

IMPLANTACIÓN DE APLICACIONES WEB

IAW

4.5 Finalizar escenario cluster web

✎ Otero Gómez, Manuela
2º ASIR
IAW
15.02.2023

Tabla de contenidos

1. Alta seguridad.....	4
2. Alto rendimiento.....	5
3. Alta disponibilidad.....	6
4. Escalabilidad horizontal y vertical.....	6
5. Resultado final:.....	7
6. GitHub	11

1. Alta seguridad

Sobre el escenario de las prácticas anteriores, hacemos un resumen de qué políticas de seguridad se han tomado y cómo queda el escenario (todas las instancias están en la misma VPC):

Por un lado tenemos la seguridad en la infraestructura montada: desde el exterior solo se puede acceder al proxy que será el que redirija el tráfico hacia las instancias del *back-end*:

Tenemos en las instancias de tipo EC2 dos grupos de seguridad independientes: el grupo al que pertenece solo la instancia del proxy (launch-wizard-1) tiene las siguientes reglas

EC2 > Grupos de seguridad > sg-08c75bacfb5d6055 - launch-wizard-1 > Editar reglas de entrada

Editar reglas de entrada [Información](#)

Las reglas de entrada controlan el tráfico entrante que puede llegar a la instancia.

ID de la regla del grupo de seguridad	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción: opcional	
sgr-0df868d3772a24e86	HTTP	TCP	80	Personaliz... <input type="text" value="0.0.0.0/0"/>		Eliminar

[Agregar regla](#)

Como hemos dicho, solo se puede acceder por el puerto 80 (http) desde cualquier IP.

Todas las instancias EC2, incluida la máquina de Windows que usamos para la administración de la base de datos y será temporal, están en otro grupo llamado SGWeb con las siguientes reglas:

EC2 > Grupos de seguridad > sg-02be7208fad77fde6 - SGWeb > Editar reglas de entrada

Editar reglas de entrada [Información](#)

Las reglas de entrada controlan el tráfico entrante que puede llegar a la instancia.

ID de la regla del grupo de seguridad	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción: opcional	
sgr-0b50b19d016f692af	SSH	TCP	22	Personaliz... <input type="text" value="0.0.0.0/0"/>		Eliminar
sgr-01d9223495d2c4abd	RDP	TCP	3389	Personaliz... <input type="text" value="0.0.0.0/0"/>		Eliminar
-	HTTP	TCP	80	Personaliz... <input type="text" value="sg-08c75bacfb5d6055"/>		Eliminar

[Agregar regla](#)

La regla del SSH que permite el tráfico desde cualquier origen está solo para la administración de las instancias. Se eliminará al terminal el despliegue.

Lo mismo ocurrirá con la regla del RDP, el escritorio remoto de Windows, que se creó para poder conectarnos a la EC2 con Windows Server pero que se eliminará cuando esta instancia también lo haga.

Por último, la regla de HTTP permite la entrada solo desde el grupo de seguridad launch-wizard-1 al que pertenece el proxy, es decir, solo podrán acceder a estas instancias las conexiones por el puerto 80 que vengan desde el la instancia Proxy.

Hay un grupo de seguridad exclusivo para la base de datos que solo tiene una regla que permite el acceso desde el grupo de seguridad SGWeb, es decir. A la base de datos solo podrán acceder las IPs de los apaches que a su vez vienen redirigidas por el proxy aportando así una capa más de filtrado:

EC2 > Grupos de seguridad > sg-04717ce684ec4cec - SeguridadMySQL > Editar reglas de entrada

Editar reglas de entrada [Información](#)

Las reglas de entrada controlan el tráfico entrante que puede llegar a la instancia.

ID de la regla del grupo de seguridad	Tipo Información	Protocolo Información	Intervalo de puertos Información	Origen Información	Descripción: opcional Información
sgr-04b9434931c2cc7df	MySQL/Aurora	TCP	3306	Personaliz... <input type="text" value="sg-02b67208fad77fde6"/>	<input type="text" value=""/>

Por último tenemos el grupo de seguridad del almacenamiento EFS. A él solo pueden acceder las máquinas de Apache que tienen montado el directorio `efs_mount` donde se almacenan todos los archivos que dan forma a la página web. El puerto es el predeterminado para este tipo de conexión.

