

DEPLOYMENT OF AN ARCHITECTURE EFS-EC2-MULTIAZ IN THE CLOUD (AWS)

EFS (web Server Multi A-Z)

Mónica Blanco Martínez

1º Daw

Arquitectura de componentes.

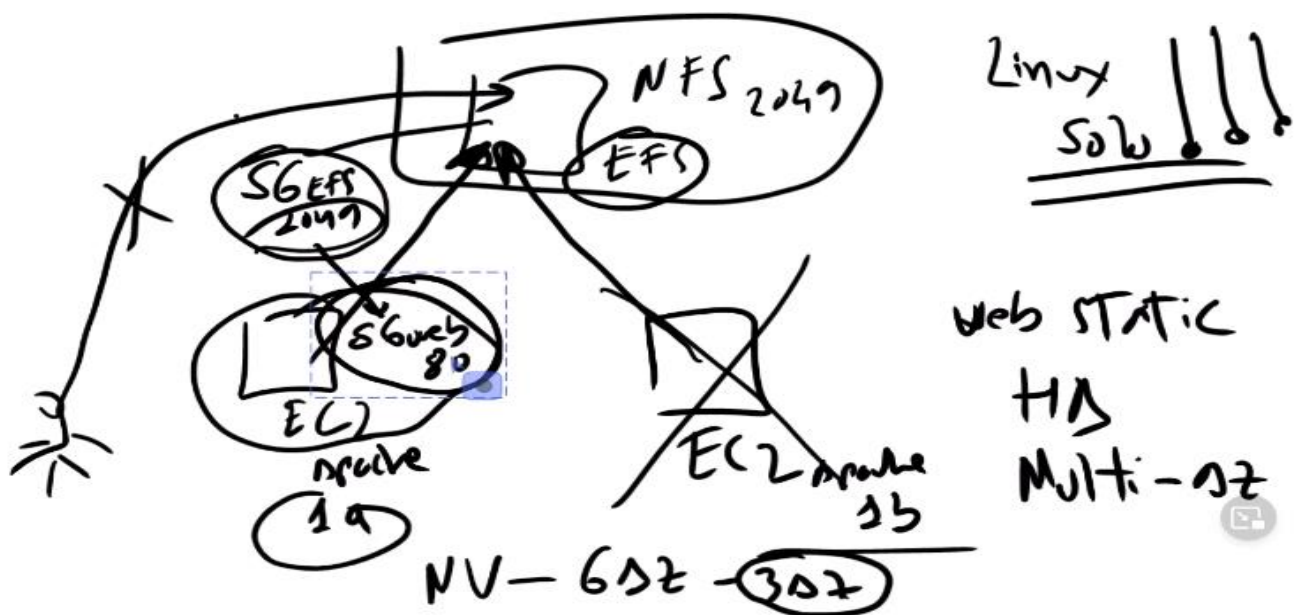
Toda esta arquitectura vamos a montarla desde el servicio AWS en el que estamos trabajando este curso.

Hay que recordar que esto sólo funciona en Linux.

Crearemos dos máquinas EC2, en la que instalaremos el servicio Apache en la zona de disponibilidad 1ª y otra EC2 con Apache en la zona 1B. Tendrán que saber leer la EC2-1 y la EC2 leer desde una estructura de ficheros EFS. Consiguiendo tener una página web estática en alta disponibilidad y en multi-A-Z.

Así si se nos cae la zona de disponibilidad 1A tenemos la 1B y al revés. Y el sistema de ficheros EFS es multi A-Z. (Lo podemos hacer como máximo es 6 máquinas en la zona de disponibilidad de Virginia, mientras que si la montamos en España serían tres).

Para ello crearemos un fichero compartido en EFS para que una vez quede instalado en una de las máquinas y un balanceador para que pueda ser visto (al utilizar como ejemplo un fichero de Netflix) desde la IP del balanceador.



INSTRUCCIONES

1. Creamos los grupos de seguridad

Que serán tanto para EFS como para las dos máquinas EC2.

