LABORATORY 03 - BASE PLATFORM AND APPLICATION LAYER PROTOCOLS

Laboratory 3 corresponding to the second term

Santiago Amaya Zapata

Julian Camilo Tinjacá Corredor

Escuela Colombiana de Ingeniería Julio Garavito

NETWORK ARCHITECTURE AND SERVICES

GROUP 3

Bogotá DC, Colombia

2025 - 2

Lab No.03 - BASE PLATFORM AND APPLICATION LAYER PROTOCOLS

Objective

Continue learning the installation of base software, particularly DNS and NTP services, complemented with knowledge of Shell programming.

Tools to be used

- Computers
- Internet Access
- Virtualization Software

Introduction

In modern enterprise networks, the proper configuration of infrastructure services is essential to ensure stability, connectivity, and availability of applications. Among these services, the **Domain Name System (DNS)** is one of the most critical components. DNS allows users and systems to translate human-friendly hostnames into machine-readable IP addresses, enabling seamless communication between distributed resources.

This laboratory practice focuses on the implementation of DNS servers in a heterogeneous environment, using different operating systems such as Solaris, Slackware, CentOS, and Windows Server. Working with multiple platforms provides students with a broader perspective of how the same service can be deployed, configured, and tested under diverse conditions. It also reinforces adaptability, since real-world IT infrastructures often integrate systems from different vendors. By configuring both primary and secondary DNS servers, as well as validating queries for internal and external domains, students will gain hands-on experience in setting up robust name resolution services.

Furthermore, the laboratory goes beyond simple configuration by emphasizing the importance of **verification and troubleshooting**. Tools such as **nslookup** and **dig**, will be employed to test the accuracy and reliability of DNS responses. This practice simulates real operational environments where administrators must not only deploy services but also validate their correct operation and detect possible misconfigurations.

Finally, the exercise also integrates **shell scripting and process management** to strengthen automation skills. Writing scripts to traverse the file system, monitor processes, or schedule tasks with cron builds a bridge between system administration and programming. This combination of service configuration and scripting prepares students to address real infrastructure challenges, where efficiency, reproducibility, and accuracy are indispensable for managing large-scale systems.

Theoretical Framework

The correct functioning of computer networks relies on a set of basic services that ensure communication, reliability, and coordination among distributed systems. One of the most relevant is the **Domain Name System (DNS)**.

Domain Name System (DNS):
DNS is a hierarchical and distributed naming system that translates human-readable hostnames (such as *www.example.com*) into numerical IP addresses (such as *192.168.1.1*), which are required for data routing at the network level. The DNS hierarchy is structured in different levels: root servers, top-level domains (TLDs), authoritative servers, and recursive resolvers. Within a local infrastructure, administrators can configure **primary (master)** and **secondary (slave)** DNS servers. The primary server stores the original zone files that define the mapping between hostnames and IP addresses, while the secondary server maintains synchronized copies, thus providing redundancy and fault tolerance. Correct DNS configuration is critical, as name resolution errors can prevent access to services and applications.

System Administration in Heterogeneous Environments:
In practice, enterprise networks often integrate multiple operating systems. Solaris, Slackware, CentOS, and Windows Server represent different philosophies and management tools, but they must coexist within the same infrastructure. Configuring DNS in these environments requires understanding the common principles and the platform-specific implementations. This experience allows system administrators to adapt to varied contexts and guarantee interoperability.

Verification and Troubleshooting Tools:
Once services are configured, their validation becomes essential. Tools such as ping, **nslookup**, and **dig** provide mechanisms to check connectivity and name resolution, while log files allow administrators to detect failures or anomalies. Similarly, the use of **cron** for scheduling tasks and shell scripting for automation reinforces the capacity to maintain reliable and efficient systems over time.

In summary, the theoretical foundation of this practice lies in understanding DNS as the backbone of communication, combined with the operational skills to deploy them in diverse environments, validate their operation, and automate their administration.

Installation of Base Software

Perform the activities listed below on the application layer protocols: DNS, as well as the specified Shell commands.

1. Linux DNS Server - BIND

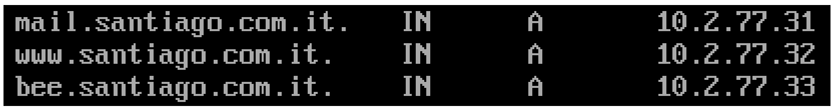
As we have seen in class, a key service in an enterprise environment is the Domain Name Resolution - DNS service. In this lab, we will configure this service using test domains. The domains to be configured, depending on the number of students in the group, are:

santiago.com.it

julian.org.uk

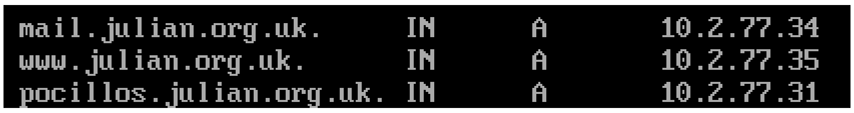
For each domain, the following must be defined:
- 3 server names with their corresponding IPv4 addresses (Use the ones from the range assigned at the beginning of the semester). For now, only name resolution will be visible; as we configure other services, we will add them to the DNS, and we will be able to access those servers by name.

santiago.com.it IPv4 server names:



```
mail.santiago.com.it.   IN      A       10.2.77.31
www.santiago.com.it.    IN      A       10.2.77.32
bee.santiago.com.it.    IN      A       10.2.77.33
```

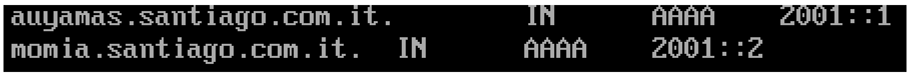
julian.org.uk IPv4 server names:



```
mail.julian.org.uk.     IN      A       10.2.77.34
www.julian.org.uk.      IN      A       10.2.77.35
pocillos.julian.org.uk. IN      A       10.2.77.31
```

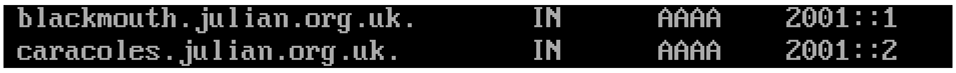
- 2 servers with their corresponding IPv6 addresses.

santiago.com.it IPv6 server names:



```
auyamas.santiago.com.it.        IN      AAAA    2001::1
momia.santiago.com.it.   IN      AAAA    2001::2
```

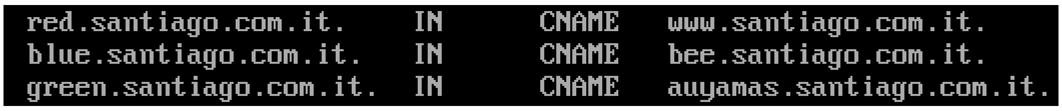
julian.org.uk IPv6 server names:



```
blackmouth.julian.org.uk.       IN      AAAA    2001::1
caracoles.julian.org.uk.        IN      AAAA    2001::2
```

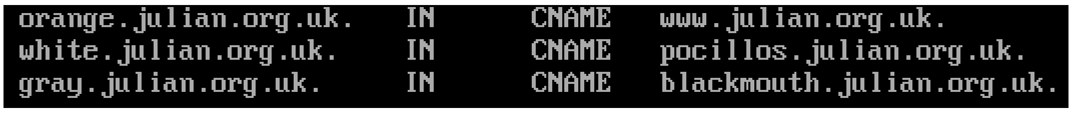
- 2 aliases for 2 servers with IPv4 addresses and 1 server with an IPv6 address (Choose any names you prefer).

santiago.com.it aliases:



```
red.santiago.com.it.    IN      CNAME   www.santiago.com.it.
blue.santiago.com.it.   IN      CNAME   bee.santiago.com.it.
green.santiago.com.it.  IN      CNAME   auyamas.santiago.com.it.
```

julian.org.uk aliases:



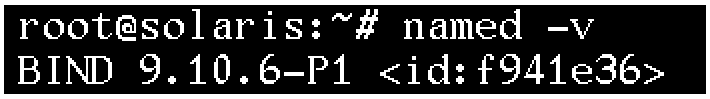
```
orange.julian.org.uk.   IN      CNAME   www.julian.org.uk.
white.julian.org.uk.    IN      CNAME   pocillos.julian.org.uk.
gray.julian.org.uk.     IN      CNAME   blackmouth.julian.org.uk.
```

The implementation should be carried out using virtual machines: one Solaris, one Windows Server, one Linux Slackware, and one CentOS (groups of 3 students), two of them located on one physical computer and the others on the other physical computer assigned to the groups. The installation should be done as follows:

- For the domain student 1.com.it:

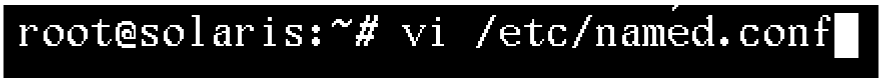
- Primary DNS server on a Solaris virtual machine.

First verify that BIND is installed



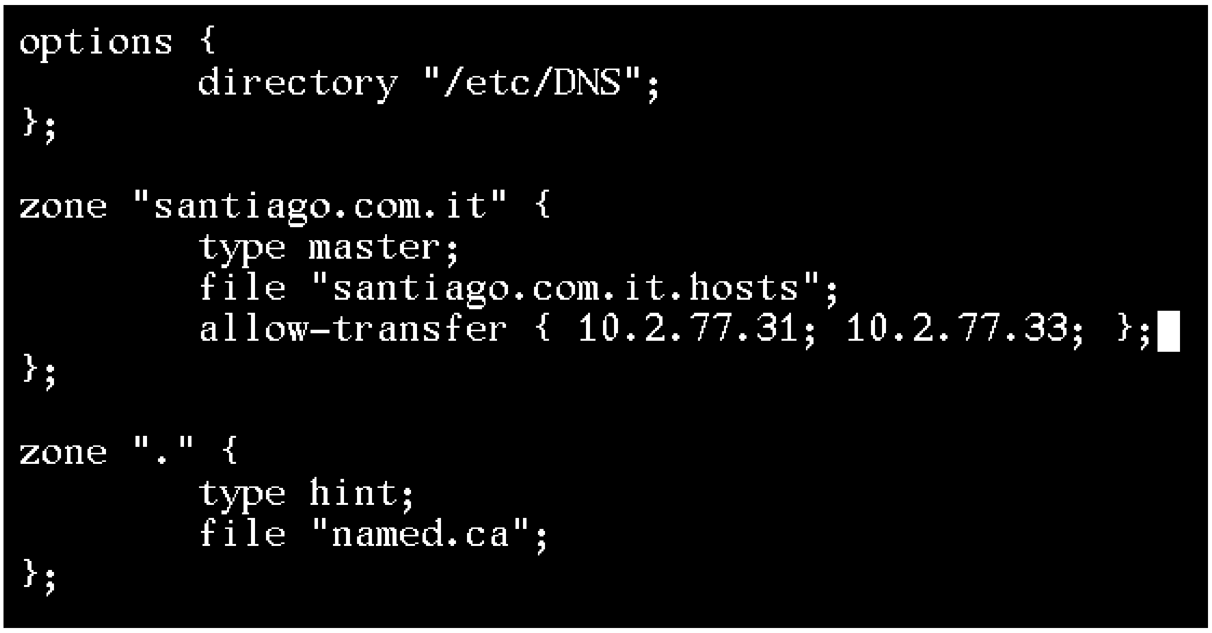
```
root@solaris:~# named -v
BIND 9.10.6-P1 <id:f941e36>
```

Now we create the file /etc/named.conf, This file is used to define the main configuration of the BIND DNS server



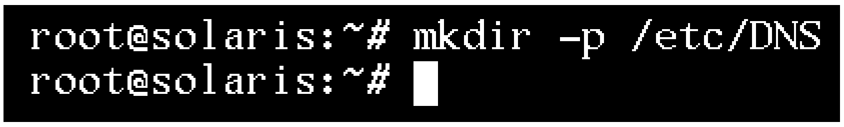
```
root@solaris:~# vi /etc/named.conf
```

The named.conf file is the main configuration file for BIND, and it defines how the DNS server operates. It specifies global options, logging rules, access controls, and the zones the server will manage, including whether it acts as a primary or secondary for each zone.



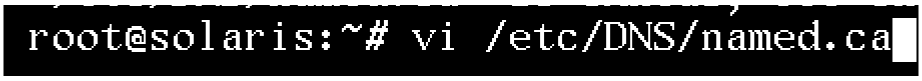
```
options {
        directory "/etc/DNS";
};

zone "santiago.com.it" {
        type master;
        file "santiago.com.it.hosts";
        allow-transfer { 10.2.77.31; 10.2.77.33; };
};

zone "." {
        type hint;
        file "named.ca";
};
```

Now, we created the directory /etc/DNS, this directory provides a dedicated location to store DNS zone files and related configurations, helping organize the BIND setup outside the default path.



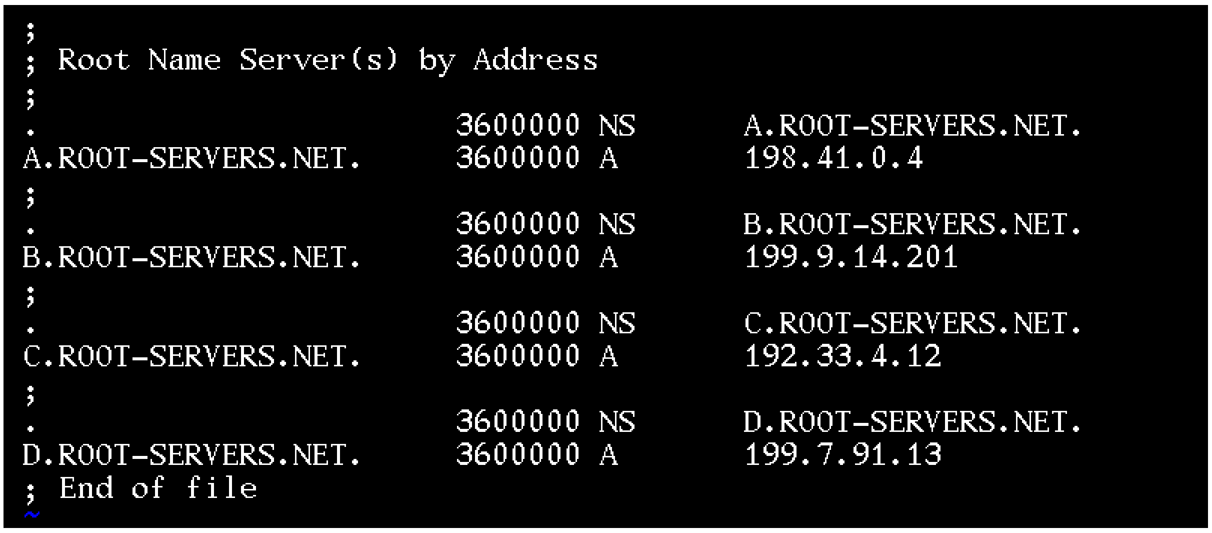
```
root@solaris:~# mkdir -p /etc/DNS
root@solaris:~#
```

Now inside that directory we just created, we create the named.ca file, this file serves as the root hints file, containing a list of root DNS servers that BIND uses to begin resolving queries when it is configured as a caching or recursive server.



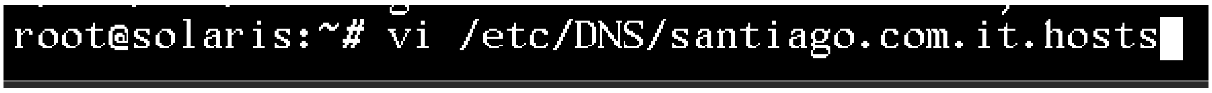
```
root@solaris:~# vi /etc/DNS/named.ca
```

Already in the VI editor, we add these root DNS servers



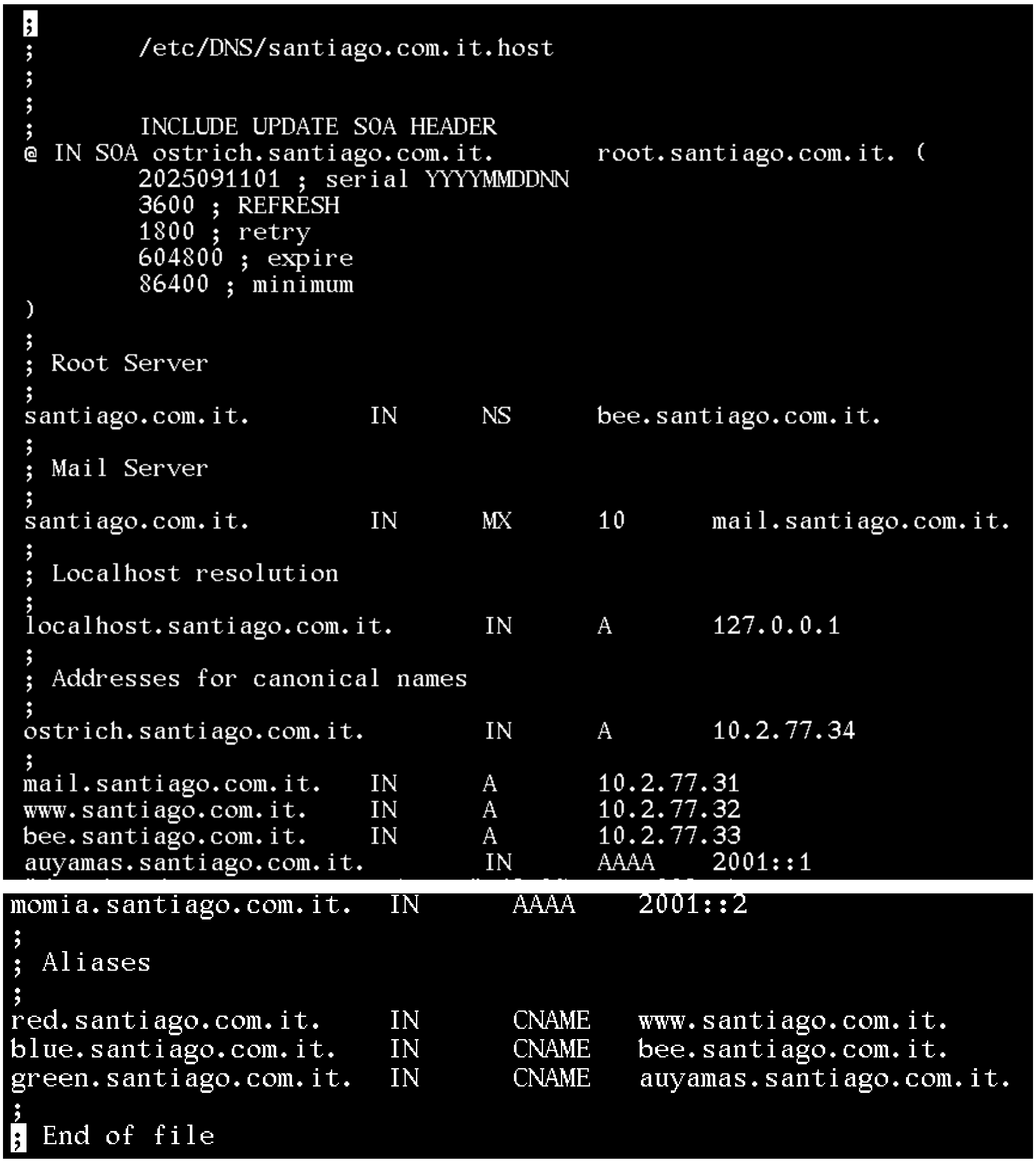
```
;
; Root Name Server(s) by Address
;
.                       3600000 NS      A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.     3600000 A       198.41.0.4
;
.                       3600000 NS      B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.     3600000 A       199.9.14.201
;
.                       3600000 NS      C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.     3600000 A       192.33.4.12
;
.                       3600000 NS      D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.     3600000 A       199.7.91.13
; End of file
~
```

Now we create the santiago.com.it.hosts file inside /etc/DNS/. The santiago.com.it.hosts file is a zone file that stores the DNS records (such as SOA, NS, A, and MX) for the domain **santiago.com.it**, allowing the DNS server to resolve names within that domain.



