
nsserver   IN      CNAME    dns
mailserver IN      A        192.168.30.6
mail       IN      CNAME    mailserver
clienteubuntu IN      A      192.168.30.18
```

Configuramos el archivo de la zona inversa `db.30.168.192`

```

;
; BIND data file for zona inversa 30.168.192
;
$TTL      604800
@         IN      SOA      dns.aragon.local. root.aragon.local. (
                                2          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800 )   ; Negative Cache TTL
;
@         IN      NS       dns.aragon.local.
5         IN      PTR      dns.aragon.local.
5         IN      PTR      dhcp.aragon.local.
5         IN      PTR      nsserver.aragon.local.
6         IN      PTR      mailserver.aragon.local.
6         IN      PTR      mail.aragon.local.
18        IN      PTR      clienteubuntu.aragon.local.

```

### 3.4. Reenviadores o forwarders

Configuramos los servidores DNS públicos para reenviar las peticiones que no pueda resolver en este archivo `/etc/bind/named.conf.options`

```

forwarders {
    1.1.1.1;
    1.0.0.1;
    8.8.8.8;
    8.8.4.4;
};

```

### 3.5. Aplicar configuración

Para aplicar la configuración basta con reiniciar el servicio:

```
systemctl restart bind9
```

## 4. Diagnostico

### 4.1. Estado del servicio

Para comprobar que el servicio está funcionando correctamente:

```
systemctl status bind9
```

Nos tendrá que salir algo como esto:

```
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset:
   enabled)
   Active: active (running) since Mon 2021-12-06 19:50:19 CET; 7min ago
     Docs: man:named(8)
  Main PID: 3931 (named)
    Tasks: 5 (limit: 2330)
   Memory: 23.8M
      CPU: 152ms
   CGroup: /system.slice/named.service
           └─3931 /usr/sbin/named -f -u bind

dic 06 19:50:19 dns named[3931]: network unreachable resolving './NS/IN':
2001:7fe::53#53
dic 06 19:50:19 dns named[3931]: zone aragon.local/IN: loaded serial 2
dic 06 19:50:19 dns named[3931]: zone 127.in-addr.arpa/IN: loaded serial 1
dic 06 19:50:19 dns named[3931]: zone localhost/IN: loaded serial 2
dic 06 19:50:19 dns named[3931]: zone 255.in-addr.arpa/IN: loaded serial 1
dic 06 19:50:19 dns named[3931]: zone 30.168.192.in-addr.arpa/IN: loaded serial 2
dic 06 19:50:19 dns named[3931]: all zones loaded
dic 06 19:50:19 dns named[3931]: running
dic 06 19:50:19 dns named[3931]: managed-keys-zone: Key 20326 for zone . is now
trusted (acceptance timer complet>
dic 06 19:50:19 dns named[3931]: resolver priming query complete
```

## 4.2. Error al cargar zona

Si nos sale un error parecido a este:

```
dic 06 14:43:10 dns named[3557]: zone aragon.local/IN: loaded serial 2
dic 06 14:43:10 dns named[3557]: /etc/bind/db.30.168.192:13: unknown RR type 'RTP'
dic 06 14:43:10 dns named[3557]: zone 30.168.192.in-addr.arpa/IN: loading from master
file /etc/bind/db.30.168.19>
dic 06 14:43:10 dns named[3557]: zone 30.168.192.in-addr.arpa/IN: not loaded due to
errors.
dic 06 14:43:10 dns named[3557]: zone 255.in-addr.arpa/IN: loaded serial 1
dic 06 14:43:10 dns named[3557]: zone localhost/IN: loaded serial 2
dic 06 14:43:10 dns named[3557]: all zones loaded
dic 06 14:43:10 dns named[3557]: running
dic 06 14:43:10 dns named[3557]: managed-keys-zone: Key 20326 for zone . is now
trusted (acceptance timer complet>
dic 06 14:43:11 dns named[3557]: resolver priming query complete
```

En este caso significa que en la zona inversa `30.168.192.in-addr.arpa` hay un error y no la ha cargado. Cuando ejecutemos el comando `nslookup` en el CMD de windows con ese error presente, el resultado será el siguiente:

```
C:\Users\Administrador>nslookup
Servidor predeterminado: UnKnown
Address: 192.168.30.5
```

Nos dirá que el servidor es **UnKnown**, ya que no puede obtener el nombre del servidor de la zona inversa ya que no está cargada. Cuando resolvamos el error de la zona inversa y hallamos reiniciado el servicio el resultado debería ser algo así:

```
C:\Users\Administrador>nslookup
Servidor predeterminado: dns.aragon.local
Address: 192.168.30.5
```

# Instalar Isc DHCP Server

---

## 1. Instalación

---

- Primero nos aseguramos de tener las interfaces de red bien configuradas en el archivo `/etc/network/interfaces`
- Instalamos el servidor DHCP `apt install isc-dhcp-server`

Nos saldrá un error de inicialización del servicio, pero eso es porque no lo hemos configurado.

## 2. Configuración

---

Configuramos las interfaces a las que va a dar servicio el DHCP en el siguiente archivo

`/etc/default/isc-dhcp-server`

```
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPDv4_PID=/var/run/dhcpd.pid
#DHCPDv6_PID=/var/run/dhcpd6.pid

# Additional options to start dhcpd with.
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="ens19"
INTERFACESv6=""
```

En este caso la interfáz se llama `ens19`

---

Configuramos los parametros del DHCP en el siguiente archivo: `/etc/dhcp/dhcpd.conf`

- El dominio por defecto:



```
option domain-name "aragon.local";
```

- El tiempo predeterminado que se le va a dar las IPs a los hosts:

```
default-lease-time 10080;           // En segundos 7 días  
max-lease-time 20160;              // 14 días
```

- DNS predeterminados:

```
option domain-name-servers 8.8.8.8, 8.8.4.4;
```

- Configuración de rango:

```
subnet 192.168.30.0 netmask 255.255.255.0 {  
    range 192.168.30.11 192.168.30.20;  
    range 192.168.30.31 192.168.30.40;  
    option routers 192.168.30.1;  
    option domain-name-servers 192.168.30.5;  
    default-lease-time 1440;           // En segundos 1 día  
    max-lease-time 2880;              // 2 días  
}
```

- Reserva de IP:

```
host mailserver {  
    hardware ethernet 32:0d:92:ba:50:d5;  
    fixed-address 192.168.30.6;  
}
```

Archivo completo:

```
# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#

# option definitions common to all supported networks...
option domain-name "example.org";
option domain-name-servers 8.8.8.8, 8.8.4.4;

default-lease-time 10080;
max-lease-time 20160;

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style none;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
#log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

#subnet 10.152.187.0 netmask 255.255.255.0 {
#}
# This is a very basic subnet declaration.

subnet 192.168.30.0 netmask 255.255.255.0 {
    range 192.168.30.11 192.168.30.20;
    range 192.168.30.31 192.168.30.40;
    option routers 192.168.30.1;
    option domain-name-servers 192.168.30.5;
    default-lease-time 1440;
    max-lease-time 2880;
}

host mailserver {
    hardware ethernet 32:0d:92:ba:50:d5;
    fixed-address 192.168.30.6;
}

# This declaration allows BOOTP clients to get dynamic addresses,
# which we don't really recommend.
```

```

#subnet 10.254.239.32 netmask 255.255.255.224 {
#  range dynamic-bootp 10.254.239.40 10.254.239.60;
#  option broadcast-address 10.254.239.31;
#  option routers rtr-239-32-1.example.org;
#}

# A slightly different configuration for an internal subnet.
#subnet 10.5.5.0 netmask 255.255.255.224 {
#  range 10.5.5.26 10.5.5.30;
#  option domain-name-servers ns1.internal.example.org;
#  option domain-name "internal.example.org";
#  option routers 10.5.5.1;
#  option broadcast-address 10.5.5.31;
#  default-lease-time 600;
#  max-lease-time 7200;
#}

# Hosts which require special configuration options can be listed in
# host statements.  If no address is specified, the address will be
# allocated dynamically (if possible), but the host-specific information
# will still come from the host declaration.

#host passacaglia {
#  hardware ethernet 0:0:c0:5d:bd:95;
#  filename "vmunix.passacaglia";
#  server-name "toccata.example.com";
#}

# Fixed IP addresses can also be specified for hosts.  These addresses
# should not also be listed as being available for dynamic assignment.
# Hosts for which fixed IP addresses have been specified can boot using
# BOOTP or DHCP.  Hosts for which no fixed address is specified can only
# be booted with DHCP, unless there is an address range on the subnet
# to which a BOOTP client is connected which has the dynamic-bootp flag
# set.
#host fantasia {
#  hardware ethernet 08:00:07:26:c0:a5;
#  fixed-address fantasia.example.com;
#}

# You can declare a class of clients and then do address allocation
# based on that.  The example below shows a case where all clients
# in a certain class get addresses on the 10.17.224/24 subnet, and all
# other clients get addresses on the 10.0.29/24 subnet.

#class "foo" {
#  match if substring (option vendor-class-identifier, 0, 4) = "SUNW";
#}

#shared-network 224-29 {
#  subnet 10.17.224.0 netmask 255.255.255.0 {

```

```
#    option routers rtr-224.example.org;
# }
# subnet 10.0.29.0 netmask 255.255.255.0 {
#    option routers rtr-29.example.org;
# }
# pool {
#    allow members of "foo";
#    range 10.17.224.10 10.17.224.250;
# }
# pool {
#    deny members of "foo";
#    range 10.0.29.10 10.0.29.230;
# }
#}
```

### 3. Iniciamos el servicio DHCP

---

```
/etc/init.d/isc-dhcp-server restart

systemctl restart isc-dhcp-server.service

service isc-dhcp-server restart
```

### 4. Comprobación de errores

---

Verificar el log del sistema:

```
cat /var/log/syslog
```

Verificar que el proceso está en ejecución:

```
ps -ef | grep dhcp
```

Comprobar que el servidor escucha por el puerto 67 y 68:

```
netstat -putona | grep :67
netstat -putona | grep :68
```

Consultar el fichero de concesiones para comprobar que todavía no existe ninguna concesión:

```
cat /var/lib/dhcp/dhcpd.leases
```

# Configure RAID 0-1-5 on debian

---

## Partitioning with fdisk

---

You don't need to partition disks before using them in an array, but partitioning does provide a couple of advantages.

- Partitioning is necessary if you want the kernel to automatically start arrays, because the md driver uses the partition type to identify member disks.

If you have a lot of disks, then you might not want to go through the trouble of partitioning each disk since this process can take a lot of time if you have more than a few drives. In that case, you can simply use a whole, unpartitioned disk as an array member ( `/dev/sda` , for example). This means that you won't be able to autostart arrays, however, so you'll have to include commands to start `md` devices in your system initialization scripts.

Next we will start creating partitions on our disks `/dev/sdb` and `/dev/sdc` using `fdisk`

Partition 1 of type Linux and of size 4 GiB is set

Command (m for help): n

Tipo de partición

p primaria (0 primaria(s), 0 extendida(s), 4 libre(s))

e extendida (contenedor para particiones lógicas)

Seleccionar (valor predeterminado p): p

Se está utilizando la respuesta predeterminada p.

Número de partición (1-4, valor predeterminado 1): 1

Primer sector (2048-1048575, valor predeterminado 2048):

Último sector, +/-sectores o +/-tamaño{K,M,G,T,P} (2048-1048575, valor predeterminado 1048575):

Crea una nueva partición 1 de tipo 'Linux' y de tamaño 511 MiB.

Command (m for help): t

Selected partition 1

Hex code (type L to list all codes): fd

Changed type of partition 'Linux' to 'Linux raid autodetect'

Command (m for help): p

Disk /dev/sdc: 4294 MB, 4294967296 bytes, 8388608 sectors

Units = sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk label type: dos

Disk identifier: 0xe215a659

Device	Boot	Start	End	Blocks	Id	System
/dev/sdc1		2048	8388607	4193280	fd	Linux raid autodetect

Command (m for help): w

The partition table has been altered!

Calling ioctl() to re-read partition table.

Syncing disks.

Update the partition table in the kernel.

```
[root@node1 ~]# partprobe
```

Now list the available partitions on your node and verify the changes. So now we have two new partitions `/dev/sdb1` and `/dev/sdc1` for setting up linear mode software raid.

```
[root@node1 ~]# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	30G	0	disk	
├─sda1	8:1	0	512M	0	part	/boot
└─sda2	8:2	0	27.5G	0	part	
├─centos-root	253:0	0	25.5G	0	lvm	/
└─centos-swap	253:1	0	2G	0	lvm	[SWAP]
sdb	8:16	0	4G	0	disk	
└─sdb1	8:17	0	4G	0	part	
sdcc	8:32	0	4G	0	disk	
└─sdcc1	8:33	0	4G	0	part	
sr0	11:0	1	1024M	0	rom	

## Create Linear Software RAID 0

```
[root@node1 ~]# mdadm -Cv -llinear -n2 /dev/md0 /dev/sd{b,c}1
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

OR you can also use the long version of this command to create linear raid

```
# mdadm --create --verbose --level=linear --raid-devices=2 /dev/md0 /dev/sdb1
/dev/sdc1
```

Here,

-C, --create  
Create a new array.

-v, --verbose  
Be more verbose.

-l, --raid-level  
Select the RAID level: linear, 0, 1, 4, or 5.

-n, --raid-disks  
Set the number of member disks in the array.

`mdadm` automatically activate newly created linear raid arrays. Information about the array and its member disks is now available via the `/proc/mdstat` pseudo file.



