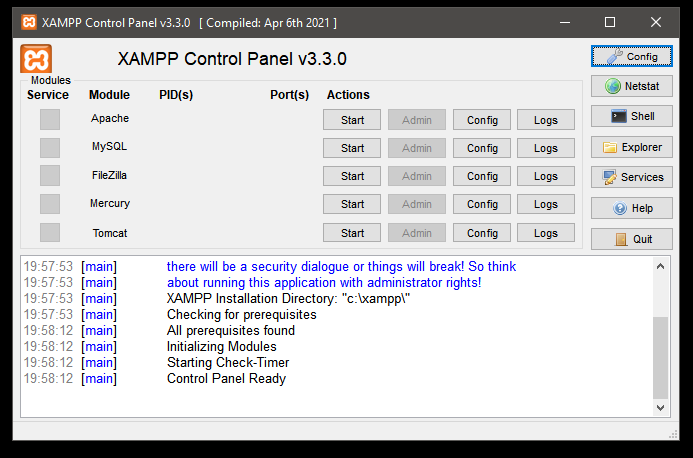
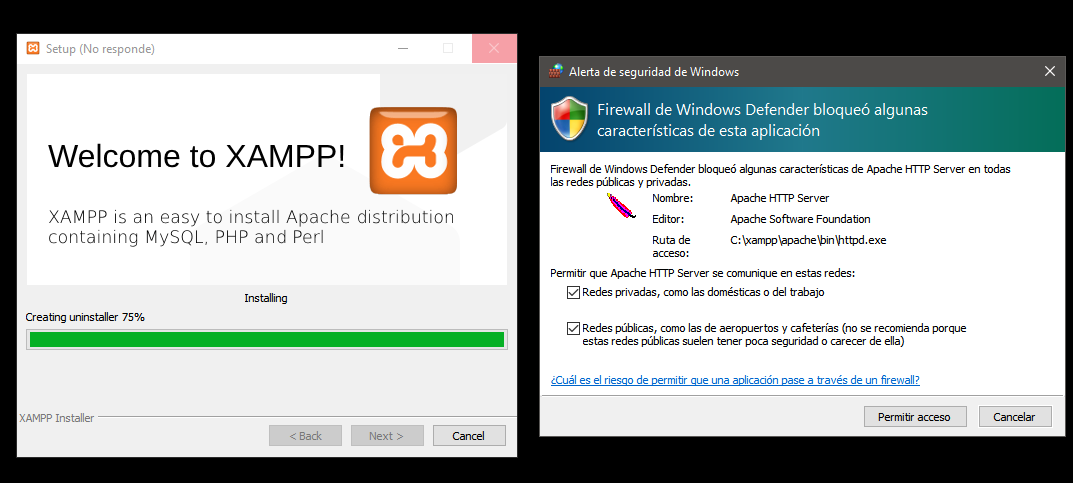
**Alta Disponibilidad**

Una característica de un sistema seguro es la Disponibilidad, es decir, que los datos estén accesibles por las personas autorizadas. Esto no siempre es posible bien a causa de ataques, accidentes o incidencias tanto de software como de hardware. Por lo tanto se hace necesario investigar e implementar aquellas soluciones que consigan mantener la disponibilidad el mayor tiempo posible.

**Trabajo - Práctica**

1. Instalar un servidor web en una máquina virtual (ej. wamp/lamp, xampp, iis, etc...) Comprobar el acceso web desde otra máquina.







Para poder acceder remotamente, buscamos en este archivo esta parte y la borramos:

—---------------------------------------------------------------------------------------------------------

#

# New XAMPP security concept

#

<LocationMatch “^/(?i:(?:xampp|security|licenses|phpmyadmin|webalizer|server-status|server-info))”>

Require local

ErrorDocument 403 /error/XAMPP\_FORBIDDEN.html.var

</LocationMatch>

—---------------------------------------------------------------------------------------------------------

Y la cambiamos por esto:

#

# New XAMPP security concept

#

# Close XAMPP security section here

<LocationMatch “^/(?i:(?:security))”>

Order deny,allow

#Deny from all

#Allow from ::1 127.0.0.0/8

Allow from all

ErrorDocument 403 /error/HTTP\_XAMPP\_FORBIDDEN.html.var

</LocationMatch>

# Close XAMPP sites here

<LocationMatch “^/(?i:(?:xampp|licenses|phpmyadmin|webalizer|server-status|server-info))”>