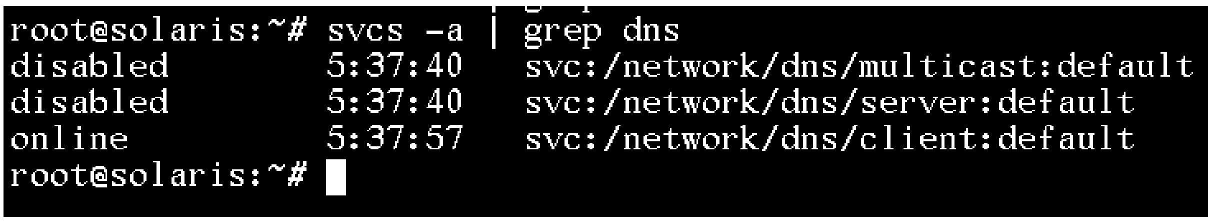
```
root@solaris:~# vi /etc/DNS/santiago.com.it.hosts
```

We write the different server names defined at the beginning of the laboratory, with their corresponding IP, either IPv4 or IPv6, in addition to the aliases associated with different server names already defined.



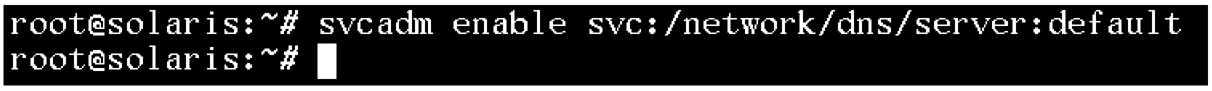
```
;
;       /etc/DNS/santiago.com.it.host
;
;
;       INCLUDE UPDATE SOA HEADER
@ IN SOA ostrich.santiago.com.it.       root.santiago.com.it. (
        2025091101 ; serial YYYYMMDDNN
        3600 ; REFRESH
        1800 ; retry
        604800 ; expire
        86400 ; minimum
)
;
; Root Server
;
santiago.com.it.        IN      NS      bee.santiago.com.it.
;
; Mail Server
;
santiago.com.it.        IN      MX      10      mail.santiago.com.it.
;
; Localhost resolution
;
localhost.santiago.com.it.      IN      A       127.0.0.1
;
; Addresses for canonical names
;
ostrich.santiago.com.it.        IN      A       10.2.77.34
;
mail.santiago.com.it.   IN      A       10.2.77.31
www.santiago.com.it.    IN      A       10.2.77.32
bee.santiago.com.it.    IN      A       10.2.77.33
auyamas.santiago.com.it.        IN      AAAA    2001::1
momia.santiago.com.it.  IN      AAAA    2001::2
;
; Aliases
;
red.santiago.com.it.    IN      CNAME   www.santiago.com.it.
blue.santiago.com.it.   IN      CNAME   bee.santiago.com.it.
green.santiago.com.it.  IN      CNAME   auyamas.santiago.com.it.
;
; End of file
```

The **svcs -a | grep dns** command in Solaris is used to check the status of DNS-related services managed by SMF (Service Management Facility), which helps verify whether the DNS server service is enabled, online, disabled, or in maintenance mode.



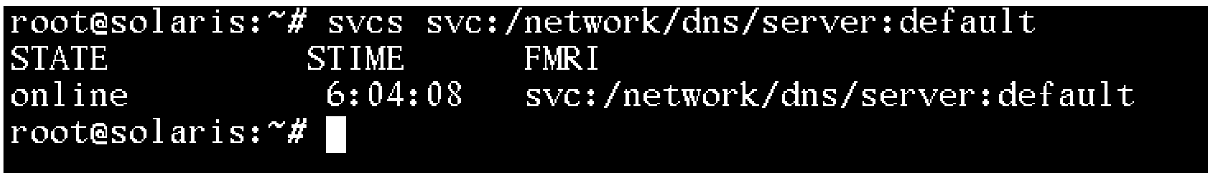
```
root@solaris:~# svcs -a | grep dns
disabled        5:37:40   svc:/network/dns/multicast:default
disabled        5:37:40   svc:/network/dns/server:default
online          5:37:57   svc:/network/dns/client:default
root@solaris:~#
```

Since the DNS server service is disabled, we enable it with the following command:



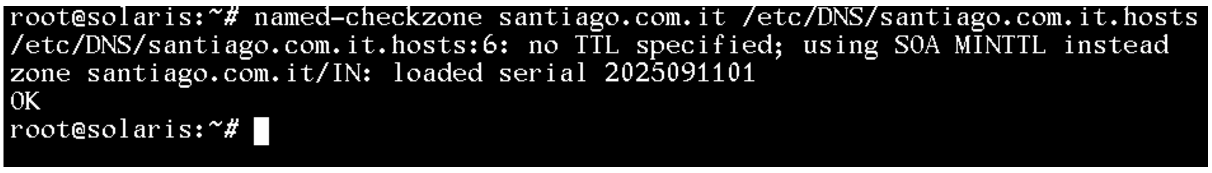
```
root@solaris:~# svcadm enable svc:/network/dns/server:default
root@solaris:~#
```

With the following command check if it was activated correctly:



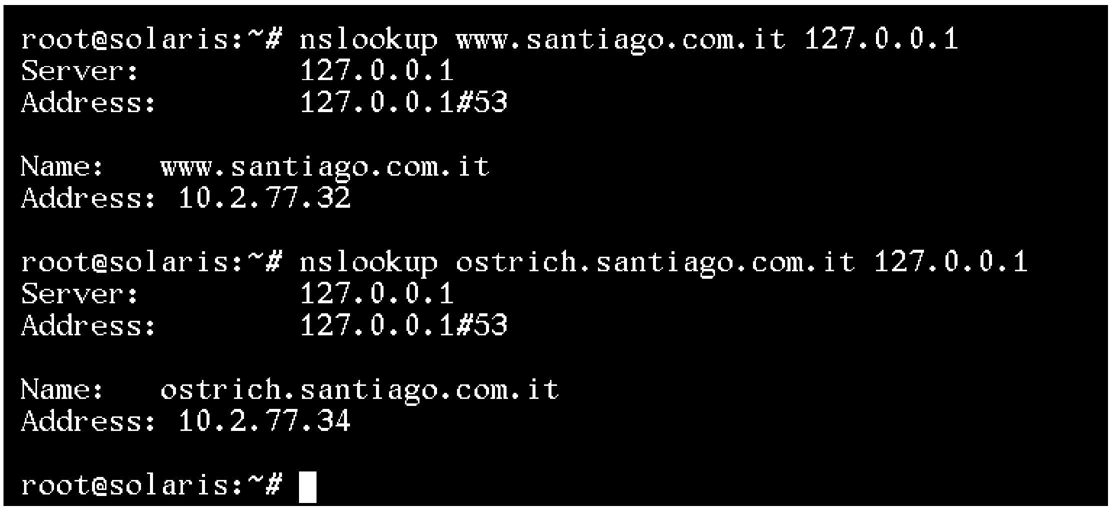
```
root@solaris:~# svcs svc:/network/dns/server:default
STATE          STIME      FMRI
online          6:04:08   svc:/network/dns/server:default
root@solaris:~#
```

After that, we check the file for syntax errors using the **named-checkzone** command. This is important because it ensures that the resources stored there are valid before they are loaded by BIND.



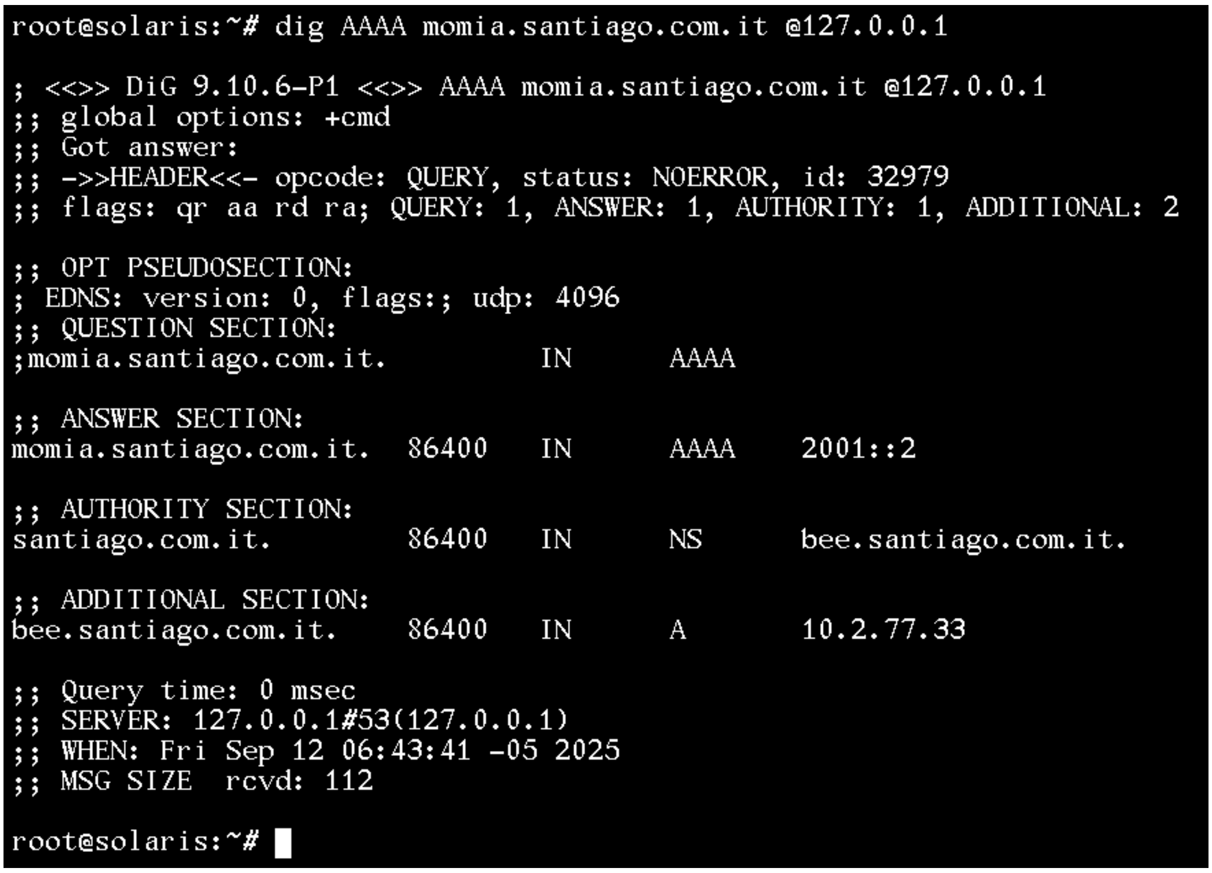
```
root@solaris:~# named-checkzone santiago.com.it /etc/DNS/santiago.com.it.hosts
/etc/DNS/santiago.com.it.hosts:6: no TTL specified; using SOA MINTTL instead
zone santiago.com.it/IN: loaded serial 2025091101
OK
root@solaris:~#
```

After that, we can verify that the primary server is working using the **nslookup** command as follows:



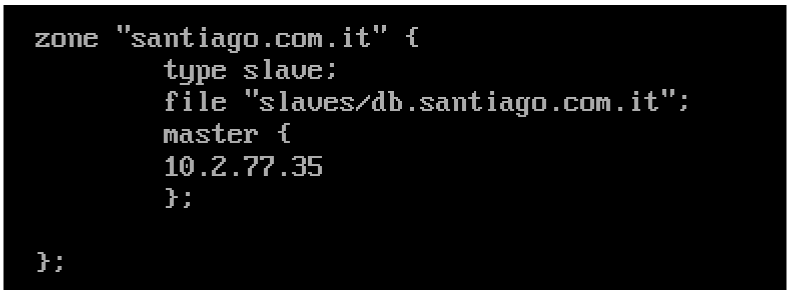
```
root@solaris:~# nslookup www.santiago.com.it 127.0.0.1
Server:         127.0.0.1
Address:        127.0.0.1#53

Name:   www.santiago.com.it
Address: 10.2.77.32

root@solaris:~# nslookup ostrich.santiago.com.it 127.0.0.1
Server:         127.0.0.1
Address:        127.0.0.1#53

Name:   ostrich.santiago.com.it
Address: 10.2.77.34

root@solaris:~#
```

Another option is to use **dig**, this command is used to query the DNS server directly, allowing you to test whether the primary DNS on the same machine is correctly serving its zones and responding with the expected records.



```
root@solaris:~# dig AAAA momia.santiago.com.it @127.0.0.1

; <<>> DiG 9.10.6-P1 <<>> AAAA momia.santiago.com.it @127.0.0.1
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32979
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;momia.santiago.com.it.         IN      AAAA

;; ANSWER SECTION:
momia.santiago.com.it.  86400   IN      AAAA    2001::2

;; AUTHORITY SECTION:
santiago.com.it.        86400   IN      NS      bee.santiago.com.it.

;; ADDITIONAL SECTION:
bee.santiago.com.it.    86400   IN      A       10.2.77.33

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Fri Sep 12 06:43:41 -05 2025
;; MSG SIZE  rcvd: 112

root@solaris:~#
```

- Secondary DNS servers on a Linux Slackware virtual machine and Windows Server.
Linux Slackware:
In the /etc/named.conf we define the slave server zone for the domain santiago.com.it

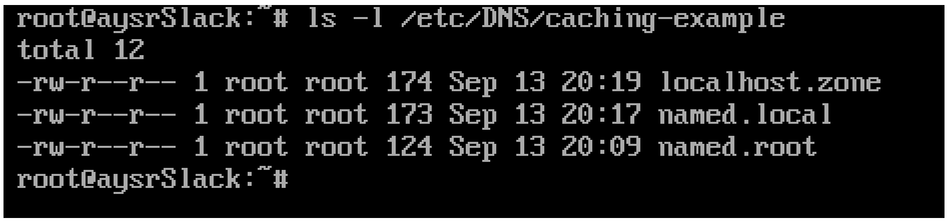


```
zone "santiago.com.it" {
        type slave;
        file "slaves/db.santiago.com.it";
        master {
        10.2.77.35
        };

};
```

Now, create the directory /etc/DNS/slaves, this directory provides a location for storing zone files received by the DNS server when it acts as a **secondary (slave)**, keeping them separate from primary zone files. The command **chown named:named /etc/DNS/slaves** sets the owner and group of the directory to the **named** user, ensuring that only the DNS service can write to it. The command **chmod 750 /etc/DNS/slaves** sets the permissions so that the owner can read, write, and execute, the group can read and execute, and others have no access, securing the slave zone files.

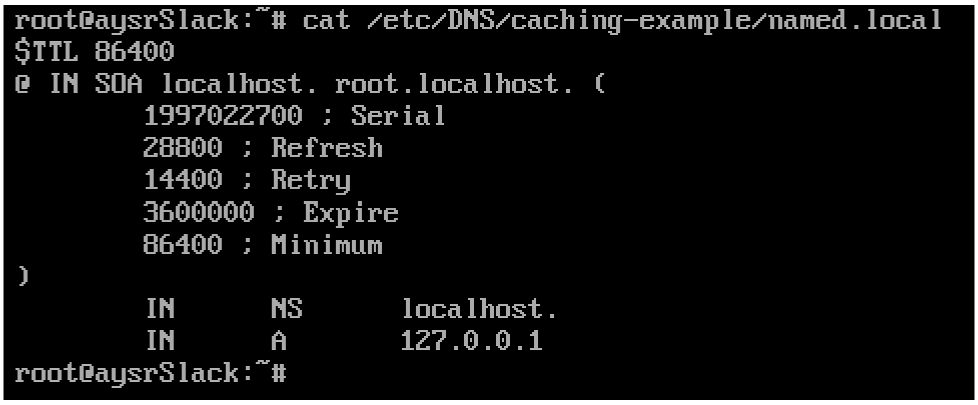
Since zone files are looked up in /etc/DNS, we need to create the following files inside of /etc/DNS/caching-example/
- /etc/DNS/caching-example/named.local
- /etc/DNS/caching-example/localhost.zone

Creating these files provides the root hints and local zone files needed for a caching DNS server to resolve queries and handle the local host domain, ensuring proper operation of BIND in caching mode.

