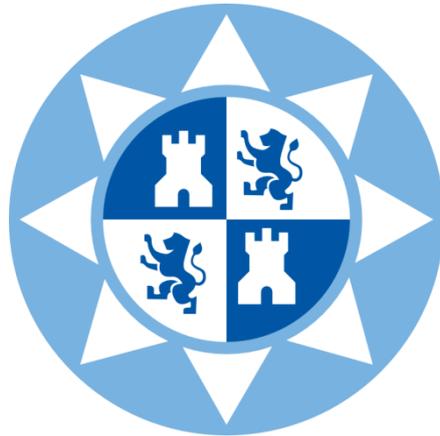


ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE
TELECOMUNICACIÓN
UNIVERSIDAD POLITÉCNICA DE CARTAGENA



Proyecto Fin de Carrera

**DESPLIEGUE EN AMAZON WEB SERVICES DE UNA
APLICACIÓN WEB PARA PROCESADO DE
IMÁGENES**



AUTOR: M^a Teresa García Muñoz
DIRECTOR: Fernando Losilla López
Octubre / 2015



Autor	M ^a Teresa García Muñoz
E-mail del Autor	tsy_gm28@hotmail.com
Director	Fernando Losilla López
E-mail del Director	fernando.losilla@upct.es
Codirector(es)	
Título del PFC	Despliegue en Amazon Web Services de una Aplicación Web para procesado de imágenes
Descriptor(es)	
Resumen	
<p>El trabajo que se presenta en la memoria de este Proyecto Final de Carrera es el despliegue de una aplicación web de procesado de imagen basado en librerías preexistentes con el fin de poder comprender el proceso de despliegue y configuración necesario en la consola de gestión de Amazon Web Services y en el propio código.</p> <p>El diseño de la aplicación permitirá solventar las peticiones de los clientes que se conecten desde su navegador y quieran procesar una imagen propia, utilizando la tecnología del Cloud Computing, concretamente utilizando uno de los más famosos y pioneros proveedores de esta tecnología, Amazon Web Services.</p> <p>Para poder realizar este despliegue, ha sido necesario comprender cómo funciona Amazon Web Services y llevar a cabo la creación de un entorno web e incluir las especificaciones necesarias para el funcionamiento correcto de la aplicación.</p> <p>En lo referente a la aplicación, se ha utilizado un código ya existente con múltiples modificaciones que han sido imprescindibles para el posible despliegue de la misma utilizando la versión de prueba gratuita que ofrece Amazon Web Services.</p>	
Titulación	Ingeniería de Telecomunicación
Intensificación	
Departamento	Tecnologías de la Información y las Comunicaciones
Fecha de Presentación	Octubre 2015

AGRADECIMIENTOS

A mi familia en primer lugar, agradezco todo el apoyo y la insistencia de continuar hacia delante a pesar de los obstáculos que me he ido encontrando a lo largo de estos años ya que no ha sido tan fácil como me esperaba, en especial a mis abuelos porque al mismo tiempo que yo aprendo de vosotros, cada día aprendéis algo nuevo en lo referente a las tecnologías y eso me llena de orgullo.

A los profesores que me han impartido clase, pues de cada uno de ellos he ido aprendiendo cosas nuevas, a mi director de proyecto Fernando Losilla por brindarme la oportunidad de trabajar con él y por ir guiándome a la hora de realizar todo el proyecto.

A mis amigos de toda la vida que no sé qué haría sin vosotros, a los compañeros de carrera que me han animado cuando he flaqueado en algún momento, a mis princesas antigonas Maria del Carmen, Rocio, Patricia, Inma, Trini y Elena por todos los días vividos en Cartagena de bibliotecas, risas, lloros, abrazos, cotilleos y notitas de motivación.

A David, que por muchos años que pasen siempre serás el hombre del futuro y podremos con todo. A Moira, Carmen, Lore y Olaya, por todas las tardes, bailes, tonterías y locuras que hemos cometido y que nos quedan. A Adrián, por estar siempre, en mis buenos y en mis malos momentos que suelen ser breves si los paso en tu compañía.

A Nuria, que es una de esas personas que parecen que llegan las últimas a tu vida pero rápidamente ocupan un gran lugar en tu corazón. A Edu por esas conversaciones telecas tan interesantes acompañadas de unas risas, bailes y cervezas. A Emi, que has estado siempre a mi lado desde que nos conocimos, por aguantarme cada día, por animarme a conseguir todo lo que me propongo, porque espero que sigamos viviendo tantas aventuras como las que hemos tenido hasta ahora y porque me has hecho ver de nuevo las cosas desde otra perspectiva y siempre con una sonrisa en la cara.

Y por último pues me agradezco a mi misma la realización de este proyecto, por haber vuelto a plantearme nuevos retos y a luchar por conseguirlos.

*“Sólo aquellos que se atreven a tener grandes fracasos,
terminan consiguiendo grandes éxitos.”*

Robert F. Kennedy.

ÍNDICE DE CONTENIDOS

DESPLIEGUE EN AMAZON WEB SERVICES DE UNA APLICACIÓN WEB PARA PROCESADO DE IMÁGENES

INTRODUCCIÓN	1
1.1 DESCRIPCIÓN DEL PROYECTO	1
1.2 OBJETIVO DEL PROYECTO	2
1.3 ESTRUCTURA DE LA MEMORIA	3
DESCRIPCIÓN DE LAS TECNOLOGÍAS EMPLEADAS	4
2.1 CLOUD COMPUTING	4
2.1.1 ¿Qué es Cloud Computing?	4
2.1.2 Modelo Cloud: Características, modelos de servicio y modelos de despliegue.....	5
2.1.3 Ventajas e Inconvenientes de Cloud Computing.....	9
2.1.4 Virtualización.....	11
2.1.5 Virtualización en Cloud Computing	13
2.2 AMAZON WEB SERVICES	14
2.2.1 ¿Qué es Amazon Web Services?	14
2.2.2 Historia de Amazon Web Services.....	15
2.2.3 Regiones de AWS.....	16
2.2.4 Seguridad en AWS.....	17
2.2.5 Productos de AWS	18
2.2.6 Amazon SQS.....	19
2.2.7 Amazon S3.....	20
2.2.8 Amazon DynamoDB.....	22
2.2.9 Amazon EC2	24
FUNCIONAMIENTO DE AMAZON WEB SERVICES	27
3.1 CREACIÓN DE UNA INSTANCIA EN AMAZON EC2	27
3.1.1 Consola de AWS.....	27
3.1.2 Creación de Instancia de Máquina Virtual.....	29
3.1.3 Configuración de red del servidor creado	37
3.1.4 Eliminación de instancias	38

3.2 CONEXIÓN A UNA INSTANCIA EN AMAZON EC2	39
3.2.1 Pasos Previos a la conexión de la Instancia	39
3.2.2 Métodos de conexión a una Instancia	42
3.2.3 Volúmenes de almacenamiento de una Instancia	48
3.3 FUNCIONAMIENTO DE SERVICIOS EN LA CONSOLA AWS	52
3.3.1 Funcionamiento de Amazon SQS.....	52
3.3.2 Funcionamiento de Amazon S3	56
3.3.3 Funcionamiento de Amazon DynamoDB.....	60
DESARROLLO DE UNA APLICACIÓN CON AMAZON WEB SERVICES.....	66
4.1 PROCESADOR DE IMÁGENES.....	66
4.1.1 Introducción.....	66
4.1.2 Requisitos de la aplicación	66
4.1.3 Funcionamiento de la Aplicación	67
4.2 CREACIÓN DE LA APLICACIÓN	68
4.2.1 Subida de la imagen a la aplicación	68
4.2.2 Pasos previos al procesado del archivo.....	69
4.2.3 Procesado del archivo	72
4.2.4 Recuperación del archivo procesado.....	76
4.2.5 Finalización del procesado de imágenes.....	77
4.3 CONFIGURACIÓN DE LA INSTANCIA EC2	78
4.3.1 Configuración del servidor LAMP	78
4.3.3 Prueba de la configuración realizada de la instancia.....	83
4.3.4 Instalación de librerías en la instancia	84
4.4 RESULTADOS OBTENIDOS	85
PRESUPUESTO	88
CONCLUSIONES Y LÍNEAS FUTURAS.....	89
6.1 CONCLUSIONES	89
6.2 LÍNEAS FUTURAS.....	89
GLOSARIO DE TÉRMINOS.....	90
REFERENCIAS Y BIBLIOGRAFÍA	92