Order deny,allow

#Deny from all

#Allow from ::1 127.0.0.0/8

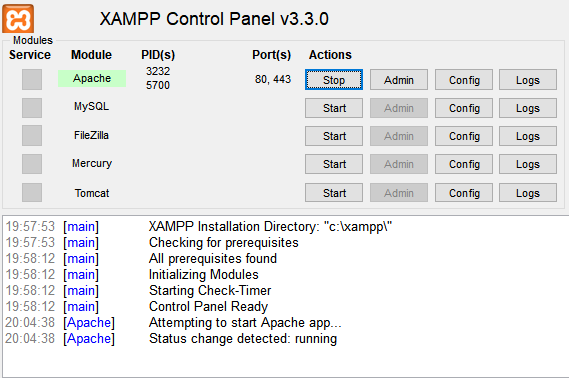
Allow from all

ErrorDocument 403 /error/HTTP\_XAMPP\_FORBIDDEN.html.var

</LocationMatch>

—---------------------------------------------------------------------------------------------------------

Y iniciamos Apache:



Ahora para acceder desde otra máquina necesitamos saber nuestra ip:

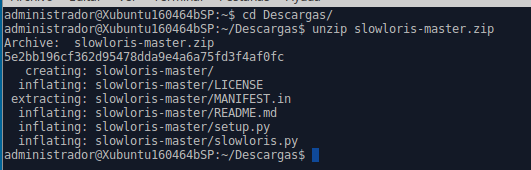


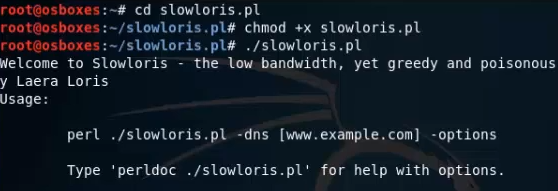


Donde hayas guardado la página (en mi caso he cogido un ejemplo de internet): **C:\xampp\htdocs\proyecto**, en otra máquina: <http://192.168.0.19/proyecto/>

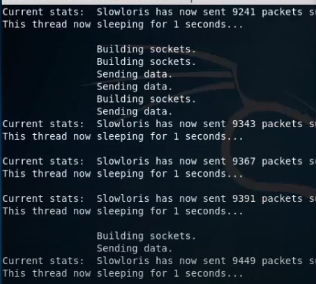


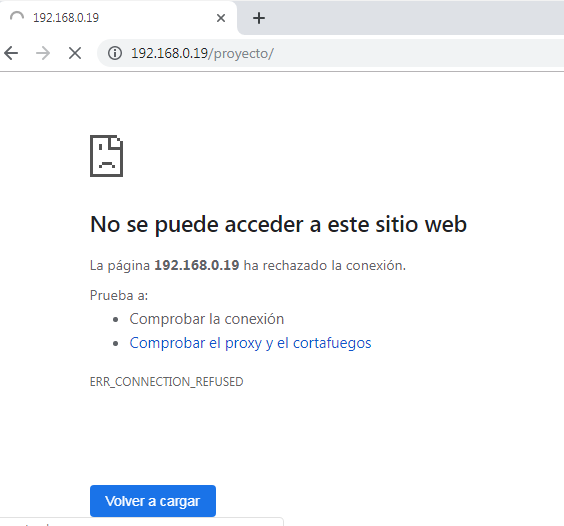
2. Realizar un ataque DoS (Denegación de Servicio) mediante un script o software al servidor creado en el punto anterior. Comprobar el acceso web desde otra máquina.





perl ./slowloris.pl -dns 192.168.0.19 -port 80 -timeout 1 -num 5000





3. En otra máquina virtual instalar el mismo servidor web y un software de cluster para que si uno de los 2 servidores caen, el otro siga dando respuesta, o bien un balanceador de carga y repartir el trabajo entre las 2 máquinas. (ej. Mscs, Kerio, Heartbeat, KeepAlived, Corosync+Pacemaker, etc...)

Comprobar el acceso web independientemente de que se apaguen alguna de las 2 máquinas con el servidor web.

**No he conseguido hacerlo adjunto instrucciones:**

**b. Configuración**

Una vez se han instalado las máquinas virtuales con Ubuntu Server 18.04 LTS, se procede a actualizar los repositorios en cada una de las maquinas desde la línea de comandos mediante la instrucción:

***sudo apt-get update***

Seguidamente se instala UNZIP a través del comando:

***apt-get install unzip.***

Hay que recordar que debe iniciar sesión con el usuario root para que se permita los cambios. Pero antes debe iniciar Putty para acceder a estos servidores. Inicialmente los servidores poseerán direcciones DHCP proporcionadas por su router. Se recomienda utilizar direcciones estáticas para garantizar el acceso a los servidores. Para este caso se han configurado los equipos con las direcciones, con dirección IP 172.23.46, 172.23.19.47, puertos 4243 y 4242 respectivamente. El acceso para la configuración se hará mediante putty.