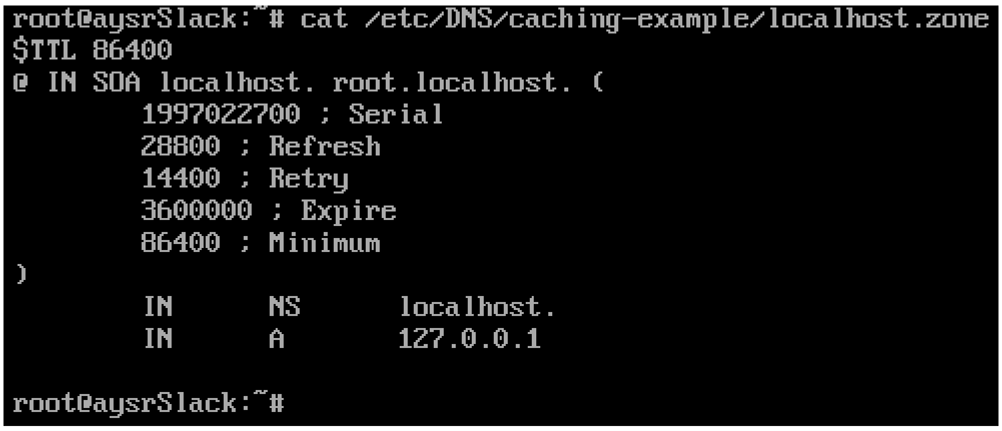


```
root@aysrSlack:~# ls -l /etc/DNS/caching-example
total 12
-rw-r--r-- 1 root root 174 Sep 13 20:19 localhost.zone
-rw-r--r-- 1 root root 173 Sep 13 20:17 named.local
-rw-r--r-- 1 root root 124 Sep 13 20:09 named.root
root@aysrSlack:~#
```

And we write on them the following:

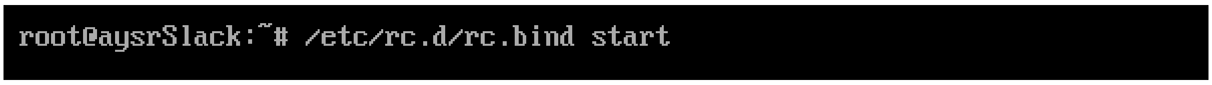


```
root@aysrSlack:~# cat /etc/DNS/caching-example/named.local
$TTL 86400
@ IN SOA localhost. root.localhost. (
        1997022700 ; Serial
        28800 ; Refresh
        14400 ; Retry
        3600000 ; Expire
        86400 ; Minimum
)
        IN      NS      localhost.
        IN      A       127.0.0.1
root@aysrSlack:~#
```



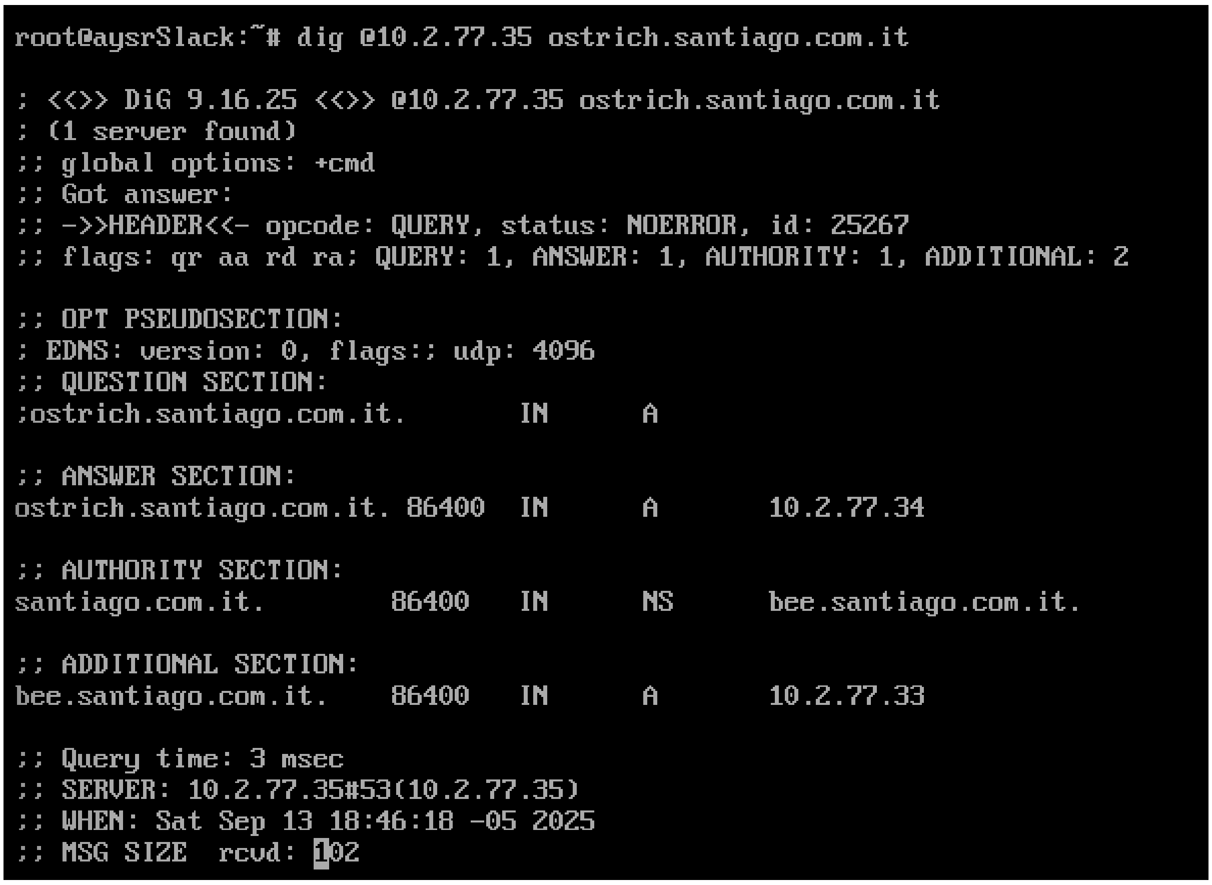
```
root@aysrSlack:~# cat /etc/DNS/caching-example/localhost.zone
$TTL 86400
@ IN SOA localhost. root.localhost. (
        1997022700 ; Serial
        28800 ; Refresh
        14400 ; Retry
        3600000 ; Expire
        86400 ; Minimum
)
        IN      NS      localhost.
        IN      A       127.0.0.1

root@aysrSlack:~#
```

Now we can start the BIND service as follows

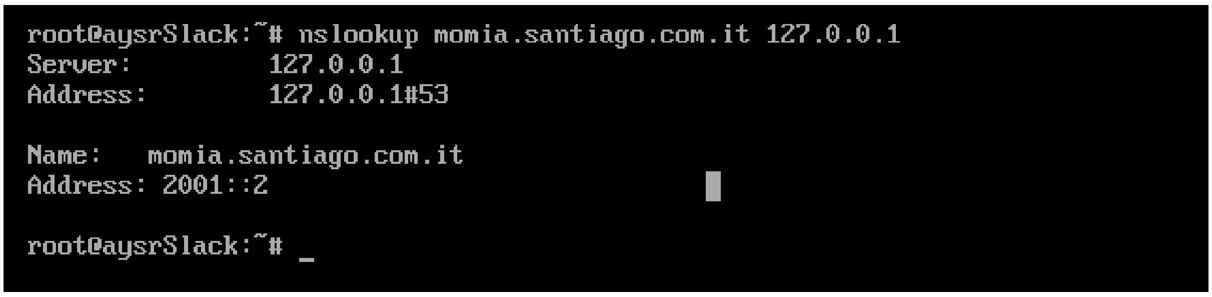


```
root@aysrSlack:~# /etc/rc.d/rc.bind start
```

And now we can prove the slave server, using **dig** or **nslookup**



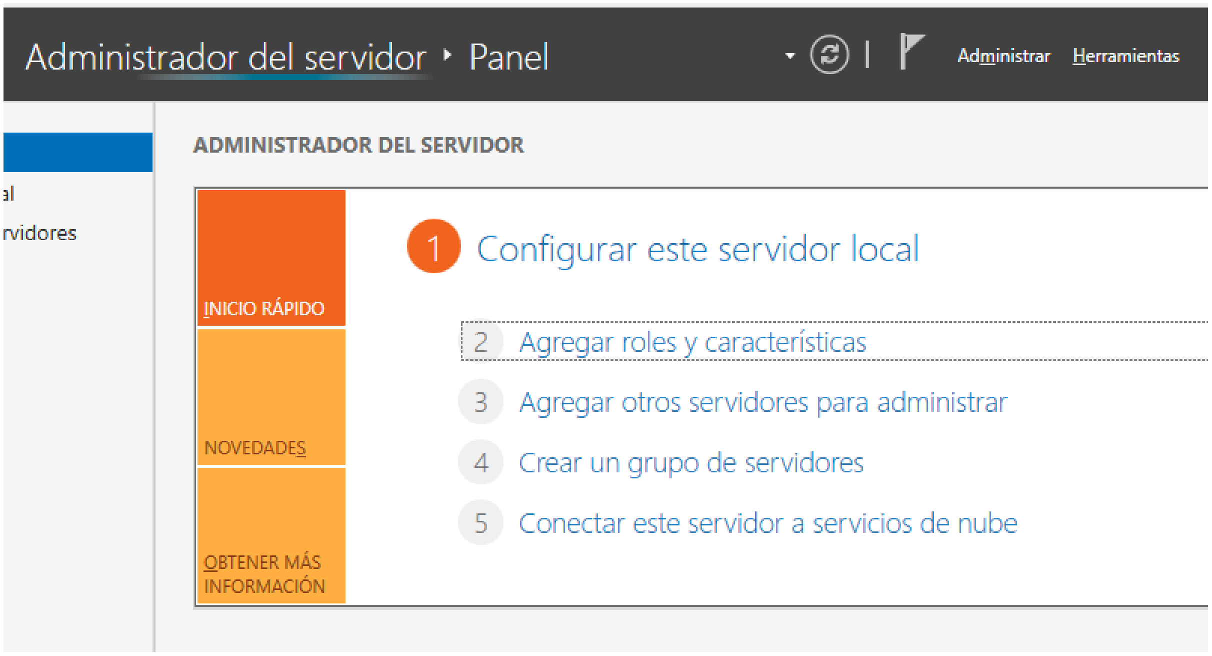
```
root@aysrSlack:~# dig @10.2.77.35 ostrich.santiago.com.it

; <<>> DiG 9.16.25 <<>> @10.2.77.35 ostrich.santiago.com.it
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25267
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;ostrich.santiago.com.it.       IN      A

;; ANSWER SECTION:
ostrich.santiago.com.it. 86400  IN      A       10.2.77.34

;; AUTHORITY SECTION:
santiago.com.it.        86400   IN      NS      bee.santiago.com.it.

;; ADDITIONAL SECTION:
bee.santiago.com.it.    86400   IN      A       10.2.77.33

;; Query time: 3 msec
;; SERVER: 10.2.77.35#53(10.2.77.35)
;; WHEN: Sat Sep 13 18:46:18 -05 2025
;; MSG SIZE  rcvd: 102
```



```
root@aysrSlack:~# nslookup momia.santiago.com.it 127.0.0.1
Server:         127.0.0.1
Address:        127.0.0.1#53

Name:   momia.santiago.com.it
Address: 2001::2

root@aysrSlack:~# _
```

Windows Server:

First we must enter the server manager. In the server administrator section, we choose the option "Add roles and features"


Administrador del servidor ‣ Panel

Administrar Herramientas

al

rvidores

ADMINISTRADOR DEL SERVIDOR

INICIO RÁPIDO

NOVEDADES

OBTENER MÁS INFORMACIÓN

1 Configurar este servidor local

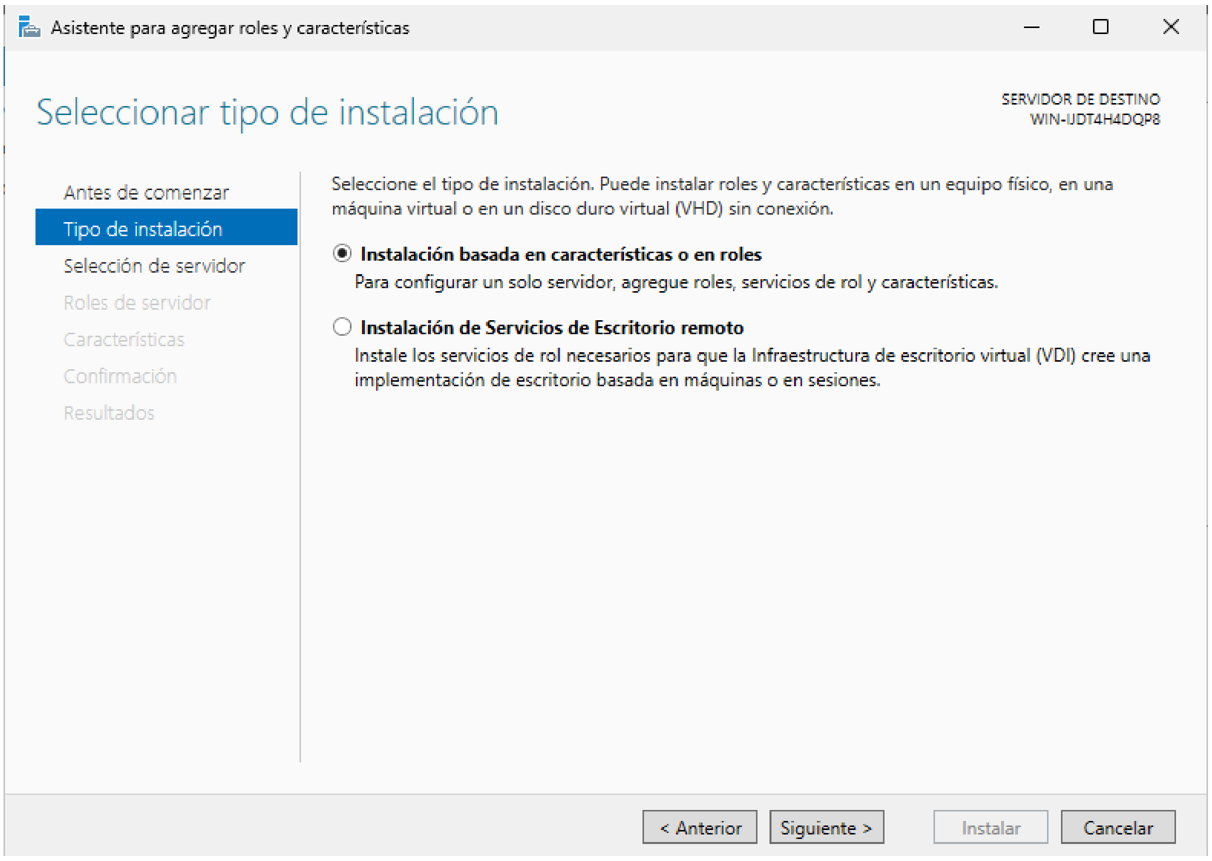
2 Agregar roles y características

3 Agregar otros servidores para administrar

4 Crear un grupo de servidores

5 Conectar este servidor a servicios de nube

For the installation type, choose the "Feature or role-based installation" option.


Asistente para agregar roles y características

Seleccionar tipo de instalación

SERVIDOR DE DESTINO
WIN-IJDT4H4DQP8

Antes de comenzar
Tipo de instalación
Selección de servidor
Roles de servidor
Características
Confirmación
Resultados

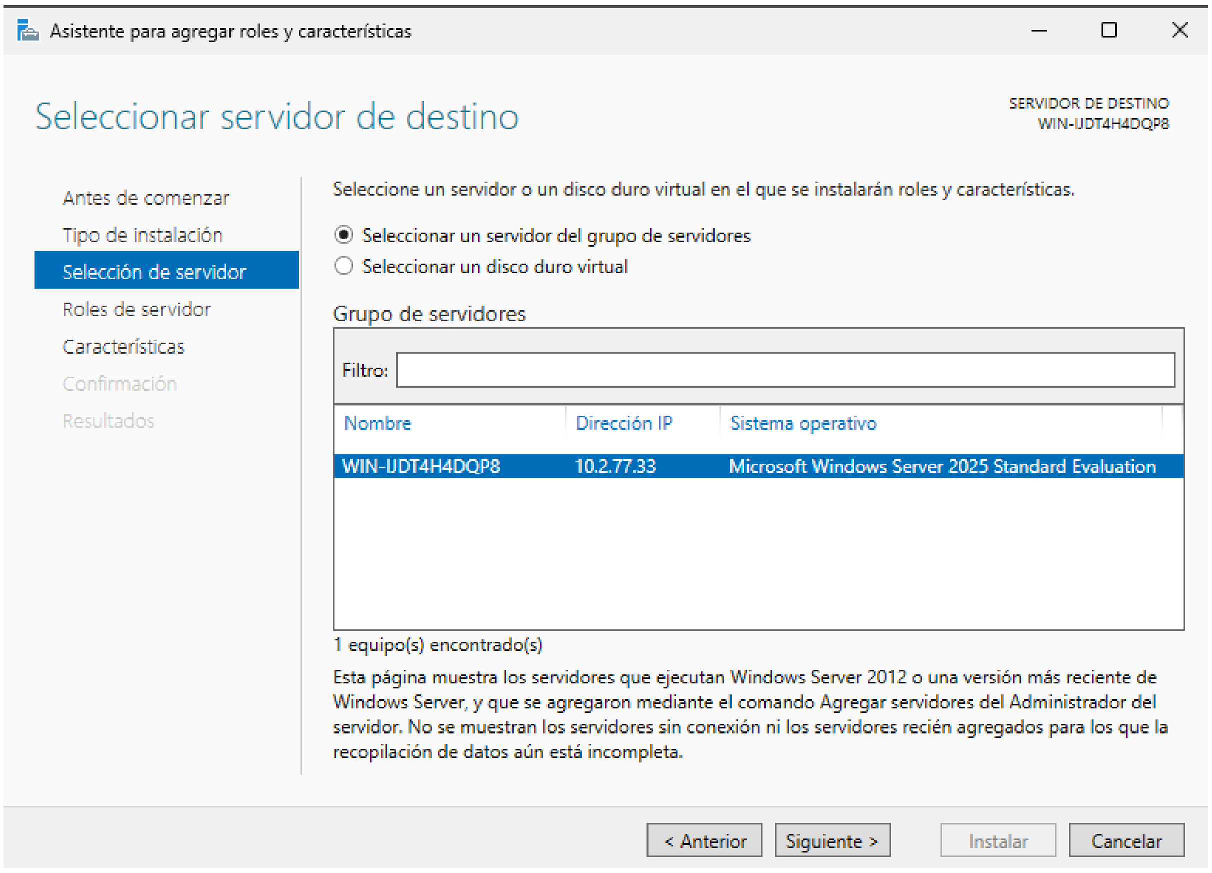
Seleccione el tipo de instalación. Puede instalar roles y características en un equipo físico, en una máquina virtual o en un disco duro virtual (VHD) sin conexión.

