

Despliegue de un arquitectura de EFS- EC2-MultiA-Z (AWS)

ALEJANDRO RUIZ CRESPO

INDICE

1.- Enunciado.....	
2.- Esquema Gráfico de la arquitectura.....	
3.- DOCUMENTACIÓN.....	
3.1.- EC2.....	
3.2.- EFS.....	
3.3.- Balanceador de carga.....	
3.4.- RDS.....	
3.5.- Página web (ficheros y objetivo)	
4.- Características.....	

1.- Práctica 4.4 (semana del 23 al 27 Enero): deployment of an architecture EFS-EC2-MultiAZ in the CCloud (AWS)

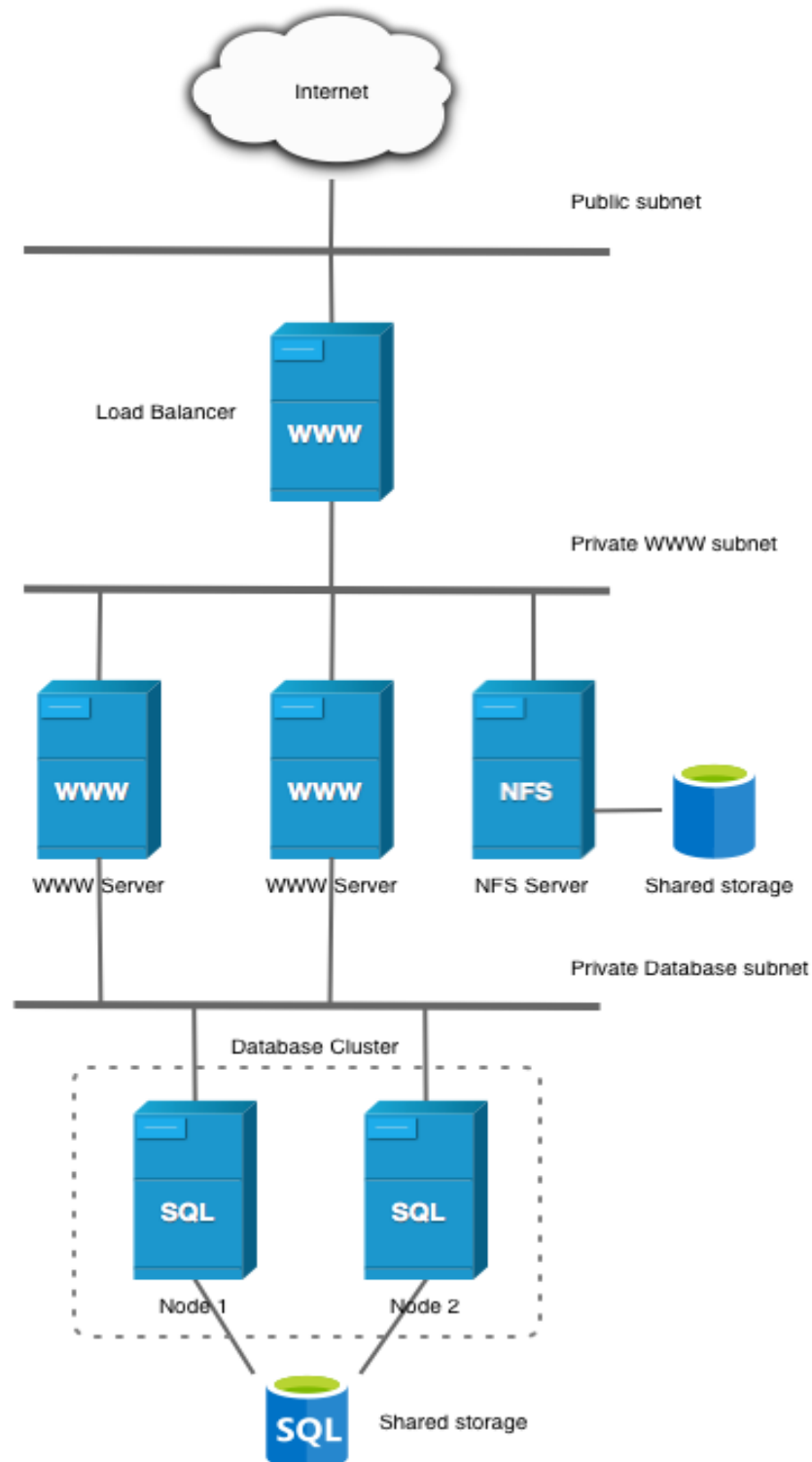
Objetivos de la práctica:

- High Avality
- Multi A-Z
- Alto rendimiento
- Alta seguridad

[VIDEO DE BASE](#)

2.- ESQUEMA GRÁFICO DE LA ARQUITECTURA

ESCENARIO : Clúster de servidores webs con balanceo de carga.



DOCUMENTACIÓN.

EC2 Y EFS

Primeramente crearemos dos máquinas EC2, las que estarán conectadas a un balanceador de carga y cada una de ellas en una región diferente utilizando Multi A-Z.

