



**Universidad**  
Internacional  
de Valencia

## Introducción a AWS

**Titulación:**

Máster Universitario en Desarrollo de  
Aplicaciones y Servicios Web

**Curso académico:** 2022-2023

**Alumno/a:** Gómez Olivencia, Rubén

**D.N.I.:** 78910013-A

**Asignatura:** Computación en la nube

# Índice general

<b>1. Introducción</b>	<b>3</b>
<b>2. Base de datos RDS</b>	<b>3</b>
2.1. Mediante CLI	5
<b>3. EC2 y despliegue de aplicación</b>	<b>6</b>
3.1. Creación de la máquina virtual	6
3.1.1. Mediante CLI	7
3.2. Configuración previa	8
3.2.1. Creación del sistema de ficheros	8
3.2.2. Instalación de dependencias	9
3.3. Instalación de la aplicación	10
<b>4. Grupos de seguridad</b>	<b>12</b>
4.1. Mediante CLI	13
<b>5. IP elástica</b>	<b>14</b>
<b>6. Bucket S3</b>	<b>15</b>
6.1. Mediante CLI	16
<b>7. Costes estimados</b>	<b>17</b>
<b>8. Conclusiones</b>	<b>18</b>

# 1. Introducción

A la hora de poner un proyecto en producción es aconsejable conocer las distintas alternativas que podemos utilizar para llevarlo a cabo. Hasta hace no muchos años, lo habitual era realizar el despliegue en un servidor propio situado en algún CPD.

Debido a las dificultades que puede suponer el gestionar unos servidores propios, junto con el no poder realizar el escalado del proyecto de manera sencilla, ha supuesto que la demanda de utilizar sistemas *cloud* crezca, ya que facilitan dicha labor.

A lo largo de este documento se va a realizar una introducción a **AWS** (Amazon Web Services) junto con el despliegue de la aplicación [Flarum](#) en un servidor creado en este *cloud*.

# 2. Base de datos RDS

Para la aplicación se va a necesitar una base de datos, por lo que va a ser lo primero que se va a desplegar.

AWS permite realizar el despliegue de distintos gestores de bases de datos a través de su servicio **RDS** (*Relational Database Service*). En este caso se va a desplegar un RDS de MySQL 5.X con las siguientes características:

- RDS de tipo db.t3.micro
- Almacenamiento 20Gib SSD de uso general (gp2).
- MultiA-Z no.
- Cifrado en reposo habilitado.
- Configurar los backups automáticos para que se ejecuten a las 3:00 UTC con período de retención de 10 días.
- Configurar la ventana de mantenimiento de actualizaciones a las 4:00 UTC.

A continuación parte de la configuración realizada durante la creación de la misma:

Versión  
 MySQL 5.7.40

db.t3.micro  
 2 vCPUs 1 GiB RAM Red: 2085 Mbps

Tipo de almacenamiento [Información](#)  
 SSD de uso general (gp2)  
 Rendimiento de referencia determinado por el tamaño del volumen

Almacenamiento asignado  
 20 GiB

Periodo de copia de seguridad [Información](#)  
 El intervalo de tiempo diario (en UTC) durante el cual RDS realiza copias de seguridad automatizadas.

☒ Elegir una ventana  
☐ Sin preferencia

Hora de inicio Duración  
 03 : 00 UTC 0.5 horas

Periodo de mantenimiento [Información](#)  
 Seleccione el periodo en el que desea que Amazon RDS aplique las modificaciones o el mantenimiento pendientes a la base de datos.

☒ Elegir una ventana  
☐ Sin preferencia

Día de inicio Hora de inicio Duración  
 Lunes 04 : 00 UTC 0.5 horas

A la hora de crear la base de datos se ha elegido crear una contraseña automática, por lo que es importante visualizarla y copiarla, ya que de no ser así, habría que volver a crear la base de datos.

