

IMPLANTACIÓN DE APLICACIONES WEB

IMPLANTACIÓN DE APLICACIONES WEB

4.2 Wordpress en tres niveles

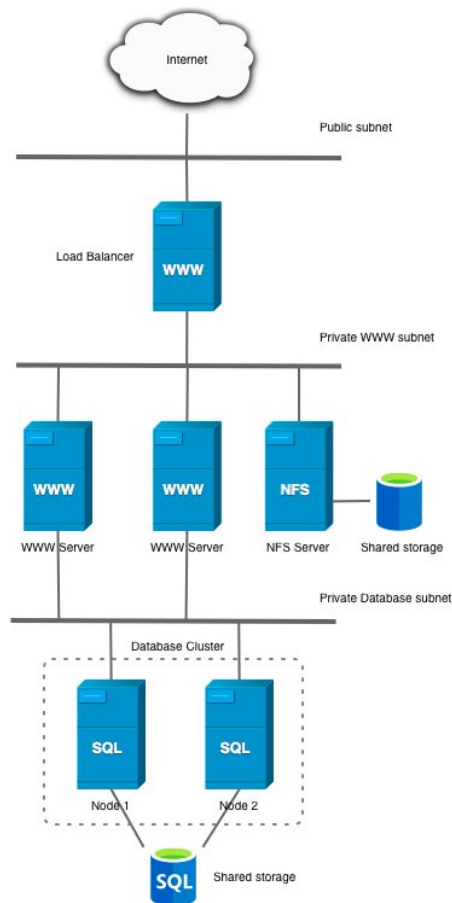
✎ Otero Gómez, Manuela
2º ASIR
IAW
25.11.2022

Tabla de contenidos

1. Objetivo.....	4
2. Modificaciones al esquema anterior.....	4
3. Escenario antes del balanceador.....	6
4. Balanceador de carga.....	6
5. NFS.....	8
5.1 Implementación	8

1. Objetivo

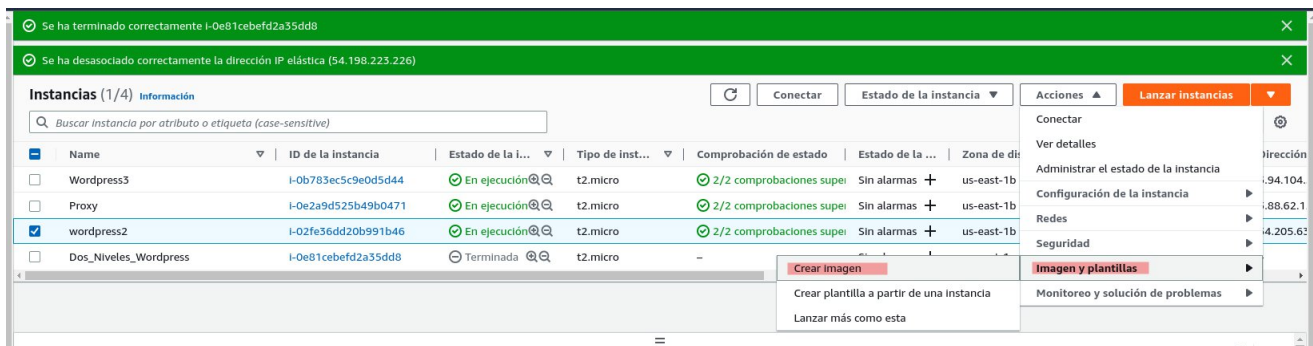
Vamos a añadir un proxy que además balancee la carga quedando un esquema similar a este:



