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*

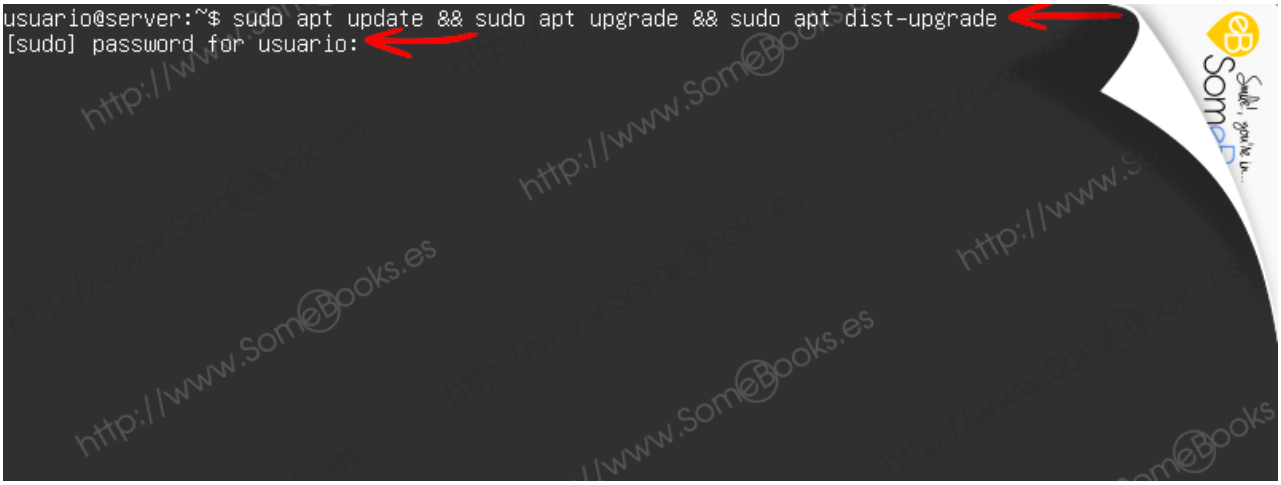
A continuación, deberemos realizar diferentes ajustes:

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

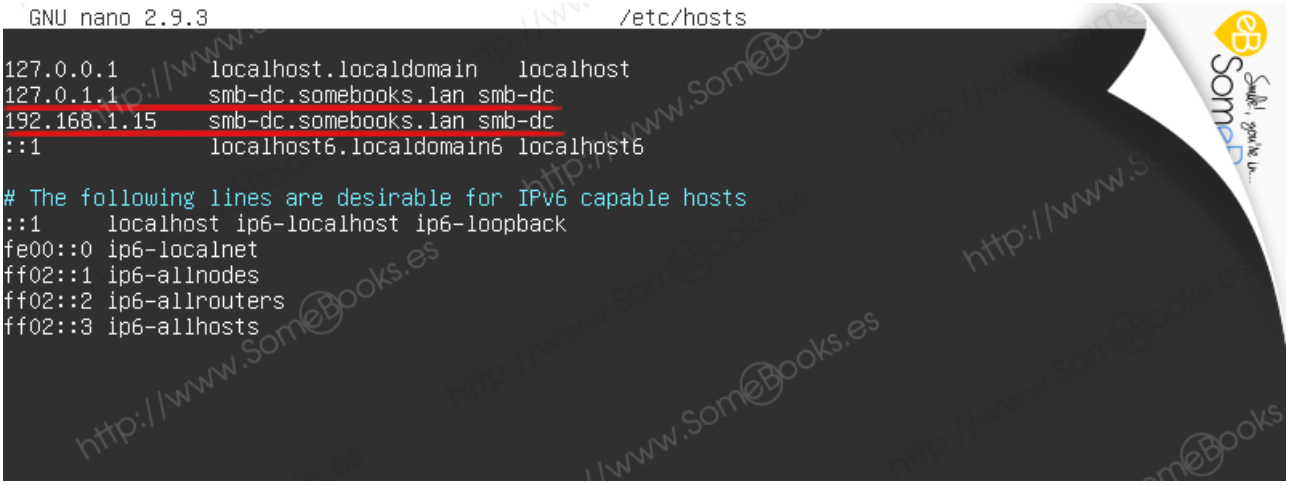
Una vez concluida la actualización, el segundo paso será adecuar el nombre del equipo a nuestras necesidades (recuerda que debe llamarse *smb-dc*).

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*).

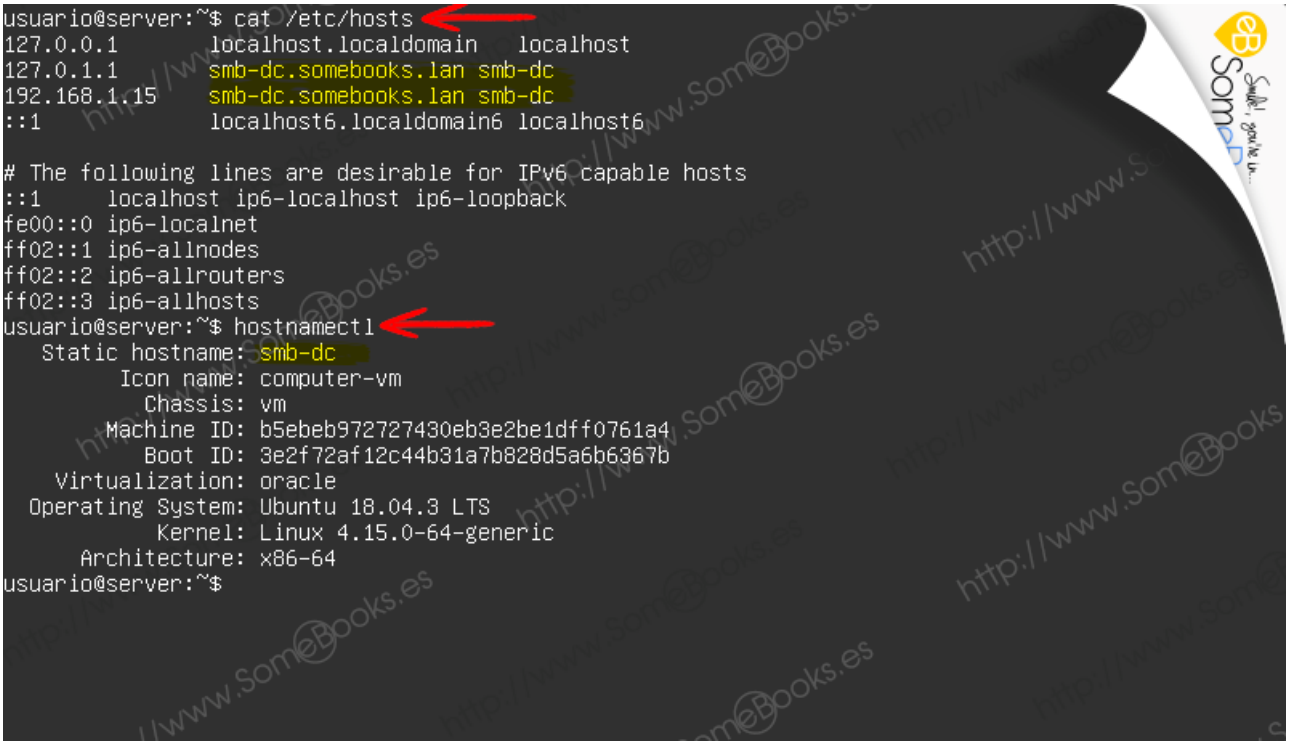
También incluiremos una referencia a la dirección IP estática que asignaremos al equipo en el siguiente apartado.

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.



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.

