Grado en Ingeniería de Software

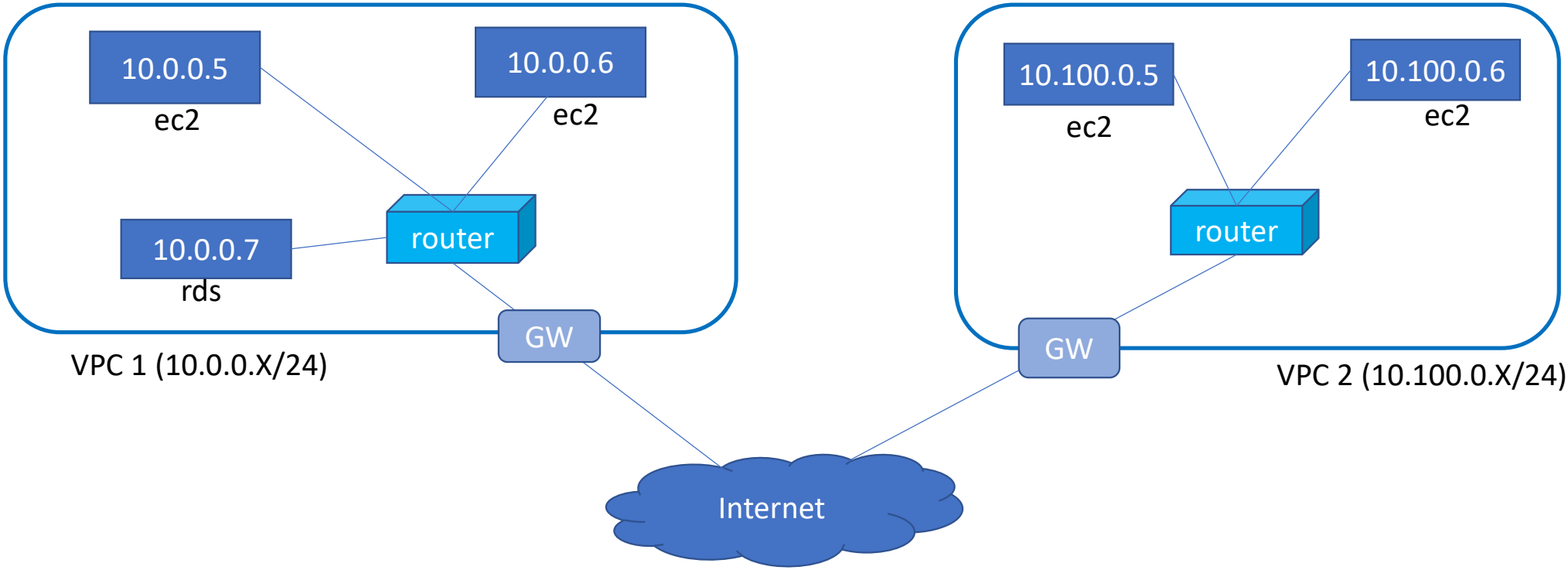
Laboratorio de Bases de Datos y Sistemas Distribuidos.

Diseño de conectividad en AWS para despliegue de una solución con un servidor WEB y un servidor de BBDD

Elena G. Gamella

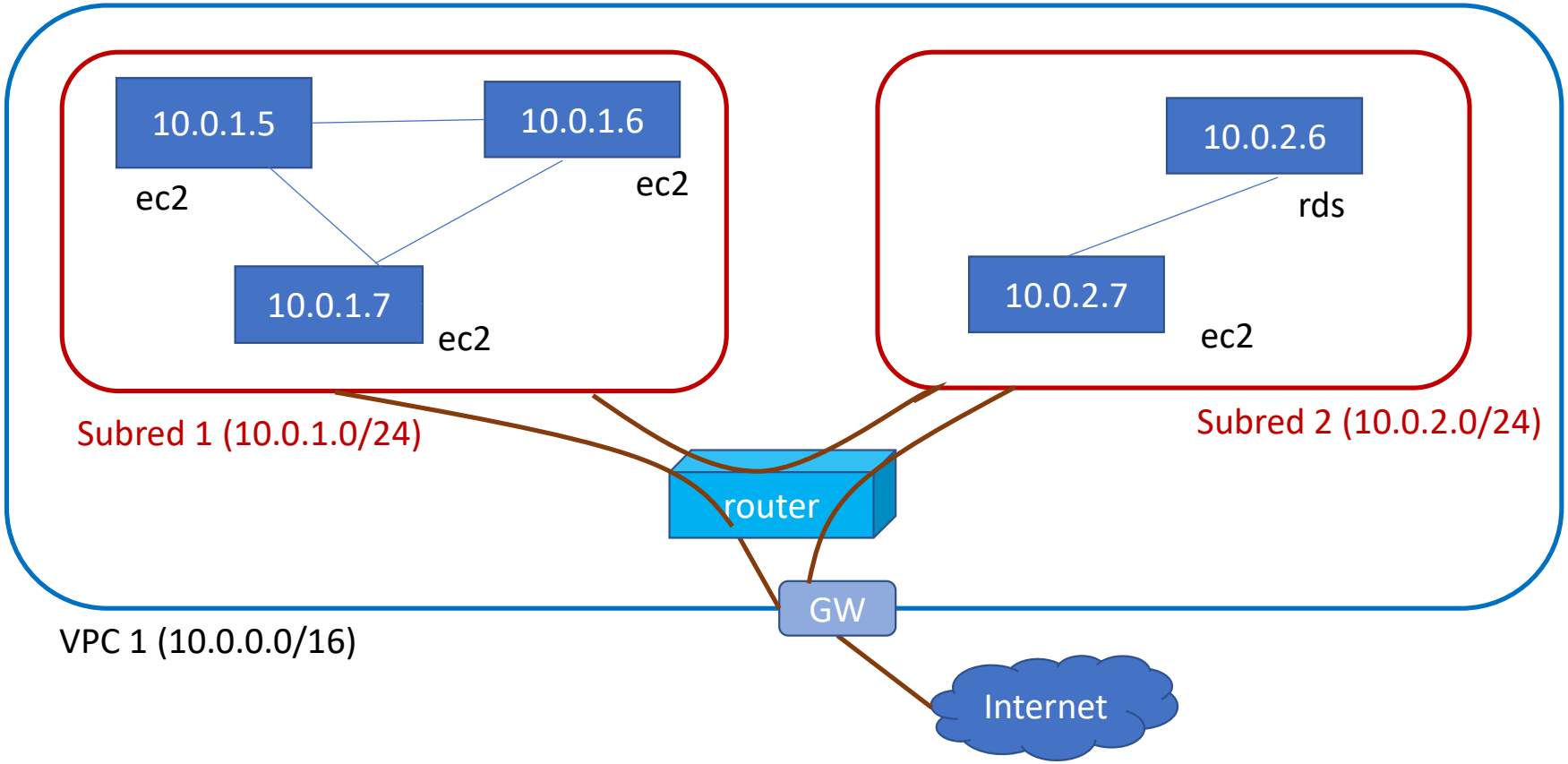
Algunos conceptos básicos antes de emepzar……

Ejemplo de dos redes de ordenadores en AWS sin conexión entre ellas, ambas conectadas a internet



Algunos conceptos básicos antes de empezar…

Ejemplo de una red con dos subredes de ordenadores en AWS conectadas entre ellas y solo una de ellas conectada a internet



VAMOS A PRACTICAR……

CREAREMOS UNA INFRAESTRUCTURA DE RED Y DOS ESCENARIOS DIFERENTES EN AWS

ESCENARIO 1

BASADO EN UNA INFRAESTRUCTURA DE RED Y DOS MÁQUINAS EC2s

Vamos a practicar en AWS: **ESCENARIO 1**

Debemos diseñar un entorno distribuido basado en Cloud Computing para dar servicio a una empresa. El sistema debe constar principalmente de dos entornos montados sobre una infraestructura de red propia, por tanto, se pide:

1. **Crear la infraestructura de red (VPC, subredes, routers, etc) que se describe en las siguientes slides.**
2. **Crear un ordenador (EC2) que hará de servidor de páginas WEBs** (SW Apache).
3. **Un ordenador (EC2) en que hará de servidor de Base de Datos** (SGBD MariaDB).