```
[root@node1 ~]# cat /proc/mdstat
Personalities : [linear]
md0 : active linear sdc1[1] sdb1[0]
      8380416 blocks super 1.2 0k rounding

unused devices: <none>
```

## Create RAID 5

Now since we have all the partitions with us, we will create software RAID 5 array on those partitions

```
[root@node1 ~]# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda                  8:0    0   30G  0 disk
├─sda1               8:1    0   512M  0 part  /boot
└─sda2               8:2    0  27.5G  0 part
   ├─centos-root     253:0    0  25.5G  0 lvm    /
   └─centos-swap     253:1    0    2G    0 lvm    [SWAP]
sdb                  8:16    0    2G    0 disk
└─sdb1               8:17    0    2G    0 part
sdc                  8:32    0    2G    0 disk
└─sdc1               8:33    0    2G    0 part
sdd                  8:48    0    2G    0 disk
└─sdd1               8:49    0    2G    0 part
sr0                 11:0    1 1024M  0 rom
```

Execute the below command to create software raid 5 array using `/dev/sdb1` , `/dev/sdc1` and `/dev/sdd1`

```
[root@node1 ~]# mdadm -Cv -l5 -c64 -n3 -pls /dev/md0 /dev/sd{b,c,d}1
mdadm: /dev/sdb1 appears to contain an ext2fs file system
      size=2096128K  mtime=Wed Jun 12 11:21:25 2019
mdadm: size set to 2094080K
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

`mdadm` defaults to the **left-symmetric** algorithm, so you can safely omit the `-p` option from the command line.

The left-symmetric algorithm will yield the best disk performance for a RAID-5, although this value can be changed to one of the other algorithms (right-symmetric, left-asymmetric, or right-asymmetric).

Here,

```
-C, --create
    Create a new array.

-v, --verbose
    Be more verbose about what is happening.

-l, --level=
    Set RAID level. When used with --create, options are: linear, raid0, 0,
stripe, raid1, 1, mirror,
raid4, 4, raid5, 5, raid6, 6, raid10, 10, multipath, mp, faulty, container.
Obviously some of these
are synonymous.

-c, --chunk=
    Specify chunk size of kilobytes.

-n, --raid-devices=
    Specify the number of active devices in the array.

-p, --layout=
    This option configures the fine details of data layout for RAID5, RAID6, and
RAID10 arrays, and controls the failure modes
for faulty.

    The layout of the RAID5 parity block can be one of left-asymmetric, left-
symmetric, right-asymmetric, right-symmetric, la,
ra, ls, rs. The default is left-symmetric.
```

## Create filesystem

Next, create a file-system on the new software raid array. We will create ext4 filesystem on our linear raid array

```
[root@node1 ~]# mkfs.ext4 /dev/md0
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
524288 inodes, 2095104 blocks
104755 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2145386496
64 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

## Create mount point

Create a mount point for accessing the software raid array

```
[root@node1 ~]# mkdir /linear_raid
```

Next mount the raid array on the created directory

```
[root@node1 ~]# mount /dev/md0 /linear_raid/
```

Check the mount status which also gives more details about mount point, available space etc details.

```
[root@node1 ~]# df -h /linear_raid/
Filesystem      Size  Used Avail Use% Mounted on
/dev/md0        7.8G   36M   7.3G   1% /linear_raid
```

Next add an entry for the array to `/etc/fstab` file so it will be mounted automatically when the system restarts.

```
[root@node1 ~]# tail -n 1 /etc/fstab
/dev/md0      /linear_raid  ext4    defaults    0 0
```

Be warned that some distributions (Red Hat, for one) halt system initialization if an `/etc/fstab` entry could not be properly checked and mounted. So if the kernel doesn't automatically start your array, an entry in `/etc/fstab` might be preventing the system from booting successfully. It's a good idea to place commands that will manually start arrays in your initialization scripts before filesystems are checked and mounted, even if you're already successfully using autodetection. This will provide additional stability and, at worst, display some innocuous warnings on the console.

## Verify the software raid changes

---

Post reboot verify the raid status

```
[root@node1 ~]# cat /proc/mdstat
Personalities : [linear]
md0 : active linear sdc1[1] sdb1[0]
      8380416 blocks super 1.2 0k rounding

unused devices: <none>
```

# Instalar PostFix Mail server

---

Tutorial de como configurar postfix en Debian 11

## Instalación

---

1. Instalaremos los paquetes `postfix` y `mailutils` con el comando `apt install postfix mailutils` y lo configuramos con `dpkg-reconfigure postfix`

```
apt install postfix mailutils
dpkg-reconfigure postfix
```

2. En tipo generico de configuración de correos: **Sitio de Internet**
3. En el nombre se sistema de coreo ponemos el dominio: **aragon.local**
4. Recipiente de correo para el administrador: -
5. Otros destinos: -
6. Forzar actualizaciones asincronas: **no**
7. Redes locales: -
8. Límite tamaño del buzón: -
9. Caracter de extensión: -
10. Protocolo de Internet: **Ipv4**

## Configuración

---

Archivo de configuracion de PostFix `/etc/postfix/main.cf`

### Cambiar banner

```
smtpd_banner = SMTP Server aragon.local
```

### Config Maildir

```
home_mailbox = Maildir/
```

### Agregar certificado SSL

Generamos el certificado ssl en la carpeta `/etc/ssl/postfix` :

```
mkdir /etc/ssl/postfix
cd /etc/ssl/postfix
openssl req -nodes -new -x509 -keyout server.key -out server.cert
```

Ponemos el certificado autofirmado que hemos generado:

```
smtpd_tls_cert_file=/etc/ssl/postfix/server.cert
smtpd_tls_key_file=/etc/ssl/postfix/server.key
```

## Límite tamaño mensaje

Agregamos la siguiente línea:

```
message_size_limit = 3072000          # 3 MB
```

## Configuración de DNS

En nuestro servidor DNS tenemos que configurar lo siguiente:

- Zona Directa

mailserver	IN	A	192.168.30.6
aragon.local.	IN	MX 10	mailserver

Donde "MX" significa que es un registro de intercambio de correo y el número "10" indica la prioridad (Menor prioridad primero), en este caso el registro está apuntando a otro de tipo "A", pero podría apuntar a uno de tipo "CNAME".

- Zona Inversa

La configuramos también ya que de no hacerlo, nuestros correos pueden acabar en el SPAM

6	IN	PTR	mailserver.aragon.local.
---	----	-----	--------------------------

## Desactivar protocolos inseguros SSLv2 y SSLv3

Escribir lo siguiente en el archivo de configuración

```
smtpd_tls_mandatory_protocols=!SSLv2,!SSLv3
smtp_tls_mandatory_protocols=!SSLv2,!SSLv3
smtpd_tls_protocols=!SSLv2,!SSLv3
smtp_tls_protocols=!SSLv2,!SSLv3
```

## Creacion de cuentas

Para crear una cuenta de correo simplemente agregamos los usuarios a la maquina del servi

```
adduser tutor39
adduser compeetidor39
adduser usuario39
```

## Diagnostico

---

### Mandar un correo de prueba

Vamos a mandar un correo con el siguiente comando a la dirección `direccion@dominio.com` con el comando:

```
echo "This is the body of the email" | mail -r root@aragon.local direccion@cominio.com
-s "This is the subject line"
```

Revisar el log a tiempo real de postfix con el comando: `tail -f /var/log/mail.log`

```
echo 'export MAIL=~/.Maildir' | sudo tee -a /etc/bash.bashrc | sudo tee -a
/etc/profile.d/mail.sh
```

# Instalar Dovecot para Postfix IMAP POP3

---

## Instalación

---

Instalamos los siguientes paquetes dependiendo de los protocolos que utilicemos:

```
apt install dovecot-imapd dovecot-pop3d
```

Una vez los instalemos ya quedan configurados.

## Configuración

---

Los archivos de configuración están en la carpeta `/etc/dovecot`

### Direcciones de escucha

Buscamos y descomentamos la línea siguiente en el archivo de configuración `dovecot.conf`:

```
listen = *, ::
```

### Mail location

Configuramos donde están guardados los correos de cada usuario en el archivo `conf.d/10-mail.conf`:

```
mail_location = maildir:~/Maildir
```

En este caso están configurados en la carpeta home de cada usuario.

### Certificado ssl

En el archivo `conf.d/10-ssl.conf` configuramos lo siguiente:

```
ssl = yes
ssl_cert = </etc/ssl/postfix/server.cert
ssl_key = </etc/ssl/postfix/server.key
```



# Configurar SASL con postfix

---

## Configuración de dovecot

Creamos o modificamos el archivo de configuración `/etc/dovecot/local.conf` añadimos lo siguiente:

```
# Space separated list of wanted authentication mechanisms:
#  plain login digest-md5 cram-md5 ntlm rpa apop anonymous gssapi otp skey
#  gss-spnego
# NOTE: See also disable_plaintext_auth setting.
auth_mechanisms = plain login

service auth {
  # auth_socket_path points to this userdb socket by default. It's typically
  # used by dovecot-lda, doveadm, possibly imap process, etc. Users that have
  # full permissions to this socket are able to get a list of all usernames and
  # get the results of everyone's userdb lookups.
  #
  # The default 0666 mode allows anyone to connect to the socket, but the
  # userdb lookups will succeed only if the userdb returns an "uid" field that
  # matches the caller process's UID. Also if caller's uid or gid matches the
  # socket's uid or gid the lookup succeeds. Anything else causes a failure.
  #
  # To give the caller full permissions to lookup all users, set the mode to
  # something else than 0666 and Dovecot lets the kernel enforce the
  # permissions (e.g. 0777 allows everyone full permissions).
  unix_listener auth-userdb {
    #mode = 0666
    #user =
    #group =
  }

  # Postfix smtp-auth
  unix_listener /var/spool/postfix/private/auth {
    mode = 0666
  }

  # Auth process is run as this user.
  #user = $default_internal_user
}
```

## Configuración de postfix

En el archivo de configuración `/etc/postfix/main.cf` añadimos lo siguiente:





As a system administrator, you are probably already familiar with the LDAP protocol. If you are working in a medium to large company, you can be sure that your company already owns a LDAP server, whether it is on Linux or Windows.

Invented in the early 80s, the LDAP protocol (for Lightweight Directory Access Protocol) was created in order to store data that should be accessed over a network.

The LDAP protocol was defined as part of the RFC 4511 specification and it was implemented by many different vendors.

In this tutorial, we are taking a look at one of the implementations of the LDAP protocol : [OpenLDAP](#). OpenLDAP is a free and open-source implementation of LDAP that provides a server (called slapd) as well as utilities and libraries for developers.

Using this tutorial, you will be able to setup a complete OpenLDAP server and

That's quite a long article so without further ado, let's start by install a simple OpenLDAP server on Debian 10.

**Install OpenLDAP server on Debian 10**

Before starting, you should make sure that you have administrator rights on your system : you will need them to install new packages.

To check if you have sudo rights, execute the "sudo" command with the "-v" option.

```
$ sudo -v
```

If you are not sure on how to provide sudo [rights for users on Debian 10 or CentOS 8](#), make sure to read our dedicated guides about it.

Also, make sure that your packages are correctly updated in order to get the latest package version from the repositories.

```
$ sudo apt-get update
```

On Linux, the OpenLDAP server is called "slapd".

It is a simple and configurable stand-alone server that is used in order to read, modify and delete from a LDAP directory.

The slapd daemon also comes with many different utilities that can be used in order to create new entries easily, or to modify entries easily : slapadd [or slapasswd just to](#) name a few.

```
$ sudo apt-get install slapd
```

When installing this new package, you will be ask to configure the slapd daemon at the end of the installation.

## Configuring slapd on Debian 10

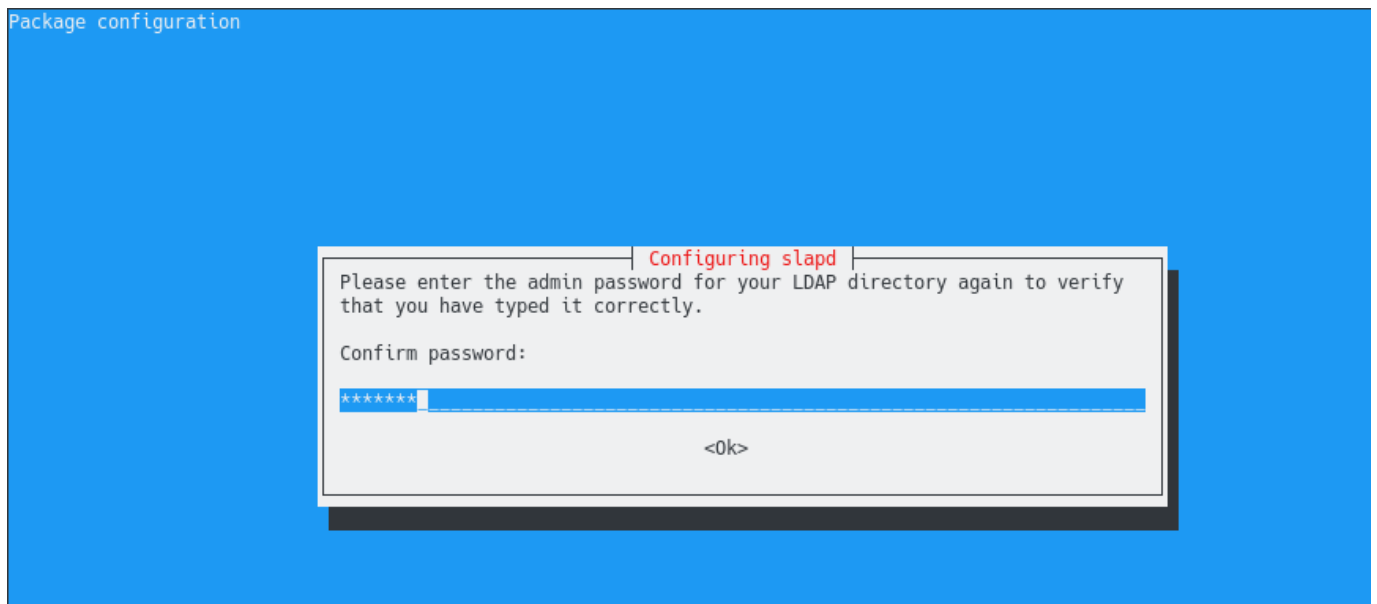
The slapd configuration comes as text-based interfaces that you need to fill in order to setup the server properly.

First, you are asked to provide an administrator password for the LDAP server.



Choose a secured password because the administrator entry in the LDAP directory has all the rights on the entire tree : add, delete and modify all the entries as well as reading all the LDAP attributes.

On the next screen, you are asked to confirm the password : simply type what you typed in the previous screen.



From there, your LDAP server should be initialized. When installing the slapd server, the installation also :

- Created a user named "openldap" on your server;
- Created an initial configuration that is available at /etc/ldap

- Created an initial and empty database that is ready to accept new entries.