● **Instalación basada en características o en roles**
Para configurar un solo servidor, agregue roles, servicios de rol y características.

○ **Instalación de Servicios de Escritorio remoto**
Instale los servicios de rol necesarios para que la Infraestructura de escritorio virtual (VDI) cree una implementación de escritorio basada en máquinas o en sesiones.

< Anterior | Siguiente > | Instalar | Cancelar

In server selection we choose the Windows Server virtual machine.


Asistente para agregar roles y características

Seleccionar servidor de destino

SERVIDOR DE DESTINO
WIN-IJDT4H4DQP8

- Antes de comenzar
- Tipo de instalación
- **Selección de servidor**
- Roles de servidor
- Características
- Confirmación
- Resultados

Seleccione un servidor o un disco duro virtual en el que se instalarán roles y características.

● Seleccionar un servidor del grupo de servidores
○ Seleccionar un disco duro virtual

Grupo de servidores

Filtro:

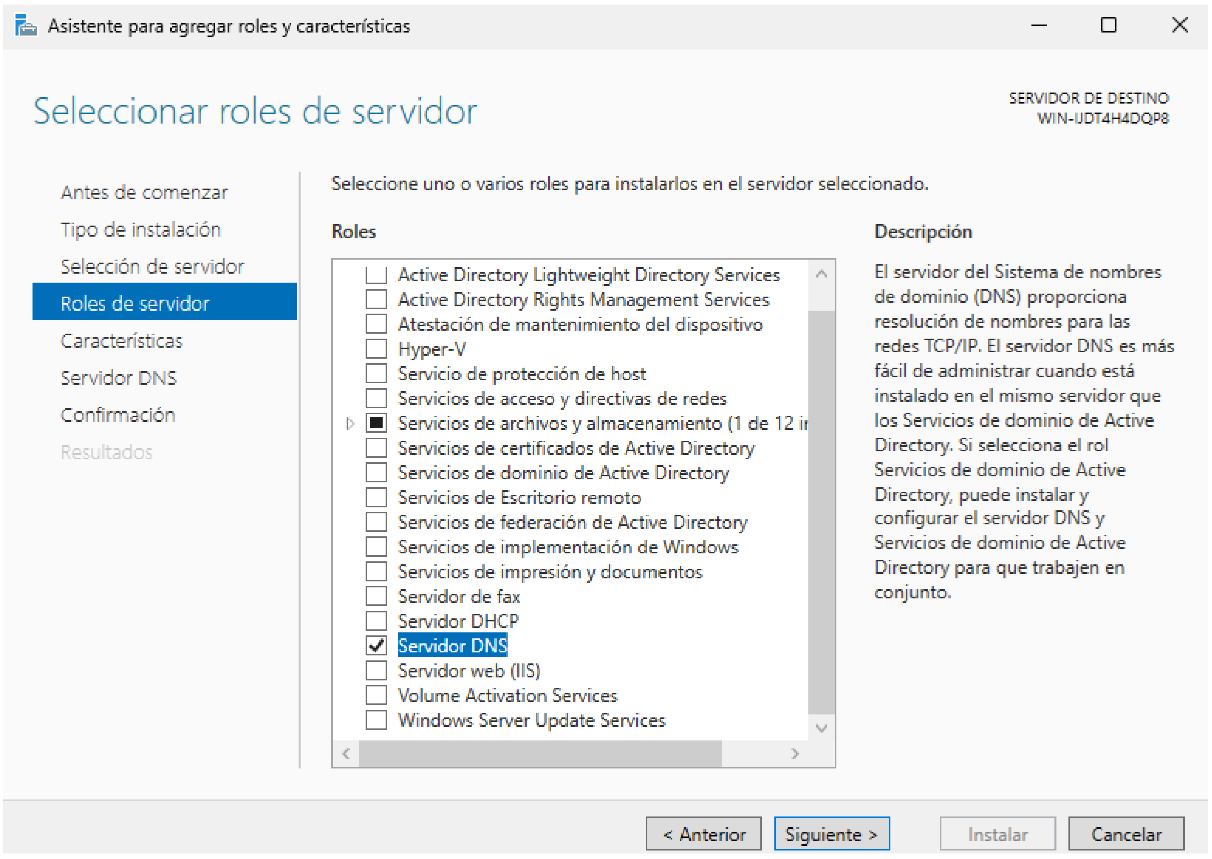
Nombre	Dirección IP	Sistema operativo
WIN-IJDT4H4DQP8	10.2.77.33	Microsoft Windows Server 2025 Standard Evaluation

1 equipo(s) encontrado(s)

Esta página muestra los servidores que ejecutan Windows Server 2012 o una versión más reciente de Windows Server, y que se agregaron mediante el comando Agregar servidores del Administrador del servidor. No se muestran los servidores sin conexión ni los servidores recién agregados para los que la recopilación de datos aún está incompleta.

< Anterior | Siguiente > | Instalar | Cancelar

In server roles we mark the one that says "DNS Server"


Asistente para agregar roles y características

Seleccionar roles de servidor

SERVIDOR DE DESTINO
WIN-IJDT4H4DQP8

- Antes de comenzar
- Tipo de instalación
- Selección de servidor
- **Roles de servidor**
- Características
- Servidor DNS
- Confirmación
- Resultados

Seleccione uno o varios roles para instalarlos en el servidor seleccionado.

Roles

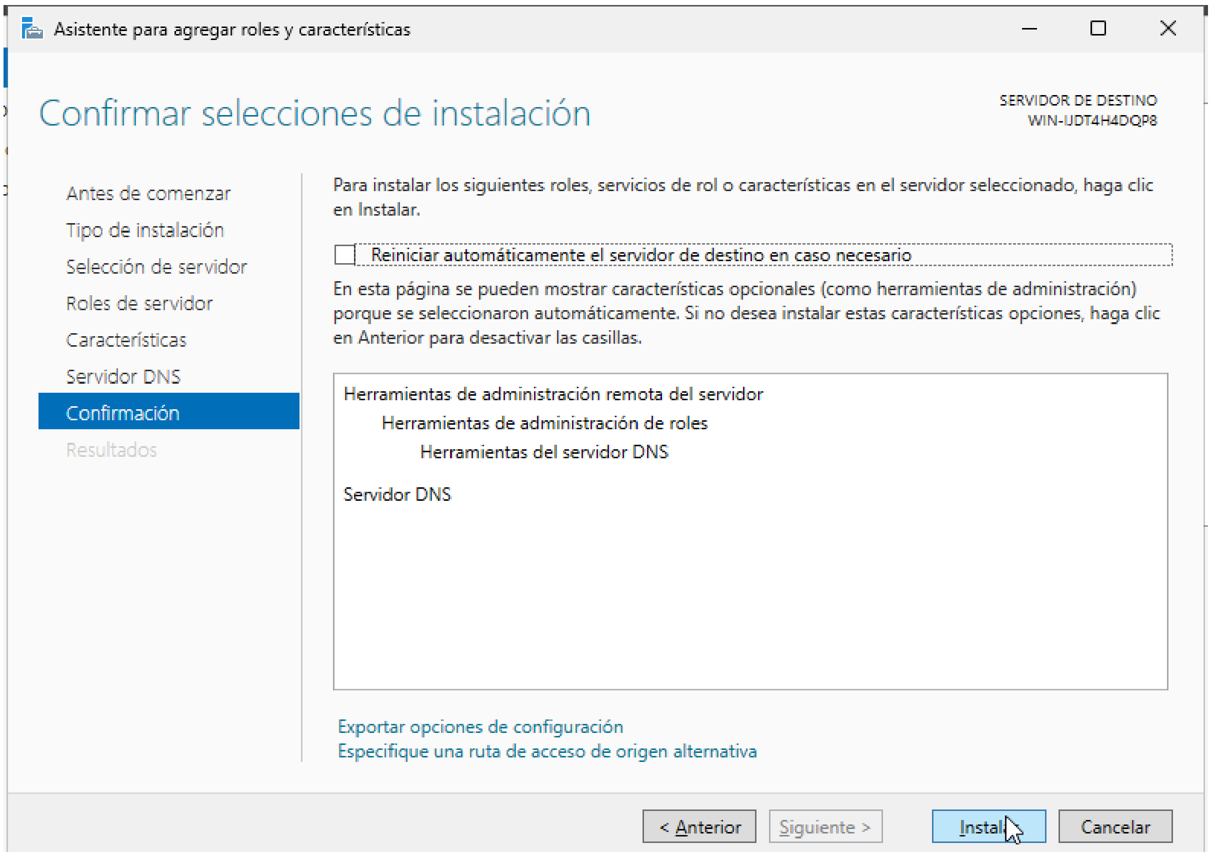
- [ ] Active Directory Lightweight Directory Services
- [ ] Active Directory Rights Management Services
- [ ] Atestación de mantenimiento del dispositivo
- [ ] Hyper-V
- [ ] Servicio de protección de host
- [ ] Servicios de acceso y directivas de redes
- [■] Servicios de archivos y almacenamiento (1 de 12 i
- [ ] Servicios de certificados de Active Directory
- [ ] Servicios de dominio de Active Directory
- [ ] Servicios de Escritorio remoto
- [ ] Servicios de federación de Active Directory
- [ ] Servicios de implementación de Windows
- [ ] Servicios de impresión y documentos
- [ ] Servidor de fax
- [ ] Servidor DHCP
- [✓] Servidor DNS
- [ ] Servidor web (IIS)
- [ ] Volume Activation Services
- [ ] Windows Server Update Services

Descripción

El servidor del Sistema de nombres de dominio (DNS) proporciona resolución de nombres para las redes TCP/IP. El servidor DNS es más fácil de administrar cuando está instalado en el mismo servidor que los Servicios de dominio de Active Directory. Si selecciona el rol Servicios de dominio de Active Directory, puede instalar y configurar el servidor DNS y Servicios de dominio de Active Directory para que trabajen en conjunto.

< Anterior | Siguiente > | Instalar | Cancelar

We verify that the configurations have been applied and click on install.



Confirmar selecciones de instalación

Para instalar los siguientes roles, servicios de rol o características en el servidor seleccionado, haga clic en Instalar.

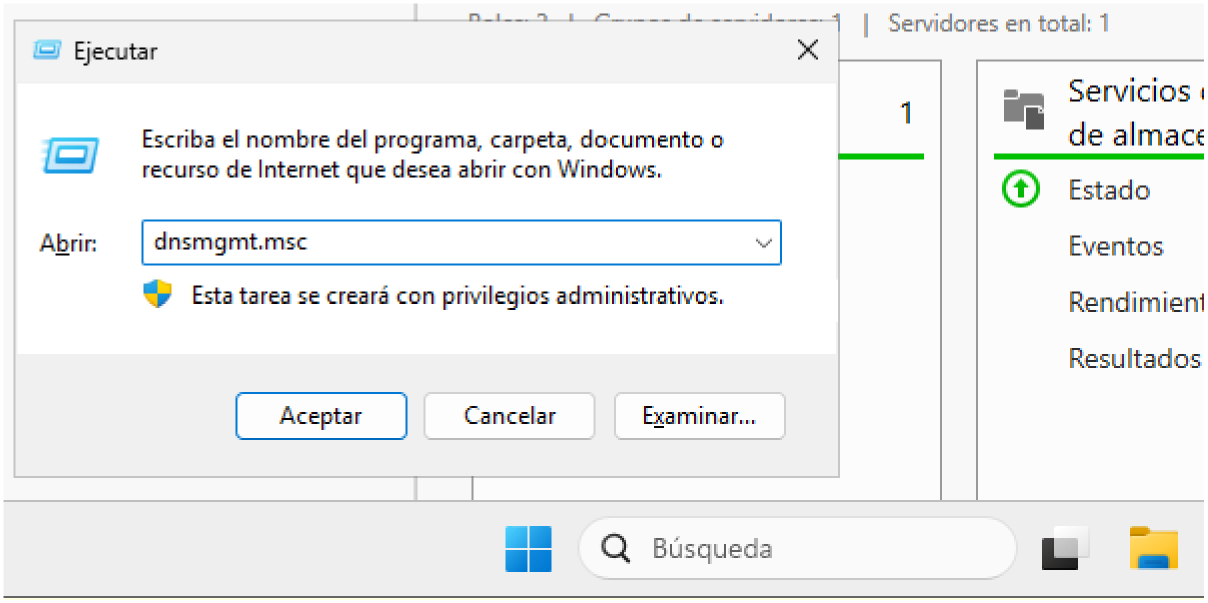
☐ Reiniciar automáticamente el servidor de destino en caso necesario

En esta página se pueden mostrar características opcionales (como herramientas de administración) porque se seleccionaron automáticamente. Si no desea instalar estas características opciones, haga clic en Anterior para desactivar las casillas.

Herramientas de administración remota del servidor
    Herramientas de administración de roles
        Herramientas del servidor DNS

Servidor DNS

We wait for it to install, then press Windows + R, and type **dnsmgmt.msc**, press enter

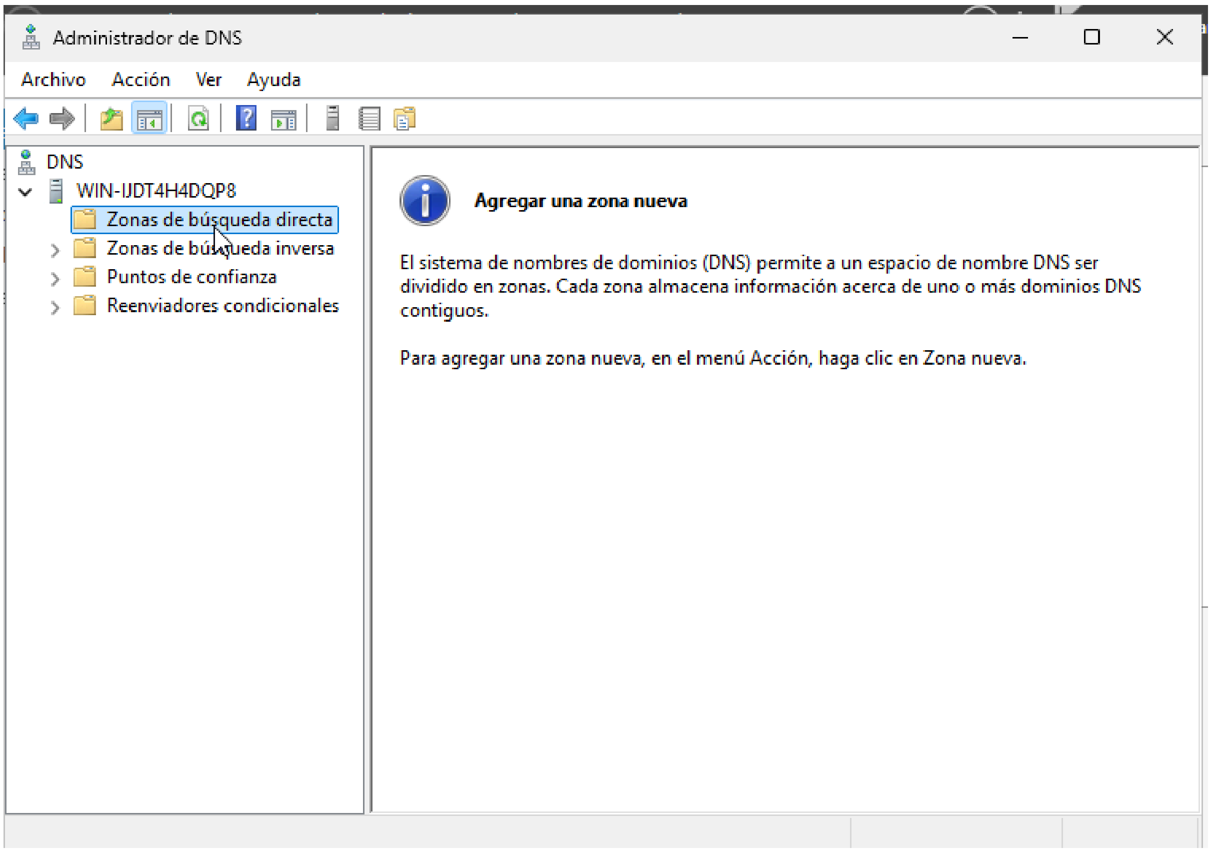


Escriba el nombre del programa, carpeta, documento o recurso de Internet que desea abrir con Windows.

Abrir: dnsmgmt.msc

Esta tarea se creará con privilegios administrativos.

Then, in the tab that appears, right-click on "direct search areas"


Administrador de DNS

Archivo Acción Ver Ayuda

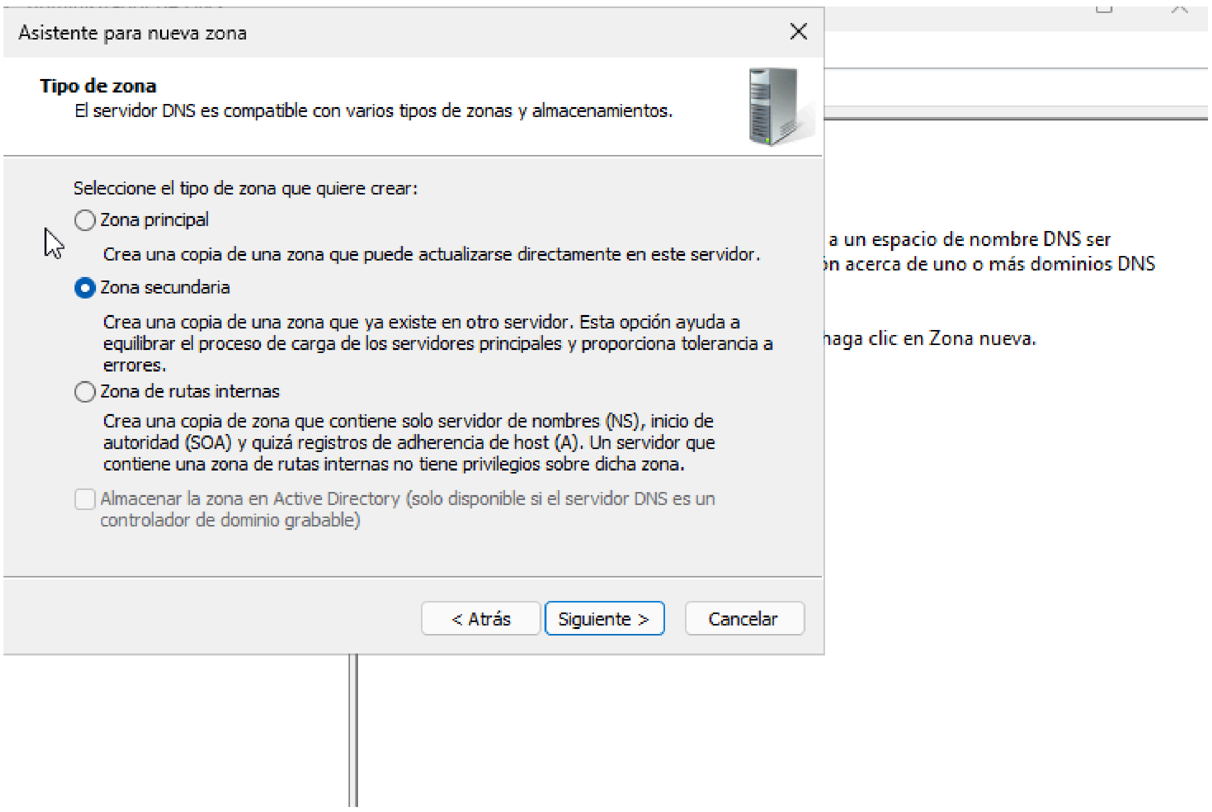
DNS
WIN-IJDT4H4DQP8
Zonas de búsqueda directa
Zonas de búsqueda inversa
Puntos de confianza
Reenviadores condicionales

Agregar una zona nueva

El sistema de nombres de dominios (DNS) permite a un espacio de nombre DNS ser dividido en zonas. Cada zona almacena información acerca de uno o más dominios DNS contiguos.

