Despliegue de aplicaciones web

▼ DHCP

DHCP (Protocolo de Configuración Dinámica de Host)

¿Qué es DHCP?

- **DHCP** significa *Dynamic Host Configuration Protocol*.
- Su función es asignar automáticamente configuraciones de red a dispositivos en una red.
- Configura parámetros como:
  - **Dirección IP**
  - **Máscara de subred**
  - **Puerta de enlace predeterminada**
  - **Servidores DNS**

Instalación del Servidor DHCP

Para instalar el servidor DHCP en sistemas basados en Debian/Ubuntu:

```
sudo apt-get update
sudo apt-get install isc-dhcp-server
```

Configuración del Servidor DHCP

- El archivo principal de configuración es `/etc/dhcp/dhcpd.conf`.
- Aquí defines cómo el servidor asignará las direcciones IP y otros parámetros.

Ejemplo de Configuración de una Subred

```
subnet 192.168.1.0 netmask 255.255.255.0 {
  range 192.168.1.60 192.168.1.90;  # Rango de IPs a asi
gnar
  option routers 192.168.1.254;     # Puerta de enlace
  option domain-name-servers 8.8.8.8, 8.8.4.4;  # Servid
ores DNS
}
```

Gestión del Servicio DHCP

Comandos para controlar el servicio:

- **Iniciar el servicio**:

  ```
  sudo systemctl start isc-dhcp-server
  ```

- **Detener el servicio**:

  ```
  sudo systemctl stop isc-dhcp-server
  ```

- **Reiniciar el servicio**:

  ```
  sudo systemctl restart isc-dhcp-server
  ```

- **Ver el estado del servicio**:

  ```
  sudo systemctl status isc-dhcp-serv
  ```

Notas Importantes

- Si el servidor tiene múltiples interfaces de red, especifica cuáles usar en `/etc/default/isc-dhcp-server`:

  ```
  INTERFACES="eth0 eth1"
  ```

▼ DNS

DNS y BIND9: Apuntes para el Examen

¿Qué es DNS?

- **DNS** significa *Domain Name System* (Sistema de Nombres de Dominio).
- Es un sistema que traduce nombres de dominio legibles (como **www.ejemplo.com**) a direcciones IP numéricas (como **192.168.1.1**).
- Facilita la navegación en Internet al permitir utilizar nombres fáciles de recordar en lugar de números.

¿Cómo Funciona DNS?

- **Estructura Jerárquica**: DNS tiene una estructura en forma de árbol, comenzando desde la **raíz (.)**.
  - **Dominios de Nivel Superior (TLD)**: Como **.com**, **.org**, **.net**.
  - **Dominios de Segundo Nivel**: Como **ejemplo.com**.
  - **Subdominios**: Como **www.ejemplo.com**.
- **Resolución de Nombres**:
  - Cuando escribes una URL, tu dispositivo pregunta a un servidor DNS por la dirección IP correspondiente.
  - La consulta puede ser resuelta localmente o pasar por varios servidores DNS hasta encontrar la respuesta.

Tipos de Registros DNS

- **SOA (Start of Authority)**: Indica el inicio de la zona y contiene información administrativa.
- **A**: Asocia un nombre de dominio con una dirección IPv4.
- **AAAA**: Asocia un nombre de dominio con una dirección IPv6.
- **NS (Name Server)**: Indica los servidores DNS para la zona.
- **MX (Mail Exchange)**: Especifica los servidores de correo para el dominio.
- **CNAME (Canonical Name)**: Alias de otro nombre de dominio.
- **PTR (Pointer Record)**: Utilizado para resolución inversa (IP a nombre de dominio).

- **TXT**: Contiene texto arbitrario, a menudo usado para verificaciones y seguridad.

Instalación de un Servidor DNS con BIND9

- **BIND9** es uno de los servidores DNS más populares.
- **Instalación en sistemas Debian/Ubuntu**:

```
$sudo apt-get update
$sudo apt-get install bind9 bind9-doc dnsutils
```

Configuración Básica de BIND9

- **Archivos de Configuración**: Se encuentran en `/etc/bind/`.

Archivo `named.conf`

- Es el archivo principal de configuración.
- Incluye referencias a otros archivos como:
  - `named.conf.options`: Configuraciones globales.
  - `named.conf.local`: Zonas definidas por el usuario.