ÍNDICE DE FIGURAS Y TABLAS

- Figura 1: Cloud Computing, 4
- Figura 2: Modelos de servicio, 6
- Figura 3: Modelos de despliegue, 8
- Figura 4: Virtualización, 11
- Figura 5: Ejemplo de Virtualización, 12
- Figura 6: Amazon Web Services, 14
- Figura 7: Línea Temporal de AWS, 15
- Figura 8: Regiones AWS, 16
- Figura 9: Amazon SQS, 19
- Figura 10: Amazon S3, 20
- Figura 11: Amazon EC2, 24
- Figura 12: Iniciar Consola de Amazon, 27
- Figura 13: Consola de Amazon, 28
- Figura 14: Lanzamiento de la Instancia, 29
- Figura 15: Selección de AMI, 30
- Figura 16: Tipo de Instancia, 31
- Figure 17: Detalles en la configuración de la instancia, 32
- Figura 18: Agregar Almacenamiento, 33
- Figura 19: Etiquetas de la Instancia, 34
- Figura 20: Configuración del Grupo de Seguridad, 35
- Figura 21: Creación del par de claves RSA, 36
- Figura 22: Aspecto de la "Lista de instancias", 37
- Figura 23: Eliminación de una instancia, 38
- Figura 24: DNS Pública de la Instancia, 39
- Figura 25 : Vista de las reglas de nuestra instancia, 40
- Figura 26: Grupos de seguridad, 41
- Figura 27: Agregar nueva regla en un grupo de seguridad, 41
- Figura 28: Descarga de PuTTY, 43
- Figura 29: Carga archivo .pem en PuTTYgen, 43
- Figura 30: Clave privada .ppk generada, 44
- Figura 31: Configuración de la Sesión PuTTY, 44
- Figura 32: Inicio de conexión en PuTTY, 45
- Figura 33: Instancia con PuTTY, 45
- Figura 34: Conexión a la instancia desde navegador, 46

Figura 35: Instancia con MindTerm, 47

Figura 36: Zona de disponibilidad de la Instancia, 48

Figura 37: Creación de un Volumen de Almacenamiento, 49

Figura 38: Adjuntar volumen de almacenamiento, 50

Figura 39: Ciclo de vida de un mensaje SQS, 52

Figura 40: Crear nueva cola SQS, 54

Figura 41: Especificaciones de la cola SQS, 54

Figura 42: Aspecto lista de cola SQS, 55

Figura 43: Crear Depósito, 57

Figura 44: Nombre del depósito, 57

Figura 45: Aspecto Lista de Depósitos, 58

Figura 47: Asistente de subida de objetos, 58

Figura 48: Subida del Archivo al depósito, 59

Figura 49: Objeto Subido, 59

Figura 50: Crear Tabla, 63

Figura 51: Clave Primaria, 63

Figura 52: Añadir índices, 64

Figura 53: Capacidad de aprovisionado, 64

Figura 54: Configuración del uso básico de alarmas, 65

Figura 55: Aspecto Lista de tablas en DynamoDB, 65

Figura 56: Procesador de Imágenes, 67

Figura 57: Servidor Web Apache en Amazon Linux, 79

Figura 58: Prueba del servidor Apache, 80

Figura 59: WinSCP, 81

Figura 60: Clave privada en WinSCP, 82

Figura 61: Transferencia de archivos con WinSCP, 83

Figura 62 : Prueba PHP en Amazon Linux, 83

Figura 63: Procesador de Miniaturas, 85

Figura 64: Thanks.php, 85

Figura 65: Imagen Cargada en la aplicación, 86

Figura 66: Imagen Procesada, 86

Figura 67: Imagen para descargar, 87

Figura 68: Archivo eliminado, 87

Figura 69: Amazon Web Simple Monthly Calculator, 88

Tabla 1: Nombres de usuarios por Sistema Operativo, 42

INTRODUCCIÓN

1.1 DESCRIPCIÓN DEL PROYECTO

El presente Proyecto Final de Carrera (PFC), mediante el trabajo en el estudio y análisis de la tecnología del Cloud Computing, presentará un estudio específico sobre el desarrollo de aplicaciones web en “La Nube” con uno de los proveedores pioneros y más conocidos, Amazon Web Services.

En la actualidad, la forma en la que se almacena nuestra información o el despliegue de servicios ha cambiado respecto a hace unos años. Antes, una persona guardaba localmente todos sus datos en un dispositivo, y si esta persona deseaba poder utilizar dicha información desde otro lugar u otro dispositivo, era necesario transportar los datos de un dispositivo a otro o a un almacenamiento externo. Ahora, una persona puede acceder a todos sus datos (imágenes, vídeos, música, documentos, etc...) desde cualquier dispositivo que tenga y desde el lugar del mundo en el que se encuentre, pues esta información se guarda en el sitio conocido como “La Nube”. La oferta de servicios en “La Nube” de la mano de los proveedores son de todo tipo, y cualquier persona u organización puede acceder a dichos servicios para su utilización.

El comienzo del “Cloud Computing”, no obstante, empezó hace décadas, cuando se envió el primer correo electrónico y se almacenaron éstos emails de forma remota. Este acontecimiento dio lugar a una auténtica revolución tecnológica que aún perdura y que supone uno de los hitos de la era actual.

Para poder realizar este estudio, hemos utilizado las herramientas que nos ofrece el proveedor Amazon Web Services para explicar cómo se desarrollaría una aplicación sencilla como es un procesador de imágenes en este entorno web.

Para trabajar con las herramientas de Amazon Web Services, se ha tenido que crear una cuenta, la cual Amazon abastece a cualquier persona de un año de prueba gratuita para la libre utilización de sus servicios.

1.2 OBJETIVO DEL PROYECTO

El objetivo de este proyecto será estudiar cómo se desarrolla desde cero una aplicación en web utilizando Cloud Computing. Se explicará paso por paso los progresos que tendría que hacer un usuario cualquiera para poder desarrollar una aplicación, un sencillo procesador de imágenes, en este entorno.

Para la realización del proyecto, hemos definido los objetivos que se muestran a continuación:

- Estudio teórico detallado de Cloud Computing y del proveedor que se ha utilizado, Amazon Web Services.
- Creación del entorno en “La Nube” que vamos a utilizar para poder desarrollar la aplicación.
- Estudio teórico de la aplicación que se va a realizar.
- Código comentado de la aplicación.
- Configuraciones de la Instancia necesarias para el correcto funcionamiento de la aplicación.

1.3 ESTRUCTURA DE LA MEMORIA

1.3.1 Introducción

Se corresponde con la primera parte de la memoria la cual contiene de forma general una breve explicación del proyecto, los objetivos de éste y los pasos realizados para resolver estos objetivos.

1.3.2 Descripción de las tecnologías utilizadas

Este capítulo desarrolla toda la teoría referente a las tecnologías utilizadas para este proyecto.

1.3.3 Funcionamiento de Amazon Web Services

En este apartado, se verá cómo se crea y se carga una máquina virtual sencilla en Amazon Web Services y explicada paso por paso para que sirva de guía a la hora de desarrollar cualquier tipo de aplicación en este entorno web.

1.3.4 Desarrollo de una aplicación con AWS

En este capítulo consta de la parte más importante del proyecto ya que se describe el proceso de desarrollo de una aplicación de procesamiento de imágenes con Amazon Web Services partiendo de la información del apartado anterior.

1.3.5 Presupuesto

El apartado de presupuesto describen los gastos necesarios para realización de este proyecto.

1.3.6 Conclusiones y Líneas futuras

Este apartado de la memoria contiene las conclusiones que se obtienen a lo largo del desarrollo del proyecto además de un apartado en el que se podrá ver a partir de este estudio qué mejoras se podrían implementar en un futuro no muy lejano.

1.3.7 Glosario de Términos

El glosario de términos incluye una lista de palabras técnicas que se utilizan a lo largo de este proyecto y útiles para una comprensión mejor del mismo.

1.3.8 Referencias y Bibliografía

DESCRIPCIÓN DE LAS TECNOLOGÍAS EMPLEADAS

Para realizar este proyecto se ha empleado uno de los proveedores más conocidos en todo el mundo, Amazon Web.

2.1 CLOUD COMPUTING

2.1.1 ¿Qué es Cloud Computing?

“Cloud Computing” también conocido como “computación en la nube”, es un sistema que se basa en que las aplicaciones software y los equipos hardware con capacidad de proceso y almacenaje de datos no se encuentran ubicados en el equipo del usuario, sino que se encuentran en un centro de almacenaje de datos permitiendo a los usuarios acceder a las aplicaciones y servicios disponibles a través de Internet (conocido como “la nube”) de forma cómoda y sencilla.

