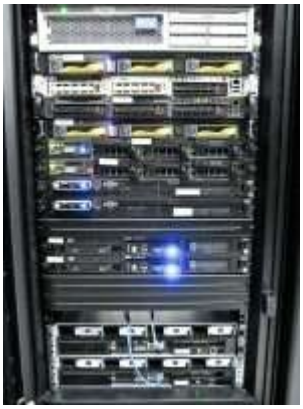
Despliegue de Aplicaciones Web

Servidores Web - Introducción

Hoy en día utilizamos Internet como una herramienta común: para el trabajo, para el ocio... Pero sin duda el elemento fundamental que usamos no es otro que el navegador, gracias al cual podemos sacar partido a todo lo que se encuentra en Internet: comprar entradas para el cine, acceder a nuestra cuenta bancaria, averiguar el tiempo que hará el fin de semana... pero nada de esto tendría sentido si detrás de cada página web a la que accedemos no existiera un servidor web, el cual permite que la página esté accesible 24x7 (24 horas al día y 7 días a la semana, es decir, siempre).



Detrás de cada página web debe existir un servidor web que ofrezca esa página, bien a los internautas, a los trabajadores de una empresa -por tratarse de una página web interna, de la empresa, no accesible a Internet-, o a todo aquel que disponga de una conexión de red con la cual pueda acceder a la página.

La configuración del servidor web dependerá de las páginas web que ofrezca, así la configuración no será la misma si la página posee contenido estático o no, o si se necesita que modifique el contenido según interacción del usuario, o si se necesita de comunicación segura en la transición de información, o si se debe tener en cuenta el control de acceso a determinados sitios de la página. Por lo tanto según las páginas web que se ofrezcan el servidor web deberá estar configurado para tal fin: con soporte PHP, con soporte de cifrado, con soporte de control de acceso, etc.

Pero ¿un servidor web pueda alojar varias páginas web o solamente una? Es más, ¿puede alojar varios sitios (Conjunto de páginas web), dominios de Internet (Nombre por el cual se reconoce a un grupo de dispositivos o equipos conectados a la red. Éstos pueden ser nombres locales, no existentes en Internet, pero son

mayoritariamente utilizados para su uso en Internet, por ejemplo: debian.org) o solamente uno, esto es, permite hosts virtuales (Dominios independientes que se pueden alojar en un mismo servidor web)?

Pues, un servidor web puede alojar varias páginas, sitios, dominios de Internet, pero hay que tener en cuenta que la elección del servidor web será muy importante para la configuración y administración de uno o múltiples sitios, ya que: ¿puede el servidor web ser modular -fácilmente se le pueden añadir o quitar características-?, o por la contra si queremos añadirle una funcionalidad que no posea en la instalación base debemos desinstalarlo e instalarlo de nuevo, por ejemplo: hasta ahora el servidor web solamente ofrecía páginas estáticas pero queremos ofrecer también páginas web dinámicas, qué hacemos: modular o nueva instalación.

También tenemos que pensar que todo puede crecer y lo que ahora era un servidor web que ofrecía x número de páginas necesitamos que ofrezca x*y, con lo cual tenemos que prever la **escalabilidad** del servidor web, y también la **estabilidad**: ¿cómo se comporta ante mútiples conexiones simultáneas?

De nada servirá tener instalado un servidor web sin saber cómo se va a comportar ofreciendo el servicio, con lo cual será muy importante previamente y durante el funcionamiento del servidor establecer unas pruebas de funcionamiento del mismo y registrar lo acontecido.

Por todo lo anteriormente uno de los recursos más utilizados es la configuración y administración d un servidor Apache, ya que soporta páginas web estáticas, dinámicas, hosts virtuales, seguridad mediante cifrado, autenticación y control de acceso, modularización y monitorización de archivos de registro.

1.1.- Servicio de ficheros estáticos.

¿Es necesario que todas las páginas web se modifiquen constantemente? ¿Un blog sería útil si el contenido no sufre cambios? ¿Y un manual? ¿Si actualizamos un manual la página deja de ser estática?

Todas aquellas páginas web que durante el tiempo no cambian su contenido no necesariamente son estáticas.