1.1 Grupo de seguridad EC2-Web

Abrimos el puerto 80 para abrir el puerto en cascada, y el puerto 22 SSH

1.2 Grupo de seguridad EFS-NFS

Le abrimos el puerto 2049 de Seguridad para el SGEFS , para todo el mundo o bien solo para las máquinas que vienen desde la WEB.

The screenshot shows the AWS Management Console interface. On the left is a navigation menu with categories like 'Panel de EC2', 'Instancias', 'Imágenes', 'Elastic Block Store', and 'Red y seguridad'. The main content area is titled 'EC2 > Grupos de seguridad > sg-06547e5f47df07e8e - SGefs'. It displays the 'Detalles' (Details) tab for this security group. The details include the name 'SGefs', the ID 'sg-06547e5f47df07e8e', the description 'SGefs', and the VPC ID 'vpc-08c7a041118560e29'. It also shows the owner '860937737830', 2 inbound rules, and 1 outbound rule. Below the details, there are tabs for 'Reglas de entrada', 'Reglas de salida', and 'Etiquetas'. The 'Reglas de entrada' tab is active, showing a table with 2 inbound rules. A banner at the top of the rules section says 'Ahora puede comprobar la conectividad de red con Reachability Analyzer' with a button 'Ejecutar Reachability Analyzer'. The table has columns for Name, ID, Version, Type, Protocol, Port Range, Origin, and Description.

	Nombre	ID de la regla del g...	Versión de IP	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción
<input type="checkbox"/>	-	sg-0c2b73c9d09c70e8	-	NFS	TCP	2049	sg-0d75b6db8e2c614...	-
<input type="checkbox"/>	-	sg-006d48d322c7ed...	-	HTTP	TCP	80	sg-0d75b6db8e2c614...	-

2.Creación Security group

1. Creamos uno llamándolo SGweb y abrimos el protocolo 80(HTTP)a todo el mundo; instalar también para poder acceder a las máquinas y abrimos el servicio SSH puerto 20.
2. Creamos el segundo Security group SGEfs el protocolo será el NFS que va por el 2049 para todo el mundo. (si sólo queremos que accedan los del grupo de SGweb lo seleccionaríamos)

3. Lanzar Instancia

Lanzamos la primera instancia "Linux_01": con t2micro, vokey, la Subnet le decimos que la coloque en la 1a, el security group es el SGweb;

The screenshot shows the AWS 'Launch Instance' wizard. The instance name is 'Linux_02'. The AMI is 'Amazon Linux 2 Kernel 5.10'. The instance type is 't2.micro'. The security group is 'SGweb'. The storage is '1 volumen(es): 8 GiB'. The summary panel on the right confirms these settings. A notification about the free tier is also present.

Y en el addition habilitamos el user data para introducir:

```
#!/bin/bash
```

```
yum update -y
```

```
yum install httpd -y
```

```
systemctl start httpd
```

```
systemctl enable httpd
```

```
yum -y install nfs-utils
```

