Práctica 4.3

Despliegue de una arquitectura EFS-EC2-MultiAZ

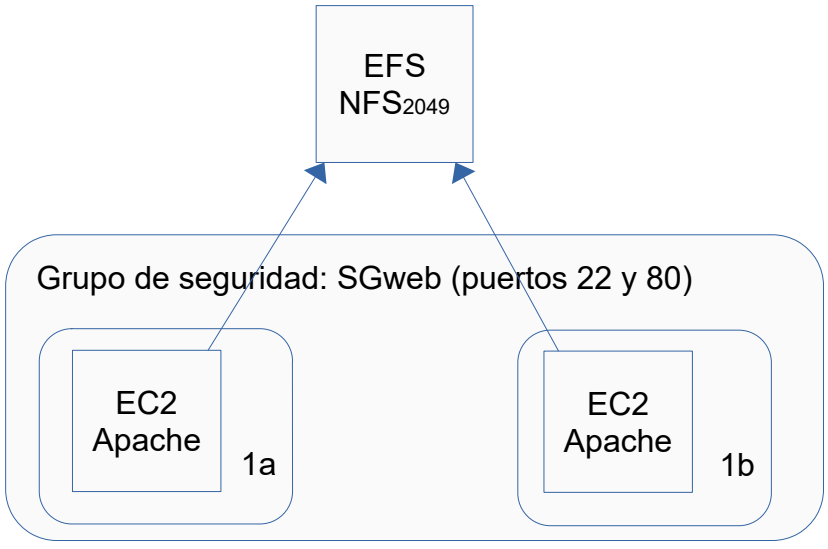
Guadalupe Luna Velázquez

Índice

1. Despliegue

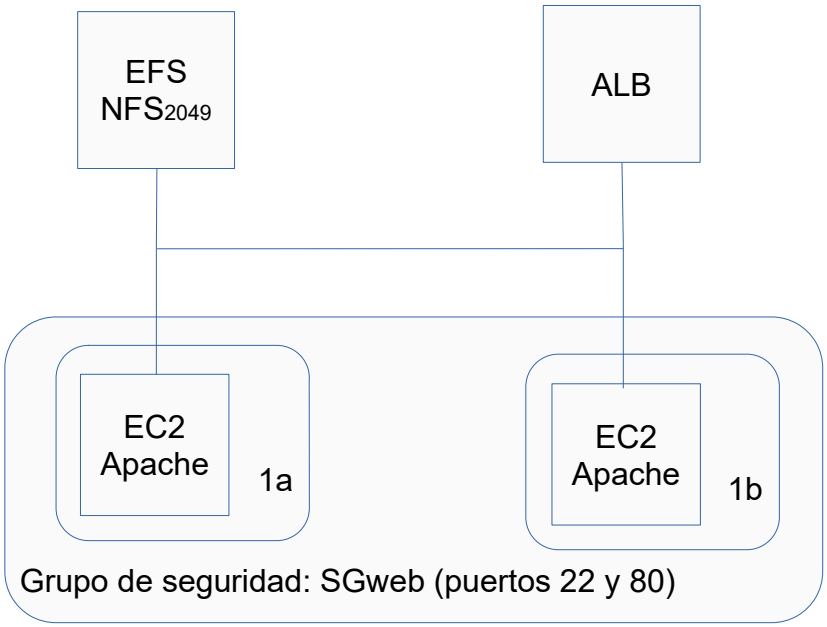
Crearemos en el servicio EFS un sistema de ficheros que será un sistema distribuido con un sistema NFS por el protocolo NFS y puerto 2049. Tendrá un grupo de seguridad con el puerto 2049 abierto y que solo puedan acceder las máquinas EC2 con apache para securizar todo.

Una máquina EC2, que tenga instalado Apache en la zona 1a.
Otra máquina EC2, que tenga instalado Apache en la zona 1b, que ambas configuradas para que sepan leer de la estructura de ficheros EFS, y con el puerto 22 y 80 abierto.



Con esto tendremos una página web estática de alta disponibilidad y Multi A-Z, por si se nos cae la zona de disponibilidad 1a trabajaría con la 1b y al contrario.

Además le añadiremos un balanceador de carga que será creado con una EC2 y se conectará a los servidores web para que distribuya automáticamente el tráfico.



Con esto ya estaría listo nuestro despliegue con su balanceador de carga que distribuya el tráfico de la red a los dos servidores web que hemos levantado, que están en diferentes zonas de disponibilidad, en la a y b, con apache instalado y operando, en estos servidores guardamos nuestra página web.
También tenemos un sistema de ficheros para que administre nuestros archivos e irá creciendo o disminuyendo en función de la cantidad de archivos que tengamos.