La solución debe diseñarse en Amazon Web Services (AWS) y la arquitectura es la que se muestra a continuación:

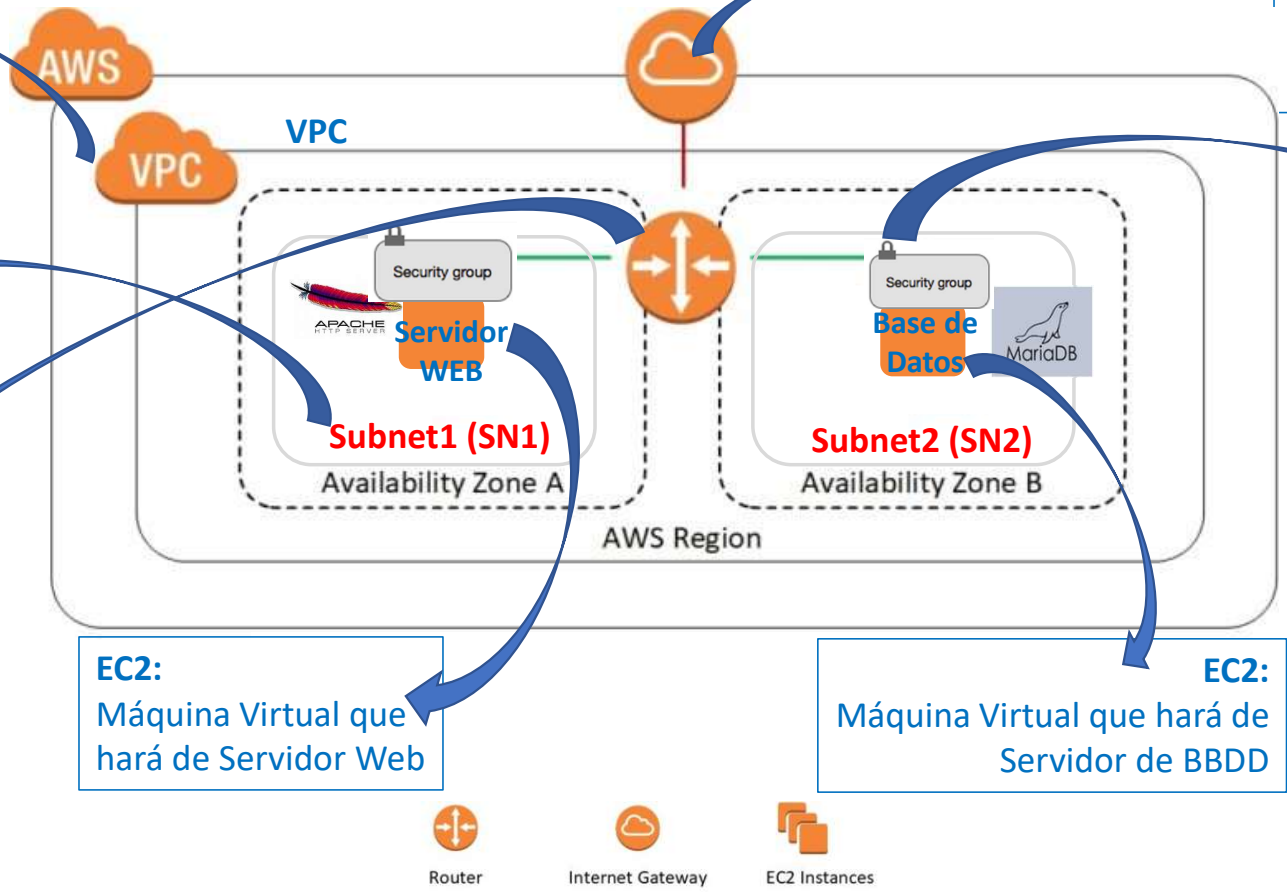
Diseño de Entorno Cloud en AWS basado en máquinas EC2

Arquitectura de la Solución en AWS:

VPC:
(Virtual Private Cloud)
Red en la que tengamos desplegados nuestros recursos

Subredes
SN1 y SN2, pertenecientes ambas a la misma VPC y en las que desplegaremos los recursos necesario.

Router (Tabla de rutas): nos permite establecer las reglas para el encaminamiento de tráfico, bien entre las subredes o bien hacia internet



EC2:
Máquina Virtual que hará de Servidor Web

EC2:
Máquina Virtual que hará de Servidor de BBDD

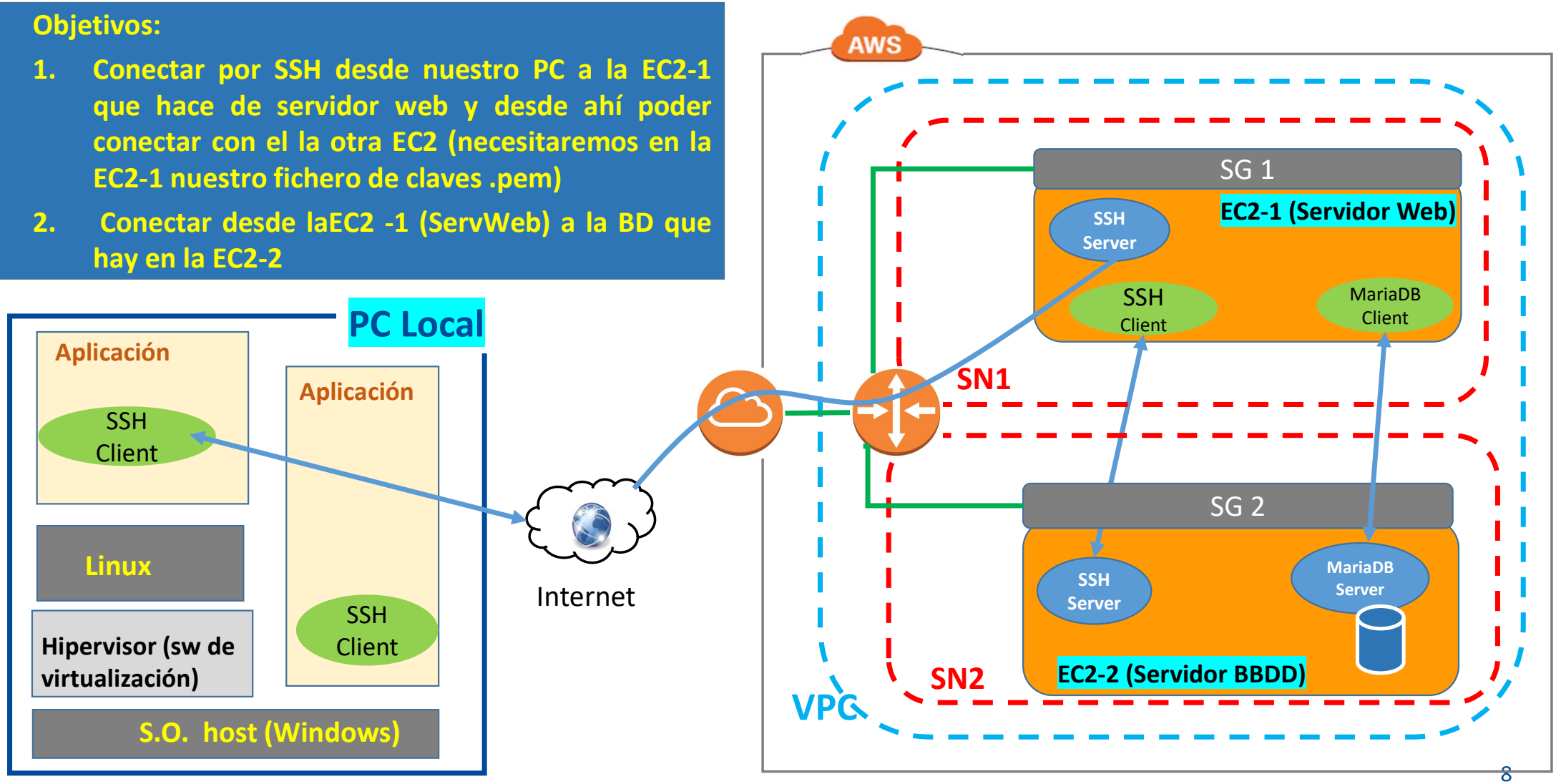
Internet Gateway:
Elemento frontera de la VPC que nos habilita el acceso "a" y "desde" internet a nuestras instancias.