Configuración de Opciones Globales

- Edita `/etc/bind/named.conf.options` para ajustar opciones como:

```
options {
  directory "/var/cache/bind";

  forwarders {
    8.8.8.8; 8.8.4.4;  # Servidores DNS a los que ree
nviar consultas
  };

  allow-query { any; };  # Permitir consultas desde c
ualquier lugar
```

```
  recursion yes;  # Habilitar resolución recursiva
};
```

Definir Zonas DNS

- **Zona**: Un conjunto de registros DNS para un dominio.

Añadir una Zona Directa (Resolución de Nombre a IP)

- Edita `/etc/bind/named.conf.local`:

```
zone "ejemplo.com" {
  type master;
  file "/etc/bind/db.ejemplo.com";
  allow-transfer { 192.168.1.2; };  # Servidor secund
ario
};
```

- Crea el archivo de zona `/etc/bind/db.ejemplo.com`:

```
$TTL 604800
@ IN SOA ns1.ejemplo.com. admin.ejemplo.com. (
  2023101001 ; Número de serie (YYYYMMDDXX)
  7200       ; Refresh
  3600       ; Retry
  1209600    ; Expire
  86400      ; Negative Cache TTL
)

; Servidores de nombres
@   IN NS ns1.ejemplo.com.
@   IN NS ns2.ejemplo.com.

; Registros A
ns1 IN A 192.168.1.1
ns2 IN A 192.168.1.2
www IN A 192.168.1.3

; Registros MX
```

```
@   IN MX 10 mail.ejemplo.com.
mail IN A 192.168.1.4
```

Añadir una Zona Inversa (Resolución de IP a Nombre)

- Edita `/etc/bind/named.conf.local`:

```
zone "1.168.192.in-addr.arpa" {
  type master;
  file "/etc/bind/db.192.168.1";
  allow-transfer { 192.168.1.2; };
};
```

- Crea el archivo de zona inversa `/etc/bind/db.192.168.1`:

```
$TTL 604800
@ IN SOA ns1.ejemplo.com. admin.ejemplo.com. (
  2023101001 ; Número de serie
  7200
  3600
  1209600
  86400
)

; Servidores de nombres
@   IN NS ns1.ejemplo.com.
@   IN NS ns2.ejemplo.com.

; Registros PTR
1 IN PTR ns1.ejemplo.com.
2 IN PTR ns2.ejemplo.com.
3 IN PTR www.ejemplo.com.
4 IN PTR mail.ejemplo.com.
```

Parámetros Importantes en los Archivos de Zona

- **$TTL**: Tiempo de vida predeterminado de los registros.
- **@**: Representa el dominio raíz de la zona.

- **SOA Record**: Contiene información básica de la zona.
  - **Serial**: Número que debe incrementarse cada vez que se modifica el archivo.
  - **Refresh**: Tiempo que espera el secundario para refrescar la zona.
  - **Retry**: Tiempo que espera para reintentar si falla la actualización.
  - **Expire**: Tiempo tras el cual el secundario considera la información no válida.
  - **Negative Cache TTL**: Tiempo que se cachean las respuestas negativas.

Comprobación de la Configuración

- **Verificar el archivo de configuración principal**:

```
named-checkconf
```

- **Verificar archivos de zona**:

```
named-checkzone ejemplo.com /etc/bind/db.ejemplo.com
named-checkzone 1.168.192.in-addr.arpa /etc/bind/db.1
92.168.1
```

Gestión del Servicio BIND9

- **Iniciar el servicio**:

```
systemctl start bind9
```

- **Detener el servicio**:

```
sudo systemctl stop bind9
```

- **Reiniciar el servicio**:

```
sudo systemctl restart bind9
```

- **Ver el estado del servicio**:

```
sudo systemctl status bind9
```

Herramientas para Probar el DNS

- **nslookup**: Herramienta sencilla para consultas DNS.

```
nslookup www.ejemplo.com
```

- **dig**: Herramienta avanzada para consultas DNS.

```
dig www.ejemplo.com
dig -x 192.168.1.3  # Consulta inversa
```

Servidores DNS Secundarios y Transferencia de Zonas

- **Servidor Secundario**: Copia los datos del servidor primario para proporcionar redundancia.
- **Configuración en el Servidor Secundario**:
  - En `/etc/bind/named.conf.local`:

```
zone "ejemplo.com" {
  type slave;
  file "/var/cache/bind/db.ejemplo.com";
  masters { 192.168.1.1; };  # IP del servidor pri
mario
};

zone "1.168.192.in-addr.arpa" {
  type slave;
  file "/var/cache/bind/db.192.168.1";
  masters { 192.168.1.1; };
};
```

▼ GIT , GITHUB & DOCUMENTACIÓN

¿Qué es Git?

