

UD2. Administración de Servidores Web

Apache / Nginx

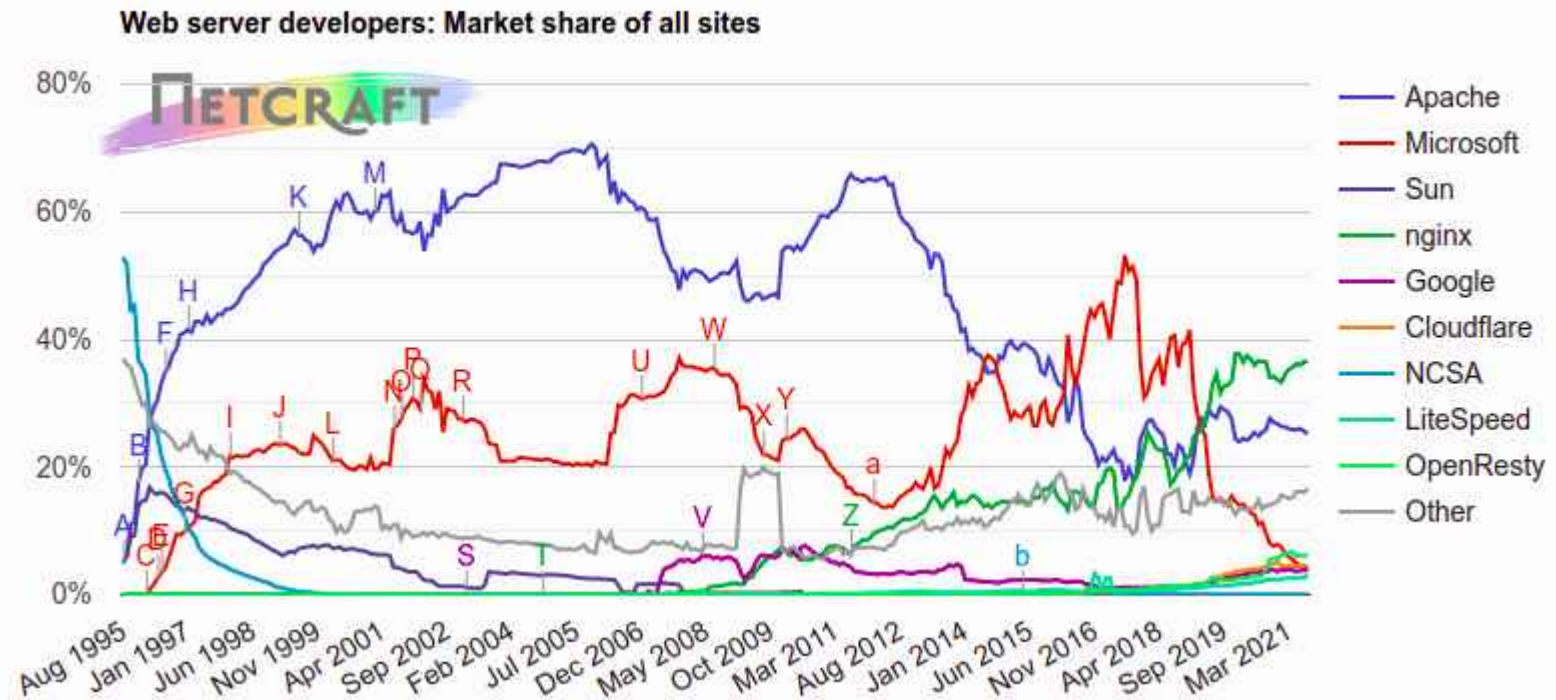


Despliegue de Aplicaciones Web
2º DAW

ÍNDEX

- INTRODUCCIÓN
- ADMINISTRACIÓN DEL SERVICIO
- HOST VIRTUALES
- MONITORIZACIÓN
- MÓDULOS

1. Introducción



1. Introducción

Servidor HTTP Apache

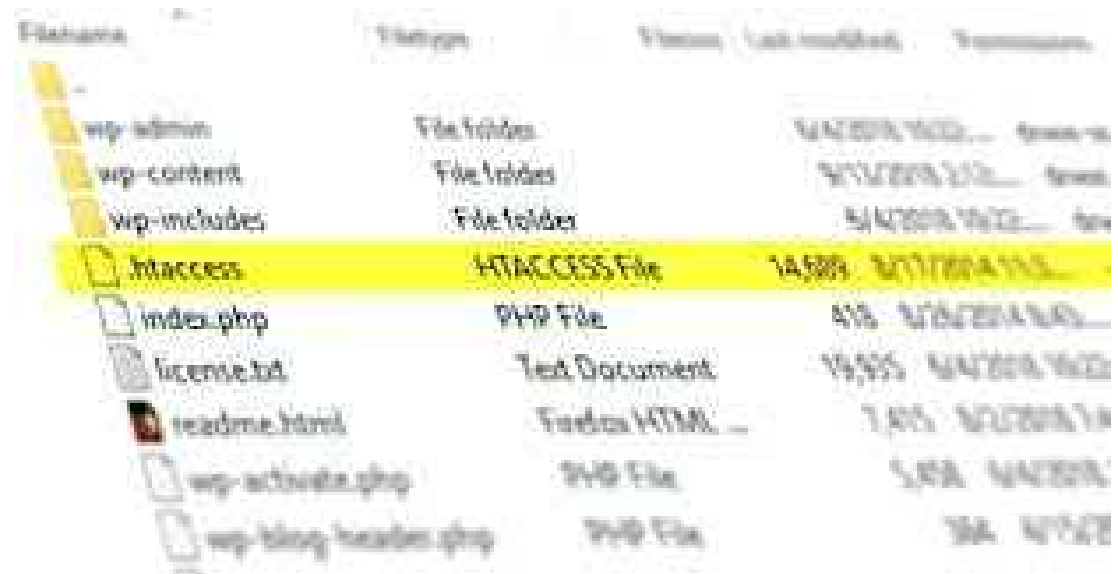
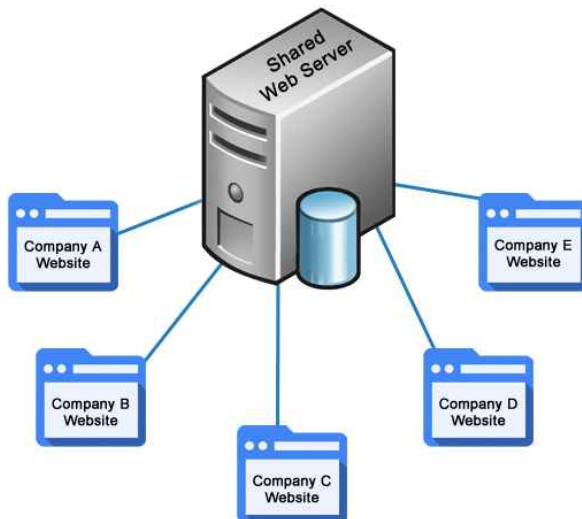
- El **servidor HTTP Apache** es un **servidor web HTTP** de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh.
- Presenta una arquitectura **modular**.
- Soporte para **HTTP/1** y **HTTP/2**
- Amplia aceptación → **Gran comunidad y documentación**



1. Introducción

Servidor HTTP Apache