Para agregar una zona nueva, en el menú Acción, haga clic en Zona nueva.

We choose the type of zone, in this case secondary since this domain will use Windows Server to host its slave server.


Asistente para nueva zona

Tipo de zona
El servidor DNS es compatible con varios tipos de zonas y almacenamientos.

Seleccione el tipo de zona que quiere crear:

( ) Zona principal
Crea una copia de una zona que puede actualizarse directamente en este servidor.

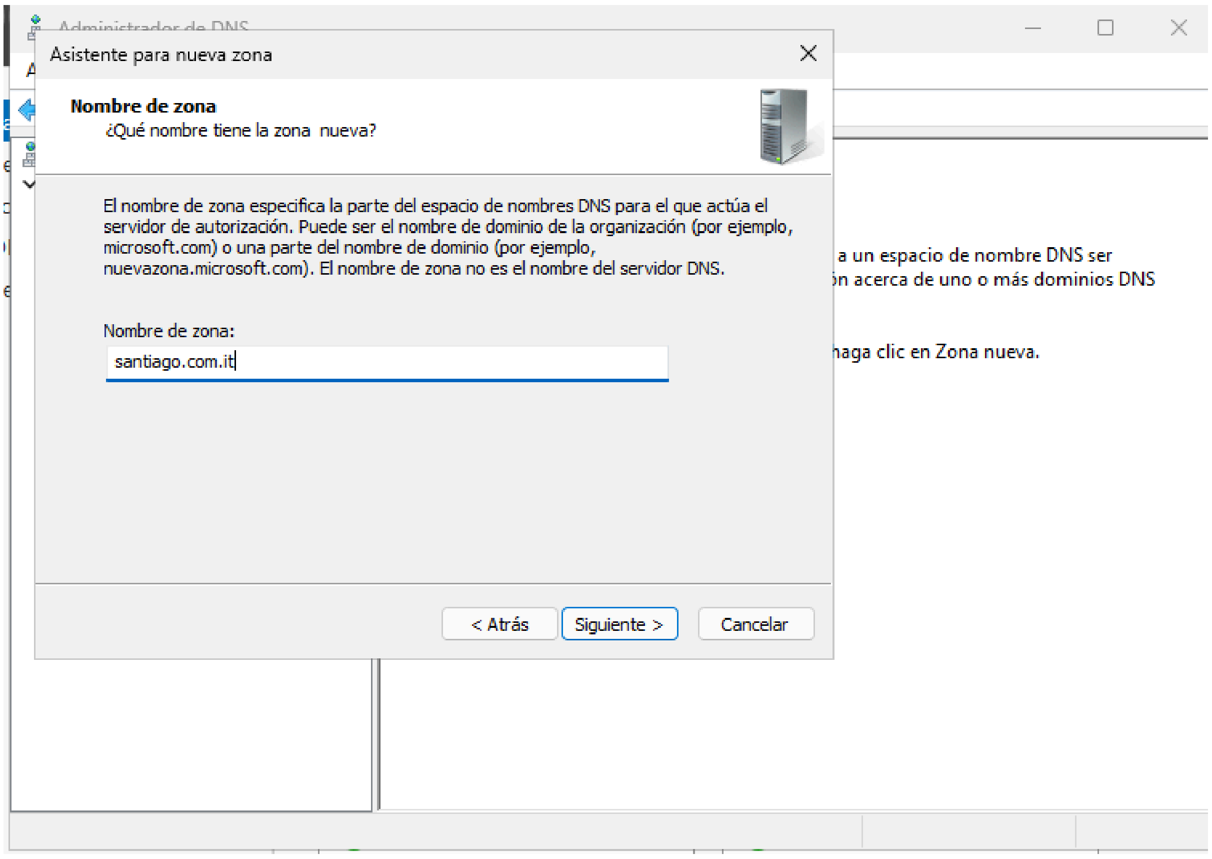
(●) Zona secundaria
Crea una copia de una zona que ya existe en otro servidor. Esta opción ayuda a equilibrar el proceso de carga de los servidores principales y proporciona tolerancia a errores.

( ) Zona de rutas internas
Crea una copia de zona que contiene solo servidor de nombres (NS), inicio de autoridad (SOA) y quizá registros de adherencia de host (A). Un servidor que contiene una zona de rutas internas no tiene privilegios sobre dicha zona.

[ ] Almacenar la zona en Active Directory (solo disponible si el servidor DNS es un controlador de dominio grabable)

< Atrás | Siguiente > | Cancelar

We write the domain name.



Asistente para nueva zona

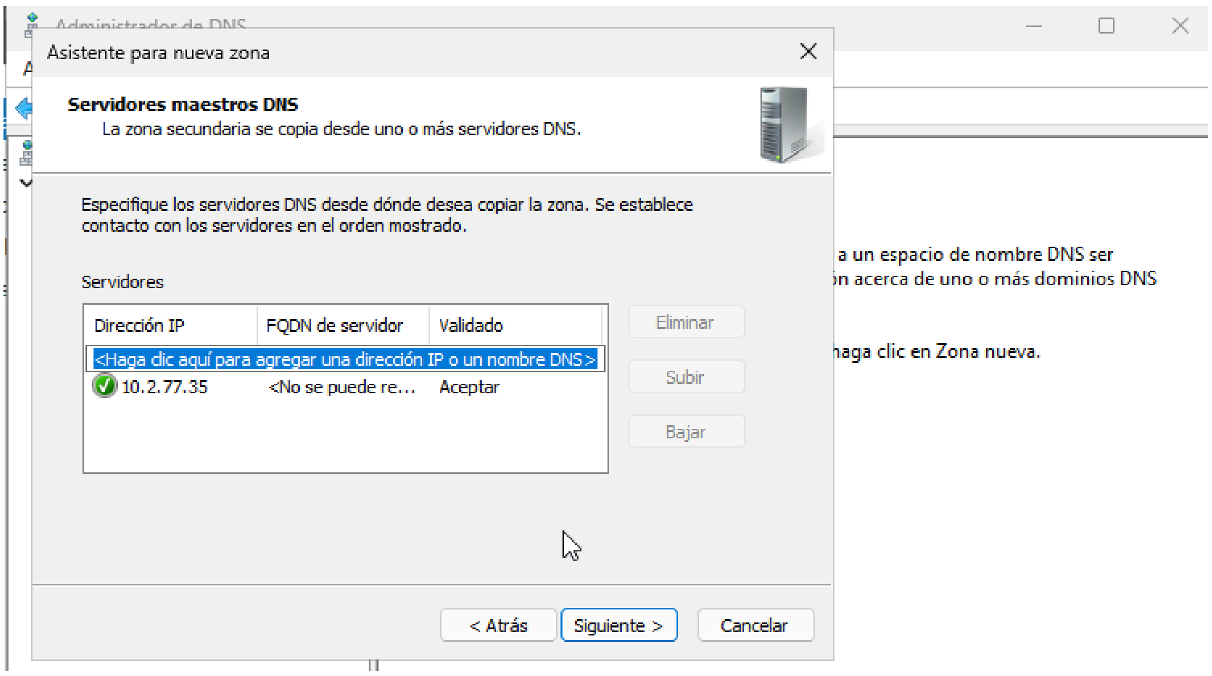
Nombre de zona
¿Qué nombre tiene la zona nueva?

El nombre de zona especifica la parte del espacio de nombres DNS para el que actúa el servidor de autorización. Puede ser el nombre de dominio de la organización (por ejemplo, microsoft.com) o una parte del nombre de dominio (por ejemplo, nuevazona.microsoft.com). El nombre de zona no es el nombre del servidor DNS.

Nombre de zona:
santiago.com.it

< Atrás | Siguiente > | Cancelar

And then, we write the IP of the virtual machine where the main or master server resides, in this case, the Solaris IP



Asistente para nueva zona

Servidores maestros DNS
La zona secundaria se copia desde uno o más servidores DNS.

Especifique los servidores DNS desde dónde desea copiar la zona. Se establece contacto con los servidores en el orden mostrado.

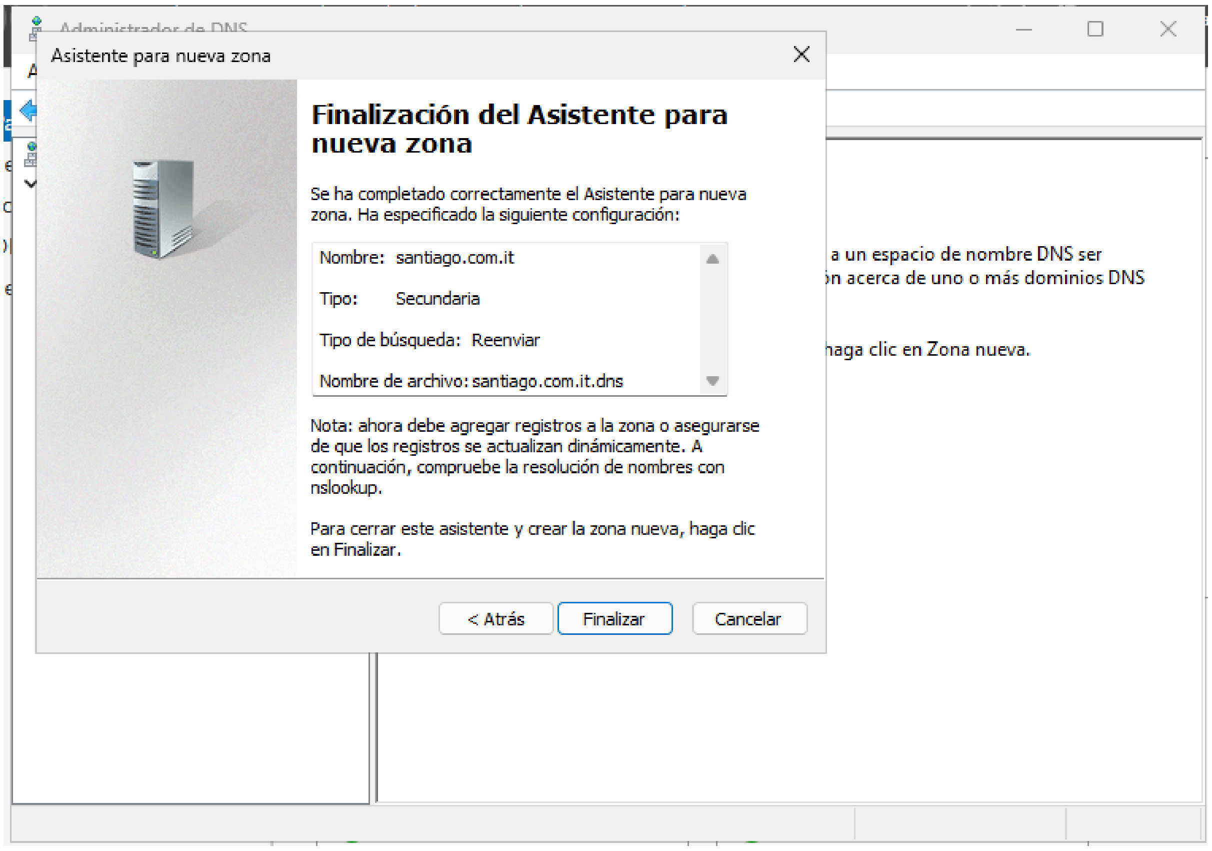
Servidores

Dirección IP	FQDN de servidor	Validado
<Haga clic aquí para agregar una dirección IP o un nombre DNS>		
✓ 10.2.77.35	<No se puede re...	Aceptar

Eliminar | Subir | Bajar

< Atrás | Siguiente > | Cancelar

We verify that the data is correct and click finish.



Finalización del Asistente para nueva zona

Se ha completado correctamente el Asistente para nueva zona. Ha especificado la siguiente configuración:

Nombre: santiago.com.it

Tipo: Secundaria

Tipo de búsqueda: Reenviar

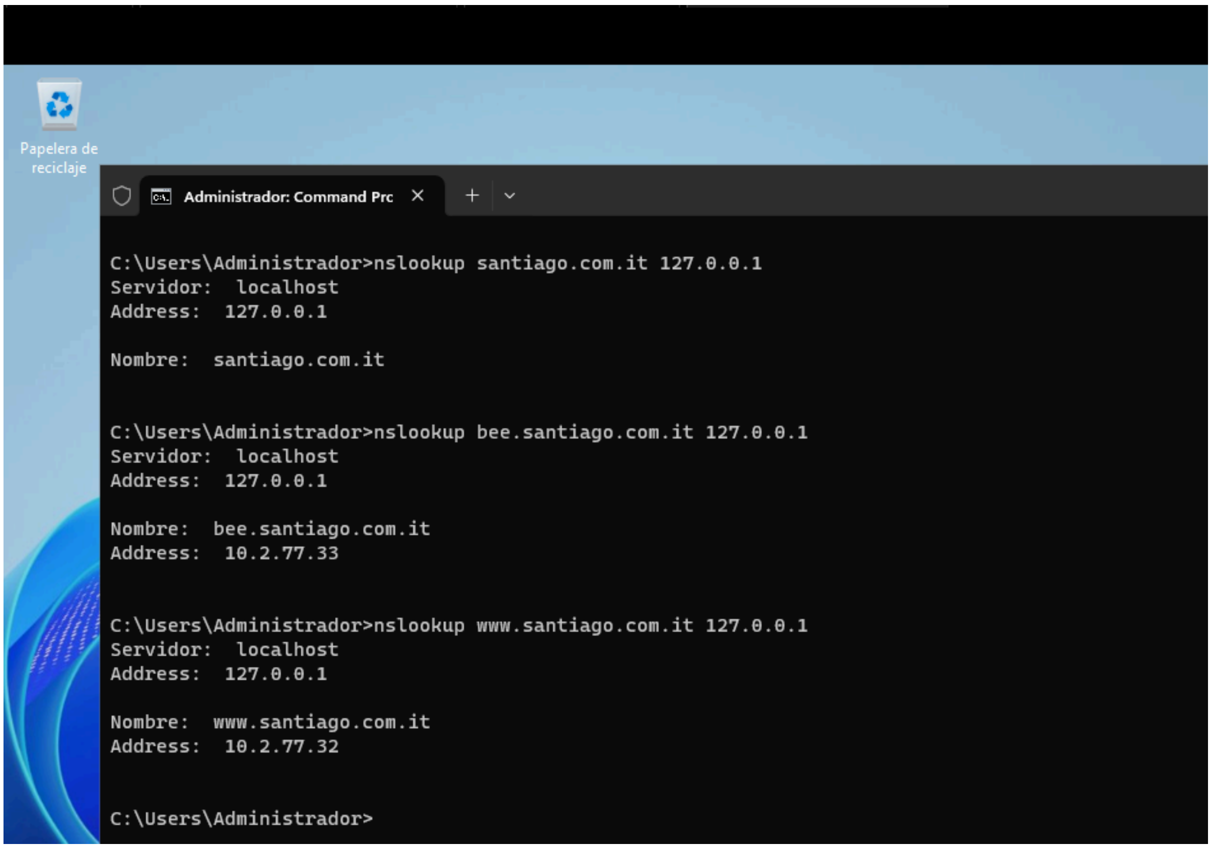
Nombre de archivo: santiago.com.it.dns

Nota: ahora debe agregar registros a la zona o asegurarse de que los registros se actualizan dinámicamente. A continuación, compruebe la resolución de nombres con nslookup.

Para cerrar este asistente y crear la zona nueva, haga clic en Finalizar.

< Atrás
Finalizar
Cancelar
a un espacio de nombre DNS ser
ón acerca de uno o más dominios DNS
haga clic en Zona nueva.

After that, we use **nslookup** to test that the slave server is running.