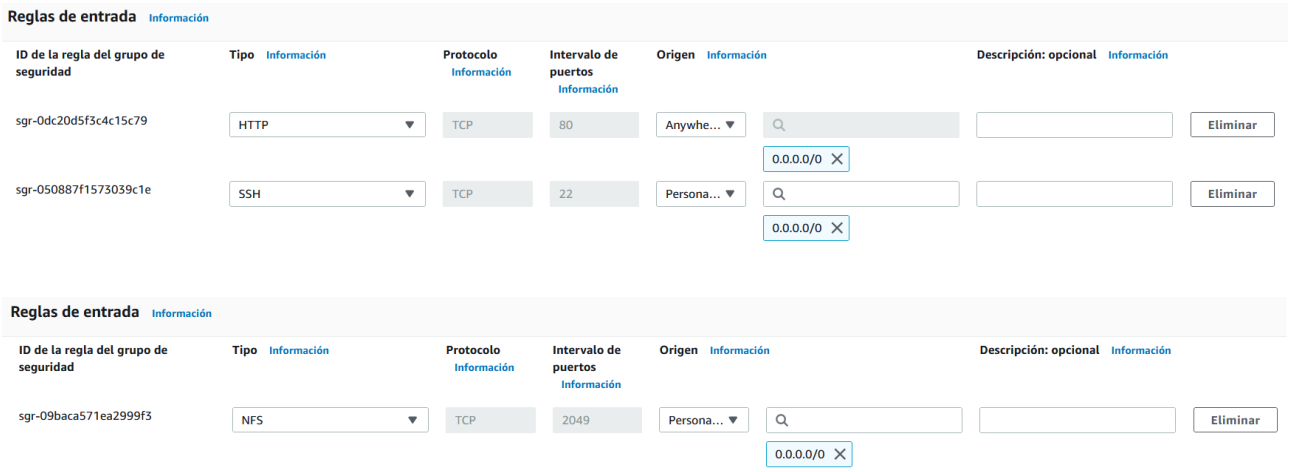
Tanto el balanceador de carga como el sistema de ficheros estarán securizados para que solo se pueda acceder desde la EC2.
Las EC2 también tendrán seguridad para que no se pueda modificar desde cualquier lado.

Guadalupe Luna Velázquez
Administración de Sistemas Informáticos en Red
Gestión de Bases de Datos

2. Desarrollo del despliegue

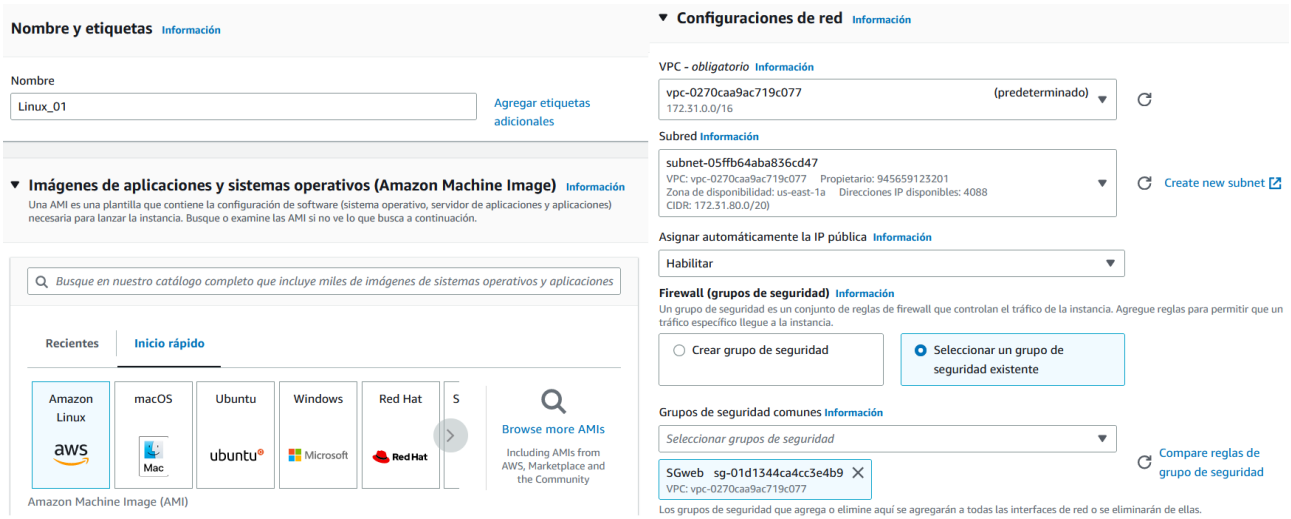
2.1 Creación de los grupos de seguridad

Para esta práctica primero en el servicio EC2, iremos a Grupos de Seguridad y creamos 2 grupos de seguridad, en uno lo llamaremos SGweb y abriremos el purto 80, HTTP desde cualquier IPv4 y el puerto 22 de SSH por si hay que modificarlo, el otro se llamará SGEfs con el puerto 2049 de NFS para cualquier IPv4. Quedando así:



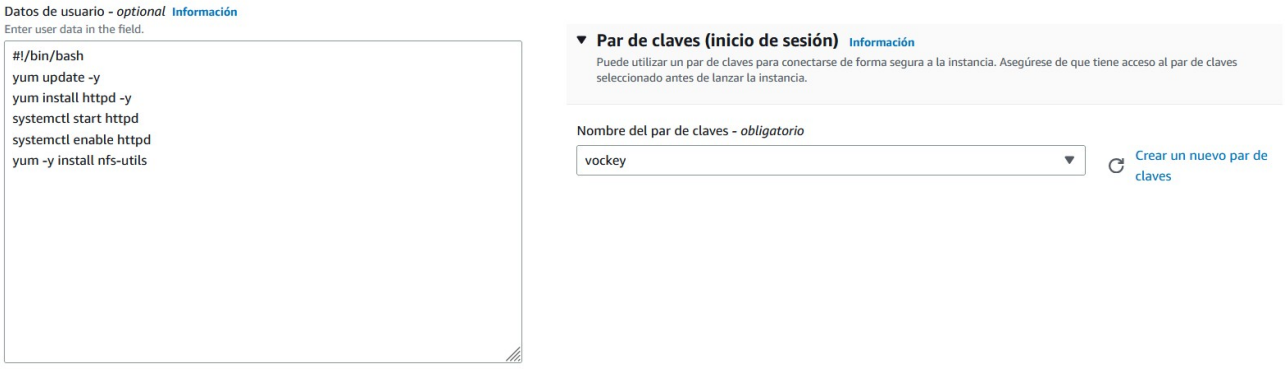
2.2 Creación de instancias nodos

Seguimos en el servicio EC2 y ahora creamos una EC2 que se llamará Linux_01, con Amazon Linux, par de claves vockey, la VPC predeterminada pero elegimos la subred a, y permitimos que asigne una ip pública, se le asigna el grupo de seguridad que antes hemos creado con el nombre Sgweb.

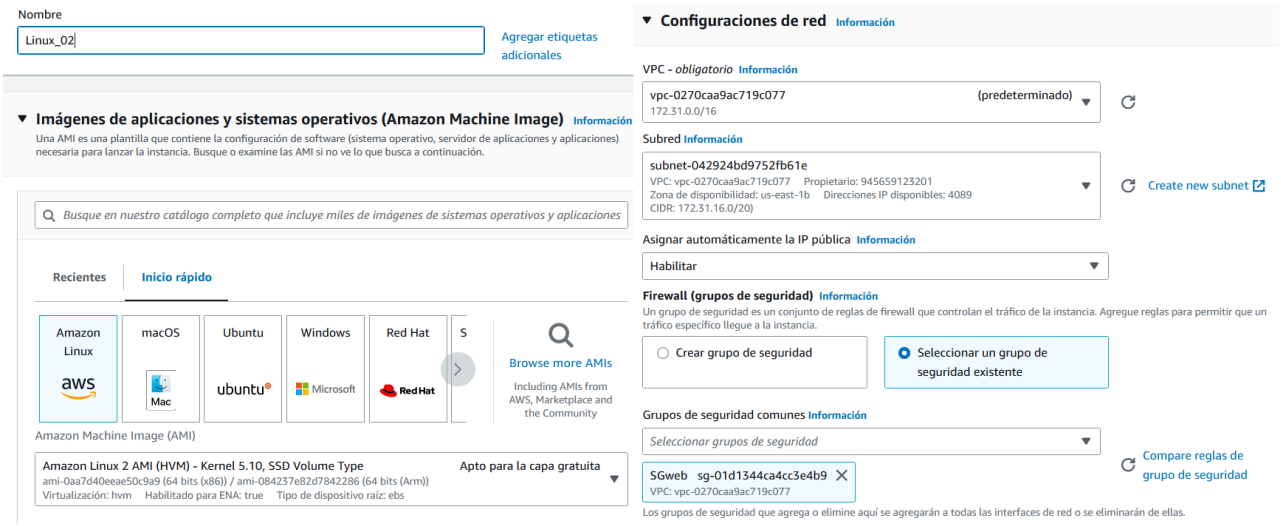


En configuración avanzada introducimos los datos de usuario que se muestran a continuación:

```
#!/bin/bash
yum update -y
yum install httpd -y
systemctl start httpd
systemctl enable httpd
yum -y install nfs-utils
```



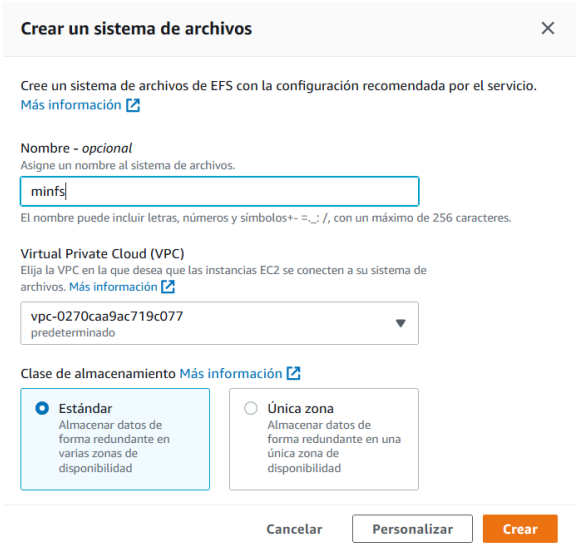
Y lanzamos la instancia, mientras crearemos otra instancia llamada Linux_02, con la misma configuración Amazon Linux,par de claves vockey, VPC predeterminada pero con subred b y el mismo grupo de seguridad llamado SGweb y volvemos a configuración avanzada para pegar los datos de usuario, por último lanzamos la instancia.