```
debian-ldap@debian-ldap:~$ ls -l /etc/ldap/
total 16
-rw-r--r-- 1 root    root      332 Feb  2  2019 ldap.conf
drwxr-xr-x 2 root    root      4096 Aug 10 14:58 sasl2
drwxr-xr-x 2 root    root      4096 Jan 18 09:48 schema
drwxr-xr-x 3 openldap openldap 4096 Jan 18 09:48 slapd.d
debian-ldap@debian-ldap:~$
```

By default, the OpenLDAP server will create a first database entry that reflects your current domain name.

However, if you did not configure your domain name properly (during the installation for example), there is a chance that your OpenLDAP server is badly configured.

To take a first look at the initial configuration of your OpenLDAP server, use the “slapcat” command and watch for the distinguished names created by slapd.

```
$ sudo slapcat
$ sudo slapcat | grep dn
```

```
debian-ldap@debian-ldap:~$ sudo slapcat | grep dn
dn: dc=nodomain
dn: cn=admin,dc=nodomain
debian-ldap@debian-ldap:~$
```

Usually, your OpenLDAP top DN should match the DNS names of your domain.

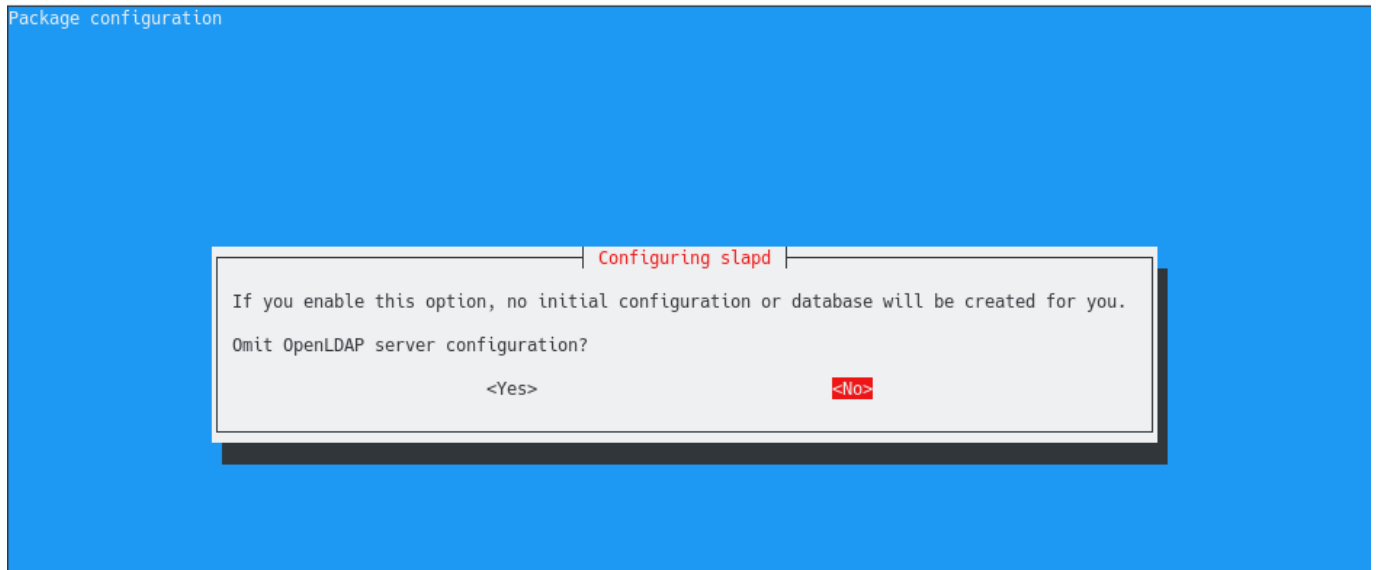
It means that if you are currently working in the “devconnected.com” domain, your OpenLDAP server should have the “dc=devconnected,dc=com” top distinguished names.

As you can see, this is not the case for now but luckily for us there is a way to reconfigure the slapd daemon.

## Reconfiguring slapd using dpkg-reconfigure

In order to reconfigure the slapd daemon, execute the “dpkg-reconfigure” command on the “slapd” daemon. Again, you need sudo privileges to reconfigure slapd.

```
$ sudo dpkg-reconfigure slapd
```



First, you are asked if you want to omit the OpenLDAP server configuration.

We obviously want to press "No" on this option because we want the initial configuration of the database to be created for us.

On the next step, you are asked to provide the base distinguished name of your LDAP server.



As you can see, the slapd daemon describes that the DNS domain name is used to build the base DN of your OpenLDAP directory.

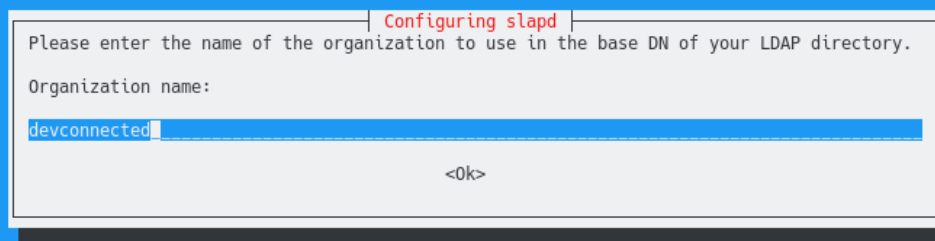
In this case, we are choosing to have "dc=devconnected,dc=com" : note that you have to modify this entry to match your current DNS settings.

If you are not sure about the domain that you belong to, simply use the "domainname" command in your terminal.

```
$ domainname devconnected.com
```

Next, you are asked to provide the name of your organization. This is exactly the same step as the one done before, simply type your organization name and hit "Ok".

Package configuration



Configuring slapd

Please enter the name of the organization to use in the base DN of your LDAP directory.

Organization name:

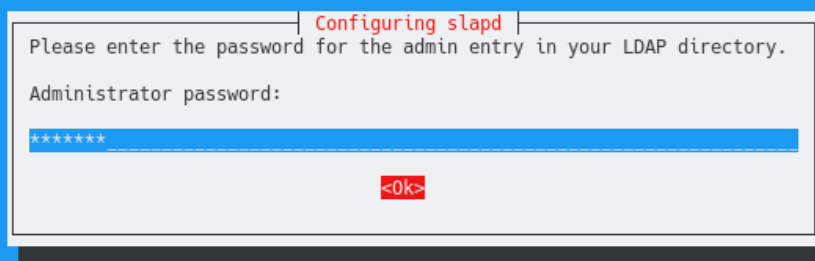
devconnected

<Ok>

Finally, similarly to the first slapd configuration, you are asked to provide admin credentials for your LDAP server.

Again, choose a strong password as it can be used in order to read and modify every single entry in the LDAP directory.

Package configuration



Configuring slapd

Please enter the password for the admin entry in your LDAP directory.

Administrator password:

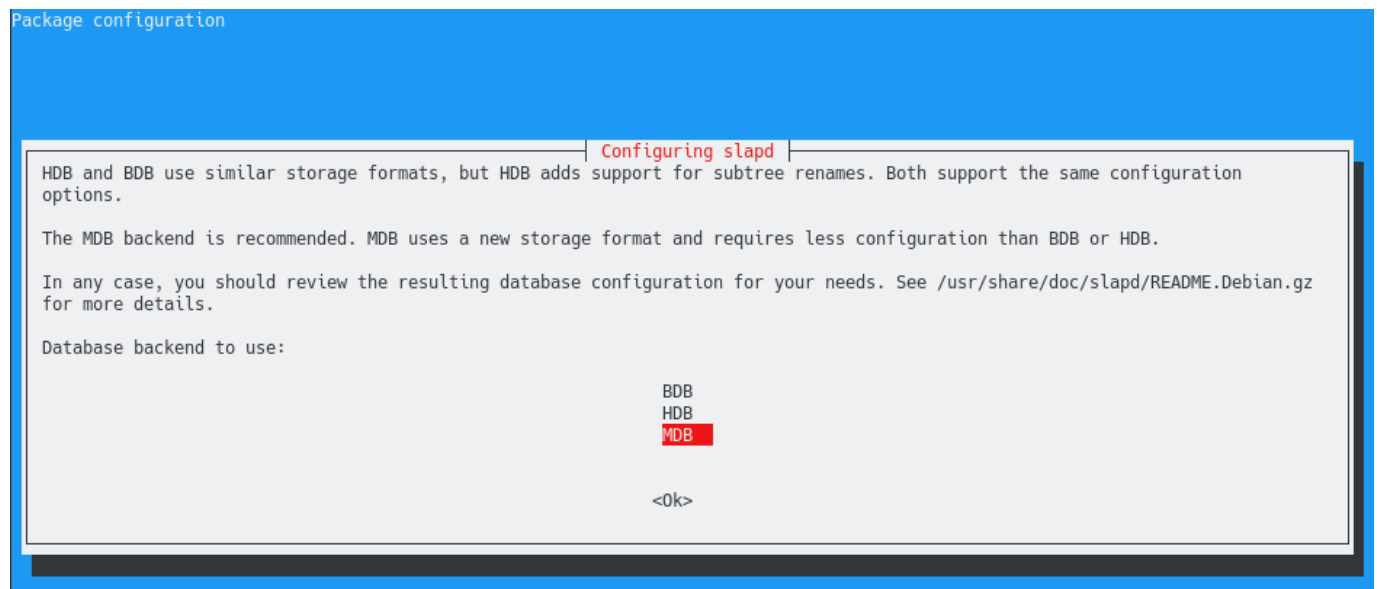
\*\*\*\*\*

<Ok>

On the next screen, you are asked to provide the back-end to be used by LDAP.



For this step, you want to keep the default values (meaning a MDB for MariaDB back - end) unless you have a reason to choose another storage backend.



Next, you are asked if you want the database to be removed when slapd is purged.

In this case, we will choose "No" : there are many situations where you simply want to update your slapd package or switch to a different LDAP server.

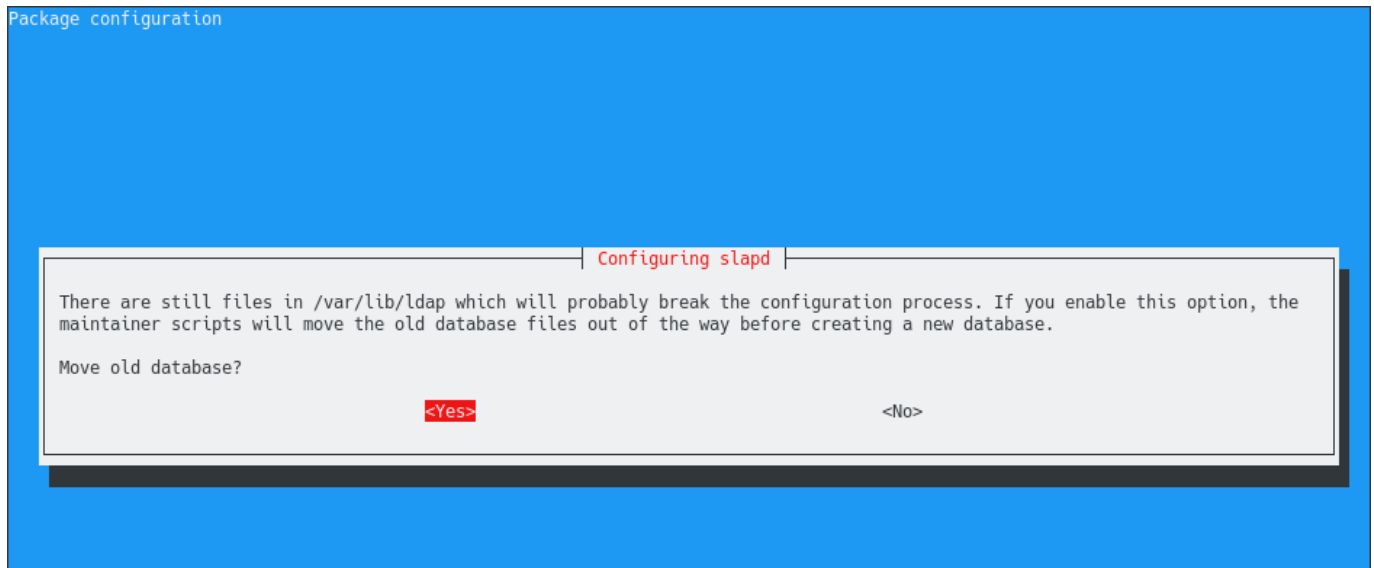
If you choose yes, your database will be removed which can be a real problem if you don't have any backups of your LDAP directory.



Finally, you are prompted with a warning : there are already some files sitting in the "/var/lib" directory of your server.

In this directory, you currently have your old database stored. As you are trying to reconfigure your OpenLDAP server, you will overwrite the content of this folder.

By choosing “Yes”, the slapd utility will backup the content of your existing database to the “/var/backups” folder.



Done!

Your slapd server is now configured properly to match your current DNS settings.

To have a first look at the content of your LDAP database, simply execute the “slapcat” (with sudo privileges if you are not currently logged as root)