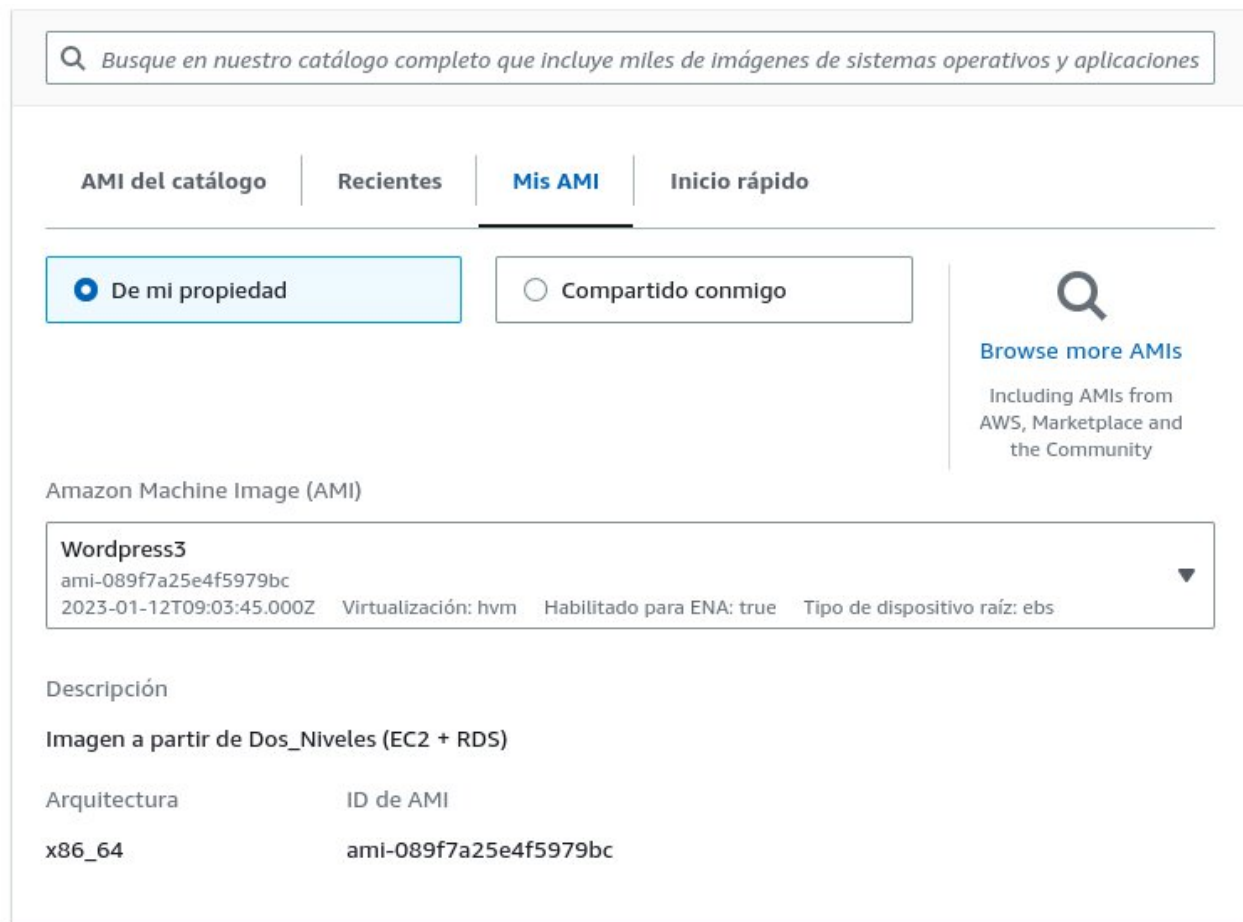
Partiremos de la instalación de wordpress en dos niveles de la práctica anterior que ya tiene una instancia RDS que aloja la base de datos de Wordpress. El objetivo es que la máquina que hace de proxy reparta las peticiones que llegan a las otras dos máquinas.

2. Modificaciones al esquema anterior

Ya teníamos la instalación de Wordpress en dos niveles: un *front-end* con los servidores de Apache corriendo en instancias del tipo EC2 y un *back-end* con una instancia del tipo RDS que aloja la base de datos. Las máquinas EC2 se crearon de manera independiente, pero se pueden clonar a partir de imágenes AMI. Para ello, elegimos la instancia que queremos clonar, desplegamos el menú de acciones y elegimos *Crear imagen* en el apartado *Imagen y plantillas*



Para lanzar una nueva instancia a partir de la imagen que hemos creado a la que deberemos haber puesto un nombre significativo y una descripción, seguimos el proceso habitual para crear instancias. Con la máquina nueva creada, ya tendrá toda la instalación de Wordpress montada y configurada para acceder a la base de datos. Esta imagen se selecciona en lugar del sistema operativo como haríamos en un lanzamiento de instancia convencional.



