

Crear un controlador de dominio de Active Directory con Samba en Ubuntu 18.04 LTS

Seguramente ya sabrás que *Samba* es un desarrollo de software libre, que implementa el protocolo *SMB/CIFS* con el objetivo de integrar sistemas *GNU/Linux*, *Mac OS X* y similares en redes *Microsoft*. En general, usando *Samba*, los equipos con éstos sistemas operativos pueden actuar como clientes o servidores en ese tipo de redes.

Además, desde la versión 4, *Samba* también permite que el equipo actúe como *Controlador de Dominio* de *Active Directory*, pudiendo compartir archivos e impresoras, autenticar usuarios y grupos, establecer permisos, políticas de grupo (GPOs), etc.

Precisamente, este último aspecto será el que resolvamos en el artículo de hoy. Utilizaremos una instalación básica de *Ubuntu 18.04 LTS* para implementar un *Controlador de dominio* donde los equipos clientes con *Microsoft Windows* puedan iniciar sesión en el dominio *Linux* y utilizar sus recursos sin notar la diferencia con un servidor *Windows Server*.

En particular, el servidor que implementaremos atenderá a los siguientes datos:

- Nombre del controlador de dominio de *Active Directory*: SMB-DC
- Nombre DNS del dominio de *Active Directory*: *somebooks.lan*
- Nombre del Reino Kerberos: *somebooks.lan*
- Nombre NetBIOS del dominio: *somebooks*
- Dirección IP fija del servidor: 192.168.1.15
- Rol del servidor: Domain Controller (DC)
- Reenviador DNS: 192.168.1.1

Los pasos que vamos a seguir para completar la tarea son los siguientes:

- Preparar el equipo
- Instalar los paquetes necesarios
- Establecer parámetros para el sistema de archivos
- Configurar *NTP*
- Instalar y configurar *Samba*
- Comprobar el funcionamiento del servidor *DNS*
- Configurar y comprobar el funcionamiento de *Kerberos*

Una vez que hemos fijado los objetivos, comencemos...

Preparar el equipo

Como hemos dicho más arriba, para este artículo partiremos de una instalación básica de *Ubuntu 18.04 LTS*

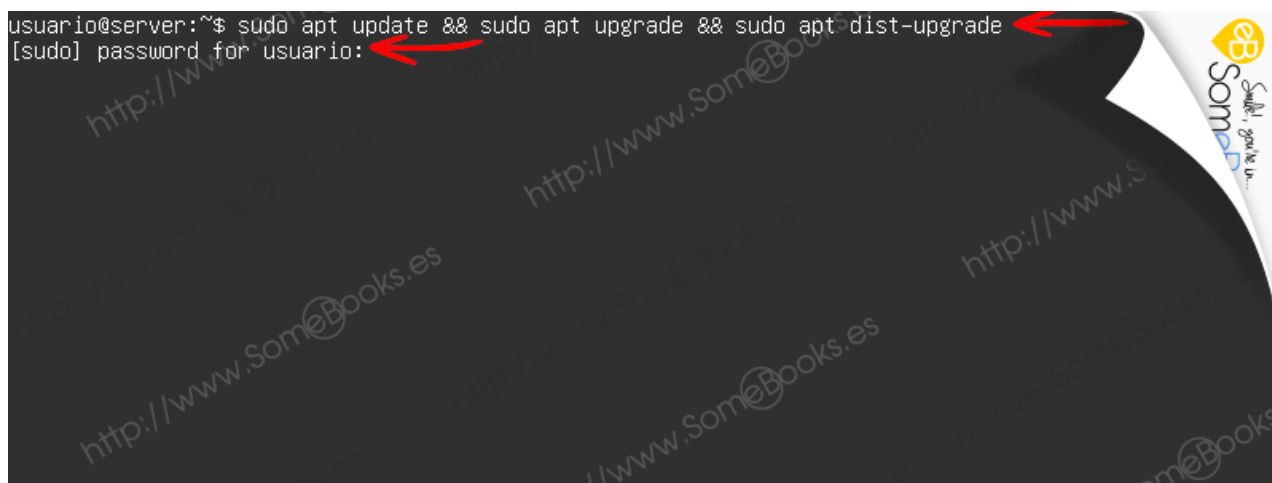
A continuación, deberemos realizar diferentes ajustes:

Actualizar el sistema

Para comenzar, nos aseguraremos de que el equipo se encuentra perfectamente actualizado. Para lograrlo, basta con escribir en la consola una orden como la siguiente:

```
sudo apt update && sudo apt upgrade && sudo apt dist-upgrade
```

Escribimos el comando y pulsamos la tecla `Intro`.



Establecer un nombre adecuado para el servidor

Una vez concluida la actualización, el segundo paso será adecuar el nombre del equipo a nuestras necesidades (recuerda que debe llamarse *smb-dc*).

El archivo **/etc/hosts**, contiene una relación de direcciones IP con sus correspondientes nombres lógicos. De esta forma, cuando hagamos referencia a un ordenador que esté identificado en esta lista, el acceso es inmediato, sin que necesitemos la mediación de un servidor *DNS* para resolverlo.

Este archivo contenía una referencia al nombre antiguo del propio servidor, que cambiaremos al seguir las indicaciones del artículo anterior. En nuestro caso, aprovecharemos para incluir tanto el nombre *DNS* como el nombre *NetBIOS* (*smb-dc.somebooks.lan* y *smb-dc*).

También incluiremos una referencia a la dirección IP estática que asignaremos al equipo en el siguiente apartado.

Por lo tanto, el aspecto final del archivo **/etc/hosts** debe ser como en la imagen, aunque con los valores que se correspondan con tus necesidades.

```
GNU nano 2.9.3 /etc/hosts
127.0.0.1    localhost.localdomain localhost
127.0.1.1    smb-dc.somebooks.ian smb-dc
192.168.1.15 smb-dc.somebooks.ian smb-dc
::1          localhost6.localdomain6 localhost6

# The following lines are desirable for IPv6 capable hosts
::1          localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

Al completar la tarea, podemos verificar que todo es correcto consultando el contenido del archivo **/etc/hosts**, por ejemplo, usando el comando **cat**, y ejecutando el comando **hostnamectl** para obtener el nombre del equipo.

La imagen muestra la salida de ambos comandos.

```
usuario@server:~$ cat /etc/hosts
127.0.0.1    localhost.localdomain localhost
127.0.1.1    smb-dc.somebooks.ian smb-dc
192.168.1.15 smb-dc.somebooks.ian smb-dc
::1          localhost6.localdomain6 localhost6

# The following lines are desirable for IPv6 capable hosts
::1          localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
usuario@server:~$ hostnamectl
  Static hostname: smb-dc
        Icon name: computer-vm
        Chassis: vm
        Machine ID: b5ebeb972727430eb3e2be1dff0761a4
        Boot ID: 3e2f72af12c44b31a7b828d5a6b6367b
        Virtualization: oracle
        Operating System: Ubuntu 18.04.3 LTS
        Kernel: Linux 4.15.0-64-generic
        Architecture: x86-64
usuario@server:~$
```

Configurar las características de red

Lo siguiente será configurar las características de red según las necesidades del servidor. Recuerda que un servidor debe tener *IP fija*.

Para nuestro ejemplo, el archivo *Netplan* acabará teniendo este aspecto:

```
GNU nano 2.9.3 /etc/netplan/50-cloud-init.yaml
# This file is generated from information provided by
# the datasource.  Changes to it will not persist across an instance.
# To disable cloud-init's network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    enp0s3:
      dhcp4: false
      addresses: [192.168.1.15/24]
      gateway4: 192.168.1.1
      nameservers:
        addresses: [8.8.8.8]
  version: 2

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text   ^J Justify    ^C Cur Pos   M-U Undo
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell   ^_ Go To Line  M-E Redo
```

Una vez completada la configuración, nos aseguramos que es correcta con el siguiente comando:

```
ip addr
```

Comprobamos la salida del comando

```
usuario@server:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:92:81:bd brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.15/24 brd 192.168.1.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe92:81bd/64 scope link
        valid_lft forever preferred_lft forever
usuario@server:~$ _
```

Antes de continuar, para asegurarnos de que la configuración inicial está correctamente aplicada, reiniciamos el equipo:

```
sudo reboot
```

Instalar los paquetes necesarios

Además de los paquetes propios de *Samba* necesitaremos disponer de algunos otros, que deberán estar preinstalados antes de comenzar con el proceso de configuración, aunque todos ellos están en los repositorios. Son los siguientes:

- samba: servidor de archivos, impresión y autenticación para SMB/CIFS.
- smbclient: clientes de línea de comandos para SMB/CIFS.
- krb5-config: Archivos de configuración para Kerberos Version 5.
- winbind: Servicio para resolver información sobre usuarios y grupos de servidores Windows NT.

Por lo tanto, usaremos un comando como este:

```
sudo apt install samba krb5-config winbind smbclient
```

Solo tienes que escribir el comando anterior y, cuando estés listo, pulsar la tecla `Intro`.

```
usuario@smb-dc:~$ sudo apt install samba krb5-config winbind smbclient
[sudo] password for usuario:
```

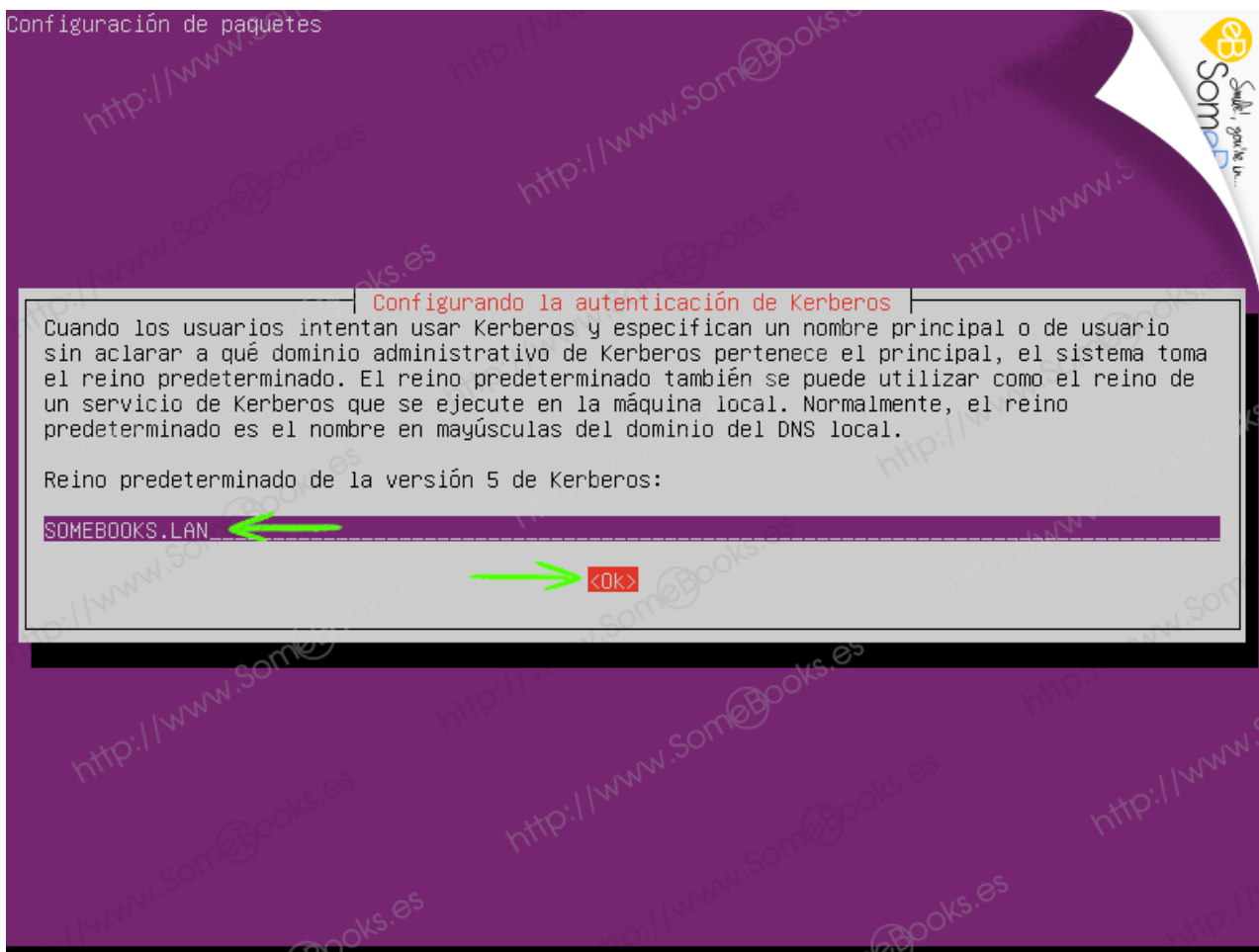
Al hacerlo, **apt** nos informará de los paquetes complementarios que deberán instalarse para completar la tarea.

Pulsamos la tecla **s** y después **Intro** para permitir la instalación.

```
usuario@smb-dc:~$ sudo apt install samba krb5-config winbind smbclient
[sudo] password for usuario:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
 attr ibverbs-providers libarchive13 libavahi-client3 libavahi-common-data libavahi-common3
 libcephfs2 libcups2 libgpgme11 libibverbs1 libjansson4 libldb1 libnl-route-3-200 libnspr4
 libnss3 libpython-stdlib libpython2.7 libpython2.7-minimal libpython2.7-stdlib librados2
 libsmbclient libtalloc2 libtdb1 libtevent0 libwbclient0 python python-crypto python-dnspython
 python-ldb python-minimal python-samba python-talloc python-tdb python2.7 python2.7-minimal
 samba-common samba-common-bin samba-dsdb-modules samba-libs samba-vfs-modules tdb-tools
Paquetes sugeridos:
 lrzip cups-common python-doc python-tk python-crypto-doc python-gpgme python2.7-doc binutils
 binfmt-support bind9 bind9utils ctdb ldb-tools ntp | chrony smbldap-tools heimdal-clients
 cifs-utils libnss-winbind libpam-winbind
Se instalarán los siguientes paquetes NUEVOS:
 attr ibverbs-providers krb5-config libarchive13 libavahi-client3 libavahi-common-data
 libavahi-common3 libcephfs2 libcups2 libgpgme11 libibverbs1 libjansson4 libldb1
 libnl-route-3-200 libnspr4 libnss3 libpython-stdlib libpython2.7 libpython2.7-minimal
 libpython2.7-stdlib librados2 libsmbclient libtalloc2 libtdb1 libtevent0 libwbclient0 python
 python-crypto python-dnspython python-ldb python-minimal python-samba python-talloc python-tdb
 python2.7 python2.7-minimal samba samba-common samba-common-bin samba-dsdb-modules samba-libs
 samba-vfs-modules smbclient tdb-tools winbind
0 actualizados, 45 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 21,1 MB de archivos.
Se utilizarán 102 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

