

UT5: El servicio http: **Apache** **2.4.**

Servicios en Red - 2º Curso CFGM SMR

Características generales.



▶ Apache:

- ▶ Aplicación “estrella” del mundo Linux.
- ▶ Es el servidor web más utilizado en el mundo en estos momentos.
- ▶ Algunas de sus características:
 - ▶ Es estable.
 - ▶ Multiplataforma: Windows, Linux, MacOS, ...
 - ▶ Altamente configurable.
 - ▶ Modular.
 - ▶ Actualmente se utiliza la versión 2.4 (Ubuntu 15.04).
 - ▶ La versión anterior era la 2.2 y algunas de las directivas (Require funcionaban de diferente manera).



Instalación y arranque del servidor.

- ▶ El nombre del paquete que debemos es **apache2**.
 - ▶ `sudo apt-get update`
 - ▶ `sudo apt-get install apache2`
- ▶ Para comprobar que se ha instalado nos conectaremos al host local <http://localhost>.
- ▶ Para arrancar, parar y comprobar el estado del servicio:
 - ▶ `sudo service apache2 start`
 - ▶ `sudo service apache2 stop`
 - ▶ `sudo service apache2 status`
 - ▶ `sudo service apache2 reload` → únicamente relee los archivos de configuración



Ficheros de configuración.

- ▶ Apache se configura poniendo directivas en archivos de configuración. El principal se llama `apache2.conf` o `httpd.conf`.
 - ▶ Además pueden existir otros archivos que han de indicarse en `apache2.conf` mediante la directiva `include`.
 - ▶ El directorio de configuración por defecto es:
 - ▶ **/etc/apache2**
 - ▶ Ficheros.
 - ▶ **Apache2.conf:** fichero de configuración genérico.
 - ▶ **Ports.conf:** puertos en los que escucha.
 - ▶ **Httpd.conf:** archivo vacío no usado en Ubuntu.
 - ▶ Directorios.
 - ▶ **Sites-available:** lista de archivos de configuración de los sitios preparados para ser usados.
 - ▶ **Sites-enabled:** enlaces simbólicos a los sitios (archivos de conf) de sites-available que están siendo usados..
 - ▶ **Mods-available:** lista de módulos disponibles.
-
- ▶ **Mods-enabled:** links simbólicos a módulos. Son los habilitados.
 - ▶ **Conf.d:** archivos de configuración de algunas aplicaciones.

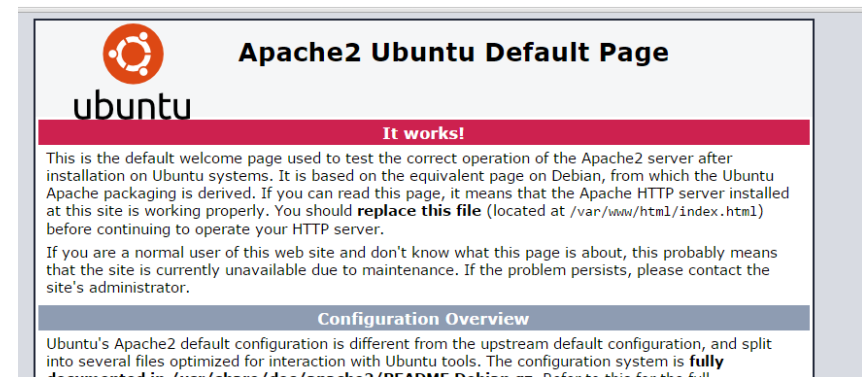
Despliegue web en Apache.

- ▶ Para poner en marcha un servidor web Apache, como todo los servidores web, hay que tener claros dos planos:
 - ▶ El contenido.
 - ▶ Se entiende por contenido, las páginas web que componen el sitio, es decir, los ficheros html, php, ...
 - ▶ La configuración del sitio.
 - ▶ Para que las páginas anteriores sean accedidas desde los clientes, debemos crear un fichero de configuración que especifique cuál será el comportamiento del sitio.

Nada más instalar apache, se crea y activa un sitio por defecto.

-El contenido está en **`/var/www/html/index.html`**.

-El fichero de configuración lo puedes ver en **`/etc/apache2/sites-available/000-default.conf`**



Despliegue web Apache: El contenido.