EC2 > Grupos de seguridad > sg-0b04d122a2b1715d7 - SGDFS > Editar reglas de entrada

Editar reglas de entrada [Información](#)

Las reglas de entrada controlan el tráfico entrante que puede llegar a la instancia.

ID de la regla del grupo de seguridad	Tipo Información	Protocolo Información	Intervalo de puertos Información	Origen Información	Descripción: opcional Información
sgr-03f64edd34b230282	NFS	TCP	2049	Personaliz... <input type="text" value="sg-02b67208fad77fde6"/>	<input type="text" value=""/>

2. Alto rendimiento

El alto rendimiento se consigue gracias a la duplicidad de las instancias EC2 que proporcionan el servicio web de Apache corriendo sobre un Sistema Operativo Amazon Linux. Estas instancias están conectadas a un servicio de almacenamiento EFS. Además el proxy tiene balanceador de carga, por lo que al aumentar el número de peticiones, estas se van a repartir entre ambas instancias de manera transparente al usuario. La configuración del proxy es la siguiente:

```

GNU nano 6.2 /etc/apache2/sites-available/000-default.conf

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# 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://mycluster>
  #Server1
  BalancerMember http://172.31.9.63
  #Server2
  BalancerMember http://172.31.86.197
</Proxy>

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

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

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

```

En esta configuración, el cuello de botella lo haría el proxy, que es una sola instancia de Ubuntu Server que tendría que gestionar, redirigir y balancear todas las peticiones hacia las máquinas del *back-end*.

3. Alta disponibilidad

En este despliegue tenemos duplicidad tanto en los servidores web con las dos instancias EC2 de Amazon Linux mencionadas antes como en dos instancias de bases de datos RDS configuradas en cluster como se explicó en la práctica anterior. Las instancias multi A-Z se encuentran además en zonas diferentes, por lo que si hay un problema físico en una de las granjas de servidores de Amazon, la otra instancia RDS seguiría funcional.

Respecto a estas instancias, solo se puede leer y escribir en una de ellas, la otra está perfectamente sincronizada, pero solo recibirá los datos provenientes de los apaches si la primera no responde.

Las instancias EC2 tienen IP elástica y es la IP privada la que aparece en el balanceador para que no cambie entre reinicios.

El almacenamiento de todos los archivos necesarios para que nuestra web dinámica funcione están alojados en un servicio EFS. Este servicio de Amazon proporciona un almacenamiento de tipo NFS cuyo montaje en los Apaches está en el directorio `/var/www/html/efs_mount` (se ha modificado el document root de Apache para que sea esa la carpeta raíz). Ahí se encuentran tanto el `index.html` y el `style.css` que da forma a la página como los archivos `.php` que proporcionan el formulario y la conexión con la base de datos. De esta manera la edición de cualquiera de estos archivos se ve inmediatamente reflejada en la otra instancia puesto que se está modificando el contenido de un punto de montaje externo.

4. Escalabilidad horizontal y vertical

Toda la configuración aquí descrita proporciona un entorno bastionado y de alta disponibilidad para el despliegue planteado en las prácticas. Pero no solo es seguro y tiene alta disponibilidad, también se puede ampliar fácilmente si nuestro sitio web requiriese de más infraestructura:

Ya sabemos crear imágenes tipo AMI en amazon, por lo que para levantar más servicios tanto proxy como web solo tendríamos que crear la imagen y levantar tantas instancias como fuesen necesarias, durante la creación los añadimos a los grupos de seguridad pertinentes y funcionarían sin afectar a la disponibilidad del sitio.

Las RDS también son escalables, de hecho, todos los servicios que hemos visto a lo largo del curso tienen la opción de autoescalado por parte de AWS.