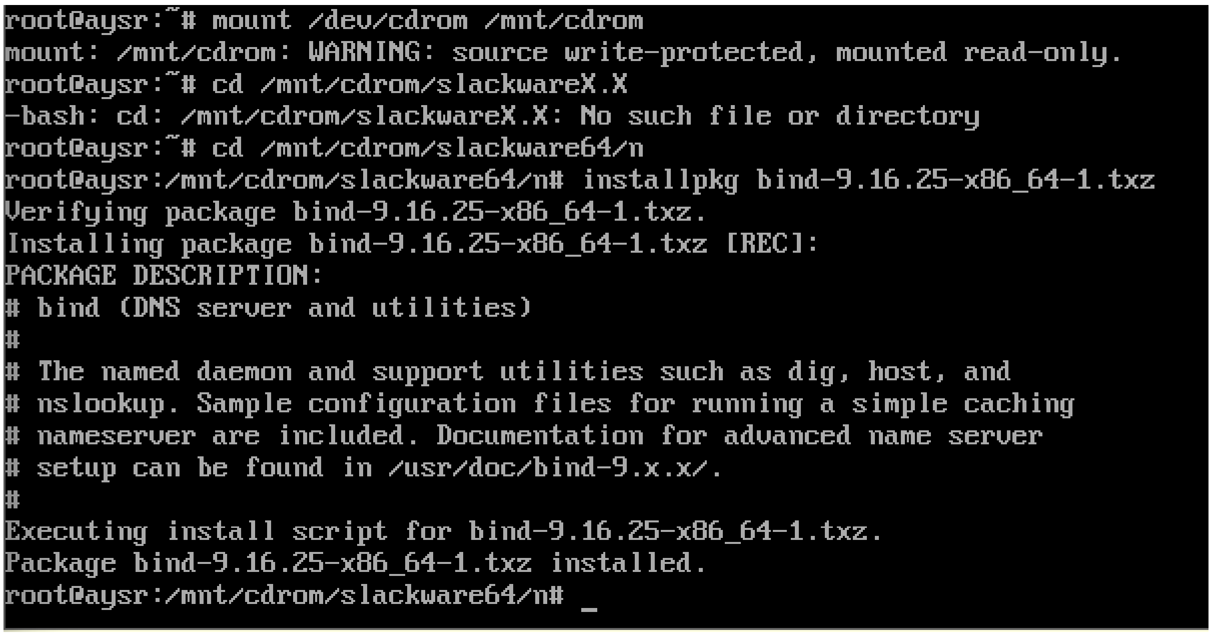
```
C:\Users\Administrador>nslookup santiago.com.it 127.0.0.1
Servidor:  localhost
Address:  127.0.0.1

Nombre:  santiago.com.it


C:\Users\Administrador>nslookup bee.santiago.com.it 127.0.0.1
Servidor:  localhost
Address:  127.0.0.1

Nombre:  bee.santiago.com.it
Address:  10.2.77.33


C:\Users\Administrador>nslookup www.santiago.com.it 127.0.0.1
Servidor:  localhost
Address:  127.0.0.1

Nombre:  www.santiago.com.it
Address:  10.2.77.32


C:\Users\Administrador>
```

- For the domain student 2.org.uk:
- Primary DNS server on a Slackware virtual machine.

Below is an example of how to configure the primary DNS service on Slackware. The highlighted yellow parts indicate what should be added to the configuration files or replaced with the names of your domains or specific IP addresses:

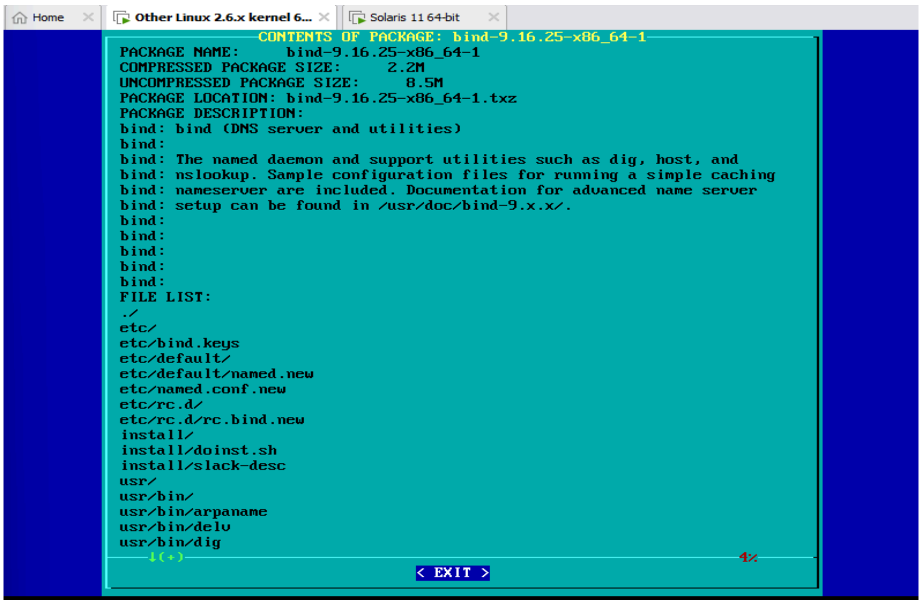
For julian.org.uk domain:

1. If required, install the DNS package from the Linux CD/Image.



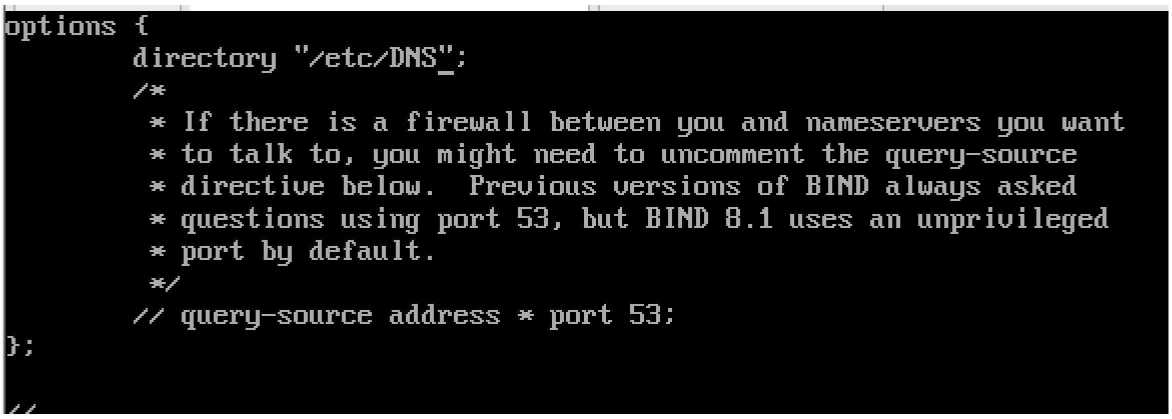
```
root@aysr:~# mount /dev/cdrom /mnt/cdrom
mount: /mnt/cdrom: WARNING: source write-protected, mounted read-only.
root@aysr:~# cd /mnt/cdrom/slackwareX.X
-bash: cd: /mnt/cdrom/slackwareX.X: No such file or directory
root@aysr:~# cd /mnt/cdrom/slackware64/n
root@aysr:/mnt/cdrom/slackware64/n# installpkg bind-9.16.25-x86_64-1.txz
Verifying package bind-9.16.25-x86_64-1.txz.
Installing package bind-9.16.25-x86_64-1.txz [REC]:
PACKAGE DESCRIPTION:
# bind (DNS server and utilities)
#
# The named daemon and support utilities such as dig, host, and
# nslookup. Sample configuration files for running a simple caching
# nameserver are included. Documentation for advanced name server
# setup can be found in /usr/doc/bind-9.x.x/.
#
Executing install script for bind-9.16.25-x86_64-1.txz.
Package bind-9.16.25-x86_64-1.txz installed.
root@aysr:/mnt/cdrom/slackware64/n# _
```

2. Check that the packages were installed (for example, on Slackware, use pkgtools to verify).

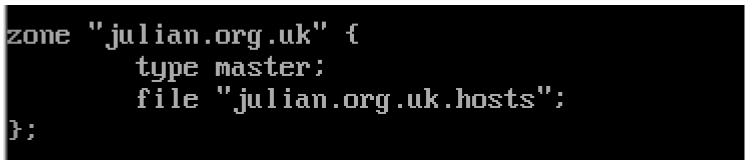


```
CONTENTS OF PACKAGE: bind-9.16.25-x86_64-1
PACKAGE NAME:     bind-9.16.25-x86_64-1
COMPRESSED PACKAGE SIZE:     2.2M
UNCOMPRESSED PACKAGE SIZE:     8.5M
PACKAGE LOCATION: bind-9.16.25-x86_64-1.txz
PACKAGE DESCRIPTION:
bind: bind (DNS server and utilities)
bind:
bind: The named daemon and support utilities such as dig, host, and
bind: nslookup. Sample configuration files for running a simple caching
bind: nameserver are included. Documentation for advanced name server
bind: setup can be found in /usr/doc/bind-9.x.x/.
bind:
bind:
bind:
bind:
bind:
FILE LIST:
./
etc/
etc/bind.keys
etc/default/
etc/default/named.new
etc/named.conf.new
etc/rc.d/
etc/rc.d/rc.bind.new
install/
install/doinst.sh
install/slack-desc
usr/
usr/bin/
usr/bin/arpaname
usr/bin/delv
usr/bin/dig
```

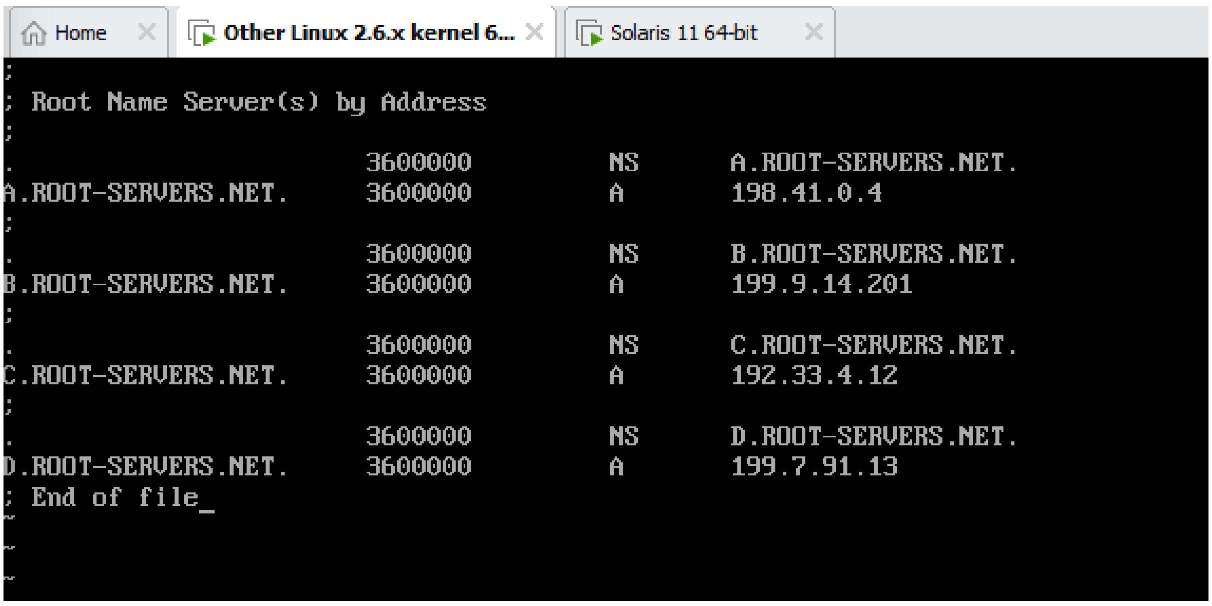
3. Configure the service.



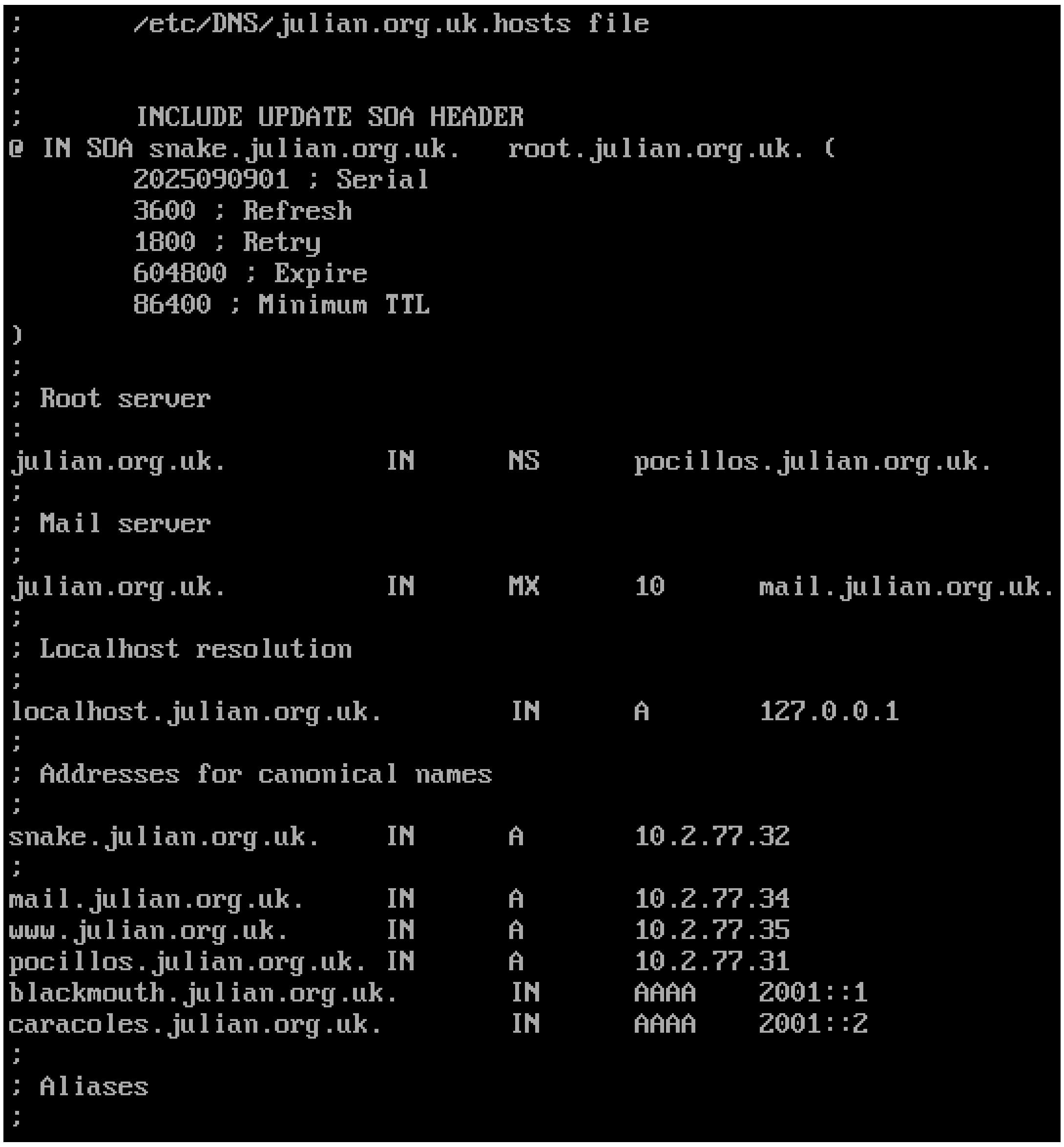
```
options {
        directory "/etc/DNS_";
        /*
         * If there is a firewall between you and nameservers you want
         * to talk to, you might need to uncomment the query-source
         * directive below.  Previous versions of BIND always asked
         * questions using port 53, but BIND 8.1 uses an unprivileged
         * port by default.
         */
        // query-source address * port 53;
};

//
```



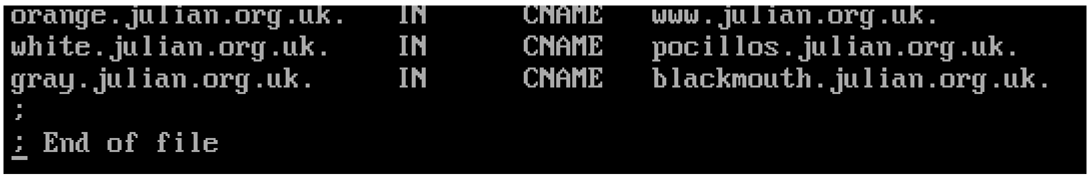
```
zone "julian.org.uk" {
        type master;
        file "julian.org.uk.hosts";
};
```



```
;
; Root Name Server(s) by Address
;
.                       3600000         NS      A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.     3600000         A       198.41.0.4
;
.                       3600000         NS      B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.     3600000         A       199.9.14.201
;
.                       3600000         NS      C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.     3600000         A       192.33.4.12
;
.                       3600000         NS      D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.     3600000         A       199.7.91.13
; End of file_
~
~
~
```



```
;       /etc/DNS/julian.org.uk.hosts file
;
;
;       INCLUDE UPDATE SOA HEADER
@ IN SOA snake.julian.org.uk.   root.julian.org.uk. (
        2025090901 ; Serial
        3600 ; Refresh
        1800 ; Retry
        604800 ; Expire
        86400 ; Minimum TTL
)
;
; Root server
;
julian.org.uk.          IN      NS      pocillos.julian.org.uk.
;
; Mail server
;
julian.org.uk.          IN      MX      10      mail.julian.org.uk.
;
; Localhost resolution
;
localhost.julian.org.uk.        IN      A       127.0.0.1
;
; Addresses for canonical names
;
snake.julian.org.uk.    IN      A       10.2.77.32
;
mail.julian.org.uk.     IN      A       10.2.77.34
www.julian.org.uk.      IN      A       10.2.77.35
pocillos.julian.org.uk. IN      A       10.2.77.31
blackmouth.julian.org.uk.       IN      AAAA    2001::1
caracoles.julian.org.uk.        IN      AAAA    2001::2
;
; Aliases
;
```



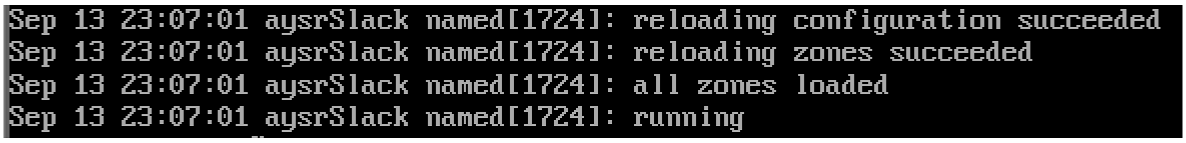
```
orange.julian.org.uk.   IN      CNAME   www.julian.org.uk.
white.julian.org.uk.    IN      CNAME   pocillos.julian.org.uk.
gray.julian.org.uk.     IN      CNAME   blackmouth.julian.org.uk.
;
; End of file
```

- Secondary DNS servers on a Solaris virtual machine and Windows Server.

4. What are the A and AAAA records in the root servers file?
The A and AAAA records in the root servers file define the IPv4 and IPv6 addresses of the servers responsible for handling DNS queries. The A records map hostnames to IPv4 addresses, while the AAAA records map hostnames to IPv6 addresses. These records allow DNS resolvers to locate and communicate with the appropriate servers using either IP protocol.

5. What are the NS, MX, A, and CNAME records in the particular domain file?
The NS record specifies the authoritative name server for the domain. The MX record indicates the mail server responsible for handling email for the domain, along with its priority. A records map hostnames to their corresponding IPv4 addresses, while CNAME records create aliases that point one hostname to another canonical hostname. These records collectively define how various services within the domain are accessed and resolved.

6. Check the system logs to verify that the service is functioning correctly.
At this moment, we had to reload BIND due to a modification, but the service, but the service still works.

