



Proxy y Servidor Web

INFRAESTRUCTURA WEB Y PROTOCOLOS DE COMUNICACIÓN

 Arquitectura de Servicios Web

¿Qué es un Proxy?

Servidor Proxy Web

Un **proxy** actúa como intermediario entre los clientes y los servidores de destino, permitiendo gestionar, filtrar y optimizar el tráfico de red.

Caching

Almacena respuestas para acelerar futuras solicitudes

Filtrado

Controla el acceso a contenido específico

Anonimato

Oculto la identidad del cliente original

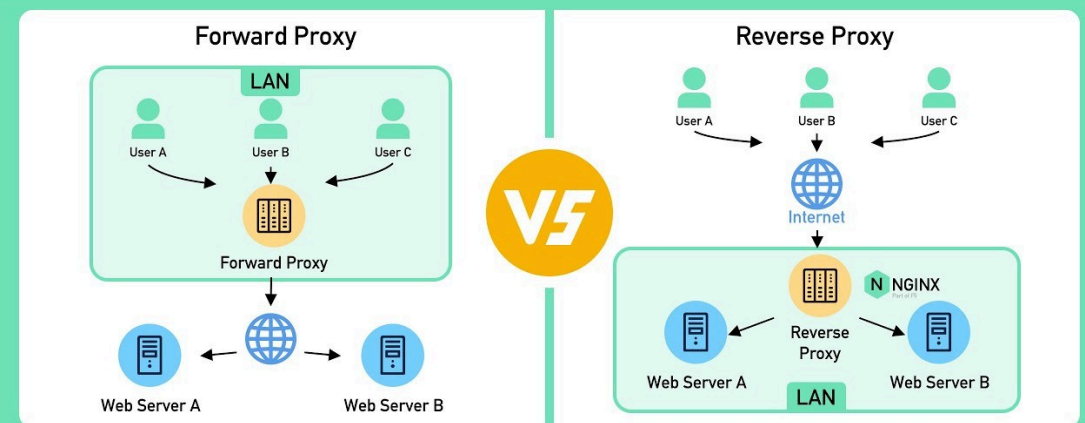
Control de Acceso

Gestiona autenticación y permisos

Balanceo de Carga

Distribuye el tráfico entre múltiples servidores

| Forward Proxy vs. Reverse Proxy



Herramientas de Proxy en Linux



Squid

Proxy de Caching y Web

Proxy de caching web con alto rendimiento, filtrado de contenido, control de acceso y soporte para múltiples protocolos.



Varnish

Acelerador HTTP

Caching de alto rendimiento para acelerar aplicaciones web, con reducción significativa de carga en servidores backend.



HAProxy

Balanceador de Carga

Balanceador de carga TCP/HTTP de alto rendimiento con detección de salud de servidores y conmutación por fallos.



Privoxy

Proxy de Filtrado

Proxy de filtrado y privacidad con capacidad de modificar encabezados HTTP, cookies y contenido web.

¿Qué es un Servidor Web?

Definición

Software que **almacena, procesa y entrega contenido web** a través de protocolos HTTP/HTTPS.

Actúa como intermediario entre el cliente (navegador) y los recursos solicitados.



