



# TELECOMUNICACIONES 3

## PROYECTO BALANCEADOR DE CARGA CON APACHE MOD\_PROXY\_BALANCER

*Anyelo Anduquia, Luis Fernando Arboleda Hurtado*



## INTRODUCCIÓN:

Implementación de un sistema de balanceo de carga utilizando tecnologías de virtualización y servicios web. El objetivo principal es distribuir de manera equitativa las peticiones de los clientes entre múltiples servidores web

## PROBLEMA:

Cuando un sitio web recibe muchas visitas simultáneas, un solo servidor puede saturarse. Esto genera lentitud, interrupciones y caída del servicio. El proyecto parte de este problema común y plantea como solución un sistema distribuido donde múltiples servidores trabajen de forma coordinada.



## ALTERNATIVAS DE SOLUCION:

- `mod_proxy_balancer lbmethod (byrequests)`: Distribuye equitativamente según la cantidad de solicitudes realizadas. Es el método por defecto.
- `mod_proxy_balancer lbmethod (bytraffic)`: Distribuye según el volumen de datos transferidos
- `Sticky Session`: Especializada un mismo usuario siempre sea atendido por el mismo servidor backend durante toda su sesión.

Se eligió `Apache mod_proxy_balancer lbmethod (byrequests)` por su simplicidad y amplia documentación.

# HERRAMIENTAS UTILIZADAS

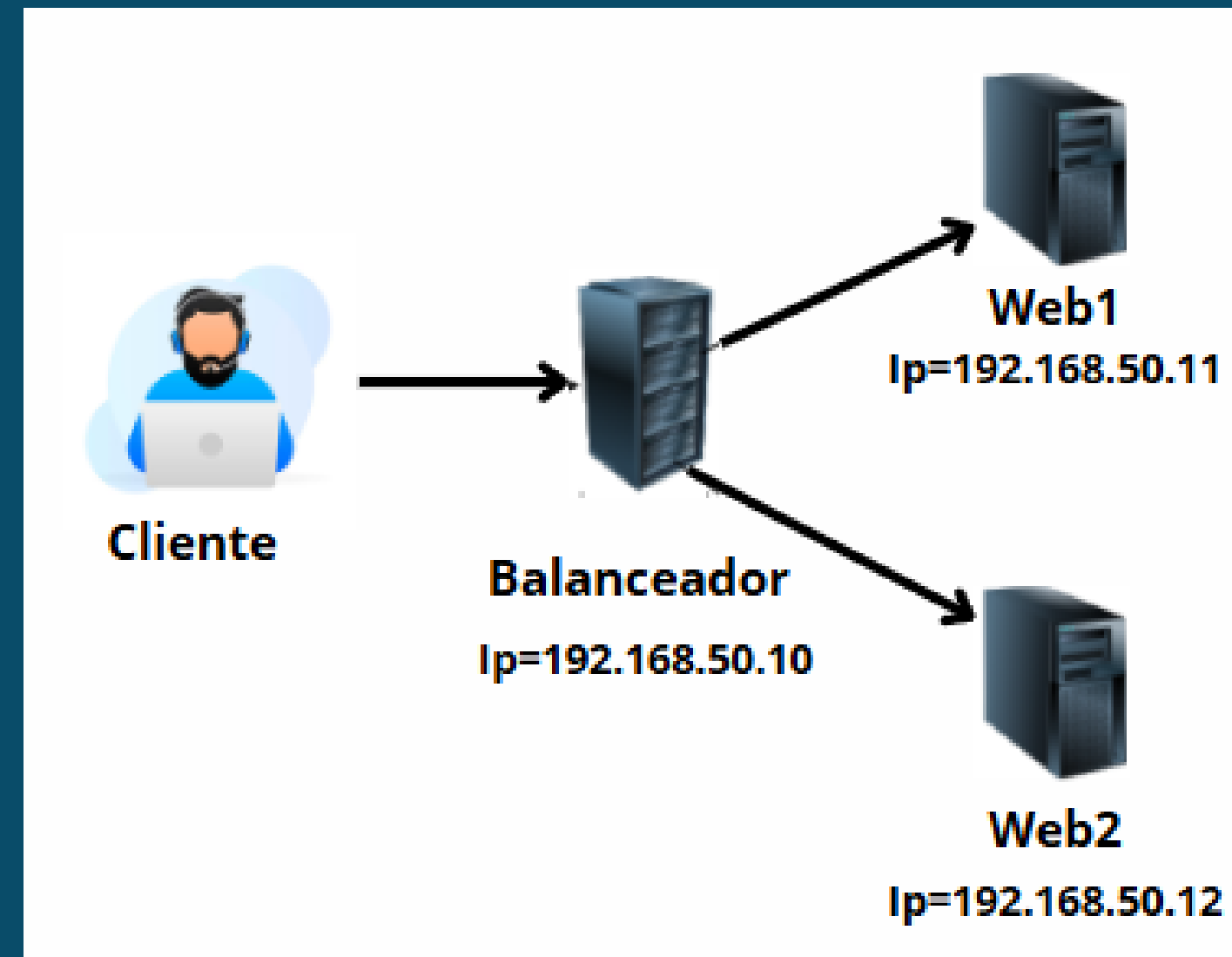
- Apache HTTP Server: servidor web con capacidades de balanceo
- mod\_proxy\_balancer: módulo para balanceo de carga
- Vagrant + VirtualBox: para crear las máquinas virtuales
- Artillery: herramienta de pruebas de carga



