**Balanceador con HAProxy**

ED2. Hasna, Toni, Miguel, David

**ÍNDICE**

[**1. Descripción.**](#_onn30bc5e14c) **3**

[**3. Instalación.**](#_qw2b8bz6ta88) **4**

[**4. Configuración.**](#_bybfvpvkfpbb) **5**

[**5. Funcionamiento.**](#_lq7ppuijczjs) **7**

[**6. Fuentes.**](#_d2cfafrzfwbd) **9**

# Descripción.

HAProxy es un software libre que actúa como balanceador de carga (load balancer) ofreciendo alta disponibilidad, balanceo de carga y proxy para comunicaciones TCP y HTTP.

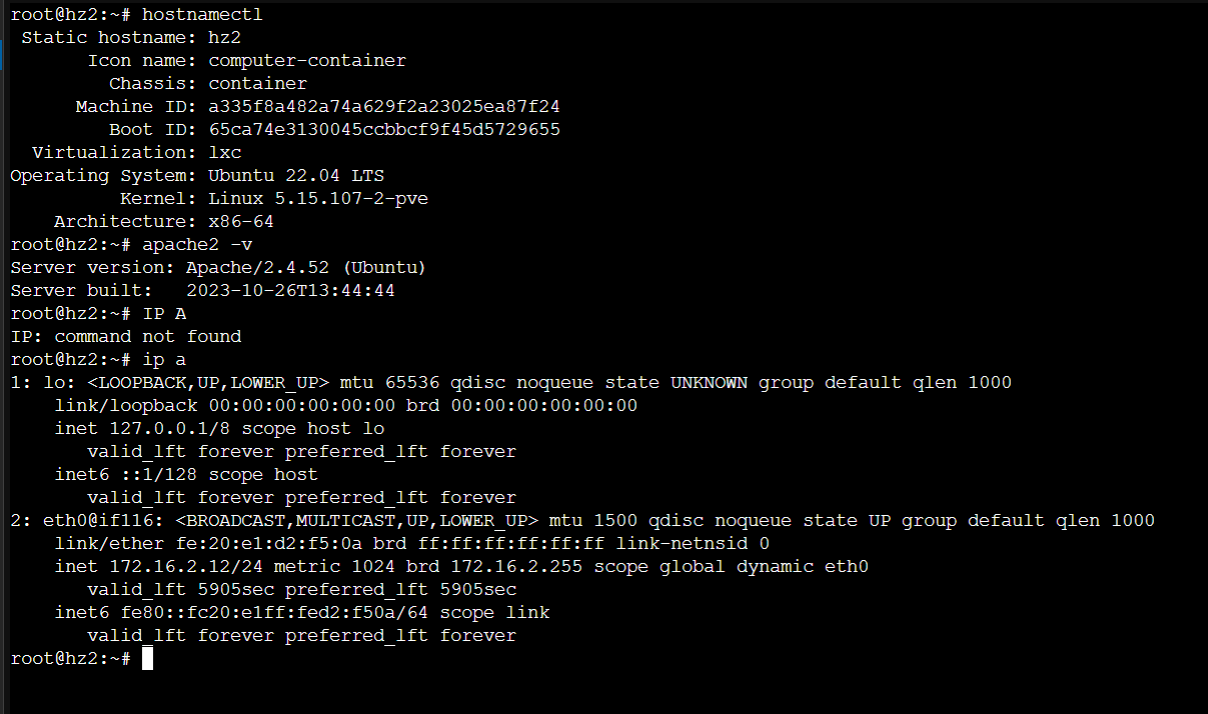
En esta configuración hemos utilizado 3 servidores: HAProxy + 2 webs.