Almacenamiento

Tipo de almacenamiento [Información](#)

SSD de IOPS provisionadas (io1)
Flexibilidad en el aprovisionamiento de E/S

Almacenamiento asignado [Información](#)

100

GiB



El valor mínimo es 100 GiB y el valor máximo es 6144 GiB

IOPS provisionadas [Información](#)

1000

IOPS

El valor mínimo es 1000 IOPS y el valor máximo es 256.000 IOPS. La proporción de tamaño de almacenamiento debe estar entre 0,1 y 1000. La proporción de tamaño de almacenamiento es la relación entre el almacenamiento asignado y las IOPS aprovisionadas.

 Las IOPS reales pueden diferir de la cantidad aprovisionada en función del tipo de instancia y de la carga de trabajo de la base de datos. [Más información](#) 

Escalado automático de almacenamiento [Información](#)

Proporciona compatibilidad con el escalado dinámico para el almacenamiento de la base de datos en función de las necesidades de la aplicación.

☒ **Habilitar escalado automático de almacenamiento**
Si se habilita esta característica, el almacenamiento podrá aumentar después de que se supere el umbral especificado.

Umbral de almacenamiento máximo [Información](#)

Los cargos se aplicarán cuando la base de datos escale automáticamente el umbral especificado.

1000

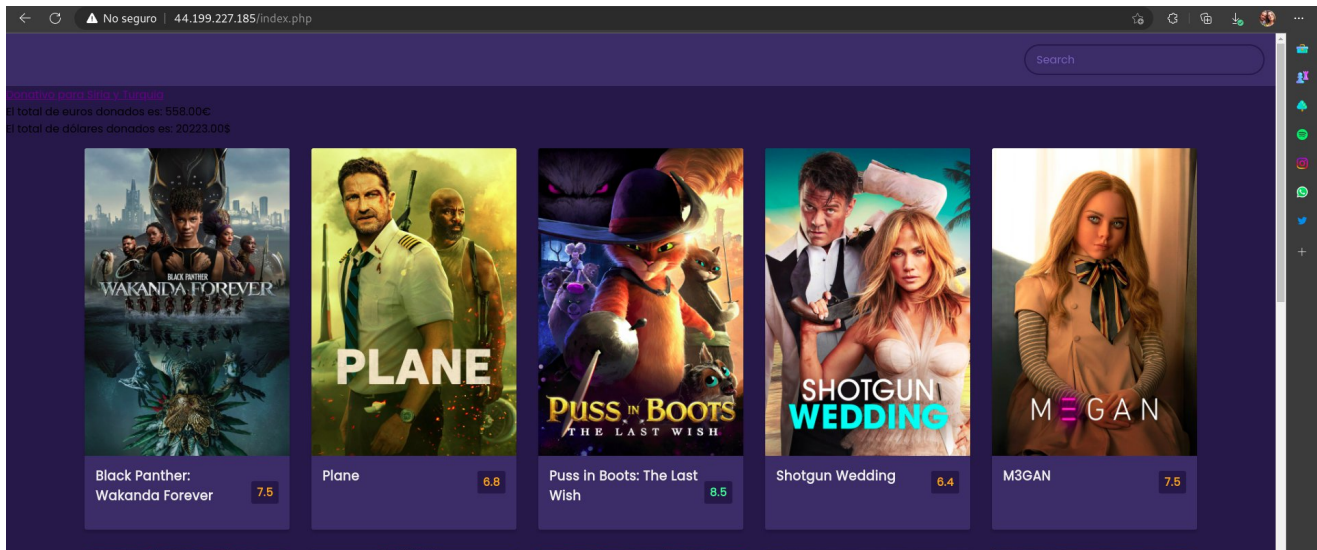
GiB

El valor mínimo es 110 GiB y el valor máximo es 6144 GiB

5. Resultado final:

De cara a un cliente lo que se vería sería esta sucesión de pantallas (mejorables en estilo pero funcionales):

Cuando se accede a la IP pública del proxy:



Si entras al enlace de los donativos:

A screenshot of a web browser displaying a donation form. The address bar shows '44.199.227.185/formulario.php'. The form has a dark purple background with the title 'DONATIVOS PARA TURQUIA Y SIRIA' in large white letters. Below the title, there are two input fields: 'Nombre:' with the value 'Javier' and 'Donativo:' with the value '1000'. There are two radio buttons for currency selection: 'Euro' (selected) and 'Dolar'. At the bottom, there is an 'Enviar' button and a link 'Volver a inicio' in purple text.

