UNIVERSIDAD DOCTOR ANDRÉS BELLO

FACULTAD:

Facultad de Tecnología e Innovación

CARRERA:

Ingeniería en sistemas y Computación

ASIGNATURA:

“Redes II”

TEMA:

“”

ALUMNOS:

Richard Alexis Avalos Garcia

Ivania Nicole Hernandez Mendez

Rene Israel Rodriguez Palacios

Andres Felipe Galan Hernandez

Índice

Introducción

Este documento tiene como propósito guiar paso a paso la configuración de servicios fundamentales en redes locales, integrando servidores DNS, Web y DHCP en sistemas Linux, así como el uso de Apache2 y la asignación de IP estática en openSUSE Tumbleweed. Se abordan configuraciones prácticas con distribuciones como Rocky Linux, AlmaLinux y openSUSE, enfocándose en el despliegue de un entorno de red funcional y eficiente.

El servidor DNS permite resolver nombres de dominio a direcciones IP dentro de la red, mejorando la gestión de recursos. El servidor web, implementado con Nginx y Apache2, proporciona el acceso a contenidos alojados en el sistema. La asignación de IP estática y la configuración mediante Wicked o NetworkManager aseguran la disponibilidad constante del servidor. Finalmente, el servidor DHCP complementa la red proporcionando direcciones IP dinámicas cuando es necesario.

El objetivo principal es ofrecer una guía clara y funcional que refleje casos reales en los que un administrador de sistemas debe implementar soluciones que garanticen el correcto funcionamiento y acceso de los servicios en la red local.

Objetivo

El objetivo principal de este documento es proporcionar una guía completa, clara y funcional para la configuración de servicios esenciales dentro de una red local, incluyendo la instalación y gestión de un servidor DNS, un servidor web (usando tanto Nginx como Apache2), y un servidor DHCP, así como la asignación de direcciones IP estáticas utilizando las herramientas Wicked y NetworkManager en sistemas Linux, especialmente en Rocky Linux, AlmaLinux y openSUSE Tumbleweed.

Este documento busca que el lector no solo ejecute correctamente los comandos y configuraciones indicadas, sino que también comprenda el propósito de cada paso y su impacto dentro de la infraestructura de red. A través de esta práctica, se pretende fortalecer los conocimientos sobre administración de sistemas y redes, preparándolo para enfrentar escenarios reales en entornos laborales, donde la correcta implementación y mantenimiento de estos servicios es clave para garantizar conectividad, disponibilidad de recursos y estabilidad en la comunicación entre dispositivos.

Implementación de un dominio Linux con clientes Linux - Windows

Definición

Dominio en DNS (Domain Name System): En el ámbito de Internet, un dominio es una dirección jerárquica que se utiliza para identificar recursos en la red, como los nombres de los sitios web.

DNS (Domain Name System): Es un sistema jerárquico y distribuido que permite traducir nombres de dominio en direcciones IP, facilitando que los dispositivos se comuniquen entre sí a través de redes como Internet. Gracias al DNS, los usuarios pueden acceder a sitios web usando nombres fáciles de recordar, en lugar de memorizar largas secuencias numéricas.

Tipos de Dominios:

1. Dominio de Nivel Superior (TLD - Top Level Domain)

Son los que aparecen al final de un nombre de dominio.

- **Genéricos (gTLD):** .com, .org, .net, .info
- **Geográficos (ccTLD):** Representan países, como .sv (El Salvador), .mx (México), .us (EE. UU.)
- **Patrocinados:** .edu, .gov, .mil, usados por instituciones específicas

2. Dominio de Segundo Nivel (SLD - Second Level Domain)

Es el nombre que elige el propietario del sitio web, justo antes del TLD. **Ejemplo:** en google.com, "google" es el segundo nivel.

3. Dominio de Tercer Nivel (Subdominios)

Son dominios creados dentro de un dominio de segundo nivel. **Ejemplo:** mail.google.com o blog.ejemplo.com Aquí "mail" o "blog" son subdominios del dominio principal.

