

BALANCEO DE CARGAS DE SERVIDORES WEB USANDO NGINX Y ARTILLERY

JUAN SEBASTIAN VARONA
JHARLING JHOAN RODRIGUEZ
ANA SOFIA LOPEZ LOPEZ
JUAN SEBASTIÁN LOSADA
JULIÁN DAVID VELÁSQUEZ

ÍNDICE DEL DOCUMENTO

- INTRODUCCIÓN
- ALTERNATIVAS DE SOLUCION
- IMPLEMENTACION
- PRUEBAS
- CONCLUSIONES

1 – INTRODUCCIÓN

INTRODUCCIÓN



RECONOCIMIENTO

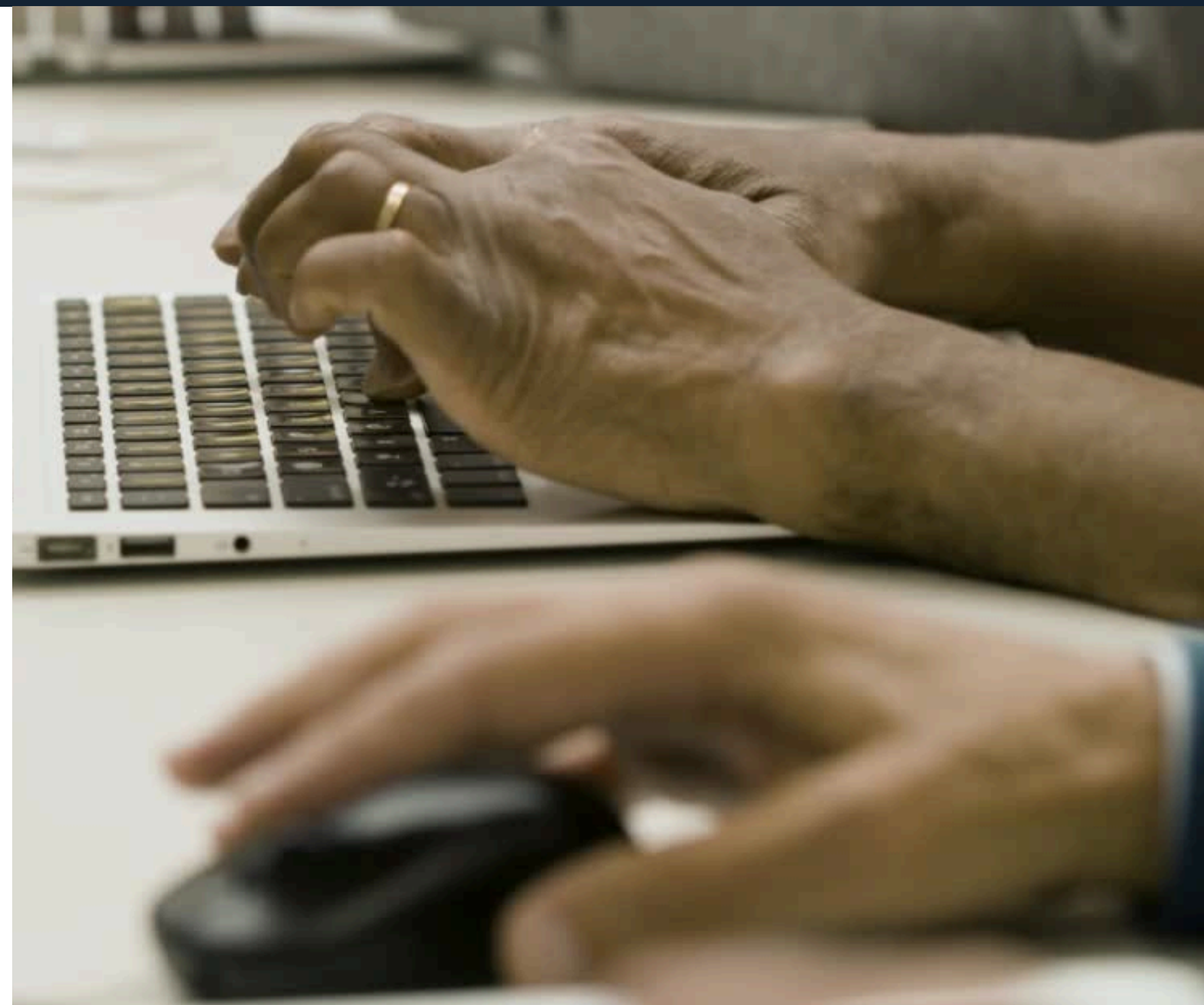
Existe una cantidad significativa de entornos web donde las personas deben acceder eficazmente cuando lo soliciten.

El conflicto al que nos estamos presentando está relacionado con la gestión eficaz de peticiones que se realizan a un solo servidor en un entorno de **alto tráfico**.