- **Git** es un **sistema de control de versiones**.
- Permite **rastrear cambios** en archivos y coordinar el trabajo entre varias personas.
- Ayuda a **volver a versiones anteriores** del código si es necesario.

Comandos Básicos de Git

1. Iniciar un Repositorio

- **Comando**: `git init`
- **Función**: Crea un nuevo repositorio de Git en tu carpeta actual.

```
git init
```

2. Añadir Archivos al Seguimiento

- **Comando**: `git add`
- **Función**: Comienza a rastrear archivos nuevos o cambios en archivos existentes.

```
git add archivo.txt      # Añade un archivo específic
o
git add .                # Añade todos los archivos e
n el directorio actual
```

3. Confirmar Cambios (Commit)

- **Comando**: `git commit`
- **Función**: Guarda tus cambios en el repositorio con un mensaje descriptivo.

```
git commit -m "Mensaje que describe los cambios"
```

4. Enviar Cambios al Repositorio Remoto (Push)

- **Comando**: `git push`
- **Función**: Envía tus commits al repositorio remoto (como GitHub).

```
git push origin main     # Envía cambios a la rama 'm
ain' en el remoto 'origin'
```

5. Actualizar Repositorio Local (Pull)

- **Comando**: `git pull`
- **Función**: Trae y fusiona cambios desde el repositorio remoto a tu repositorio local.

```
git pull origin main     # Trae cambios de la rama 'm
ain' del remoto 'origin'
```

6. Trabajar con Ramas

- **Crear una Rama Nueva**:

```
git branch nombre_rama
```

- **Cambiar de Rama**:

```
git checkout nombre_rama
```

- **Crear y Cambiar a una Rama Nueva al Mismo Tiempo**:

```
git checkout -b nombre_rama
```

- **Fusionar Ramas**:
  - Cambia a la rama donde quieres fusionar y ejecuta:

```
git merge nombre_rama
```

7. Ver el Estado del Repositorio

- **Comando**: `git status`
- **Función**: Muestra el estado de los archivos (modificados, preparados, sin rastrear).

```
git status
```

8. Ver el Historial de Cambios

- **Comando**: `git log`
- **Función**: Muestra el historial de commits.

```
git log
```

¿Qué es GitHub?

- **GitHub** es una plataforma en línea que **aloja repositorios Git**.
- Permite colaborar con otros desarrolladores.
- Ofrece características como:
  - **Control de versiones**.
  - **Seguimiento de problemas** (Issues).
  - **Solicitudes de extracción** (Pull Requests).
  - **Wiki y documentación**.

1. Conectar Repositorio Local con GitHub

- En tu repositorio local, añade el repositorio remoto:

```
git remote add origin https://github.com/tu_usuario/tu_repositorio.git
```

2. Enviar Cambios a GitHub

- Después de hacer commits:

```
git push -u origin main
```

3. Clonar un Repositorio Existente

- Para obtener una copia local de un repositorio:

```
git clone https://github.com/usuario/repositorio.git
```

Documentar el Código

La documentación es esencial para entender y mantener el código.

Etiquetas Estandarizadas:

- `@param`: Describe los parámetros.
- `@return`: Describe el valor de retorno.
- `@throws`: Indica excepciones que puede lanzar.

Herramientas para Documentar

1. phpDocumentor (Para PHP)

- **Función**: Genera documentación a partir de comentarios en el código PHP.
- **Instalación**:

```
composer global require phpdocumentor/phpdocumentor
```

- **Ejemplo de Comentario PHPDoc**:

```
/**
 * Suma dos números.
 *
 * @param int $a El primer número.
 * @param int $b El segundo número.
 * @return int La suma de $a y $b.
 */
function sumar($a, $b) {
    return $a + $b;
}
```

- **Uso**:
  - Añade comentarios PHPDoc en tu código.
  - Ejecuta:

```
phpdoc
```

2. JSDoc (Para JavaScript)

- **Función**: Genera documentación para proyectos JavaScript.
- **Instalación**:

```
npm install -g jsdoc
```

- **Ejemplo de Comentario JSDoc**:

```
/**
 * Multiplica dos números.
 * @param {number} a - El primer número.
 * @param {number} b - El segundo número.
 * @returns {number} El resultado de la multiplicació
n.
 */
function multiplicar(a, b) {
    return a * b;
}
```

- **Uso**:
  - Añade comentarios JSDoc en tu código.
  - Ejecuta:

```
jsdoc archivo.js
```