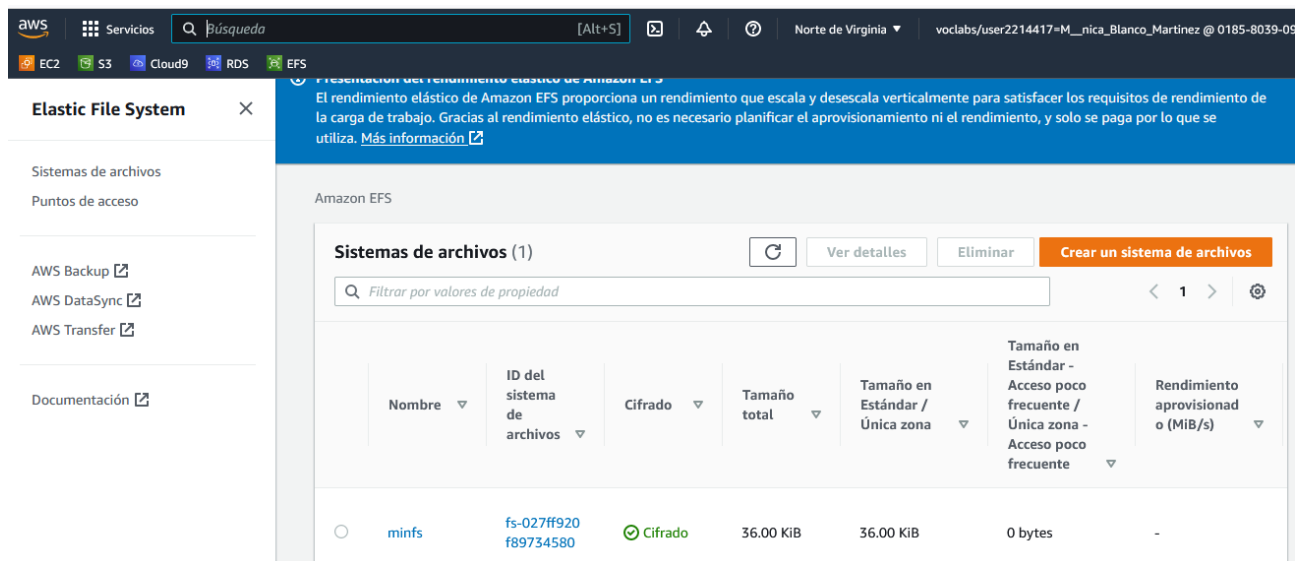
A continuación lanzamos la instancia y hacemos exactamente igual para la instancia Linux_02, a excepción de que Subnet le decimos que la coloque en la 1b.

4. Montamos el fichero EFS

Le damos a crear el fichero, lo instalamos “minfs”, las máquinas virtuales la dejamos en la que aparezca por defecto y en todas las zonas de disponibilidad. Y la creamos.

Automáticamente se crea y cogemos el ID que se ha generado, para podernos conectar.

Vamos a la parte de red y cambiaremos en los puntos de montaje para que los grupos de seguridad tengan todos SGEfs, lo cambiaremos uno a uno todos.



5. Conexión a ambas máquinas

Nos conectamos a la primera máquina y a la segunda. Cuando tengamos el prom de ellas ponemos para saber si se han instalado bien `sudo systemctl status httpd`.

Lo hacemos en las dos.

Para montar la base de ficheros creamos una carpeta:

- `ls /var/www/html`
- `cd /var/www/html`
- `mkdir efs-mount`
- `sudo mount -t nfs -o nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport.efs.us-east-1.amazonaws.com:/ /var/www/html/efs-mount`

(Será el ID de EFS que copiamos al principio, por lo que estará personalizado para cada uno, en nuestro caso ese es el válido).

Lo ejecutamos y vemos con `df -h` para saber a que carpeta corresponde.

Si entro en `cd efs-mount`

ls verá que tengo vacía la carpeta.

Subimos la página web wget <https://s3.eu-west-1.amazonaws.com/www.profesantos.cloud/Netflix.zip>