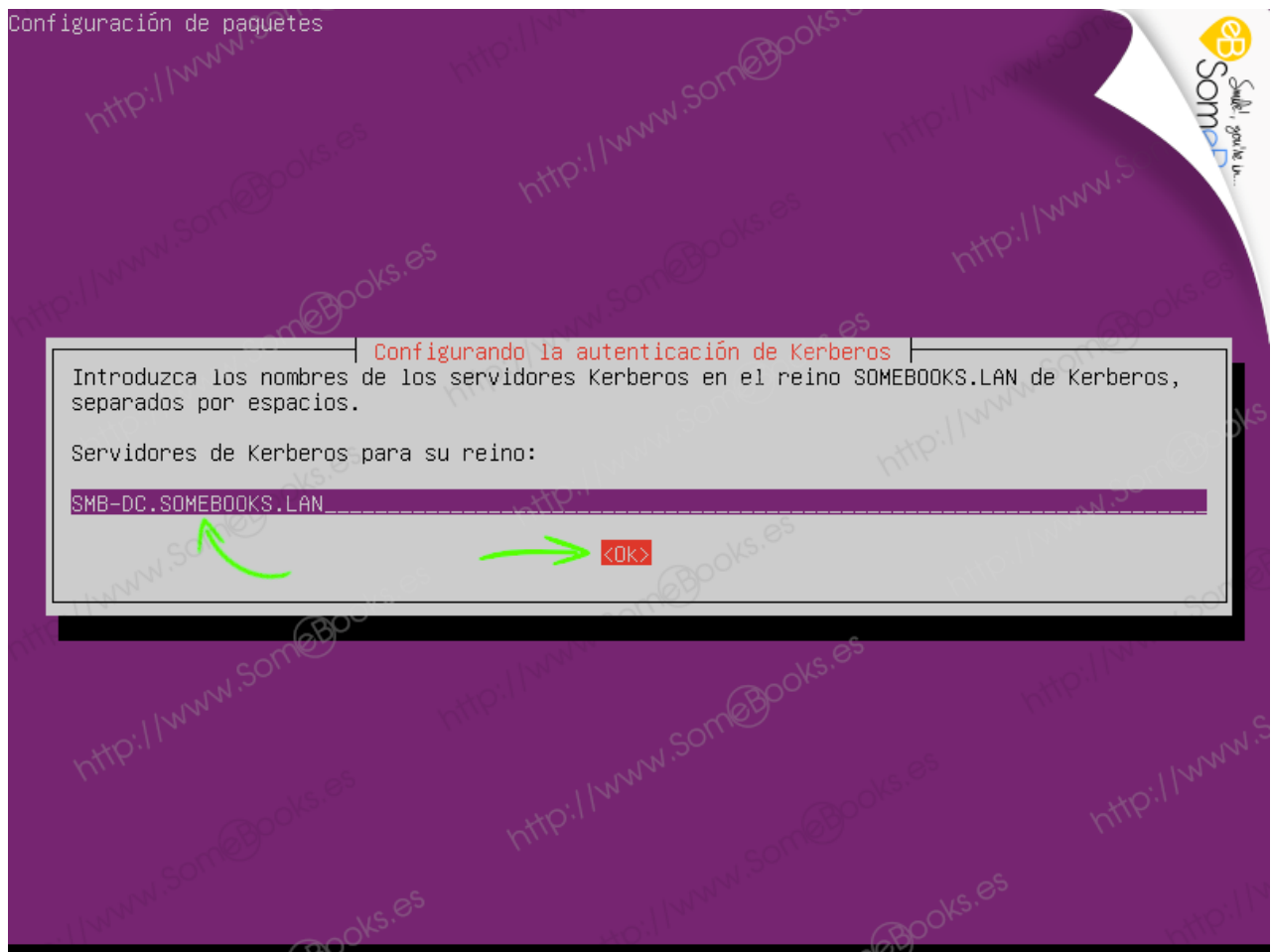
Cuando llegue el momento de instalar *Kerberos*, nos preguntará por el reino (*realm* en inglés). En realidad, se refiere al nombre del dominio y, en el caso de nuestro ejemplo, escribiremos **SOMEBOOKS.LAN**.

A continuación, pulsamos la tecla de tabulación para seleccionar *Ok* y, a continuación, la tecla *Intro*.



Después, el instalador nos solicita el nombre de los *servidores de Kerberos para nuestro reino*. En este caso, se refiere a los controladores de dominio que tengamos definidos. En nuestro ejemplo sólo tenemos uno y se llama SMB-DC.SOMEBOOKS.LAN, por lo que procedemos a escribir su nombre.

Como antes, nos desplazamos hasta *OK* y pulsamos la tecla *Intro*.



Por último, nos solicita el *servidor administrativo para nuestro reino de Kerberos*, como sólo tenemos uno, volvemos a escribir su nombre (SMB-DC.SOMEBOOKS.LAN).

Y de nuevo seleccionamos *Ok*.

Configuración de paquetes

Configurando la autenticación de Kerberos

Introduzca el nombre del servidor administrativo (cambio de contraseña) para el reino SOMEBOOKS.LAN de Kerberos.

Servidor administrativo para su reino de Kerberos:

SMB-DC.SOMEBOOKS.LAN

<OK>

Después de esto, la instalación continuará un poco más, pero sin necesitar que aportemos más información.

Poco después, la instalación habrá terminado.

```

Configurando libavahi-client3:amd64 (0.7-3.1ubuntu1.2) ...
Configurando libcups2:amd64 (2.2.7-1ubuntu2.7) ...
Configurando python-ldb:amd64 (2:1.2.3-1ubuntu0.1) ...
Configurando samba-libs:amd64 (2:4.7.6+dfsg~ubuntu-0ubuntu2.11) ...
Configurando samba-vfs-modules (2:4.7.6+dfsg~ubuntu-0ubuntu2.11) ...
Configurando python-samba (2:4.7.6+dfsg~ubuntu-0ubuntu2.11) ...
Configurando libsmclient:amd64 (2:4.7.6+dfsg~ubuntu-0ubuntu2.11) ...
Configurando smbclient (2:4.7.6+dfsg~ubuntu-0ubuntu2.11) ...
Configurando samba-common-bin (2:4.7.6+dfsg~ubuntu-0ubuntu2.11) ...
Configurando samba-dsdb-modules (2:4.7.6+dfsg~ubuntu-0ubuntu2.11) ...
Configurando winbind (2:4.7.6+dfsg~ubuntu-0ubuntu2.11) ...
mkdir: created directory '/var/lib/samba/winbindd_privileged'
changed group of '/var/lib/samba/winbindd_privileged' from root to winbindd_priv
mode of '/var/lib/samba/winbindd_privileged' changed from 0755 (rwxr-xr-x) to 0750 (rwxr-x---)
Created symlink /etc/systemd/system/multi-user.target.wants/winbind.service → /lib/systemd/system/winbind.service.
Configurando samba (2:4.7.6+dfsg~ubuntu-0ubuntu2.11) ...
Adding group `sambashare' (GID 115) ...
Done.
Samba is not being run as an AD Domain Controller, masking samba-ad-dc.service.
Please ignore the following error about deb-systemd-helper not finding samba-ad-dc.service.
Created symlink /etc/systemd/system/multi-user.target.wants/nmbd.service → /lib/systemd/system/nmbd.service.
Failed to preset unit: Unit file /etc/systemd/system/samba-ad-dc.service is masked.
/usr/bin/deb-systemd-helper: error: systemctl preset failed on samba-ad-dc.service: No such file or directory
Created symlink /etc/systemd/system/multi-user.target.wants/smbd.service → /lib/systemd/system/smbd.service.
Procesando disparadores para mime-support (3.60ubuntu1) ...
Procesando disparadores para ureadahead (0.100.0-21) ...
Procesando disparadores para libc-bin (2.27-3ubuntu1) ...
Procesando disparadores para systemd (237-3ubuntu10.29) ...
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...
Procesando disparadores para ufw (0.36-0ubuntu0.18.04.1) ...
usuario@smb-dc:~$ _