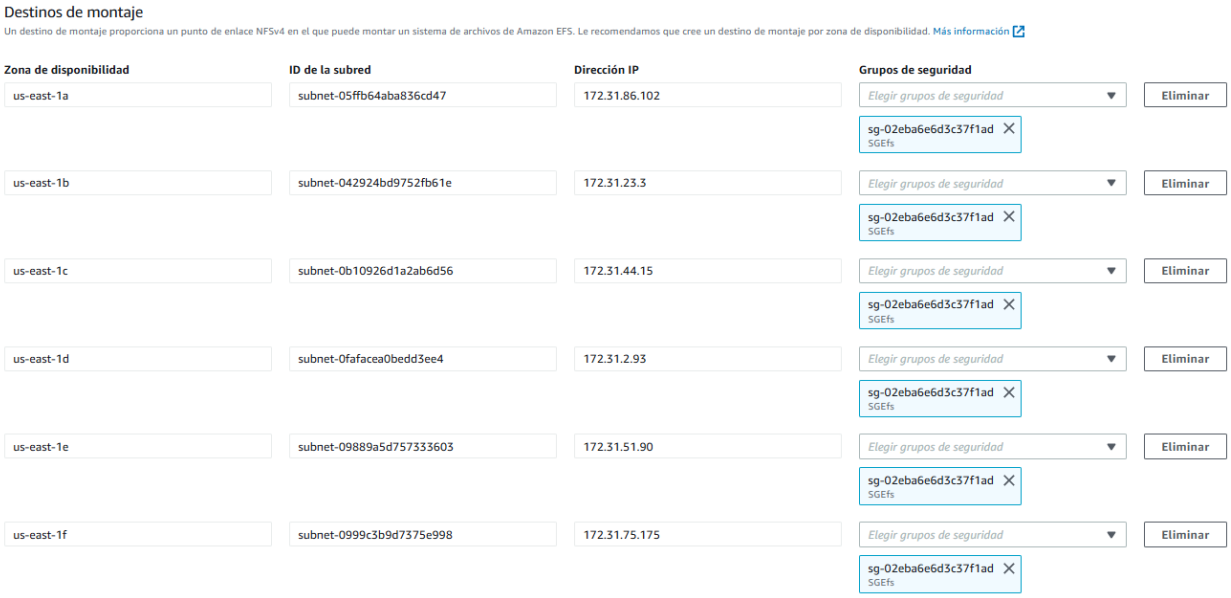
Además creamos un par de ips elásticas para que al cerrar el laboratorio, esta no cambie, las creamos y las asociamos una por máquina.

2.3 Creación del Sistema de archivos

En el servicio EFS, vamos a crear un sistema de ficheros, llamado minfs y tendrá el VPC por defecto y elegimos la opción Estándar para que esté disponible en todas las zonas de disponibilidad.



Entramos en nuestro sistema de ficheros llamado miefs y accederemos a los grupos de seguridad y en todas las subredes le asignaremos el grupo de seguridad SGEfs.



2.4 Configuración de los servidores web

Cuando se han terminado de crear, conectaremos a ellas donde podemos verificar que se haya instalado correctamente Apache y después de verificar entraremos en /var/www/html y crearemos una carpeta llamada efs-mount con el comando “mkdir efs-mount”.

```
[ec2-user@ip-172-31-94-123 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
  Drop-In: /usr/lib/systemd/system/httpd.service.d
           └─php-fpm.conf
   Active: active (running) since Fri 2023-02-10 15:02:25 UTC; 29min ago
     Docs: man:httpd.service(8)
 Main PID: 3149 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
   CGroup: /system.slice/httpd.service
           ├─3149 /usr/sbin/httpd -DFOREGROUND
           ├─3161 /usr/sbin/httpd -DFOREGROUND
           ├─3162 /usr/sbin/httpd -DFOREGROUND
           ├─3163 /usr/sbin/httpd -DFOREGROUND
           ├─3164 /usr/sbin/httpd -DFOREGROUND
           └─3165 /usr/sbin/httpd -DFOREGROUND

Feb 10 15:02:25 ip-172-31-94-123.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Feb 10 15:02:25 ip-172-31-94-123.ec2.internal systemd[1]: Started The Apache HTTP Server.
[ec2-user@ip-172-31-94-123 ~]$ cd /var/www/html
[ec2-user@ip-172-31-94-123 html]$ mkdir efs-mount
```

