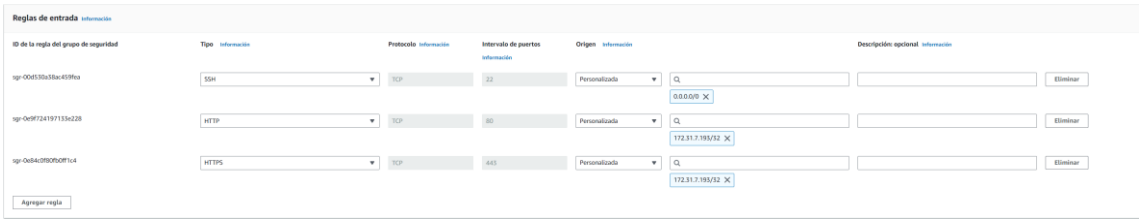
Entrevista técnica

Para empezar con el problema, he tenido que buscar cuales son las mejores opciones para implementar el sistema. Para este caso, he utilizado máquinas virtuales de aws, y un balanceador de carga hecho con nginx. Para la máquina virtual con la aplicación wordpress, he utilizado una máquina con la instalación base a hecha, ya que, al no requerir funciones complejas, he decidido optimizar mi tiempo. Para la máquina virtual con el balanceador si que he instalado manualmente nginx porque es una tarea muy simple.

El primer problema es que no sabía usar nginx, por lo que he tenido que aprender su funcionamiento, y la forma de desplegarlo. Esto lo he hecho a partir de la documentación oficial: https://docs.nginx.com/nginx-management-suite/, algunas páginas dónde explican que es nginx cómo: https://kinsta.com/es/base-de-conocimiento/que-es-nginx/ y de videos que muestran cómo se ha de configurar e instalar https://www.youtube.com/watch?v=YlD2FvLCOLw&t=79s.

Para conseguir el objetivo que se me plantea, y tras aprender cómo se utiliza nginx, he empezado simplificando el problema. Para esto, he ejecutado las dos máquinas virtuales, y he conectado el balanceador a la aplicación sin ninguna restricción.

Tras conseguir que todo funcione como deseado, he empezado a solucionar los diferentes problemas que se plantean por separado. Primero he restringido el tráfico a la máquina virtual con la aplicación. Para esto, he añadido en el grupo de seguridad de aws EC2 una restricción dónde únicamente se puede acceder a esa máquina a través de la ip privada de la maquina con nginx. Es la privada y no la pública porque están conectadas a la misma red privada de aws EC2. Para comprobar esto, si estuviese abierto a todo internet, se debería de poder acceder a la maquina virtual con el wordpress a través del siguiente link, pero no se puede: https://ec2-18-201-194-242.eu-west-1.compute.amazonaws.com .



En cuanto al acceso ssh seguro a las máquinas, es bastante sencillo ya que cuando lanzas las instancias aws te da la opción de crear un par de claves para asegurar esto.

Siguiendo el entregable, tendremos una url para acceder al wordpress a través del balanceador: https://ec2-34-241-56-176.eu-west-1.compute.amazonaws.com y las 2 conexiones ssh, a la maquina nginx mediante: **ssh -i clave.pem ec2-user@ec2-34-241-56-176.eu-west-**

1.compute.amazonaws.com y otra a la maquina con wordpress mediante **ssh -i clave.pem bitnami@ec2-18-201-194-242.eu-west-1.compute.amazonaws.com**

En cuanto a la configuración del balanceador, dejo una captura de lo que creo que es más relevante, pero se puede ver entero en el nginx.conf que hay en el git.

```
    upstream backend{
            server ec2-18-201-194-242.eu-west-1.compute.amazonaws.com;
    }

server {
    listen       80;
    listen       [::]:80;
    listen       443 ssl;
    listen       [::]:443 ssl;

    server_name  ec2-34-241-56-176.eu-west-1.compute.amazonaws.com;
    root         /usr/share/nginx/html;

    location / {
            proxy_pass http://backend;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Proto $scheme;
    }

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    ssl_certificate "/etc/pki/nginx/server.crt";
    ssl_certificate_key "/etc/pki/nginx/server.key";
    ssl_session_cache shared:SSL:1m;
    ssl_session_timeout  10m;
    ssl_ciphers PROFILE=SYSTEM;
    ssl_prefer_server_ciphers on;


    error_page 404 /404.html;
    location = /404.html {
    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
    }
}
```

Uno de los problemas de mi estructura es que el certificado ssl no está verificado, por lo que cuando accedes el navegador alerta al usuario de que la configuración no es privada. Esto se puede solucionar fácilmente reemplazando los certificados de nginx por unos certificados.