```

Configurar Samba

Una vez completada la instalación de los paquetes, ha llegado el momento de comenzar a configurar nuestro servidor. Pero antes, debemos cambiar de nombre (o de sitio) el archivo *smb.conf* que contiene la configuración predeterminada.

De esta forma, evitamos que *Samba* intente usarlo durante el proceso de configuración y estaremos seguros de que todos los datos del nuevo archivo de configuración se producen desde cero. Además, también podremos recuperarlo en caso de que algo salga mal.

Para cambiar el nombre al archivo, sólo tendremos que escribir la siguiente orden:

```
sudo mv /etc/samba/smb.conf /etc/samba/smb.conf.old
```

Renombramos el archivo **smb.conf**.



```
usuario@smb-dc:~$ sudo mv /etc/samba/smb.conf /etc/samba/smb.conf.old
usuario@smb-dc:~$ _
```

A red arrow points to the command `sudo mv /etc/samba/smb.conf /etc/samba/smb.conf.old`. The terminal background has a watermark for 'SomeBooks.es' and a logo in the top right corner.

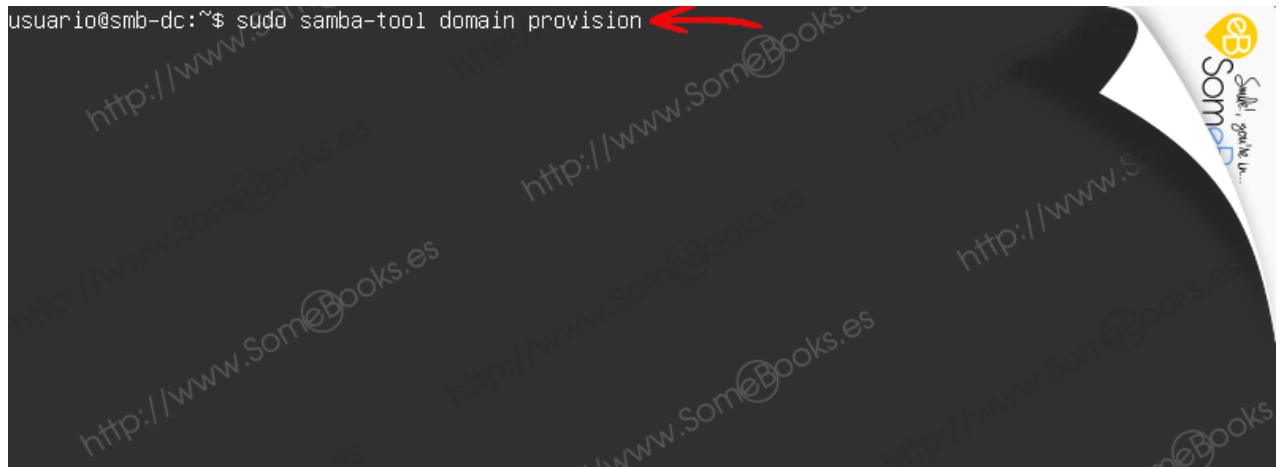
Después de esto, ya estamos listos para promover nuestro equipo como controlador de un dominio *Samba 4* que actúe como un reemplazo completo de un servidor de dominio de *Active Directory*.

Para lograrlo, usaremos el comando **samba-tool domain provision**, y lo haremos de forma interactiva, para que sea el propio comando el que nos solicite los valores que necesite y, cuando sea posible, nos sugiera sus valores predeterminados. Así, si éstos coinciden con los que nosotros esperamos, será muy probable que los pasos anteriores hayan sido los correctos.

Así pues, comenzamos escribiendo la siguiente orden:

```
sudo samba-tool domain provision
```

Iniciamos la provisión del controlador de dominio.



```
usuario@smb-dc:~$ sudo samba-tool domain provision
```

A red arrow points to the command `sudo samba-tool domain provision`. The terminal background has a watermark for 'SomeBooks.es' and a logo in the top right corner.

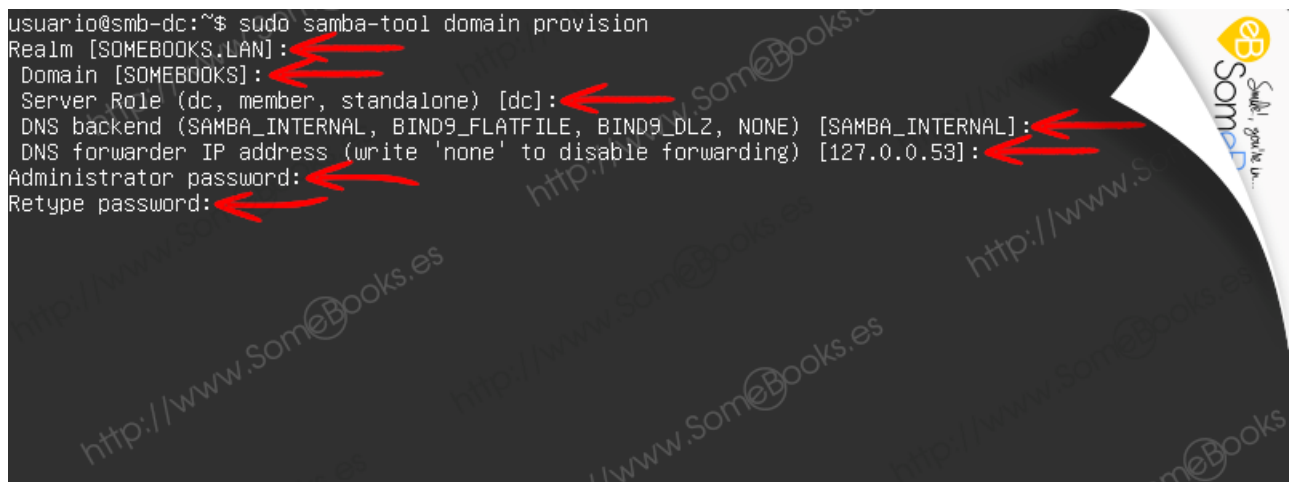
Como puedes ver en la imagen siguiente, los valores predeterminados del comando son los esperados para nuestro ejemplo:

- Realm: SOMEBOOKS.LAN (el nombre del dominio).
- Domain: SOMEBOOKS (el nombre *NetBIOS* del dominio).
- Server Role: dc (domain controller, es decir, controlador de dominio).
- DNS backend: SAMBA_INTERNAL (el servidor *DNS* que utilizaremos, que será el propio *Samba*).
- DNS forwarder IP address: 127.0.0.53 (el reenviador *DNS*, es decir, el servidor al que recurrirá cuando no sepa resolver alguna referencia).