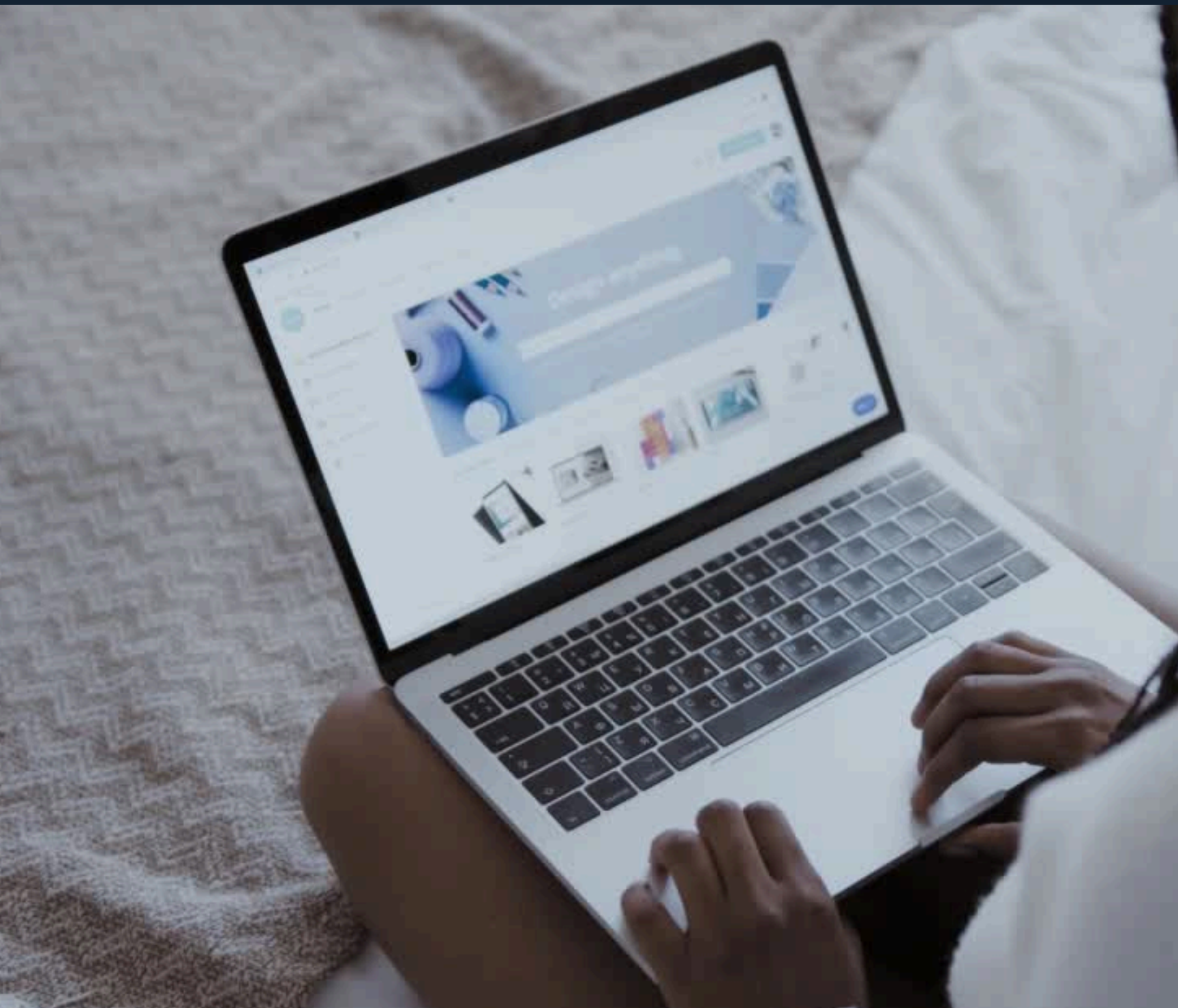


ANÁLISIS

En este contexto, la cantidad de peticiones en tiempo real genera sobrecarga en los servidores dificulta el manejo del volumen de peticiones y resulta en tiempos de respuesta lentos y fallos en el sistema.

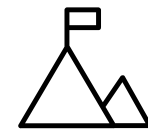


INTRODUCCIÓN



PLANEACIÓN

La solución propuesta incluye un balanceador de carga configurado con NGINX y tres servidores backend, todos ellos gestionados en un entorno de máquinas virtuales usando Vagrant. Para las pruebas de rendimiento y carga, se empleará Artillery.



EJECUCIÓN

Por esto, se explicara paso a paso el como lograr que la solución propuesta con NGINX sea un método para balancear las cargas y asi tener un mejor rendimiento del aplicativo

2 — ALTERNATIVAS DE SOLUCIÓN

ALTERNATIVAS DE SOLUCIÓN

BALANCEADOR DE CARGAS

SOLUCION NO. 1

NGINX

NGINX es un servidor web/proxy inverso ligero de alto rendimiento. Este también puede actuar como un balanceador de carga que distribuye las solicitudes entrantes a múltiples servidores.