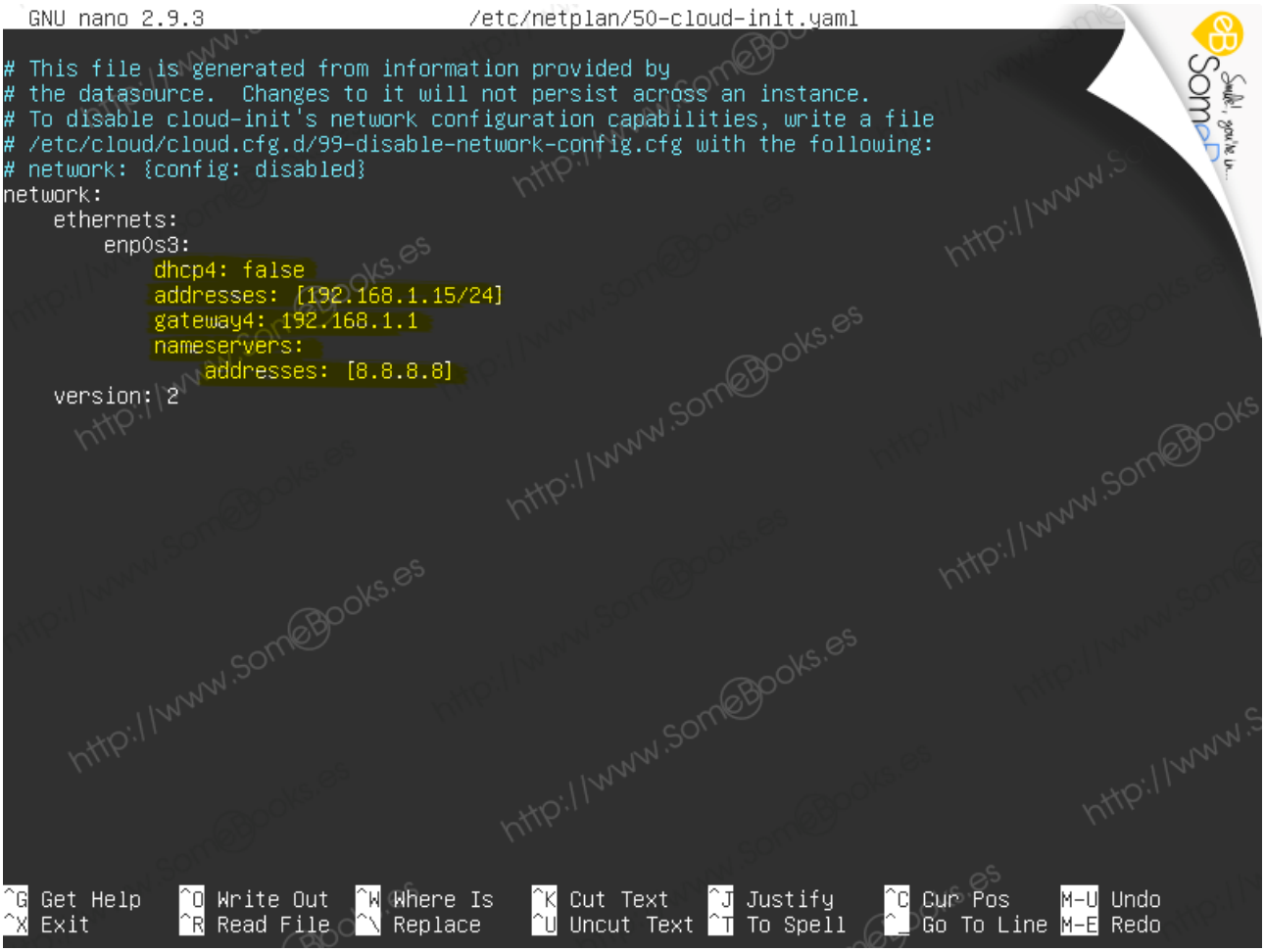
La imagen muestra la salida de ambos comandos.



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*.

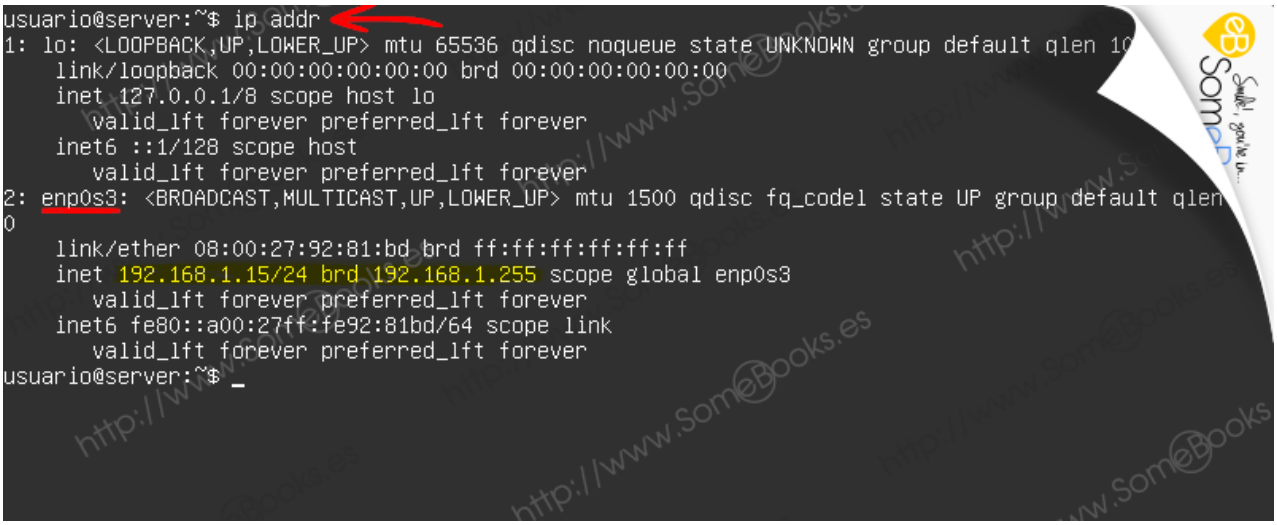
Para nuestro ejemplo, el archivo *Netplan* acabará teniendo este aspecto:



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

```
ip addr
```

Comprobamos la salida del comando



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

```
sudo reboot
```

Instalar los paquetes necesarios

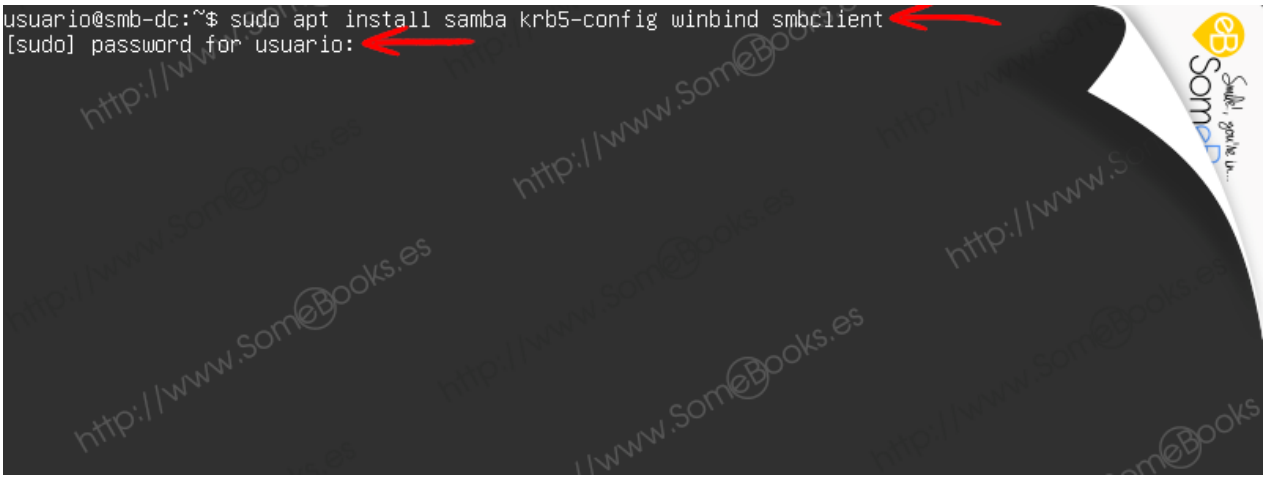
Además de los paquetes propios de *Samba* necesitaremos disponer de algunos otros, que deberán estar preinstalados antes de comenzar con el proceso de configuración, aunque todos ellos están en los repositorios. Son los siguientes:

- 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:

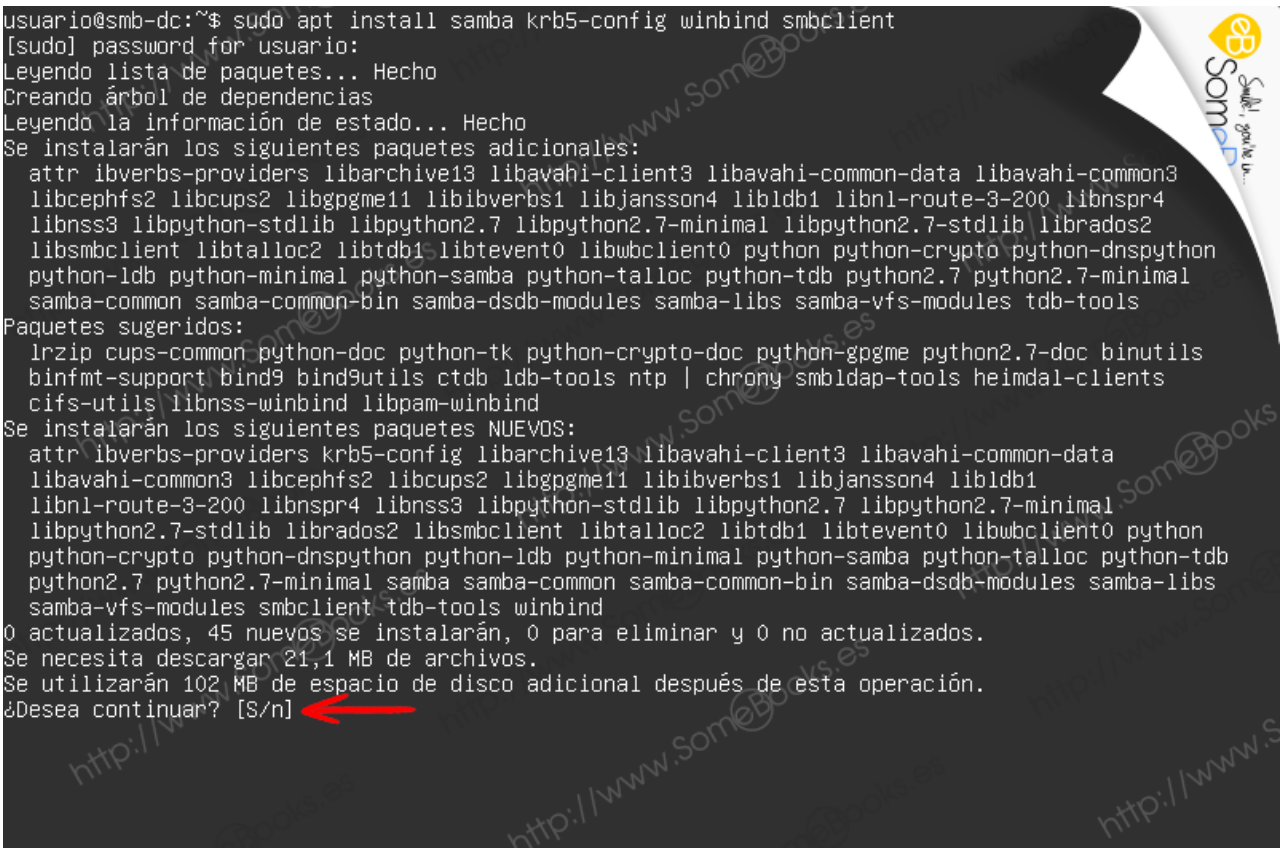
```
sudo apt install samba krb5-config winbind smbclient
```

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



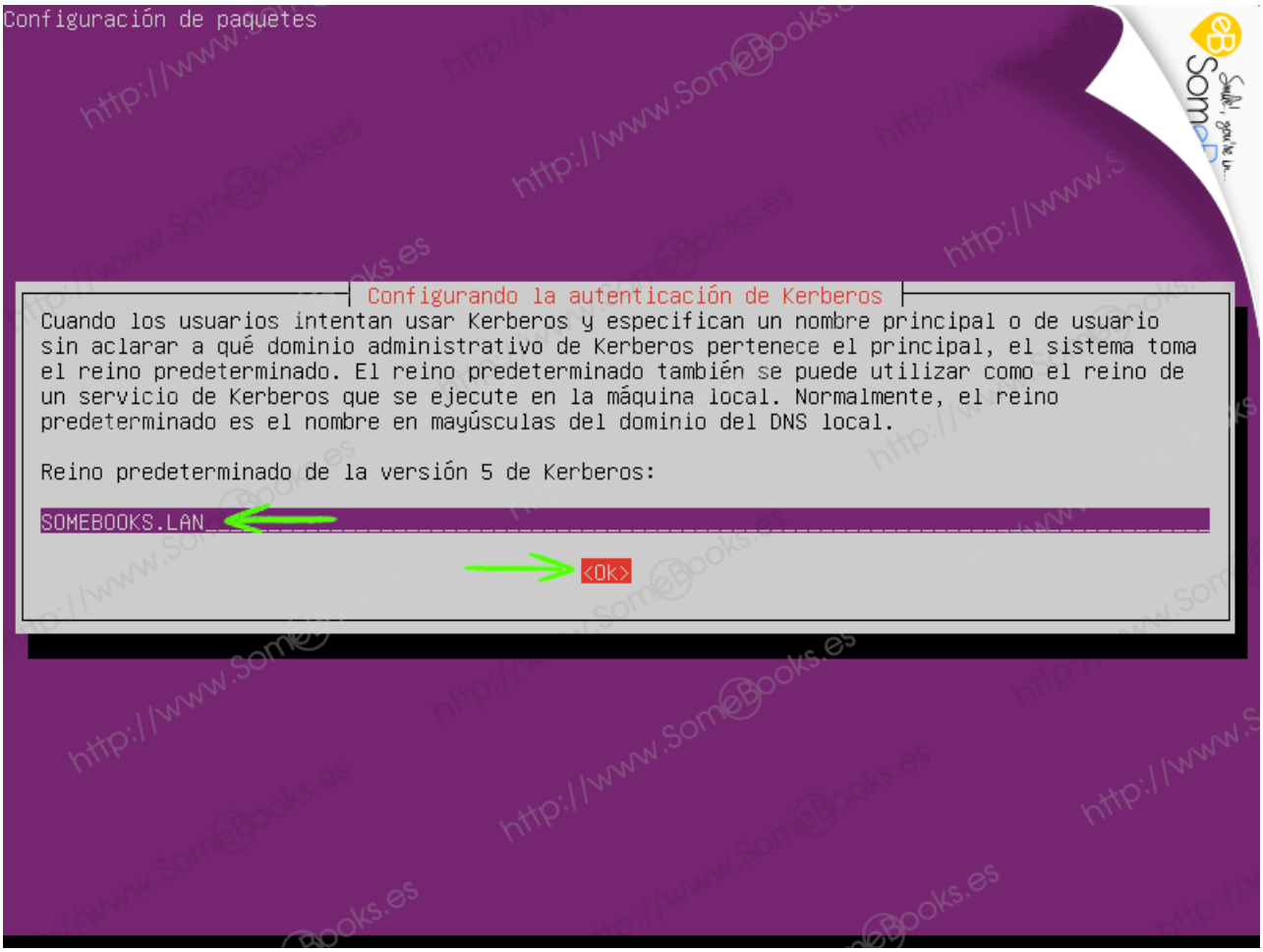
Al hacerlo, **`apt`** nos informará de los paquetes complementarios que deberán instalarse para completar la tarea.

Pulsamos la tecla `S` y después `Intro` para permitir la instalación.