Hay que tener especial cuidado con el acceso al puerto 3306, por lo que se ha creado un grupo de seguridad propio que limitará únicamente el acceso a la IP pública de mi conexión y desde la IP privada de la instancia que se comentará cómo se ha creado posteriormente.

ID de la regla del g...	Versión de IP	Tipo	Protocolo	Intervalo de puertos	Origen
sgr-0572d6dfd71fc81f3	IPv4	MySQL/Aurora	TCP	3306	172.31.81.57/32
sgr-01e456771bf86f151	IPv4	MySQL/Aurora	TCP	3306	47.61.134.184/32

No se ha añadido la regla del tráfico saliente, ya que mantenemos la regla creada por defecto que habilita todo el tráfico que sale de la base de datos. Para mayor seguridad, podríamos hacer lo mismo, pero no es necesario.

De esta manera, a partir de ahora se podrá acceder al puerto 3306 con los credenciales que nos ha creado el asistente y podremos realizar la conexión utilizando el host **database-1.c92nioexause.us-east-1.rds.amazonaws.com**

Para asegurar que tenemos acceso, se utiliza el programa [Dbeaver](#) para comprobar que se puede realizar la conexión y de paso crear la base de datos “flarum” para instalar posteriormente la aplicación:



Tras crear la base de datos, se puede pasar a realizar la creación de la máquina virtual.

## 2.1. Mediante CLI

Si queremos crear la instancia de RDS mediante el interfaz de línea de comandos se puede realizar a través del siguiente comando:

### </> Crear instancia RDS mediante CLI

```
ddd_v1_w_Hox_1818884@runweb70100:~$ aws --region us-east-1 \
  rds create-db-instance \
    --db-name mysqlcli \
    --db-instance-identifier rds-cli \
    --allocated-storage 20 \
    --db-instance-class db.t3.micro \
    --engine mysql \
    --engine-version 5.7 \
    --master-username admin \
    --master-user-password p4isQW73QW9130aS \
    --storage-encrypted \
    --no-multi-az \
    --backup-retention-period 10 \
```

```
--preferred-backup-window 03:00-03:30 \  
--vpc-security-group-ids sg-03172af1cf7d57f66
```

## 3. EC2 y despliegue de aplicación

**EC2**, o *Elastic Compute Cloud*, es la capa de negocio de Amazon que nos permite crear máquinas virtuales dentro de su sistema *cloud*.

### 3.1. Creación de la máquina virtual

Para realizar el despliegue se va a necesitar una instancia EC2 con las siguientes características:

- Tipo: **t2.micro**
- Arquitectura 64bits (x86)
- Sistema Operativo Amazon Linux 2
- Volumen raíz 10GiB SSD (gp2)
- Volumen adicional EBS de 5 GiB

Para crear la instancia debemos ir al interfaz web, buscar la sección **EC2** y darle al botón **Lanzar instancias** que nos abrirá el asistente para seleccionar las opciones descritas.

Durante el asistente se ha configurado el grupo de seguridad para permitir el acceso por SSH, al puerto 80 y al 443 desde mi IP personal. En apartados posteriores se detallarán más opciones acerca de los grupos de seguridad.

### Firewall (grupos de seguridad) Información

Un grupo de seguridad es un conjunto de reglas de firewall que controlan el tráfico de la instancia. Agregue reglas para que el tráfico específico llegue a la instancia.