En este caso seleccionamos además un par de claves de las que ya tenemos generadas para los otros dos wordpress y el mismo grupo de seguridad.

3. Escenario antes del balanceador

En estos momentos tenemos tres instancias EC2 exactamente iguales dentro de la misma VPC y del mismo grupo de seguridad. Serán el *front-end* con Apache y Wordpress. Estas instancias tienen también asignada una ip elástica. Tenemos también una instancia de tipo RDS que aloja la base de datos y que está en la misma VPC que las EC2 pero en otro grupo de seguridad para poder añadir reglas específicas que controlen que el tráfico hasta ella llega solo desde Wordpress.

Instancias (4) Información								
<input type="text" value="Buscar instancia por atributo o etiqueta (case-sensitive)"/>								
<input type="checkbox"/>	Name	ID de la instancia	Estado de la instancia	Dirección IP...	IP elástica	Nombre del grupo de seguri...	Nombre de la clave	Hora de lanzamiento
<input type="checkbox"/>	Proxy	i-0e2a9d525b49b0471	En ejecución	3.88.62.122	–	launch-wizard-1	proxy	2023/02/07 17:15 GMT+1
<input type="checkbox"/>	Wordpress1	i-0cf250d5c2dafa554	En ejecución	54.152.13.214	54.152.13.214	Dos_Niveles	Dos_Niveles_Wordpress	2023/02/07 20:36 GMT+1
<input type="checkbox"/>	wordpress2	i-02fe36dd20b991b46	En ejecución	54.205.63.198	54.205.63.198	Dos_Niveles	Dos_Niveles_Wordpress	2023/02/07 17:15 GMT+1
<input type="checkbox"/>	Wordpress3	i-0b783ec5c9e0d5d44	En ejecución	3.94.104.58	3.94.104.58	Dos_Niveles	Dos_Niveles_Wordpress	2023/02/07 17:15 GMT+1

4. Balanceador de carga

Tenemos también una cuarta instancia del tipo EC2 que estará en la misma VPC pero con un grupo de seguridad propio y en la que solo se instalará Apache con las directivas del balanceador. Tendrá también su ip elástica y será la única cuyas reglas de entrada permitan el tráfico desde el exterior por el puerto 80 para http y por el 22 para la administración por ssh.

Instancias (1/4) Información									
<input type="text" value="Buscar instancia por atributo o etiqueta (case-sensitive)"/>									
<input checked="" type="checkbox"/>	Name	ID de la instancia	Estado de la i...	Tipo de inst...	Comprobación de estado	Estado de la ...	Zona de dispon...	DNS de IPv4 pública	Dirección
<input checked="" type="checkbox"/>	Proxy	i-0e2a9d525b49b0471	En ejecución	t2.micro	2/2 comprobaciones supe...	Sin alarmas +	us-east-1b	ec2-3-88-62-122.comp...	3.88.62.1
<input type="checkbox"/>	Wordpress1	i-0cf250d5c2dafa554	En ejecución	t2.micro	2/2 comprobaciones supe...	Sin alarmas +	us-east-1e	ec2-54-152-13-214.co...	54.152.12
<input type="checkbox"/>	wordpress2	i-02fe36dd20b991b46	En ejecución	t2.micro	2/2 comprobaciones supe...	Sin alarmas +	us-east-1b	ec2-54-205-63-198.co...	54.205.62
<input type="checkbox"/>	Wordpress3	i-0b783ec5c9e0d5d44	En ejecución	t2.micro	2/2 comprobaciones supe...	Sin alarmas +	us-east-1b	ec2-3-94-104-58.comp...	3.94.104.