Actualizamos con ls y vemos que lo tenemos descomprimido

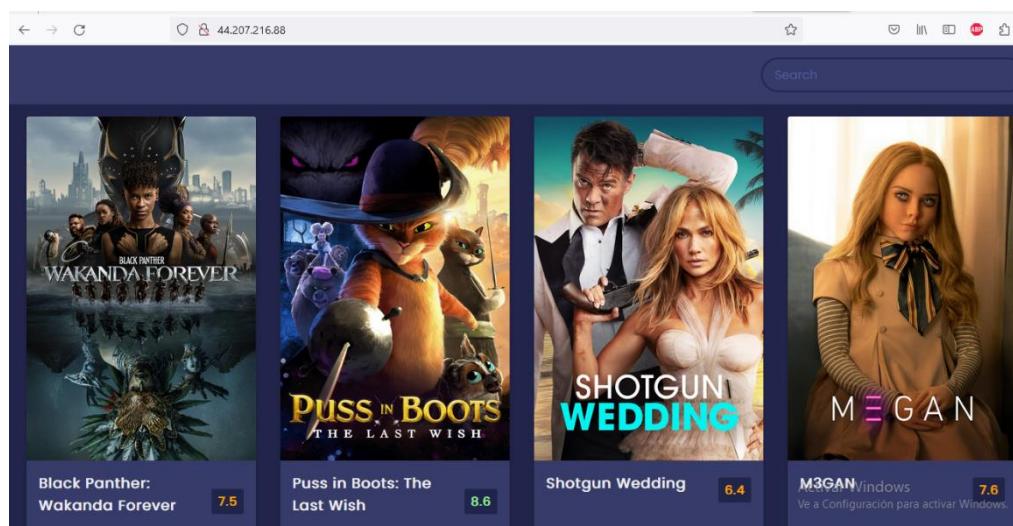
```
aws
Servicios
Búsqueda [Alt+S]
Norte de Virginia voclabs/user2214417-M_nica_Blanco_Martinez @ 0185
EC2 S3 Cloud9 RDS
tmpfs 483M 0 483M 0% /sys/fs/cgroup
/dev/xvda1 8.0G 1.7G 6.4G 21% /
tmpfs 97M 0 97M 0% /run/user/1000
[ec2-user@ip-172-31-90-5 html]$ sudo mount -t nfs -o nfsvers=4.1,rsize=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport fs-027ff9204
efs.us-east-1.amazonaws.com:/ /var/www/html/efs-mount
[ec2-user@ip-172-31-90-5 html]$ df -h
-bash: df: command not found
[ec2-user@ip-172-31-90-5 html]$ mkdir efs-mount
mkdir: cannot create directory 'efs-mount': File exists
[ec2-user@ip-172-31-90-5 html]$ cd efs-mount
[ec2-user@ip-172-31-90-5 efs-mount]$ ls
index.html Netflix.zip Netflix.zip.1 script.js style.css
[ec2-user@ip-172-31-90-5 efs-mount]$ wget https://s3.eu-west-1.amazonaws.com/www.profesantos.cloud/Netflix.zip
--2023-02-08 19:29:33-- 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)... 52.218.92.227, 52.218.96.146, 52.218.97.107, ...
Connecting to s3.eu-west-1.amazonaws.com (s3.eu-west-1.amazonaws.com) [52.218.92.227]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1993 (1.9K) [application/zip]
Netflix.zip.2: Permission denied

Cannot write to 'Netflix.zip.2' (Success).
[ec2-user@ip-172-31-90-5 efs-mount]$ ls
index.html Netflix.zip Netflix.zip.1 script.js style.css
[ec2-user@ip-172-31-90-5 efs-mount]$
```

Hacemos exactamente igual en la otra máquina. Así tenemos los mismos ficheros en ambas máquinas aunque solo lo hayamos instalado en una. Si copiamos la IP pública y lo pegamos en el navegador veremos la web de Netflix.

Si modificamos el fichero el documentRoot y cambiamos la ruta por /etc/httpd/conf/httpd.conf no hará que añadamos a la IP /efs-mount/index.html