Lo siguiente es descargar e instalar el paquete Heartbeat en cada uno de los servidores virtuales (CMaster y CSlave) a través del comando:

***apt-get install heartbeat.***

Una vez descargados en ambos servidores, Heartbeat, buscamos el paquete en la siguiente ruta: ***usr/share/doc/heartbeat***

En esta carpeta se encontrarán los archivos comprimidos: ha.cf.gz, haresources.gz y authkeys. Se procederá a descomprimir los archivos en cada uno de los servidores con el comando:

***gzip -d ha.cf.gz***

***gzip -d haresources.gz***

A continuación, se deben copiar los archivos ha.cf, haresources y authkeys a la carpeta /etc/ha.d.

***cp ha.cf /etc/ha.d***

***cp haresources /etc/ha.d***

***cp authkeys /etc/ha.d***

Se debe ubicar en el directorio /etc/ha.d. Ahora, se procederá a la edición de los archivos ha.cf, haresources y authkeys. Para la edición de estos archivos se utilizará el editor de Linux nano.

***nano ha.cf***

En el archivo ha.cf, se deben “descomentar” las siguientes líneas:

• keepalive 2, sirve para verificar la comunicación entre los servidores.

• deadtime 30 se debe cambiar a deadtime 5, para iniciar un servicio.

• udpport 694. • bcast eth0, mediante el comando ifconfig, verifique el nombre de su tarjeta de red. Este nombre es distinto dependiendo del software de máquina virtual que esté utilizando. Para este caso es ens18.

• mcast eth0 225.0.0.1 694 1 0, cambiar al que corresponda en su equipo.

• ucast eth0 192.168.1.2, Si se trata del archivo de configuración del master, esta dirección IP debe apuntar al IP del esclavo, y si es la configuración del esclavo, esta dibe direijirse a la IP del maestro. En nuestro caso sería, ucast ens18 172.23.19.47. Es importante aclarar que luego de “descomentar” se debe adicionar abajo la línea upd ens18.

• auto\_failback on, validar que este esté en “on”.

• node ken3, ken3 sería el nombre del nodo maestro. En nuestro caso, node master-cluster.

• node kathy, cambiar al nombre del nodo esclavo. En nuestro caso, node slave-cluster. Nota: Estos mismos pasos, en la configuración del servidor maestro se realizan en el servidor esclavo con la diferencia en que en el apartado o línea “ucast ens18…” se coloca la dirección IP del servidor maestro.

El segundo archivo que se debe modificar es “haresources”. Al editarlo, se debe ir hasta el final de la página y agregar el nombre del equipo o nodo maestro, la dirección IP virtual que va a ser de referencia para uso del servicio por parte de los clientes y el servicio utilizado. Para cada equipo, nodo maestro y nodo esclavo se asignará la ip 172.23.19.48, la cual corresponderá a la IP virtual. La línea quedará escrita de la siguiente manera:

Del lado del servidor maestro: ***master-cluster 172.23.19.48 apache2***

Del lado del servidor esclavo: ***slave-cluster 172.23.19.48 apache2***

El tercer y último archivo para modificar es “authkeys”, al cual se deberá agregar las líneas siguientes al final: “auth 1 y 1 crc”. Aplicarlo en cada servidor.

Nota: es importante colocarle los permisos de control total al archivo “authkeys” a sólo administradores ya que de forma predeterminada no viene establecidos. Este se valida a través del comando “ls -l”.

***chmode 600 authkeys***

***./heartbeat restart***

Para verificar que todo se encuentra funcionando, ejecute el comando:***./heartbeat status***

Este comando le mostrará todo lo concerniente al servicio y detalles de errores si existiese.

Nuestro clúster de alta disponibilidad es un servicio web, por lo que se procederá a instalar el servidor web en los equipos máster y esclavo. Para instalar apache2 en los equipos, se ejecutará la siguiente instrucción:

***sudo apt-get install apache2***

**c. Funcionamiento**

Para validar el funcionamiento del clúster se editará la página del “index.html” de apache2 para ver como el clúster alta disponibilidad funciona. Para eso nos colocamos en la ruta “/var/www/html” y modificamos la cabecera que aparece en la figura (MASTER-CLUSTER ON y SLAVE-CLUSTER ON”), de forma tal que se pueda identificar que equipo esta brindando el servicio por medio de la IP virtual que introducirá el cliente en un navegador.