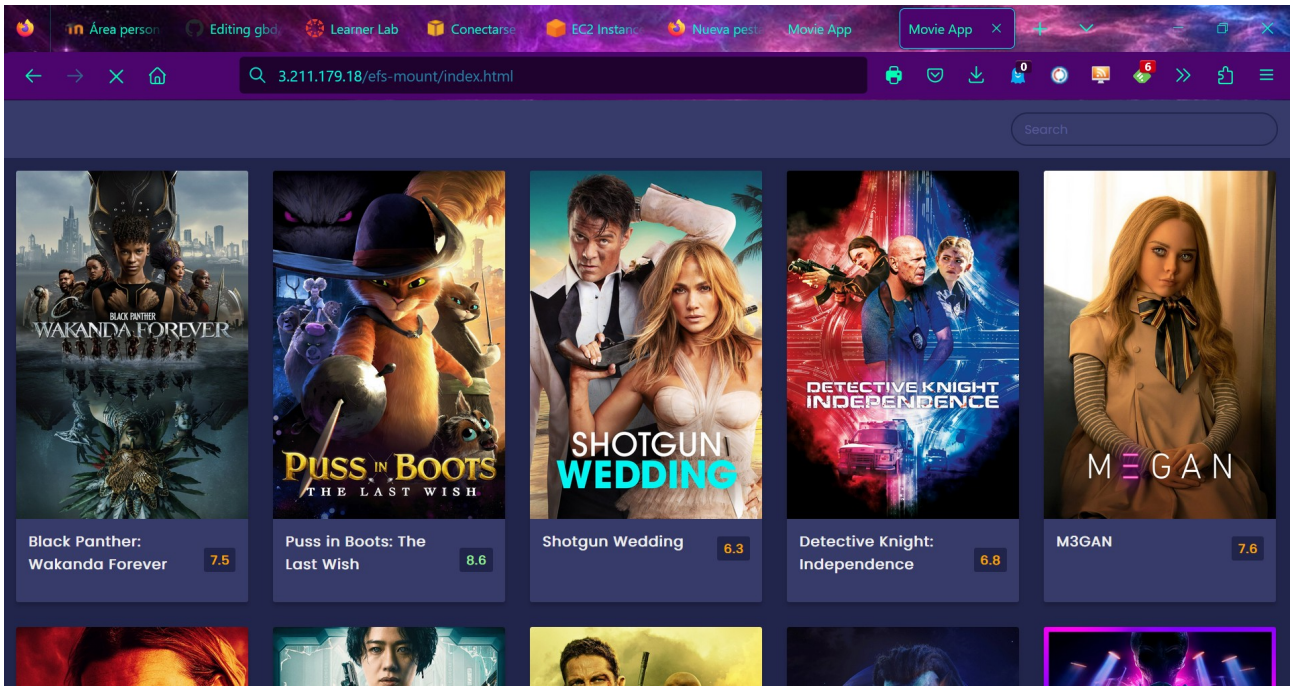
Ahora vamos a utilizar el comando para montar en un sistema nfs sobre la carpeta que hemos creado, y ejecutamos el comando cambiando el id por el nuestro.
“sudo mount -t nfs -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-0b3af357a01238bb9.efs.us-east-1.amazonaws.com:/ efs-mount”.

Ahora con otro comando nos descargaremos la página web:
“wget https://s3.eu-west-1.amazonaws.com/www.profesantos.cloud/Netflix.zip”, lo descomprimimos con el comando “unzip Netflix.zip” y ya estaría la página web.

```
[ec2-user@ip-172-31-94-123 efs-mount]$ wget https://s3.eu-west-1.amazonaws.com/www.profesantos.cloud/Netflix.zip
--2023-02-10 15:56:36--  https://s3.eu-west-1.amazonaws.com/www.profesantos.cloud/Netflix.zip
Resolving s3.eu-west-1.amazonaws.com (s3.eu-west-1.amazonaws.com)... 52.218.42.11, 52.218.62.35, 52.218.89.139, ...
Connecting to s3.eu-west-1.amazonaws.com (s3.eu-west-1.amazonaws.com)|52.218.42.11|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1993 (1.9K) [application/zip]
Netflix.zip: Permission denied

Cannot write to ‘Netflix.zip’ (Success).
[ec2-user@ip-172-31-94-123 efs-mount]$ sudo wget https://s3.eu-west-1.amazonaws.com/www.profesantos.cloud/Netflix.zip
--2023-02-10 15:56:44--  https://s3.eu-west-1.amazonaws.com/www.profesantos.cloud/Netflix.zip
Resolving s3.eu-west-1.amazonaws.com (s3.eu-west-1.amazonaws.com)... 52.218.45.32, 52.218.57.3, 52.218.96.18, ...
Connecting to s3.eu-west-1.amazonaws.com (s3.eu-west-1.amazonaws.com)|52.218.45.32|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1993 (1.9K) [application/zip]
Saving to: ‘Netflix.zip’

100%[==========================================================================================================================================>] 1,993       --.-K/s   in 0s

2023-02-10 15:56:45 (27.5 MB/s) - ‘Netflix.zip’ saved [1993/1993]

[ec2-user@ip-172-31-94-123 efs-mount]$ sudo unzip Netflix.zip
Archive:  Netflix.zip
  inflating: index.html
  inflating: script.js
  inflating: style.css
[ec2-user@ip-172-31-94-123 efs-mount]$
```

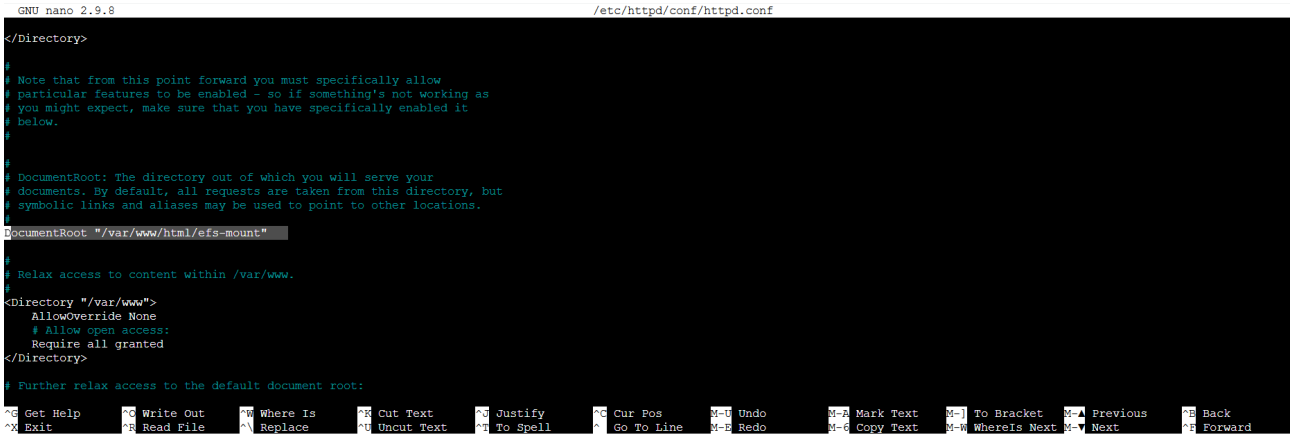
Guadalupe Luna Velázquez
Administración de Sistemas Informáticos en Red
Gestión de Bases de Datos

