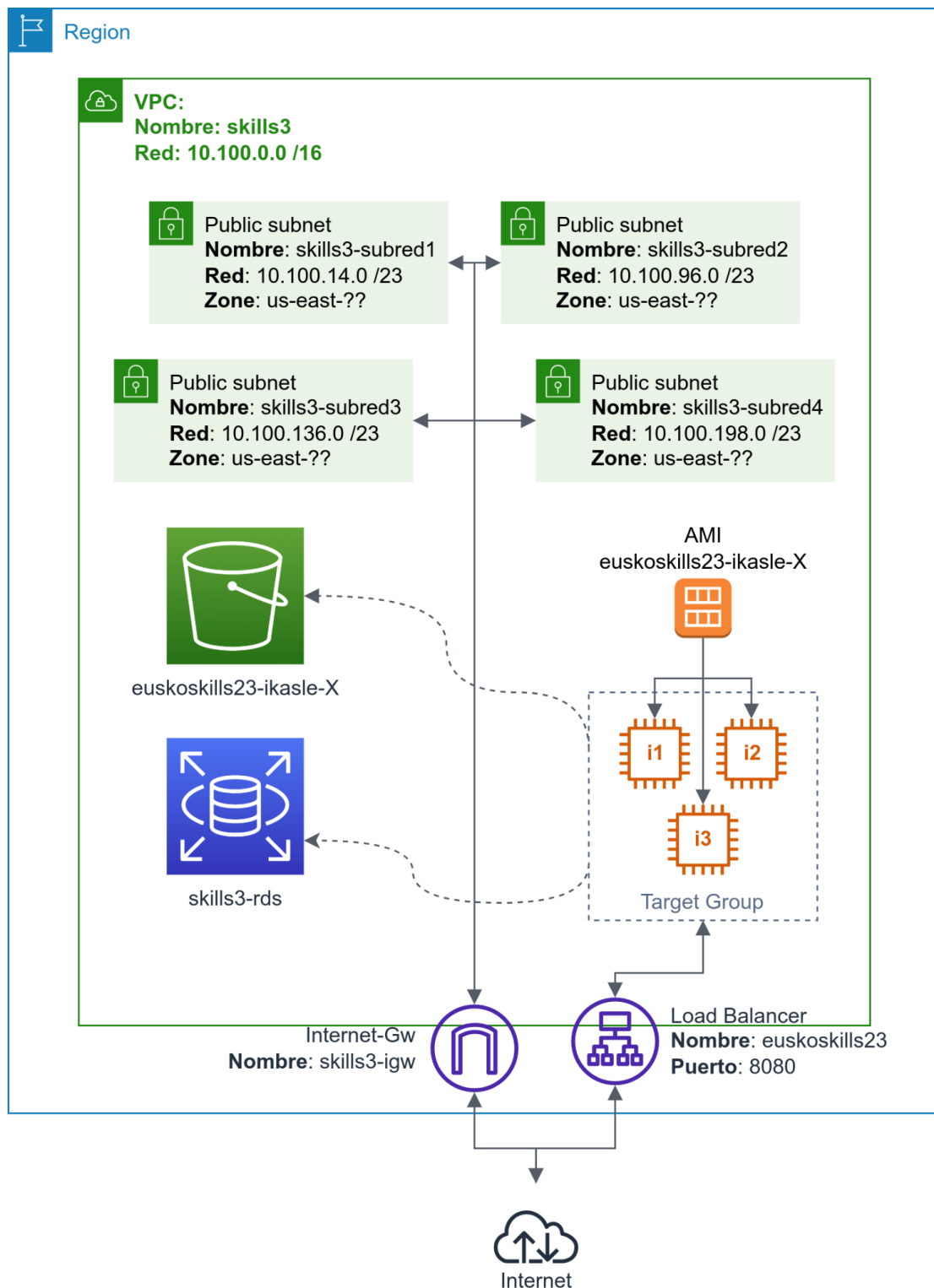


## Prueba 3 - Cloud en Alta Disponibilidad (3h)

Se necesita crear una infraestructura en Alta Disponibilidad dentro de la infraestructura AWS. El esquema resumido de la prueba es el siguiente:



**¡Atención! Prestad atención a:**

- Los nombres que se deben asignar deben ser iguales a los especificados en la rúbrica (cuidado con mayúsculas, minúsculas y guiones).
- Al crear las instancias haced uso de las claves SSH "**vockey**" que están creadas por defecto en el laboratorio.
  - Para acceder a la instancia privada necesitáis la clave en la instancia pública.
  - Para las instancias se usará el **sistema operativo Ubuntu**.

