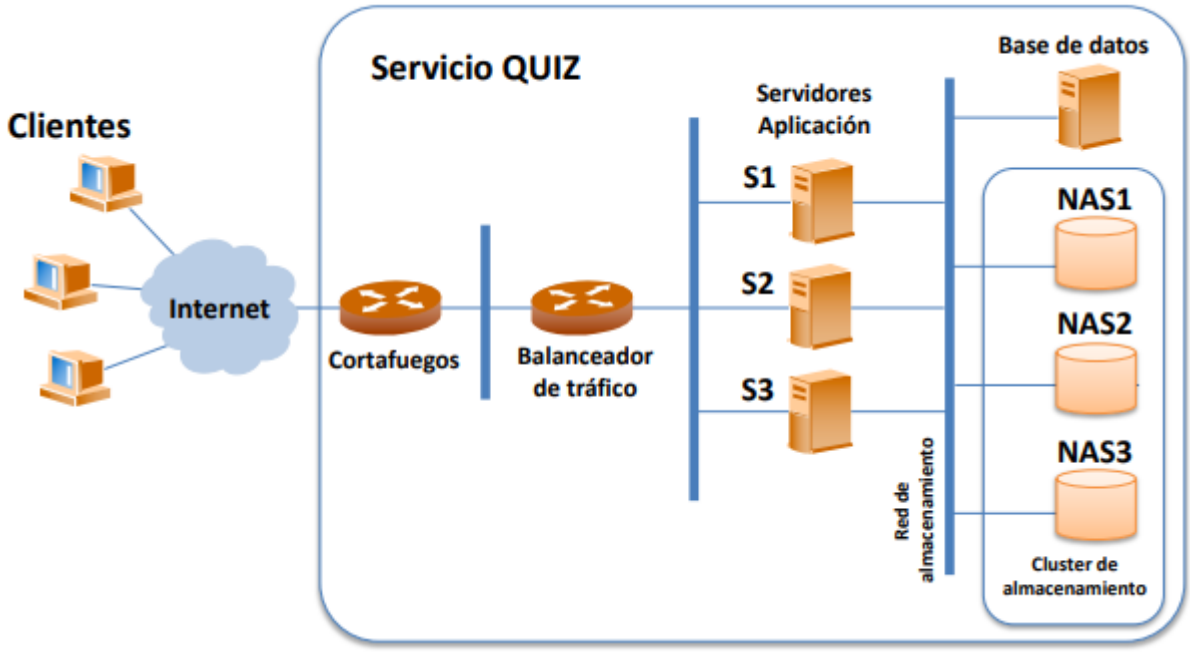
Entrega Práctica Creativa 2



Alejandro Aguilera Alcalde
Pablo Barba Sabín

Decisiones de Diseño e Implementación

Para la realización de esta práctica se ha utilizado el laboratorio con sistema operativo Ubuntu.
El escenario virtual consta de varias máquinas virtuales ligeras (lxc) que desempeñan cada una su función específica.

Partes Opcionales

El firewall debe permitir únicamente el acceso mediante ping y al puerto 80 de TCP de la dirección balanceador de tráfico. Cualquier otro tráfico debe estar prohibido.

Se ha creado un firewall adicional para implementar esta mejora, que también permite el tráfico en el puerto 8001 del balanceador para poder mostrar en el navegador las estadísticas de balanceo de carga.
Como consecuencia de esta mejora, la única forma de acceder a la aplicación a través del navegador es con la dirección del balanceador de carga, que se encarga de decidir a qué servidor redirige la petición.

Crear el procedimiento para añadir y configurar un nuevo servidor front-end adicional

Para implementar esta mejora hemos modificado pc2.xml para que se cree y se añada a la red un nuevo servidor al iniciar el escenario y después se ha configurado de forma análoga al resto de servidores.

Realizar mejoras en el algoritmo de balanceo de carga

Para implementar esta mejora hemos editado el fichero de configuración del haproxy para que en vez de utilizar el algoritmo round-robin utilice el algoritmo leastconn que envía las peticiones al servidor con menor número de conexiones.

Indicar cómo cambiaría el despliegue en caso de desplegarlo en OpenStack,Amazon AWSo GoogleCloud:

Para cambiar el despliegue en caso de usar GoogleCloud se crearía un cluster de kubernetes similar a lo usado en la práctica 6 de la asignatura teniendo en cuenta el número de nodos que son necesarios.

Ahora ya tendríamos un cluster de ordenadores donde podemos desplegar Pods, formados por contenedores Docker. Más tarde hay que realizar los distintos Dockerfiles y usar los contenedores que se usarán en el despliegue del cluster.

En la práctica 6 se usa este despliegue para crear una tienda online formada por múltiples microservicios y en nuestro caso desplegaríamos nuestra página web para jugar a QUIZ_2021.

Análisis de Puntos Débiles de la Arquitectura

Un punto débil de la arquitectura es la existencia de un único balanceador de carga, que en caso de fallo haría que la aplicación dejara de estar disponible. Una solución a esto sería añadir otro balanceador de carga para evitar que haya un único punto de fallo.

En cuanto a la escalabilidad, la aplicación es escalable ya que hemos demostrado que se puede añadir más servidores, pero en caso de un gran tráfico y para una mejor escalabilidad sería recomendable emplear una plataforma de cloud online para el despliegue de la aplicación. Por ejemplo, se podría utilizar Google Cloud y Kubernetes, como se explica en el punto anterior.