Observa que solo hemos tenido que escribir el valor para el primer campo.

Por último, nos pedirá, como siempre por duplicado, la contraseña para el administrador.

Cuando el valor deseado coincida con el que aparece entre corchetes, bastará con pulsar la tecla `Intro`.



```

usuario@smb-dc:~$ sudo samba-tool domain provision
Realm [SOMEBOOKS.LAN]:
Domain [SOMEBOOKS]:
Server Role (dc, member, standalone) [dc]:
DNS backend (SAMBA_INTERNAL, BIND9_FLATFILE, BIND9_DLZ, NONE) [SAMBA_INTERNAL]:
DNS forwarder IP address (write 'none' to disable forwarding) [127.0.0.53]:
Administrator password:
Retype password:
  
```

Después del proceso de configuración, el comando nos ofrece un resumen de las acciones que ha llevado a cabo para que podamos comprobarlas con detenimiento.

Nos aseguramos de que los valores son los esperados.

```
Setting up the registry
Setting up the privileges database
Setting up idmap db
Setting up SAM db
Setting up sam.ldb partitions and settings
Setting up sam.ldb rootDSE
Pre-loading the Samba 4 and AD schema
Adding DomainDN: DC=somebooks,DC=lan
Adding configuration container
Setting up sam.ldb schema
Setting up sam.ldb configuration data
Setting up display specifiers
Modifying display specifiers
Adding users container
Modifying users container
Adding computers container
Modifying computers container
Setting up sam.ldb data
Setting up well known security principals
Setting up sam.ldb users and groups
Setting up self join
Adding DNS accounts
Creating CN=MicrosoftDNS,CN=System,DC=somebooks,DC=lan
Creating DomainDnsZones and ForestDnsZones partitions
Populating DomainDnsZones and ForestDnsZones partitions
Setting up sam.ldb rootDSE marking as synchronized
Fixing provision GUIDs
A Kerberos configuration suitable for Samba AD has been generated at /var/lib/samba/private/krb5.conf
f
Once the above files are installed, your Samba AD server will be ready to use
Server Role: active directory domain controller
Hostname: smb-dc
NetBIOS Domain: SOMEBOOKS
DNS Domain: somebooks.lan
DOMAIN SID: S-1-5-21-239877117-2419810215-3004548455

usuario@smb-dc:~$
```

Con esto, además de configurarse Samba de forma adecuada para nuestros propósitos, se ha generado un archivo de configuración para Kerberos en la ruta `/var/lib/samba/private/krb5.conf`.

Solo tenemos que copiarlo a la ubicación adecuada:

```
sudo cp /var/lib/samba/private/krb5.conf /etc/
```

Copiamos `krb5.conf` al directorio `/etc`.

```
usuario@smb-dc:~$ sudo cp /var/lib/samba/private/krb5.conf /etc/
usuario@smb-dc:~$
```

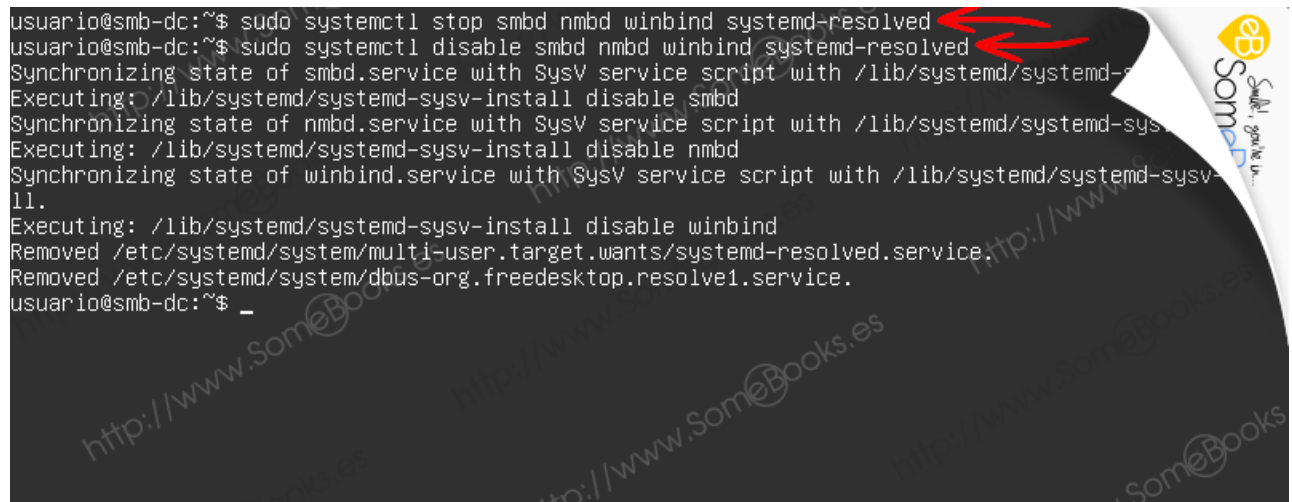
Lo siguiente será ajustar la resolución de nombres, y comenzaremos deteniendo los servicios implicados:

```
sudo systemctl stop smbd nmbd winbind systemd-resolved
```

También los deshabilitaremos, para que no vuelvan a iniciarse si reiniciamos el equipo:

```
sudo systemctl disable smbd nmbd winbind systemd-resolved
```

Escribimos ambos comandos y comprobamos que se ejecutan correctamente.

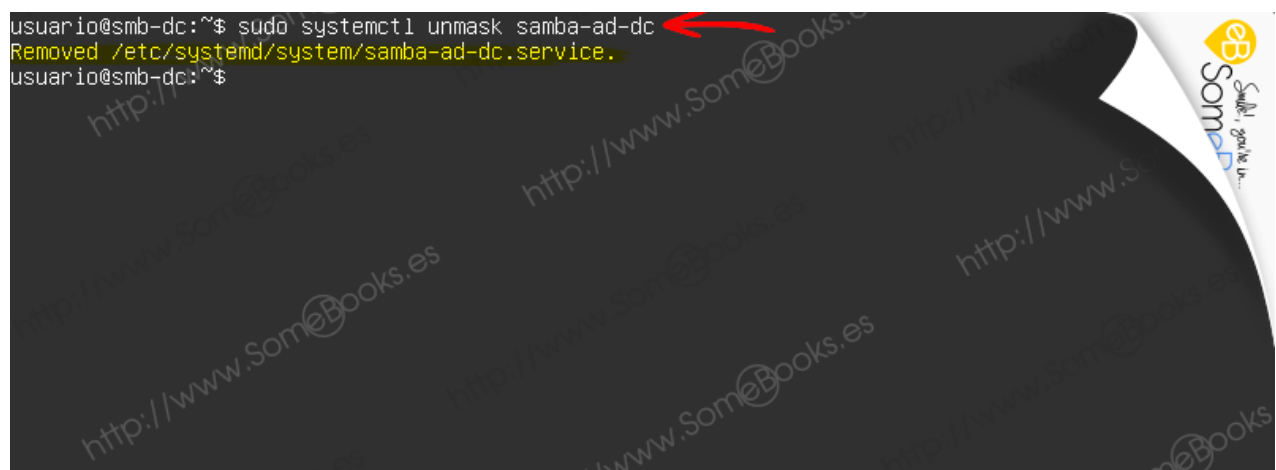
A terminal window screenshot showing the execution of two systemctl commands. The first command is 'sudo systemctl stop smbd nmbd winbind systemd-resolved' and the second is 'sudo systemctl disable smbd nmbd winbind systemd-resolved'. The output shows the synchronization of service states and the execution of disable scripts for each service. Red arrows point to the command lines. A watermark 'http://www.SomeBooks.es' is visible across the terminal output. A logo for 'SomeBooks' is in the top right corner.

```
usuario@smb-dc:~$ sudo systemctl stop smbd nmbd winbind systemd-resolved
usuario@smb-dc:~$ sudo systemctl disable smbd nmbd winbind systemd-resolved
Synchronizing state of smbd.service with SysV service script with /lib/systemd/systemd-sysv-install
Executing: /lib/systemd/systemd-sysv-install disable smbd
Synchronizing state of nmbd.service with SysV service script with /lib/systemd/systemd-sysv-install
Executing: /lib/systemd/systemd-sysv-install disable nmbd
Synchronizing state of winbind.service with SysV service script with /lib/systemd/systemd-sysv-install
Executing: /lib/systemd/systemd-sysv-install disable winbind
Removed /etc/systemd/system/multi-user.target.wants/systemd-resolved.service.
Removed /etc/systemd/system/dbus-org.freedesktop.resolve1.service.
usuario@smb-dc:~$ _
```