☒ Crear grupo de seguridad
 ☐ Seleccionar un grupo de seguridad existente

Crearemos un nuevo grupo de seguridad denominado "launch-wizard-1" con las siguientes reglas:

☒ Permitir el tráfico de SSH desde Internet  
Ayuda a establecer conexión con la instancia

☒ Permitir el tráfico de HTTPS desde Internet  
Para configurar un punto de enlace, por ejemplo, al crear un servidor web

☒ Permitir el tráfico de HTTP desde Internet

Número de instancias Información

1

Imagen de software (AMI)

Amazon Linux 2 Kernel 5.10 AMI 2.0.20221210.1  
x86\_64 HVM gp2  
ami-0b5eea76982371e91

Tipo de servidor virtual (tipo de instancia)

t2.micro

Firewall (grupo de seguridad)

Nuevo grupo de seguridad

Almacenamiento (volúmenes)

2 volúmen(es): 15 GiB

Tras asegurar que las opciones elegidas son las correctas, se confirma la creación de la instancia y nos aparecerá la nueva instancia en el panel principal de EC2:

Name	ID de la instancia	Estado de la i...	Tipo de inst...	Comprobación ...
Actividad 1	i-05e975b9b5af915ae	En ejecución	t2.micro	Inicializando

### 3.1.1. Mediante CLI

Si queremos realizar la creación de la máquina virtual a través del interfaz de consola **aws**, deberíamos ejecutar el siguiente comando:

#### Crear instancia EC2 mediante CLI

```
ddd_v1_w_Hox_1818884@runweb70100:~$ aws --region us-east-1 \
ec2 run-instances \
  --instance-type t2.micro \
  --key-name vockey \
  --image-id ami-0b5eea76982371e91 \
  --block-device-mappings \
    '[{"DeviceName":"/dev/xvda","Ebs":{"VolumeSize":10}},
    {"DeviceName":"/dev/xvdb","Ebs":{"VolumeSize":5}}]' \
  --security-group-ids sg-0d13061c1e89dec15
```

La sección del parámetro "**block-device-mappings**" se podría pasar a un fichero **JSON** externo en lugar de ponerlo en el comando.

## 3.2. Configuración previa

Para conectarnos a la instancia se ha obtenido la **clave SSH** que otorga AWS, ya que el acceso por usuario+contraseña está deshabilitado por defecto. Para realizar el acceso se ha usado el siguiente comando:

### </> Acceso SSH a la instancia





```
ruben@vega:~$ ssh -i labsuser.pem ec2-user@44.211.145.168
```

### 3.2.1. Creación del sistema de ficheros

Dado que se ha añadido un segundo volumen de 5GiB a la instancia, tenemos que hacer que este sea accesible. Este nuevo volumen está situado en la ruta


 /dev/xvdb


Por defecto no está formateado, por lo que debemos realizar:

- Crear partición a través del comando  `fdisk`. Usaremos todo el volumen.
- Crear sistema de ficheros EXT-4 con el comando  `mkfs.ext4`
- Modificar el fichero  `/etc/fstab` para que la nueva partición se monte en  `/data` en caso de que la instancia se reinicie:

### </> Contenido de /etc/fstab

```
UUID=a6741dbd-5f1a-4fab-8ea1-629052b08877 /data ext4 defaults,noatime 0 0
```

Se podría haber utilizado  `/dev/xvdb1` en lugar del UUID, pero así nos aseguramos que no existe confusión con el posible cambio de orden de los volúmenes.

Una vez realizado esto, y creado el directorio de destino, con el comando  `mount -a` se montará el nuevo volumen, quedando:



#### </> Título

```
[root@ip-172-31-81-57 ec2-user]# df -h
```

S.ficheros	Tamaño	Usados	Disp	Uso%	Montado en
devtmpfs	474M	0	474M	0%	/dev
tmpfs	483M	0	483M	0%	/dev/shm
tmpfs	483M	416K	482M	1%	/run
tmpfs	483M	0	483M	0%	/sys/fs/cgroup
/dev/xvda1	10G	1,6G	8,5G	16%	/
tmpfs	97M	0	97M	0%	/run/user/1000
/dev/xvdb1	4,8G	24K	4,6G	1%	/data

### 3.2.2. Instalación de dependencias

La aplicación a desplegar, [Flarum](#), es un desarrollo creado con el *framework* [Laravel](#), por lo que es un desarrollo web que requiere de las siguientes dependencias:

- Servidor web Apache o Nginx
- PHP 7.4 o superior (con las extensiones: curl, dom, fileinfo, gd, json, mbstring, openssl, pdo\_mysql, tokenizer, zip)
- Base de datos MySQL 5.6 o superior

Para la gestión de la base de datos lo haremos mediante la instancia **RDS** que se ha creado previamente. Para realizar la instalación de Apache lo haremos mediante los repositorios de la distribución:

#### </> Instalación y activación de Apache

```
[root@ip-172-31-81-57 ec2-user]# yum install apache2
[root@ip-172-31-81-57 ec2-user]# systemctl enable httpd
[root@ip-172-31-81-57 ec2-user]# systemctl start httpd
```

Se ha decidido utilizar la versión 8.0 de [PHP](#) y realizar la instalación a través de los repositorios “extra” de Amazon, así como los módulos necesarios:

### </> Instalación de PHP y las dependencias necesarias

```
[root@ip-172-31-81-57 ec2-user]# amazon-linux-extras install php8.0
[root@ip-172-31-81-57 ec2-user]# yum install php-mbstring php-dom php-gd
```

## 3.3. Instalación de la aplicación

Para realizar la instalación de la aplicación Flarum se ha seguido la [guía de instalación](#), por lo que se usará [Composer](#) para la descarga del código.

### </> Descarga e instalación de Composer

```
[root@ip-172-31-81-57]# php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
[root@ip-172-31-81-57]# php composer-setup.php --install-dir=/bin --filename=composer
```

La instalación utilizará la ruta  `/data/www` donde se alojará la aplicación.

### </> Descarga e instalación de Composer

```
[root@ip-172-31-81-57 ec2-user]# mkdir /data/www
[root@ip-172-31-81-57 ec2-user]# cd /data/www/
[root@ip-172-31-81-57 www]# composer create-project flarum/flarum .
```

Y tras esto hay que añadir la configuración correspondiente del VirtualHost en el fichero  `/etc/httpd/conf/httpd.conf` y reiniciar el servicio.

### </> Configuración Apache

```
<VirtualHost *:80>
    DocumentRoot "/data/www/public"
    ServerName actividad1-08masw.universidadviu.com
    DirectoryIndex index.php
    <Directory "/data/www/public">
        AllowOverride All
        Options Indexes FollowSymLinks
        Require all granted
    </Directory>
</VirtualHost>
```

Tal como se puede ver en la configuración, se ha añadido un dominio a la configuración, **actividad1-08masw.universidadviu.com**, por lo que para

acceder deberemos modificar nuestro fichero `/etc/hosts`. Tras esto, en el navegador podremos ver cómo aparece la web de configuración de Flarum:

Forum Title	Actividad 1
MySQL Host	database-1.c92nioexause.us-east-1.rds.am
MySQL Database	flarum
MySQL Username	admin
MySQL Password	*****
Table Prefix	

Se ha introducido la configuración obtenida de la creación de la base de datos RDS, y tras esto ya tenemos acceso a la aplicación. Para comprobar su correcto funcionamiento se han creado dos posts de pruebas:

Actividad 1

Start a Discussion All Discussions Following Tags General

Latest ▾

Rubén Gómez Olivencia  
rugoli started a few seconds ago

VIU Actividad 1  
rugoli started 2 minutes ago

## 4. Grupos de seguridad

Para facilitar la seguridad de las instancias existen los **grupos de seguridad**. Con ellos, podemos crear reglas de entrada y reglas de salida, para distintos puertos y protocolos que nos interese filtrar.

Podemos crear grupos de seguridad que posteriormente se asignan a instancias, y de esta manera centralizar la seguridad. En caso de necesitar añadir o quitar IPs o protocolos, al modificar el grupo de seguridad se aplicará a las instancias que hagan uso de dicho grupo de seguridad.

