

Examen 2 Evaluación Despliegue de aplicaciones Web:

Alumno: Brais Bea Mascato

Actividad 1

Configuraremos el servidor en Linux para restringir el acceso a los recursos a usuarios y grupos.

Con autenticación HTTP Basic:

1. Crear un archivo de contraseñas con dos usuarios: `cliente1` y `cliente2`.

```
sudo htpasswd -c /etc/apache2/.htpasswd cliente1
```

```
sudo htpasswd /etc/apache2/.htpasswd cliente2
```

2. Crear el grupo `cartera`, el cual contiene a ambos.

```
sudo nano /etc/apache2/.htgroup
```

Añadimos la línea:

```
cartera: cliente1 cliente2
```

3. Crea la carpeta `/var/www/examen` con algún archivo HTML y configura adecuadamente el Apache de manera que:

```
mkdir /var/www/examen
```

```
sudo nano /var/www/examen/index.html
```

añadimos un html de prueba:

```
<h1> Hola Mundo! </h1>
```

- 1) Que pueda acceder `cliente2`, pero no `cliente1`.
- 2) Que puedan acceder los miembros del grupo `cartera`.
- 3) Que pueda acceder cualquier usuario válido.

```
sudo nano /etc/apache2/sites-available/examen.conf
```

```
<Directory "/var/www/examen">
    AuthType Basic
    AuthName "Zona restringida"
    AuthUserFile /etc/apache2/.htpasswd
    AuthGroupFile /etc/apache2/.htgroup
    Require valid-user
    Require group cartera
    Require user cliente2
</Directory>
```

Posteriormente habilitamos el sitio web con:

```
sudo a2ensite examen.conf
```

Resetear el servidor apache para que todo funcione correctamente.

```
sudo service apache2 restart
```

(4 puntos)

Actividad 2

Se solicita, en entorno Windows:

1. Crear el directorio `C:\Users\Administrador\prueba` e le asignamos un alias para que sea accesible desde `http://192.168.0.2/prueba`. Creamos el archivo `prueba.html` dentro.
2. Permitir el empleo de ficheros `.htaccess` en ese directorio para todas las directivas admitidas.

```
mkdir "C:\Users\Administrador\prueba"  
notepad "C:\Users\Administrador\prueba\prueba.html"
```

Editar el archivo `httpd.conf`:
`C:\Program Files\Apache Group\Apache2\conf\httpd.conf`

Añade el alias al final del archivo y la directiva de acceso al directorio.

```
Alias /prueba "C:/Users/Administrador/prueba"  
  
<Directory "C:/Users/Administrador/prueba">  
    Options Indexes FollowSymLinks  
    AllowOverride All  
    Require all granted  
</Directory>
```

(2 puntos)

Test.

Selecciona una o más de las respuestas posibles:

1. La plataforma LAMP emplea como servidor web:
 - a) **Apache.**
 - b) CGI.
 - c) SQL.
 - d) Arapahoe.
2. El servidor web Apache puede dar servicio de contenido:
 - a) Dinámico pero no estático.
 - b) **Estático y dinámico.**
 - c) Estático pero no dinámico.
 - d) Ni estático ni dinámico.

3. El servidor web Apache...

- a) **Está estructurado en módulos.**
- b) Disponible sólo para Windows.
- c) **Es de código abierto.**
- d) Es una mejor opción que NGINX para aquellos que tienen un sitio web más estático.

4. Señala las correspondencias correctas de campos con sus identificadores:

- a) **date %t.**
- b) **status %>s.**
- c) **authuser %u.**
- d) **host %h.**

5. Para habilitar el módulo SSL en Apache, ¿qué comando/s podemos usar?

- a) a2dismod ssl.
- b) a2enabled ssl.
- c) a2habmod ssl.
- d) **a2enmod ssl.**

6. Rotatelog, proporcionado por Apache, permite:

- a) **Rotar logs.**
- b) Solamente rotar logs por el tamaño de los mismos.
- c) **Rotar logs por el tamaño de los mismos.**
- d) **Rotar logs por intervalos de tiempo, siempre y cuando en el intervalo de tiempo definido existan nuevos datos.**

7. La directiva DocumentRoot en Apache tiene por objetivo:

- a) Configurar la documentación propia del servidor.
- b) **Especificar dónde Apache debe buscar los archivos que forman el sitio.**
- c) Definir la dirección de correo del administrador.
- d) No existe tal directiva en Apache.

8. De las siguientes tecnologías, ¿cuál no es interpretada por el servidor web?

- a) PHP.
- b) CGI.
- c) ASP.
- d) **CSS.**

(2 puntos)

Respuestas:

1. a)
2. b)
3. a), c)
4. a), b), c), d)
5. d)
6. a), c), d)
7. b)
8. d)

Desarrollo.

Explica, usando tus propias palabras, las ventajas e inconvenientes de Apache frente a NGINX y por qué elegirías uno u otro para un trabajo que propongamos.

(2 puntos)

Las principales diferencias a nivel de arquitectura entre apache y nginx podrían resumirse en que:

Apache: Utiliza un modelo basado en procesos, creando un nuevo proceso o hilo para cada petición. Esto puede llevar a un mayor uso de recursos bajo carga pesada.

- **Ventajas:** Funciona tanto en windows como en linux con el mismo rendimiento, servidor ligero.
- **Inconvenientes:** Más lento que NGINX al servir contenido estático y el consumo de recursos aumenta exponencialmente en entornos de alta demanda.

NGINX: Utiliza una arquitectura asíncrona, basada en eventos, que es más eficiente en el manejo de alta concurrencia. Crea un proceso controlador y múltiples procesos trabajadores, permitiendo a cada trabajador gestionar múltiples peticiones sin bloquearse.

- **Ventajas:** Servidor ligero con alta escalabilidad. Sirve contenido estático el doble de rápido que Apache. Posee soporte empresarial.
- **Inconvenientes:** Su rendimiento en entornos Windows es peor que en otras plataformas.

NGINX por tanto es un servidor más escalable para altos volúmenes de peticiones que apache. En este contexto si nuestra aplicación va a tener un alto volumen de peticiones y eventos concurrentes, como por ejemplo una red social, tendría mucho más sentido utilizar NGINX antes que Apache.

Si por el contrario nuestra plataforma no va a estar sujeta a una alta demanda, no va a gestionar múltiples eventos y usuarios un servidor apache sería más que suficiente, en este contexto estaríamos hablando de webs corporativas, o incluso webs de reservas, pudiendo esta última aplicación verse mejorada por la implementación de un sistema híbrido Apache-NGINX.