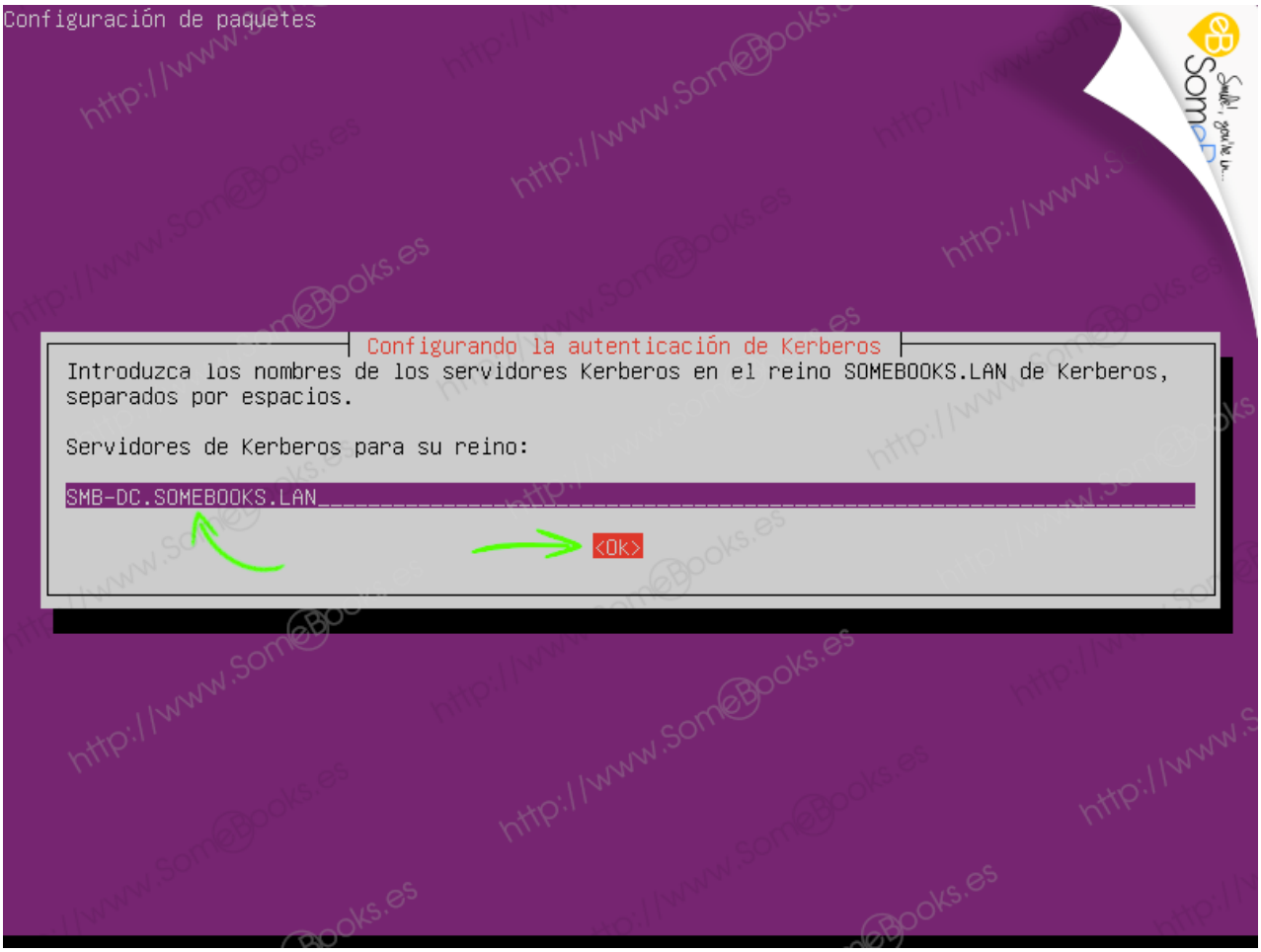
Cuando llegue el momento de instalar *Kerberos*, nos preguntará por el reino (*realm* en inglés). En realidad, se refiere al nombre del dominio y, en el caso de nuestro ejemplo, escribiremos SOMEBOOKS.LAN.

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



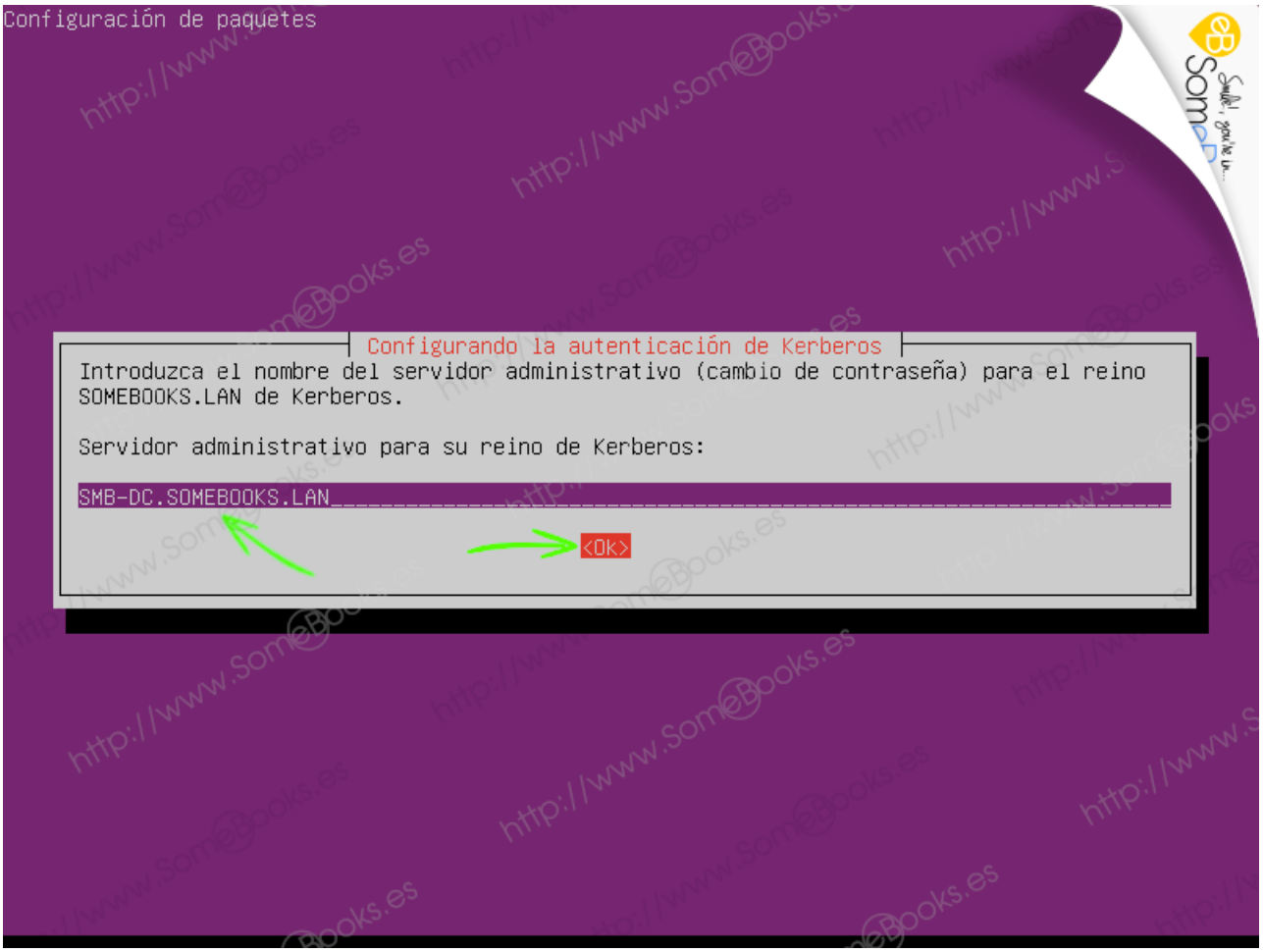
Después, el instalador nos solicita el nombre de los *servidores de Kerberos para nuestro reino*. En este caso, se refiere a los controladores de dominio que tengamos definidos. En nuestro ejemplo sólo tenemos uno y se llama SMB-DC.SOMEBOOKS.LAN, por lo que procedemos a escribir su nombre.