1. Máquinas y paquetes utilizados.

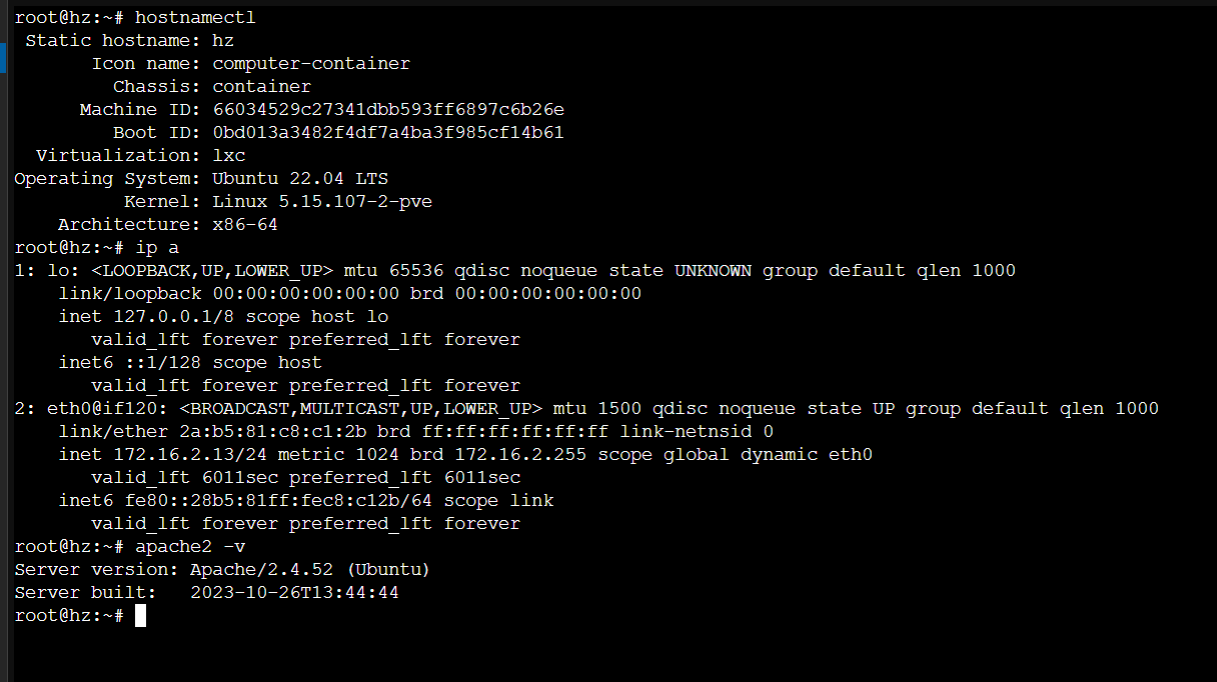
Donde tenemos instalado el haproxy. Se encuentra en DMZ

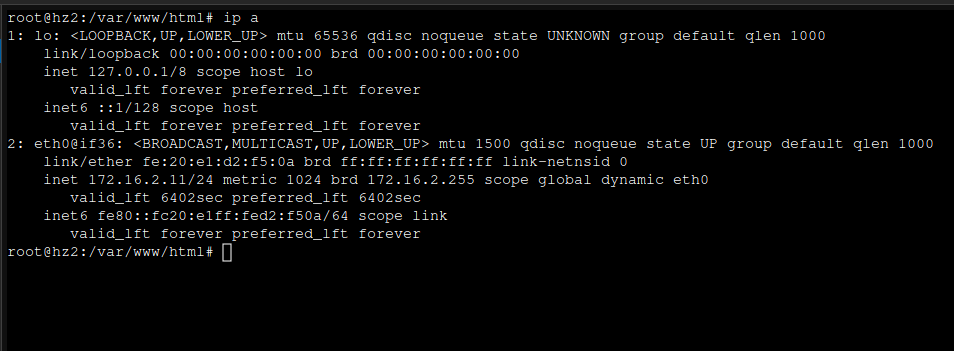


Server1, se encuentra en la LAN



Server2, se encuentra en la LAN



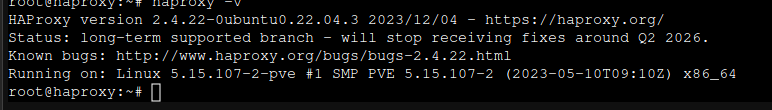


# Instalación.

En la VM de ubuntu instalamos el haproxy con los siguientes comandos:

apt update && apt upgrade

apt install haproxy



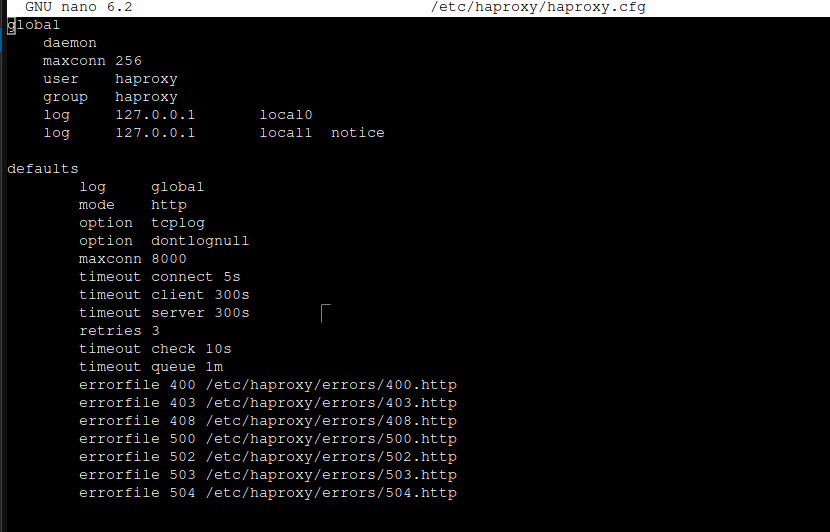
En los contenedores instalamos apache2 con los siguientes comandos:

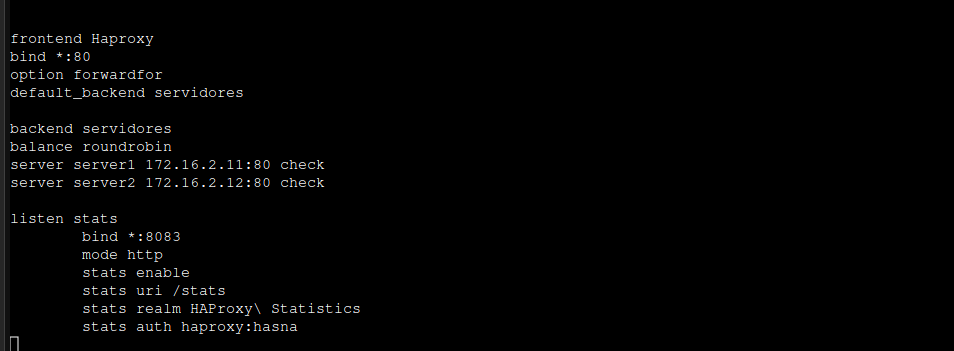
sudo apt update

sudo apt install apache2

# Configuración.

Editamos el fichero de configuración.





**Sección Global:**

daemon: Indica que HAProxy se ejecutará como un demonio en segundo plano.

maxconn 256: Establece el número máximo de conexiones concurrentes permitidas.

user haproxy y group haproxy: Especifica el usuario y grupo bajo los cuales se ejecutará HAProxy.

log 127.0.0.1 local0 y log 127.0.0.1 local1 notice: Configura la dirección del registro. Los registros se envían al sistema local en los puertos 127.0.0.1:local0 y 127.0.0.1:local1 con nivel de notificación.

**Sección Defaults:**

log global: Establece la dirección para los registros globales.

mode http: Configura el modo de HAProxy como HTTP.

option tcplog y option dontlognull: Opciones de registro para tráfico TCP y para no registrar solicitudes nulas.

maxconn 8000: Establece el número máximo de conexiones simultáneas.

timeout connect 5s, timeout client 300s, timeout server 300s: Establece los tiempos de espera para diferentes fases de la conexión.

retries 3: Número de reintentos antes de marcar un servidor como no disponible.

timeout check 10s: Tiempo de espera para verificar el estado del servidor.

timeout queue 1m: Tiempo de espera para las colas.

errorfile: Rutas a los archivos de error personalizados para códigos HTTP específicos.

**Sección Frontend:**

bind \*:80: Enlaza el frontend a todas las interfaces en el puerto 80.

option forwardfor: Agrega la dirección IP real del cliente a los encabezados HTTP.

default\_backend servidores: Define el backend predeterminado al que se enviarán las solicitudes.

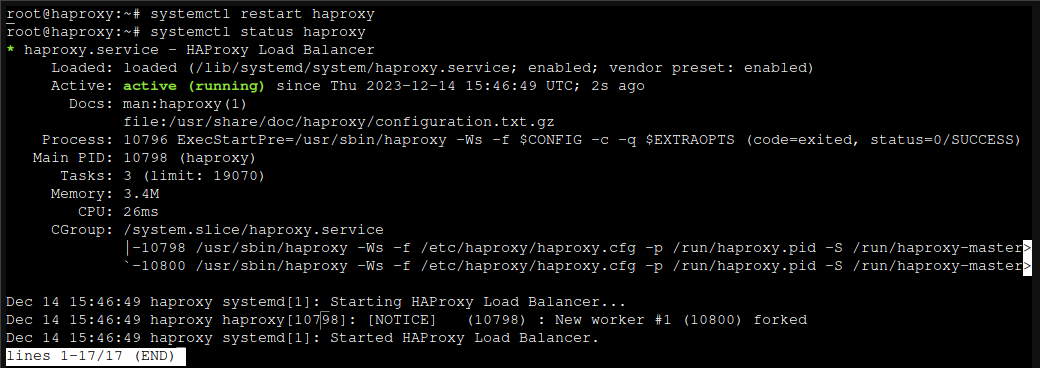
**Sección Backend (servidores):**

balance roundrobin: Utiliza el algoritmo de equilibrio de carga round-robin.