```
$ sudo slapcat
```

```
debian-ldap@debian-ldap:~$ sudo slapcat
[sudo] password for debian-ldap:
dn: dc=devconnected,dc=com
objectClass: top
objectClass: dcObject
objectClass: organization
o: devconnected
dc: devconnected
structuralObjectClass: organization
entryUUID: 6336fd4a-ce52-1039-80e5-ffdbb5f0ddef
creatorsName: cn=admin,dc=devconnected,dc=com
createTimestamp: 20200118152439Z
entryCSN: 20200118152439.469541Z#000000#000#000000
modifiersName: cn=admin,dc=devconnected,dc=com
modifyTimestamp: 20200118152439Z

dn: cn=admin,dc=devconnected,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e1NTSEF9TTlpUEVEEdzNKSUlCS29oY2I2dTdZVGE3MmcvQLZ1Tk0=
structuralObjectClass: organizationalRole
entryUUID: 63377e96-ce52-1039-80e6-ffdbb5f0ddef
creatorsName: cn=admin,dc=devconnected,dc=com
createTimestamp: 20200118152439Z
entryCSN: 20200118152439.472875Z#000000#000#000000
modifiersName: cn=admin,dc=devconnected,dc=com
modifyTimestamp: 20200118152439Z
```

With this initial setup :

- Your configuration files are stored in `"/etc/ldap"` : they are storing the ldap configuration file, the schemas that you can add to slapd, as well as the slapd.d directory used for server customization;
- Your database is stored at `"/var/lib/ldap"` under the `"data.mdb"` database : you should probably setup backups of this file in order to make sure that you won't lose everything.

## Configuring firewall rules for LDAP

If you are using a firewall, it is very likely that you will need to accept inbound requests to your LDAP server.

As a quick reminder, OpenLDAP runs on port 389.

To make sure that it is running correctly, run the `"systemctl status"` command on the `"slapd"` server.

```
$ sudo systemctl status slapd
```

```

debian-ldap@debian-ldap:/var/lib/ldap$ sudo systemctl status slapd
● slapd.service - LSB: OpenLDAP standalone server (Lightweight Directory Access Protocol)
   Loaded: loaded (/etc/init.d/slapd; generated)
   Active: active (running) since Sat 2020-01-18 10:24:39 EST; 18min ago
     Docs: man:systemd-sysv-generator(8)
  Process: 6423 ExecStart=/etc/init.d/slapd start (code=exited, status=0/SUCCESS)
    Tasks: 3 (limit: 4915)
   Memory: 3.2M
    CGroup: /system.slice/slapd.service
            └─6430 /usr/sbin/slapd -h ldap:/// ldapi:/// -g openldap -u openldap -F /etc/ldap/slapd.d

```

If you are using recent distributions of Debian, you are probably using UFW as the default firewall.

To enable OpenLDAP on UFW, execute the “ufw allow” command on the port 389 of your server. You can accept TCP and UDP connections all together.

```
$ sudo ufw allow 389
```

```
Rule added
```

```
Rule added (v6)
```

You can then verify that the rule was correctly created using the status command.

```

debian-ldap@debian-ldap:/$ sudo ufw status
Status: active

To Action From
--
389 ALLOW Anywhere
389 (v6) ALLOW Anywhere (v6)

```

In the next section, we are going to see how you can easily add new entries to your LDAP server using LDIF files.

## Add OpenLDAP entries using LDIF files

When adding new entries to your OpenLDAP server, you could use the “slapadd” utility in order to add a new LDIF file.

However, this is not the utility that we are going to use, instead we are going to use “ldapadd”.

## Difference between slapadd and ldapadd

Before adding new entries, it is important for you to know the difference between slap utilities and ldap utilities.

Both utilities take LDIF formatted files as an argument and they had the content to the database.

However, when using slapadd, you will have to restart your LDAP server for the changes to be applied.

This is not the case when using ldap utilities such as "ldapadd" : modifications are directly performed on the directory tree.

In order to be able to use "ldapadd", "ldapsearch" and other LDAP utilities, you need to install the "ldap-utils" package on your system.

```
$ sudo apt-get install ldap-utils
```

## Creating your first LDIF file

As explained earlier, if you are using the console line, you will need to create LDIF files and add them to your current LDAP configuration or database.

The LDIF format is a format used in order to add or modify existing entries in a LDAP directory.

Using LDIF, you specify the distinguished name of the node that you want to modify and you describe the modifications to be performed.

As an example, let's say that you want to create a new node in your LDAP directory named "users".

## Adding a users group

To achieve that, create a new LDIF file named "users.ldif" and add the following content in it.

```
$ sudo touch /etc/ldap/users.ldif

# Content of the users file
dn: ou=People,dc=devconnected,dc=com objectClass: organizationalUnit
ou: People
```

As you can see, we are provided the complete DN of the node to be added, we specify the object class and the name of the node to be created.

In order to add this entry to your LDAP directory, you have to use the "ldapadd" command and specify the LDIF file to be used.

```
$ sudo ldapadd -D "cn=admin,dc=devconnected,dc=com" -W -H ldapi:/// -f users.ldif
```

Enter LDAP Password:

```
added new entry "ou=People,dc=devconnected,dc=com"
```

If you are not familiar with ldap utility options, here is a description of the options provided :

- -D : used to specify a node to bind to. When adding new entries to a LDAP server, you can choose your authentication mechanism but you usually want to bind to the admin node in order to gain all privileges on the tree;
- -W : used in order to specify that we want the password to be prompted when connecting;
- -H : used in order to specify the LDAP server to connect to. In this case, we are connecting to a LDAP server available at localhost;
- -f : to specify the LDIF file to be added to the LDAP server.

Note that you can not use an external authentication in order to add new entries to LDAP by default : ACL are not configured to do that.

Now that your node is added to your tree, you can try to find it using the "ldapsearch" command.

```
$ sudo ldapsearch -x -b "dc=devconnected,dc=com" ou
```

```
debian-ldap@debian-ldap:/etc/ldap$ ldapsearch -x -b "dc=devconnected,dc=com" ou
# extended LDIF
#
# LDAPv3
# base <dc=devconnected,dc=com> with scope subtree
# filter: (objectclass=*)
# requesting: ou
#
# devconnected.com
dn: dc=devconnected,dc=com

# admin, devconnected.com
dn: cn=admin,dc=devconnected,dc=com

# People, devconnected.com
dn: ou=People,dc=devconnected,dc=com
ou: People

# search result
search: 2
result: 0 Success

# numResponses: 4
# numEntries: 3
```

Great!

Now that the "People" organizational unit was added, let's add some users to your LDAP tree.

### Adding new users to LDAP

In order to add new users, we are going to follow the same logic : creating a LDIF file containing individual entries for users.

As described before, OpenLDAP uses schemas in order to define "objects" that can be added to the directory.

In this case, we are going to use the ["posixAccount" schema which is](#) already added to your database configuration by default.

The "posixAccount" object has several fields that can be used to describe a Linux user account such as the username, the surname but most importantly the user password. Create a new LDIF file and add the following content in it :

```
$ sudo touch /etc/ldap/new\_users.ldif
```

- Content of new\\_users LDIF file

```
dn: cn=john,ou=People,dc=devconnected,dc=com
objectClass: top
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: person
cn: john
uid: john
uidNumber: 10001
gidNumber: 10001
homeDirectory: /home/john
userPassword: <password>
loginShell: /bin/bash
```

When you are done, save your file and use the `ldapadd` command in order to add your entry to the LDAP directory tree.

```
$ sudo ldapadd -D "cn=admin,dc=devconnected,dc=com" -W -H ldapi:/// -f new\_users.ldif
```

```
Enter LDAP Password:
added new entry "cn=john,ou=People,dc=devconnected,dc=com"
```

Congratulations, you now have your first user stored in OpenLDAP.

You can read the user information by issuing a LDAP search command. Note that you won't be able to read the user password as you are restricted by ACLs.

```
$ sudo ldapsearch -x -b "ou=People,dc=devconnected,dc=com"
```



```
# People, devconnected.com
dn: ou=People,dc=devconnected,dc=com
objectClass: organizationalUnit
ou: People

# john, People, devconnected.com
dn: cn=john,ou=People,dc=devconnected,dc=com
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
cn: john
uid: john
uidNumber: 10001
gidNumber: 10001
homeDirectory: /home/john
loginShell: /bin/bash
```

Awesome, now that your LDAP server is configured, let's configure a client in order to configure central authentication.

## Configuring LDAP clients for centralized authentication

In the last section of this OpenLDAP server setup, we are going to see how you can configure LDAP clients (i.e your host machines) in order for them to connect using LDAP information.

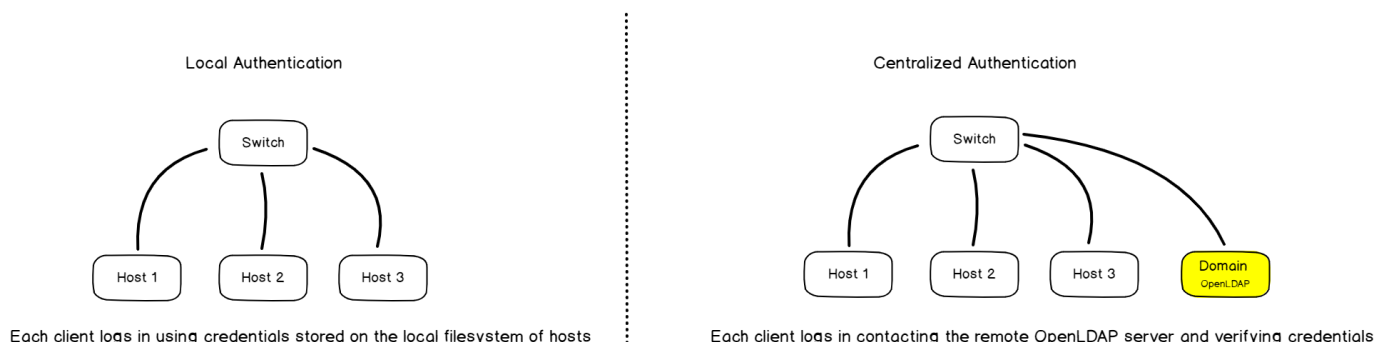
### How LDAP client authentication works

Before issuing any commands, it is important for you to have a global understanding of what we are actually building.

Before LDAP (and NIS), if you wanted to configure users and group permissions over multiple computers of a network, you would have to connect to them one by one and change their settings.

LDAP comes as a great solution for this : LDAP will centralize user information in one single place on your network.

### Client Authentication over Networks



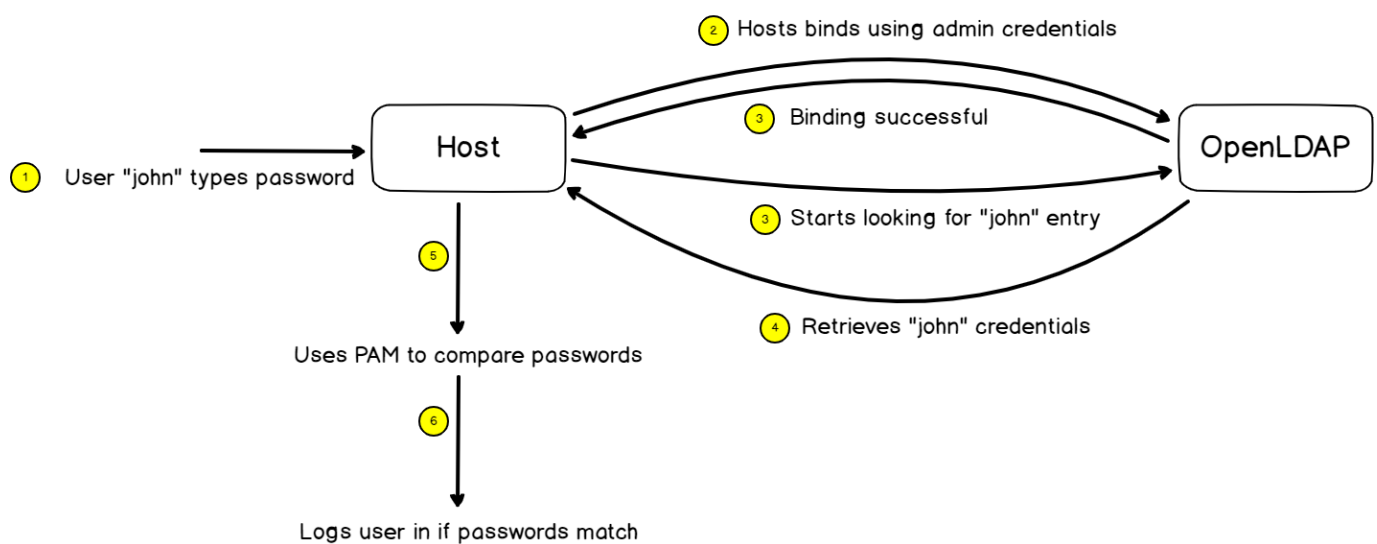
When a client connects to any machine of your domain, the host will first contact the LDAP server and verify that the user password provided is correct.

The client library will bind (or authenticate) to the remote LDAP server using the admin account and retrieve the information associated with the user trying to connect.

Next, it will retrieve the password associated with the account and compare it with the password you typed when you logged in.

If the passwords match, you will be logged in your account, otherwise you will be denied.

### How clients connect using OpenLDAP



### Setup Client LDAP authentication on Debian

In order to setup client LDAP authentication, you will need to install the "libnss-ldap" package on your client.