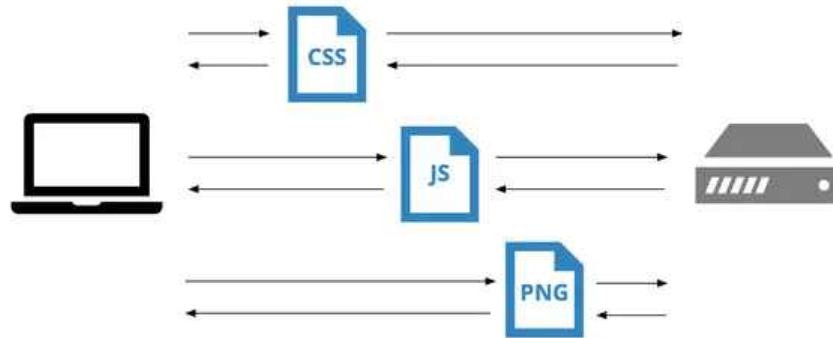
- Permite configuración **descentralizada** a través de fichero ***.htaccess***



1. Introducción

Servidor HTTP Apache

- Conexiones **keep-alive**: utilizadas para no tener que abrir una conexión diferente para cada elemento a enviar



```
curl -I https://www.keycdn.com
HTTP/1.1 200 OK
Server: nginx
Date: Mon, 31 Oct 2016 19:37:22 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 20677
Connection: keep-alive
Vary: Accept-Encoding
Set-Cookie: keycdn=v2bboirnhapd98884fhi66s0v80m71f0ohj4qh727bqdmgb0v6i
Expires: Mon, 31 Oct 2016 19:37:21 GMT
Strict-Transport-Security: max-age=31536000; includeSubdomains; preloa
```