Según el NIST (National Institute of Standards and Technology) Cloud Computing se define como “el modelo tecnológico que permite el acceso ubicuo, adaptado y bajo demanda en red a un conjunto compartido de recursos de computación configurables (por ejemplo: redes, servidores, equipos de almacenamiento, aplicaciones y servicios) que pueden ser desplegados rápidamente, con mínimo esfuerzo de gestión o interacción con el proveedor del servicio.”



Figura 1: Cloud Computing

2.1.2 Modelo Cloud: Características, modelos de servicio y modelos de despliegue

El modelo Cloud se compone según el NIST de 5 características esenciales, 3 modelos de servicio y 4 modelos de despliegue.

Las características esenciales para considerar un servicio Cloud son las siguientes:

- **Accesibilidad desde la red:** Los usuarios pueden acceder a los servicios que se encuentran en la nube desde cualquier ordenador, smartphone, tablet, etc.
- **Autoservicio de servicios bajo demanda:** Un consumidor puede acceder a una serie de recursos de computación disponibles en la nube, sin interacción humana del proveedor, además de que puede aumentar o disminuir los recursos en función de lo que necesite en cada instante.
- **Compartición de recursos:** Todos los usuarios tienen acceso a los servicios disponibles en la nube, y los recursos, ubicados por centros de datos, países, provincias, etc., son asignados conforme la demanda del usuario en cada momento.
- **Capacidad de rápido crecimiento:** Según las necesidades del cliente la cantidad de recursos y servicios pueden incrementarse o decrementarse, en algunos casos puede automatizarse en función de la demanda. Al cliente le debe parecer que los recursos del servidor son infinitos, y podrá acceder a cualquier servicio y servicio en cualquier momento.
- **Servicio medido:** Se controla y se mide el uso de recursos computacionales por el proveedor, que debe notificar al cliente, evitando posibles colapsos de algún servicio o recurso. Los costes del servicio son predecibles.

Los modelos de Servicio se muestran a continuación:

Infrastructure as a Service (IaaS)

Este modelo ofrece servicios basados en utilizar recursos informáticos hardware (infraestructuras) a los usuarios que contraten estas prestaciones dependiendo en cada momento de sus necesidades, como por ejemplo recursos de computación, almacenamiento, redes, servidores, etc.

Las entidades o clientes que contraten estos servicios podrán utilizar estos recursos como si los tuvieran físicamente, es decir, que podrán desarrollar su software en la infraestructura del proveedor, pero no podrá gestionar dicha infraestructura. El acceso a estas infraestructuras se hace mediante una interfaz accesible a través de Internet y con un navegador.

Ejemplo: Amazon Web Services (AWS).

Platform as a Service (PaaS)

El proveedor ofrece al cliente una plataforma en la que se encuentran herramientas de desarrollo software, máquinas virtuales y entornos virtuales para poder crear nuevas aplicaciones y testearlas. En este modelo existe una capa de abstracción sobre los recursos del proveedor para que el cliente sólo se preocupe del despliegue de las aplicaciones accesibles desde cualquier dispositivo con acceso a Internet.

Ejemplo: Google App Engine.

Software as a Service (SaaS)

El servicio que se ofrece a los clientes es la ejecución de las aplicaciones software sobre la infraestructura que se encuentra en la nube, de esta manera, el usuario al contratar el servicio, accederá a través del navegador web de forma remota, pagando sólo por el volumen de utilización de los servicios ofrecidos.

Este modelo además es multi-tenancy, es decir la aplicación que se ejecuta en el proveedor, es utilizada por múltiples usuarios finales. Además estos usuarios no tienen porqué descargar actualizaciones de dichas aplicaciones, pues cada vez que utilicen los servicios ofrecidos por el proveedor llevarán las modificaciones necesarias para su uso.

Ejemplos: Google Apps, CRM, juegos online, etc.

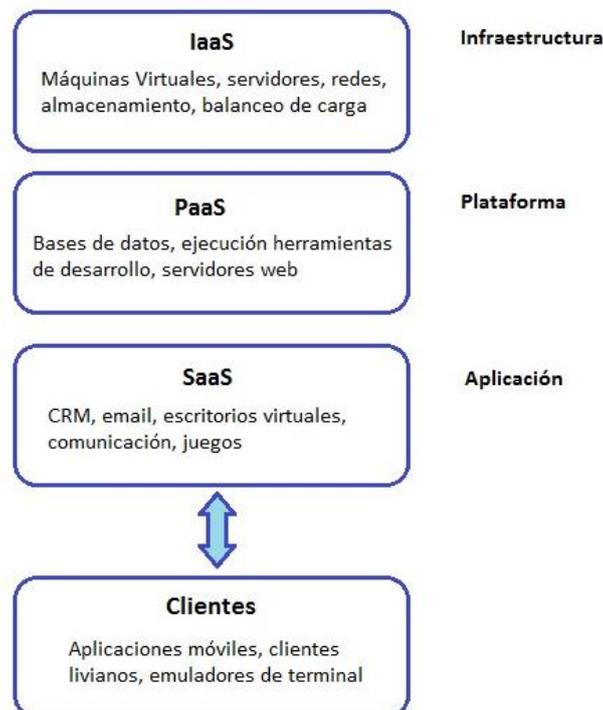


Figura 2: Modelos de servicio

Por último, los modelos de despliegue que tiene Cloud Computing son:

Nube privada

La nube privada tiene un mayor nivel de seguridad y control de datos e información que el resto de modelos de despliegue, suele encontrarse en las instalaciones de determinadas organizaciones para el uso de los miembros de la misma organización. La propia entidad decide qué, quiénes y dónde se pueden ejecutar los procesos, lo cual hace que no se ofrezcan servicios a personas fuera de la organización.

Nube pública

Los servicios que ofrecen se encuentran con servidores externos donde el acceso a estos servicios se puede hacer de forma gratuita o mediante el pago de un alquiler por uso. La información de los usuarios estará mezclada entre los distintos sistemas de almacenamiento, servidores e infraestructuras de esta nube.

Posee una alta capacidad de procesamiento y almacenamiento sin que el usuario tenga que invertir ni pagar por mantenimiento de ésta, sólo se pagará por el uso que se realice en ella. Además los servicios ofrecidos por el proveedor deberán estar siempre en línea ya que un fallo de conexión puede hacer que el cliente no pueda acceder a la información, aplicación o datos que desee.

Nube Comunitaria

Este tipo de nubes que no son tan comunes en la actualidad, en ella existe una infraestructura comunitaria compartida entre distintas organizaciones con intereses comunes. Estas nubes pueden ser controladas y administradas tanto internamente como por terceros, al igual que su localización puede encontrarse tanto en estas organizaciones como externas a ellas.

Nube Híbrida

Este tipo de nubes es resultado de la combinación de dos tipos de nubes vistas anteriormente, las nubes privadas y las públicas. De este modo, el cliente será propietario de algunas partes y compartirá otras manteniendo el control de la estructura. Requiere una inversión inicial menor que la de una nube privada, pero es más compleja cuando se distribuyen las aplicaciones a través de estos dos tipos de nubes ya descritos anteriormente.

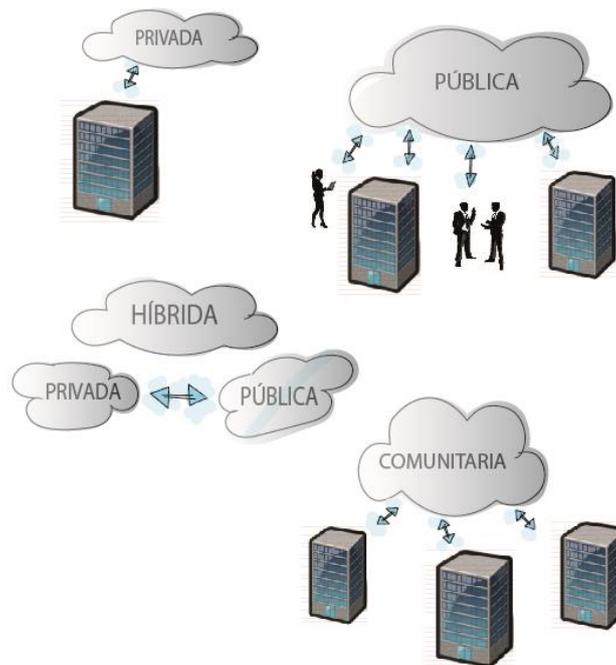


Figura 3: Modelos de despliegue

2.1.3 Ventajas e Inconvenientes de Cloud Computing

La tecnología Cloud Computing ha revolucionado el mundo tecnológico de gran significado, por ello, es necesario conocer tanto los beneficios como las amenazas de esta tecnología.