4. Dominio raíz

Representado por un punto (.), es el nivel más alto en la jerarquía del DNS. No suele mostrarse, pero siempre está al final del dominio (por ejemplo: www.google.com.).

Tipos de DNS

1. Servidor DNS Recursivo

Es el primero que recibe la solicitud cuando escribes un dominio. Se encarga de buscar la IP correspondiente, preguntando a otros servidores si es necesario. Guarda respuestas en caché para acelerar futuras consultas.

2. Servidor Raíz (Root DNS Server)

Es el nivel más alto del sistema DNS. Redirige al servidor correspondiente del TLD (.com, .org, .sv, etc.). Hay 13 servidores raíz principales a nivel mundial (con muchas réplicas).

3. Servidor TLD (Top-Level Domain)

Maneja dominios de nivel superior como .com, .net, .org, .sv. Redirige hacia el servidor autoritativo del dominio consultado.

4. Servidor Autoritativo

Tiene la respuesta final: la dirección IP del dominio buscado. Es el único que puede dar respuestas oficiales para los dominios que administra.

Arquitectura

1. Descripción General

En esta infraestructura se ha implementado un Servidor DNS basado en Rocky Linux utilizando el servicio BIND. El objetivo principal de este servidor es gestionar la resolución de nombres dentro de la red local, facilitando el acceso a servicios mediante nombres de dominio amigables en lugar de IPs.

La dirección IP asignada al servidor DNS es 192.168.0.100, mientras que el servidor DHCP está en 192.168.0.1 (normalmente el router).

2. Servicios que Ofrece El Servidor DNS

Resolución de nombres interna: Traduce nombres de dominio locales, www.mesa4.unab, hacia las IPs internas de los servidores.

Gestión de zonas DNS:

- Zona directa: Mapea nombres de dominio a direcciones IP.
- Zona inversa: Mapea direcciones IP a nombres de dominio.

3. Relación del DNS con los Servidores Web

El servidor DNS facilita la conexión entre los usuarios y los servidores web:

- Servidor Web Principal:
- Sistema operativo: AlmaLinux
- Servidor Web: NGINX
- IP: 192.168.0.5
- Dominios atendidos: www.mesa4.unab

Servidor Web de Respaldo

- Sistema operativo: OpenSUSE
- Servidor Web: Apache
- Funciona como backup en caso de falla del servidor principal

4. Ventajas de esta Arquitectura

Separación de roles: Cada servicio (DNS, Web principal, Web de respaldo) corre en su propio servidor, mejorando la eficiencia y seguridad.

- **Alta disponibilidad:** Existe un servidor de respaldo preparado para asumir funciones en caso de fallo del servidor principal.

- **Control total:** Al tener nuestro propio servidor DNS, no dependemos de servicios externos para gestionar dominios internos.
- **Facilidad de escalabilidad:** Nuevos servidores o servicios pueden ser añadidos fácilmente al DNS.

5. Tecnologías Clave Utilizadas

- **Rocky Linux:** Sistema operativo estable, ideal para servidores.
- **BIND:** Servicio DNS líder para entornos de red empresarial.
- **NGINX y Apache:** Dos de los servidores web más potentes y usados en el mundo.
- **OpenSUSE:** Sistema operativo confiable para el servidor de respaldo.