```
$ sudo apt-get install libnss-ldap
```

When installing this package, you will be prompted with many different questions in order to configure client centralized authentication.

First, you are asked to provide the URL of your LDAP server : it is recommended to setup an IP address (configured as static obviously) in order to avoid problems in DNS resolutions.

On the server, [identify your IP address with the ip command](#) and fill the corresponding field on the client.

- On the server
- `$ ip a`

#### Package configuration

##### Configuring ldap-auth-config

Please enter the URI of the LDAP server to use. This is a string in the form of `ldap://<hostname or IP>:<port>/`. `ldaps://` or `ldapi://` can also be used. The port number is optional.

Note: It is usually a good idea to use an IP address because it reduces risks of failure in the event name service problems.

LDAP server Uniform Resource Identifier:

`ldap://192.168.178.29/`

<Ok>

Note : make sure that you are using the LDAP protocol and not the LDAPI protocol. For some reason, your server won't be reachable if you use the LDAPI protocol.

Next, you are asked to provide the root distinguished name of your LDAP server. If you are not sure, you should run a `ldapsearch` command on the server to get this information.

## Package configuration

### Configuring ldap-auth-config

Please enter the distinguished name of the LDAP search base. Many sites use the components of their domain names for this purpose. For example, the domain "example.net" would use "dc=example,dc=net" as the distinguished name of the search base.

Distinguished name of the search base:

dc=devconnected,dc=com

<Ok>

On the next screen, you are asked the LDAP version that you want to use : choose the LDAP version 3 for now.

## Package configuration

### Configuring ldap-auth-config

Please enter which version of the LDAP protocol should be used by ldapns. It is usually a good idea to set this to the highest available version.

LDAP version to use:

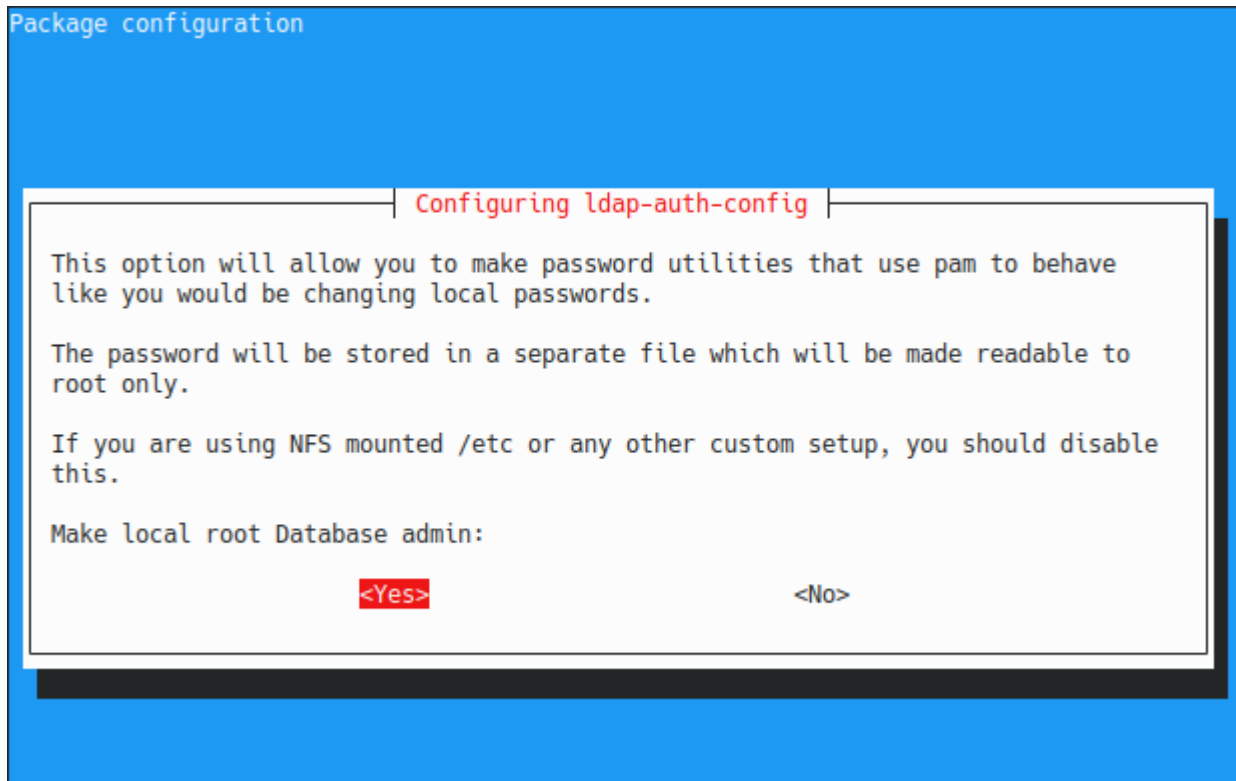
3  
2

<Ok>

Next, you are asked if you want to make the local root the database admin.

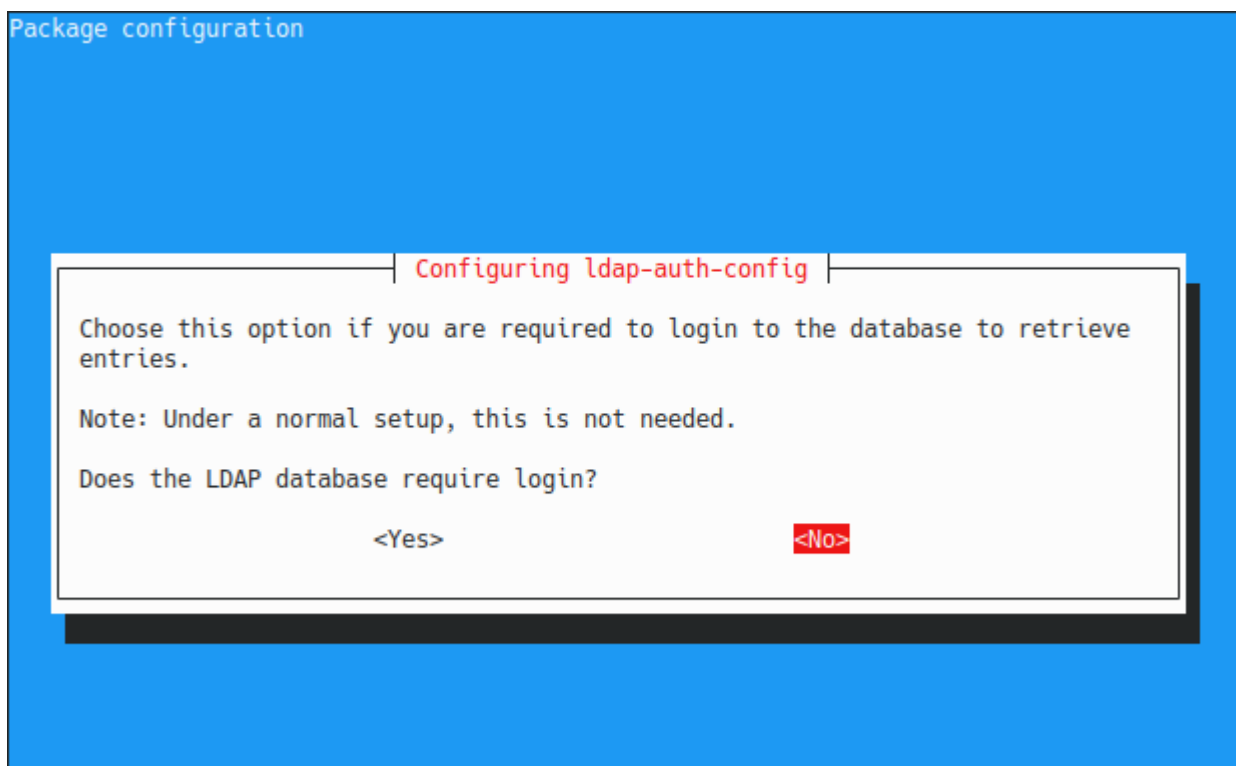
You want to type "Yes" to this option as you want to change the user password directly from the host machine.

With this option, you will be able to run the “passwd” and have the password modified directly in the LDAP directory, which is pretty useful.



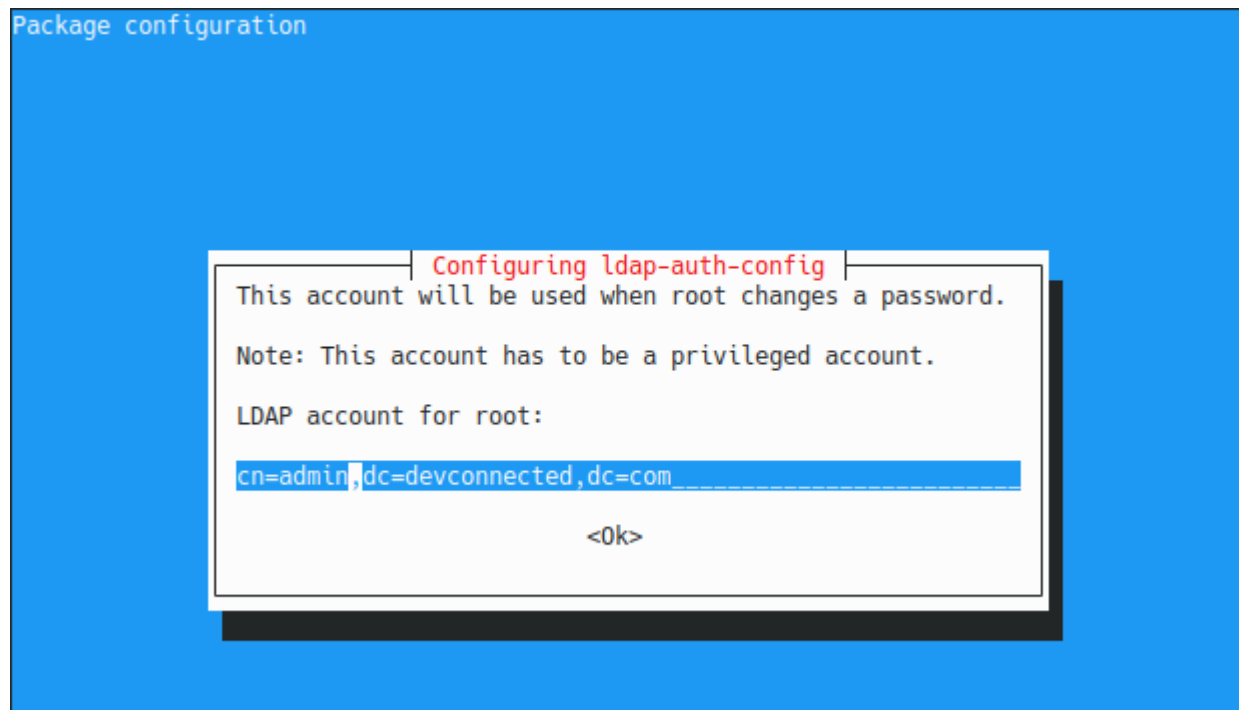
By default, the LDAP database does not require a login, so you can type “No” on this option.

Note : the LDAP database has no login but you have an admin account at the top of your LDAP directory. Those are two different concepts that are very different one from another.

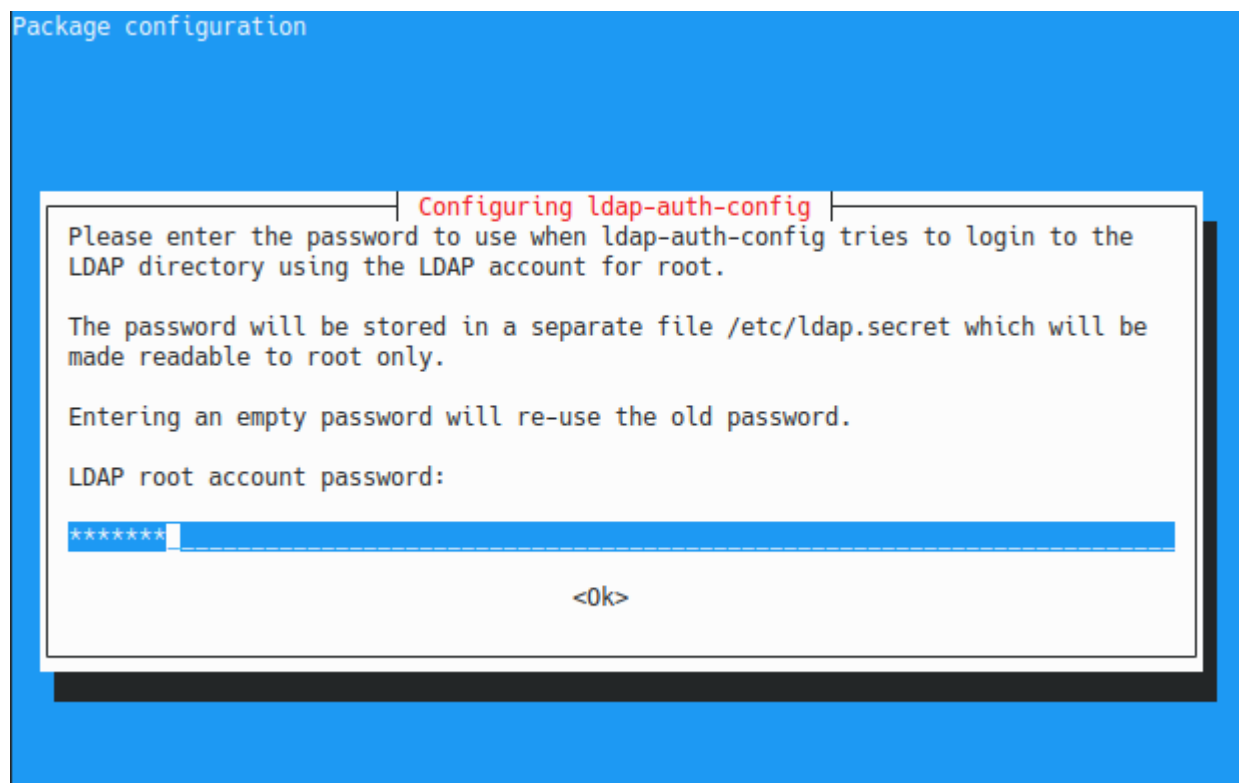


Next, type the LDAP administrator account to be used for bindinds.

As a reminder, this is the account that will be used in order to get the user password information from the server.



Finally, type the password associated with the admin account on the LDAP server.



Done, you should now be able to query your LDAP server.

## Linking client information to LDAP

In order to link your client information (such as username and password) to the LDAP directory, you need to modify the nsswitch file.

As a reminder, the nsswitch file is used in order to link some information on your system (such as users, groups or hosts) to various different sources (local, LDAP, NIS or others).

Edit the /etc/nsswitch.conf file and add a "ldap" entry to the first four sections : passwd, group, shadow, gshadow.