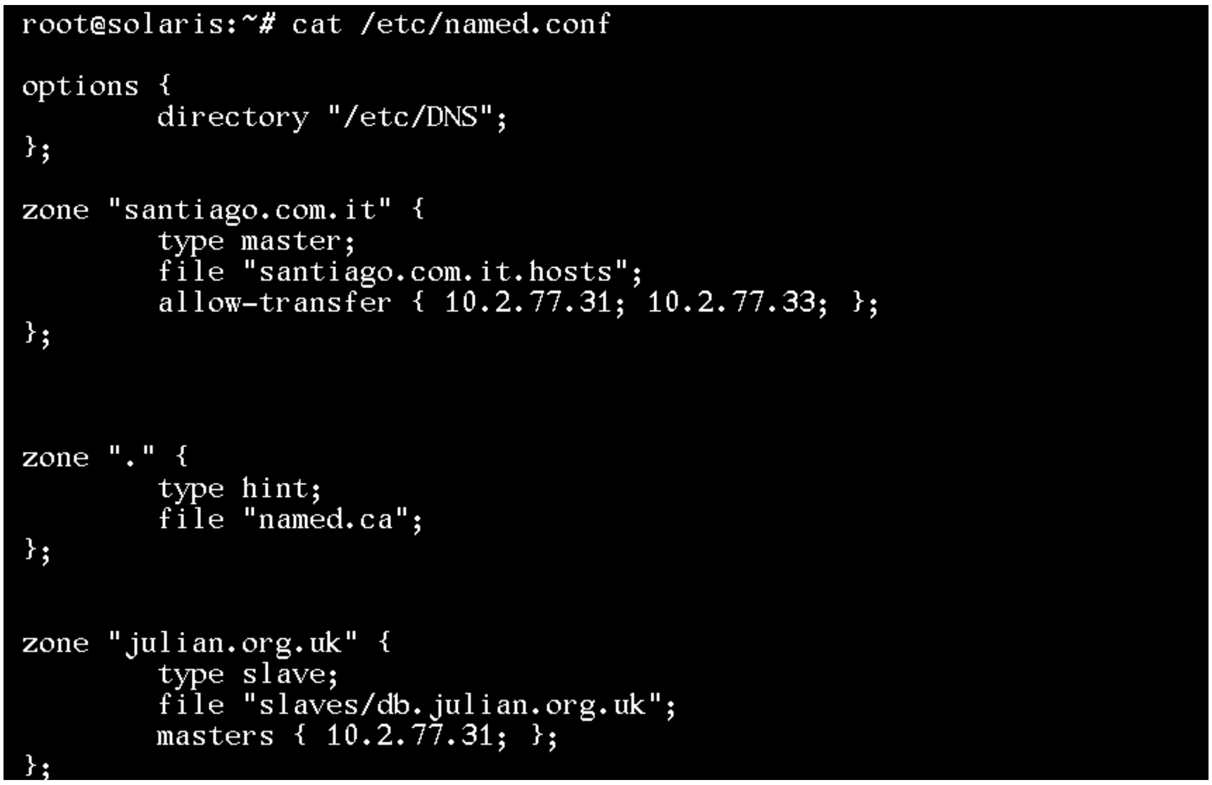


```
Sep 13 23:07:01 aysrSlack named[1724]: reloading configuration succeeded
Sep 13 23:07:01 aysrSlack named[1724]: reloading zones succeeded
Sep 13 23:07:01 aysrSlack named[1724]: all zones loaded
Sep 13 23:07:01 aysrSlack named[1724]: running
```

Now, let's configure the slave DNS servers of the second domain on Windows server and Solaris

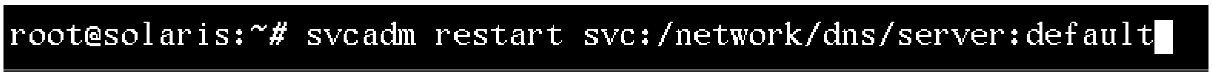
On Oracle Solaris:

First, we added the zone for julian.org.uk domain



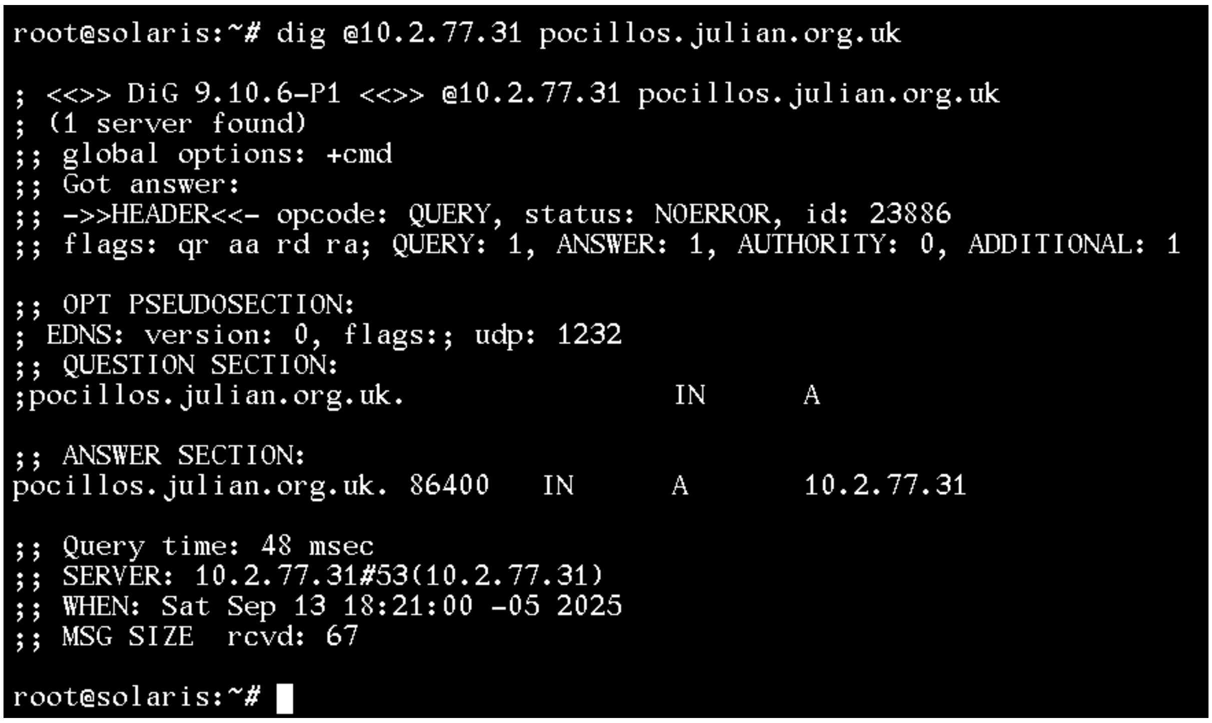
```
root@solaris:~# cat /etc/named.conf

options {
        directory "/etc/DNS";
};

zone "santiago.com.it" {
        type master;
        file "santiago.com.it.hosts";
        allow-transfer { 10.2.77.31; 10.2.77.33; };
};



zone "." {
        type hint;
        file "named.ca";
};


zone "julian.org.uk" {
        type slave;
        file "slaves/db.julian.org.uk";
        masters { 10.2.77.31; };
};
```

We add the slaves folder where the copy from the primary server will be saved, next, we need to restart the DNS server to apply changes



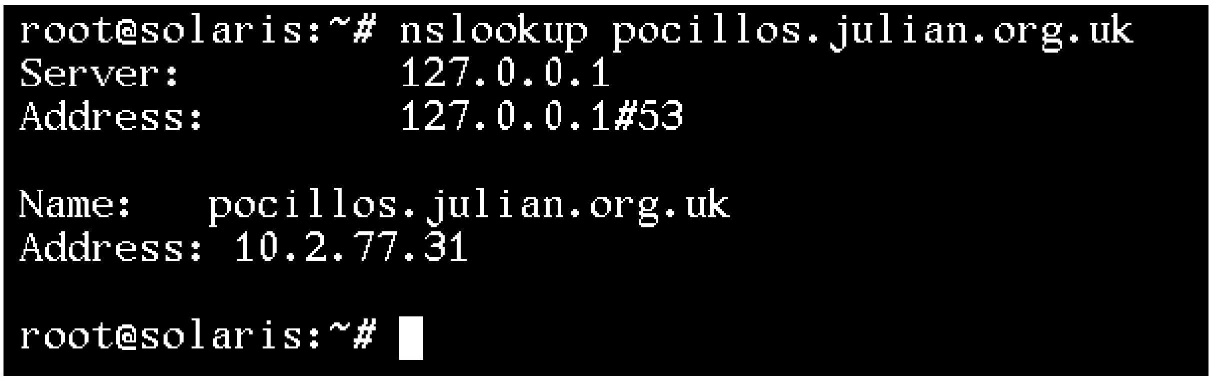
```
root@solaris:~# svcadm restart svc:/network/dns/server:default
```

Now, we can check its funcionality
Connection to the primary server using **dig**:



```
root@solaris:~# dig @10.2.77.31 pocillos.julian.org.uk

; <<>> DiG 9.10.6-P1 <<>> @10.2.77.31 pocillos.julian.org.uk
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 23886
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;pocillos.julian.org.uk.                IN      A

;; ANSWER SECTION:
pocillos.julian.org.uk. 86400   IN      A       10.2.77.31

;; Query time: 48 msec
;; SERVER: 10.2.77.31#53(10.2.77.31)
;; WHEN: Sat Sep 13 18:21:00 -05 2025
;; MSG SIZE  rcvd: 67

root@solaris:~#
```

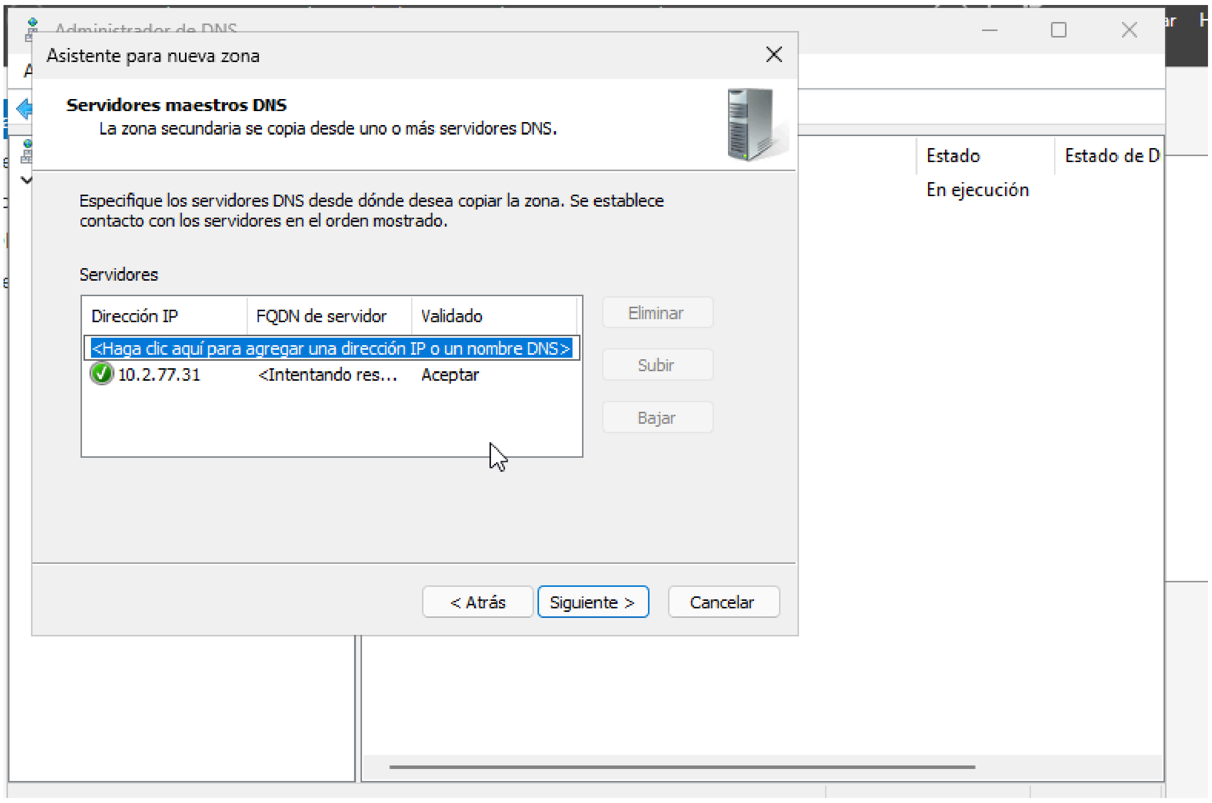
Connection to the slave server using **nslookup**



```
root@solaris:~# nslookup pocillos.julian.org.uk
Server:         127.0.0.1
Address:        127.0.0.1#53

Name:   pocillos.julian.org.uk
Address: 10.2.77.31

root@solaris:~#
```

On Windows Server:

After installing the DNS server we follow the same steps described in the part where we configure the slave DNS server for santiago.com.it
We just put julian.org.uk as the name and write the IP of the machine where the primary DNS server is, in this case, the IP of Linux Slackware


Asistente para nueva zona

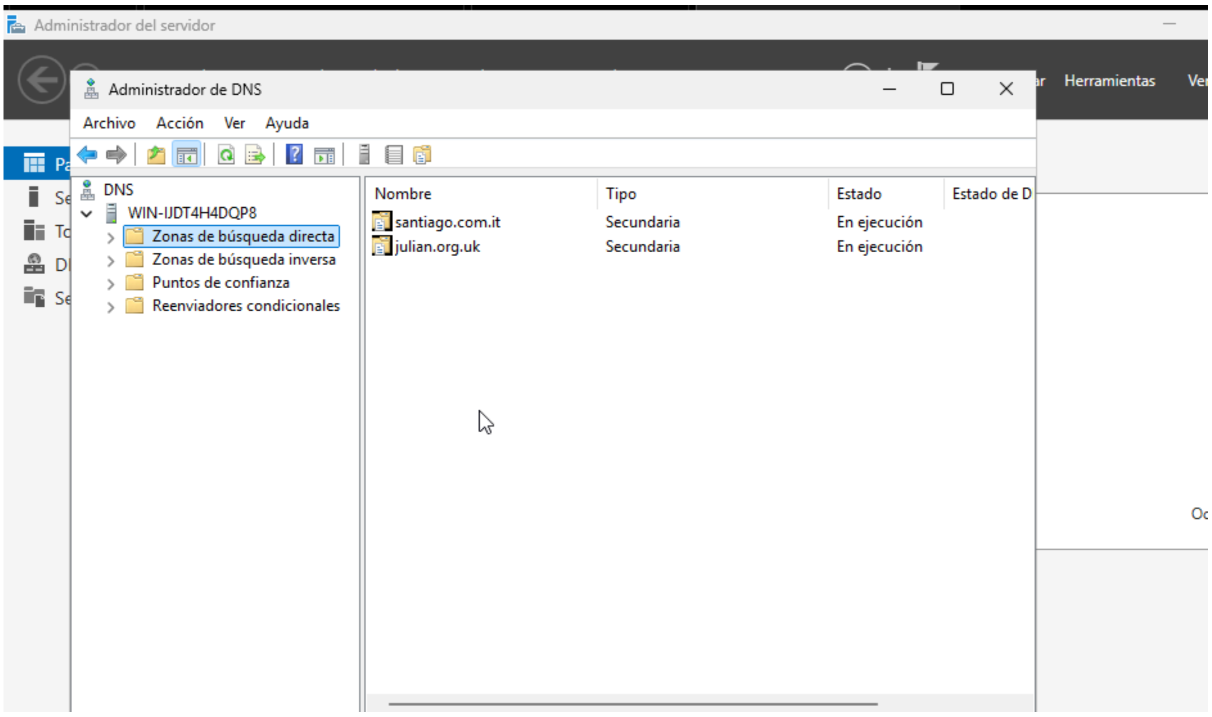
Servidores maestros DNS

La zona secundaria se copia desde uno o más servidores DNS.

Especifique los servidores DNS desde dónde desea copiar la zona. Se establece contacto con los servidores en el orden mostrado.

Servidores

Dirección IP	FQDN de servidor	Validado
<Haga dic aquí para agregar una dirección IP o un nombre DNS>		
10.2.77.31	<Intentando res...	Aceptar


Administrador de DNS

Administrador del servidor
Archivo
Acción
Ver
Ayuda
Herramientas
DNS
WIN-IJDT4H4DQP8
Zonas de búsqueda directa
Zonas de búsqueda inversa
Puntos de confianza
Reenviadores condicionales

Nombre	Tipo	Estado	Estado de D
santiago.com.it	Secundaria	En ejecución	
julian.org.uk	Secundaria	En ejecución	

7. Test its functionality on a client.

i. Configure a client computer to use the DNS server you just set up.

ii. Use the nslookup command to check its operation. Make a video of no more than 5 minutes to explain it.

A. What is the nslookup command used for?

B. Test its operation.

C. Change the DNS server to the school's DNS server and repeat the same queries from the previous point. Document the results.

D. Use the command set type=NS. What did you get? Explain the results.

E. Use the command set debug. What did you get? Explain the results.

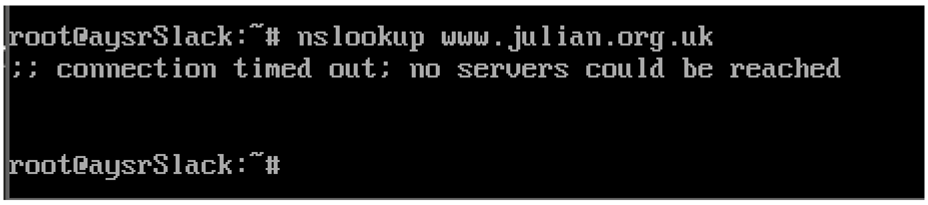
F. Use the command set type=A. What did you get? Explain the results.