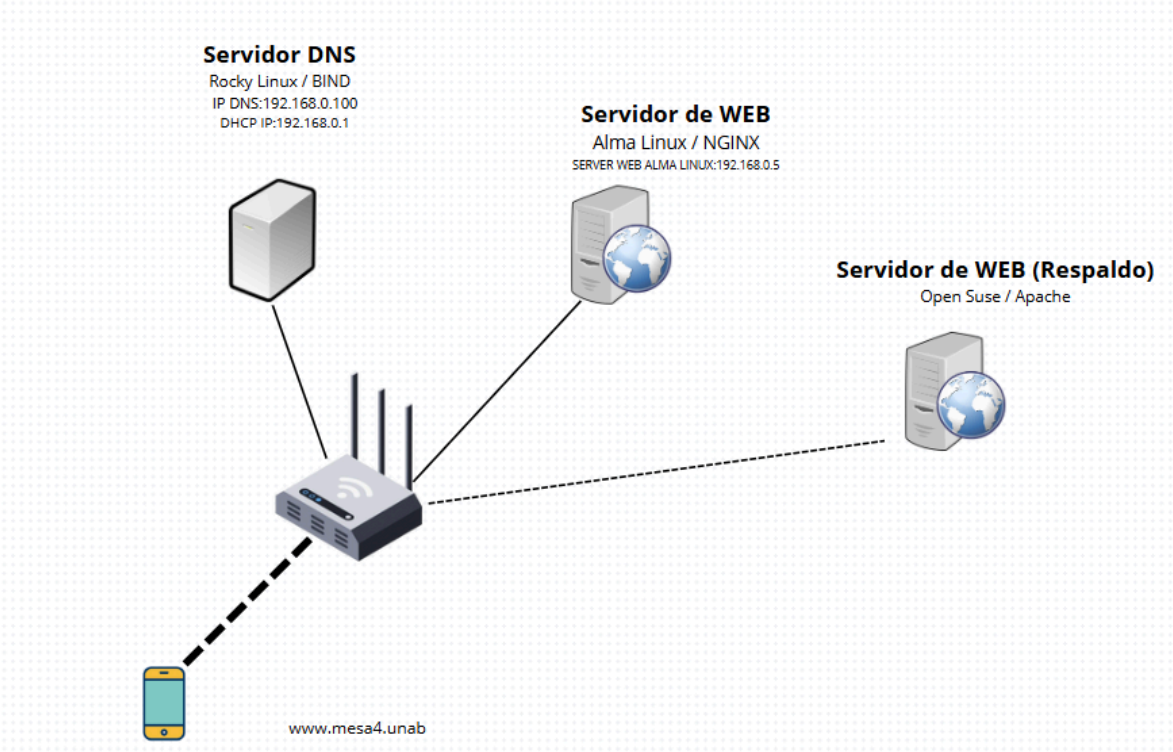
6. Mejoras Futuras

Para hacer la infraestructura aún más robusta, se pueden implementar las siguientes mejoras:

- **Implementar DNSSEC:** Para proteger la integridad y autenticidad de las respuestas DNS.
- **Configuración de alta disponibilidad (HA):** Tener un segundo servidor DNS como respaldo automático.
- **Agregar resolución externa:** Usar reenviadores hacia servidores DNS públicos para resolver dominios de Internet.

- **Monitoreo de servicios:** Implementar herramientas como Zabbix o Nagios para recibir alertas de caídas o problemas de los servidores.

Ilustración:



Tecnologías

Rocky Linux

Es una distribución de Linux está fundamentada en Red Hat Enterprise Linux (RHEL), se desarrolló como sustituto de CentOS, proporcionando estabilidad y respaldo a largo plazo.

Es perfecto para ambientes de producción y servidores gracias a su seguridad, fiabilidad y compatibilidad con empresas.

Alma Linux

Otra distribución fundamentada en RHEL, que también se originó como opción frente a CentOS. La comunidad la mantiene y CloudLinux la patrocina. Se centra en proporcionar una plataforma gratuita, firme y compatible con aplicaciones de negocios.

OpenSuse

Se trata de una distribución de Linux respaldada por la comunidad y SUSE. Se dirige a usuarios avanzados, programadores y gerentes de sistemas. Proporciona instrumentos sólidos para la administración y configuración del sistema.

TumbleWeed

Se trata de la versión de lanzamiento constante (rolling release) de openSUSE. En contraposición a las versiones estándar, Tumbleweed siempre se actualiza con los paquetes y tecnologías más recientes, perfecto para aquellos que buscan lo más reciente en el software Linux.

Bind

Es el programa DNS más empleado en los sistemas Unix/Linux. Facilita la transformación de nombres de dominio en direcciones IP (y a la inversa), operando como el núcleo de un servidor DNS. Es altamente personalizable y se utiliza en redes tanto locales como internacionales.