Estos pasos también la hacemos en la segunda máquina EC2 para así que se vea la misma página web en los dos servidores. Si buscamos nuestra ip en internet y en la ruta accedemos al html de la página web se visualizará.

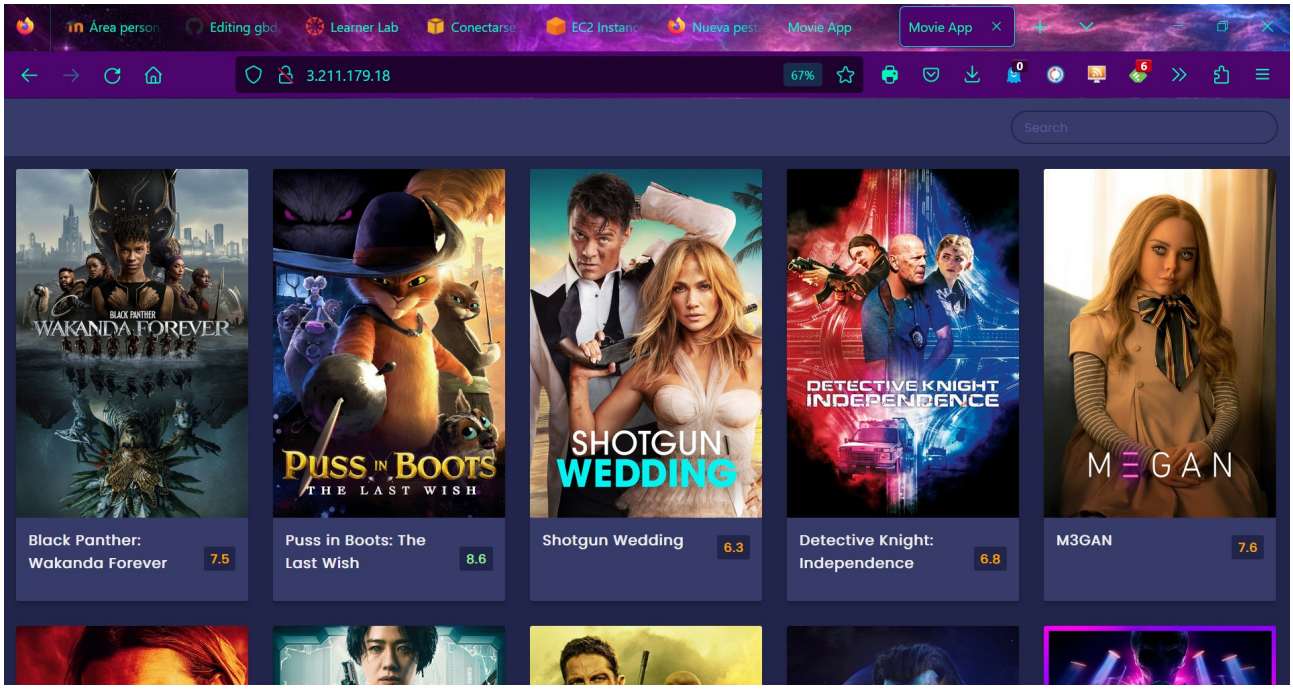


Ahora vamos a modificar el archivo de Apache para simplemente acceder a la página con nuestra ip.

Así que modificaremos el archivo /etc/httpd/conf/httpd.conf con “sudo nano” o “vim” y modificaremos el DocumentRoot a DocumentRoot "/var/www/html/efs-mount", y después reiniciaremos el servicio httpd con “systemctl restart httpd”, esto se hará en los dos servidores.