Recibir Solicitudes

Escucha peticiones
HTTP/HTTPS



Procesar

Interpreta y ejecuta solicitudes



Servir Contenido

Envía respuestas al cliente



Múltiples Conexiones

Maneja concurrencia de usuarios



Gestión de Seguridad

Implementa SSL/TLS, autenticación y control de acceso



Web servers

cloudspinx.com

cloudspinx.com

HTTP vs Protocolos Anteriores

Características de HTTP



Sin Estado

Cada solicitud es independiente



Texto Plano

Formato legible por humanos



Cliente-Servidor

Arquitectura de dos capas



Portabilidad

Funciona en cualquier plataforma



Flexibilidad

Métodos, encabezados y contenido extensibles

Protocolos Anteriores

Gopher

Menús jerárquicos

FTP

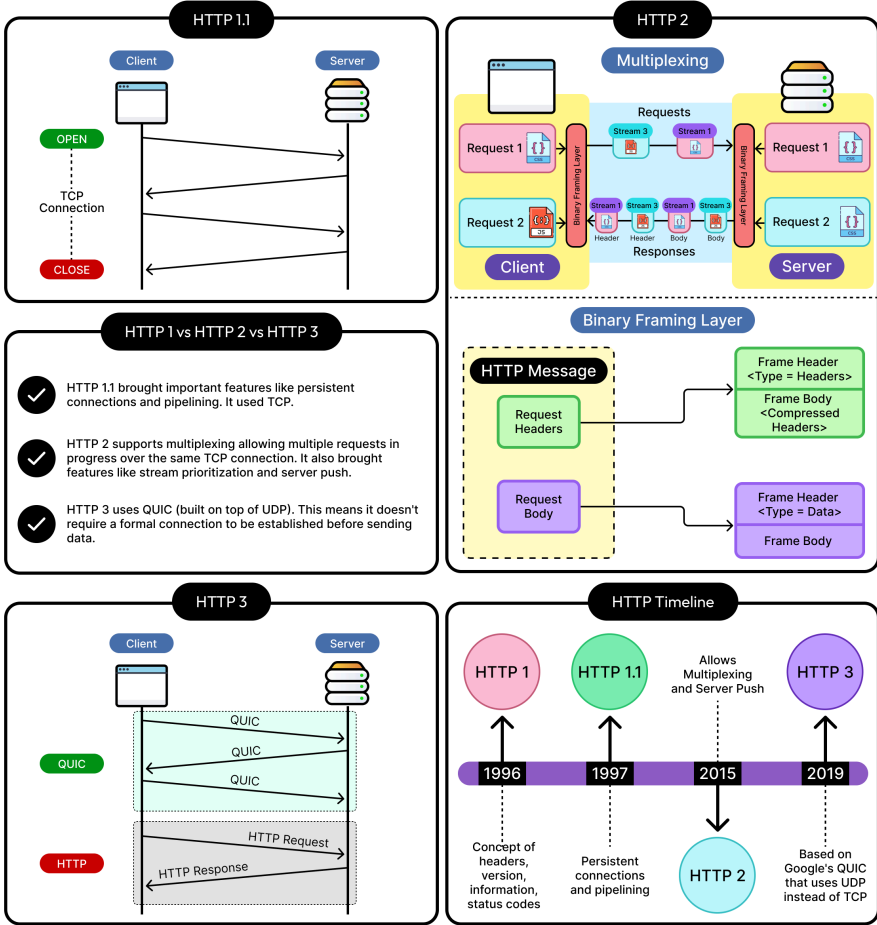
Transferencia de archivos

Telnet

Acceso remoto

A Deep Dive Into HTTP: From HTTP/1.1 to HTTP/3

ByteByteGo



Métodos de HTTP



GET

Recuperar recursos del servidor

Ej: GET /index.html



POST

Enviar datos al servidor

Ej: POST /login



PUT

Actualizar recursos completos

Ej: PUT /user/123



DELETE

Eliminar recursos del servidor

Ej: DELETE /user/123



HEAD

Obtener solo cabeceras

Ej: HEAD /image.jpg



OPTIONS

Comunicar opciones permitidas

Ej: OPTIONS /api



PATCH

Modificar recursos parcialmente

Ej: PATCH /user/123



CONNECT

Establecer túneles

Ej: CONNECT proxy:8080



TRACE

Diagnosticar rutas

Ej: TRACE /test

HTTPS: Uso Actual y Razones

¿Por qué HTTPS es Esencial Hoy?

Cifrado TLS/SSL

Protege datos en tránsito

Autenticación

Verifica identidad del servidor

Protección de Datos

Evita intercepciones

Requisito Navegadores

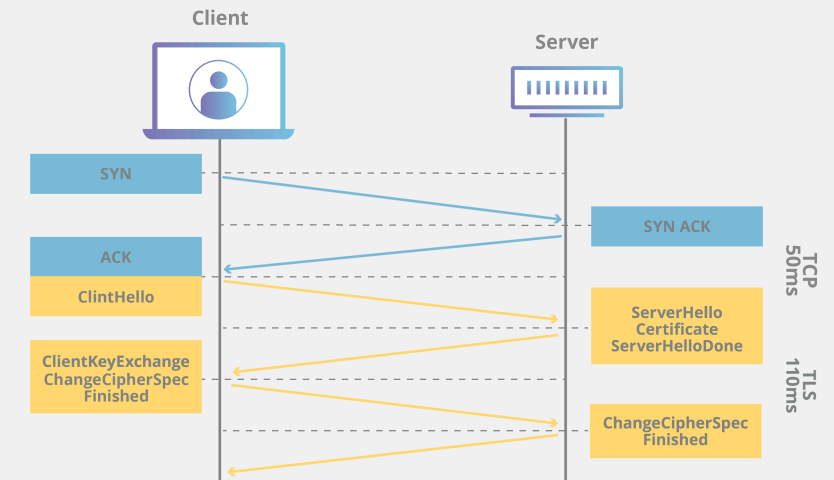
Estándar web moderno

Beneficios SEO

Mejor posicionamiento en resultados de búsqueda

Proceso TLS/SSL Handshake

El cliente y servidor intercambian certificados y claves para establecer un canal seguro cifrado



Servidores Web en Linux: Apache y Nginx



Apache HTTP Server

- ✓ Arquitectura modular
- ✓ Soporte .htaccess
- ✓ Integración PHP nativa
- ✓ Base de código maduro



Nginx

- ✓ Alto rendimiento
- ✓ Conexiones asíncronas
- ✓ Proxy reverso eficiente
- ✓ Menor consumo de recursos

Casos de Uso Apache

- Hosting compartido
- Proyectos PHP
- Necesidad de .htaccess
- Compatibilidad heredada