Un **grupo de seguridad** funciona como un firewall virtual que controla el tráfico de una o varias instancias (habilita tráfico mediante la apertura de los puertos correspondientes)

Servidor MariaDB y Clientes: Instalación en la nube

Objetivos:

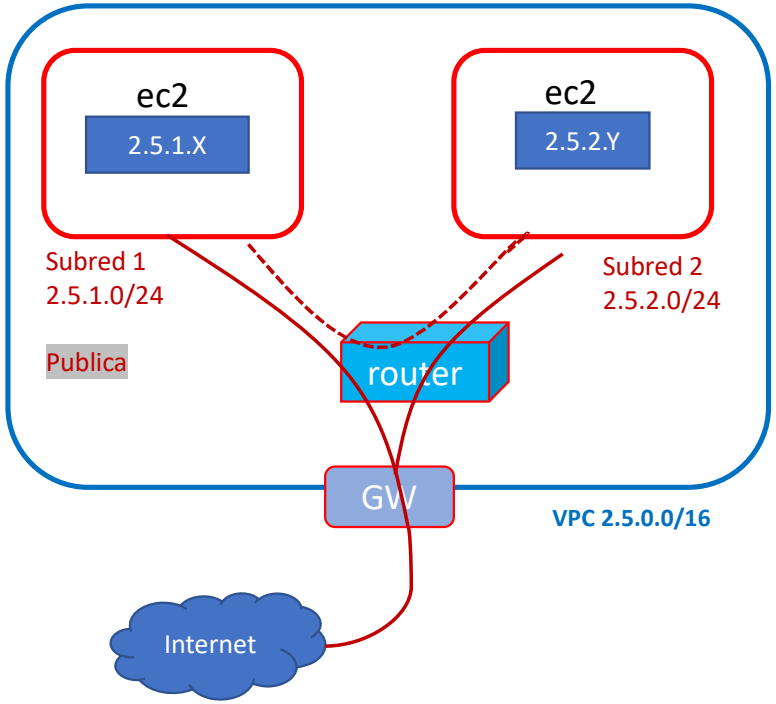
1. Conectar por SSH desde nuestro PC a la EC2-1 que hace de servidor web y desde ahí poder conectar con el la otra EC2 (necesitaremos en la EC2-1 nuestro fichero de claves .pem)
2. Conectar desde laEC2 -1 (ServWeb) a la BD que hay en la EC2-2



Diseño de Entorno Cloud en AWS basado en máquinas EC2

Datos relevantes para crear la solución

- La ***VPC*** tiene que tener el rango de direcciones**: 2.5.0.0/16**. Se deben habilitar los servicios de nombre de dominio.
- Debe constar de dos subnetes, cada una de ellas en una zona de disponibilidad diferente para garantizar la alta disponibilidad. **Los rangos de direcciones de cada subred (subnets) deben ser: SN1=2.5.1.0/24, SN2=2.5.2.0/24, SN3=2.5.3.0/24**
- La VPN debe contener un ***Internet Gateway*** para salir a Internet y una única ***Tabla de Rutas*** que permita la conexión con Internet.
- En la **SN1 se debe crear una instancia EC2, en la que debemos instalar el servidor WEB Apache**, el firewall (o SG) para este entorno debe permitir **sólo** los **servicios de SSH y HTTP/HTTPS**
- **En la SN2 se creará otra EC2 en la que debemos instalar un Gestor de BD MariaDB, en este caso su firewall (o SG) debe permitir sólo los servicios de SSH y MySQL (puerto 3306).**
- Sólo la máquina que hará de servidor WEB tendrá direccionamiento público IPv4 para que pueda ser accesible desde Internet.



***Nota**: En el Escenario 1 no usaremos SN3*

Diseño Entorno Cloud en AWS basado en EC2

Se pide:

1. Definir la VPC y todos los elementos asociados (Subnets, Internet Gateway, Security Groups y Tabla de Rutas) con los parámetros indicados.
2. Definir y arrancar las dos instancias EC2. En la instancia de la SN1 instalar (o asegurar que está instalado) el servidor Apache y en la instancia de la SN2 instalar el Gestor de BD MariaDB (o asegurar que está instalado).
3. Crear en la primera EC2 una página HTML sencilla que contenga en alguna de ella el nombre del alumno (Se debe hacer en el directorio raíz de documentos de apache: /var/www/html).
4. Acceder desde un navegar externo y presentar la página WEB de muestra con el nombre del alumno para asegurar que Apache funciona.
5. Crear en la EC2 una BD de ejemplo (**Practica_Escenario1**) que contenga una tabla que se llame **Alumnos**. La tabla debe tener los siguientes campos **ID_alumno**, **Nombre** y **Apellidos**. Rellenar esta tabla con el nombre y apellidos del alumno que ha realizado la práctica.
6. Modificar la configuración del servidor MariaDB para que se pueda acceder desde el exterior y crear un usuario que tenga acceso externo. Esto nos permitirá conectar desde la EC2 en la que se ha instalado el servidor Web a la BD instalada en la otra EC2.