Con esto conseguiremos una alta estabilidad y disponibilidad debido a la desviación de carga gracias al balanceador y a los soportes diferentes en las diferentes regiones. Al crear estas, debemos asegurarnos de crearlas en diferentes regiones, esta es su única diferencia.

Contaremos también con una EFS, la cual nos dotará de almacenamiento elástico, y una RDS que servirá de base de datos para el almacenamiento de estos. La EFS también nos ayudará mas adelante para almacenar un fichero que debemos montar en la EC2, esta en su creación mantendremos los valores por defecto donde le aplicaremos el grupo de seguridad previamente creado (SGEFS).

Una vez creada, nos conectaremos a las instancias EC2 y trabajaremos paralelamente con estas dos en las dos zonas de disponibilidad hasta enlazarlas.

Seguidamente, en la terminal nos iremos a la ruta donde trabajaremos la mayor parte del tiempo, `$cd var/www/html/` y ahí creamos el fichero que nombraremos “efs-mount” y dentro de este el `index.html`, que será la de netflix.

Por temas de comodidad, modificaremos el root cambiando el documento raíz y añadiremos el efs-mount para la visualización directa de los ficheros de netflix.

Una vez terminado, nos encargaremos del balanceador, este se encargara de distribuir la carga por igual a las dos maquinas EC2 para no saturar el tráfico, esto nos permitirá acceder al contenido de las maquinas desde la propia ip del balanceador. Antes de todo recomiendo asignarle una ip elástica a cada máquina con la que vamos a trabajar. Y también debemos secularizar las dos maquinas de forma que solo se puedan acceder mediante el balanceador y que no tengan salida directa a Internet.

CONFIGURACIÓN DEL BALANCEADOR DE CARGA.

Para configurar el balanceador será necesario conectarnos a este, y en la terminal deberemos ejecutar varios comando para instalar los diferentes módulos;

```
sudo apt install apache2
```

```
sudo a2enmod proxy
```

```
sudo a2enmod proxy_http
```

```
sudo a2enmod proxy_ajp
```

```
sudo a2enmod rewrite
```

```
sudo a2enmod deflate
```

```
sudo a2enmod headers
```

```
sudo a2enmod proxy_balancer
```

```
sudo a2enmod proxy_connect
```

```
sudo a2enmod proxy_html
```

```
sudo a2enmod lbmethod_byrequests
```

```
sudo systemctl restart apache2
```

Y para el Proxy Inverso abriremos el siguiente archivo con el comando ;

```
sudo nano /etc/apache2/sites-enabled/000-default.conf
```

Este nos permitirá editarlo, le adjuntaremos este escrito.

```
<Proxy balancer://mycluster>
```

```
    # Server 1
```

```
    BalancerMember http://IP-HTTP-SERVER-1
```

```
    # Server 2
```

```
    BalancerMember http://IP-HTTP-SERVER-2
```

```
</Proxy>
```

```
ProxyPass / balancer://mycluster/
```

```
<Location /balancer-manager>
```

```
    SetHandler balancer-manager
```

```
    Order Deny,Allow
```

```
    Allow from all
```

```
</Location>
```

```
ProxyPassReverse / balancer://mycluster/
```

En lo subrayado de azul deberemos poner las ip privadas de las máquinas EC2 que deseemos conectar.

Este último nos permite acceder a la información del tráfico de las máquinas y poder hacer cambios respecto a estas.

RDS (BASE DE DATOS) y PÁGINA WEB

Para el almacenaje de los datos, debemos conectarnos a las máquinas EC2, en este caso solo a una, ya que comparten carpeta de montaje y los cambios afectan a las dos.

En la ruta indicada anteriormente crearemos los siguientes ficheros para darle forma a nuestra página web;

- Index.html
- Grabar.html
- Formulario.php
- Conexión.php
- Prueba.php

El Index es la página principal, la que lleva todo el contenido de netflix y que hemos ampliado con un botón de donaciones, este nos llevara a formulario.php donde introduciremos los datos de la donación que posteriormente serán guardados en la base de datos y que serán mostrados en la siguiente página que nos ofrece información sobre las cantidad total creada.

Para la creación del script de la base de datos utilizaremos una aplicación externa que previamente hemos conectado con la RDS y le introduciremos la tabla a recolectar dichos datos.

Y con esto ultimo finalizaremos el montaje del escenario.

4.- CARACTERÍSTICAS

Amazon EFS proporciona almacenamiento de archivos totalmente elástico y sin servidor que le permite compartir datos de archivos sin necesidad de aprovisionar o administrar la capacidad de almacenamiento y rendimiento.

- Alta disponibilidad y durabilidad
- Elasticidad y escalabilidad
- Clases de almacenamiento y administración del ciclo de vida

Multi A-Z En una implementación Multi-AZ de Amazon RDS, Amazon RDS crea de manera automática una instancia de base de datos (DB) principal y replica de manera síncrona los datos en una instancia de una AZ diferente. Cuando detecta un error, Amazon RDS conmuta por error automáticamente a una instancia en espera sin necesidad de intervención manual.

- Conmute por error de manera automática
- Proteja el rendimiento de la base de datos
- Mejora la durabilidad
- Aumenta la disponibilidad