Casos de Uso Nginx

- Alto tráfico
- Proxy reverso
- Servicios web modernos
- Balanceo de carga

NGINX
VS
Apache



bloggerpilot.com

Configuración de Apache

Ficheros Principales

`/etc/apache2/apache2.conf`

Configuración global (Debian/Ubuntu)

`/etc/httpd/conf/httpd.conf`

Configuración global (RHEL/CentOS)

`/etc/apache2/sites-available/`

VirtualHost habilitados

Comandos Básicos

▶ `systemctl start apache2`

■ `systemctl stop apache2`

↻ `systemctl restart apache2`

🔄 `systemctl reload apache2`

📄 `systemctl status apache2`

Gestión de Módulos

`a2enmod nombre_modulo`

`a2dismod nombre_modulo`

Estructura de Configuración

VirtualHost

Sitios virtuales por IP/Puerto

Directory

Configuración por directorio

Location

Configuración por URL

Módulos Principales

`mod_ssl`

`mod_rewrite`

`mod_php`

Sitios Web Virtuales (FQDN)

VirtualHost FQDN

Configuración basada en nombre de dominio

Múltiples Dominios

Varios sitios en un solo servidor

Estructura del Bloque VirtualHost

```
<VirtualHost *:80>
  ServerName www.ejemplo.com
  DocumentRoot /var/www/ejemplo
  ServerAlias ejemplo.com
  ErrorLog ${APACHE_LOG_DIR}/error.log
</VirtualHost>
```

ServerName

Nombre de dominio

DocumentRoot

Directorio del sitio

ServerAlias

Dominios alternos

Ejemplo Práctico

```
/etc/apache2/sites-
available/ejemplo.com.conf
```

→ Habilitar: a2ensite ejemplo.com

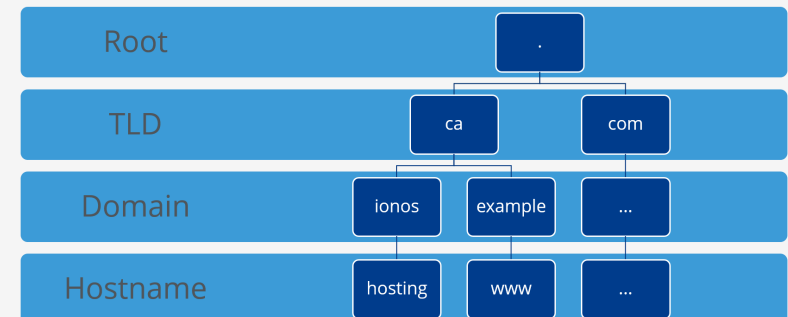
```
/etc/apache2/sites-
available/tienda.com.conf
```

→ Habilitar: a2ensite tienda.com

Reiniciar: `systemctl reload apache2`

Ambos sitios activos simultáneamente

FQDN



Configuración de Nginx

Ficheros Principales

`/etc/nginx/nginx.conf`

Configuración global

`/etc/nginx/sites-available/`

Archivos de sitios disponibles

`/etc/nginx/sites-enabled/`

Enlaces simbólicos activos

Comandos Básicos

- ▶ `systemctl start nginx`
- `systemctl stop nginx`
- 🔄 `systemctl restart nginx`
- 🔄 `systemctl reload nginx`
- 📄 `systemctl status nginx`

Gestión de Sitios

`ln -s sites-available/ sitio sites-enabled/`

Estructura server

 **listen**
Puerto de escucha (80, 443)

 **server_name**
Nombre del dominio

 **root**
Directorio raíz del sitio

Proxy Reverso

```
location / {  
    proxy_pass http://backend;  
}
```

Preguntas de Test

1 ¿Cuál es la función principal de un servidor proxy web?

- a) Almacenar datos de bases de datos
- b) Actuar como intermediario entre cliente y servidor
- c) Compilar código del lado del servidor
- d) Gestionar bases de datos relacionales

2 ¿Qué herramienta de proxy en Linux se especializa en caching de alto rendimiento HTTP?

- a) Squid
- b) Varnish
- c) HAProxy
- d) Privoxy

3 ¿Qué directiva en Apache define el directorio raíz de un VirtualHost?

- a) Requiere conexión persistente
 - a) Directory
- b) Cada solicitud es independiente
 - b) DocumentRoot
- c) Almacena estado del cliente
 - c) ServerRoot
- d) Mantiene sesión activa
 - d) HomeDir

4 ¿Qué método HTTP se utiliza para actualizar un recurso parcialmente?

- a) PUT
- b) POST
- c) PATCH
- d) UPDATE

5 ¿Cuál es la principal ventaja de usar HTTPS sobre HTTP?

- a) Mayor velocidad de carga
- b) Cifrado y autenticación
- c) Menor consumo de recursos
- d) Soporte para más protocolos

8 ¿En qué ruta se encuentra el archivo de configuración principal de Nginx?

- a) /etc/nginx/nginx.conf
- a) Apache
- b) /etc/nginx/main.conf
- b) Nginx
- c) /etc/nginx/config/nginx.conf
- c) IIS
- d) /etc/nginx/nginx/main.conf
- d) Lighttpd