A continuación, nos aseguraremos de que el servicio *samba-ad-dc* se podrá iniciar sin dificultades, evitando cualquier enmascaramiento que pueda existir:

```
sudo systemctl unmask samba-ad-dc
```

Ejecutamos el comando

A terminal window screenshot showing the execution of the 'sudo systemctl unmask samba-ad-dc' command. The output shows the removal of the service file from the multi-user target. A red arrow points to the command line. A watermark 'http://www.SomeBooks.es' is visible across the terminal output. A logo for 'SomeBooks' is in the top right corner.

```
usuario@smb-dc:~$ sudo systemctl unmask samba-ad-dc
Removed /etc/systemd/system/samba-ad-dc.service.
usuario@smb-dc:~$
```

Después, eliminamos el archivo *resolv.conf* que, en realidad, será un enlace a *../run/systemd/resolve/stub-resolv.conf*. Para comprobarlo, comenzamos por consultar su entrada en el directorio:

```
sudo ls -l /etc/resolv.conf
```

La comprobación es correcta.



```
usuario@smb-dc:~$ sudo ls -l /etc/resolv.conf
lrwxrwxrwx 1 root root 39 feb 14 2019 /etc/resolv.conf -> ../run/systemd/resolve/stub-resolv.conf
usuario@smb-dc:~$ _
```

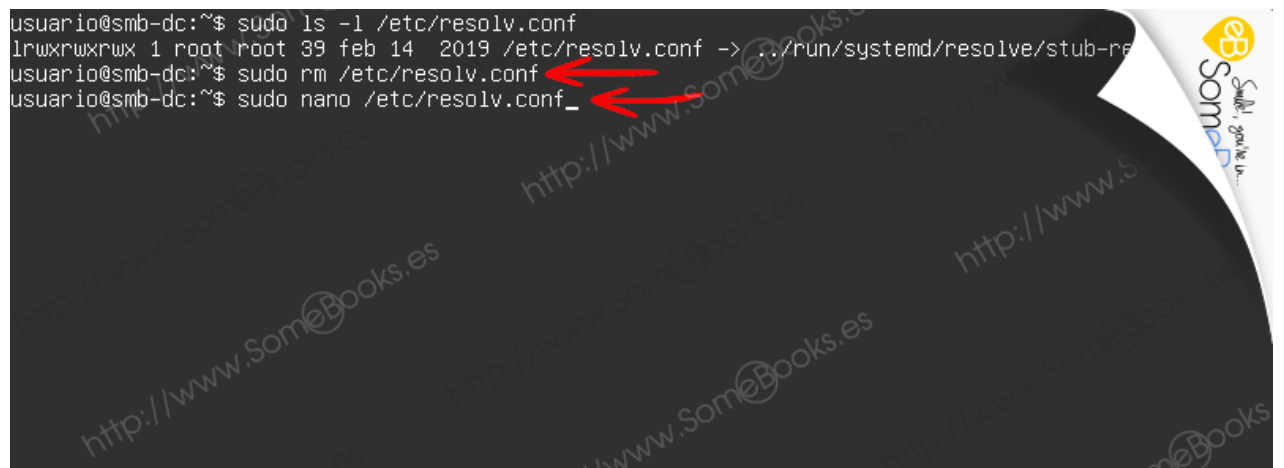
Por lo tanto, procedemos a eliminar el enlace:

```
sudo rm /etc/resolv.conf
```

Y utilizamos el editor *nano* para crear un nuevo archivo que lo sustituya:

```
sudo nano /etc/resolv.conf
```

Ejecutamos ambos comandos.



```
usuario@smb-dc:~$ sudo ls -l /etc/resolv.conf
lrwxrwxrwx 1 root root 39 feb 14 2019 /etc/resolv.conf -> ../run/systemd/resolve/stub-resolv.conf
usuario@smb-dc:~$ sudo rm /etc/resolv.conf
usuario@smb-dc:~$ sudo nano /etc/resolv.conf _
```

Una vez que nos encontremos en el entorno de trabajo de *nano*, escribiremos los valores adecuados para nuestro dominio:

```
domain somebooks.lan
nameserver 127.0.0.1
```

Cuando hayamos terminado, salimos del editor *nano* pulsando la combinación de teclas Ctrl + X.



```
GNU nano 2.9.3 /etc/resolv.conf
domain somebooks.lan
nameserver 127.0.0.1
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^_ Replace ^U Uncut Text ^T To Spell ^_ Go To Line M-E Redo

Al salir, *nano* nos pregunta si queremos guardar los cambios. Lógicamente, la respuesta debe ser *Sí* (Yes).

Pulsamos la tecla *y*.

```
GNU nano 2.9.3 /etc/resolv.conf
domain somebooks.1an
nameserver 127.0.0.1
```

```
Save modified buffer? (Answering "No" will DISCARD changes.)
Y Yes
N No      ^C Cancel
```

Como última precaución, *nano* nos ofrece la posibilidad de cambiar el nombre del archivo, por si no queremos reescribirlo.

Como preferimos reescribirlo, pulsamos la tecla `Intro`.

```
GNU nano 2.9.3 /etc/resolv.conf
domain somebooks.lan
nameserver 127.0.0.1
```



File Name to Write: /etc/resolv.conf

^G Get Help M-D DOS Format M-A Append M-B Backup File
^C Cancel M-M Mac Format M-P Prepend ^T To Files

Y ya solo nos queda iniciar el servicio *samba-ad-dc*:

```
sudo systemctl start samba-ad-dc
```

Y habilitarlo, para que también se inicie automáticamente al arrancar el sistema:

```
sudo systemctl enable samba-ad-dc
```

Ejecutamos ambos comandos.