SOLUCION NO. 2

MOD_PROXY_BALANCER

En el caso de Apache, el módulo `mod_proxy_balancer` permite al servidor web actuar como un proxy inverso reenviando las solicitudes a otro servidor y devolviendo la respuesta al cliente.

SOLUCION NO. 3

AWS ELASTIC LOAD BALANCING

Es un servicio que distribuye automáticamente el tráfico de aplicaciones entrantes entre varios destinos y dispositivos virtuales en una o varias zonas de disponibilidad.

Tabla Comparativa			
Característica	Nginx	mod_proxy_balancer	AWS Elastic Load Balancer
Ventajas			
Rendimiento	Alta eficiencia y baja latencia	Suficiente para aplicaciones de tamaño medio	Optimizado para alta disponibilidad y escalabilidad automática
Configuración	Flexible y altamente configurable	Integrado en Apache, fácil de configurar	Fácil de usar y gestionar desde la consola AWS
Seguridad	Buenas características de seguridad	Compatible con la seguridad de Apache	Integrado con AWS WAF y otros servicios de seguridad AWS
Costos	Open source, sin costos adicionales	Incluido con Apache, sin costos adicionales	Costos basados en uso, puede ser más costoso a largo plazo
Integración	Compatible con muchas aplicaciones y servicios	Fácil integración con aplicaciones basadas en Apache	Integración profunda con otros servicios AWS
Desventajas			
Complejidad	Puede ser complejo para usuarios novatos	Limitado en funcionalidad comparado con Nginx	Puede ser complejo para configuraciones avanzadas
Escalabilidad	Requiere configuración manual para escalar	Menos eficiente en escalabilidad	Escalabilidad automática pero dependiente de la infraestructura AWS
Soporte	Comunidad activa, pero soporte comercial limitado	Principalmente soporte de la comunidad Apache	Soporte profesional a través de AWS
Dependencia	Requiere conocimientos en Linux/Unix para configuraciones avanzadas	Depende de Apache, no es un producto independiente	Dependencia de la infraestructura de AWS
Mantenimiento	Requiere mantenimiento y actualizaciones manuales	Requiere mantenimiento junto con Apache	Mantenimiento y actualizaciones gestionadas por AWS

NGINX

SOLUCIÓN USADA



NGINX destaca como balanceador de carga debido a su flexibilidad y capacidad de personalización, su rendimiento superior en situaciones de alta carga, sus funcionalidades avanzadas como el balanceo de carga HTTP y TCP/UDP, así como su costo más bajo y el control directo sobre la infraestructura en comparación con opciones gestionadas en la nube como ELB de AWS, lo que lo hace ideal para aplicaciones que requieren ajustes específicos, alto rendimiento y un control más directo sobre la infraestructura.

<i>Característica</i>	<i>Artillery Cloud</i>	<i>Apache JMeter</i>
Ventajas		
<i>Facilidad de uso</i>	Intuitivo, fácil de configurar y usar	Amplia documentación y tutoriales disponibles
<i>Escalabilidad</i>	Buen rendimiento en pruebas distribuidas en la nube	Soporta pruebas distribuidas y de gran escala
<i>Soporte de protocolos</i>	Soporta HTTP, WebSocket, Socket.io, Kinesis	Amplio soporte de protocolos (HTTP, HTTPS, FTP, JDBC, JMS, etc.)
<i>Flexibilidad y personalización</i>	Configuración sencilla, soporte para scripts JavaScript	Alta personalización mediante scripting y plugins
<i>Integración CI/CD</i>	Fácil integración con herramientas de CI/CD como Jenkins, GitHub Actions	Buenas opciones de integración con Jenkins, Bamboo, etc.
<i>Análisis y reportes</i>	Reportes claros y visuales, fáciles de interpretar	Reportes detallados, amplia capacidad de análisis
<i>Costo</i>	Planes gratuitos y de pago disponibles	Open-source, gratuito
<i>Soporte y comunidad</i>	Soporte de Artillery.io, comunidad en crecimiento	Gran comunidad y soporte activo
Desventajas		
<i>Facilidad de uso</i>	Interfaz limitada comparada con JMeter	Curva de aprendizaje más pronunciada
<i>Escalabilidad</i>	Dependencia de la infraestructura de Artillery	Configuración de pruebas distribuidas puede ser compleja
<i>Soporte de protocolos</i>	Menos protocolos soportados comparado con JMeter	Algunos protocolos requieren plugins adicionales
<i>Flexibilidad y personalización</i>	Menos opciones de personalización	Configuración avanzada puede ser compleja
<i>Integración CI/CD</i>	Menos popularidad puede limitar integración con algunas herramientas	Requiere configuración adicional para integración
<i>Análisis y reportes</i>	Menos opciones de personalización en los reportes	Requiere configuración para obtener reportes detallados
<i>Costo</i>	Funcionalidades avanzadas requieren suscripción paga	Puede incurrir en costos indirectos por tiempo de configuración y mantenimiento
<i>Soporte y comunidad</i>	Comunidad más pequeña comparada con JMeter	El soporte oficial puede ser costoso