- ▶ **Antes de empezar a configurar el servidor debemos de tener el contenido. Para ello debemos saber y crear:**
 - ▶ **Cuál será el nombre a través del cuál será accesible el sitio.** Ejemplo: www.sri.com
 - ▶ **En qué puerto va a servir el sitio web anterior.** Ejemplo: puerto 80.
 - ▶ **Decidir y crear cuál será el directorio raíz en el cual se almacenará el sitio.** Ejemplo: /var/www/html/sri
 - ▶ **Crear la estructura de directorios.** Ejemplo: si va a tener subdirectorios o directorios virtuales.
 - ▶ **Crear y almacenar las páginas web que serán los documentos predeterminados.** Ejemplo: por ejemplo index.html.
 - ▶ **Tener claros qué permisos tendrán los clientes al acceder a las páginas en cada directorio.** Ejemplo: si se pedirá autenticación para algunas páginas del sitio.
 - ▶ **Tener claras que otras características de los directorios.** Ejemplo: mostrar contenidos de directorios, etc.
 - ▶ **Si algún subdirectorio podrá ser administrado por un usuario sin intervención del administrador.** Ejemplo: se dispone de ficheros .htaccess.
 - ▶ **Saber si se tratará de un sitio seguro.** Ejemplo: la web es accesible con <https://www.sri.com>.
-



Despliegue web Apache: Configuración.

- ▶ Para aplicar las especificaciones anteriores hay que crear un fichero de configuración que se almacenará en `/etc/apache2/sites-available` y cuyo nombre deberá acabar en `.conf`.

Ejemplo: `wwwasir2.conf`.

- ▶ Los ficheros de configuración están formados por directivas.
- ▶ Contendrán una directiva por línea.
- ▶ En las directivas no se hace distinción entre mayúsculas y minúsculas.
- ▶ Para chequear la sintaxis se pueden usar los siguientes comandos:
 - ▶ `apache2ctl configtest`
- ▶ El log de errores del servicio están en archivo:
 - ▶ `/var/log/apache2/error.log`
 - ▶ `tail -f /var/log/apache2/error.log`
- ▶ También existe un log de accesos al servidor que se almacena en:
 - ▶ `/var/log/apache2/access.log`



Configuración básica del servidor.

► Directivas

- Determinan la manera en que debe comportarse el servidor.
- Se incluyen dentro de los archivos de configuración.
- Dependiendo del entorno en el que se escriban afectarán al servidor general, a un servidor virtual, a un directorio, fichero, etc.
- Hay directivas que actúan como contenedores de otras.
- Podemos dividir las en tres grupos:
 - **Directivas configuración general:** definen los parámetros del servidor principal y las opciones por defecto para los hosts virtuales.
 - **Directivas de secciones:** configuran partes concretas del funcionamiento de apache.
 - **Optimización:** configuran los recursos y la eficacia de apache.

Configuración básica del servidor.

► Directivas de configuración general.

► ServerAdmin

- Dirección de correo del administrador de la web.
- ServerAdmin root@asir.net

► ServerName

- Especifica el nombre del host que se indicará en la
- ServerName www.asir.net

► DocumentRoot

- Carpeta raíz donde se ubica el servidor. Ojo, si hay Alias no es esta.
- DocumentRoot “/var/www/asir”



Configuración básica del servidor.

► Directivas de configuración general.

► DirectoryIndex

- Fichero a buscar en caso de que no se especifique ninguno.
- DirectoryIndex index.html indice.html red.html

► Alias

- Permite colocar contenido web fuera de DocumentRoot.
- Alias “url-path” “recurso”
- Alias /wiki /home/alumno/wiki
- Se utiliza para crear directorios virtuales.

OJO!!!! partir de Apache2.4 en el fichero de configuración /etc/apache2/apache2.conf, existe una directiva que impide el acceso al contenido bajo el directorio /. Por esa razón es necesario permitirlo explícitamente utilizando directivas Require explicadas más adelante.



Configuración básica del servidor.

► Directivas de configuración general.

- Permite colocar contenido web fuera de DocumentRoot.
- Alias “url-path” “recurso”
- Alias /wiki /home/alumno/wiki
- Se utiliza para crear directorios virtuales

OJO!!!! partir de Apache2.4 en el fichero de configuración /etc/apache2/apache2.conf, existe una directiva que impide el acceso al contenido bajo el directorio /. Por esa razón es necesario permitirlo explícitamente utilizando directivas Require explicadas más adelante.



Actividad.