Una página estática puede modificarse, actualizando su contenido y seguir siendo estática, por lo que debemos diferenciar cuando accedemos a una página web entre código ejecutable en el lado del servidor y en el lado del cliente -equipo que solicita la página mediante el cliente web (navegador)-.

Si al acceder a una página web no es necesaria la intervención de código en el lado del servidor -por ejemplo código PHP- o en el lado del cliente -por ejemplo javascript- entonces entenderemos que la página es estática, si por el contrario es necesaria la intervención en el lado del servidor y/o en el lado del cliente entenderemos que la página es dinámica.

Ofrecer páginas estáticas es simple, puesto que solamente se necesita que el servidor web disponga de soporte html/xhtml/css o incluso solamente html/xhtml. En cuanto a configuración y administración del servidor es el caso más simple: solamente se necesita un soporte mínimo base de instalación del servidor Apache, esto es, no se necesita por ejemplo soporte PHP.

1.2.- Contenido dinámico.

Muchas veces seguro que te encuentras visitando una página web y la información te parece tan interesante que procedes y guardas en Favoritos la dirección URL (dirección de Internet de un recurso válida para su posible utilización a través de Internet, la cual permite que el navegador la encuentre y la muestre de forma adecuada, por ejemplo: http://www.debian.org) para una posterior visión, pero cuando de nuevo deseas ver la página resulta que lo que estás viendo no tiene nada que ver o es distinto de lo que esperabas, ¿qué ha ocurrido? Pues puede que la página haya cambiado su contenido o que la página que visitas posee contenido no estático, dinámico, dependiente del código ejecutado en el servidor o en el cliente al acceder a la página.

Imagínate que accedes a una página web y dependiendo si posees una cuenta de usuario u otra el contenido es distinto, o que presionas en una imagen de la página y se produce un efecto en la misma, o que el contenido cambia dependiendo del navegador. De cualquier forma la página ha sido modificada mediante una interacción con el usuario y/o el navegador, por lo tanto nos encontramos con una página dinámica.

Como bien puedes pensar, una página dinámica, necesita más recursos del servidor web que una página estática, ya que consume más tiempo de CPU y más memoria que una página estática. Además la configuración y administración del servidor web será más compleja: cuantos más módulos tengamos que soportar, más tendremos que configurar y actualizar. Esto también tendrá una gran repercusión en la seguridad del servidor web: cuántos más módulos más posibilidades de problemas de seguridad, así si la página web dinámica necesita, para ser ofrecida, de ejecución en el servidor debemos controlar que es lo que se ejecuta.

1.3.- Protocolo HTTP y HTTPS.

¿Quieres conservar la información de forma confidencial? ¿Quieres transferir información de forma segura? Si estás pensando en este tipo de preguntas necesariamente estás pensando en el protocolo HTTPS (protocolo basado en el protocolo HTTP, destinado a la transferencia segura de datos mediante cifrado, es decir, es la versión segura de HTTP) yno en el protocolo HTTP (protocolo usado en cada transacción de la World Wide Web).

El protocolo HTTPS permite que la información viaje de forma segura entre el cliente y el servidor, por la contra el protocolo HTTP envía la información en texto claro, esto es, cualquiera que accediesea la información transferida entre el cliente y el servidor puede ver el contenido exacto y textual de la información.

Para asegurar la información, el protocolo HTTPS requiere de certificados y siempre y cuando sean validados, la información será transferida cifrada. Pero cifrar la información requiere un tiempo de computación, por lo que será perjudicado el rendimiento del servidor web.

Así, ¿es necesario que toda, absolutamente toda, la información sea transferida entre el cliente y servidor de forma cifrada? A lo mejor solamente es necesario que sea cifrada la autenticación a dicha información, por eso en algunas páginas web puede que el servidor esté configurado para que en todo el dominio esté cifrada su información o simplemente el intento de acceso a la misma.

Un servidor web, como Apache, puede emitir certificados, pero puede que en algún navegador sea interpretado como peligroso, esto suele ser debido a que los navegadores poseen en su configuración una lista de Entidades Certificadoras que verifican, autentican y dan validez a los certificados.

¿Tú, confiarías en un DNI que no fuese certificado por una entidad de confianza como el Ministerio del Interior? Pues, lo mismo le pasa a los navegadores, solamente

confían en quien confían. Eso no quiere decir que no puedes crear tus certificados en un servidor web, de hecho muchas empresas lo hacen, sobre todo para sitios internos o externos en los que solamente puede acceder personal autorizado por la propia empresa.