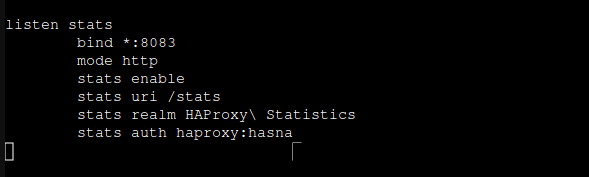
server server1 172.16.2.12:80 check: Define el servidor 1 con su dirección IP, puerto y verifica su estado.

server server2 172.16.2.11:80 check: Define el servidor 2 de manera similar.

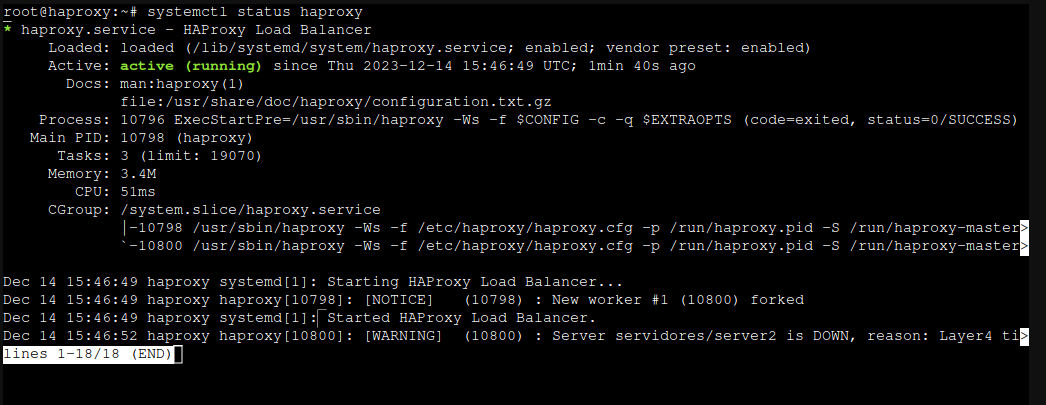
Reiniciamos la herramienta.

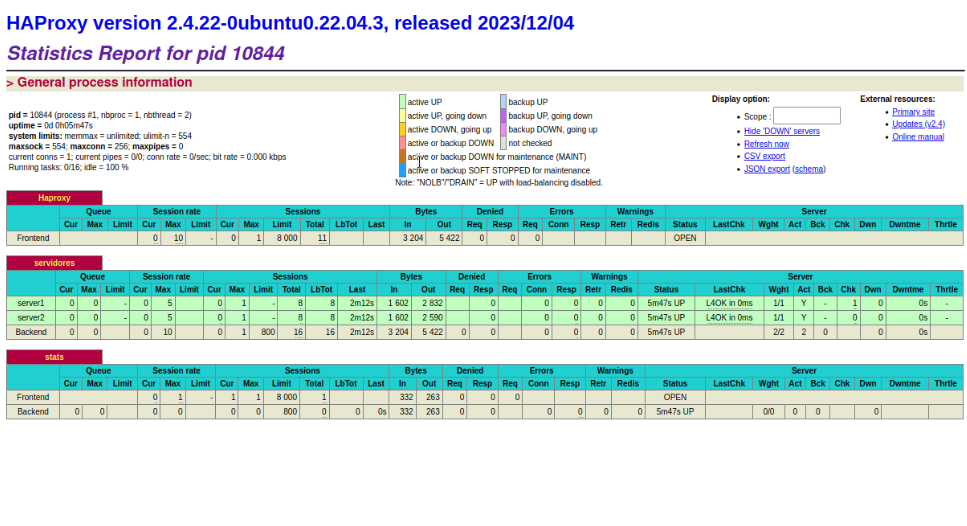


Para visualizar las estadísticas por vía web configuraremos el mismo archivo, es decir habilitar la consulta de estadísticas en directo del funcionamiento del servidor. Aquí podemos escoger el puerto que queramos. También es importante definir un usuario y contraseña para el acceso:

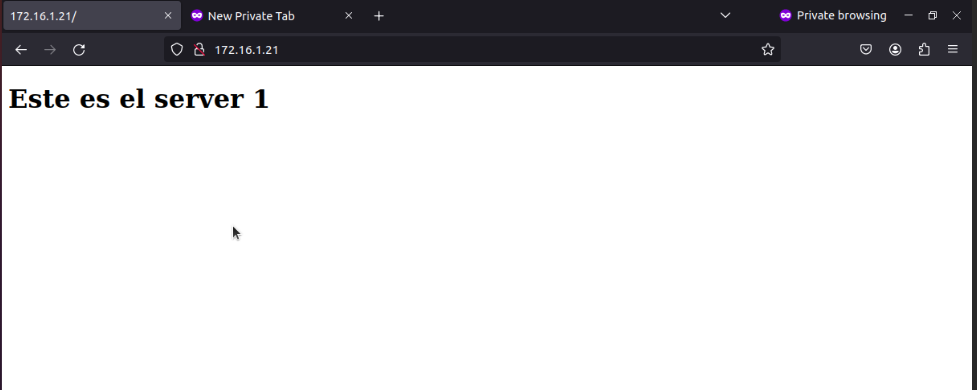


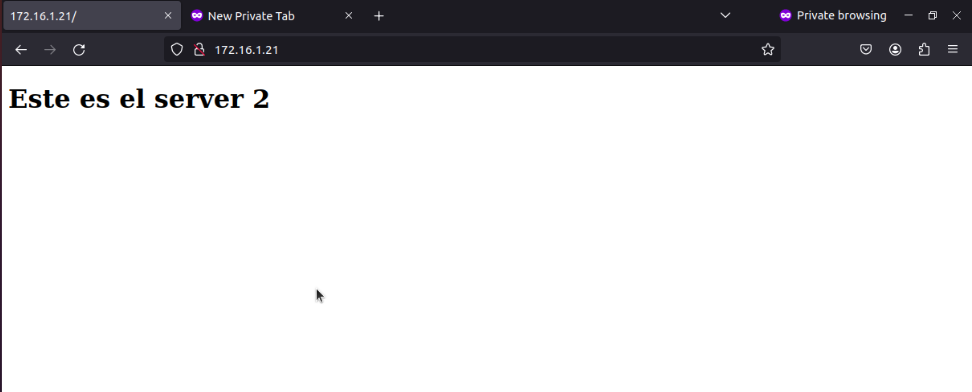
# Funcionamiento.

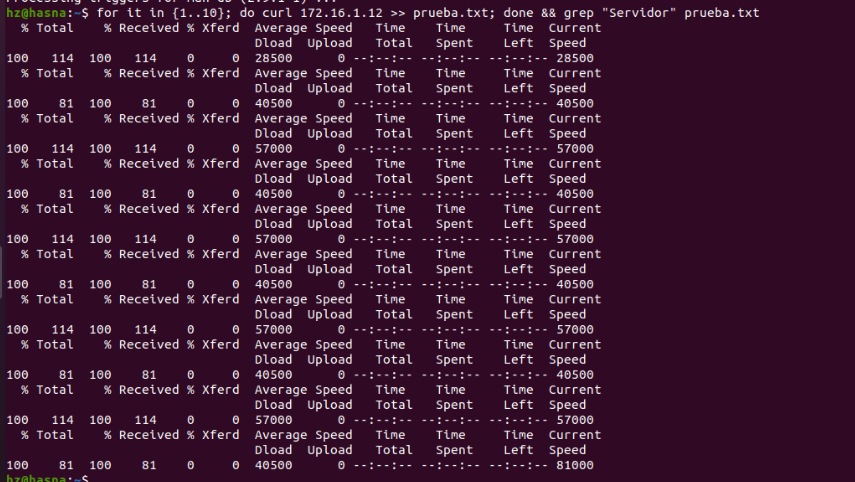




Cambiar la página por defecto por servidor 1 y servidor 2









# 

# Fuentes.

https://unpocodejava.com/2013/07/03/que-es-un-balanceador-de-carga-que-es-haproxy/

<https://help.clouding.io/hc/es/articles/4801456483356-Balancear-servicio-web-con-HAProxy-con-certificado-SSL-en-Ubuntu-22-04-LTS>

<https://www.ochobitshacenunbyte.com/2019/12/17/balanceo-de-carga-con-haproxy-en-ubuntu-18-04/>

<https://help.clouding.io/hc/es/articles/360019908839-C%C3%B3mo-configurar-un-servidor-de-balanceo-de-carga-Nginx-en-Ubuntu-20-04>

<https://help.clouding.io/hc/es/articles/4801456483356-Balancear-servicio-web-con-HAProxy-con-certificado-SSL-en-Ubuntu-22-04-LTS>