G. Use the command set q=MX. What did you get? Explain the results.

Link to the video containing the explanation of point 7:
https://youtu.be/al7TEe9mBtM

8. Test its functionality on the DNS server.
i. Perform the previous step directly on the DNS server. Does it work? Why?

On slackware:

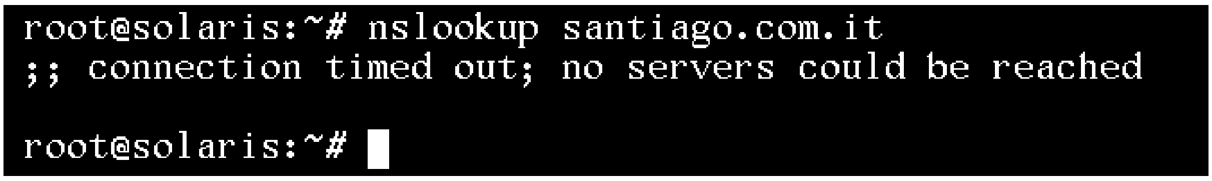


```
root@aysrSlack:~# nslookup www.julian.org.uk
;; connection timed out; no servers could be reached


root@aysrSlack:~#
```

The Slackware server cannot resolve the name because:

1. The /etc/resolv.conf file is not configured to use the local DNS
2. It does not have a working DNS resolver configured
3. It acts as an authoritative server but not as a resolver

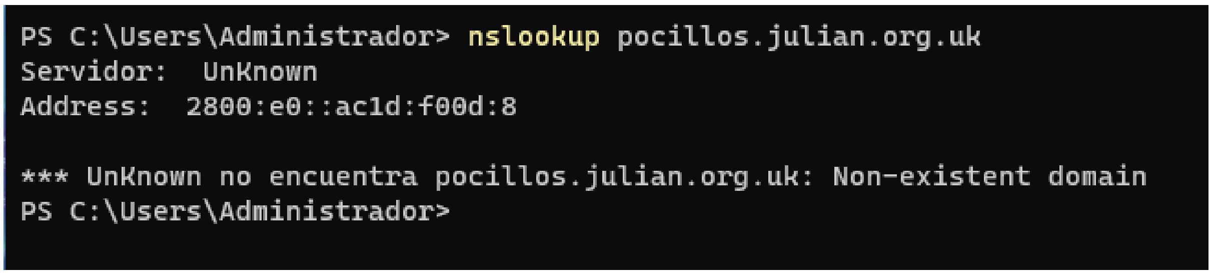
On solaris:



```
root@solaris:~# nslookup santiago.com.it
;; connection timed out; no servers could be reached

root@solaris:~#
```

It doesn't work for the same reasons as Slackware

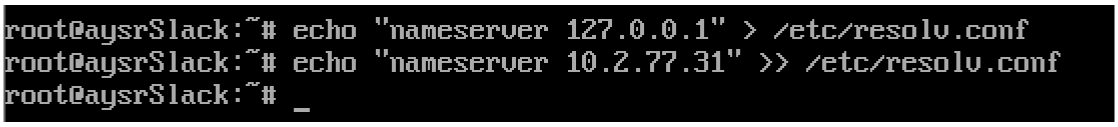
On Windows Server:



```
PS C:\Users\Administrador> nslookup pocillos.julian.org.uk
Servidor:  UnKnown
Address:  2800:e0::ac1d:f00d:8

*** UnKnown no encuentra pocillos.julian.org.uk: Non-existent domain
PS C:\Users\Administrador>
```

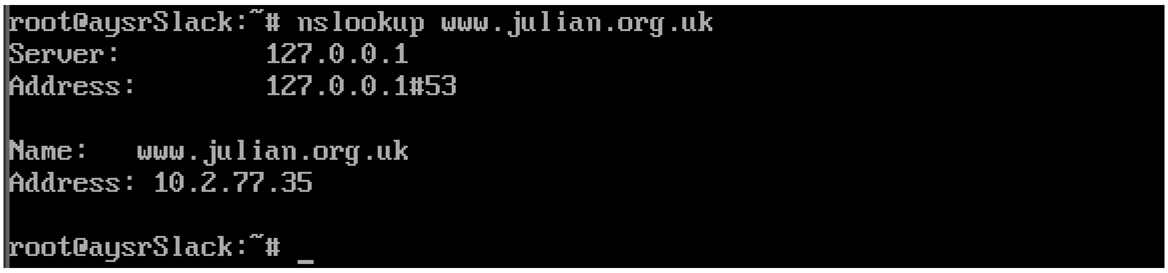
ii. Solve the problem and show the final IP configuration of the server.

On Slackware:



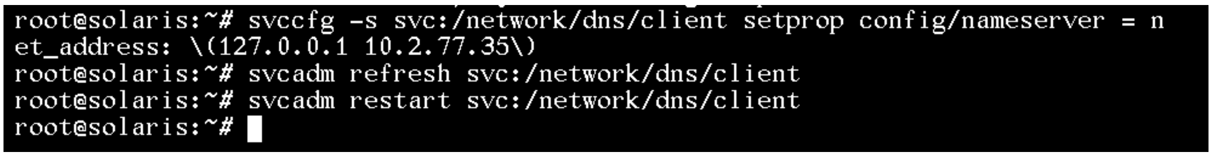
```
root@aysrSlack:~# echo "nameserver 127.0.0.1" > /etc/resolv.conf
root@aysrSlack:~# echo "nameserver 10.2.77.31" >> /etc/resolv.conf
root@aysrSlack:~# _
```

Result:



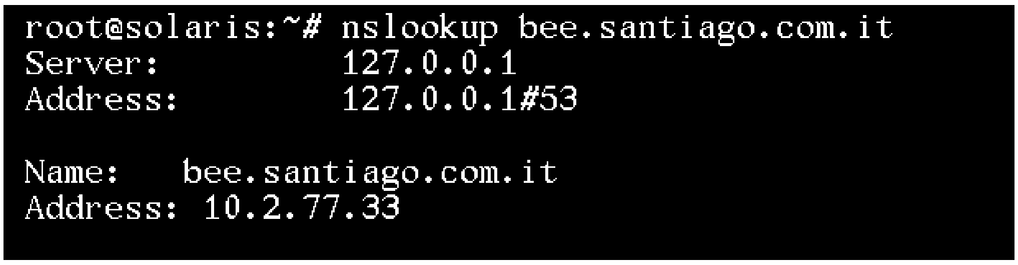
```
root@aysrSlack:~# nslookup www.julian.org.uk
Server:         127.0.0.1
Address:        127.0.0.1#53

Name:   www.julian.org.uk
Address: 10.2.77.35

root@aysrSlack:~# _
```

On Solaris:



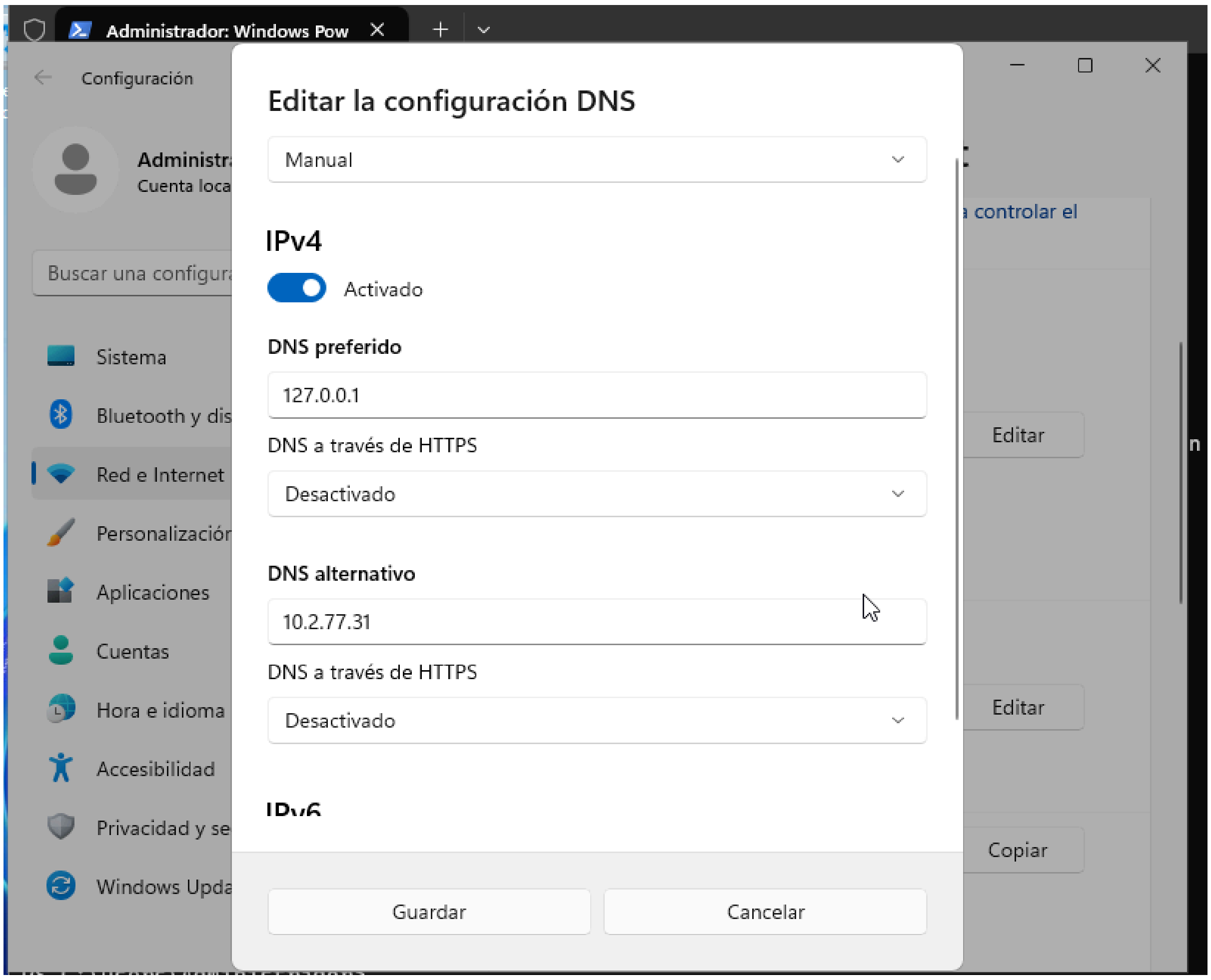
```
root@solaris:~# svccfg -s svc:/network/dns/client setprop config/nameserver = n
et_address: \(127.0.0.1 10.2.77.35\)
root@solaris:~# svcadm refresh svc:/network/dns/client
root@solaris:~# svcadm restart svc:/network/dns/client
root@solaris:~# █
```

Result:



```
root@solaris:~# nslookup bee.santiago.com.it
Server:         127.0.0.1
Address:        127.0.0.1#53

Name:   bee.santiago.com.it
Address: 10.2.77.33
```

On Windows Server:



Editar la configuración DNS

Manual

IPv4

Activado

DNS preferido

127.0.0.1

DNS a través de HTTPS

Desactivado

DNS alternativo

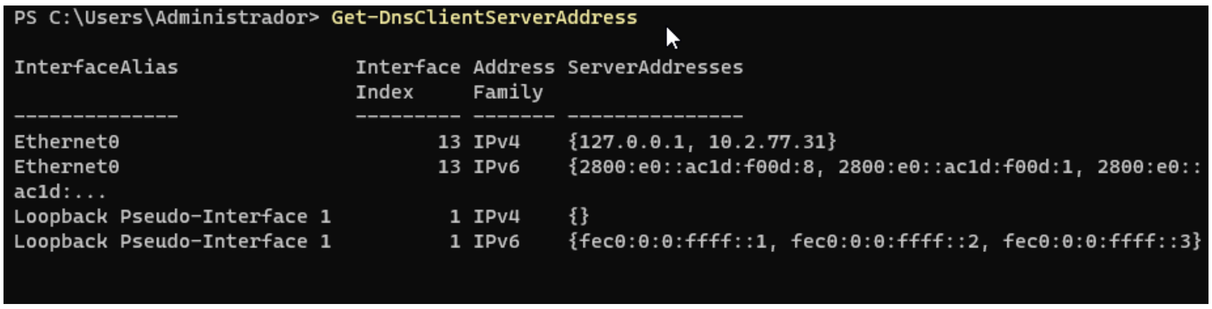
10.2.77.31

DNS a través de HTTPS

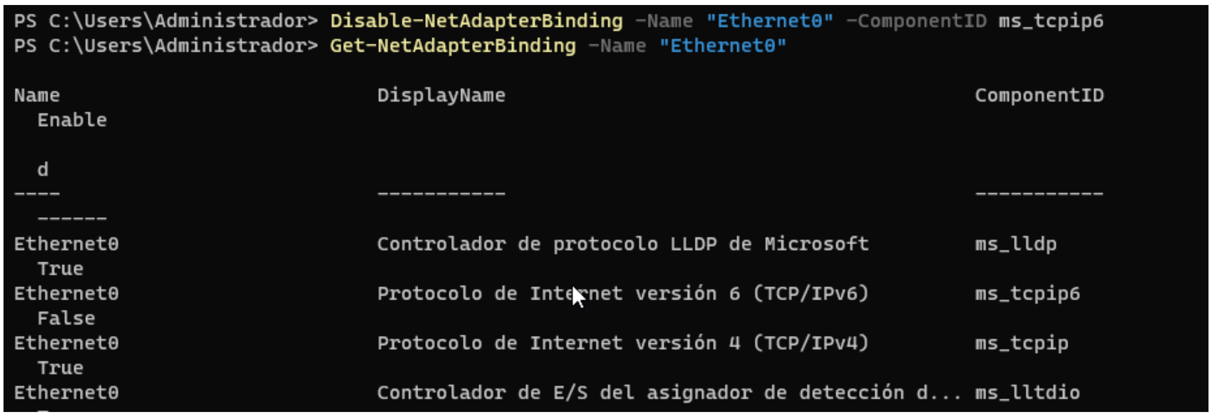
Desactivado

IPv6

Guardar | Cancelar

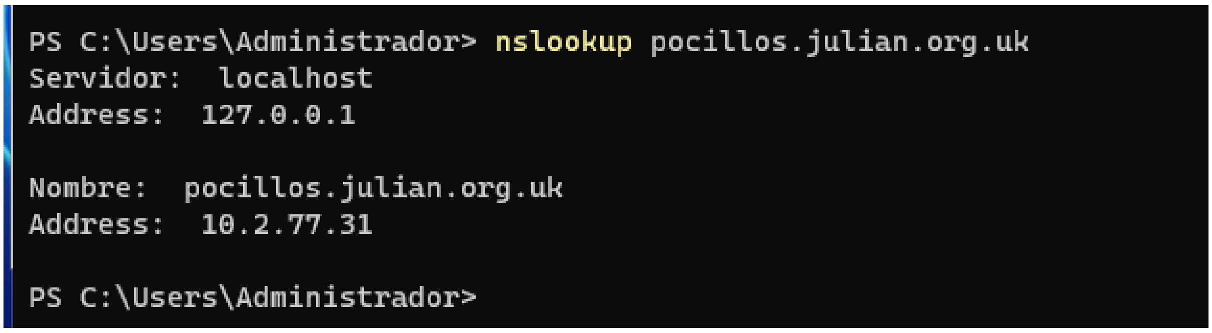


```
PS C:\Users\Administrador> Get-DnsClientServerAddress

InterfaceAlias               Interface Address ServerAddresses
                             Index     Family
--------------               --------- ------- ---------------
Ethernet0                           13 IPv4    {127.0.0.1, 10.2.77.31}
Ethernet0                           13 IPv6    {2800:e0::ac1d:f00d:8, 2800:e0::ac1d:f00d:1, 2800:e0::
ac1d:...
Loopback Pseudo-Interface 1          1 IPv4    {}
Loopback Pseudo-Interface 1          1 IPv6    {fec0:0:0:ffff::1, fec0:0:0:ffff::2, fec0:0:0:ffff::3}
```



```
PS C:\Users\Administrador> Disable-NetAdapterBinding -Name "Ethernet0" -ComponentID ms_tcpip6
PS C:\Users\Administrador> Get-NetAdapterBinding -Name "Ethernet0"

Name                           DisplayName                                        ComponentID
  Enable

  d
----                           -----------                                        -----------
  ------
Ethernet0                      Controlador de protocolo LLDP de Microsoft         ms_lldp
  True
Ethernet0                      Protocolo de Internet versión 6 (TCP/IPv6)         ms_tcpip6
  False
Ethernet0                      Protocolo de Internet versión 4 (TCP/IPv4)         ms_tcpip
  True
Ethernet0                      Controlador de E/S del asignador de detección d... ms_lltdio
```

Result:



```
PS C:\Users\Administrador> nslookup pocillos.julian.org.uk
Servidor:  localhost
Address:  127.0.0.1

Nombre:  pocillos.julian.org.uk
Address:  10.2.77.31

PS C:\Users\Administrador>
```

9. Configure the domain resolution service – DNS (DNS Server) so that it is activated during system startup.
The DNS service was already configured to start automatically on all three servers. It was verified that the services started correctly upon system boot, so no additional changes were necessary.

10. Show the configuration to your instructor.

2. Other Useful Commands

Write Shell programs for the Solaris and Linux Slackware servers that:
(a) Allow configuring a task to run periodically on the system. The user will specify the task to be executed and its frequency via the command line. The parameters should NOT be prompted interactively. For example:

```
solaris# ./schedult-task-script.sh [frequency]
solaris# ./schedult-task-script.sh * * * * *
```

In /Scripts/scheduleTask.sh

(b) Build a Shell with a menu of options, where one option is to exit, and the others execute the desired command and return to the options menu. The menu should allow:

- Displaying the processes currently running on a server. Show the process name, its identifier, memory usage percentage, and CPU usage percentage.
- Searching for a given process by the user and displaying its full information.
- Killing/closing a running process.
- Restarting a running process.

In /Scripts/processes.sh

(c) Create a Shell that allows traversing the file system from a given directory, including subdirectories, and shows the n smallest files within a size specified by the user. The output should indicate: file name, path where it is located, and size. The execution should look like:

```
slackware# ./files-script.sh [no_files] [max_size]
slackware# ./files-script.sh 10 1GB
```

In /Scripts/files-script.sh

Conclusions

The practice demonstrated the importance of DNS services as fundamental component in the stability and reliability of network infrastructures. Through the configuration of primary and secondary DNS servers, it was possible to guarantee both redundancy and consistency across different systems.

It was observed that even when working with heterogeneous operating systems such as Solaris, Slackware, and Windows Server, the principles of configuration remain aligned with POSIX standards, which facilitates interoperability. This highlighted the value of adopting portable scripts and standardized practices to ensure cross-platform compatibility.

Additionally, the use of diagnostic tools such as nslookup, dig, and log inspection proved essential for validating the correct functionality of the services. These tools not only allowed the detection of potential errors but also strengthened the ability to troubleshoot and maintain critical network components.

Finally, this practice emphasized that proper service configuration, combined with systematic verification and automation through shell scripting, significantly contributes to the robustness, scalability, and reliability of enterprise systems.