Ahora si, si utilizas certificados mediante Apache en un sitio visible a través de Internet y accesible por cualquier usuario, o bien eres una empresa o entidad en la que de por si confía el usuario o la imagen de la empresa o entidad quedará muy mal parada, ya que lo más probable es que el usuario no aceptará la comunicación, por visionar en el navegador un aviso de problema de seguridad.

El protocolo HTTPS utiliza cifrado sobre SSL/TLS (protocolos criptográficos que proporcionan comunicaciones seguras por una red, comúnmente Internet) que proporcionan autenticación y privacidad. Entonces, si necesitas que la información viaje cifrada debes emplear el protocolo HTTPS, en caso contrario el protocolo HTTP. Hay que dejar claro que la utilización del protocolo HTTPS no excluye ni impide el protocolo HTTP, los dos pueden convivir en un mismo dominio.

Bien, pero, ¿cómo funcionan? En el protocolo HHTP cuando escribes una dirección URL en el navegador, por ejemplo http://www.debian.org/index.es.html, antes de ver la página en el navegador existe todo un juego de protocolos, sin profundizar en todos ellos básicamente lo que ocurre es lo siguiente:

Se traduce el dominio DNS (sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a Internet o a una red privada, por ejemplo: apache.org determina un dominio org (organización) y un subdominio que identifica en este caso la máquina o conjunto de máquinas de nombre apache) por una IP, una vez obtenida la IP se busca en ella si un servidor web aloja la página solicitada en el puerto 80 (número utilizado en las comunicaciones cliente/servidor, en transmisiones TCP o UDP comprendido entre 1 y 65535, que indica por donde tiene lugar la conexión con un servidor.

Están estandarizados, esto es, un servidor suele estar activo siempre por definición en un puerto determinado, pero éste puede que sea modificado en la configuración del servidor. Por ejemplo un servidor web espera en el puerto TCP 80) , puerto TCP

(es uno de los protocolos fundamentales en Internet. Garantiza que los datos serán entregados en su destino sin errores y una vez recogidos ponerlos en el mismo orden en que se transmitieron) asignado por defecto al protocolo HTTP. Si el servidor web aloja la página ésta será transferida a tu navegador.

Sin embargo cuando escribes en el navegador una dirección URL con llamada al protocolo HTTPS, el procedimiento es similar al anterior pero un poco más complejo, así se traduce el dominio DNS por una IP, con la IP se busca el servidor web que aloja la página solicitada en el puerto 443, puerto TCP asignado por defecto al protocolo HTTPS, pero ahora antes de transferir la página a tu navegador se inicia una negociación SSL, en la que entre otras cosas el servidor envía su certificado -el navegador aunque es poco habitual también puede enviar el suyo-. Si el certificado es firmado por un Entidad Certificadora de confianza se acepta el certificado y se cifra la comunicación con él, transfiriendo así la página web de forma cifrada.

Puedes hacer que un servidor web para una determinada página espere los protocolos HHTP y HTPS en puertos TCP distintos del 80 y 443 respectivamente. Eso sí, cuando visites la página web a mayores en la dirección URL debes especificar el puerto TCP, por ejemplo: http://www.tupagina.local:8080

De esta forma el servidor web espera la petición de la página www.tupagina.local en el puerto 8080.

Del mismo modo en la dirección URL: https://www.tupagina.local:4333 espera la petición de la página www.tupagina.local en el puerto 4333.

Como ves, puedes configurar los puertos, pero ten en cuenta que cualquiera que quisiera acceder a esas páginas debería saber el puerto TCP de la solicitud.

Entonces, quiere decir que ¿aunque no escribas el puerto TCP en las direcciones URL estas se interpretan en el puerto 80 y 443 para el protocolo HTTP y HTTPS respectivamente?

Pues si, así es. Es lo mismo escribir:

http://www.tupagina.local:80 que: http://tupagina.local.

De igual forma es lo mismo escribir:

http://www.tupagina.local:443 que: https://www.tupagina.local

En la página http://www.warriorsofthe.net/index.html puedes encontrar un vídeo muy ameno sobre el funcionamiento de Internet.