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



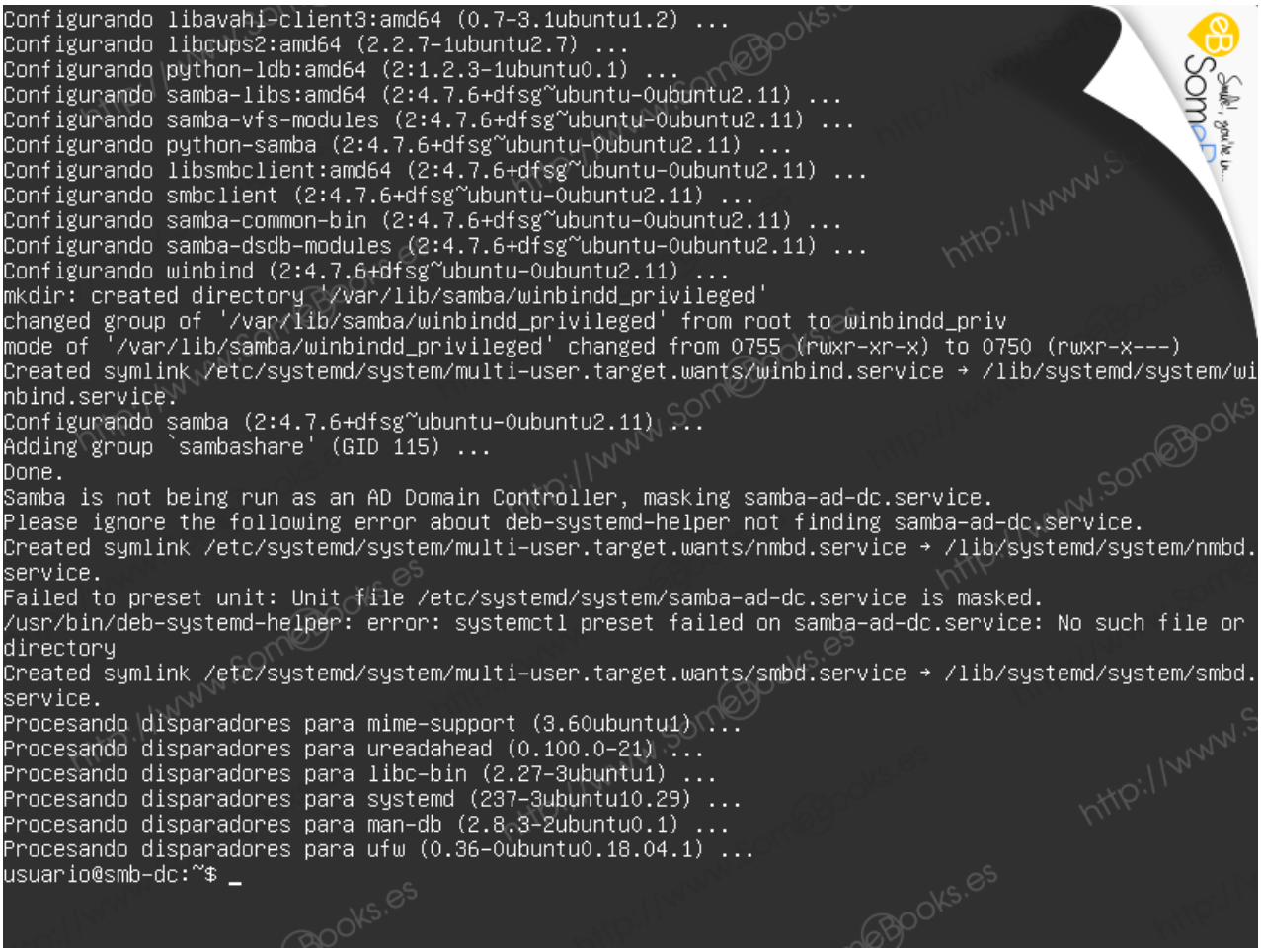
Por último, nos solicita el *servidor administrativo para nuestro reino de Kerberos*. como sólo tenemos uno, volvemos a escribir su nombre (SMB-DC.SOMEBOOKS.LAN).

Y de nuevo seleccionamos *Ok*.



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

Poco después, la instalación habrá terminado.



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`**.

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.

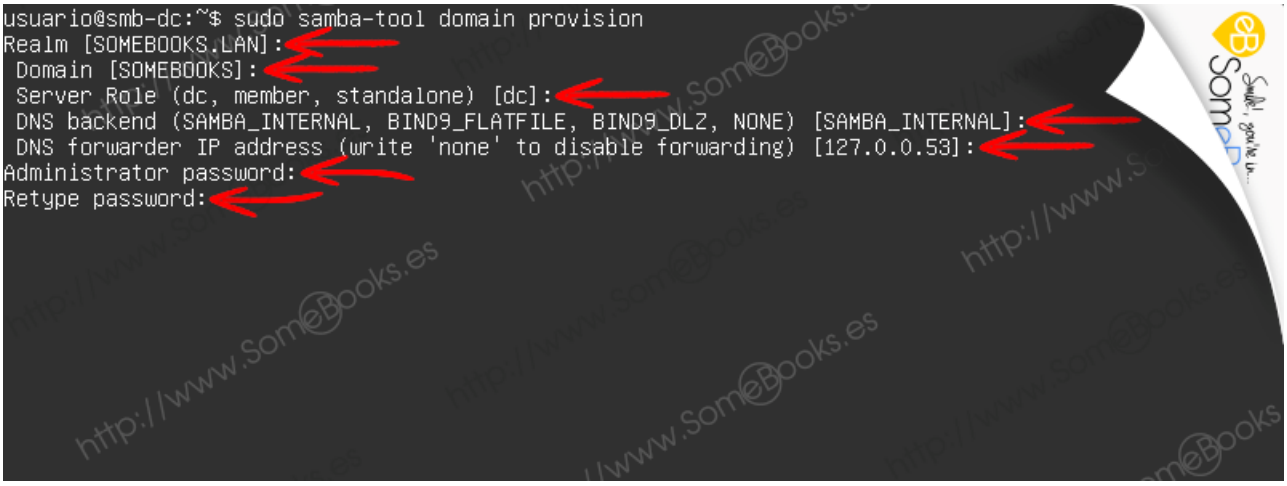
Como puedes ver en la imagen siguiente, los valores predeterminados del comando son los esperados para nuestro ejemplo:

- Realm: SOMEBOOKS.LAN (el nombre del dominio).
- Domain: SOMEBOOKS (el nombre *NetBIOS* del dominio).
- Server Role: dc (domain controler, 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.

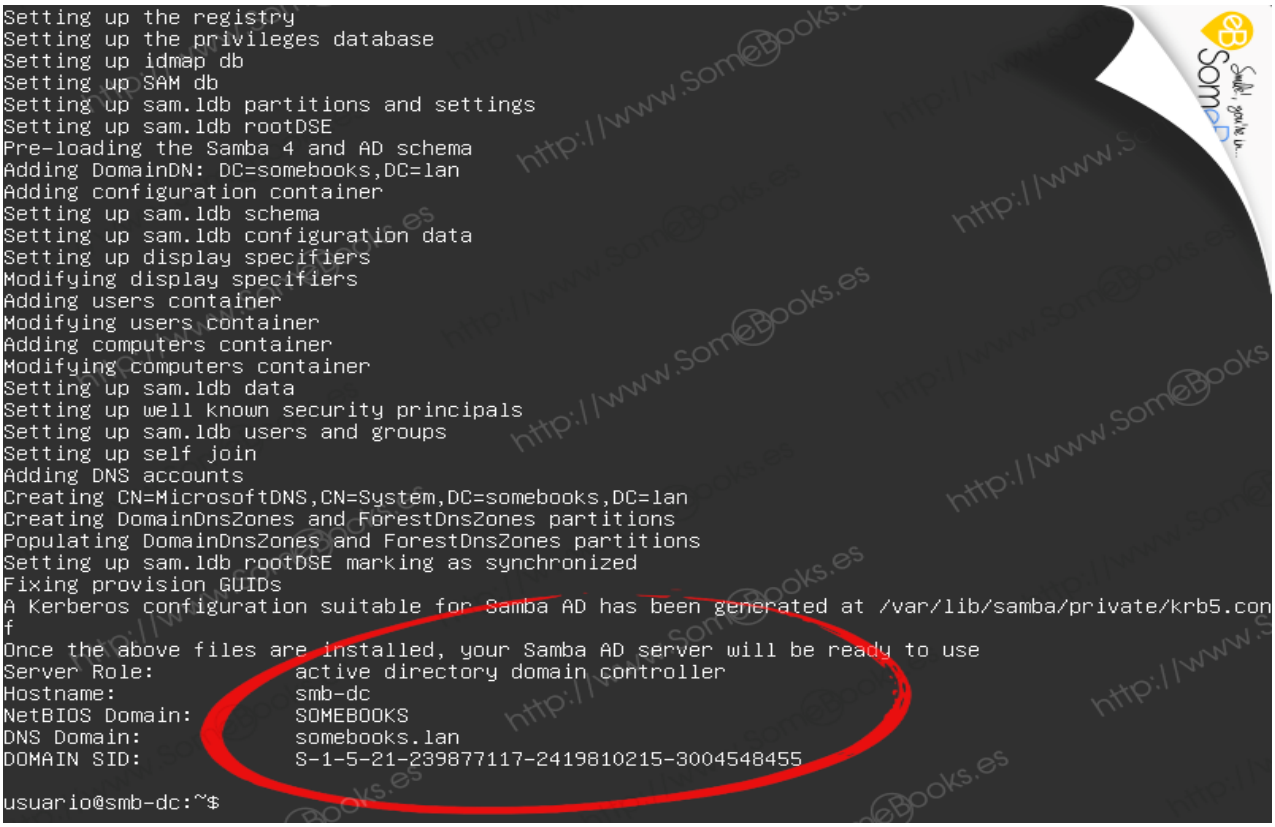
Por último, nos pedirá, como siempre por duplicado, la contraseña para el administrador.