DHCP (Dynamic Host Configuration Protocol)

Es un protocolo de red que otorga direcciones IP de manera automática a los dispositivos en una red. Esto previene conflictos de IP y simplifica la gestión, particularmente en redes amplias o con numerosos dispositivos vinculados.

Nginx

Es un servidor web de gran eficiencia, reconocido por su eficacia, escaso uso de recursos y habilidad para gestionar numerosas conexiones al mismo tiempo. Además, puede desempeñarse como proxy inverso, equilibrador de carga o servidor de correo electrónico.

Apache2

Es uno de los servidores web más famosos y utilizados a nivel global. Su estructura modular facilita su personalización de acuerdo a las exigencias del sitio web. Es perfecto para contextos que demandan flexibilidad, compatibilidad y una extensa documentación.

Wicked

Es una herramienta de configuración de red utilizada en openSUSE. Permite gestionar interfaces de red mediante archivos de configuración manuales. Es más común en entornos sin interfaz gráfica y es útil para configuraciones estáticas o avanzadas.

NetworkManager

Es un recurso para administrar conexiones de red en Linux, particularmente en ambientes gráficos. Promueve la conexión a redes de cables, inalámbricas, VPN, entre otras, y resulta más sencillo para aquellos usuarios que prefieren no modificar archivos de forma manual.

ARP (Address Resolution Protocol)

ARP es un protocolo de red utilizado para resolver direcciones IP en direcciones MAC dentro de una red local. En otras palabras, permite que un dispositivo descubra la dirección física (MAC) de otro equipo a partir de su dirección IP, lo cual es esencial para la comunicación a nivel de red Ethernet. Es una parte fundamental del funcionamiento de las redes locales.

EPEL (Extra Packages for Enterprise Linux)

EPEL es un repositorio de software mantenido por la comunidad Fedora que proporciona paquetes adicionales para distribuciones basadas en RHEL (como Rocky Linux y AlmaLinux). Muchos programas que no están en los repositorios oficiales de estas distribuciones se pueden instalar fácilmente desde EPEL. Se activa normalmente con el paquete epel-release.

net-tools

Es un paquete clásico en sistemas Linux que contiene utilidades de red como ifconfig, netstat, route, arp, entre otros. Aunque muchas de estas herramientas han sido reemplazadas por comandos más modernos (como ip de iproute2), net-tools sigue siendo útil para compatibilidad y diagnósticos rápidos en entornos tradicionales.

Configuración DNS

1. ¿Por qué Rocky Linux Minimal?

Elegimos Rocky Linux Minimal porque es una distribución ligera, eficiente y sencilla de manejar. Ideal para un servidor que no requiere entorno gráfico y que optimiza recursos.

2. Configurar la Red Manualmente

Primero, necesitamos configurar la interfaz de red.

ip a

Fíjate en la interfaz que aparece en la línea con el número 2 (por ejemplo, enp0s3).

Edita el archivo de configuración:

```
sudo vi /etc/sysconfig/network-scripts/ifcfg-enp0s3
```

O usa nano si prefieres:

```
sudo nano /etc/sysconfig/network-scripts/ifcfg-enp0s3
```

Configuración esencial:

```
TYPE=Ethernet
BOOTPROTO=static
NAME=enp0s3
DEVICE=enp0s3
ONBOOT=yes
```

```
IPADDR=192.168.0.10

NETMASK=255.255.255.0

GATEWAY=192.168.0.1

DNS1=8.8.8.8
```

Guarda y reinicia el servicio de red

```
sudo systemctl restart network
```

3. Instalar BIND (Berkeley Internet Name Domain)

BIND es el software más utilizado para implementar servidores DNS.

¿Qué hace BIND?

Convierte nombres de dominio como misitio.com en direcciones IP como 192.168.0.10 (y viceversa).

Instalación:

```
sudo dnf install bind bind-utils
```

4. Crear el Archivo de Zona DNS

Este archivo define cómo resolver los nombres de dominio a IPs en tu red.

