Prueba técnica

Para realizar esta pequeña prueba técnica, el primer paso realizado ha sido la elección de Amazon Web Services (AWS) para ejecutar las máquinas virtuales. Una vez elegido este servicio, me informé ligeramente acerca de los servicios que proporciona AWS. El servicio mediante el que se implementará la solución es EC2, que es el que nos permite utilizar servidores virtuales en la nube. Los contenidos de este servicio son similares a los ofrecidos por la plataforma Azure de Microsoft, con la que ya había trabajado previamente y por tanto ya tenía nociones básicas en creación y gestión de instancias virtuales.

Una vez en la consola de AWS, es necesario crear los elementos que nos permitan implementar la arquitectura solicitada (Cabe destacar que no se va a explicar la configuración especifica de cada elemento). El primer elemento a crear es una instancia en la cual instalaremos WordPress. Para ello nos conectaremos a la máquina ya sea mediante la consola del panel EC2 o mediante el SSH configurado al crear la máquina. Para poder ejecutar WordPress, es necesario instalar un servidor web Apache con soporte para PHP y MariaDB, también llamado servidor web LAMP. En el siguiente enlace, se encuentran los pasos a realizar para tener operativo este servidor en una instancia con el sistema operativo Amazon Linux 2.

<https://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/ec2-lamp-amazon-linux-2.html>

Una vez hecho esto, se ha seguido la siguiente guía para instalar y configurar WordPress en la máquina.

<https://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/hosting-wordpress.html>.

Tras realizar estos pasos ya tendremos un WordPress instalado por defecto. Sin embargo no podremos acceder a él debido a la configuración requerida para la máquina, por lo que es el momento de crear y configurar el balanceador de carga.

El balanceador de carga debe recibir las peticiones HTTP y HTTPS y redirigirlas al grupo destino, que en este caso es la aplicación WordPress. Es importante destacar que para configurar la redirección de peticiones HTTPS es necesario incorporar certificados a la configuración por lo que tan solo se ha realizado para peticiones HTTP. Una vez creado el balanceador, el ultimo paso es configurar debidamente los grupos de seguridad asociados a cada elemento. Este punto es muy importante ya que si no se realiza como es debido podemos encontrar fallos en el funcionamiento deseado.

El grupo de seguridad del balanceador de carga debe permitir como entrada todo el trafico IPv4 de tipo HTTP en el puerto 80. Como salida no es necesario restringir nada, con lo que se permitirá todo el trafico IPv4 en cualquier puerto. En el grupo de seguridad de la maquina WordPress, las reglas de salida se configurarán del mismo modo. En cuanto a las reglas de entrada, será necesario permitir el trafico HTTP por el puerto 80 tan solo del balanceador, por lo que debemos configurar como origen el grupo de seguridad creado para el balanceador de carga. También es necesario permitir el tráfico de tipo SSH de cualquier dirección IPv4 en el puerto 22. Por tanto mediante esta regla podremos conectarnos a la maquina por SSH y, al haber creado una clave de seguridad previa durante la creación de la máquina, tan solo podremos hacerlo si disponemos de las claves adecuadas.

Para conectar mediante SSH utilizaremos el archivo clavesSSH.pem mediante el siguiente comando:

ssh -i .\clavesSSH.pem ec2-user@13.38.56.105

Una vez configurado todo correctamente, ya es posible acceder a nuestra aplicación. Para ello utilizaremos la dirección DNS que nos proporciona AWS para nuestro balanceador de carga:

<application-load-balancer-867082386.eu-west-3.elb.amazonaws.com>

Bonus track

También se ha realizado la parte de ampliación. Para ello se ha instalado Terraform en la maquina mediante los tres simples comandos encontrados en esta guía: <https://medium.com/@rajkanchole1/steps-to-install-terraform-on-amazon-linux-15e6eb8763f2>. Una vez instalado, crearemos un nuevo directorio donde guardar los distintos archivos de configuración. En este directorio crearemos el archivo main.tf, que contendrá lo necesario para crear nuevas instancias. Tras esto tan solo tendremos que inicializar Terraform mediante terraform init, aprovisionar los recursos mediante terraform apply, y destruirlos mediante terraform destroy.

Por último para realizar estos pasos mediante una línea se ha creado un archivo deploy.sh que incluye los comandos init y apply.

Problemas encontrados

Durante la implementación de esta solución no se han encontrado problemas destacables. Ya que las guías de instalación eran bastante completas el principal problema encontrado ha sido entender la forma de configurar los distintos elementos de AWS, cosa que tampoco ha resultado muy costosa debido a la experiencia previa ya comentada anteriormente con Azure. También encontré el problema ya comentado de los certificados para HTTPS. La solución es sencilla ya que, al tratarse de una prueba, no hay necesidad de implementarlo ya que es posible establecer conexión por HTTP.

Debido a esto y, aunque en un principio pensaba que me resultaría mas costoso ya que siempre aparecen problemas inesperados en este tipo de implementaciones, me ha resultado muy sencillo realizar la prueba ya que tan solo he necesitado dedicar un par de días de trabajo.