1. Introducción

Servidor Nginx

- Primera **publicación** en **2004**
- Servidor de **código abierto ligero**
- Da solución al problema de los **C10k** (10.000 conexiones concurrentes).
- Inicialmente pensado para servir **contenidos estáticos**.
- El **contenido dinámico** es ejecutado solo a través de **procesos externos** (FastCGI...)
- **No** permite **configuración descentralizada**



1. Introducción

Servidor Nginx

- Es utilizado tanto como **Web Server** como de Proxy (**Reverse Proxy**)
- Presenta una arquitectura orientada a eventos
 - Cada hilo → miles de conexiones
 - Menor consumo de recursos
- Buena documentación → nginx wiki
- Presenta una estructura ***modular***
- Hasta **2.5 veces** más rápido en servidor contenido estático que Apache.

1. Introducción

- Otros **servidores web**
 - **IIS**: servidor web privativo propiedad de Microsoft específicamente para su plataforma de sistemas operativos Windows
 - **Lighthttp**: software libre y escrito **en C** se distribuye bajo la licencia BSD. Funciona en GNU/Linux y UNIX de forma oficial.
 - **Caddy**: uno de los primeros servidores web que ofrece HTTPS de forma automática cada vez que activas un host, valida tu sitio contra los servidores de **Let's Encrypt**.
 - **Google Web Server**: servidor web propiedad de Google que utiliza para su infraestructura

LIGHTTPD



2. Host Virtuales

- Permiten servir **diferentes sitios** web dentro del mismo **servidor http**. Cada virtualHost podrá definir su propia configuración.
 - ***Puertos de escucha***
 - ***Ficheros de log de acceso y error***
 - ***DocumentRoot***
 - ***ServerName*** (*Nombre de dominio que atenderá*)



Welcome!