Crear archivo de zona:

```
sudo vi /var/named/mesa4.unab.db
```

Contenido del archivo mesa4.unab.db:

```
$TTL 86400
@   IN  SOA     mesa4.unab. root.mesa4.unab. (
            2025042501  ; Serial
            3600        ; Refresh
            1800        ; Retry
            604800      ; Expire
            86400 )     ; Minimum TTL

@       IN  NS      mesa4.unab.
@       IN  A       192.168.0.10
www     IN  A       192.168.0.5
```

```
www     IN  A       192.168.0.6
web     IN  A       192.168.0.5
web     IN  A       192.168.0.6
```

5. Configurar el Archivo Principal de BIND

Editamos el archivo principal de configuración para definir nuestras zonas y dirección IP.

Editar named.conf:

```
sudo vi /etc/named.conf
```

En la sección options agrega tu IP:

```
options {
    listen-on port 53 { 127.0.0.1; 192.168.0.10; };
    allow-query     { localhost; 192.168.0.0/24; };
    recursion yes;
    ...
};
```

Agrega la zona:

```
zone "mesa4.unab" IN {
    type master;
    file "/var/named/mesa4.unab.db";
};
```

6. Reiniciar y Habilitar el Servicio

```
sudo systemctl restart named
sudo systemctl enable named
```

Configuración WEB

1. Configurar Red y DNS

Primero debemos asignar una IP estática al servidor web y definir como DNS el servidor Rocky (que ya configuramos como DNS).

Verifica la interfaz de red

```
ip a
```

Fíjate en el nombre de la interfaz (por ejemplo, enp0s3).

Edita el archivo de configuración de red:

```
sudo vi /etc/sysconfig/network-scripts/ifcfg-enp0s3
```

Ejemplo de configuración:

```
TYPE=Ethernet
BOOTPROTO=static
NAME=enp0s3
DEVICE=enp0s3
ONBOOT=yes
IPADDR=192.168.0.5
NETMASK=255.255.255.0
```

```
GATEWAY=192.168.0.1

DNS1=192.168.0.10
```

Reinicia el servicio de red:

```
sudo systemctl restart network
```

Puedes cambiar las IPs a las que mejor se adapten a tu red.

2. Instalar y activar Apache y nginx

Instala Nginx:

```
sudo dnf install nginx -y
```

Instala Apache:

```
sudo zypper install apache2
```

Inicia y habilita Nginx:

```
sudo systemctl start nginx

sudo systemctl enable nginx
```

Inicia y habilita Apache:

```
sudo systemctl enable apache2

sudo systemctl start apache2
```

Verifica el estado:

```
sudo systemctl status nginx

sudo systemctl status apache2
```

3. Abrir puertos HTTP/HTTPS en el firewall

Para que el servidor web sea accesible desde otros equipos:

```
sudo firewall-cmd --permanent --add-service=http

sudo firewall-cmd --permanent --add-service=https

sudo firewall-cmd --reload
```

4. Crear archivo de configuración del sitio en Nginx y Apache

Creamos un archivo de configuración para nuestro sitio web.

```
Nginx: sudo vi /etc/nginx/conf.d/mesa4.conf

Apache: sudo nano /etc/apache2/vhosts.d/misitio.conf
```

Ejemplo del archivo mesa4.conf:

```
server {
    listen 80;
    server_name mesa4.unab;

    root /var/www/mesa4.unab;
    index index.html;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

Ejemplo del archivo misitio.conf:

```
<VirtualHost *:80>
    ServerAdmin admin@misitio.com
    ServerName misitio.com
    ServerAlias www.misitio.com

    DocumentRoot /srv/www/misitio

    <Directory /srv/www/misitio>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog /var/log/apache2/misitio_error.log
    CustomLog /var/log/apache2/misitio_access.log combined
</VirtualHost>
```

5. Crear el directorio raíz del sitio

```
sudo mkdir -p /var/www/mesa4.unab
```

6. Crear una página web de prueba

```
echo "<h1>Hola desde Nginx en AlmaLinux</h1>" | sudo tee
/var/www/mesa4.unab/index.html
```

7. Verificar configuración y recargar Nginx

Verificar sintaxis:

```
sudo nginx -t
```

Recargar Nginx:

```
sudo systemctl reload nginx
```

Recargar Apache:

```
sudo systemctl restart apache
```

Conexión del DNS al Servidor Web

Ya que en el servidor DNS configurado en Rocky agregamos un registro como este en mesa4.unab.db:

```
www     IN  A   192.168.0.5
```

Entonces, cualquier cliente que use ese DNS y acceda a www.mesa4.unab será redirigido a tu servidor web en AlmaLinux. Así se completa la conexión entre el servidor DNS y el servidor web.

Configuración de Keepalived para Failover de Servidor Web

1. Objetivo

Configurar alta disponibilidad (HA) de servicios web usando Keepalived con una IP virtual (VIP) 192.168.0.150, entre:

Servidor Master: openSUSE Tumbleweed (Apache, IP 192.168.0.6)

Servidor Backup: AlmaLinux 8.10 (Nginx, IP 192.168.0.5)

2. Red

Componente	Datos
VIP	192.168.0.150
Master	192.168.0.6 (openSUSE + Apache)
Backup	192.168.0.5 (AlmaLinux + Nginx)

Interfaz de red	enp0s3

3. Instalación de Keepalived

En openSUSE Tumbleweed:

sudo zypper refresh

sudo zypper install keepalived

En AlmaLinux 8.10:

```
sudo dnf install epel-release -y
sudo dnf install keepalived -y
```

4. Configuración de Keepalived

En openSUSE (Master)

Archivo: /etc/keepalived/keepalived.conf

```
vrrp_instance VI_1 {
    state MASTER
    interface enp0s3
    virtual_router_id 51
    priority 150
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass claveSegura123
    }
    virtual_ipaddress {
        192.168.0.150
```

```
    }
}
```

En AlmaLinux (Backup)

Archivo: /etc/keepalived/keepalived.conf

```
vrrp_instance VI_1 {
    state BACKUP
    interface enp0s3
    virtual_router_id 51
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass claveSegura123
    }
    virtual_ipaddress {
        192.168.0.150
    }
}
```

5. Configuración del Firewall

En openSUSE (usa firewalld):

```
# Asegurar que firewalld está activo
sudo systemctl start firewalld
sudo systemctl enable firewalld

# Abrir HTTP (puerto 80)
sudo firewall-cmd --permanent --add-service=http
sudo firewall-cmd --reload
```

```
# (VRRP usa el protocolo 112 que ya suele estar permitido para
tráfico interno)
```

En AlmaLinux (también usa `firewalld`):

```
# Asegurar firewalld activo
sudo systemctl start firewalld
sudo systemctl enable firewalld

# Abrir HTTP (puerto 80)
sudo firewall-cmd --permanent --add-service=http
sudo firewall-cmd --reload
```

6. Administración de Keepalived

En openSUSE:

```
sudo systemctl enable keepalived
sudo systemctl start keepalived
sudo systemctl status keepalived
```

En AlmaLinux:

```
sudo systemctl enable keepalived
sudo systemctl start keepalived
sudo systemctl status keepalived
```

7. Servicios Web

Sistema	Servicio Web	Instalación	Puerto
openSUSE	Apache	sudo zypper install apache2	80
AlmaLinux	Nginx	sudo dnf install nginx	80

Configuracion Router

Es esencial que el enrutador esté correctamente configurado para garantizar que los dispositivos de la red se comuniquen bien entre sí y con los servicios que hemos establecido, como el servidor DNS. Si el proveedor de internet (ISP) no ha impuesto restricciones al enrutador, podemos configurarlo fácilmente y asignarle la dirección IP del servidor DNS que hemos configurado en Rocky Linux. Por ejemplo, durante esta práctica, se utilizó la dirección 192.168.0.100, pero se puede seleccionar cualquier otra dirección IP dentro del mismo rango de red.

Es crucial que tanto los servidores como el enrutador estén en la misma subred. Esto implica que las direcciones IP deben estar en el mismo rango para permitir la comunicación. Por ejemplo:

- Si el enrutador tiene la dirección IP 192.168.1.1 y el servidor DNS tiene la IP 192.168.0.100

No podrán comunicarse porque están en diferentes subredes. Para resolver esto, se puede cambiar la dirección IP del enrutador o ajustar las direcciones IP de los servidores para que todos estén, por ejemplo, en la subred 192.168.0.x.

¿Y si el enrutador tiene restricciones del ISP?

Configuramos el enrutador en modo puente. Este modo desactiva las funciones de enrutamiento y DHCP, permitiendo que otro dispositivo asuma estas tareas.

Retos tecnológicos

1. Restricciones del Router impuestas por el ISP

Uno de los principales retos fue la imposibilidad de modificar los ajustes del DNS en un router proporcionado por el proveedor de servicios de internet (modelo Arris). Este dispositivo presentaba bloqueos en sus opciones de configuración, exigiendo claves o permisos especiales para alterar parámetros esenciales como el servidor DNS.

Solución: Se optó por configurar el router en modo Bridge, desactivando su función de asignación de IP y transfiriendo esa responsabilidad a un servidor DHCP configurado manualmente en Rocky Linux. Esto permitió un control total sobre la red sin limitaciones del hardware del proveedor.

2. Incompatibilidades entre versiones de distribuciones y paquetes

Otro reto frecuente fue la incompatibilidad entre versiones de software al trabajar con distintas distribuciones Linux (Rocky Linux, AlmaLinux y openSUSE Tumbleweed). Por ejemplo:

- En algunas versiones, ciertos comandos o rutas de archivos no estaban disponibles.

Paquetes como bind, dhcp-server, o configuraciones de Apache/Nginx variaban levemente en sintaxis o ubicación de archivos de configuración dependiendo de la versión de la distro. Esto obligó a investigar y ajustar varios pasos de la guía para cada sistema, especialmente en Tumbleweed, que por ser una distribución rolling release, cambia constantemente los paquetes disponibles.

Solución: Se recurrió a la documentación oficial de cada distribución y al uso de comandos como dnf info y zypper search para validar la disponibilidad y compatibilidad de los paquetes. Además, se adaptaron las rutas y configuraciones específicas según la distro, documentando las diferencias encontradas para futuras implementaciones.

Plan de contingencias

1. Problemas con el Servidor DNS

Posibles causas:

- El servicio DNS se detiene por error del sistema.
- Hay errores en los archivos de configuración.
- El archivo de zona DNS está mal escrito o dañado.
- Cambios en la red que afectan la conectividad.

Medidas correctivas:

- Verificar el estado del servidor DNS y reiniciarlo si es necesario.
- Revisar y corregir configuraciones, especialmente archivos de zona y parámetros de red.
- Restaurar archivos de respaldo si los originales fallan.

Medidas preventivas:

- Realizar respaldos periódicos de la configuración.
- Validar los archivos antes de aplicarlos.

- Documentar las direcciones IP y configuraciones de red para facilitar futuras verificaciones.

2. Problemas con el Servidor Web

Posibles causas:

- El servicio web se detiene o no inicia correctamente.
- Archivos del sitio están mal ubicados o no existen.
- Configuraciones incorrectas en el servidor web.
- El firewall o red impide el acceso.

Medidas correctivas:

- Verificar si el servidor está en funcionamiento y reiniciarlo en caso necesario.
- Comprobar que el contenido del sitio esté en su lugar.
- Revisar configuraciones del servidor y red.

Medidas preventivas:

- Hacer una copia del contenido web y la configuración del servidor.