```
$ sudo nano /etc/nsswitch.conf
```

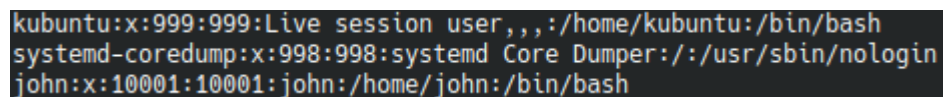


```
GNU nano 4.3 /etc/nsswitch.conf
/etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc-reference' and `info' packages installed, try:
# `info libc "Name Service Switch"' for information about this file.

passwd:    files systemd ldap
group:     files systemd ldap
shadow:    files ldap
gshadow:   files ldap
```

Save your file and you should now be able to list users from the LDAP server. \$ getent passwd

Note : if you are not familiar with the getent command, here are all the commands used to [list users on Linux](#).



```
kubuntu:x:999:999:Live session user,,,:/home/kubuntu:/bin/bash
systemd-coredump:x:998:998:systemd Core Dumper:/:usr/sbin/nologin
john:x:10001:10001:john:/home/john:/bin/bash
```

Awesome!

Now that your user can be retrieved via LDAP, you will be able to log to this account by using the user password you have specified in the LDAP directory.

```
$ su - john

<Type password specified in LDAP>
john@client:/home/john
```

To generate the home directory automatically, follow the next steps:

Install `libpam-ldap`

```
apt install libpam-ldap
```

Add the following line to the file /etc/pam.d/common-session

```
session optional pam_mkhomedir.so skel=/etc/skel umask=077
```

To log-in into the desktop enviroment we need one more steep, install `nsldap` and the configuration should be done

```
apt install nsldap
```

## Example config file

Config file of aragon.local

Grupos	Nombres de usuario
Competidores	competidor01 competidor39
tutores	tutor01 tutor39
gestores	gestor01 gestor39



## # Organizational Units

```
dn: ou=People,dc=aragon,dc=local
objectClass: organizationalUnit
ou: People
```

```
dn: ou=Groups,dc=aragon,dc=local
objectClass: organizationalUnit
ou: Groups
```

## # Groups

```
dn: cn=competidores,ou=Groups,dc=aragon,dc=local
objectClass: top
objectClass: posixGroup
gidNumber: 2000
cn: competidores
```

```
dn: cn=tutores,ou=Groups,dc=aragon,dc=local
objectClass: top
objectClass: posixGroup
gidNumber: 3000
cn: tutores
```

```
dn: cn=gestion,ou=Groups,dc=aragon,dc=local
objectClass: top
objectClass: posixGroup
gidNumber: 4000
cn: gestion
```

## # Users

```
dn: uid=competidor01,ou=People,dc=aragon,dc=local
objectClass: top
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: person
uid: competidor01
cn: competidor01
sn: competidor01
uidNumber: 10001
gidNumber: 2000
homeDirectory: /home/competidor39
userPassword: IVSZ2e12
loginShell: /bin/bash
```

```
dn: uid=competidor39,ou=People,dc=aragon,dc=local
objectClass: top
objectClass: inetOrgPerson
objectClass: posixAccount
```

```
objectClass: person
uid: competidor39
cn: competidor39
sn: competidor39
uidNumber: 10039
gidNumber: 2000
homeDirectory: /home/competidor39
userPassword: IVSZ2e12
loginShell: /bin/bash
```

```
dn: uid=tutor01,ou=People,dc=aragon,dc=local
objectClass: top
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: person
uid: tutor01
cn: tutor01
sn: tutor01
uidNumber: 20001
gidNumber: 3000
homeDirectory: /home/tutor01
userPassword: IVSZ2e12
loginShell: /bin/bash
```

```
dn: uid=tutor39,ou=People,dc=aragon,dc=local
objectClass: top
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: person
uid: tutor39
cn: tutor39
sn: tutor39
uidNumber: 20039
gidNumber: 3000
homeDirectory: /home/tutor39
userPassword: IVSZ2e12
loginShell: /bin/bash
```

```
dn: uid=gestor01,ou=People,dc=aragon,dc=local
objectClass: top
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: person
uid: gestor01
cn: gestor01
sn: gestor01
uidNumber: 30001
gidNumber: 4000
homeDirectory: /home/gestor01
userPassword: IVSZ2e12
loginShell: /bin/bash
```

```
dn: uid=gestor39,ou=People,dc=aragon,dc=local
objectClass: top
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: person
uid: gestor39
cn: gestor39
sn: gestor39
uidNumber: 30039
gidNumber: 4000
homeDirectory: /home/gestor39
userPassword: IVSZ2e12
loginShell: /bin/bash
```

## Script to remove all structure aragon.local

---

```
ldapdelete -x -D cn=admin,dc=aragon,dc=local -W uid=competidor01,ou=People,dc=aragon,dc=local
ldapdelete -x -D cn=admin,dc=aragon,dc=local -W uid=competidor39,ou=People,dc=aragon,dc=local
ldapdelete -x -D cn=admin,dc=aragon,dc=local -W uid=tutor01,ou=People,dc=aragon,dc=local
ldapdelete -x -D cn=admin,dc=aragon,dc=local -W uid=tutor39,ou=People,dc=aragon,dc=local
ldapdelete -x -D cn=admin,dc=aragon,dc=local -W uid=gestor01,ou=People,dc=aragon,dc=local
ldapdelete -x -D cn=admin,dc=aragon,dc=local -W uid=gestor39,ou=People,dc=aragon,dc=local

ldapdelete -x -D cn=admin,dc=aragon,dc=local -W cn=competidores,ou=Groups,dc=aragon,dc=local
ldapdelete -x -D cn=admin,dc=aragon,dc=local -W cn=tutores,ou=Groups,dc=aragon,dc=local
ldapdelete -x -D cn=admin,dc=aragon,dc=local -W cn=gestion,ou=Groups,dc=aragon,dc=local

ldapdelete -x -D cn=admin,dc=aragon,dc=local -W ou=People,dc=aragon,dc=local
ldapdelete -x -D cn=admin,dc=aragon,dc=local -W ou=Groups,dc=aragon,dc=local
```

## Config sshd LDAP auth

---

First we have to config LDAP Auth on the machine we want to access over ssh with LDAP users

Modifi the `/etc/ssh/sshd_config` file and edit the following line

```
PAMAuthenticationViaKbdInt yes
```

Restart the service and done

```
systemctl restart sshd
```

# Apache2

---

## Installation

---

Install the package `apache2`

```
apt install apache2
```

## Configuration files

---

The configuration folder is located in the following folder `/etc/apache2`

### Explanation of the configuration files

- **apache2.conf:** This is the main configuration file for the server. Almost all configuration can be done from within this file, although it is recommended to use separate, designated files for simplicity. This file will configure defaults and be the central point of access for the server to read configuration details.
- **ports.conf:** This file is used to specify the ports that virtual hosts should listen on. Be sure to check that this file is correct if you are configuring SSL.
- **conf.d/:** This directory is used for controlling specific aspects of the Apache configuration. For example, it is often used to define SSL configuration and default security choices.
- **sites-available/:** This directory contains all of the virtual host files that define different web sites. These will establish which content gets served for which requests. These are available configurations, not active configurations.
- **sites-enabled/:** This directory establishes which virtual host definitions are actually being used. Usually, this directory consists of symbolic links to files defined in the "sites-available" directory.
- **mods-[enabled,available]/:** These directories are similar in function to the sites directories, but they define modules that can be optionally loaded instead.

## Global Configuration Section

---

### Timeout

By default, this parameter is set to "300", which means that the server has a maximum of 300 seconds to fulfill each request.

This is probably too high for most set ups and can safely be dropped to something between 30 and 60 seconds.

## KeepAlive

This option, if set to "On", will allow each connection to remain open to handle multiple requests from the same client.

If this is set to "Off", each request will have to establish a new connection, which can result in significant overhead depending on your setup and traffic situation.

## MaxKeepAliveRequests

This controls how many separate request each connection will handle before dying. Keeping this number high will allow Apache to serve content to each client more effectively.

Setting this value to 0 will allow Apache to serve an unlimited amount of request for each connection.

## KeepAliveTimeout

This setting specifies how long to wait for the next request after finishing the last one. If the timeout threshold is reached, then the connection will die.

This just means that the next time content is requested, the server will establish a new connection to handle the request for the content that make up the page the client is visiting.

## Enabling Sites and Modules in Apache

---

Once you have a Virtual Host file that meets your requirements, you can use the tools included with Apache to transition them into live sites.

To automatically create a symbolic link in the "sites-enabled" directory to an existing file in the "sites-available" directory, issue the following command:

```
a2ensite website_config_file
```

After enabling a site, issue the following command to tell Apache to re-read its configuration files, allowing the change to propagate:

```
service apache2 reload
```

There is also a companion command for disabling a Virtual Host. It operates by removing the symbolic link from the "sites-enabled" directory:

```
a2dissite website_config_file
```

Again, reload the configuration to make the change happen:

```
service apache2 reload  
or  
systemctl reload apache2
```

Modules can be enabled or disabled by using the "a2enmod" and "a2dismod" commands respectively. They work in the same way as the "site" versions of these commands.

Remember to reload your configuration changes after modules have been enabled or disabled as well.

## SSL Module

---

```
a2enmod ssl
```

## Redirect http request to https

```
<IfModule mod_ssl.c>  
    <VirtualHost 127.0.0.1:80>  
        ServerName midominio.com  
        Redirect / https://midominio.com/  
    </VirtualHost>  
    ...  
</IfModule>
```

## LDAP Module

---

Enable the module

```
a2enmod authnz_ldap
```

Config directory `/var/web/intranet` to be access by an ldap valid user and password

```
<Directory /var/web/intranet>
    Options Indexes FollowSymLinks
    AllowOverride None

    AuthName "LDAP Authentication"
    AuthType Basic
    AuthBasicProvider ldap
    AuthLDAPURL ldap://ldap.aragon.local/dc=aragon,dc=local?uid?sub?
(objectClass=*)
    Require valid-user
</Directory>
```

## The Require Directives

### Require ldap-user

The Require ldap-user directive specifies what usernames can access the resource.

```
Require ldap-user "Barbara Jenson"
Require ldap-user "Fred User"
Require ldap-user "Joe Manager"
```

If the uid attribute was used instead of the cn attribute in the URL above, the above three lines could be condensed to

```
Require ldap-user bjenson fuser jmanager
```

### Require ldap-group

This directive specifies an LDAP group whose members are allowed access.