ARTILLERY

SOLUCIÓN USADA PARA PRUEBAS

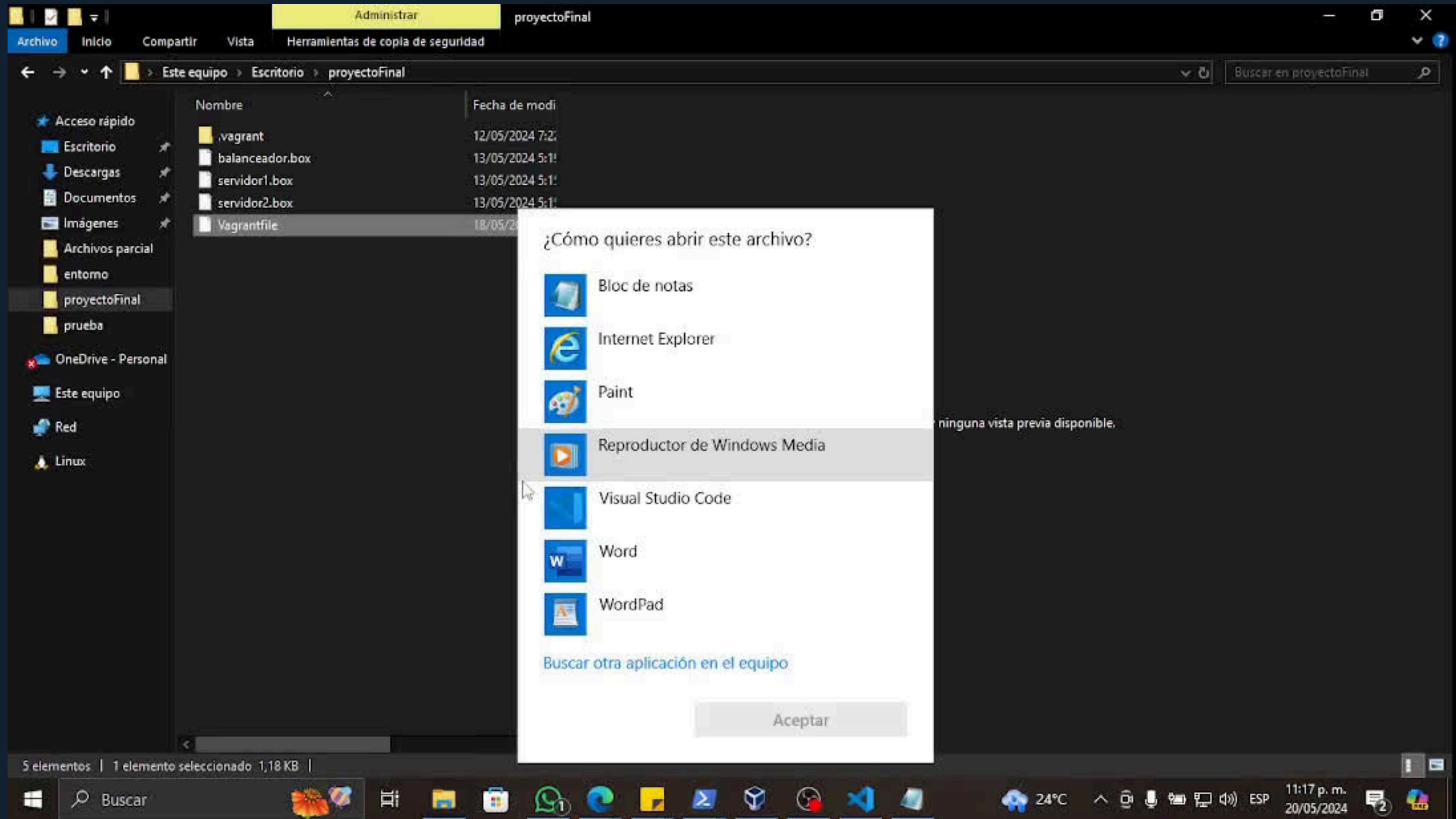


La elección de Artillery Cloud para nuestras pruebas de balanceo de cargas se justifica por su capacidad para simular cargas reales de usuarios concurrentes, su facilidad de configuración y ejecución, así como por su capacidad para proporcionar monitorización y análisis detallados del rendimiento de nuestra aplicación.