Como vemos abajo ésta es la IP de Linux_01

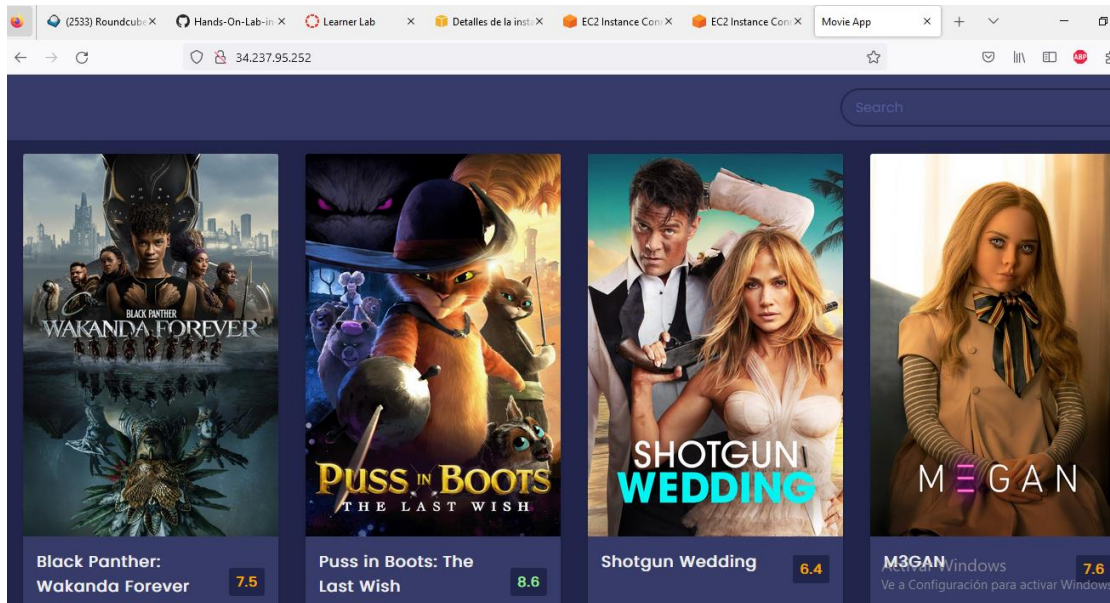


Mónica Blanco Martínez

1ºDAW

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

La ip de Linux_02



6. Instalamos el Balanceador

Nos conectamos a la instancia del balanceador e instalamos Apache:

```
aws
Servicios
Búsqueda [Alt+S]
Norte de Virginia
voclabs/user2214417-M_nica_Blanco_Martinez @ 0185-8039-091

7 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Last login: Tue Feb  7 17:30:39 2023 from 18.206.107.28
ubuntu@ip-172-31-0-121:~$ sudo systemctl restart
Too few arguments.
ubuntu@ip-172-31-0-121:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [107 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [874 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [192 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [587 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [90.9 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [801 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [143 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [8600 B]
Fetched 3033 kB in 1s (3548 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
13 packages can be upgraded. Run 'apt list --upgradable' to see them.

i-0102eba80016a53fe (Balanceador)
```

```
sudo apt update
```

```
sudo apt install apache2
```

Instalamos los módulos necesarios para *proxyficar* una conexión y gestionar balanceo de *requests*:

```
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
```

```
Reiniciamos apache2
sudo systemctl restart apache2
```

Editamos el fichero /etc/apache2/sites-enabled/000-default.conf para configurar nuestro gestor de balanceo:

```
ProxyPass /balancer-manager !
<Proxy balancer://mycluster>
    BalancerMember http://172.31.31.218
    BalancerMember http://172.31.33.21
</Proxy>
```

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

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

Y reiniciamos Apache2 nuevamente.

Si ahora visitamos la IP del balanceador en lugar de cualquiera de los nodos, vamos a ver la página Web que está en ambos nodos:

Tras lo cual, la página Web sigue funcionando cuando es accedida a través del balanceador:

←

→

↺

🔒

54.198.87.236/balancer-manager

☆

📧

📄

🔍

📑

🔖

Load Balancer Manager for 54.198.87.236

Server Version: Apache/2.4.52 (Ubuntu)
 Server Built: 2023-01-23T18:34:42
 Balancer changes will NOT be persisted on restart.
 Balancers are inherited from main server.
 ProxyPass settings are inherited from main server.

LoadBalancer Status for [balancer://mycluster](#) [pf4b4eaa9_mycluster]

MaxMembers	StickySession	DisableFailover	Timeout	FailoverAttempts	Method	Path	Active
2 [2 Used]	(None)	Off	0	1	byrequests	/	Yes

Worker URL	Route	RouteRedir	Factor	Set	Status	Elected	Busy	Load	To	From
http://172.31.4.177			1.00	0	Init Err	8	0	0	0	0
http://172.31.90.5			1.00	0	Init Err	8	0	0	0	0

Apache/2.4.52 (Ubuntu) Server at 54.198.87.236 Port 80