Diseño de Entorno Cloud en AWS basado en máquinas EC2

Guía de los pasos a seguir en la consola de AWS:

1. **Crear la VPC eligiendo la opción de "Solo la VPC"**. Añadir un nombre y el rango de direcciones que se defina (ver imagen de ejemplo). Campo CIDR IPv4, entrada manual .
2. Al crear el VPC se crea sin estar activo el servicio de nombres de domino. Se puede activar a posteriori en las opciones de la VPC.
3. **Crear las tres subredes con sus rangos de direcciones y asociarlas a la anterior VPC**. Las subredes las crearemos en zonas de disponibilidad distintas. Si las creamos en diferentes zonas de disponibilidad nos aseguramos que ante caídas no perdamos el servicio. En nuestro ejemplo eso no es muy relevante pero en la vida real sí lo es.
4. **Creamos una tabla de rutas (router), asociamos las subredes creadas en el punto anterior y veremos que por defecto se crea la regla de encaminamiento de tráfico local.**
5. **Creamos el Internet GW para la salida a Internet y se lo asociamos a la VPC creada**.
6. **Añadimos a la tabla de rutas creada anteriormente, la ruta para redirigir el tráfico "que va a cualquier destino" hacia el GW de Internet**
7. **Haremos la Subred 1 de acceso público**. Para ello permitiremos la asignación de IPv4 públicas a dicha subred. Seleccionamos la subred → "Acciones" → Editar la configuración de la subred. Al hacer esto, nos permitirá crear instancias EC2 en esta subred con la opción habilitada para asignar IPs públicas ("enable"). Si la subred no tiene acceso público entonces al crear la máquina aparecerá por defecto "disable"

Diseño Entorno Cloud en AWS basado en EC2

Guía de los pasos a seguir en la consola de AWS:

8. Crearemos una máquina (EC2) en SN1 y otra en SN2. Una EC2 será la que haga de Servidor Web y la otra de Servidor de BBDD
9. Crearemos un Security Group diferente para cada EC2. Los podemos crear al tiempo de crear las instancias.
10. En el SG de la instancia EC2 de la subred 1 (es en la que ejecutaremos el servidor web) abriremos el tráfico SSH y HTTP/HTTPS**. Nos referiremos a él como SG1.**
11. En el otro ordenador (EC2 de la subred 2), es donde vamos a tener instalado el servidor de MariaDB, por tanto abriremos en su SG el puerto 3306 de MariaDB para el tráfico MySQL pero solo para el tráfico procedente del security group creado para nuestro ordenador 1. Abrimos el SSH también para conectarnos a la máquina. **Nos referiremos a este SG como SG2.**
12. *Una vez creada toda la infraestructura de red y las máquinas:*
    - *Asegurarnos que en la EC2 de la subred 1 el servidor Web Apache y PHP están instalados y funcionando. Hacer las pruebas correspondientes para asegurar que ambos funcionan. Instalar también el cliente de MariaDB si no está instalado. (que todo este software esté o no instalado depende de la versión de AMI que elijamos al crear la EC2. En el caso de utilizar el tipo cloud9ubuntu ya viene todo instalado por defecto).*
    - *Asegurarnos que en la EC2 de la subred 2 el servidor de MariaDB está instalado y funciona (conectarnos desde la propia máquina). Configurar MariaDB para que soporte acceso remoto (conexión desde otras máquinas) y crear un usuario al que se le otorgue el permiso de conectar remotamente.*

Instalación de SW en las EC2s

OJO!!!! Estos pasos sólo deben hacerse si el sw necesario no está previamente instalado

En la EC2-1, es decir en la máquina que hace de Serv.Web, necesitamos el siguiente sw: Apache (como servidor web), PHP y cliente MariaDB. Los pasos para instalarlo serían:

1. Instalar el servidor web
$ sudo apt-get -y install apache2
$ ps -ef | grep apache

2. Comprobamos el estado del servidor:
$ sudo systemctl status apache2

3. Instalamos PHP, el módulo PHP para apache2 (servidor web) y para mysql:
$ sudo apt-get install php libapache2-mod-php php-mysql

4. Rearrancamos el servidor Web:
$ sudo systemctl restart apache2

5 . Instalar cliente de mariadb
$ sudo apte-get install mariadb-client

En AWS Academy utilizando AMIs del entorno ***clou9ubuntu*** para crear las máquinas virtuales ya viene instalado el servidor web Apache y PHP por lo que no son necesarios los pasos de la instalación de paquetes.

Tampoco es necesario instalar el cliente de MariaDB para conectarnos a la BD que estará en la otra EC2.

Instalación de SW en las EC2s

OJO!!!! Estos pasos sólo deben hacerse si el sw necesario no está previamente instalado

En la EC2-2, es decir en la que máquina que hará de Servidor de BD necesitamos el siguiente sw: Cliente y Servidor de María DB. Necesitamos ejecutar el siguiente comando:

1 . Instalar cliente y servidor de mariadb

$ sudo apte-get install mariadb-client mariadb-server

En AWS Academy utilizando AMIs del entorno ***clou9ubuntu*** para crear las máquinas ya viene preinstalado el sw de mysql/mariadb

Creación de una BD en la EC2-2

Una vez que hayamos asegurado que en la EC2-2 (la que hace de Servidor de BD) tenemos MariaDB instalado crearemos la Base de Datos que nos indica el enunciado:.

En la EC2-2 (la que hace de servidor de BD) hacer:

```
$ sudo mysql –u root

MySQL> CREATE DATABASE Practica_Escenario1;

MySQL> use Practica_Escenario1;

MySQL> CREATE TABLE Alumnos (id_alumno INT, nombre_apellidos VARCHAR(50));

MySQL> INSERT INTO Alumnos (id_alumno, nombre_apellidos) values (1, 'Carlos Lopez Fuentes'), (2, 'Lucia Navas Garcia');
```

Configuración del servidor de MariaDB en la EC2-2 para que nos deje conectar desde la EC2-1

A. Configurar el servidor mariadb para que nos deje conectarnos desde cualquier host remoto, para ello editar el fichero de configuración:

$ sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf

Buscar la línea en la que se da valor a la variable bind-address y:

Y donde pone:

bind-address = 127.0.0.1

Comentar esa línea y sustituirla por:

bind-address = 0.0.0.0

Salvar el fichero y rearrancar el servidor:

$ sudo systemctl restart mysql

Comprobar su estado:

$ sudo systemctl status mysql

Por defecto, la instalación de MariaDB no permite conexiones externas desde otras máquinas, es decir, por seguridad sólo admite conexiones desde la propia máquina local (localhost).

Por tanto, para conectarnos desde la primera máquina a la BD que hay en la segunda máquina **es necesario modificar la configuración de MariaDB, para que ademita conexiones remotas y no solo desde el localhost**.

Como en este caso nos queremos conectar a MariaDB desde la EC2 que hace de servidor web **es necesario realizar los pasos A y B descritos.**

Configuración del servidor de MariaDB para crear un usuario con el que nos podamos conectar a la BD

B. Crear un usuario en la BD y darle permisos usuario para conectar remotamente…

Conectarnos a MariaDB como root y ejecutar **una de las dos opciones siguientes**:

B.1 → Permite acceso solo desde máquinas que pertenecen a la subred 2.5.1.%:

```
>CREATE USER 'admin_lab'@ '2.5.1.%' IDENTIFIED BY 'admin_lab123';
> GRANT ALL PRIVILEGES ON  *.*  TO   'admin_lab'@'2.5.1.%';
>FLUSH PRIVILEGES;
```

B.2 → Permite acceso desde cualquier IP:

```
>CREATE USER 'admin_lab'@%' IDENTIFIED BY 'admin_lab123';
>GRANT ALL PRIVILEGES ON  *.*  TO   'admin_lab'@'%';
>FLUSH PRIVILEGES;
```

Probemos…
Una vez hecho todo, debemos probar que somos capaces de conectarnos desde la EC2-1 a la BD que está en la EC2-2

Con esto ya nos podemos conectar desde nuestro EC2-1 (la que hace de servidor WEB) a la BD que tenemos en la otra EC2-2:

Por ejemplo, si la dirección IP de la EC2-2 fuera 5.2.2.185 , el comando para conectar desde la EC2-1 seria:

```
$ sudo  mysql –h 5.2.2.185 –u admin_lab -p
```

Comprobar que efectivamente podemos acceder a la BD creada anteriormente llamada “Practica_Escenario1” y que podemos acceder a los datos de la tabla “Alumnos”

ESCENARIO 2

BASADO EN UNA INFRAESTRUCTURA DE RED, UNA MÁQUINA EC2 Y UNA RDS

Vamos a practicar en AWS: **ESCENARIO 2**

Debemos diseñar un entorno distribuido basado en Cloud Computing para dar servicio a una empresa. El sistema debe constar principalmente de dos entornos montados sobre una infraestructura de red propia, por tanto se pide:

1. **Crear la infraestructura de red (VPS, subredes, routers, etc) que se describe en las siguientes slides.**
2. **Crear un ordenador (EC2) que hará de servidor de páginas WEBs** (SW Apache).
3. **Crear un RDS en el que instanciaremos un servidor MariaDB.**

La solución debe diseñarse en Amazon Web Services (AWS) y la arquitectura es la que se muestra a continuación:

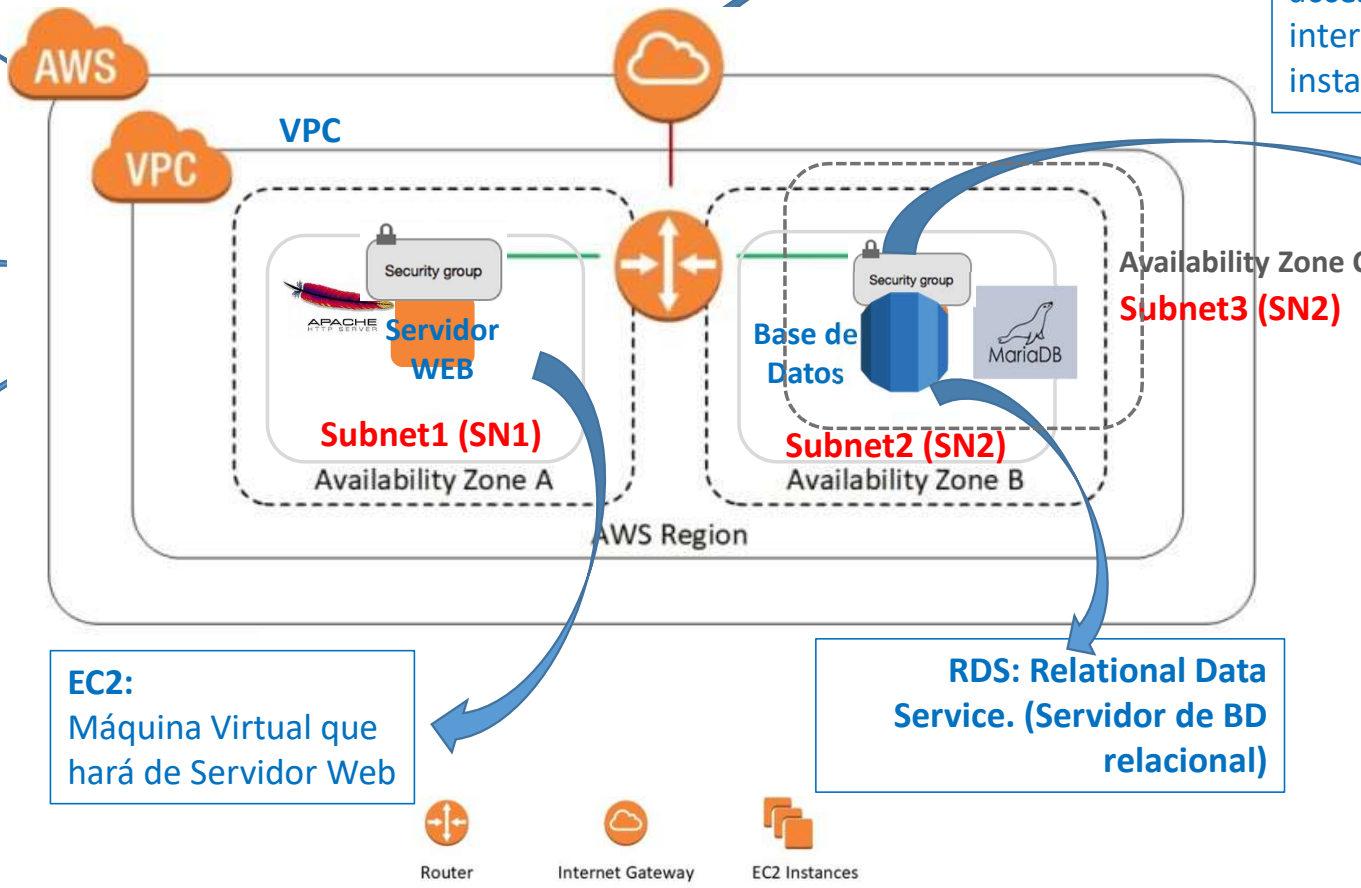
Diseño de Entorno Cloud en AWS basado en EC2 y RDS

Arquitectura de la Solución en AWS:

VPC:
(Virtual Private Cloud)
Red en la que tengamos desplegados nuestros recursos

Subredes
SN1 y SN2, pertenecientes ambas a la misma VPC y en las que desplegaremos los recursos necesario.

Router (Tabla de rutas): nos permite establecer las reglas para el encaminamiento de tráfico, bien entre las subredes o bien hacia internet



EC2:
Máquina Virtual que hará de Servidor Web

RDS: Relational Data Service. (Servidor de BD relacional)