Las ventajas de Cloud Computing serán las siguientes:

- **Costes reducidos:** Debido a que no se necesita la adquisición tanto de hardware como de software se reducen los costes de la infraestructura, mantenimiento y energía. En la nube sólo se paga por los recursos y servicios utilizados, por tanto es más económica que la instalación y mantenimiento de un servidor propio o contratar los servicios de un proveedor.
- **Escalabilidad:** El servicio de nube se paga de acuerdo a la demanda. Esto beneficia tanto al cliente, ya que puede ampliar o reducir instantáneamente la provisión de sus recursos en base a sus necesidades, como al proveedor, ya que será más eficiente a la hora de la repartición de recursos.
- **Movilidad y alta disponibilidad:** Los datos de una empresa al estar ubicados en la nube pueden ser consultados por los empleados desde cualquier lugar del planeta. Además, los servicios tienen un tiempo de disponibilidad alto, lo cual hace que los recursos puedan estar siempre disponibles para los clientes.
- **Focalización:** Las organizaciones pueden centrarse en su negocio principal. La nube simplifica el proceso de desarrollo en el sentido de que ofrecerá a los clientes balanceados de carga, almacenamiento de datos, etc. haciendo así que la empresa ahorre tiempo y dinero en inversiones tecnológicas y que les sea más fácil continuar con los objetivos que se proponga como la expansión hacia nuevos mercados.
- **Ecología:** La utilización de la nube reduce la huella de carbono de una empresa al ahorrar recursos y componentes que pasan de estar almacenados en componentes físicos a ser virtuales además del ahorro en consumo de energía con sus beneficios al medio ambiente.

La facilidad para realizar copias de respaldo y recuperación de datos en caso de fallos, rápida integración e interfaz sencillo para los clientes, son otras de las ventajas que ofrece Cloud Computing.

Hay una serie de razones por las que es posible que haya usuarios no quieran adoptar la computación en nube para sus necesidades particulares. Las desventajas o inconvenientes de Cloud Computing serán las siguientes:

- Dependencia del proveedor: Si el proveedor falla, los clientes pueden perder todos los servicios y datos desplegados en él, por lo que se recomienda contratar los servicios cloud a grandes empresas, ya que serán más fiables en cuanto a fallos.
- Privacidad y seguridad de los datos: Algunos datos de los clientes pueden quedar expuestos a los ojos de los proveedores, normalmente hay de por medio un contrato de confidencialidad, por ello deben tomarse medidas de seguridad para evitar filtraciones de la información a terceros.
- Transferencia de datos: Si el proveedor no gestiona bien las redes y su capacidad de ancho de banda pueden haber problemas a la hora de transferir grandes cantidad de datos. Además, si deseamos portar aplicaciones de un proveedor a otro será difícil si no se rediseñan.
- Latencia: Los clientes pueden encontrarse lejos de los servidores, esto quiere decir que los servicios que requieran bajos niveles de latencia podrían verse afectados si se despliegan en plataformas cloud, por ello, algunos proveedores ofrecen la posibilidad de crear máquinas virtuales en servidores físicos ubicados donde desee el cliente.
- Muchas aplicaciones no requieren de Cloud Services, esto quiere decir que no se requiere escalabilidad, se van a ofrecer pocos clientes, los recursos están infrautilizados, aunque aun así podrán ser más económicas con proveedores Cloud.
- Bases de datos: El escalado de estas bases de datos será difícil ya que es muy costoso en cloud, se usarán bases de datos no relacionales para bases de datos grandes.

2.1.4 Virtualización

El concepto de virtualización es muy sencillo, la virtualización es la técnica empleada sobre las características físicas de algunos recursos computacionales para ocultarlas de otros sistemas, usuarios o aplicaciones que interactúen con ellos, es decir, un recurso físico, como un servidor, un sistema operativo o un dispositivo de almacenamiento, puede aparecer como si fueran varios recursos lógicos a la vez, o que varios recursos físicos aparezcan como un único recurso lógico. Todos los usuarios, aplicaciones y recursos disponibles podrán interactuar con éstos recursos virtualizados como si fueran recursos físicos reales.

En la virtualización, existen el sistema operativo anfitrión y el sistema operativo huésped. En el primero, el sistema operativo anfitrión es la máquina en la que se instala un software de virtualización y que presta parte de sus recursos hardware a la máquina virtual creada. El segundo, es la máquina virtual creada mediante software de virtualización que funciona gracias a los recursos prestados por el anfitrión.

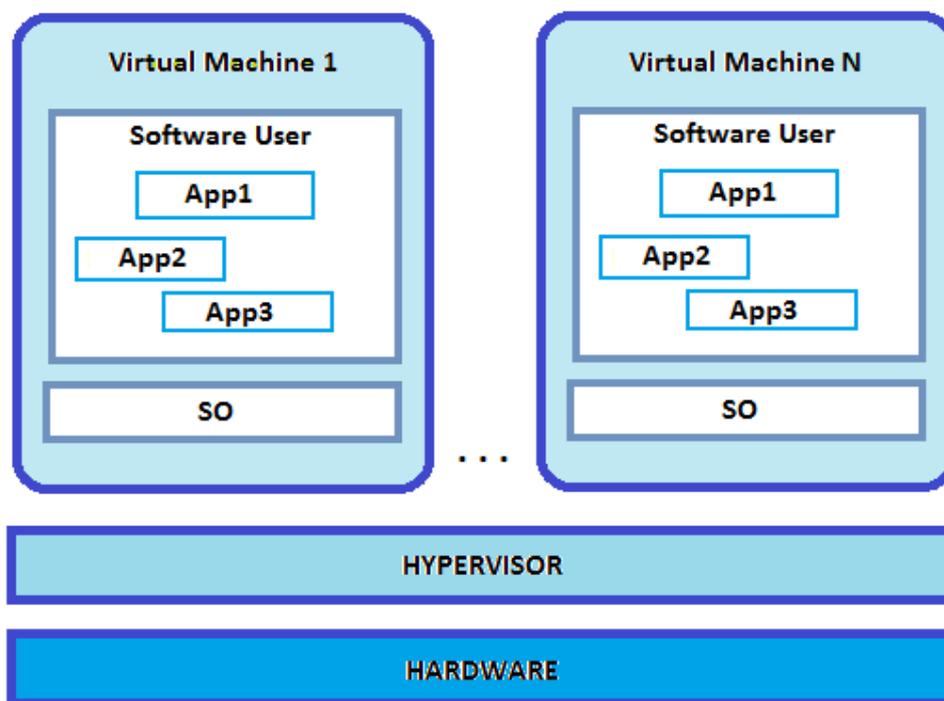


Figura 4: Virtualización

Como podemos observar en la anterior imagen, a partir de una máquina física podemos tener varias máquinas virtuales, cada una con un sistema operativo distinto si se desea y con varias aplicaciones y recursos que se necesiten para cada máquina virtual.

El Hypervisor o *Virtual Machine Monitor* (VMM), es la capa de software que permite la gestión de los recursos (CPU, memoria, conexiones de red y almacenamiento) de la máquina física real. Esta gestión permite ejecutar varias máquinas virtuales sobre un único Hardware con el que cuenta la máquina real física, y es donde el Hypervisor repartirá de forma dinámica dichos recursos entre las máquinas virtuales que se están ejecutando.

El hecho de poder ejecutar distintas máquinas virtuales hace que podamos ejecutar al mismo tiempo aplicaciones en cada máquina virtual, teniendo en cuenta las limitaciones de la máquina física y los recursos que nos asigna el Hypervisor, esto quiere decir que podemos ejecutar por ejemplo una aplicación de correo en una máquina virtual como si ésta fuera la máquina física.



Figura 5: Ejemplo de Virtualización

2.1.5 Virtualización en Cloud Computing

La virtualización tiene un papel importante con respecto a la computación en la nube, ya que la nube se apoya en las bases de la virtualización. No es necesario pero sí recomendable el virtualizar a la hora de pasarse a la nube, pues la virtualización ayuda a la hora de facilitar las necesidades de los clientes en cada momento, compartiendo los recursos y teniendo unas infraestructuras más flexibles y rápidas, entre otros.

Debido a la abstracción de los recursos físicos de la virtualización en Cloud Computing, provee de los siguientes beneficios a la nube:

- **Ahorro de energía, espacio y costes de operación:** La virtualización reduce el consumo total de energía que se utiliza respecto al uso de energía que se utilizaba antes de ésta. Además se ahorra mucho espacio en servidores físicos, ya que cuenta con varios servidores virtuales por cada servidor físico. El gasto en mantenimiento quedará reducido ya que no se invertirá tanto dinero en la infraestructura realizada.
- **Seguridad:** Se facilita nuevas opciones de soluciones para la recuperación de información frente a fallos. Las máquinas virtuales están aisladas entre sí, lo que quiere decir que si una máquina virtual cae, el resto seguirá funcionando. Además no se podrá acceder a datos confidenciales de otra máquina virtual lo cual aumenta la seguridad ante posibles ataques de terceros.
- **Particionado y elasticidad:** gracias a la virtualización un dispositivo se verá como varios virtuales y podrá ser compartido por varios clientes a la vez, además de que gracias a la abstracción que se tiene respecto al hardware se podrán tener tantas máquinas virtuales como se deseen.