```
usuario@smb-dc:~$ sudo systemctl start samba-ad-dc
usuario@smb-dc:~$ sudo systemctl enable samba-ad-dc
Synchronizing state of samba-ad-dc.service with SysV service script with /lib/systemd/s
ninstall.
Executing: /lib/systemd/systemd-sysv-install enable samba-ad-dc
usuario@smb-dc:~$
```

Comprobar la instalación

Si has llegado hasta aquí, lo más probable es que la instalación sea correcta. Sin embargo, antes de dar el trabajo por concluido, vamos a realizar una serie de comprobaciones...

Consultar el nivel del dominio y crear un nuevo usuario

Para consultar el nivel de nuestro dominio, es decir, el tipo de servidor que hemos implementado, solo tenemos que usar el comando **samba-tool** con la siguiente sintaxis:

```
sudo samba-tool domain level show
```

Al hacerlo, comprobamos que el nivel del dominio, y del bosque donde se encuentra equivale a una instalación *Windows Server 2008 R2*... Lo que no está nada mal.

Salida del comando

```
usuario@smb-dc:~$ sudo samba-tool domain level show
Domain and forest function level for domain 'DC=somebooks,DC=lan'

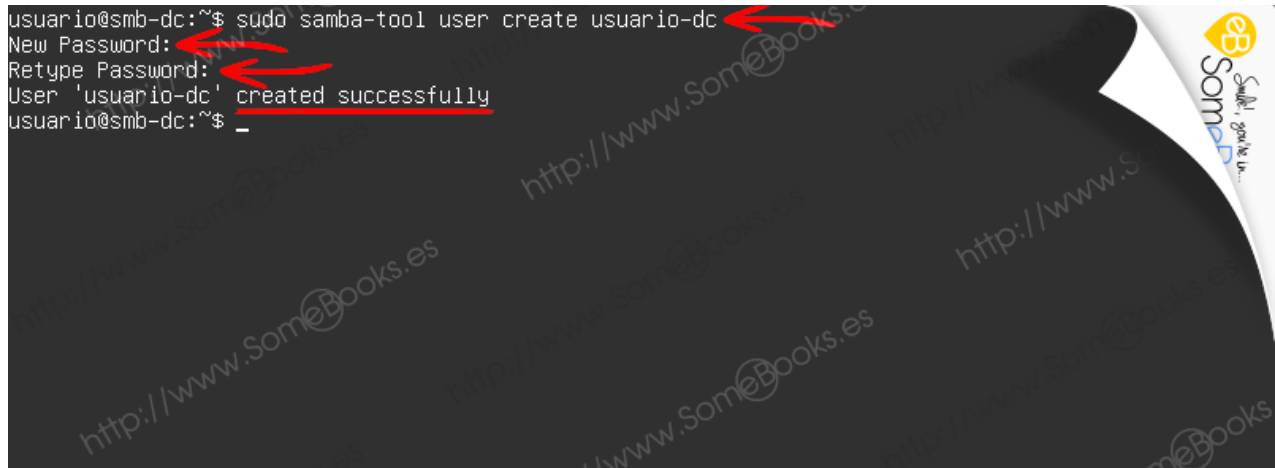
Forest function level: (Windows) 2008 R2
Domain function level: (Windows) 2008 R2
Lowest function level of a DC: (Windows) 2008 R2
usuario@smb-dc:~$ _
```

Como parece que la instalación es correcta, probaremos a crear una nueva cuenta de usuario en el dominio. Algo que conseguiremos con el comando:

```
sudo samba-tool user create usuario-dc
```

Como es lógico, el nombre de nuestro usuario de ejemplo será *usuario-dc*.

Para crear la cuenta, el comando nos solicita su contraseña por duplicado.

A terminal window screenshot showing the execution of the command 'sudo samba-tool user create usuario-dc'. The prompt is 'usuario@smb-dc:~\$'. The command is entered, and the system prompts for a 'New Password:' and 'Retype Password:'. Both prompts are followed by redacted input (asterisks). The final output is 'User 'usuario-dc' created successfully', which is underlined. The prompt returns to 'usuario@smb-dc:~\$'. There are red arrows pointing to the command and the success message. A watermark 'http://www.SomeBooks.es' is visible across the terminal output. On the right side, there is a logo for 'eb SomeBooks' with the tagline 'Smile, you're in...'.

```
usuario@smb-dc:~$ sudo samba-tool user create usuario-dc
New Password:
Retype Password:
User 'usuario-dc' created successfully
usuario@smb-dc:~$
```

Confirmar que el servidor DNS funciona de forma adecuada

Para que *Samba 4* funcione correctamente, es muy importante que el servidor *DNS* que tenga asociado también lo haga. En este caso, el servidor *DNS* interno del propio *Samba* y en particular, los registros *SRV*.

Los registros *SRV* son los encargados de almacenar, dentro de la base de datos *DNS*, la relación entre el nombre de un servicio y el nombre *DNS* del ordenador que ofrece dicho servicio.

Lo primero será comprobar el servicio *LDAP* sobre el protocolo *TCP*, para lo que escribiremos la siguiente orden:

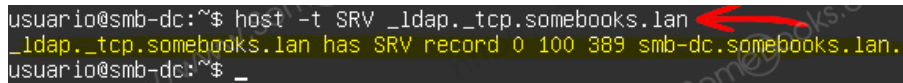
```
host -t SRV _ldap._tcp.somebooks.lan
```

Si todo es correcto, recibiremos una respuesta parecida a esta:

```
_ldap._tcp.somebooks.lan has SRV record 0 100 389 smb-dc.somebooks.lan
```

Lo que significará que todo funciona como debe.

Observamos que la salida del comando no produce ningún error.

A terminal window with a dark background. The prompt is 'usuario@smb-dc:~\$'. The command 'host -t SRV _ldap._tcp.somebooks.lan' is entered. The output is '_ldap._tcp.somebooks.lan has SRV record 0 100 389 smb-dc.somebooks.lan.'. A red arrow points to the command. The background has a watermark 'http://www.SomeBooks.es' and a logo 'eB SomeBooks' in the top right corner.

```
usuario@smb-dc:~$ host -t SRV _ldap._tcp.somebooks.lan
_ldap._tcp.somebooks.lan has SRV record 0 100 389 smb-dc.somebooks.lan.
usuario@smb-dc:~$ _
```

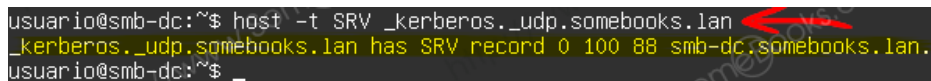
A continuación, comprobaremos el registro *SRV* para el protocolo *Kerberos* sobre *UDP*, para lo que usamos la siguiente orden:

```
host -t SRV _kerberos._udp.somebooks.lan
```

La respuesta debe ser parecida a la anterior, y podremos comprobar que el registro *SRV* es correcto:

```
_kerberos._udp.somebooks.lan has SRV record 0 100 88 smb-dc.somebooks.lan
```

De nuevo comprobamos que no se ha producido ningún error en la salida.

A terminal window with a dark background. The prompt is 'usuario@smb-dc:~\$'. The command 'host -t SRV _kerberos._udp.somebooks.lan' is entered. The output is '_kerberos._udp.somebooks.lan has SRV record 0 100 88 smb-dc.somebooks.lan.'. A red arrow points to the command. The background has a watermark 'http://www.SomeBooks.es' and a logo 'eB SomeBooks' in the top right corner.

```
usuario@smb-dc:~$ host -t SRV _kerberos._udp.somebooks.lan
_kerberos._udp.somebooks.lan has SRV record 0 100 88 smb-dc.somebooks.lan.
usuario@smb-dc:~$ _
```

Por último, comprobamos la resolución del nombre de nuestro servidor:

```
host -t A smb-dc.somebooks.lan
```

... Y si todo va bien, la respuesta debe ser parecida a la siguiente:

```
smb-dc.somebooks.lan has address 192.168.1.15
```

Nos aseguramos de que la resolución del nombre del servidor también es correcta.