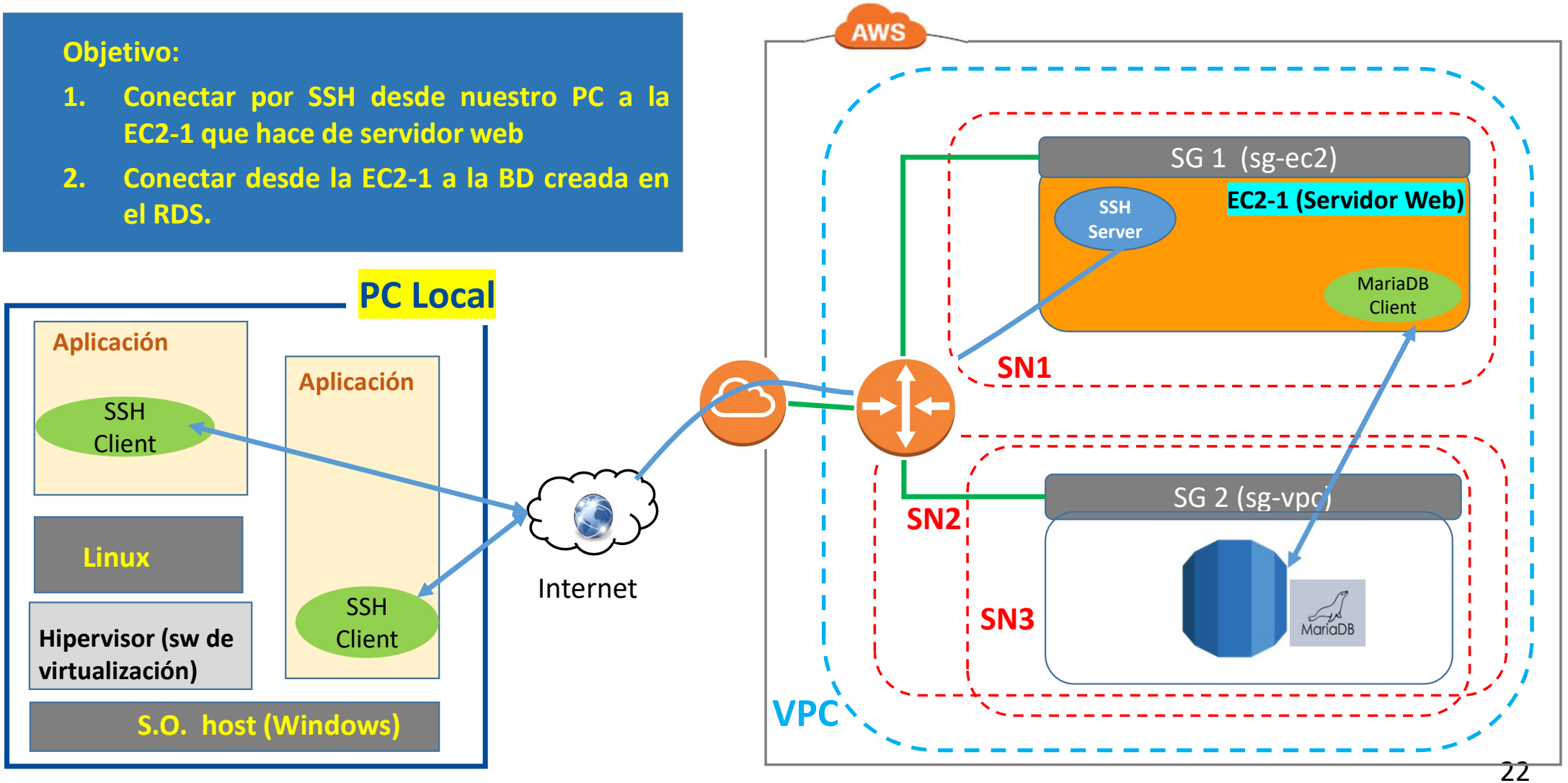
Internet Gateway:
Elemento frontera de la VPC que nos habilita el acceso "a" y "desde" internet a nuestras instancias.

Un **grupo de seguridad** funciona como un firewall virtual que controla el tráfico de una o varias instancias (habilita tráfico mediante la apertura de los puertos correspondientes)

Servidor MariaDB y Clientes: Instalación en la nube

Objetivo:

1. **Conectar por SSH desde nuestro PC a la EC2-1 que hace de servidor web**
2. **Conectar desde la EC2-1 a la BD creada en el RDS.**



Diseño de Entorno Cloud en AWS basado en EC2 y RDS

Pasos a seguir en la consola de AWS:

Partimos de la infraestructura de red creada en el ESCENARIO 1, es decir:

- **Utilizaremos la misma VPC, las subredes (en este caso las 3 subredes SN1, SN2 y SN3), la tabla de rutas, el Internet Gatewaty y los SGs.**
- **Utilizaremos la misma EC2 que hacía de servidor Web.**

Los pasos que debemos dar para montar este segundo escenario son:

1. **Crear un "grupo de subredes de Base de Datos" utilizando las subredes SN2 y SN3.**
2. **Instanciar un RDS privado** en la nuestra VPC indicando que utilice "el grupo de subredes de BBDD" creado en el apartado anterior.
3. Utilizar usuario por defecto: **admin y password la que cada uno quiera.**
4. Utilizaremos para el RDS el Security Group que ya estábamos utilizando en el ESCENARIO 1 para la EC2 que hacía de Servidor de BD, **es decir usaremos el SG2**. Debe tener admitir tráfico SSH desde cualquier origen y tráfico MySQL desde el SG1.

No tenemos que instalar ningún software ni hacer ninguna configuración adicional para probar este escenario.

Creación de la BD en la RDS

Una vez que hayamos creado la RDS nos conectamos desde la EC2-1 (la que hace de servidor web) para crear la BD que nos pide el enunciado. Debemos conocer la cadena de conexión del RDS que aparece en el apartado de los "detalles", el usuario para conectar y la password (los utilizados al crear la RDS).

En la EC2-1 hacer:

$ sudo mysql –h "cadena de conexión de la RDS creada" –u admin -p

```
MySQL> CREATE DATABASE Practica_Escenario2;

MySQL> use Practica_Escenario2;

MySQL> CREATE TABLE Alumnos (id_alumno INT, nombre_apellidos VARCHAR(50));

MySQL> INSERT INTO Alumnos (id_alumno, nombre_apellidos) values (1, 'Carlos Lopez Fuentes'), (2, 'Lucia Navas Garcia');
```

Terminaremos este escenario conectando a la BD desde una página web dinámica hecha en PHP que encontrarás en BB

Descargar de BB el fichero Conexión_MySQL.php y copiarlo a la EC2-1 mediante el comando scp:

En tu máquina Ubuntu local debes ejecutar:

```
$ scp –i “labsuser.pem” Conexión_MySQL.php  ubuntu@IPpublica_o_DNS_publico_de_EC2-1”:/home/ubuntu/.
```

En la EC2-1 debes ejecutar:

```
$sudo cp Conexión_MySQL.php  /var/www/html/.
```

```
$cd /var/www/html/
```

Asegúrate que el fichero Conexión_MySQL.php tiene permisos “644”, si no los tiene ejecuta:

```
$sudo chmod 644 Conexión_MySQL.php
```

Modificaremos el fichero entregado como plantilla para adaptarlo a nuestro RDS (BD)

Edita el fichero Conexión_MySQL.php y modifica los parámetros específicos de tu RDS para que la página web (fichero .php) se pueda conectar a tu BD:

En la EC2-1 debes ejecutar:

```
$sudo nano Conexión_MySQL.php
```

Aparecerá el contenido que se muestra a continuación. Debes hacer los cambios indicados dentro de ese fichero y salir salvando cuando los hayas termianado

Contenido del fichero Conexión_MySQL.php

elenagg@elenagg-LRSO: ~/AWS_21_22/ConexPHPMySQL

```php
<?php
        // Ejemplo de conexión a base de datos MySQL con PHP.
        //


        // Datos de la base de datos. Sustituir los valores entre "" por los correspondientes de la BBDD sobre la que
        // se quiere yrabajar
        $usuario = "usuario";
        $password = "password";
        $servidor = "endpoint";
        $basededatos = "nombre_BD";
        $tabla = "nombre_tabla";

        // creación de la conexión a la base de datos con mysql_connect()
        $conexion = mysqli_connect( $servidor, $usuario, $password ) or die ("No se ha podido conectar al servidor de Base de datos");

        // Selección del a base de datos a utilizar
        $db = mysqli_select_db( $conexion, $basededatos ) or die ( "Upps! Pues va a ser que no se ha podido conectar a la base de datos" );

        // establecer y realizar consulta. guardamos en variable.
        $consulta = "SELECT * FROM $tabla";
        $resultado = mysqli_query( $conexion, $consulta ) or die ( "Algo ha ido mal en la consulta a la base de datos");

        // Motrar el resultado de los registro de la base de datos. Aqui podemos poner los valores que queramos a Campo1, Campo2, etc.
        // pero es una práctica ponerle nombres similares a los de la base de datos
        // Encabezado de la tabla
        echo "<table borde='2'>";
        echo "<tr>";
        echo "<th>campo1</th>";
        echo "<th>campo2</th>";
        echo "<th>Campor3</th>";
        echo "<th>Campo4</th>"
        echo "</tr>";
```