2.2 AMAZON WEB SERVICES

2.2.1 ¿Qué es Amazon Web Services?

Amazon Web Services (AWS) es el conjunto de servicios web (computación en la nube) que forman una plataforma de Cloud Computing ofrecida por Amazon.com y es uno de los proveedores de la computación en nube más importantes de todo el planeta.



Figura 6: Amazon Web Services

Todos los servicios que se ofrecen en Amazon Web se clasifican en distintos grupos, los más importantes serán almacenamiento, bases de datos, entrega de contenido, mensajería y procesamiento de datos.

Muchas son las aplicaciones que utilizan los servicios de Amazon Web, algunas tan populares como Dropbox, Foursquare o HootSuite entre otras. Además, AWS compite directamente con otros servicios tan importantes como Microsoft Azure y Google Cloud Platform.

2.2.2 Historia de Amazon Web Services

En el año 2002, Amazon creó el AWS inicial, una interfaz para acceder a las funciones de Amazon, siendo uno de los pioneros en ofrecer este tipo de servicios web al mundo. Con el paso de los años, la compañía empezó a vender y construir nuevos servicios, lo que haría un gran cambio en las tecnologías de información (IT).

Fue lanzado oficialmente el 14 de Marzo de 2006, ofreciendo servicios en línea para otros sitios web o aplicaciones del lado del cliente, aunque la mayoría de estos servicios se ofrecen para que los clientes finales utilicen las aplicaciones de otros desarrolladores.

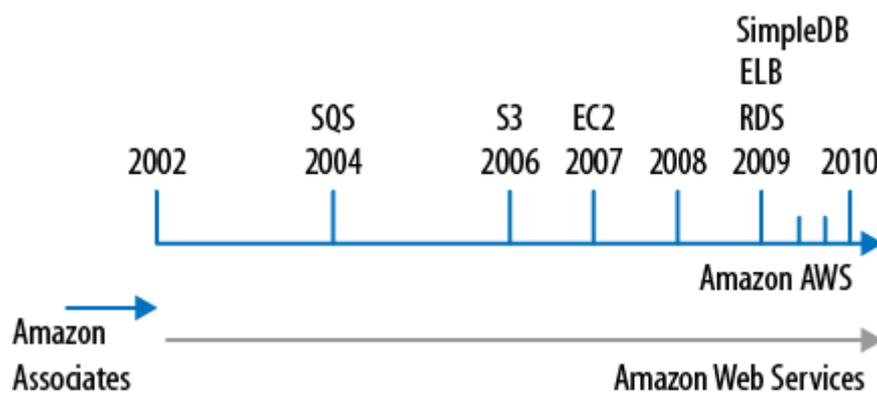


Figura 7: Línea Temporal de AWS

De entre todos los servicios que lanzaron, los más destacados son Amazon Simple Queue Service (Amazon SQS) lanzado en el año 2004, Amazon Simple Storage Service (Amazon S3) lanzado en 2006 y Amazon Elastic Compute Cloud (Amazon EC2) lanzado oficialmente en 2007. En futuros apartados se explicarán la funcionalidad de estos servicios que han permitido que AWS se convierta en uno de los líderes de mercado mundiales en servicios IaaS.

2.2.3 Regiones de AWS

Amazon Web Services dispone de centros de datos distribuidos por todo el mundo, estos centros de datos se construyen en grupos y parte el mundo en regiones. Los clientes tienen actualmente acceso a once regiones AWS, éstos, pueden optar por utilizar todas las regiones, una única región, o una combinación. Las ubicaciones de las regiones de AWS serán las siguientes:

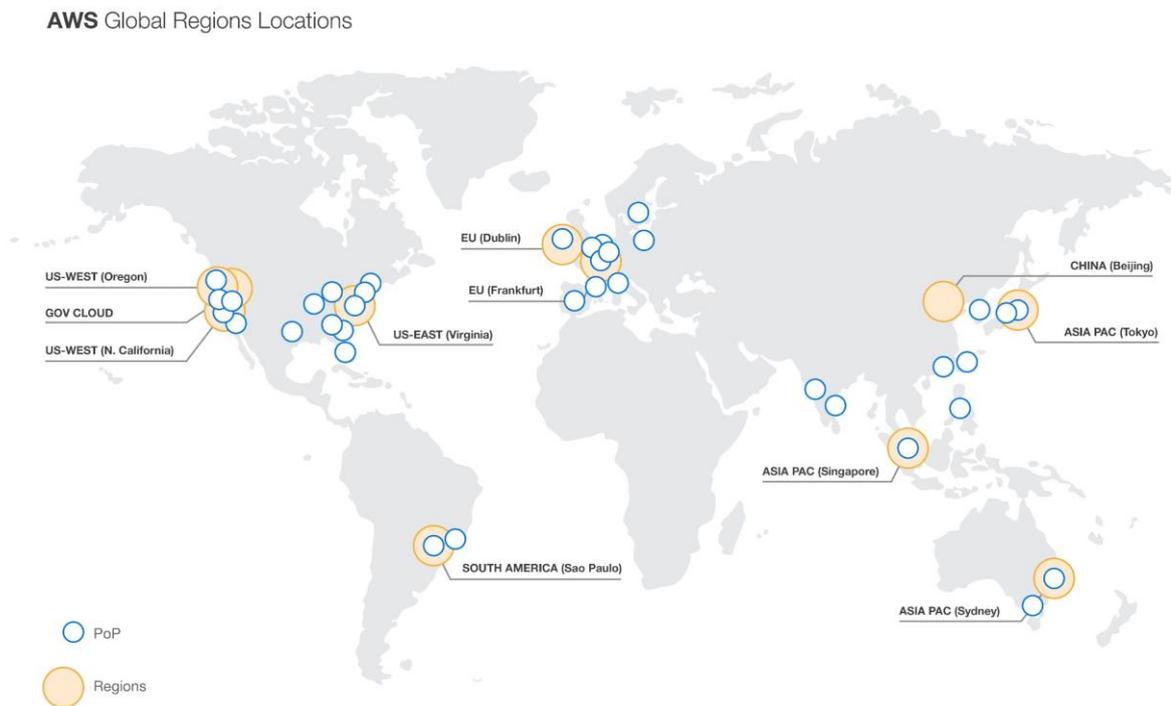


Figura 8: Regiones AWS

Los beneficios de que haya varias regiones de centros de datos AWS es trivial, pues mejorará la latencia de los usuarios pertenecientes a esa región, minimizará los costes ya que las comunicaciones entre los servicios de una misma región son más baratas o incluso gratuitas y por último pero no menos importante, cumplen la legislación en lo referente a los datos personales de los usuarios.

Las regiones AWS, estarán aisladas entre sí, y la transferencia de datos entre dos regiones se facturará por la salida y entrada de datos según las tarifas de Transferencia de datos en Internet. Como se ha dicho anteriormente, los clientes de AWS pueden elegir las regiones en las que se ubicará su información y servidores, lo que permite que los clientes puedan establecerse en las zonas geográficas según sus necesidades. Por ejemplo, un cliente de AWS elige implementar sus servicios de AWS en la región Asia Pacífico (Tokyo) y almacenar los contenidos en Japón, esto quiere decir que la información se encuentra en Japón a no ser que este cliente decida mover los datos.

Los clientes podrán replicar y realizar copias de seguridad de los contenidos en más de una región, pero estos contenidos AWS no se moverán o replicarán fuera de la región o regiones en las que se encuentre si este no lo permite. Para la realización de las réplicas de datos entre regiones se deberá solicitar y pagar dicho proceso.

Por último, un cliente externo podrá acceder a servicios de cualquier otra región, pero con ciertas limitaciones, como por ejemplo peor latencia.

Cada región, como podemos ver en la figura anterior, posee distintas ubicaciones distintas denominadas *zonas de disponibilidad*. Estas zonas están diseñadas por si otras zonas fallan en un momento determinado y para proporcionar conectividad de red económica de baja latencia con otras zonas de la misma región. Si un usuario pone recursos en zonas de disponibilidad distintas, podrá proteger su aplicación en caso de que se produzca un fallo en una de estas zonas.