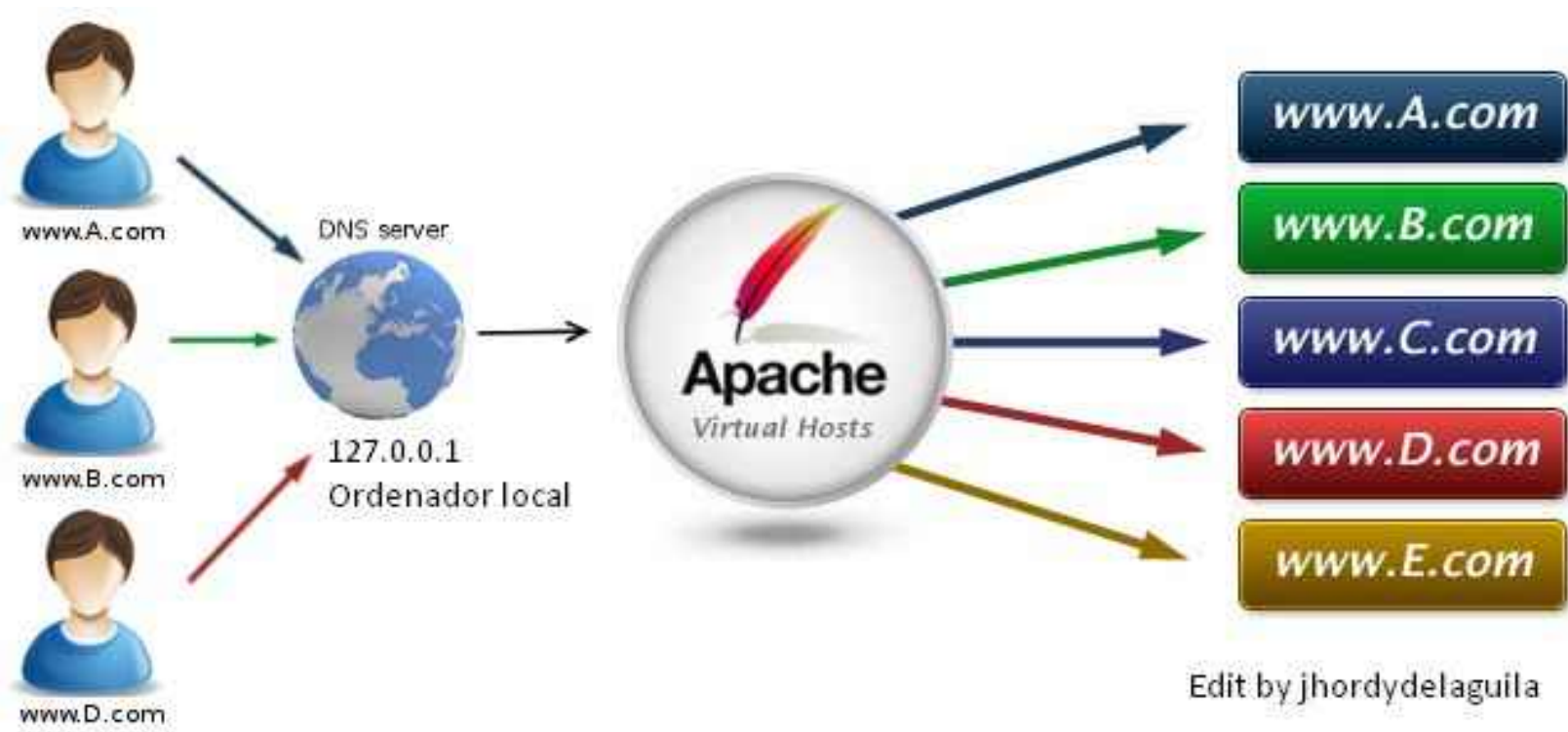
This is Example1.com



Welcome!

This is Example2.com

2. Host Virtuales



2. Host Virtuales

- Archivos de configuración
 - **/sites-available:** Contiene archivos de configuración para diversos hosts virtuales
 - `000-default.conf`
 - **/sites-enabled:** contiene enlaces simbólicos de aquellos sitios presentes en sites-available que el administrador quiere activar.
 - **a2ensite** nombre_del_sitio_web
 - **a2dissite** nombre_del_sitio_web

2. Host Virtuales

- Es importante definir cada host virtual en su propio **fichero de configuración**

```
/etc/apache2/  
|-- apache2.conf  
|   |-- ports.conf  
|-- mods-enabled  
|   |-- *.load  
|   |-- *.conf  
|-- conf-enabled  
|   |-- *.conf  
|-- sites-enabled  
|   |-- *.conf 001-www.severoochoa.es.conf
```

2. Host Virtuales

- Ejemplo Apache

```
<VirtualHost *:80>
  #Establece el dominio que atenderá el host virtual
  #ServerName www.example.com
  #Establece a quien debemos consultar problemas con servidor
  ServerAdmin webmaster@localhost
  #Path de la aplicación
  DocumentRoot /var/www/html
  # Ficheros de log y accesos al servidor,
  ErrorLog ${APACHE_LOG_DIR}/error.log
  CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

3. Monitorización

- Las directivas **ErrorLog** y **AccessLog** indican la ubicación del fichero indican la **ubicación del fichero** de registro de **errores** y **accesos** al servidor
- Es Importante **separar los registros** de los distintos hosts virtuales y/o aplicaciones que tengamos para su posterior análisis.
 - Estadísticas, accesos indebidos, ataques
- Debemos tener en cuenta el tamaño de los mismos → mantenimiento (Ej. **log-rotate**)

4. Monitorización