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



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.



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*.

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:

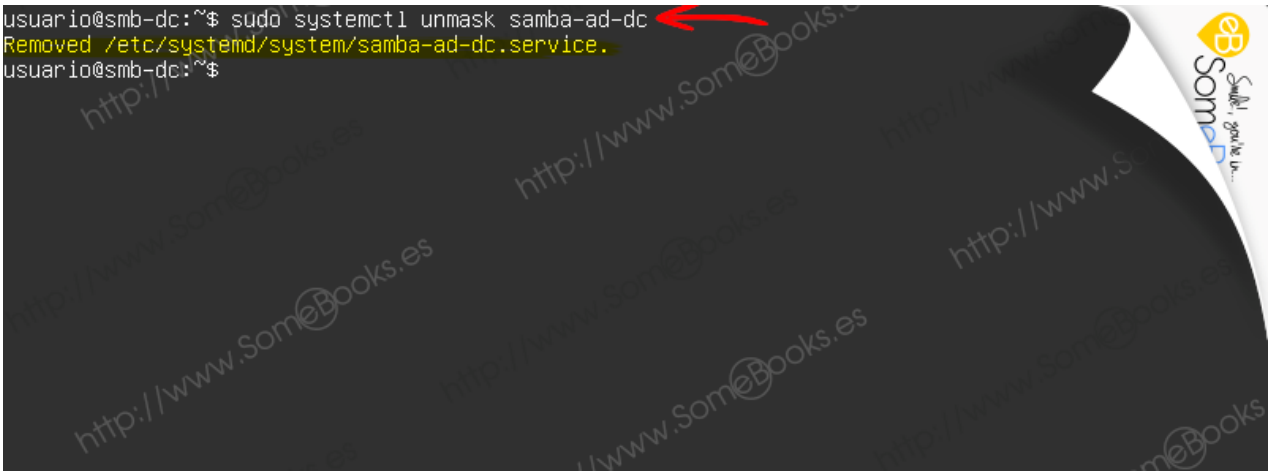
```
sudo systemctl disable smbd nmbd winbind systemd-resolved
```

Escribimos ambos comandos y comprobamos que se ejecutan correctamente.

A 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



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.

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.

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`.

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

Pulsamos la tecla `Y`.

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`.

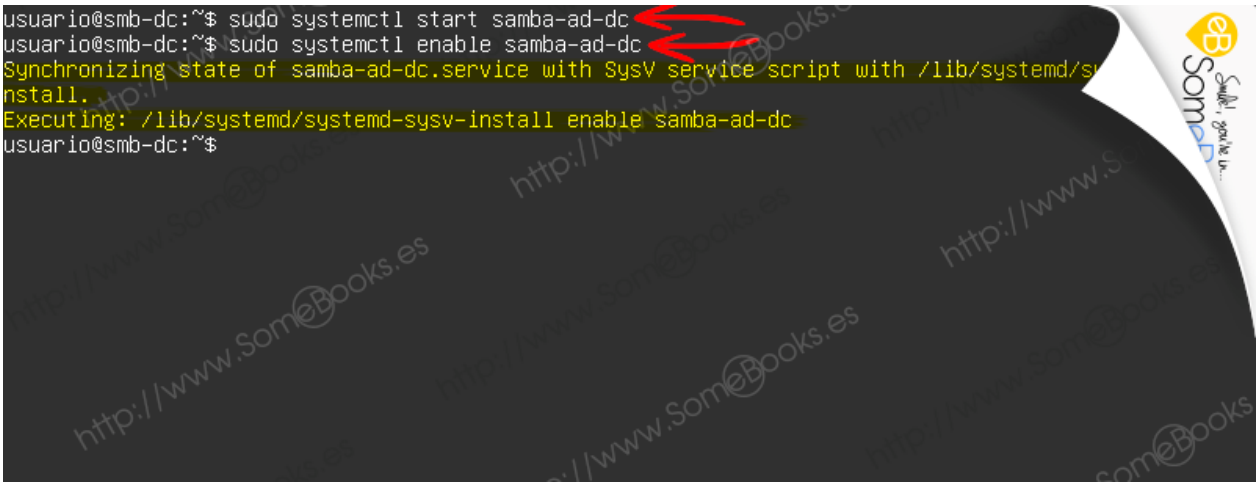
Y ya solo nos queda iniciar el servicio *samba-ad-dc*:

```
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.



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.

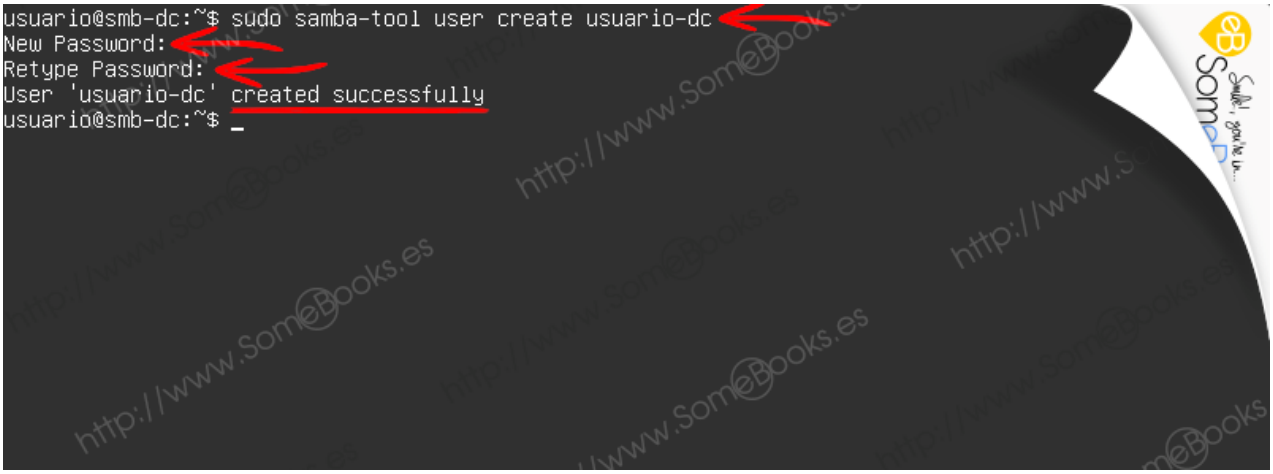
Salida del comando

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.



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:

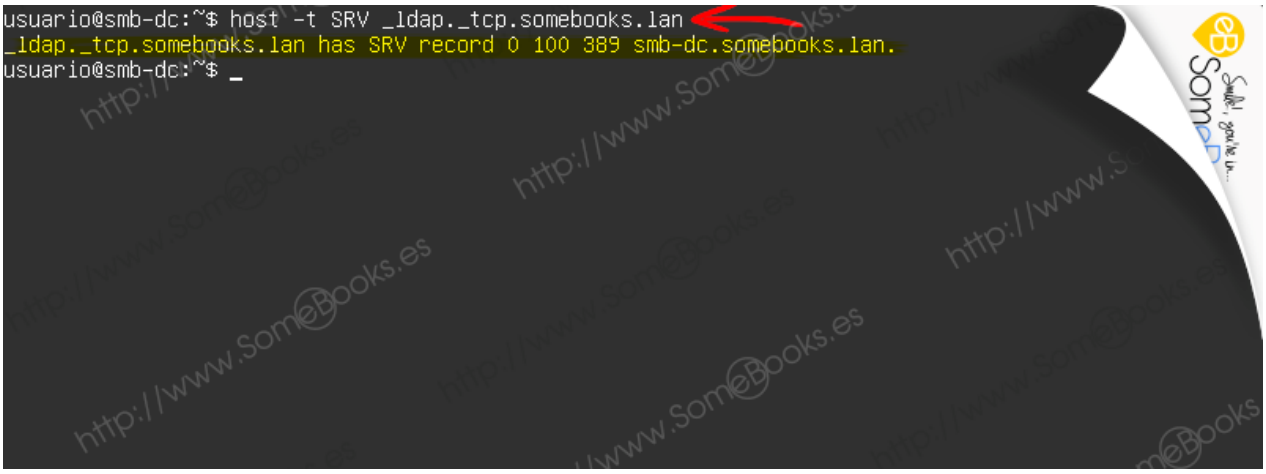
```
host -t SRV _ldap._tcp.somebooks.lan
```

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.



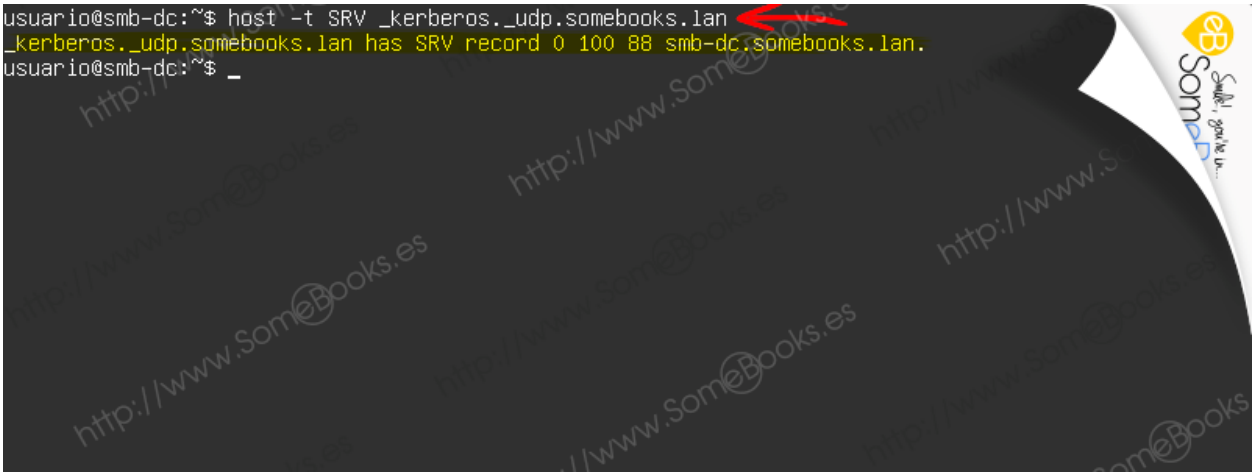
A continuación, comprobaremos el registro *SRV* para el protocolo *Kerberos* sobre *UDP*, para lo que usamos la siguiente orden:

```
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.



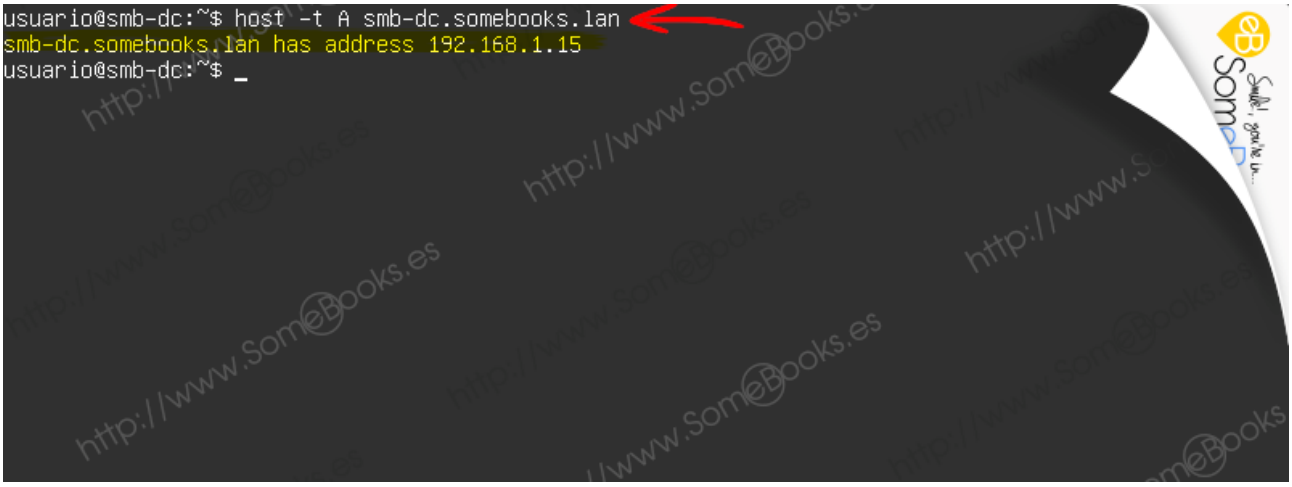
Por último, comprobamos la resolución del nombre de nuestro servidor:

```
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.



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*.

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:

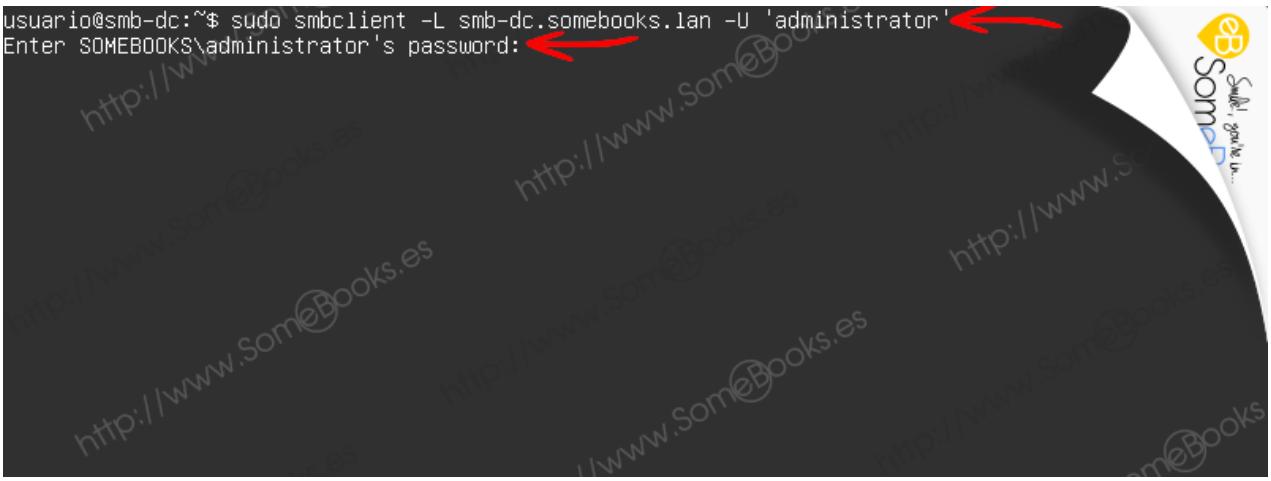
```
sudo smbclient -L smb-dc.somebooks.lan -U 'administrator'
```

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.

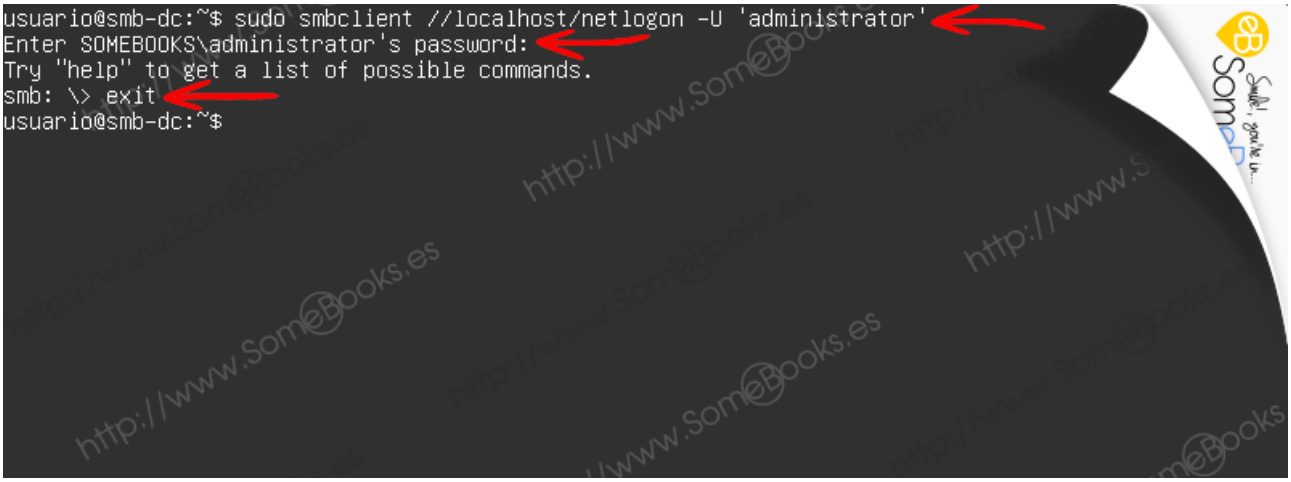
… Y el comando nos responde con la información solicitada.

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'
```