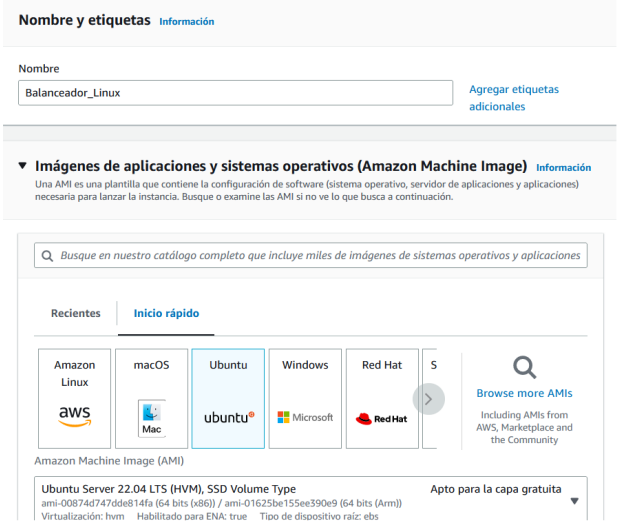


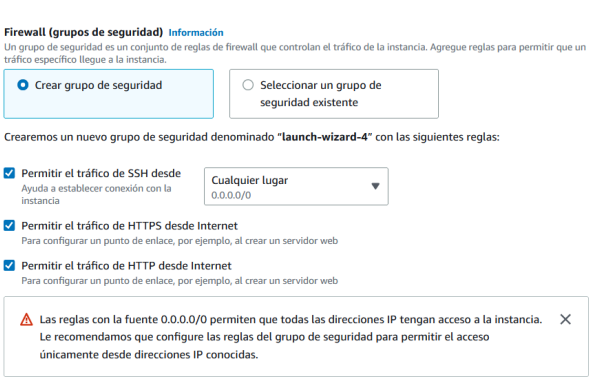
Con esto ya estaría los servidores web configurados y si ponemos simplemente la ip de los servidores nos mostrará la página web.



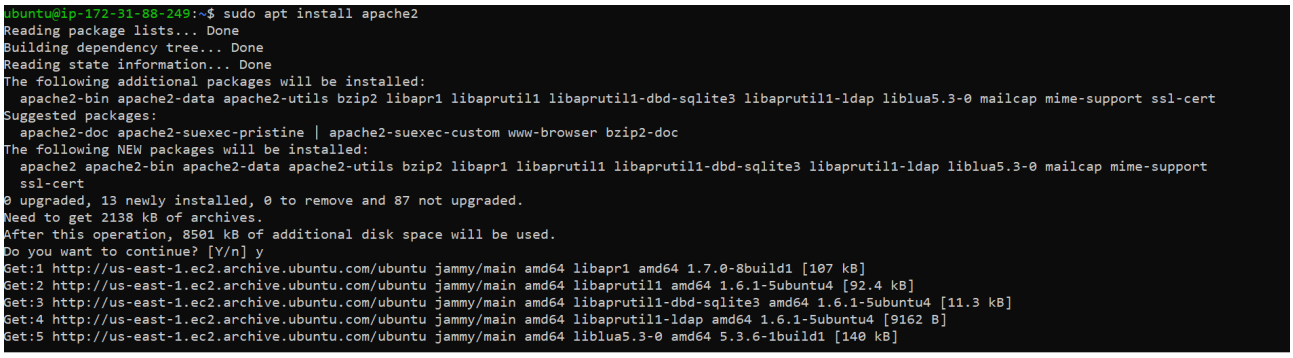
2.5 Creación y configuración del balanceador

Creamos otra EC2 más para hacer nuestro balanceador, se llamará Balanceador_Linux, con sistema Ubuntu, con par de claves vockey y le creamos un nuevo grupo de seguridad, donde abriremos los puertos SSH,HTTP y HTTPS.
También le asignaremos una ip elástica como a los nodos y se la asociaremos.





Cuando se cree el balanceador, nos conectaremos a él e instalaremos Apache con "sudo apt install apache2", también debemos reiniciar el servicio con el comando "sudo systemctl restart apache2".