**¡Seguid el orden de la rúbrica para guiaros a la hora de realizar la prueba!**

## VPC

Se pide crear un VPC personalizado con un direccionamiento global (**10.100.0.0 /16**) en el que queremos crear 4 subredes en 4 zonas de disponibilidad diferentes. Los nombres y direccionamientos son:

- skills3-subred1 : 10.100.14.0 /23
- skills3-subred2 : 10.100.96.0 /23
- skills3-subred3 : 10.100.136.0 /23
- skills3-subred4 : 10.100.198.0 /23

La rúbrica consta de **4 puntos** dividido en:

VPC creado con <b>10.100.0.0 /16</b>	0,5
El VPC tiene el nombre " <b>skills3</b> "	0,5
Se han creado 4 subredes públicas distintas en <b>4 zonas de disponibilidad diferentes</b>	0,5
Los nombres de las subredes son: - skills3-subred1 - skills3-subred2 - skills3-subred3 - skills3-subred4	0,5
Los direccionamientos de las subredes son: - 10.100.14.0 /23 - 10.100.96.0 /23 - 10.100.136.0 /23 - 10.100.198.0 /23	0,5
Se ha creado un Internet-GW llamado <b>skills3-igw</b>	0,5
La subred pública tiene una ruta 0.0.0.0 a <b>Internet-GW</b>	1

## RDS

Se tiene que crear una Base de Datos Relacional a través del servicio RDS que debe cumplir con las siguientes características:

- **MariaDB** versión **10.6.8**
  - Nombre: **skills3-rds**
  - Usuario maestro: **admin**
  - Contraseña maestra: **3usk0sk1lls2023**
  - Plantilla: **capa gratuita**
  - Clase de instancia: **db.t3.micro**
  - Almacenamiento asignado: **20GB**

El acceso público al RDS debe estar permitido, ya que se requiere poder acceder a través de su IP pública para realizar lo siguiente:

- Crear base de datos: **prueba3**
- Importar el backup obtenido de <https://euskoskills23.s3.amazonaws.com/dump.sql>

**¡NOTA!** Se recomienda hacer uso del cliente [DBeaaver](#) para confirmar que el acceso funciona.

La rúbrica consta de **5 puntos** dividido en:

Ha creado un RDS con <b>MariaDB</b> llamado <b>skills3-rds</b>	0,5
La instancia RDS tiene todas las características especificadas en el documento	2
El RDS creado está en el nuevo VPC creado	0,25
Ha creado la base de datos " <b>prueba3</b> "	0,25
Ha importado los datos correctamente en <b>prueba3</b>	1
El acceso desde el exterior al puerto 3306 está permitido	0,5
Tiene un security group propio " <b>rds</b> " que sólo permite el acceso al puerto 3306	0,5

## S3

A continuación se requiere la creación de un bucket S3 **público** cuyo nombre debe ser **euskoskills23-ikasle-X** (donde X es tu número de alumno).

Para confirmar que este bucket es público debes:

- Subir un fichero **prueba.txt** cuyo contenido sea tu propio nombre. El fichero se debe poder descargar haciendo uso de la URL que se genera para el objeto.

Aparte, este bucket será utilizado para la siguiente prueba, por lo que tienes que crear la ruta siguiente dentro del bucket:

- **backups/datos**

La rúbrica consta de **3 puntos** dividido en:

Se ha creado un bucket S3	0,5
El nombre del bucket es "euskoskills23-ikasle-X"	0,5
El bucket es público	0,5
Se ha subido un fichero <b>prueba.txt</b> cuyo contenido es el nombre del ikasle y se puede descargar públicamente	1
Se ha creado la estructura de carpetas " <b>backups/datos</b> "	0,5

## Creación de una AMI con una instancia preparada

Se quiere crear una instancia con servicios instalados y pre-configurados para después crear una **AMI** que será desplegada en la última sección.

La instancia para crear la AMI debe tener las siguientes especificaciones:

- El sistema operativo es **Debian 11**
  - El usuario para acceder con las claves SSH es **admin**
- Debe tener el servicio Apache instalado
- Al acceder a su puerto 80, se tiene que ejecutar directamente el script **index.php** que debe:
  - Mostrar el **hostname** de la instancia en la que está siendo ejecutado.
  - Lista la tabla **tutores** del **RDS** creado previamente.
- Se ha configurado un **CRON** (en la configuración del sistema de **/etc/crontab**):
  - La ejecución como **root** cada **10 minutos** del script **/opt/backups.sh** que debes crear y debe hacer:
    - Comprime **/var/www/** en un fichero comprimido con extensión **".tar.gz"**

- **¡NOTA!** Para optar a más puntos se pide que el nombre sea: `/tmp/hostname_YYYYMMDD_HHmm.tar.gz` donde:
  - hostname → el nombre de la instancia
  - YYYYMMDD → fecha en formato AñoMesDía. Ejemplo 20230509 (ejemplo para el 9 de mayo de 2023)
  - HHmm → hora en formato HoraMinutos. Ejemplo: 1443
- Una vez generado el fichero comprimido el script lo sube al **bucket S3** creado previamente, en la ruta **backups/datos/**

**¡NOTA!** Para poder subir ficheros a S3 tienes que utilizar el comando **aws** instalado en Debian. Los datos para hacer funcionar el comando están en la parte **“AWS Details”** al **arrancar el Learner Lab**. El resto, queda de vuestra parte investigar cómo se usa.