Instancia: i-0e2a9d525b49b0471 (Proxy)

▼ Detalles de seguridad

Rol de IAM	ID del propietario	Hora de lanzamiento
–	980773835652	Tue Feb 07 2023 17:15:19 GMT+0100 (hora estándar de Europa central)

Grupos de seguridad

sg-083f3f8895d6f963c (launch-wizard-1)

▼ Reglas de entrada

Nombre	ID de la regla del grupo d...	Intervalo de pu...	Protocolo	Origen	Grupos de seguridad	Descripción
–	sg-0012452e46050a562	22	TCP	0.0.0.0/0	launch-wizard-1	–
–	sg-078c713d76732be2b	80	TCP	0.0.0.0/0	launch-wizard-1	–

En el archivo `/etc/apache2/sites-available/000-default.conf` añadimos la directiva:

```
ProxyPass /balancer-manager !
```

```

#Server1
BalancerMember http://IP_privada_wordpress1 loadfactor=3 timeout=1
#Server2
BalancerMember http://IP_privada_wordpress2 loadfactor=2
#Server3
BalancerMember http://IP_privada_wordpress3 loadfactor=1

ProxyPass "/" "balancer://mycluster"
ProxyPassReverse "/" "balancer://mycluster"

SetHandler balancer-manager
Order Deny,Allow
Allow from all

```

```

<Proxy balancer://mycluster>
    #Server1 Wordpress1
    BalancerMember http://172.31.60.172 loadfactor=3 timeout=1
    #Server2 Wordpress3
    BalancerMember http://172.31.80.198 loadfactor=2
    #Server3
    BalancerMember http://172.31.82.72 loadfactor=1
</Proxy>
ProxyPass "/" "balancer://mycluster"
ProxyPassReverse "/" "balancer://mycluster"
<Location /balancer-manager>
    SetHandler balancer-manager
    Order Deny,Allow
    Allow from all
</Location>

```

Instancia: i-0b783ec5c9e0d5d44 (Wordpress3)

Detalles Seguridad Redes Almacenamiento Comprobaciones de estado

Resumen de instancia Información

ID de la instancia Dirección IPv4 pública 3.94.104.58 | dirección

Dirección IPv6 Estado de la instancia

Recordad que para que el proxy funcione sobre http hay que habilitar el módulo correspondiente con `sudo a2enmod proxy_http`

Ya podemos configurar las reglas de entrada del grupo de seguridad al que pertenecen todas las instancias del *front-end* que en mi caso se llama *Dos_Niveles* porque es el mismo de la práctica anterior. En estas reglas permitiremos el tráfico de entrada por el puerto 22 (para administración) y 80 pero solo desde el grupo al que pertenece el proxy. De esa manera, las máquinas EC2 con el gestor de contenidos quedan aisladas como en el esquema que planteamos al comienzo.

Las reglas del grupo de seguridad de entrada se han modificado correctamente en el grupo de seguridad (sg-0a3e7e612878cf69a | Dos_Niveles)

► Detalles

Grupos de seguridad (1/3) Información

🔄 Acciones ▼ Exportar los grupos de seguridad a CSV ▼ Crear grupo de seguridad

🔍 Filtrar grupos de seguridad

	Name	ID del grupo de segu...	Nombre del grupo ...	ID de la VPC	Descripción	Propietario	Número de reglas d...	Número de reg
<input checked="" type="checkbox"/>	-	sg-0a3e7e612878cf69a	Dos_Niveles	vpc-0f09b09ede7696d55	Created by RDS manag...	980773835652	2 Entradas de permisos	1 Entrada de p
<input type="checkbox"/>	-	sg-07745539ef6afe24	default	vpc-0f09b09ede7696d55	default VPC security gr...	980773835652	1 Entrada de permiso	1 Entrada de p
<input type="checkbox"/>	-	sg-083f3f8895d6f963c	launch-wizard-1	vpc-0f09b09ede7696d55	launch-wizard-1 create...	980773835652	2 Entradas de permisos	1 Entrada de p

sg-0a3e7e612878cf69a - Dos_Niveles

Detalles Reglas de entrada Reglas de salida Etiquetas

📌 Ahora puede comprobar la conectividad de red con Reachability Analyzer Ejecutar Reachability Analyzer ✕

Reglas de entrada (1/2)

🔄 Administrar etiquetas Editar reglas de entrada

🔍 Filtrar reglas de grupo de seguridad

	Name	ID de la regla de de...	Versión de IP	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción
<input type="checkbox"/>	HTTP desde el proxy	sgr-06bfc7ef09dc5...	-	HTTP	TCP	80	sg-083f3f8895d6f963...	Acceso desde el proxy
<input checked="" type="checkbox"/>	Administración	sgr-0d574e3a3478...	IPv4	SSH	TCP	22	0.0.0.0/0	-