```
$ tail -f /var/log/apache2/access.log
```

```
alecogi@docker:~$ tail -f /var/log/apache2/access.log
192.168.1.102 - - [23/Sep/2019:21:29:46 +0000] "-" 408 0 "-" "-"
192.168.1.102 - - [23/Sep/2019:21:30:17 +0000] "GET /~alecogi HTTP/1.1" 301 584 "-" "Mozilla/5.0 (Macintosh; I
WebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.132 Safari/537.36"
192.168.1.102 - - [23/Sep/2019:21:30:17 +0000] "GET /~alecogi/ HTTP/1.1" 200 296 "-" "Mozilla/5.0 (Macintosh;
eWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.132 Safari/537.36"
192.168.1.102 - - [23/Sep/2019:21:30:18 +0000] "GET /~alecogi/ HTTP/1.1" 304 178 "-" "Mozilla/5.0 (Macintosh;
eWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.132 Safari/537.36"
192.168.1.102 - - [23/Sep/2019:21:30:19 +0000] "GET /~alecogi/ HTTP/1.1" 304 178 "-" "Mozilla/5.0 (Macintosh;
eWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.132 Safari/537.36"
192.168.1.102 - - [23/Sep/2019:21:30:19 +0000] "GET /~alecogi/ HTTP/1.1" 304 178 "-" "Mozilla/5.0 (Macintosh;
eWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.132 Safari/537.36"
127.0.0.1 - - [23/Sep/2019:23:37:25 +0000] "GET /index.html HTTP/1.1" 200 3084 "-" "Wget/1.19.4 (linux-gnu)"
127.0.0.1 - - [23/Sep/2019:23:37:44 +0000] "GET /index.html HTTP/1.1" 200 11229 "-" "Wget/1.19.4 (linux-gnu)"
127.0.0.1 - - [23/Sep/2019:23:42:38 +0000] "GET /index.html HTTP/1.1" 200 3084 "-" "Wget/1.19.4 (linux-gnu)"
127.0.0.1 - - [23/Sep/2019:23:53:11 +0000] "GET /index.html HTTP/1.1" 200 3084 "-" "Wget/1.19.4 (linux-gnu)"
```


5. Módulos

- La arquitectura del servidor **Apache** es modular.
- El servidor consta de una serie de funcionalidad **básica** **que** se complementa mediante la adición de módulos
 - **Autenticación:** mod_dav, mod_auth_ldap
 - **Proxy:** mod_http_proxy, mod_proxy_fcgi
 - **Compresión:** mod_deflate
 - **Cifrado de conexiones:** mod_ssl

<https://httpd.apache.org/docs/current/es/mod/>

5. Módulos

- **mods_available:** este directorio contiene una serie de archivos **.load** y **.conf**.
 - Los archivos **.load** tienen directivas de configuración Apache para la carga del módulo en cuestión.
 - El correspondiente **.conf**, contiene las directivas de configuración necesarias para la utilización del módulo



5. Módulos

- **Estructura de directorios**
 - **mods_enabled**: enlaces simbólicos con los **.load** presentes en **mods_available**. También para **.conf** si existe.
 - **a2enmod** [[-q|--quiet] module]
 - **a2dismod** [[-q|--quiet] module]

a2enmod imagemap

6. Administración del servicio

- **Administración del servicio**
 - *sudo systemctl stop apache2*
 - *sudo systemctl start apache2*
 - *sudo systemctl restart apache2*
 - *sudo systemctl reload apache2*
 - *sudo systemctl disable apache2* #inicio del sistema
 - *sudo systemctl enable apache2* #inicio del sistema
 - *sudo systemctl apache2 status*

6. Administración del servicio

- Información y estructura de directorios

- `$ sudo apache2ctl -V`

```
Server version: Apache/2.4.29 (Ubuntu)
Server built: 2019-09-16T12:58:48
Server's Module Magic Number: 20120211:68
Server loaded: APR 1.6.3, APR-UTIL 1.6.1
Compiled using: APR 1.6.3, APR-UTIL 1.6.1
Architecture: 64-bit
Server MPM: event
Server compiled with....
-D APR_HAS_SENDFILE
-D APR_HAS_MMAP
-D APR_HAVE_IPV6 (IPv4-mapped addresses enabled)
-D APR_USE_SYSVSEM_SERIALIZE
-D APR_USE_PTHREAD_SERIALIZE
-D SINGLE_LISTEN_UNSERIALIZED_ACCEPT
-D APR_HAS_OTHER_CHILD
-D AP_HAVE_RELIABLE_PIPED_LOGS
-D DYNAMIC_MODULE_LIMIT=256
-D HTTPD_ROOT="/etc/apache2"
-D SUEXEC_BIN="/usr/lib/apache2/suexec"
-D DEFAULT_PIDLOG="/var/run/apache2.pid"
-D DEFAULT_SCOREBOARD="logs/apache_runtime_status"
-D DEFAULT_ERRORLOG="logs/error_log"
-D AP_TYPES_CONFIG_FILE="mime.types"
-D SERVER_CONFIG_FILE="apache2.conf"
```

6. Administración del servicio

- Estructura de directorios
 - *Apache* → */etc/apache/*

```
[alecogi@ddaw:~$ ls /etc/apache2/
apache2.conf  conf-enabled  magic          mods-enabled  sites-available
conf-available  envvars      mods-available ports.conf    sites-enabled
alecogi@ddaw:~$
```

6. Administración del servicio

- **Archivos de configuración general**
 - **apache2.conf**: archivo de configuración principal.
 - **/conf_available/**: tiene archivos de configuración asociados a módulos
 - **/conf_enabled/**: contiene los módulos de configuración activos.
 - Son **enlaces simbólicos** a los archivos de conf available.

```
a2enconf security
a2disconf security
```

6. Administración del servicio

- **Archivos de configuración general**
 - **ports.conf:** directivas de configuración que indican puertos y direcciones IP a escuchar.

```
Listen 80
<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>
```


6. Administración del servicio

- Eso es todo...de momento :-)