```
dn: cn=Administrators, o=Example
objectClass: groupOfUniqueNames
uniqueMember: cn=Barbara Jenson, o=Example
uniqueMember: cn=Fred User, o=Example
```

## Restrict folders to Hosts

---

### Require ip



The ip provider allows access to the server to be controlled based on the IP address of the remote client. When `Require ip ip-address` is specified, then the request is allowed access if the IP address matches.

A full IP address:

```
Require ip 10.1.2.3
Require ip 192.168.1.104 192.168.1.205
```

An IP address of a host allowed access

A partial IP address:

```
Require ip 10.1
Require ip 10 172.20 192.168.2
```

The first 1 to 3 bytes of an IP address, for subnet restriction.

A network/netmask pair:

```
Require ip 10.1.0.0/255.255.0.0
```

A network a.b.c.d, and a netmask w.x.y.z. For more fine-grained subnet restriction.

A network/nnn CIDR specification:

```
Require ip 10.1.0.0/16
```

## Require host

The host provider allows access to the server to be controlled based on the host name of the remote client. When `Require host host-name` is specified, then the request is allowed access if the host name matches.

A (partial) domain-name

```
Require host example.org
Require host .net example.edu
```

## Require local

The local provider allows access to the server if any of the following conditions is true:

the client address matches 127.0.0.0/8 the client address is ::1 both the client and the server address of the connection are the same This allows a convenient way to match connections that originate from the local host:

```
Require local
```

## Authorization Containers

The authorization container directives `<RequireAll>`, `<RequireAny>` and `<RequireNone>` may be combined with each other and with the `Require` directive to express complex authorization logic.

The example below expresses the following authorization logic. In order to access the resource, the user must either be the superadmin user, or belong to both the admins group and the Administrators LDAP group and either belong to the sales group or have the LDAP dept attribute sales. Furthermore, in order to access the resource, the user must not belong to either the temps group or the LDAP group Temporary Employees.

```
<Directory "/www/mydocs">
  <RequireAll>
    <RequireAny>
      Require user superadmin
    <RequireAll>
      Require group admins
      Require ldap-group "cn=Administrators,o=Airius"
    <RequireAny>
      Require group sales
      Require ldap-attribute dept="sales"
    </RequireAny>
  </RequireAll>
</RequireAny>
<RequireNone>
  Require group temps
  Require ldap-group "cn=Temporary Employees,o=Airius"
</RequireNone>
</RequireAll>
</Directory>
```

## Configuration examples

[www.aragon.local](http://www.aragon.local)

## Parte01

- Sitio web público, debe ser accesible desde cualquier sitio, pero sólo mediante el protocolo HTTPS, por lo que las peticiones HTTP serán redirigidas a HTTPS.
- Utilizará el primer certificado creado en la tarea anterior (su "Common Name" es [www.CCAA.com](http://www.CCAA.com)).
- Cuando se accede a un directorio del sitio el recurso que se abrirá automáticamente será "main.html". Si no existe ese recurso, se abrirá "main2.html". Si no existiese ninguno de los dos, mostrará el contenido del directorio.
- Este sitio estará en la ruta /var/web/www. Crea los directorios y archivos necesarios para demostrar que funciona.

## Parte02

- Cada usuario LDAP puede tener su propia página web, que estará ubicada en la ruta /home/ldap/usuario/www.
- Mediante navegador web, se accederá a ella utilizando la ruta [www.CCAA.com/~usuario](http://www.CCAA.com/~usuario).
- Sólo será accesible desde la red 192.168.z.0/24.
- Cuando se accede a un directorio del sitio el recurso que se abrirá automáticamente será "public.html". Si no existe ese recurso, no se mostrará el contenido del directorio.

```

<IfModule mod_ssl.c>
    <VirtualHost www.aragon.local:443>
        ServerAdmin webmaster@localhost

        DocumentRoot /var/web/www

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        SSLEngine on

        SSLCertificateFile      /root/certs/www/www.aragon.local.crt
        SSLCertificateKeyFile /root/certs/www/www.aragon.local.key

        <Directory /var/web/www>
            Options Indexes FollowSymLinks
            AllowOverride None
            Require all granted
            DirectoryIndex main.html main2.html
        </Directory>

        UserDir /home/ldap/*/www

        <Directory /home/ldap>
            Options -Indexes +FollowSymLinks
            AllowOverride None
            Require all granted
            Require ip 192.168.30.0/24
            DirectoryIndex public.html
        </Directory>
    </VirtualHost>

    <VirtualHost www.aragon.local:80>
        ServerName www.aragon.local
        Redirect / https://www.aragon.local/
    </VirtualHost>
</IfModule>

```

## intranet.aragon.local

- Sitio web que requiere autenticación contra el servidor LDAP de la máquina ldap. Sólo estará accesible mediante el protocolo HTTPS, por lo que las peticiones HTTP serán redirigidas a HTTPS.
- Utilizará el segundo certificado creado en la tarea anterior (su "Common Name" es intranet.CCAA.com).

- Dentro de la estructura del sitio web, debe haber un directorio llamado "private" al que sólo se podrá acceder desde la máquina ldap y desde la propia máquina server.
- Este sitio estará en la ruta /var/web/intranet. Crea los directorios y archivos necesarios para demostrar que funciona.

```
<IfModule mod_ssl.c>
    <VirtualHost intranet.aragon.local:443>
        ServerAdmin webmaster@localhost

        DocumentRoot /var/web/intranet

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        SSLEngine on

        SSLCertificateFile      /root/certs/intranet/intranet.aragon.local.crt
        SSLCertificateKeyFile /root/certs/intranet/intranet.aragon.local.key

        <Directory /var/web/intranet>
            Options Indexes FollowSymLinks
            AllowOverride None

            AuthName "LDAP Authentication"
            AuthType Basic
            AuthBasicProvider ldap
            AuthLDAPURL ldap://ldap.aragon.local/dc=aragon,dc=local?uid?
sub?(objectClass=*)
            Require valid-user
        </Directory>
        <Directory /var/web/intranet/private>

            Require ip 192.168.30.98 192.168.30.99
        </Directory>
    </VirtualHost>

    <VirtualHost intranet.aragon.local:80>
        ServerName intranet.aragon.local
        Redirect / https://intranet.aragon.local:443/
    </VirtualHost>
</IfModule>
```

---

## Preparations

---

Change DNS setting to refer to Active Directory

```
nano /etc/resolv.conf

nameserver 10.0.0.100 # IP of your AD
```

## Install packages

---

```
apt -y install realmd sssd sssd-tools adcli krb5-user packagekit samba-common
samba-common-bin samba-lsbs
```

If DHCP server is running, select [Yes], if not, select [No]

```
+-----+ Samba server and utilities +-----+
|
| If your computer gets IP address information from a DHCP server on the
| network, the DHCP server may also provide information about WINS servers
| ("NetBIOS name servers") present on the network. This requires a change
| to your smb.conf file so that DHCP-provided WINS settings will
| automatically be read from /var/lib/samba/dhcp.conf.
|
| The dhcp-client package must be installed to take advantage of this
| feature.
|
| Modify smb.conf to use WINS settings from DHCP?
|
|                                     <Yes>                                     <No>
|
+-----+
```

Specify Realm Name

```

+-----+ Configuring Kerberos Authentication +-----+
| When users attempt to use Kerberos and specify a principal or user name |
| without specifying what administrative Kerberos realm that principal    |
| belongs to, the system appends the default realm. The default realm may |
| also be used as the realm of a Kerberos service running on the local    |
| machine. Often, the default realm is the uppercase version of the local  |
| DNS domain.                                                              |
|                                                                           |
| Default Kerberos version 5 realm:                                       |
|                                                                           |
| SRV.WORLD_____                                                        |
|                                                                           |
|                               <Ok>                                       |
+-----+

```

## Specify AD DS Hostname

```

+-----+ Configuring Kerberos Authentication +-----+
| Enter the hostnames of Kerberos servers in the SRV.WORLD Kerberos realm |
| separated by spaces.                                                    |
|                                                                           |
| Kerberos servers for your realm:                                       |
|                                                                           |
| fd3s.srv.world_____                                                  |
|                                                                           |
|                               <Ok>                                       |
+-----+

+-----+ Configuring Kerberos Authentication +-----+
| Enter the hostname of the administrative (password changing) server for |
| the SRV.WORLD Kerberos realm.                                          |
|                                                                           |
| Administrative server for your Kerberos realm:                         |
|                                                                           |
| fd3s.srv.world_____                                                  |
|                                                                           |
|                               >Ok<                                       |
+-----+

```

## Join in Active Directory Domain.

---

```
realm join serverdc.tutorialesit.com -U 'Administrador' -v
```

## Config auto home folder creation

---

Command:

```
pam-auth-update
```

Check mkhomedir.

To finish we have to restart the sssd service

```
sudo systemctl restart sssd
```



# Unir un cliente Ubuntu a un dominio de Active Directory

Para agregar este nuevo registro simplemente tenéis que editar el fichero hosts.

```
sudo nano /etc/hosts
```

Agregar el siguiente registro que básicamente consistirá en relacionar la ip del servidor de dominio con su nombre.

```
127.0.0.1 localhost
127.0.1.1 server1

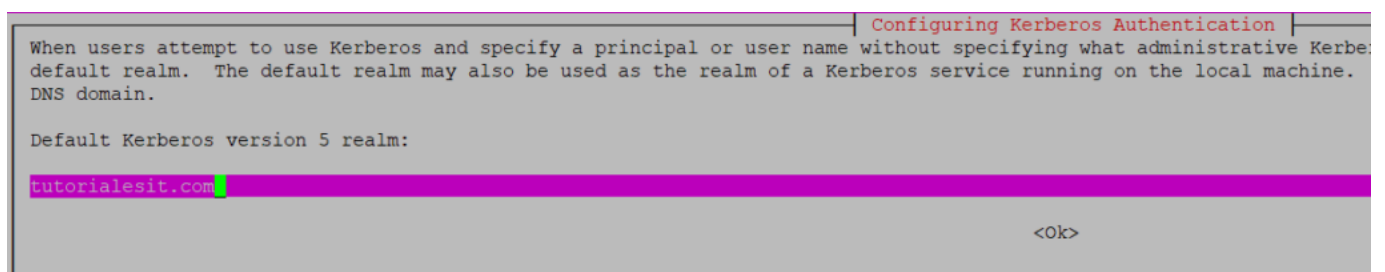
# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe00::0  ip6-localnet
ff00::0  ip6-mcastprefix
ff02::1  ip6-allnodes
ff02::2  ip6-allrouters

192.168.126.200 serverdc.tutorialesit.com
```

## Instalación de los paquetes necesarios

```
sudo apt -y install realmd sssd sssd-tools samba-common krb5-user packagekit
samba-common-bin samba-libs adcli ntp
```

En el transcurso de la instalación se nos preguntará por el nombre del dominio al que nos queremos unir así que lo especificamos y pulsamos en OK para continuar.



A continuación nos preguntará por el nombre del servidor donde tenemos desplegado Active Directory, lo especificamos y continuamos con el proceso de instalación.

Configuring Kerberos Authentication

Enter the hostnames of Kerberos servers in the tutorialsit.com Kerberos realm separated by spaces.

Kerberos servers for your realm:

serverdc.tutorialsit.com

<Ok>

Cuando nos pregunte por el servidor administrativo, en la siguiente pantalla, nuevamente pondremos el nombre de nuestro servidor de dominio.

Configuring Kerberos Authentication

Enter the hostname of the administrative (password changing) server for the tutorialsit.com Kerberos realm.

Administrative server for your Kerberos realm:

serverdc.tutorialsit.com

<Ok>

## Unir el cliente Ubuntu a nuestro dominio

Una vez hemos instalado y configurado los paquetes necesarios el siguiente paso será unir nuestro cliente de Ubuntu a nuestro dominio para ello utilizaremos el siguiente comando **(recordar cambiar el nombre del servidor del ejemplo por el vuestro)**.

```
sudo realm join serverdc.tutorialsit.com -U 'Administrador' -v
```

## Configuración inicio sesión usuarios

```
sudo pam-auth-update
```

Marcamos la opción `activate mkhomedir` y pulsamos OK.

Por último para que el cambio de configuración que hemos hecho surta efecto reiniciamos el servicio `sssd`.