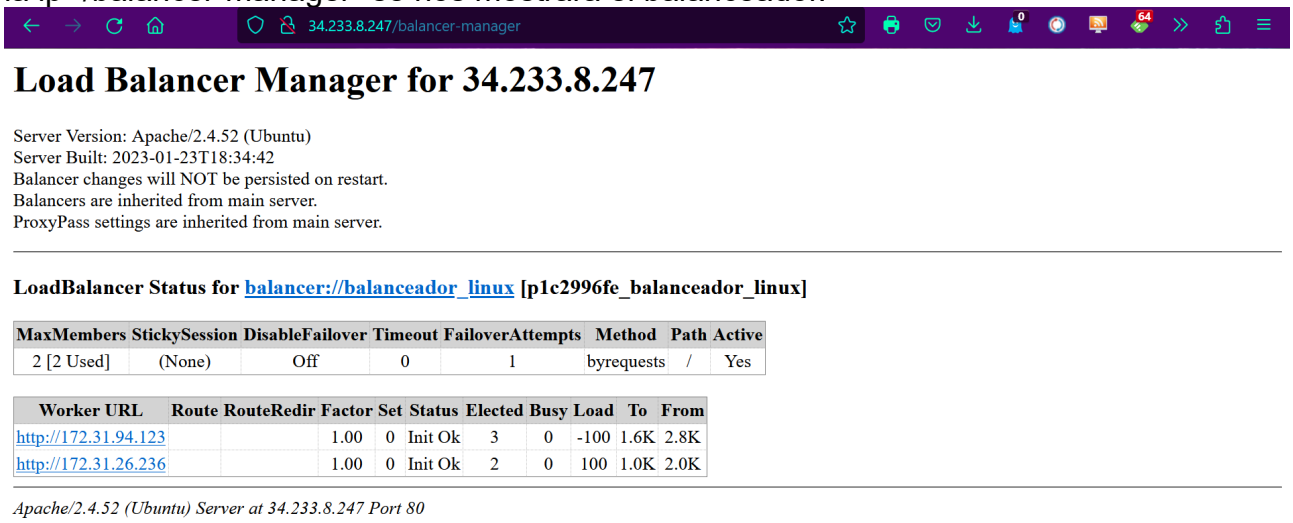


A continuación, editamos el fichero /etc/apache2/sites-enabled/000-default.conf para configurar nuestro gestor de balanceo y pondremos lo siguiente:

```
ProxyPass /balancer-manager !

<Proxy balancer://Balanceador_Linux>
  # Server 1
  BalancerMember http://172.31.94.123

  # Server 2
  BalancerMember http://172.31.26.236
</Proxy>
  ProxyPass / balancer:/Balanceador_Linux/
  ProxyPassReverse / balancer:/Balanceador_Linux/

<Location /balancer-manager>
  SetHandler balancer-manager
  Order Deny,Allow
  Allow from all
</Location>
```

```
 GNU nano 6.2                                    /etc/apache2/sites-enabled/000-default.conf
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf


ProxyPass /balancer-manager !

<Proxy balancer://Balanceador_Linux>
        # Server 1
        BalancerMember http://172.31.94.123

        # Server 2
        BalancerMember http://172.31.26.236
</Proxy>

        ProxyPass / balancer://Balanceador_Linux/
        ProxyPassReverse / balancer://Balanceador_Linux/

<Location /balancer-manager>
        SetHandler balancer-manager
        Order Deny,Allow
        Allow from all
</Location>
```

Tras esto reiniciaremos Apache de nuevo con "sudo systemctl restart apache2" y al buscar la ip de nuestro balanceador saldrá la página web, si ponemos a continuacion de la ip "/balancer-manager" se nos mostrará el balanceador.

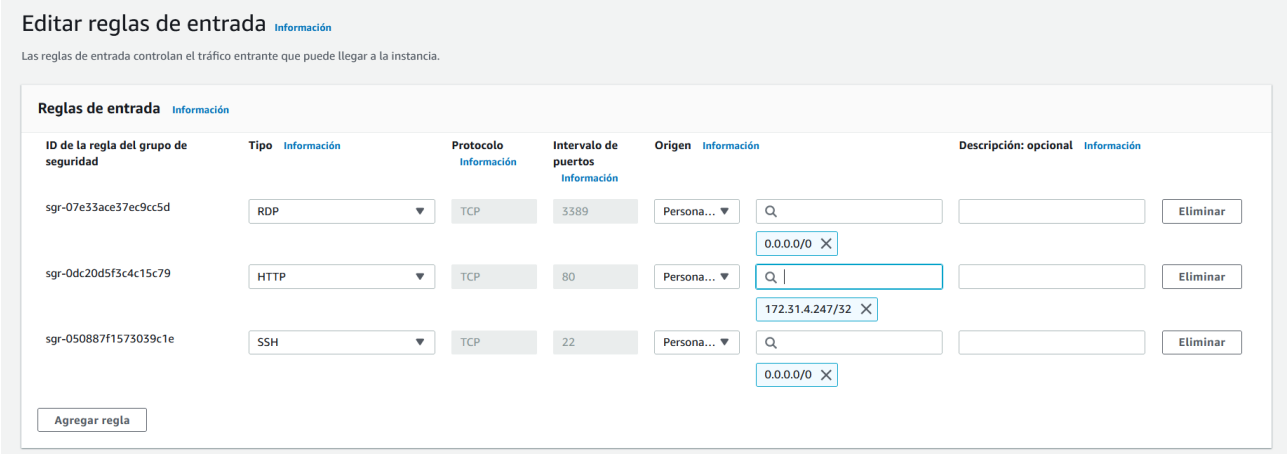


Con esto el balanceador funciona y si en algún momento se cae un nodo, el otro seguiría mostrando la página web.

2.7 Securización de los puertos

Por último, debemos proteger y securizar el puerto 80 de los nodos que contienen la página Web para que solo se puedan acceder por el balanceador y no estén expuestos al público.

Iremos al grupo de seguridad de estos que es el SGWeb y en la regla HTTP, en vez de Anywhere IPv4, pondremos la ip privada del balanceador.



Aun modificando esto nuestra página web se verá igual de bien y no será tan fácilmente accesible.

3. Servicios y sus Ventajas

El servicio que se utilizará será EC2 principalmente para crear las instancias que tienen instalado Apache que servirán como servidores web, las ventajas de usar este servicio es que es fácil de manejar y presenta muchas opciones muy útiles para crear infraestructuras, al añadir una ip elástica, crear y modificar al gusto los grupos de seguridad, se pueden crear varias instancias con especificar la configuración en una sola vez, también su fácil conexión a la instancia a través de SSH.

Además este servicio ofrece una gran escalabilidad lo que nos permitiría añadir servidores web o lo que fuera necesario en caso de necesitar ampliar el despliegue o añadir más funciones, también ha hecho posible darle a nuestro despliegue la gran disponibilidad que ofrece.

Este servicio también se ha usado para crear el balanceador de carga con el que conseguimos distribuir el tráfico de red

También utilizaremos el servicio EFS para crear un sistema de ficheros que montaremos en nuestro servidor web, este servicio nos ha permitido crearlo fácil y rápidamente sin tener muchos conocimientos sobre ellos y configurarlo a nuestra manera decidiendo las zonas de disponibilidad y administrándolas conjuntamente con el servicio EC2.
Este servicio es una manera segura y fiable de acceder a nuestros archivos creando un sistema de archivos completamente administrado y diseñado que también contará con gran durabilidad y disponibilidad.