- Evitar modificar configuraciones directamente sin respaldo previo.
- Probar los cambios antes de ponerlos en producción.

3. Fallos en la Red

Posibles causas:

- Error en la configuración de la IP.
- El servidor no puede comunicarse con los demás equipos.
- El gateway o el DNS no están configurados correctamente.

Medidas correctivas:

- Confirmar que la red esté bien configurada y que los equipos se pueden comunicar.
- Ajustar la configuración si se detecta alguna inconsistencia.

Medidas preventivas:

- Asignar direcciones IP fijas correctamente documentadas.
- Realizar pruebas de conexión entre los servidores de forma regular.
- Evitar cambios en la red sin registrar las configuraciones anteriores.

Respaldos Recomendados

Es importante contar con copias de seguridad de:

- Archivos de configuración del servidor DNS.
- Archivos del sitio web y configuraciones del servidor web.
- Configuración de red de ambos servidores.

Verificación Regular

Se recomienda establecer una rutina de revisión para asegurarse de que:

- El servidor DNS está resolviendo nombres correctamente.
- El servidor web está mostrando la página como se espera.
- Ambos servicios están activos y accesibles desde otros equipos.

Esto se puede hacer de forma diaria o al iniciar cada jornada de trabajo.

Plan de Recuperación

En caso de un fallo mayor:

1. Identificar qué servicio está fallando (DNS, Web o red).
2. Consultar la documentación y los respaldos.
3. Restaurar las configuraciones o archivos desde el respaldo.
4. Verificar que los servicios estén funcionando nuevamente.

Los respaldos deben hacerse cada vez que se realicen cambios importantes o, como mínimo, una vez por semana.

Conclusiones

A lo largo de esta práctica se logró implementar un entorno de red funcional mediante la configuración de servidores DNS, Web y DHCP, así como la correcta asignación de direcciones IP estáticas en distintas distribuciones Linux. Este proceso no solo permitió aplicar conocimientos técnicos, sino también enfrentar y resolver situaciones reales que ocurren en la administración de redes.

Entre los aprendizajes más destacados se encuentran:

- La implementación de un servidor DNS con BIND demostró ser fundamental para gestionar nombres de dominio dentro de la red local, facilitando la identificación de dispositivos sin necesidad de recurrir a direcciones IP.
- Se configuraron servidores web utilizando tanto Nginx como Apache2, lo que permitió comparar su uso, estructura y características, y garantizar la correcta publicación de sitios desde diferentes entornos Linux.

- La configuración de un servidor DHCP dentro de la red permitió asumir el control de la asignación dinámica de IPs, especialmente útil cuando se trabaja con routers que presentan limitaciones o restricciones por parte del ISP.
- Se abordó la asignación de IP estática usando Wicked y también el uso de NetworkManager, comprendiendo las ventajas, desventajas y cuándo es adecuado usar cada herramienta, dependiendo del entorno gráfico o de la automatización deseada.
- Se resolvieron retos técnicos importantes, como la incompatibilidad entre versiones de distribuciones, el bloqueo de configuraciones DNS en routers cerrados por el ISP, y la correcta adaptación de configuraciones específicas según la distribución utilizada.

En conclusión, esta práctica no solo refuerza habilidades técnicas esenciales en el área de redes y servidores Linux, sino que también fomenta el pensamiento crítico, la investigación y la capacidad de resolución de problemas que se presentan comúnmente en entornos reales de trabajo. El conocimiento adquirido sienta una base sólida para futuros proyectos relacionados con la administración de redes y servicios.

Biografía

¿Qué es Rocky Linux y debería considerarlo? (s. f.). https://es.linux-console.net/?p=13299

Cynthia Sanchez: front-end and UI, Zvezdana Marjanovic: graphic design. (s. f.). *The makers' choice for sysadmins, developers and desktop users.* openSUSE. https://www.opensuse.org/

De Luz, S. (2024, 8 octubre). Configura un servidor DNS con Bind9 en tu servidor Linux. *RedesZone*. https://www.redeszone.net/tutoriales/servidores/configurar-servidor-dns-bind-linux/#447771-que-es-bind

Welcome to The Apache Software Foundation. (s. f.). https://www.apache.org/

nginx documentation. (s. f.). https://nginx.org/en/docs/

The Linux Foundation. (2025, 23 enero). Resources - Linux Foundation - Education. Linux Foundation - Education.

https://training.linuxfoundation.org/resources/?_sft_content_type=tutorial