Aun así, queremos recalcar que otra alternativa para pruebas como Apache JMeter

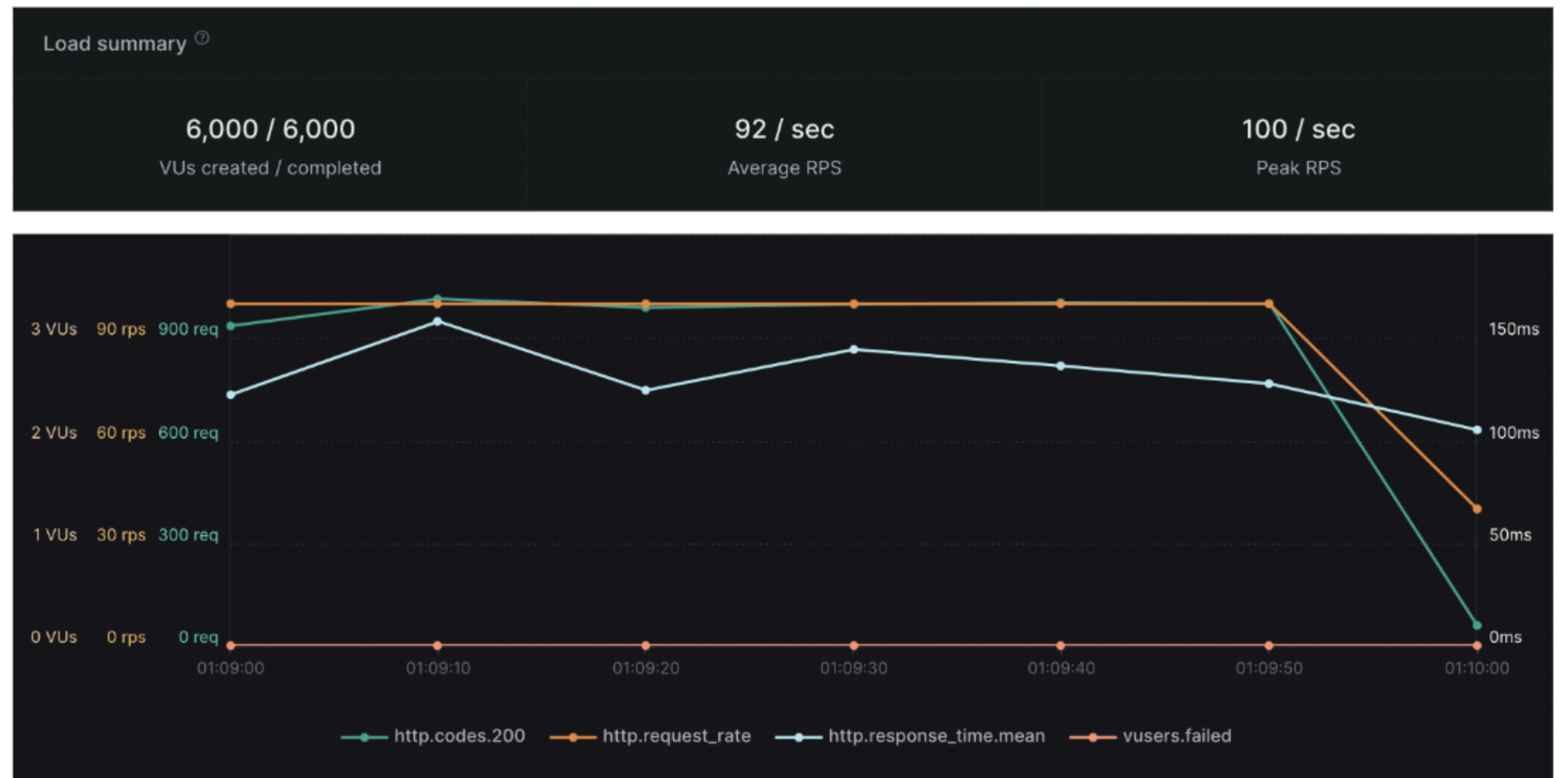
3 – IMPLEMENTACION Y PRUEBAS

VIDEOS



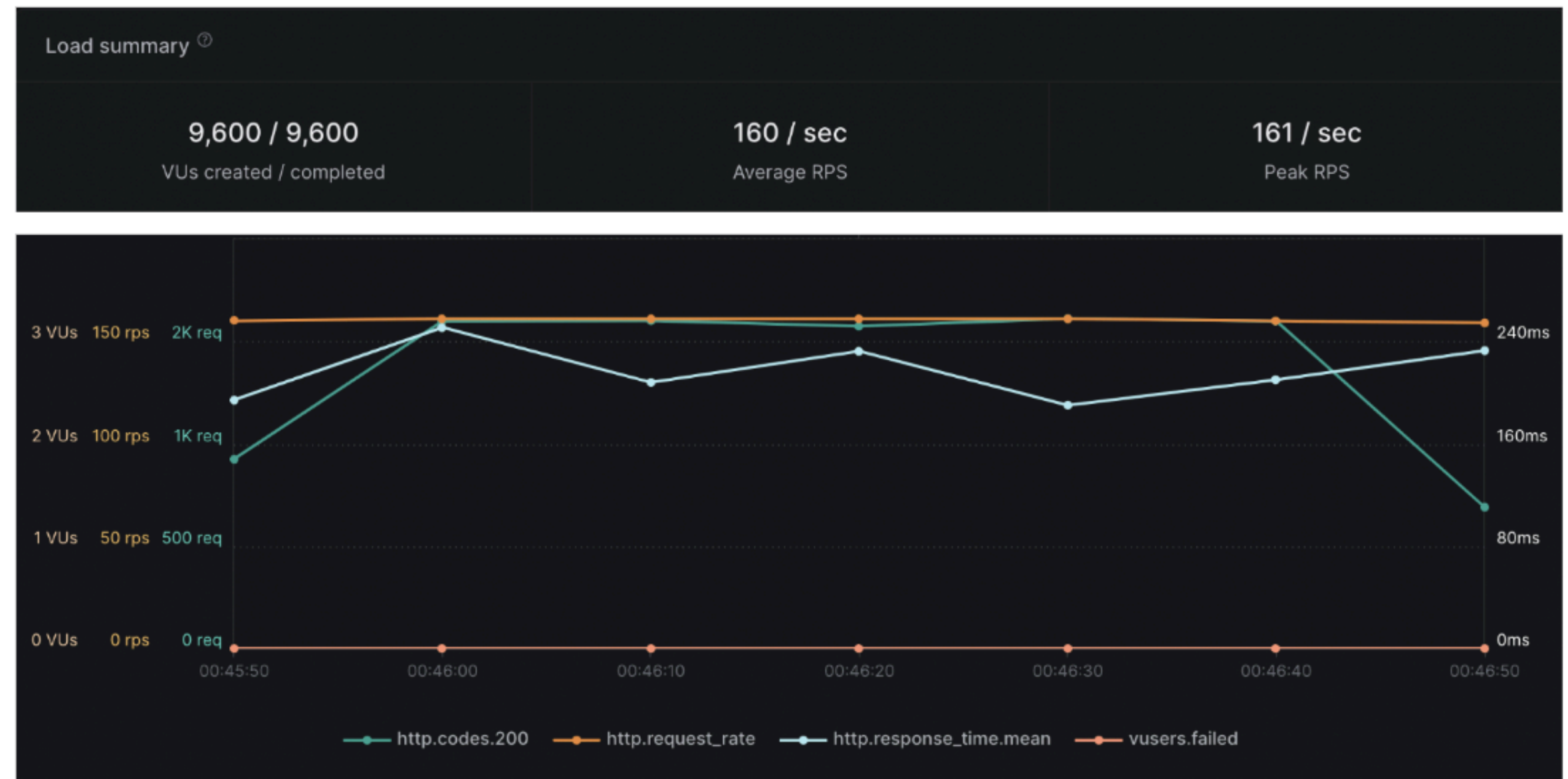
PRUEBAS

2 servidores



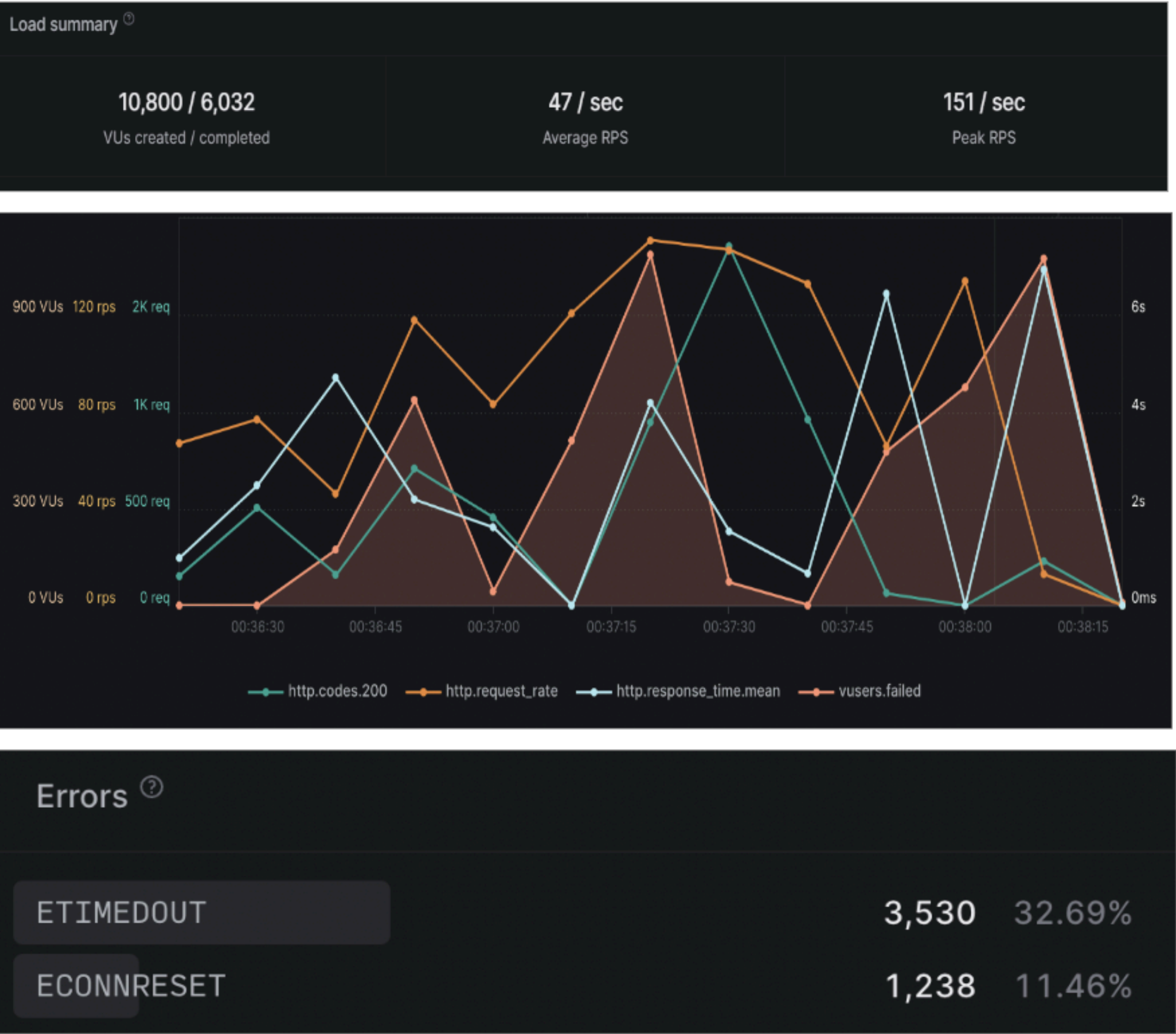
PRUEBAS

2 servidores



PRUEBAS