Cuando envías el donativo:



[Volver a inicio](#)

En la parte de atrás hay un proxy con reenviador cuya configuración ya está descrita más arriba. El Index ahora tiene extensión .php porque se ha incluido la consulta para que se muestre el total de donaciones acumuladas:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <link rel="stylesheet" href="style.css" />
  <title>Movie App</title>
</head>
<body>
  <header>
    <form id="form">
      <input type="text" id="search" class="search" placeholder="Search">
    </form>
  </header>
  <a href="formulario.php">Donativo para Siria y Turquía</a><br>

  <?php

    include ('conexion.php');
    $consulta_euros = "SELECT SUM(donativo) AS total_euros FROM donativos WHERE tipoMoneda = 'euro'";
    $consulta_dolares = "SELECT SUM(donativo) AS total_dolares FROM donativos WHERE tipoMoneda = 'dolar'";

    $resultado_euros = mysqli_query($conn, $consulta_euros);
    $resultado_dolares = mysqli_query($conn, $consulta_dolares);

    $row_euros = mysqli_fetch_assoc($resultado_euros);
    $row_dolares = mysqli_fetch_assoc($resultado_dolares);

    $total_euros = $row_euros['total_euros'];
    $total_dolares = $row_dolares['total_dolares'];

    echo "El total de euros donados es: " . $total_euros . "€";
    echo "<br>";
    echo "El total de dólares donados es: " . $total_dolares . "$";

    mysqli_close($conn);

  ?>

  <main id="main"></main>
  <script src="script.js"></script>
</body>
</html>
```

En el formulario.php hemos añadido el head del index para que conserve la gama de colores y el resultado final es:

```
</html>
```

```

<head>
<meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <link rel="stylesheet" href="style.css" />
  <title>Movie App</title>
</head>
<body>
<h1> DONATIVOS PARA TURQUIA Y SIRIA </h1>
<form action="grabar.php" method="post">

  <label for="name">Nombre:</label>
  <input type="text" id="nombre" name="nombre">
  <br>
  <label for="donativo">Donativo</label>
  <input type="text" id="donativo" name="donativo">
  <br>

  <input type="radio" name="tipoMoneda" value="euro" checked > Euro<br>

  <input type="radio" name="tipoMoneda" value="dolar"> Dolar<br>
  <br>

  <input type="submit" value="Enviar" name="enviar" />
  <a href="/index.php">Volver a inicio</a>
</form>
</body>
</html>

```

La configuración del archivo que permite la conexión con la base de datos:

```

<?php
$servername = "cluster-rds.ccnndtyrk3ef.us-east-1.rds.amazonaws.com";
$username = "admin";
$password = "Root1234$";
$db = "cluster";
// Create connection
$conn = new mysqli($servername, $username, $password, $db);

// Check connection
if ($conn->connect_error) {
    die("Conexión falla " . $conn->connect_error);
}
//echo "Conexión exitosa";
?>

```

Y el del que inserta los datos en la RDS:

```

<?php
include ('conexion.php');
echo "<h1>Donación guardada correctamente. Gracias por su colaboración</h1><br>";

$sql = "INSERT INTO cluster.donativos (id, nombre, donativo, tipoMoneda)

```

```
values (NULL, "", $_POST['nombre'], "", $_POST['donativo'], "", $_POST['tipoMoneda'], "");

if (mysqli_query($conn, $sql)) {
//  echo "New record created successfully<br>";
} else {
    echo "Error:" . $sql. "<br>" . mysqli_error($conn);
}

mysqli_close($conn);

echo "<a href='index.php'>Volver a inicio</a>"

?>
```

6. GitHub

https://github.com/a21otgoma/2ASIR_IAW