- ▶ Instalar Apache.
 - ▶ Comprobar que funciona.
 - ▶ Comprobar los procesos y que el servidor se está ejecutando.
 - ▶ Acceder desde un cliente y comprobar en el archivo de accesos al servidor la IP de la máquina cliente.
 - ▶ ¿Cuáles son los sitios disponibles? ¿Cuál es el sitio que está activo en este momento?
 - ▶ Verificar los valores de las directivas en el archivo de configuración del sitio activo.
 - ▶ Verificar el funcionamiento de DirectoryIndex del sitio activo.
 - ▶ Para ello crear un documento html que muestre “Bienvenido al servidor web por defecto de TuNombre” y que sea ese el archivo que se muestre por defecto.
 - ▶ Prueba ahora a poner DirectoryIndex apuntando a un archivo que no existe e intenta explicar lo que ocurre.
 - ▶ Crear un directorio virtual apuntando al home del usuario que estás usando, almacena un fichero html y comprueba que puedes accederlo.
-

Configuración básica del servidor.

► Directivas de secciones:

► <Directory /directorio> ...</Directory>

Engloba una o más directivas de configuración que sólo se aplican al directorio y subdirectorios especificados. Dentro, en el lugar de los ... se usan las directivas siguientes:

- **DirectoryIndex:** Página de inicio
- **Options:** controla que características estarán disponibles para un directorio particular
 - **[+/-]indexes:** muestra el contenido del directorio si no encuentra un archivo índice.
 - **[+/-] FollowSymLinks:** permite seguir enlaces simbólicos dentro del directorio.
 - **[+/-] Multiviews:** sirve para no tener que especificar en la URL la extensión de un archivo situado en este directorio. Por ejemplo: si un servidor tiene en el directorio /var/www/misitio/web.html cuando un cliente escribe en la URL <http://www.loquesea.com/misitio/web>, si la opción multiviews está puesta para el directorio /var/www/misitio/ buscará algún archivo llamado web.* y mostrará el que encuentre. También serviría si pusiéramos DirectoryIndex indice y no especificáramos extensión.
 - Si ponemos **Options All** activará todas las opciones disponibles salvo Multiviews que hay que especificarla.
 - Por defecto se heredan los parámetros de los directorios superiores. Para eliminar una opción, deberemos poner un signo "-" delante. Ejemplo: -Indexes.



Configuración básica del servidor.

► Códigos de error.

► ErrorDocument

- Permite mostrar mensajes y páginas de error personalizadas.
- Si queremos que se muestre un texto concreto para un error concreto la sintaxis es la siguiente:

ErrorDocument XXX "Texto" donde XXX es el código de error correspondiente.

- Ejemplo: **ErrorDocument 404 "Página no encontrada en el sitio"**
- Si queremos que muestre un página:

ErrorDocument XXX Fichero, donde en fichero debes especificar el fichero incluyendo la ruta para accederlo.

- Ejemplo: **ErrorDocument 404 /404.html**, donde 404.html está en la raíz del sitio (si es /var/www, ese será el directorio donde debes colocar la página 404.html).



Configuración básica del servidor.

► Ficheros de log.

► Loglevel

- Nivel de “gravedad” de los mensajes de error que se guardarán en el fichero de log.
- Por defecto, warn (avisos).

► ErrorLog

- Especifica donde se va a encontrar el fichero de log de errores.
- Por defecto es `/var/log/apache2/error.log`.

► CustomLog

- En esta directiva especificaremos en qué fichero queremos registrar los accesos al sitio web que estamos configurando y el formato que se utilizará para guardar dichos registros.
- Por defecto, se guardan en `/var/log/apache2/access.log`, y el formato será combined (mirar la documentación para entender qué campos son los que se registran con este formato).



Host virtuales. Creación, configuración y utilización.

- ▶ Configurando hosts virtuales podemos alojar varios sitios web (varios dominios) en el mismo servidor (en la misma máquina), siendo uno totalmente independiente del otro.
- ▶ El sitio por defecto que viene configurado (000-default) en apache es un host virtual más.
- ▶ Existen varios tipos de host virtuales:
 - ▶ **Basados en nombre:** permite alojar varios nombres de host o varios dominios en una misma máquina. Todos los hosts que compartan la misma IP deben declararse mediante la directiva NameVirtualHost.
 - ▶ **Basados en puerto:** se responde de diferente manera si se accede a un puerto u otro.
 - ▶ **Basados en IP:** una máquina tiene varias direcciones IP y a cada una de ellas servirá un sitio web diferente.
- ▶ Por ser los más utilizados, nos centramos en los basados en nombre.



Host virtuales. Creación, configuración y utilización.