A modo de ejemplo se ha creado el grupo de seguridad “Servidores Web”, que cuenta con las siguientes reglas de entrada:

- Aceptar cualquier conexión al puerto TCP 80 (conexión HTTP).
- Aceptar cualquier conexión al puerto TCP 443 (conexión HTTPS).
- Aceptar conexiones al puerto 22 (SSH) sólo desde la IP 83.138.41.161 (administrador remoto).
- Aceptar conexiones al puerto 22 (SSH) sólo desde la IP 47.61.134.184 (administrador remoto).

Reglas de entrada (4)

Filtrar reglas de grupo de seguridad

Administrar etiquetas

Editar reglas de entrada

<input type="checkbox"/>	Name	ID de la regla del g...	Versión de IP	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción
<input type="checkbox"/>	-	sgr-0b17188770af2f7e3	IPv4	SSH	TCP	22	83.138.41.161/32	-
<input type="checkbox"/>	-	sgr-016f13114870e656f	IPv4	HTTP	TCP	80	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-083c7feec9aee4dd7	IPv4	SSH	TCP	22	47.61.134.184/32	-
<input type="checkbox"/>	-	sgr-06144a14ea9e2a3...	IPv4	HTTPS	TCP	443	0.0.0.0/0	-

De esta manera, podríamos aplicar este grupo de seguridad a todas las instancias que creemos a partir de ahora que van a hacer uso de un servidor web. Y en caso de querer tener alguna restricción más, modificando este grupo, hará que se aplique a todas las instancias, en lugar de ir una a una realizando la modificación.

Tras crear el grupo lo que tenemos que hacer es asignarlo a la instancia creada previamente:

Instancia: i-05e975b9b5af915ae (Actividad 1)	
Detalles	Seguridad
▼ Detalles de seguridad	
Rol de IAM -	ID del propietario 314890671564
Grupos de seguridad sg-0d84732b4b07b1c7c (Servidores Web)	

También podremos realizar la asignación en el momento en el que se crea una instancia nueva, de esta manera nos aseguramos que ya desde el momento de la creación sólo se tiene acceso desde las IPs que nos interese.

## 4.1. Mediante CLI

Los grupos de seguridad también se pueden crear a través de línea de comandos, pero es necesario hacerlo en varios pasos. Primero se crea el “*security group*” y después se le añaden las reglas.

Se va a crear el ejemplo expuesto previamente a través del CLI, y para añadir las reglas es necesario conocer el **GroupID** que devuelve el primer comando:

### ❏ Crear *security group* y añadirle reglas

```
ddd_v1_w_Hox_1818884@runweb70100:~$ aws --region us-east-1 \
  ec2 create-security-group \
    --group-name servidores-web \
    --description "para servidores web"

ddd_v1_w_Hox_1818884@runweb70100:~$ aws --region us-east-1 \
  ec2 authorize-security-group-ingress \
    --group-id sg-0d13061c1e89dec15 \
    --protocol tcp --port 22 \
    --cidr 83.138.41.161/32

ddd_v1_w_Hox_1818884@runweb70100:~$ aws --region us-east-1 \
  ec2 authorize-security-group-ingress \
```

```
--group-id sg-0d13061c1e89dec15 \  
--protocol tcp --port 22 \  
--cidr 47.61.134.184/32  
  
ddd_v1_w_Hox_1818884@runweb70100:~$ aws --region us-east-1 \  
ec2 authorize-security-group-ingress \  
--group-id sg-0d13061c1e89dec15 \  
--protocol tcp --port 80 \  
--cidr 0.0.0.0/0  
  
ddd_v1_w_Hox_1818884@runweb70100:~$ aws --region us-east-1 \  
ec2 authorize-security-group-ingress \  
--group-id sg-0d13061c1e89dec15 \  
--protocol tcp --port 443 \  
--cidr 0.0.0.0/0
```

Es importante darse cuenta que a la hora de crear mediante línea de comandos la instancia EC2 y la RDS se han añadido distintos IDs de *security-groups*, que lógicamente deben existir previamente.

