

Implementación de Balanceadores de Carga, Distribución de Sesiones Web y Contenedores de Aplicaciones.

Objetivos:

- Implementar un balanceador de carga con el algoritmo Round-robin con tolerancia a fallo.
- Configurar un balanceador de carga para aplicaciones que utilicen sesiones para la trazabilidad de los usuarios (cookies y sesiones).
- Implementar contenedores de aplicaciones en una solución informática.

Desarrollo de la práctica.

Los estudiantes en grupo de 2, realizarán lo requerido en el desarrollo de la práctica.

Forma de Entrega:

La entrega de la práctica se debe realizar un reporte con lo trabajado, **conjuntamente con una presentación del proyecto en (10 minutos) con un vídeo bajo la plataforma youtube con su respectivo reporte**; la presentación debe cubrir cada uno de los requerimientos solicitados en el proyecto trabajado. La práctica será entregada según la fecha indicada en la PVA.

Desarrollo

Para nuestra práctica, vamos a utilizar como base la práctica llamada “**Uso Docker y Docker Compose**”, que contiene la aplicación de clone de Mockup API Server – Spring Boot. Nos interesa poder implementar un esquema de alta disponibilidad utilizando balanceo de carga en nuestra aplicación.

Para simplificar la implementación, estaremos utilizando como balanceador HAProxy en una maquina virtual con una IP pública asignada y un host (tipo A) de un dominio DNS que tengan disponible (namecheap.com o name.com). En la maquina virtual, estaremos subiendo varias instancias del software requerido (3 como mínimo), utilizar Docker y Docker Compose para la configuración del esquema completo de la práctica (necesario para la corrección). El algoritmo de balanceo será Round Robin. La aplicación debe mostrar el puerto o el id de la instancia para fines de validar el balanceo. Ver ilustración 2.

El balanceador estará configurado en el modo **Terminación SSL** (SSL/TLS), donde la comunicación entre el cliente y el balanceador se realiza encriptada vía HTTPS y desde el balanceador. Pueden ver información [aquí](#) y proyecto para implementarlo [aquí](#). La generación del certificado asociado al host disponible, pueden utilizar <https://letsencrypt.org/>. Las peticiones al puerto 80 debe redireccionar al puerto seguro 443. Ver ilustración 1.

HAProxy SSL Termination (HTTPS)

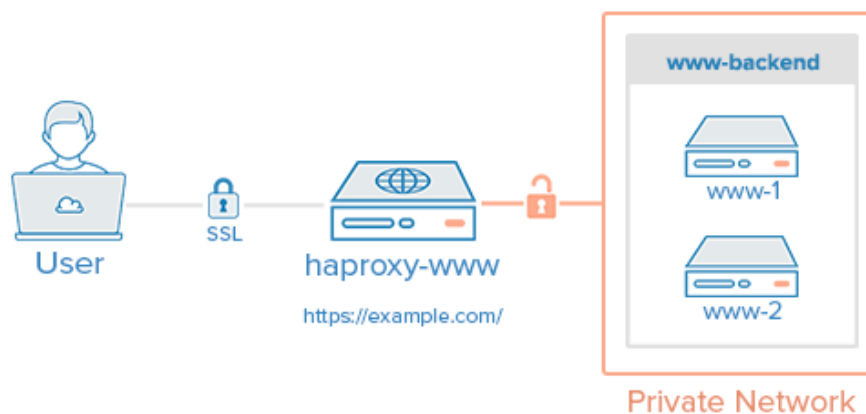


Ilustración 1: Diagrama de Balanceador en Modo Terminación SSL

Para la distribución de las sesiones entre los servidores (Spring Session como ejemplo), pueden implementar un servidor basado en “in memory data grid” IMDG que consideren, como ejemplos:

- Hazelcast.
- Memcached.
- Redis.
- Apache Ignite.

El esquema solicitado debe ser de la siguiente:

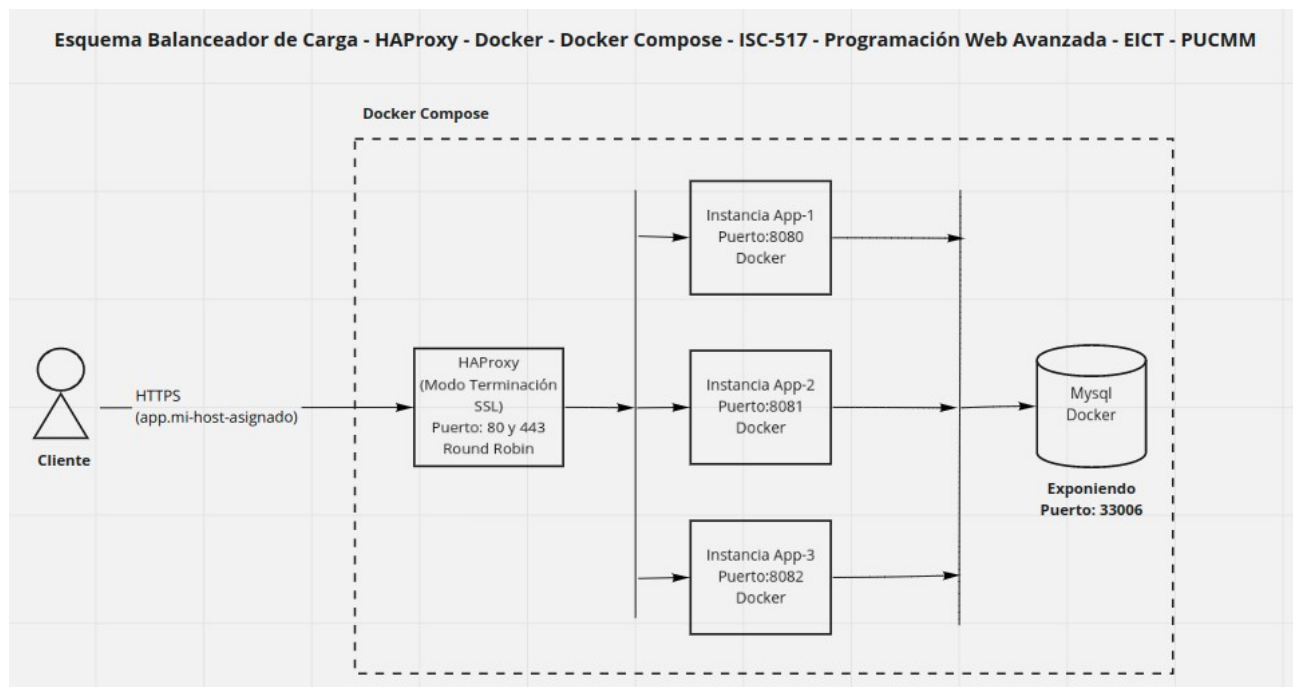


Ilustración 2: Esquema Arquitectura

Para sortear las limitaciones de las 4 horas de trabajo en AWS Academy, es necesario crear un vídeo mostrando la implementación de lo solicitado. En el repositorio debe estar disponible la configuración y los DockerFile y Docker Compose.