Debes modificar estas variables con los valores adecuados para tu RDS y tu BD

Debes adaptar estos rótulos según el número de columnas que tenga tu tabla y como quieres que aparezcan en los resultados que muetre la página web

Contenido del fichero Conexión_MySQL.php

elenagg@elenagg-LRSO: ~/AWS_21_22/ConexPHPMySQL

```
$tabla = "nombre_tabla";

// creación de la conexión a la base de datos con mysql_connect()
$conexion = mysqli_connect( $servidor, $usuario, $password ) or die ("No se ha podido conectar al servidor de Base de datos");

// Selección del a base de datos a utilizar
$db = mysqli_select_db( $conexion, $basededatos ) or die ( "Upps! Pues va a ser que no se ha podido conectar a la base de datos" );

// establecer y realizar consulta. guardamos en variable.
$consulta = "SELECT * FROM $tabla";
$resultado = mysqli_query( $conexion, $consulta ) or die ( "Algo ha ido mal en la consulta a la base de datos");

// Motrar el resultado de los registro de la base de datos. Aqui podemos poner los valores que queramos a Campo1, Campo2, etc.
// pero es una práctica ponerle nombres similares a los de la base de datos
// Encabezado de la tabla
echo "<table borde='2'>";
echo "<tr>";
echo "<th>campo1</th>";
echo "<th>campo2</th>";
echo "<th>Campor3</th>";
echo "<th>Campo4</th>"
echo "</tr>";

// Bucle while que recorre cada registro y muestra cada campo en la tabla. Los valores de Campo1, Campo2, etc. deben coincidir de forma exacta
// con los campos de la BBDD
while ($columna = mysqli_fetch_array( $resultado ))
{
    echo "<tr>";

    echo "<td>" . $columna['Campo1'] . "</td><td>" . $columna['Campo2'] . "</td><td>" . $columna['Campo3'] . "</td><td>". $columna['Campo4'] . "</td>";

    echo "</tr>";
}

echo "</table>"; // Fin de la tabla

// cerrar conexión de base de datos
```

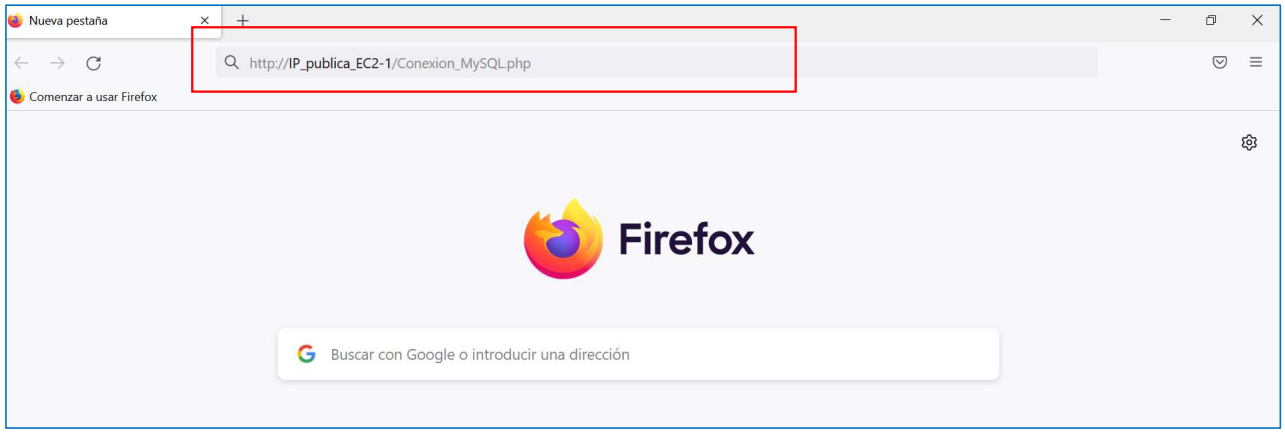
Luego debes adaptar esta línea para que sólo aparezcan los campos de la tabla en la que vas a consultar los datos. Recuerda que aquí el orden y el nombre de estos campos deben coincidir exactamente con las columnas de la tabla

43,5-12 84%

Probemos…..

En un navegador pondremos la url correspondiente a la dirección de la IP publica de la EC2-1 (la que hace de Servidor Web) y a continuación el nombre del fichero Conexión_MySQL.php (página web dinámica que se conectará a la BD del RDS y nos mostrará el resultado.

Es decir:



Deberemos obtener el resultado de la información de la taba consultada de nuestra BD, según los cambios que hemos hecho en el fichero .php

ANEXO 1

Repaso:
Direccionamiento IP

Formato Numeración de nodos en redes IPv4

- Todos los nodos de la red IP tienen asociado una dirección IP.
- Una dirección IP (versión 4) es un número de 32 bits, que se representa con **4 números decimales de 8 bits separados** por un punto.
- El valor más alto que se puede obtener con 8 bits es 255.
- El mayor número IPv4 es 255.255.255.255 y el más pequeño es 0.0.0.0

Direcciones IP públicas y Privadas

- Debido al alto número de nodos IP y la necesidad de segmentación de subredes IP, se creó el concepto de **redes IP públicas y privadas**.
- Los nodos que están en la red pública (WAN) tienen una dirección IP pública, visible y única en Internet.
- Los nodos de las redes de área local (LAN) normalmente tienen direcciones privadas, que no son visibles ni accesibles desde la red publica (el router es el único elemento público).
- Las direcciones privadas se pueden repetir infinitas veces.
- Para la conversión de red privada a pública se utiliza, los routers usan un mecanismo llamado NAT (Network Address Translation)

Numeración de nodos en redes IPv4

- El número máximo de nodos de la red IPv4 pública es:
  $2^8 x 2^8 x 2^8 x 2^8 = 256x256x256x256 = 4.294.967.296.$
- A pesar de su elevado número, se están agotando las direcciones asignadas por la ICANN a países como China e India. Por ello surgieron las redes privadas e IPv6.
- Un nodo de red privada tiene una dirección IPv4, y una máscara de red y también la dirección de servidores DNS.
- La máscara de red indica los nodos de la red o subred IP privada con los que tiene comunicación. Con los nodos fuera de la mascara de red no hay comunicación. Es decir dentro de una red IP física podemos hacer subredes aisladas

Máscara de Red

- La máscara de red permite distinguir dentro de la dirección IP, los bits que identifican a la red y los bits que identifican al host.

- La máscara se forma poniendo a 1 los bits que identifican la red y 0 en los bits que identifican al host. De esta forma, la siguiente máscara 255.255.255.0 nos indicaría que en la dirección IP asociada, los 3 primeros octetos identificarían la red a la que pertenece el dispositivo y el último octeto representa la dirección de dicho dispositivo o host.

- La máscara también puede ser representada de la siguiente forma **10.2.1.2/24** donde **el /24 indica que los 24 bits más significativos de la IP están destinados a identificar la red** o lo que es lo mismo, los primeros 24 bits a 1, es decir **/24 = 255.255.255.0**.

- Análogamente (/16 = 255.255.0.0).