## 5. IP elástica

Por cómo funciona las instancias EC2, la asignación de IPs públicas es temporal, por lo que en el momento en el que una instancia se para, al volverla a levantar no obtendrá la misma IP pública.


Para evitar esto están las **IP elásticas**, que se pueden crear desde el menú de EC2, en “**Red y Seguridad > Direcciones IP elásticas**”. Desde este apartado podremos realizar la solicitud de una IP elástica, y para poder utilizarla tiene que ser asignada a una instancia.

Es por ello que hay que seleccionar la IP elástica recién creada, y a través del menú “Acciones” pulsar la opción “Asociar la dirección IP elástica. De esta manera podremos asociarla a la instancia que nos interese. En nuestro caso a la instancia

creada previamente:

<input checked="" type="checkbox"/>	Name ▾	Dirección IPv4 asig... ▾	Tipo ▾	ID de asignación ▾	ID de la instancia asociada ▾
<input checked="" type="checkbox"/>	-	34.194.139.23	IP pública	eipalloc-00a27acda1ca1f0df	i-05e975b9b5af915ae <a href="#">↗</a>

De esta manera, aunque ahora la instancia se pare, no pasará nada, ya que cuando se vuelva a levantar mantendrá la IP.

Es importante acordarse que hemos modificado el fichero  `/etc/hosts`, por lo que ahora deberemos reflejar la nueva IP para poder acceder al dominio de la aplicación. Lo mismo para el acceso por SSH.

## 6. Bucket S3

Otro de los servicios que ofrece AWS es **S3** (de “*Simple Storage Service*”), donde podremos almacenar objetos a través de un interfaz REST (o a través de los SDK). Se han elegido las siguientes opciones al crear un bucket:

### Configuración general

Nombre del bucket

rg-actividad1-08masw

El nombre del bucket debe ser único en todo el mundo y no debe contener espacios ni letras mayúsculas.  
[Consulte las reglas para la denominación de los buckets](#) [↗](#)

Región de AWS

EE. UU. Este (Norte de Virginia) us-east-1 ▾

### Cifrado predeterminado

Cifre automáticamente los nuevos objetos almacenados en este bucket. [Más información](#) [↗](#)

Cifrado del lado del servidor

☐ Desactivar

☒ Habilitar

Tipo de clave de cifrado

Para cargar un objeto con una clave de cifrado proporcionada por el cliente (SSE-C), utilice la CLI de AWS, el SDK de AWS o la API REST de Amazon S3.

☒ Claves administradas por Amazon S3 (SSE-S3)

Una clave de cifrado que Amazon S3 crea, administra y utiliza por usted. [Más información](#) [↗](#)

☐ Clave de AWS Key Management Service (SSE-KMS)

Una clave de cifrado protegida por AWS Key Management Service (AWS KMS).  
[Más información](#) [↗](#)

Una vez creado el bucket, desde el interfaz se pueden crear carpetas y subir ficheros. Dependiendo de si queremos que sean públicos o privados, deberemos de crearlos con la ACL (*access control list*) adecuada. A continuación dos ficheros añadidos al bucket que son accesibles públicamente:

- [docu1.pdf](#)
- [imagen1.png](#)

## 6.1. Mediante CLI

Si queremos crear el bucket desde la línea de comandos se puede hacer con el siguiente comando:

### </> Crear bucket S3

```
ddd_v1_w_Hox_1818884@runweb69965:~$ aws s3 mb s3://rg-actividad1-08masw
```

Una vez creado, si queremos que el bucket esté cifrado, deberemos asignarle el tipo de cifrado:

### </> Añadir encriptación al bucket S3

```
ddd_v1_w_Hox_1818884@runweb69965:~$ aws --region us-east-1 \
s3api put-bucket-encryption \
--bucket rg-actividad1-08masw2 \
--server-side-encryption-configuration \
'{"Rules":[{"ApplyServerSideEncryptionByDefault":{"SSEAlgorithm": "AES256"}}}]'
```

Se pueden añadir ficheros a través del interfaz web o desde el la consola de AWS. Si queremos que sea público, tendremos que indicarlo mediante otro comando:

### </> Subir fichero PDF al bucket S3 y hacerlo público

```
ddd_v1_w_Hox_1818884@runweb69965:~$ aws s3 cp docu1.pdf \
s3://rg-actividad1-08masw/actividad1/docu1.pdf

ddd_v1_w_Hox_1818884@runweb69965:~$ aws s3api put-object-acl \
--bucket rg-actividad1-08masw \
--key actividad1/docu1.pdf \
```



```
--acl public-read
```

En lugar de hacerlo mediante dos comandos, se podría utilizar directamente `>_ aws s3api put-object ...`, pero es posible que quizá haya ficheros que nos interese mantener privados.

## 7. Costes estimados

Debido a todas las opciones con las que cuenta AWS, puede resultar confuso el precio final que se termina pagando. Es por eso que Amazon cuenta con el [Amazon Pricing Calculator](#) para poder estimar los costes.

Se ha tenido en cuenta las siguientes características para realizar una estimación de precios:

- Instancias bajo demanda sin reserva.
- Estimación tráfico de entrada servidor web de 15 GB/mes
- Estimación tráfico de salida servidor web de 30GB/mes.
- Tráfico de salida del servicio de S3 10GB/mes.

En la siguiente tabla se van a comparar los costes estimados por mes para dos regiones (España y USA-Virginia) teniendo en cuenta todo lo visto hasta ahora:

	España	N. Virginia
<b>Instancia EC2 t2.micro Linux, 15GB EBS gp2, con tráfico estimado</b>	\$9,97/mes *	\$9,97/mes
<b>IP elástica</b>	\$0/mes	\$0/mes
<b>Base de datos RDS MySQL db.t3.micro, single-AZ, sin proxy, almacenamiento 20GB, 40GB copias de seguridad **</b>	\$20,54/mes	\$18,51/mes

<b>Bucket S3, 1GB de almacenamiento, con tráfico estimado de salida ***</b>	\$0,02/mes	\$0,02/mes
---	------------	------------

\* En España sólo se pueden coger instancias **t3**.micro, que tiene más CPU.

\*\* La estimación para los backups al comienzo del proyecto podría ser menor, pero si se pretende llegar a los 20GB de espacio, para una retención de 10 días los 40GB se quedarán cortos.

\*\*\* Si el número de peticiones está por debajo de 100.000 el precio no varía. Tampoco se tiene en cuenta el cargar los datos iniciales.

Tal como se ha podido comprobar en la tabla, los precios son similares en ambas regiones. En el caso de la instancia EC2, por el mismo precio obtenemos una máquina virtual mejor, mientras que la instancia de base de datos es un poco más cara en el CPD de Amazon en España.

De todas formas, también hay que tener en cuenta la latencia de acceso a las instancias. En el mejor de los casos el acceso a Virginia es dos veces más lento que a España, y en el peor hasta 6 veces. Para comprobarlo se pueden utilizar distintas herramientas que comprueban la latencia ([enlace 1](#), [enlace 2](#)).

## 8. Conclusiones

Tal como se ha podido ver, la creación de una instancia de máquina virtual junto con una base de datos separada, para realizar el despliegue de una aplicación web, es una labor de unos pocos clicks a través del interfaz web o unos pocos comandos.

Este documento sólo ha sido una introducción para unos pocos servicios, sin profundizar en todas las características, ya que AWS también nos permitirá realizar una infraestructura mucho más compleja, pero a la hora de crearla será igual de sencillo hacerlo.