Una vez creada esta instancia, hay que crear la AMI.

- La AMI generada tiene que llamarse: **euskoskills23-ikasle-X**

Con todo esto, al levantar una instancia con esta AMI generada, tanto el servicio Apache como el CRON deben de funcionar.

La rúbrica consta de **8 puntos** dividido en:

La AMI generada se llama <b>euskoskills23-ikasle-X</b>	0,5
El sistema operativo es Debian 11	0,5
Tiene instalado Apache	0,25
Al acceder al puerto 80 de la instancia se ejecuta el <b>index.php</b> automáticamente	0,25
El index.php muestra el hostname de la instancia	0,5
El index.php lista los datos de la tabla <b>"tutores"</b> del RDS	1,5
Se ha configurado un CRON que se ejecuta cada 10 minutos como root en /etc/crontab y ejecuta el script <b>/opt/backup.sh</b>	0,5
El script <b>/opt/backup.sh</b> comprime /var/www/ en un fichero con extensión <b>.tar.gz</b>	0,5
El script <b>/opt/backup.sh</b> comprime /var/www/ como <b>/tmp/hostname_YYYYMMDD_HHmm.tar.gz</b>	1,5
El script sube al bucket S3 el fichero recién creado del script en la ruta <b>"backups/datos/"</b>	2

## Creación de un balanceador

Tras crear la AMI, es momento de hacer uso de ella, y para ello previamente se va a crear un **Balanceador de Carga** (o **Load Balancer**).

El balanceador se debe llamar **euskoskills23** y debe de cumplir:

- Estar creado en el nuevo VPC de esta prueba
- Utilizar todas las subredes
- Utilizará el puerto **8080** en internet
- Usará sólo un security-group llamado **euskoskills23-lb**, que sólo acepta peticiones en el puerto 8080
- Utilizará un target-group llamado **euskoskills23-tg**

A la hora de crear el target-group **euskoskills23-tg**:

- Tendrá en cuenta que el puerto **80** de las instancias responde, de esta manera sabrá si hay que mandarle peticiones o no.

Sobre las **instancias** que deben ir en este target-group, se debe cumplir:

- Debe existir 3 instancias en el target-group **sin IP pública**
  - **¡NOTA!** Para asegurar que las peticiones del balanceador llegan a las instancias, deben estar “registradas” y en modo “healthy”.
- Se deben de haber generado utilizando la AMI previamente creada
- Cada instancia pertenece a una zona de disponibilidad diferente dentro del VPC
- Las instancias sólo tienen un security-group llamado **euskoskills23-tg**
  - Sólo acepta peticiones en el puerto **80** desde la red del VPC

**¡NOTA!** Para comprobar y asegurar que el balanceador está utilizando todas las instancias, debes utilizar la URL del balanceador en el navegador en **modo privado**, o hacer uso del comando **curl** en la consola de Linux.

La rúbrica consta de **14 puntos** dividido en:

Se ha creado un balanceador llamado <b>euskoskills23</b> en el VPC creado previamente	2
El balanceador utiliza todas las subredes del nuevo VPC	0,5
El balanceador escucha en el protocolo HTTP en el puerto 8080	0,5
El balanceador sólo tiene un security group propio llamado " <b>euskoskills23-lb</b> " que sólo acepta conexiones en el puerto 8080 desde cualquier IP	0,5

Se ha creado un target group llamado <b>euskoskills23-tg</b>	1
El target group comprueba el puerto 80 de las instancias para mandarles peticiones	1
El target group está asociado al balanceador	1
Se han creado 3 instancias <b>generadas con la AMI</b>	2
Las instancias <b>no tienen IP pública</b>	1
Las 3 instancias están registradas en el target-group <b>euskoskills23-tg</b> asignado al balanceador	1
Las instancias generadas con la AMI sólo tienen un security group " <b>euskoskills23-tg</b> " que sólo permite el acceso al puerto 80 desde sólo la red del VPC	0,5
Las 3 instancias generadas pertenecen a zonas de disponibilidad diferentes	1
Al realizar peticiones a la URL del balanceador en el puerto 8080, alternativamente sale el hostname de distintas instancias y los datos del RDS (mirar index.php del apartado anterior)	2