# DISEÑO DE SOLUCION

Se configuraron tres máquinas virtuales:

- Balanceador: 192.168.50.10
- Web1: 192.168.50.11
- Web2: 192.168.50.12



# IMPLEMENTACION

## **a. Configuración de los servidores backend**

En cada máquina Web1 y Web2 se instaló Apache y se creó una página HTML identificando al servidor:

```
sudo apt install apache2 -y  
sudo mkdir -p /var/www/web1
```

Se instaló Apache en ambas máquinas y se creó una página HTML personalizada con un timestamp dinámico. Cada sitio fue habilitado con un VirtualHost, y se aseguraron permisos de acceso.

## b. Configuración del balanceador

En la máquina balanceador se instalaron y activaron los siguientes módulos de Apache:

`proxy`, `proxy_http`, `proxy_balancer`, `lbmethod_byrequests`, `status`, y `headers`.

En la configuración del archivo `balanceador.conf` se definió el clúster como `balancer://webcluster`, especificando los servidores backend y los parámetros de failover, método de balanceo, y chequeos de estado (`ProxySet hcmethod=GET hcuri=/`).



## PRUEBAS

Se realizaron pruebas con Artillery simulando distintos niveles de tráfico:

- Carga sostenida (50 usuarios/ 30 seg)
- Pico de tráfico (100 usuarios/ 60 seg)
- Estrés máximo (200 usuarios/ 30 seg)

Las respuestas se analizaron para evaluar tiempo de respuesta, errores, y distribución de carga entre servidores.