2.2.4 Seguridad en AWS

La infraestructura global que ofrece AWS es segura y tiene ciertas características para proteger los datos del cliente en la nube. Estas características son:

- Tanto el acceso físico a los centros de datos de AWS como el acceso a la red de AWS se controla, supervisa y se analiza estrictamente.
- Se pueden gestionar credenciales de seguridad que permitan a los usuarios acceder a su cuenta de AWS mediante AWS Identity and Access Management (IAM).
- Puede crear permisos con control preciso para los recursos de AWS y aplicarlos a usuarios o a grupos de usuarios y también puede crear permisos de tipo ACL para los datos y cifrado de datos inactivos.
- Puede configurar una red privada virtual (VPC), red virtual aislada de forma lógica de otras redes virtuales y se puede controlar si la red se puede enrutar directamente hacia Internet.
- Puede controlar y configurar el sistema operativo del servidor virtual, además de poder configurar un grupo de seguridad actuando como firewall virtual para controlar el tráfico entrante y saliente de los servidores virtuales.
- Se pueden especificar un par de claves al lanzar el servidor virtual, utilizado para cifrar la información de inicio de sesión.

2.2.5 Productos de AWS

Los servicios que proporciona AWS se distribuyen en las siguientes categorías:

Servicios de informática y redes: Puede proporcionar servidores virtuales, configurar un firewall, acceso a Internet, asignar y enrutar direcciones IP y escalar la infraestructura para satisfacer la demanda del cliente y las necesidades que éste tiene para sus aplicaciones. Estos servicios se complementarán para proporcionar una solución completa de informática, procesamiento de consultas y almacenamiento de las aplicaciones. Algunos de los servicios de esta categoría son Amazon Route 53, Amazon Virtual Private Cloud y AWS Direct Connect.

Servicios de Almacenamiento y Entrega de Contenidos (CDN): Posee una gran variedad de productos para que el usuario pueda satisfacer todas las necesidades que pueda tener en cuanto a almacenamiento de sus datos. Los servicios de Almacenamiento y Entrega de contenido que son Amazon S3 (más adelante se explica de manera más específica este servicio), Amazon EBS (Elastic Block Store), Volúmenes de Almacén de Instancias, Amazon CloudFront, AWS Import/Export, AWS Storage Gateway y Amazon Glacier.

Servicios de Base de datos: AWS ofrece servicios de bases de datos NoSQL y relacionales totalmente gestionados, además de un almacenamiento de caché en memoria como una solución de almacén de datos de mucha capacidad y como servicio. Algunos servicios que ofrece AWS para bases de datos serán Amazon RDS, Amazon DynamoDB, Amazon ElastiCache y base de datos alojada en Amazon EC2.

Servicios de Análisis para AWS: Se ofrecen herramientas de análisis escalables eficaces y económicas hasta almacenes de datos muy grandes. Amazon EMR, AWS Data Pipeline y Amazon Kinesis serán los servicios de esta categoría.

Servicios de Aplicaciones: Son aquellos servicios gestionados para su uso con aplicaciones, incluyendo servicios de transmisión, puesta en cola, notificaciones de inserción, transcodificación y entrega de correo electrónico. Destacaremos Amazon SQS, Amazon SES y AppStream de los servicios que oferta para esta categoría.

Servicios de Implementación: Estos servicios simplifican la creación de una solución en AWS si tener que gestionar los servicios por separado. AWS CloudFormation y AWS OpsWorks forman parte, entre otros, de esta categoría.

Servicios de Gestión: Estos servicios ayudarán a gestionar los recursos de AWS. Algunos de los servicios que ofrece AWS en esta categoría son AWS IAM, Amazon CloudWatch y AWS Directory Service.

En el apartado anterior se mostraban las categorías de servicios que ofrece AWS, en los siguientes apartados, se explicará más detenidamente los servicios más destacados que AWS tiene.

2.2.6 Amazon SQS

Amazon Simple Queue Service (SQS) es un sistema de colas de mensajes rápidos, escalable para almacenar los mensajes mientras que van de unos sistemas a otros, fiable y totalmente gestionado. Los desarrolladores podrán transferir datos entre componentes de aplicaciones que realicen distintas tareas sin perder mensajes.

Con Amazon SQS se puede desacoplar los componentes de una aplicación en la nube de manera sencilla y económica, además, se puede eliminar la carga administrativa que supone ejecutar y escalar un clúster de mensajería de gran disponibilidad.



Figura 9: Amazon SQS

Las ventajas que ofrece este servicio son:

Fiabilidad: Los mensajes se almacenan de forma redundante en varios centros de datos y servidores para que no se pierdan o dejen de estar disponibles en caso de fallo.

Escalabilidad: SQS está diseñado para permitir que un número ilimitado de servicios de mensajería puedan leer y escribir un número ilimitado de mensajes. Además, esta escalabilidad puede ser automática, según las necesidades del cliente.

Sencillez: Con una sola API los desarrolladores podrán comenzar a utilizar Amazon SQS, cuando un cliente envía una petición para realizar una tarea, ésta se procesa cuando una máquina quede libre.

2.2.7 Amazon S3

Amazon Simple Storage Service (S3) es un servicio de almacenamiento de objetos seguro, duradero, escalable, económico y fácil de utilizar pues posee una interfaz sencilla de servicios para almacenar y recuperar la información que se desee en cualquier momento desde cualquier lugar. Se ofrece una infraestructura igual a la utilizada por Amazon para el mantenimiento de su red mundial de sitios web.

Puede utilizarse tanto por separado como unido a otros servicios de Amazon como Amazon EC2, Amazon EBS y Amazon Glacier, además de repositorios y puertas de enlace de almacenamiento a terceros.

Algunos de los casos para utilización de Amazon S3 serán datos de aplicaciones en la nube, copias de seguridad y archivado, recuperación de desastres, distribución de contenidos y análisis de big data.



Figura 10: Amazon S3

Los datos se almacenan dentro de los recursos denominados “buckets”. El usuario puede almacenar todos los objetos que desee, además de poder realizar operaciones de lectura, escritura y eliminación de objetos almacenados, siendo éstos de hasta 5 terabytes.

El cliente tiene el control de acceso al bucket, ver los registros de acceso y a sus objetos además de elegir la región AWS en la que se almacene un bucket optimizando la latencia, minimizando los costos o abordando los requisitos normativos.

Las características de este servicio son:

Réplicas entre diferentes regiones: las réplicas de AWS entre distintas regiones son automáticas, fiables y rápidas. Cuando se carga un objeto en el bucket de S3 se replica automáticamente en otro bucket de otra región de AWS elegida por el usuario.

Notificaciones de eventos: Cuando se cargan o eliminan objetos en Amazon S3 puede activarse la opción de enviar una notificación al cliente por medio de otros servicios de AWS como por ejemplo Amazon SQS.

Control de versiones: El usuario puede habilitar con Amazon S3 el control de versiones, que le permite conservar, recuperar o restaurar todas las versiones de todos los objetos almacenados en un bucket.

Cifrado: La carga y descarga de objetos puede realizarse a través de puntos de enlace con cifrado SSL. Los datos inactivos pueden cifrarse de forma automática y con la opción de gestionar claves mediante Amazon Key Management Service (AWS KMS), de el propio Amazon S3 o del propio cliente.

Gestión de acceso y seguridad: Hay distintos mecanismos para el control y la supervisión de quién puede acceder a los datos, además de cómo, cuándo y dónde puede acceder a la información.

Gestión del ciclo de vida de los datos: Amazon S3 ofrece varias opciones como el archivado automatizado con Amazon Glacier, que es un servicio de bajo coste para realizar dicha tarea.

Acceso programático con los SDK de AWS: Los SDK (Software Development Kit) de AWS para Java, PHP, .NET, Python, Node.js, Ruby y el SDK para móviles de AWS admiten Amazon S3.

Supervisión y controles del coste: Cuando un cliente por ejemplo necesita un nuevo bucket para almacenar más información, Amazon S3 se encarga de ayudar a gestionar a este cliente a gestionar y controlar los costes para que el cliente esté enterado en todo momento de lo que está pagando por este nuevo bucket.

Opciones de almacenamiento flexibles: La durabilidad y disponibilidad de los objetos es de casi el 100%. Además se incluyen las opciones para los datos menos importantes y la opción de que con la ayuda de Amazon Glacier archivar los archivos de vital importancia a un precio asequible.

2.2.8 Amazon DynamoDB

Amazon DynamoDB es un servicio de datos NoSQL que ofrece un rendimiento rápido y predecible por ser un servicio escalable y totalmente gestionado. Los clientes pueden evitar así las cargas administrativas de las bases de datos distribuidas a AWS debido a que no tienen que preocuparse ni de las tareas de instalación de hardware, ni del mantenimiento de estas bases de datos.