5. NFS

NFS (Net File System) es un sistema de archivos en red de Linux que permite conectar distintas máquinas con una que almacena todos los ficheros y directorios necesarios para el correcto funcionamiento de las primeras. De esta manera, en clusters con alta disponibilidad, la información está alojada en una máquina accesible por cada uno de los servidores que forman el cluster y cualquier modificación es sincronizada automáticamente en cada uno de dichos servidores.

5.1 Implementación

El servicio NFS lo vamos a implementar en una nueva instancia EC2. Como vamos a alojar ahí el wordpress que ya tenemos creado, la máquina la vamos a crear con la misma imagen AMI de antes.

```
ubuntu@ip-172-31-63-151:~$ sudo systemctl status nfs-kernel-server.service
● nfs-server.service - NFS server and services
   Loaded: loaded (/lib/systemd/system/nfs-server.service; enabled; vendor preset: enabled)
   Active: active (exited) since Tue 2023-02-07 22:41:35 UTC; 38s ago
     Main PID: 2552 (code=exited, status=0/SUCCESS)
       CPU: 4ms

Feb 07 22:41:34 ip-172-31-63-151 systemd[1]: Starting NFS server and services...
Feb 07 22:41:34 ip-172-31-63-151 exportfs[2551]: exportfs: can't open /etc/exports for reading
Feb 07 22:41:35 ip-172-31-63-151 systemd[1]: Finished NFS server and services.
ubuntu@ip-172-31-63-151:~$
```

Resumen de instancia de i-0d2a2a4afc45168f8 (NFS) Información

Se ha actualizado hace less than a minute

Para que el servicio NFS comparta los archivos con las máquinas del *front-end* necesitamos modificar el archivo `/etc/exports` como se indica a continuación:

```
#para el wordpress1
/var/www/html 172.31.60.172(rw,sync,no_root_squash,no_subtree_check)
```

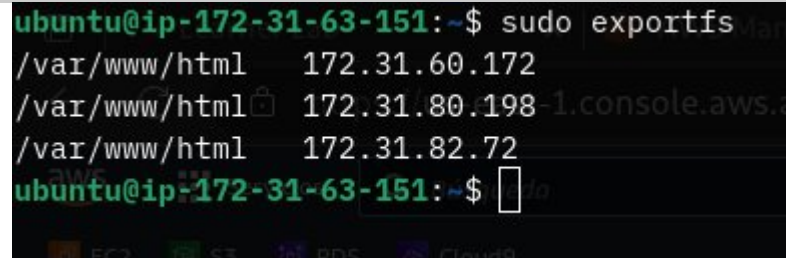


```
#para el wordpress2
```

```
/var/www/html 172.31.80.198(rw,sync,no_root_squash,no_subtree_check)
```

```
#para el wordpress3
```

```
/var/www/html 172.31.82.72(rw,sync,no_root_squash,no_subtree_check)
```



```
ubuntu@ip-172-31-63-151:~$ sudo exportfs
/var/www/html 172.31.60.172
/var/www/html 172.31.80.198
/var/www/html 172.31.82.72
ubuntu@ip-172-31-63-151:~$
```

El servicio está activo y compartiendo, pero las instancias no están accediendo a estos ficheros aun sino a los que tienen en local. La vinculación de NFS se hace como un montaje normal de un dispositivo, solo que indicando la IP de origen y la ruta (instalamos primero `nfs-client`) y lo añadimos con `sudo mount 172.31.63.151:/var/www/html /var/www/html`. Hay que tener en cuenta añadir una regla más al grupo de seguridad para que pueda conectarse entre si. Para que los cambios sean permanentes y se sincronice, tenemos que editar el archivo `/etc/fstab`

```
172.31.63.151:/var/www/html /var/www/html nfs auto,nofail,noatime,nolock,intr,tcp,actimeo=1800 0 0
```

Y hacemos esto para cada una de las EC2 de Wordpress