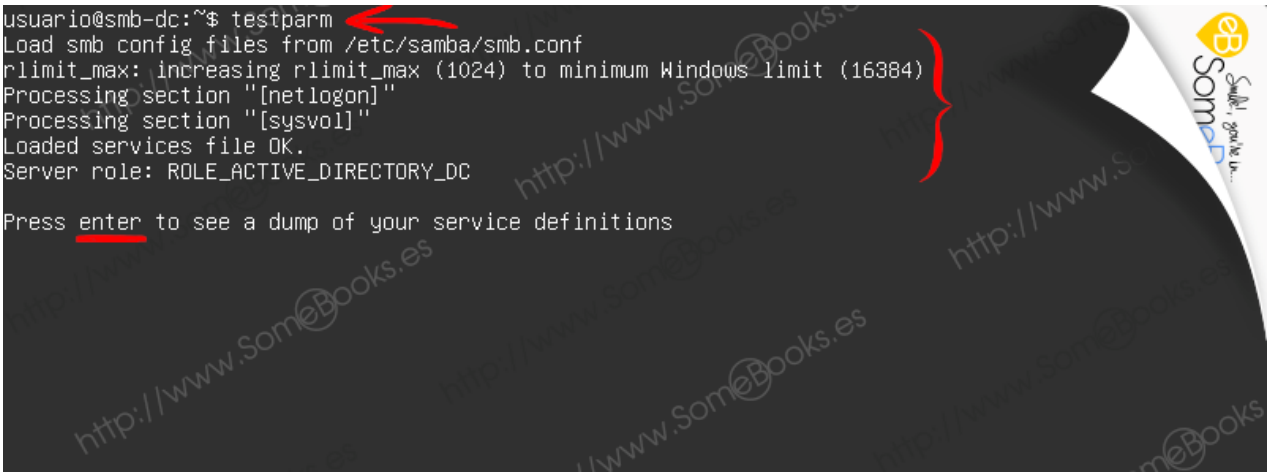
El *prompt* cambia para indicar que es el subsistema de *Samba* quien espera órdenes, por lo que la *autenticación* habrá sido correcta.

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



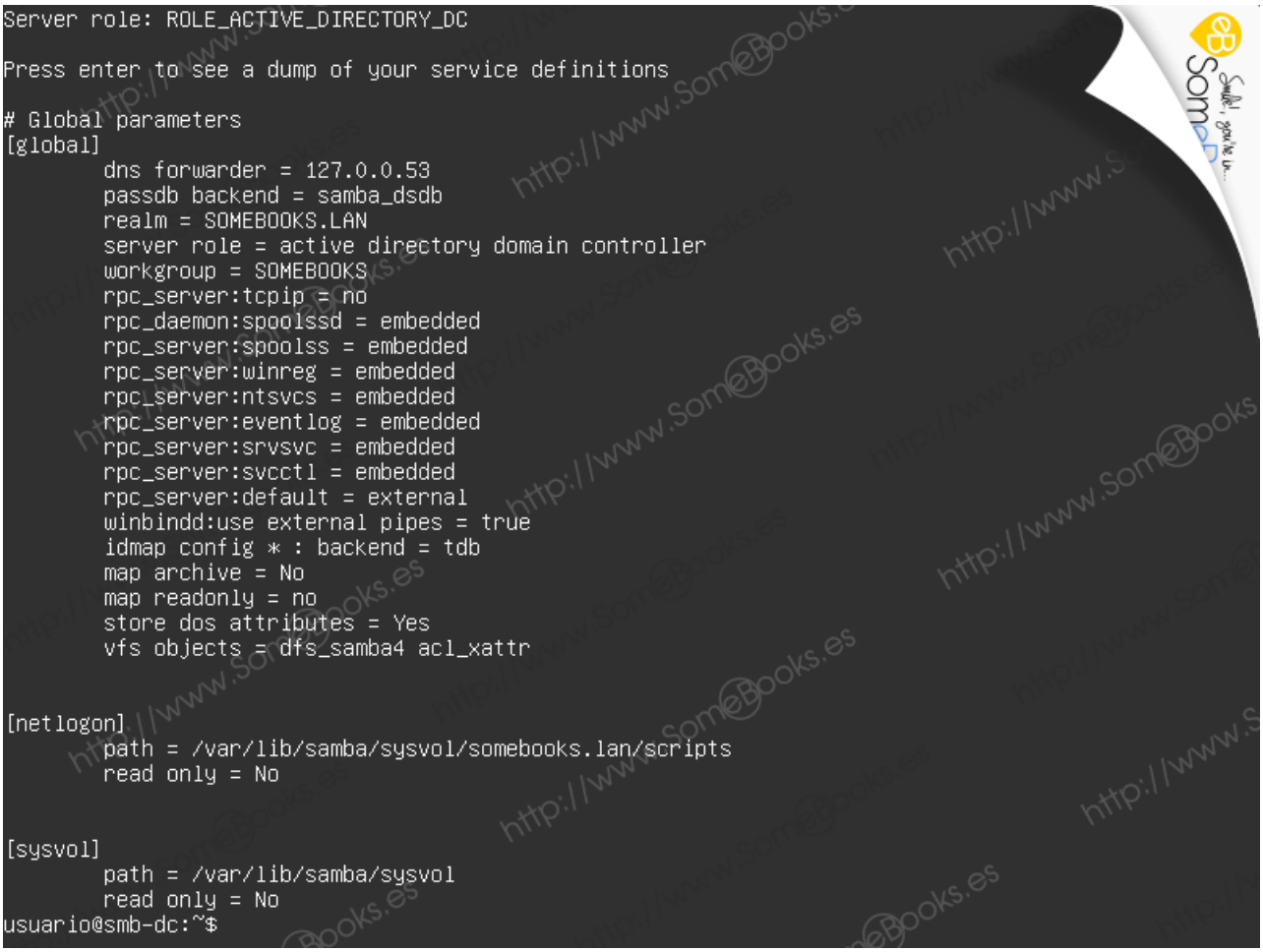
Para concluir podemos utilizar el comando **`testparm`**, que verifica la integridad del archivo de configuración de *Samba*:

Comprobamos que los datos son correctos



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

Pulsamos la tecla `Intro`.



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.