Amazon DynamoDB gestionará el software de la base de datos y el aprovisionamiento del hardware necesario para ejecutar la base de datos, haciendo así que los usuarios puedan implementar su base de datos en minutos. Además realiza de forma automática particiones de forma constante de los datos, ofrece la capacidad adicional al servidor según sus necesidades y replica de forma sincronizada los datos a través de tres instalaciones en una región AWS ofreciendo así alta disponibilidad y durabilidad de los datos.

Características de Amazon DynamoDB:

Compatibilidad con modelos de datos de documentos: Mediante el uso de herramientas de desarrollo de AWS, se puede escribir en las aplicaciones que almacenen documentos JSON directamente en tablas de Amazon DynamoDB. Este servicio permite el almacenamiento, realización de consultas y actualización de documentos, reduciendo con esta característica código nuevo que inserte, actualice y recupere documentos JSON.

Compatibilidad con modelos de datos clave-valor: es compatible con estructuras de datos clave-valor, esto quiere decir, que cada elemento (fila), constituye una pareja clave-valor, donde el valor principal es el único atributo para los elementos de la tabla e identifica cada uno de ellos. Dado que DynamoDB no contiene esquemas, cada elemento puede poseer cualquier número de atributos (columnas) pudiendo ser posible la consulta de atributos no principales.

Escalado perfecto y alta disponibilidad: Como se ha descrito anteriormente, Amazon DynamoDB ofrece un rendimiento y escalado de almacenamiento perfectos mediante la API de la consola de AWS y con alta disponibilidad gracias a la replicación de datos automática y sincrónica de tres instalaciones incluidas en cada región, protegiendo los datos antes posibles fallos en las máquinas virtuales o instalaciones.

Transmisiones: Los cambios a nivel de elemento pueden mantenerse al día gracias a esta función u obtener todas las actualizaciones a nivel de elemento y usar los datos para crear aplicaciones creativas que faciliten la replicación, copias de seguridad, integración con otros servicios y vistas materializadas.

Replicación entre regiones: Es posible replicar las tablas de DynamoDB entre regiones de AWS de forma automática. Esta característica permite crear aplicaciones distribuidas a nivel mundial que disfrutan de menor latencia al acceder a datos, gestionar mejor el tráfico, migración de datos más sencillas y recuperación ante posibles desastres.

Activadores: Amazon DynamoDB se integra con AWS Lambda para que con estos activadores se puedan ejecutar automáticamente una función personalizada que detecte los cambios a nivel de elemento en una tabla de DynamoDB.

Sin esquemas: El esquema de base de datos con el que cuenta Amazon DynamoDB es flexible. Los elementos de datos de una tabla no necesitan tener los mismos atributos ni el mismo número de los mismos. Al existir varios tipos de datos este modelo de datos se ve enriquecido.

Coherencia alta y contadores atómicos: Amazon DynamoDB permite desarrollar de manera sencilla, ya que le permite realizar lecturas de coherencia alta con el fin de garantizar que se lean los valores más actualizados. También admite de forma nativa los contadores atómicos los cuales permiten aumentar o reducir los atributos numéricos utilizando una única solicitud a la API.

Supervisión integrada y seguridad: El servicio incluye Amazon CloudWatch que permite ver la latencia y el rendimiento de la solicitud de cada tabla de DynamoDB. Este servicio además utiliza métodos criptográficos para autenticar a los usuarios y prevenir el acceso a terceros.

Por último, Amazon DynamoDB en la consola de AWS permite al usuario crear, actualizar, eliminar y consultar tablas, ajustar el rendimiento y definir alarmas de manera rápida y sencilla, además ofrece una amplia funcionalidad con un número reducido de API para poder así centrarse en desarrollar las aplicaciones que se deseen.

2.2.9 Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) es el servicio web de Amazon que ofrece en la nube capacidad informática de tamaño modificable. La función de este servicio reside en facilitar a los desarrolladores recursos informáticos escalables y basados en la Web, pagando por su uso en cada instante. La interfaz que tiene este servicio es tan sencilla que permite obtener y configurar la capacidad de manera simple teniendo un control completo sobre los recursos informáticos del cliente.

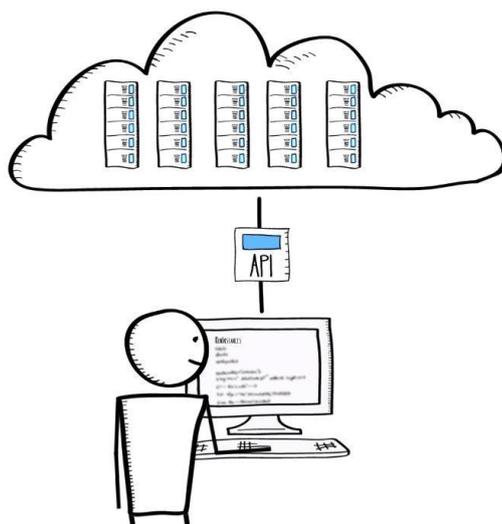


Figura 11: Amazon EC2

Amazon EC2 permite el escalado de la capacidad de forma rápida según las necesidades del cliente debido a que se reduce el tiempo necesario para obtener y arrancar nuevas instancias de servidor en minutos. Además, proporciona a los desarrolladores las herramientas necesarias para crear aplicaciones resistentes a errores y para aislarse de los fallos más comunes.

Este servicio de AWS cuenta con un entorno informático virtual, en el que se pueden utilizar interfaces de servicio web para iniciar instancias con distintos sistemas operativos, mantener y gestionar los permisos de acceso a la red, cargar las instancias con su propio entorno de aplicaciones y ejecutar esta misma imagen utilizando los sistemas que el usuario desee.

Necesitaremos para ejecutar Amazon EC2 una imagen de máquina de Amazon (AMI), configuración de la seguridad y el acceso de red en la instancia, elegir el tipo o los tipos de instancias que se van a utilizar a través de las API o de las herramientas de gestión que se ofrecen, determinar si se desea una ejecución en varias ubicaciones y por último, pagar sólo por los recursos que realmente se consumen, como las horas de uso de las instancias o la transferencia de datos.

Amazon EC2 posee una serie de características para compilar las aplicaciones escalables, eficaces ante fallos y de tipo empresarial, a continuación, se muestran estas características:

Auto Scaling: Permite escalar automáticamente la capacidad de Amazon EC2, con seguridad de que el número de instancias que se estén utilizando aumenten sin interrupciones si incrementa la demanda con el fin de mantener el rendimiento, o se reduzca en caso de que esta demanda ceda para minimizar los costos.

Varias Ubicaciones: Permite la colocación de instancias en distintas ubicaciones, donde si éstas instancias se ubican en distintas “zonas de disponibilidad”, se pueden proteger las aplicaciones en caso de error de una única ubicación.

Amazon Elastic Block Store (EBS) Amazon EBS ofrece almacenamiento persistente para las instancias de Amazon EC2. Los volúmenes de Amazon EBS, que son fiables y con una gran disponibilidad, se adjuntan a redes y persisten con independencia de la vida de una instancia.

Direcciones IP elásticas: Éstas direcciones son direcciones IP estáticas diseñadas para la informática dinámica en la nube. Cada dirección IP elástica está asociada a su cuenta, no a una instancia, por lo que se puede controlar dicha IP hasta que se decida liberarla.

Amazon Virtual Private Cloud: Amazon VPC aprovisiona una sección aislada de forma lógica de la nube de AWS, donde puede lanzar recursos de AWS en una red virtual que defina.

Amazon CloudWatch: Servicio web que supervisa las aplicaciones y los recursos en la nube de AWS, permite visualizar la utilización de los recursos, el funcionamiento operativo y los patrones de demanda general además de otras tantas funciones.

Elastic Load Balancing: Distribuye automáticamente el tráfico entrante de las aplicaciones entre varias instancias de Amazon EC2, además consigue aún más tolerancia a fallos en sus aplicaciones ya que proporciona la capacidad de equilibrio de carga necesaria como respuesta al tráfico entrante de aplicaciones.

Clústeres de Informática de alto rendimiento (HPC): Estos clústeres han sido diseñados para proporcionar una funcionalidad de red de alto rendimiento permitiendo a las aplicaciones alcanzar el rendimiento de red o baja latencia necesario para la comunicación de nodo a nodo. Adecuados para los clientes con cargas de trabajo informáticas complejas los cuales se beneficiarán de la elasticidad, flexibilidad y del precio de Amazon EC2.