Una vez realizado los cambios en ambos archivos de apache2, se debe reiniciar el servicio con el comando apache2 restart.

Desde una maquina conectada a la misma red de los servidores, se abre el navegador y en la barra de direcciones se colocará la dirección de cada servidor para comprobar su disponibilidad en forma independiente.

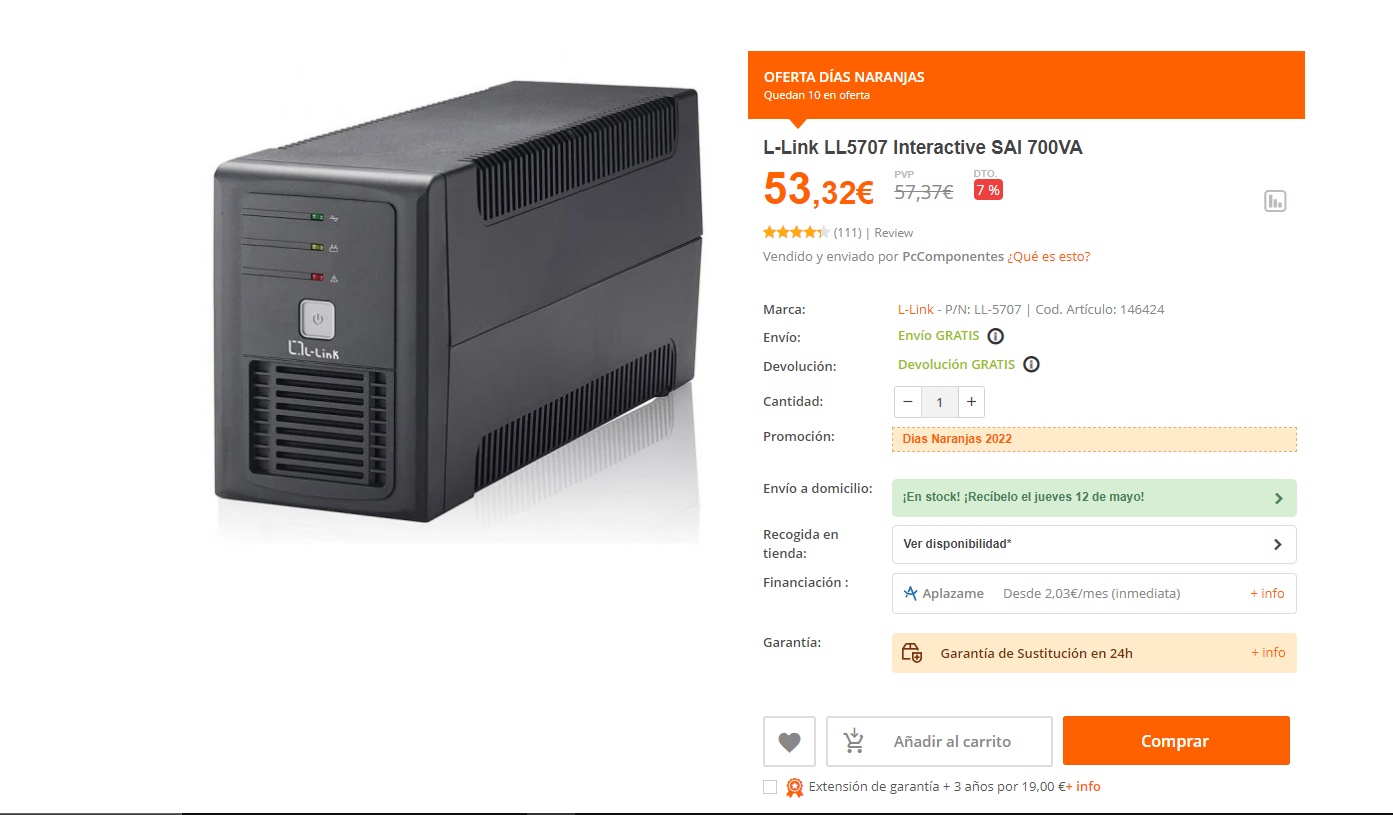
Las direcciones IP se han encerrado en un recuadro en rojo para identificar el equipo que está brindando servicio.