# PRUEBAS

## Load summary 📊

**13.500**

vusers created

**13.160 completed**

**97.48%**

**340 failed**

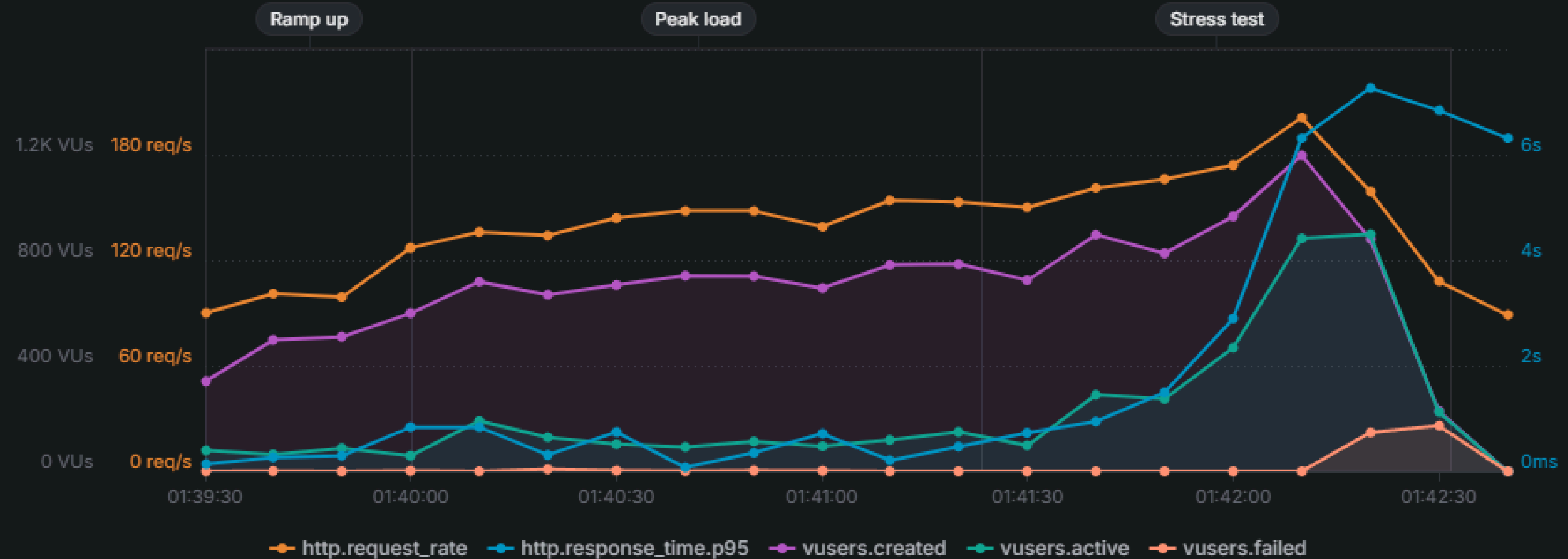
**2.52%**

**139**

average req/s

**201**

peak req/s



## DISCUSION DE LAS PRUEBAS

La integración de servicios adicionales como MySQL, SSL y APIs REST provocó cambios importantes:

- Redirecciones incorrectas al activar HTTPS
- Aumento en la carga del balanceador por conexiones a base de datos
- Fallos de rutas en APIs al no estar gestionadas por el proxy

Estos problemas exigieron ajustes en las configuraciones del proxy, los virtual hosts y los parámetros del clúster.

## DISCUSION DE LAS PRUEBAS

Las pruebas confirmaron que las solicitudes eran distribuidas adecuadamente entre Web1 y Web2, incluso bajo escenarios de alta concurrencia. Al detener un servidor, se observaron errores que permitieron analizar el comportamiento del balanceador frente a fallos.

## CONCLUSIONES

- La arquitectura balanceada es clave para sistemas que buscan alta disponibilidad y tolerancia a fallos.
- Integrar servicios adicionales como MySQL y SSL requiere ajustes adicionales y pruebas cuidadosas.

# REFERENCIAS

[1] “Configure Apache load balancer with mod proxy - LogicalDOC Community Wiki”. LogicalDOC Community Wiki:

[https://wiki.logicaldoc.com/wiki/Configure\\_Apache\\_load\\_balancer\\_with\\_mod\\_proxy](https://wiki.logicaldoc.com/wiki/Configure_Apache_load_balancer_with_mod_proxy)

[2] “Artillery”. Artillery:

[https://app.artillery.io/share/sh\\_1597275c5d5c057dff03fcb5556c2d5cf969cd1957de2094ecdcae504417c0db,](https://app.artillery.io/share/sh_1597275c5d5c057dff03fcb5556c2d5cf969cd1957de2094ecdcae504417c0db)

[3] “mod\_proxy\_balancer - Apache HTTP Server Version 2.4”. Welcome! - The Apache HTTP Server Project:

[https://httpd.apache.org/docs/current/mod/mod\\_proxy\\_balancer.html](https://httpd.apache.org/docs/current/mod/mod_proxy_balancer.html)