2 servidores



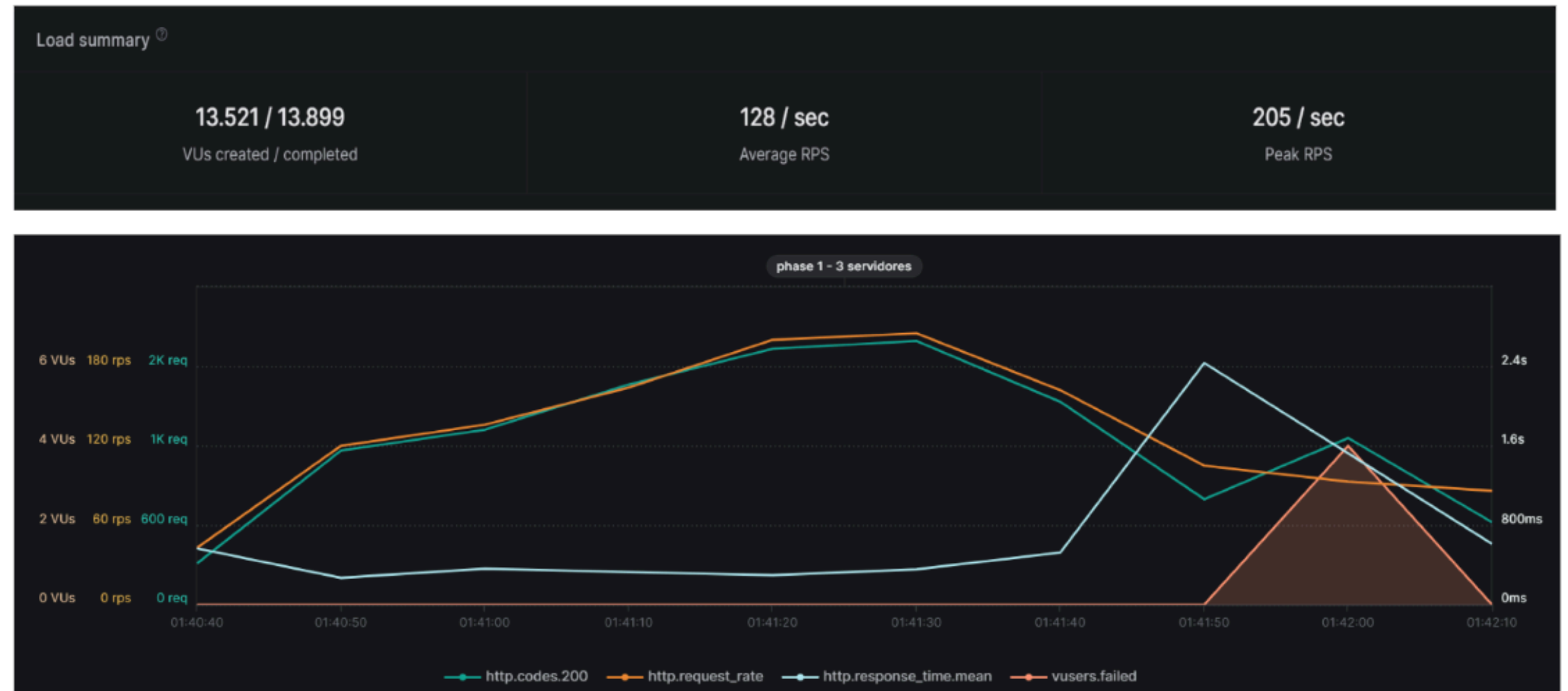
PRUEBAS

3 servidores



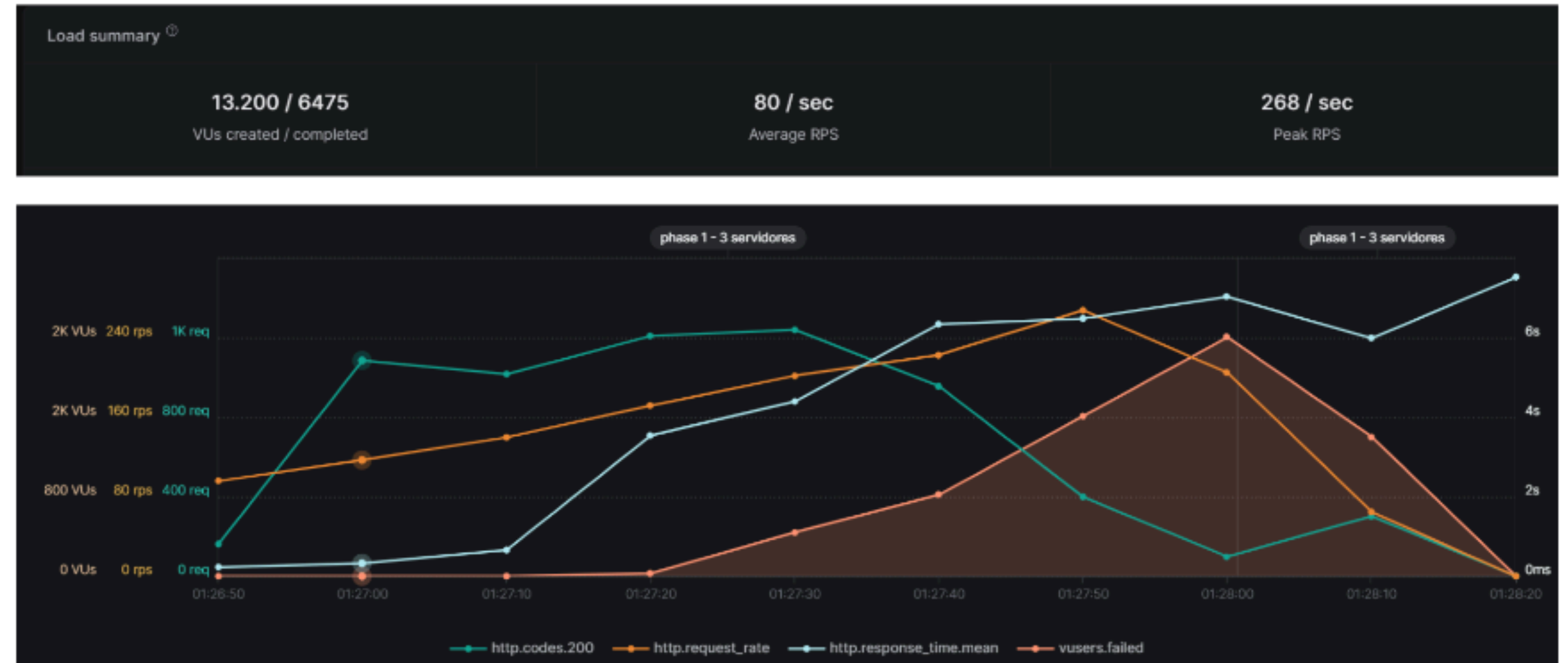
PRUEBAS

3 servidores



PRUEBAS

3 servidores



4 – CONCLUSIONES

CONCLUSIONES

1

El balanceador de carga con tres servidores demostró una mayor capacidad para gestionar peticiones en comparación con dos servidores, gracias a una distribución de recursos más amplia.

2

Se observó que los servidores tienden a tener inicios más lentos, pero luego normalizan la cantidad de peticiones manejadas en unos 10 segundos.

3

El tiempo de respuesta tiende a ser relativamente lineal en situaciones de estrés bajo/moderado, pero fluctúa significativamente en situaciones de estrés alto, con picos pronunciados.

4

Se recomienda que los servidores operen a un 60%-70% de su capacidad máxima para mantener un rendimiento eficiente, dejando un margen para picos inesperados y minimizando errores.

ХАНА
ХОВ