```
sudo systemctl restart sssd
```

# Config Fail2Ban for Apache2

---

## Install the package

---

```
apt install fail2ban
```

## Show the status of the active rules

---

```
fail2ban-client status
```

## Block error 403 (Forbidden)

---

Create a file on `/etc/fail2ban/jail.d` named `apache-forbidden.conf` and write the following:

```
[apache-forbidden]
enabled = true
filter = apache-forbidden
logpath = /var/log/apache2/access.log
bantime = 2m
maxretry = 3
findtime = 2m
port = 80,443
banaction = iptables-multiport
```

In folder `/etc/fail2ban/filter.d` create a custom filter named `apache-forbidden.conf`

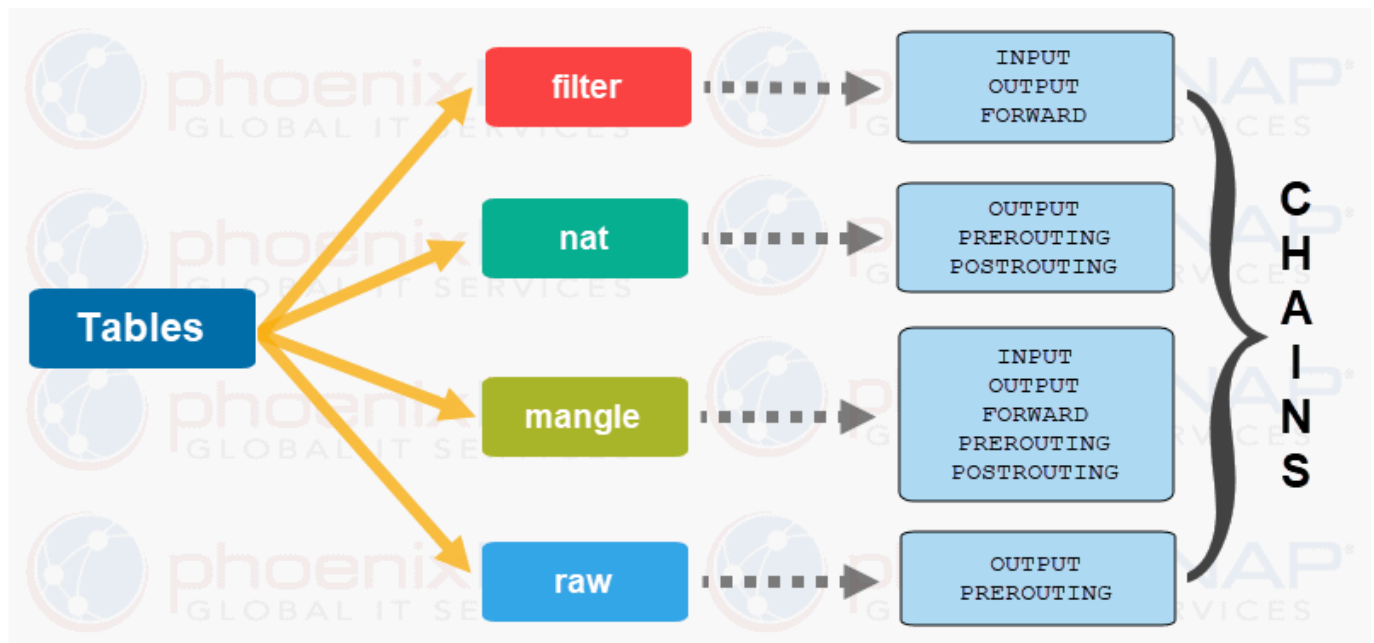
```
[Definition]
failregex = <HOST> - - .*HTTP/[0-9]+(.[0-9]+)?" 403
```

This will block for 2 minutes the hosts who try to access 3 times in 2 minutes to a content that is not allowed with error 403 (Forbidden)

Once we have created the config file and the custom filter we have to restart the service to activate the new rule

```
systemctl restart fail2ban
```

# Iptables Linux



## Commands

- Show actual config `iptables -nvL`
- Change policy `iptables -P <INPUT, OUTPUT, FORWARD> <DROP, ACCEPT, RETURN>`

## ICMP

Allow ICMP from WAN interface

```
iptables -A INPUT -p icmp -j ACCEPT
```

Allow ICMP between LANs

```
iptables -t nat POSTROUTING -p icmp -o <OUTPUT interface> -s <source network IP 192.168.40.0/24> -d <destination network IP 192.168.30.0/24> -j MASQUERADE
```

## NAT

Activate IP-forwarding in the kernel uncommenting the following line in `/etc/sysctl.conf`

```
net.ipv4.ip_forward=1
```

Make the nat rules:

```
iptables -t nat -A POSTROUTING -o ens18 -j MASQUERADE  
iptables -A FORWARD -m state --state ESTABLISH,RELATED,NEW -f ACCEPT
```

## Enable services on DMZ or LAN to be accesed from WAN

HTTP

```
iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to-destination  
192.168.202.105:80  
iptables -t nat -A POSTROUTING -p tcp -d 192.168.202.105 --dport 80 -j MASQUERADE
```

## Defining Chain Rules

Defining a rule means appending it to the chain. To do this, you need to insert the **-A** option (**Append**) right after the iptables command, like so:

```
iptables -A
```

It will alert iptables that you are adding new rules to a chain. Then, you can combine the command with other options, such as:

- **i (interface)** — the network interface whose traffic you want to filter, such as eth0, lo, ppp0, etc.
- **p (protocol)** — the network protocol where your filtering process takes place. It can be either **tcp**, **udp**, **udplite**, **icmp**, **sctp**, **icmpv6**, and so on. Alternatively, you can type **all** to choose every protocol.
- **s (source)** — the address from which traffic comes from. You can add a hostname or IP address. – **dport (destination port)** — the destination port number of a protocol, such as 22 (SSH), 443 (https), etc.
- **j (target)** — the target name (**ACCEPT**, **DROP**, **RETURN**). You need to insert this every time you make a new rule.

If you want to use all of them, you must write the command in this order:

```
iptables -A <chain> -i <interface> -p <protocol (tcp/udp) > -s <source> --dport <port no.> -j <target>
```

Once you understand the basic syntax, you can start configuring the firewall to give more security to your server. For this iptables tutorial, we are going to use the INPUT chain as an example.

## Enabling Traffic on Localhost

---

To allow traffic on localhost, type this command:

```
iptables -A INPUT -i lo -j ACCEPT
```

For this iptables tutorial, we use lo or loopback interface. It is utilized for all communications on the localhost. The command above will make sure that the connections between a database and a web application on the same machine are working properly.

## Enabling Connections on HTTP, SSH, and SSL Port

---

Next, we want http (port 80), https (port 443), and ssh (port 22) connections to work as usual. To do this, we need to specify the protocol (-p) and the corresponding port (--dport). You can execute these commands one by one:

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -p tcp --dport 443 -j ACCEPT
```

## Filtering Packets Based on Source

---

Iptables allows you to filter packets based on an IP address or a range of IP addresses. You need to specify it after the -s option. For example, to accept packets from 192.168.1.3, the command would be:

```
iptables -A INPUT -s 192.168.1.3 -j ACCEPT
```

If you want to drop packets from a range of IP addresses, you have to use the -m option and iprange module. Then, specify the IP address range with --src-range. Remember, a hyphen

should separate the range of ip addresses without space, like this:

```
iptables -A INPUT -m iprange --src-range 192.168.1.100-192.168.1.200 -j DROP
```

## Allow internet Access when filter chain is on DROP

Create the following rules to allow access to internet:

```
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state NEW,ESTABLISHED -j ACCEPT
```

## Deleting Rules

If you want to remove all rules and start with a clean slate, you can use the **-F** option (**flush**):

```
iptables -F
```

This command erases all current rules. However, to delete a specific rule, you must use the **-D** option. First, you need to see all the available rules by entering the following command:

```
iptables -L --line-numbers
```

List rules of other chain:

```
iptables -t nat -nL --line
```

You will get a list of rules with numbers:

```
Chain INPUT (policy ACCEPT)

num  target      prot opt source                destination
1     ACCEPT      all  --  192.168.0.4            anywhere
2     ACCEPT      tcp  --  anywhere              anywhere tcp dpt:https
3     ACCEPT      tcp  --  anywhere              anywhere tcp dpt:http
4     ACCEPT      tcp  --  anywhere              anywhere tcp dpt:ssh
```

To delete a rule, insert the corresponding chain and the number from the list. Let's say for this iptables tutorial, we want to get rid of rule number three of the INPUT chain. The command should be:

```
iptables -D INPUT 3
```

Delete rule from another chain:

```
iptables -t nat -D POSTROUTING 1
```

## Persisting Changes

---

Install the next package to save the active rules and persist the rules after reboot

```
apt install iptables-persistent
```

To show the active rules:

```
/sbin/iptables-save
```

Save the rules:

```
/sbin/iptables-save > /etc/iptables/rules.v4
```



# Create Self-Signed certificate

---

First we have to install the package `openssl`

```
apt install openssl
```

Then we are going to create the certificate

```
openssl req -x509 -sha256 -newkey rsa:2048 -keyout certificate.key -out  
certificate.crt -days 1024 -nodes
```

## Parameters guide

---

- **-newkey rsa:2048** indica que queremos que la clave se genere usando el algoritmo RSA con 2048 bits.
- **-nodes** indica que no queremos que la clave privada sea encriptada con una contraseña.
- **-keyout** es la ruta + nombre del archivo donde queremos que se escriba la clave.
- **-out** es la ruta + nombre del archivo donde queremos que se escriba el documento CSR.
- **-sha256** es el algoritmo hash que queremos usar para generar la huella digital (si la CA a la que queramos enviar el CSR no soporta sha256 sencillamente eliminar esta línea).
- **-x509** indica que queremos crear un certificado autofirmado.
- **-days** indica el número de días durante los cuales es útil (válido) el certificado. Cuando pasen estos días, el navegador (o la aplicación que usemos para realizar la conexión segura) nos indicará que el certificado ya ha expirado. Tengamos en cuenta que el algoritmo de comunicación por Diffi-Hellman se basa en el problema de calcular la inversa del logaritmo en módulo N que aún no ha sido resuelto matemáticamente hablando, solo puede ser deducido mediante fuerza bruta. Dependiendo de la capacidad de computación actual y el número de bits usado en la clave privada, un usuario malicioso puede tardar más o menos tiempo en deducir la clave privada a partir de la clave pública. Para ello ponemos un límite en el cual suponemos que un usuario malicioso podría haber «roto» la clave y en ese momento el certificado SSL ya no sería seguro.

# SSL Certificates Cheat-Sheet

X.509 is an ITU standard defining the format of public key certificates. X.509 are used in TLS/SSL, which is the basis for HTTPS. An X.509 certificate binds an identity to a public key using a digital signature. A certificate contains an identity (hostname, organization, etc.) and a public key (RSA, DSA, ECDSA, ed25519, etc.), and is either signed by a Certificate Authority or is Self-Signed.

## Self-Signed Certificates

### Generate CA

1. Generate RSA

```
openssl genrsa -aes256 -out ca-key.pem 4096
```

2. Generate a public CA Cert

```
openssl req -new -x509 -sha256 -days 365 -key ca-key.pem -out ca.pem
```

### Generate Certificate

1. Create a RSA key

```
openssl genrsa -out cert-key.pem 4096
```

2. Create a Certificate Signing Request (CSR)

```
openssl req -new -sha256 -subj "/CN=yourcn" -key cert-key.pem -out cert.csr
```

3. Create a `extfile` with all the alternative names

```
echo "subjectAltName=DNS:your-dns.record,IP:257.10.10.1" >> extfile.cnf
```

```
# optional
```

```
echo extendedKeyUsage = serverAuth >> extfile.cnf
```

#### 4. Create the certificate

```
openssl x509 -req -sha256 -days 365 -in cert.csr -CA ca.pem -CAkey ca-key.pem -out cert.pem -extfile extfile.cnf -CAcreateserial
```

#### 5. Combine both certificates into one

```
cat cert.pem > fullchain.pem  
cat ca.pem >> fullchain.pem
```

## Certificate Formats

X.509 Certificates exist in Base64 Formats **PEM (.pem, .crt, .ca-bundle)**, **PKCS#7 (.p7b, p7s)** and Binary Formats **DER (.der, .cer)**, **PKCS#12 (.pfx, p12)**.

## Convert Certs

COMMAND	CONVERSION
<code>openssl x509 -outform der -in cert.pem -out cert.der</code>	PEM to DER
<code>openssl x509 -inform der -in cert.der -out cert.pem</code>	DER to PEM
<code>openssl pkcs12 -in cert.pfx -out cert.pem -nodes</code>	PFX to PEM

## Verify Certificates

```
openssl verify -CAfile ca.pem -verbose cert.pem
```

## Install the CA Cert as a trusted root CA

### On Debian & Derivatives

- Move the CA certificate (`ca.pem`) into `/usr/local/share/ca-certificates/ca.crt`.
- Update the Cert Store with:

```
sudo update-ca-certificates
```

Refer the documentation [here](#) and [here](#).

### On Fedora

- Move the CA certificate (`ca.pem`) to `/etc/pki/ca-trust/source/anchors/ca.pem` or `/usr/share/pki/ca-trust-source/anchors/ca.pem`
- Now run (with sudo if necessary):

```
update-ca-trust
```

Refer the documentation [here](#).

## On Arch

System-wide – Arch(p11-kit)

(From arch wiki)

- Run (As root)

```
trust anchor --store myCA.crt
```

- The certificate will be written to `/etc/ca-certificates/trust-source/myCA.p11-kit` and the "legacy" directories automatically updated.
- If you get "no configured writable location" or a similar error, import the CA manually:
- Copy the certificate to the `/etc/ca-certificates/trust-source/anchors` directory.
- and then

```
update-ca-trust
```

wiki page [here](#)

## On Windows

Assuming the path to your generated CA certificate as `C:\ca.pem`, run:

```
Import-Certificate -FilePath "C:\ca.pem" -CertStoreLocation  
Cert:\LocalMachine\Root
```

- Set `-CertStoreLocation` to `Cert:\CurrentUser\Root` in case you want to trust certificates only for the logged in user.

OR

In Command Prompt, run:

```
certutil.exe -addstore root C:\ca.pem
```

- `certutil.exe` is a built-in tool (classic `System32` one) and adds a system-wide trust anchor.

## On Android

The exact steps vary device-to-device, but here is a generalised guide:

1. Open Phone Settings
2. Locate `Encryption and Credentials` section. It is generally found under `Settings > Security > Encryption and Credentials`
3. Choose `Install a certificate`
4. Choose `CA Certificate`
5. Locate the certificate file `ca.pem` on your SD Card/Internal Storage using the file manager.
6. Select to load it.
7. Done!