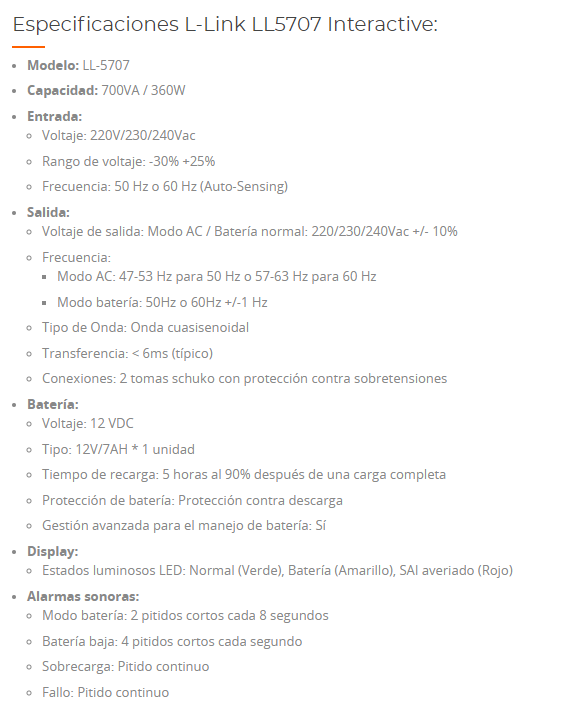
Para verificar el funcionamiento total del clúster, se ha de suspender el servicio de apache2 en el servidor maestro. Desde el navegador se introducirá la IP virtual, la cual deberá redirigir el servicio hacia el servidor esclavo automáticamente. Como se observa en la figura No. 18, el servidor maestro no responde directamente la solicitud de apache2 ya que se ha detenido el servicio.

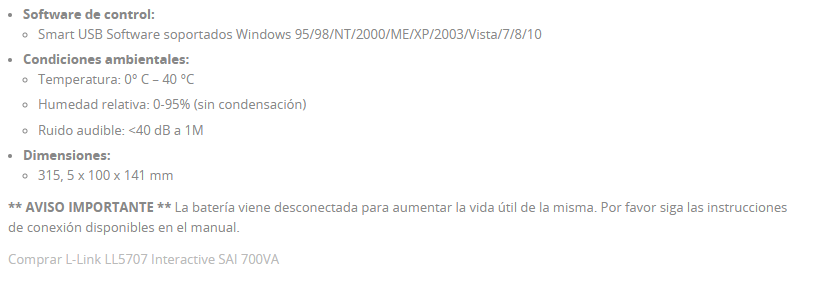
Al refrescar la página que posee la IP virtual, esta inmediatamente dirige la solicitud para que sea atendida por el servidor esclavo

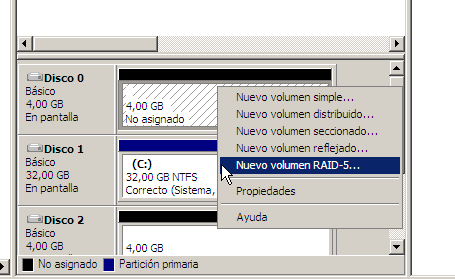
Finalmente restauramos el servicio de apache2 en el servidor maestro y también reiniciamos el servicio de Heartbeat en el servidor maestro para devolver el control a este último.

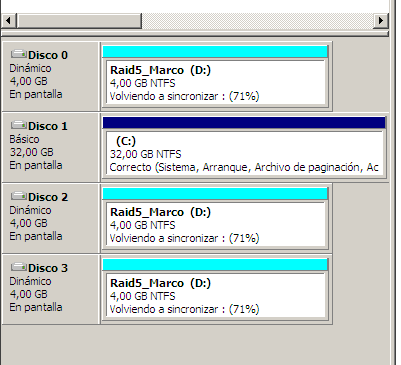
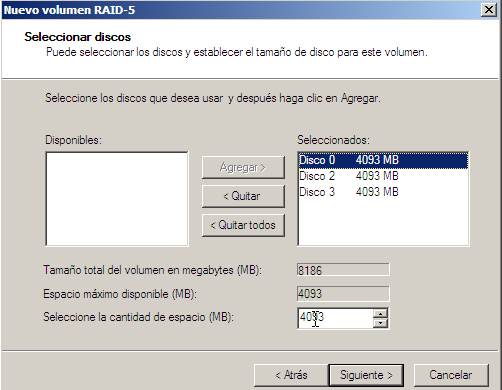
4. Busca un modelo de SAI, e indica sus características.





5. Implementar un RAID-5 en Windows o Linux.





Nota: para evitar problemas técnicos, sobrecarga de uso de máquinas virtuales, esta práctica puede realizarse en pareja.