```
usuario@smb-dc:~$ host -t A smb-dc.somebooks.lan  
smb-dc.somebooks.lan has address 192.168.1.15  
usuario@smb-dc:~$ _
```

Y para completar la comprobación, nos aseguramos de que se resuelven correctamente los nombres y las IPs del dominio, ejecutando el siguiente comando:

```
nslookup
```

Al ejecutarlo sin argumentos, aparece un signo '*mayor que*' a modo de *prompt*, donde podemos ir escribiendo argumentos.

Para comenzar, escribiremos el comando *server*, seguido de la *dirección IP* del equipo y comprobamos que se resuelve satisfactoriamente.

Después, establecemos el tipo de consulta para que utilice los registros SRV y añadimos el nombre de servicio a consultar. Al hacerlo, comprobamos que también se resuelve de forma satisfactoria.

Para terminar, escribimos la palabra *exit*.

```
usuario@smb-dc:~$ nslookup  
> server 192.168.1.15  
Default server: 192.168.1.15  
Address: 192.168.1.15#53  
> set type=SRV  
> _ldap._tcp.somebooks.lan  
Server:      192.168.1.15  
Address:     192.168.1.15#53  
  
_ldap._tcp.somebooks.lan service = 0 100 389 smb-dc.somebooks.lan.  
> exit  
usuario@smb-dc:~$
```

Comprobar el funcionamiento de Kerberos

Para comprobar el funcionamiento de *Kerberos* podemos, por ejemplo, usar el comando *smbclient* para comprobar los servicios que puede obtener un determinado usuario. Por ejemplo, el usuario *administrator*.

Para conseguirlo, usaremos la siguiente sintaxis:

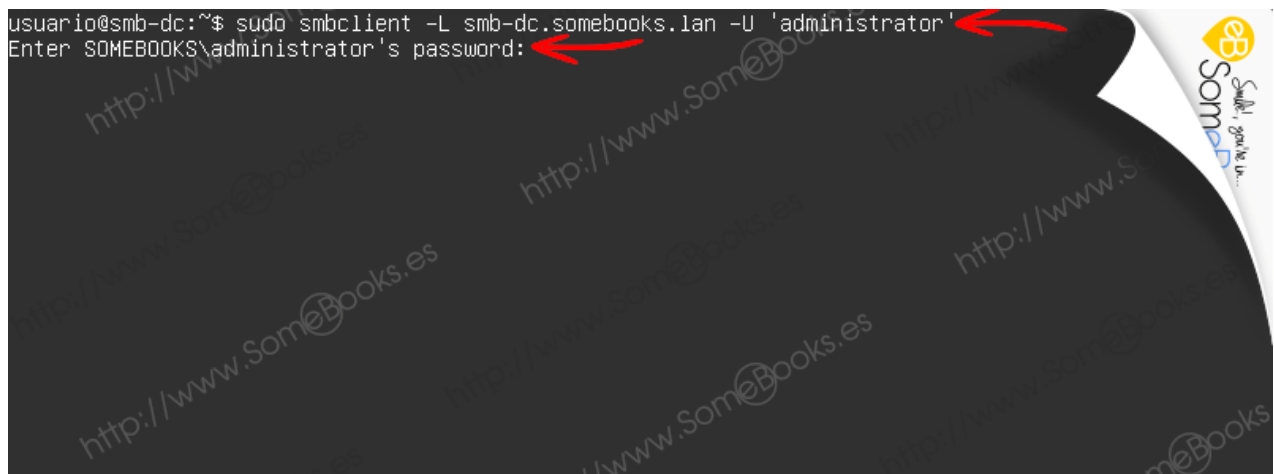
```
sudo smbclient -L smb-dc.somebooks.lan -U 'administrator'
```

Al hacerlo, deberemos escribir la contraseña del usuario *administrator*

Esta contraseña la escribimos durante la promoción del equipo como controlador de dominio, casi al principio del artículo, en respuesta al comando:

```
sudo samba-tool domain provision
```

Escribimos la contraseña y pulsamos la tecla `Intro`.



Si la autenticación es correcta, ya sabremos que *Kerberos* está haciendo su trabajo.

... Y el comando nos responde con la información solicitada.

```
usuario@smb-dc:~$ sudo smbclient -L smb-dc.somebooks.1an -U 'administrator'
Enter SOMEBOOKS\administrator's password:
```

Sharename	Type	Comment
netlogon	Disk	
sysvol	Disk	
IPC\$	IPC	IPC Service (Samba 4.7.6-Ubuntu)

Reconnecting with SMB1 for workgroup listing.

Server	Comment
Workgroup	Master
WORKGROUP	SMB-DC

```
usuario@smb-dc:~$ _
```

Si queremos, incluso podemos iniciar sesión en el servidor usando la cuenta de administración. Para lograrlo, usaremos una sintaxis como esta:

```
sudo smbclient //localhost/netlogon -U 'administrator'
```

El *prompt* cambia para indicar que es el subsistema de *Samba* quien espera órdenes, por lo que la *autenticación* habrá sido correcta.

Como se trata de una comprobación, y no pretendemos realizar ninguna tarea, nos limitamos a salir usando la orden *exit*.

```
usuario@smb-dc:~$ sudo smbclient //localhost/netlogon -U 'administrator'
Enter SOMEBOOKS\administrator's password:
Try "help" to get a list of possible commands.
smb: \> exit
usuario@smb-dc:~$
```

Para concluir podemos utilizar el comando **testparm**, que verifica la integridad del archivo de configuración de *Samba*:

Comprobamos que los datos son correctos

```
usuario@smb-dc:~$ testparm
Load smb config files from /etc/samba/smb.conf
rlimit_max: increasing rlimit_max (1024) to minimum Windows limit (16384)
Processing section "[netlogon]"
Processing section "[sysvol]"
Loaded services file OK.
Server role: ROLE_ACTIVE_DIRECTORY_DC
```

Press enter to see a dump of your service definitions

Incluso podemos obtener un volcado completo del archivo pulsando la tecla Intro.

Pulsamos la tecla Intro.

```
Server role: ROLE_ACTIVE_DIRECTORY_DC
Press enter to see a dump of your service definitions
# Global parameters
[global]
    dns forwarder = 127.0.0.53
    passdb backend = samba_dsdb
    realm = SOMEBOOKS.LAN
    server role = active directory domain controller
    workgroup = SOMEBOOKS
    rpc_server:tcpip = no
    rpc_daemon:spoolssd = embedded
    rpc_server:spoolss = embedded
    rpc_server:winreg = embedded
    rpc_server:ntsvcs = embedded
    rpc_server:eventlog = embedded
    rpc_server:svrsvc = embedded
    rpc_server:svccctl = embedded
    rpc_server:default = external
    winbindd:use external pipes = true
    idmap config * : backend = tdb
    map archive = No
    map readonly = no
    store dos attributes = Yes
    vfs objects = dfs_samba4 acl_xattr

[netlogon]
    path = /var/lib/samba/sysvol/somebooks.lan/scripts
    read only = No

[sysvol]
    path = /var/lib/samba/sysvol
    read only = No
usuario@smb-dc:~$
```

El camino ha sido un poco largo, pero si todo ha ido bien, acabas de obtener un reemplazo completo, para un controlador de dominio de *active directory*, usando únicamente software libre.