- ▶ Se trata de servir desde la misma máquina, en la misma IP y a través del mismo puerto, dos sitios web, por ejemplo: `www.web1.com` y `www.web2.com`
 - ▶ Cada sitio deberá de tener su propio DocumentRoot, es decir, el directorio donde se almacenará el sitio.
 - ▶ Los pasos a seguir serían de manera esquemática los siguientes:
 - ▶ Crear para cada sitio el esquema de directorios donde se almacenará el contenido.
 - ▶ Crear las páginas web (el contenido) y almacenarlo en los directorios correctos.
 - ▶ Crear un fichero de configuración por cada sitio (en `sites-available`).
 - ▶ MUY IMPORTANTE: Hay que especificar las directivas `ServerName` y `DocumentRoot` pues será lo que diferencie a ambos sitios.
 - ▶ Activar ambos sitios (`a2ensite nombresitio`).
 - ▶ Añadir en el servidor DNS los dos dominios y los registros `www` correspondientes (tendrán la misma IP).
-

Actividad

- ▶ Crear dos hosts virtuales basados en nombre:
 - ▶ El sitio www.cocinillas.net, se almacena en /var/www/cocinillasnet y la página de inicio será index.html.
 - ▶ El sitio www.depesca.com, se almacena en /var/www/depescacom y la página de inicio se llamará depesca.html.
 - ▶ Comprueba que puedes acceder correctamente desde un cliente diferente a la propia máquina de ubuntu.



Autenticación y control de acceso.

- ▶ Se trata de directivas que **permitirán o denegarán el acceso a partes del sitio web** que se esté configurando.
- ▶ Como hablamos de “PARTES” todas las directivas que se mencionan a continuación estarán siempre incluidas **dentro de una directiva <Directory>**.
- ▶ También podrán encontrarse en ficheros tipo **.htaccess**.
- ▶ Directiva Require: **Require [not] entity-name [entity-name.]**
 - Esta directiva comprueba si la entidad especificada en entity-name se autoriza o no se autoriza.
 - **Require ip ip-address.** *//Autoriza a la ip-address a acceder.*
 - **Require host host-name.** *//Permite el acceso a host-name.*
 - **Require all granted** *// Acceso permitido a todos incondicionalmente.*
 - **Require all denied** *// Acceso denegado a todos incondicionalmente.*
 - **(***) Require not ip ip-address.** *//Deniega el acceso a ip-address.*
 - **(***) Require not host host-name.** *//Deniega el acceso a host-name*

(*) CUANDO QUERAMOS USAR SENTENCIAS CON NOT SE DEBEN INCLUIR DENTRO DE UN BLOQUE EN EL QUE APAREZCA UN RANGO MAYOR. VER EJEMPLOS.**

Autenticación y control de acceso.

- **Ejemplo1:** En la siguiente imagen las páginas almacenadas bajo el directorio /var/www/angelica sólo serán accesibles por la IP 192.168.1.35

```
<Directory /var/www/angelica>  
    Require all denied  
    Require ip 192.168.1.35  
</Directory>
```

- **Ejemplo2:** En la siguiente situación, todas las IP tendrán acceso salvo el equipo 192.168.1.35. Aquí vemos que aparece “not ip” pero dentro de un bloque RequireAll.

```
<Directory /var/www/angelica>  
<RequireAll>  
    Require all granted  
    Require not ip 192.168.1.35  
</RequireAll>  
</Directory>
```

- **Ejemplo3:** En el siguiente ejemplo, sólo 192.168.1.35 tendrá acceso.

```
<Directory /var/www/angelica>  
<RequireAll>  
    Require all granted  
    Require ip 192.168.1.35  
</RequireAll>  
</Directory>
```



Autenticación y control de acceso.

- **Ejemplo 4:** En la siguiente imagen todos los equipos tendrán acceso.

```
<Directory /var/www/angelica>
<RequireAll>
    Require all granted
    Require ip 192.168.1.35
</RequireAll>
</Directory>
```

- **Ejemplo5:** Mismo efecto que Ejemplo 1. Sólo 192.168.1.35 tiene acceso.

```
<Directory /var/www/angelica>
<RequireAny>
    Require all denied
    Require ip 192.168.1.35
</RequireAny>
</Directory>
```

- **Ejemplo6:** En el siguiente ejemplo nadie tiene acceso.

```
<Directory /var/www/angelica>
<RequireAll>
    Require all denied
    Require ip 192.168.1.35
</RequireAll>
</Directory>
```