Ejemplo de Numeración de nodos en redes IPv4

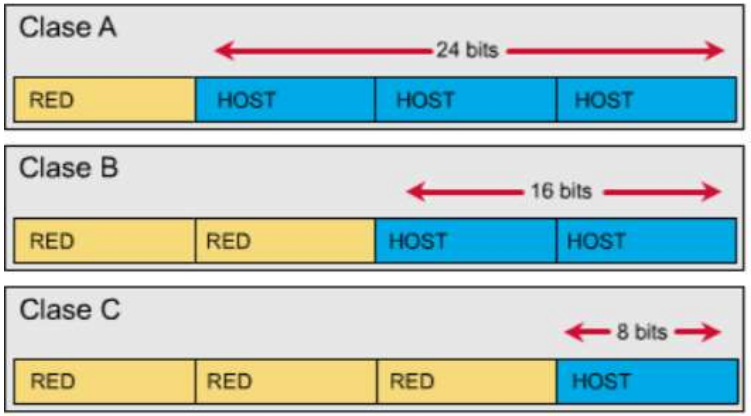
- Imaginemos un router con una única dirección pública (ej. 83.39.97.73) y una dirección privada **192.168.1.1/24**, o lo que es lo mismo una IP **192.168.1.1** con una máscara de red **255.255.255.0**.
- Su máscara de red nos indica que:
  - **Los primeros 24 bits** (en rojo), **192.168.1.X** llamados *"dirección de red"* corresponden a la dirección de la subred a la que pertenece el router. Serían el inicio de la dirección IP de todos los nodos de la subred.
  - **Los últimos 8 bits de dirección IP (en azul):** 192.168.1.**X,** representan la dirección concreta de los nodos de la subred red. Dicho valor puede variar entre 0 hasta 255 (valor máx. de 8 bits).
  - En la subred 192.168.1.**X**/24, el rango de direcciones de los nodos de la subred privada, iría de 192.168.1.**0** a 192.168.1.**255**, aunque precisamente el valor 0 (indicador de la propia subred) y el valor 255 (broadcast) están reservados. Es decir podría haber hasta 256 – 2 = 254 nodos en la subred.

Clases de Redes según su direccionamiento

Existe una clasificación rígida y de uso limitado, de redes IP, en función del número de bits , bloques de direcciones por octetos, que implicaban prefijos «naturales» de 8, 16 y 24 bits reservados al nombre de subred y a los nodos de red.

Son las redes de clase A, B, C, D, E, las redes tipo D y E están es desuso.

Esta forma de diseñar las redes, con máscara única o máscara fija, basada en asignar bloques de direcciones con los límites de los octetos «naturales» de 8, 16 y 24 bits, no proporciona mucha flexibilidad.



Direcciones IP reservadas

No todas las posibles direcciones IPv4 se pueden usar en las redes IP públicas, hay conjuntos que se usan para propósitos especiales; redes privadas, loopback, broadcast, etc.

En la tabla se presentan los principales bloques.

En total hay unos 592 M de direcciones reservadas

https://es.wikipedia.org/wiki/Anexo:Direcciones_IP_reservadas

Bloque	Rango	Núm. direcciones	Uso	Descripción
0.0.0.0/8	0.0.0.0 – 0.255.255.255	16,7 M	Red actual	Dirección de origen
10.0.0.0/8	10.0.0.0 – 10.255.255.255	16,7 M	Red privada (nodos)	Direcciones IP de nodos
100.64.0.0/10	100.64.0.0–100.127.255.255	4,19 M	Red privada ISP – CLI	Proveedor de servicios
192.168.0.0/16	192.168.0.0–192.168.255.255	65.536	Red privada (nodos)	Direcciones de IP nodos
198.18.0.0/15	198.18.0.0–198.19.255.255	131.072	Red privada	Puentes entre subredes
127.0.0.0/8	127.0.0.0–127.255.255.255	16,7 M	Dir. de loopback	Uso por el nodo (host)
224.0.0.0/4	224.0.0.0–239.255.255.255	268 M	IP Multicast	(antigua clase D)
240.0.0.0/4	240.0.0.0–255.255.255.255	268 M	Sin uso	(antigua clase E)
	255.255.255.255	1	Broadcast	Subred

Máscara de Subred de Longitud Variable - VLSM

- *En el caso del diseño de las LAN, la clasificación con máscara flexible (class-less) da mucha más flexibilidad en la creación de subredes* [(1)].

- CIDR (***Classless Inter-Domain Routing***) se introdujo en 1993 y reemplazó la sintaxis previa para nombrar las clases de redes usando ahora la técnica **VLSM** (*variable length subnet mask*, en español «**Máscara de subred de longitud variable**»), para hacer posible la asignación de prefijos de longitud arbitraria.

- Esta técnica permite diseñar un esquema de direccionamiento usando la máscara en función de la cantidad de hosts que se necesitan, es decir, la cantidad de hosts determina la longitud de la máscara o longitud del prefijo de red. Es una mejora respecto al uso de máscara fija que sólo permitía elegir una máscara de subred o longitud del prefijo de red, lo cual implicaba que la red más grande mandaba y que las redes más pequeñas estaban obligadas a ser ineficientes porque tendrían obligatoriamente una capacidad sin uso que, probablemente, nunca se iba a necesitar y nunca se podría recuperar porque el esquema sólo admite una sola máscara.

- Con VLSM, dependiendo de: numero de usuarios de la subred, numero de subredes necesario o mascara de red se puede calcular la configuración de red manualmente o incluso mediante aplicaciones web (http://www.subnet-calculator.com/cidr.php)

(1) https://es.wikipedia.org/wiki/Classless_Inter-Domain_Routing

Numeración de nodos en redes IPv4

- Otras mascaras de red IPv4, numero de direcciones IP

Composición (Red y Hosts)

Sufijo CIDR	Máscara (en binario)	Máscara	Número de bits para host (n)	Número de direcciones (2^n)
/32	255.255.11111111.11111111	255.255.255.255	0	1
/31	255.255.11111111.11111110	255.255.255.254	1	2
/30	255.255.11111111.11111100	255.255.255.252	2	4
/29	255.255.11111111.11111000	255.255.255.248	3	8
/28	255.255.11111111.11110000	255.255.255.240	4	16
/27	255.255.11111111.11100000	255.255.255.224	5	32
/26	255.255.11111111.11000000	255.255.255.192	6	64
/25	255.255.11111111.10000000	255.255.255.128	7	128
/24	255.255.11111111.00000000	255.255.255.0	8	256
/23	255.255.11111110.00000000	255.255.254.0	9	512

Direcciones MAC

- Cada dispositivo tiene una dirección MAC (Medium Access Control), que es única. A cada fabricante de dispositivos red se le asigna un rango de direcciones MAC para sus productos.
- Las direcciones MAC se graban de forman de forma inalterable en cada dispositivo que se conecta a la red IP. La dirección MAC también se conoce como dirección física y es independiente del protocolo que se utilice.
- Hay elementos de red como routers y switches que conocen la relación univoca entre las direcciones MAC y las direcciones IP de una subred, mediante el protocolo ARP.
- Los routers, que tienen implementado y activado el protocolo DHCP (Dynamic Host Configuration Protocol), asignan direcciones IP basados en la dirección MAC de los nodos de red.

U-tad

Calle Playa de Liencres, 2 bis
(entrada por calle Rozabella)
Parque Europa Empresarial
Edificio Madrid
28290 Las Rozas, Madrid

900 373 379 info@u-tad.com

SOLICITA MÁS INFORMACIÓN

CENTRO ADSCRITO A:

Universidad Camilo José Cela

PROYECTO COFINANCIADO POR:

GOBIERNO DE ESPAÑA · MINISTERIO DE EDUCACIÓN, CULTURA Y DEPORTE