Instancias de GPU: Para los clientes que necesiten una alta capacidad de rendimiento en paralelo se beneficiarán de las instancias GPU (Unidad de Procesamiento Gráfico). Estas instancias resultan ideales para aplicaciones de gráficos en 3D y cargas de trabajo informáticas, la creación de modelos financieros y el diseño para la ingeniería.

AWS Marketplace: Tienda en línea que ayuda a buscar, comprar e implementar rápidamente el software que se ejecuta en AWS. De esta forma, solo se le cobrará por lo que se use, ya sea por hora o por mes.

Redes mejoradas: Las “Enhanced Networking” permiten un rendimiento de paquetes por segundo (PPS) significativamente superior, una reducción del ruido de red y latencias menores. Mejora el rendimiento de E/S y reduce el uso de CPU en comparación con las implementaciones tradicionales.

FUNCIONAMIENTO DE AMAZON WEB SERVICES

3.1 CREACIÓN DE UNA INSTANCIA EN AMAZON EC2

En apartados anteriores, se ha hablado del servicio de Amazon Web Services llamado Amazon Elastic Compute Cloud (Amazon EC2) que provee a sus usuarios de máquinas virtuales dedicadas de múltiples tamaños y de coste mínimo.

En esta parte del proyecto se mostrará cómo poder crear una instancia de servidor dedicado con una carga pequeña y utilizar el servicio sin coste en Amazon EC2.

3.1.1 Consola de AWS

En primer lugar, realizaremos los pasos necesarios para conseguir una cuenta en Amazon Web Services, como se ha dicho anteriormente, se puede disponer de una prueba gratuita de uso de los servicios de AWS durante un año. Una vez nos se haya creado la cuenta, accedemos a la consola de Amazon.

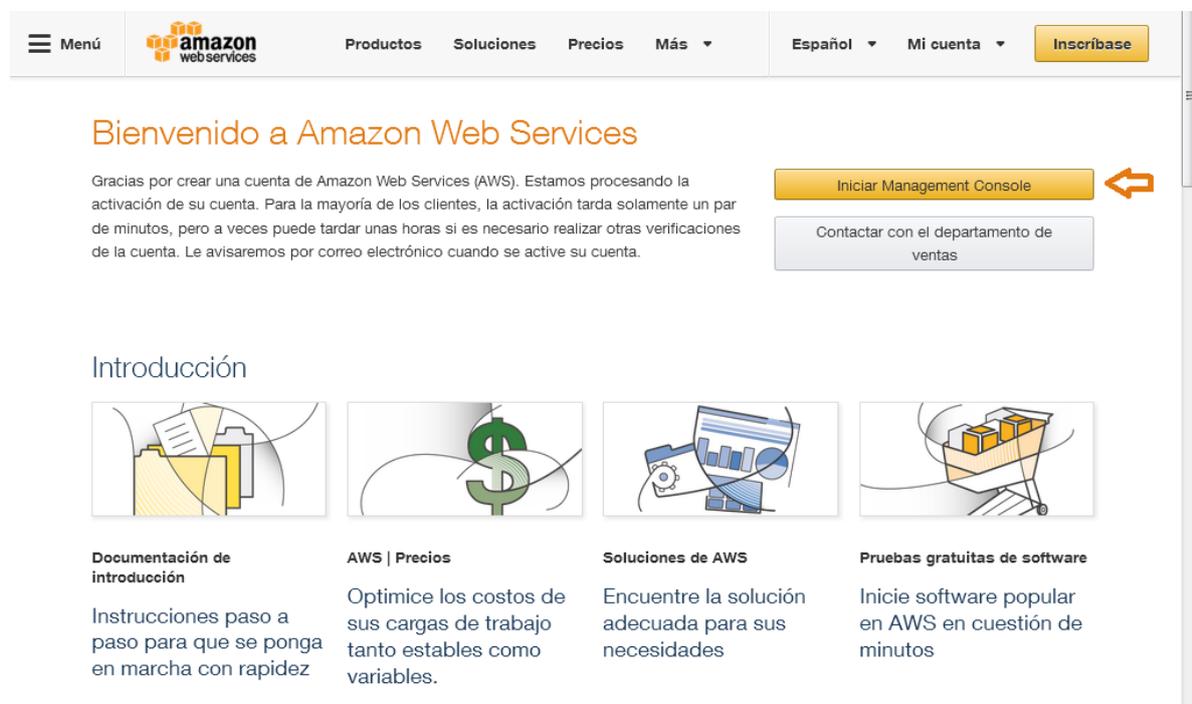


Figura 12: Iniciar Consola de Amazon

Una vez que accedemos a la consola de Amazon, podemos ver toda la lista de servicios organizados por su correspondiente categoría, es decir, los servicios pertenecientes por ejemplo a bases de datos, se encontrarán en la sección "Database".

El aspecto de la consola de Amazon, será el de la siguiente figura, una vez visualizado todos los servicios que nos puede ofrecer AWS, haremos click en EC2, “virtual servers in the cloud”.

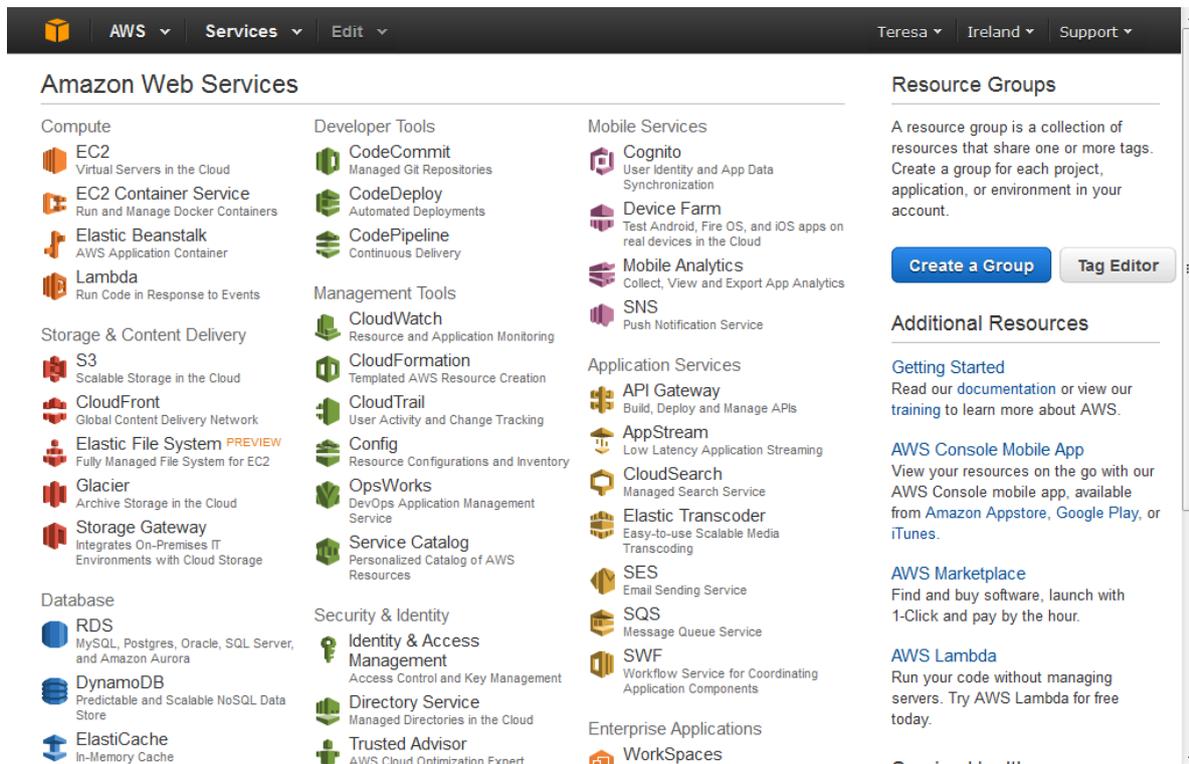


Figura 13: Consola de Amazon

Otro paso previo para realizar una instancia en EC2, será la de elegir en qué zona del mundo se van a desplegar las máquinas del usuario. Posee una gran utilidad si el servicio que se va a ofrecer se realiza en una ubicación del mundo concreta, ya que puede incrementar la latencia del cliente si éste se encuentra a mucha distancia del servidor. En este caso, se ha escogido la zona de Irlanda para poseer cobertura en toda Europa.

3.1.2 Creación de Instancia de Máquina Virtual

Una vez realizados los dos pasos del apartado anterior, se pasa a la creación de la instancia en Amazon EC2. Para la creación de la Instancia de EC2, pulsaremos el botón “Launch Instance” en el apartado “Create Instance” pulsando en el panel de navegación “EC2 Dashboard”:

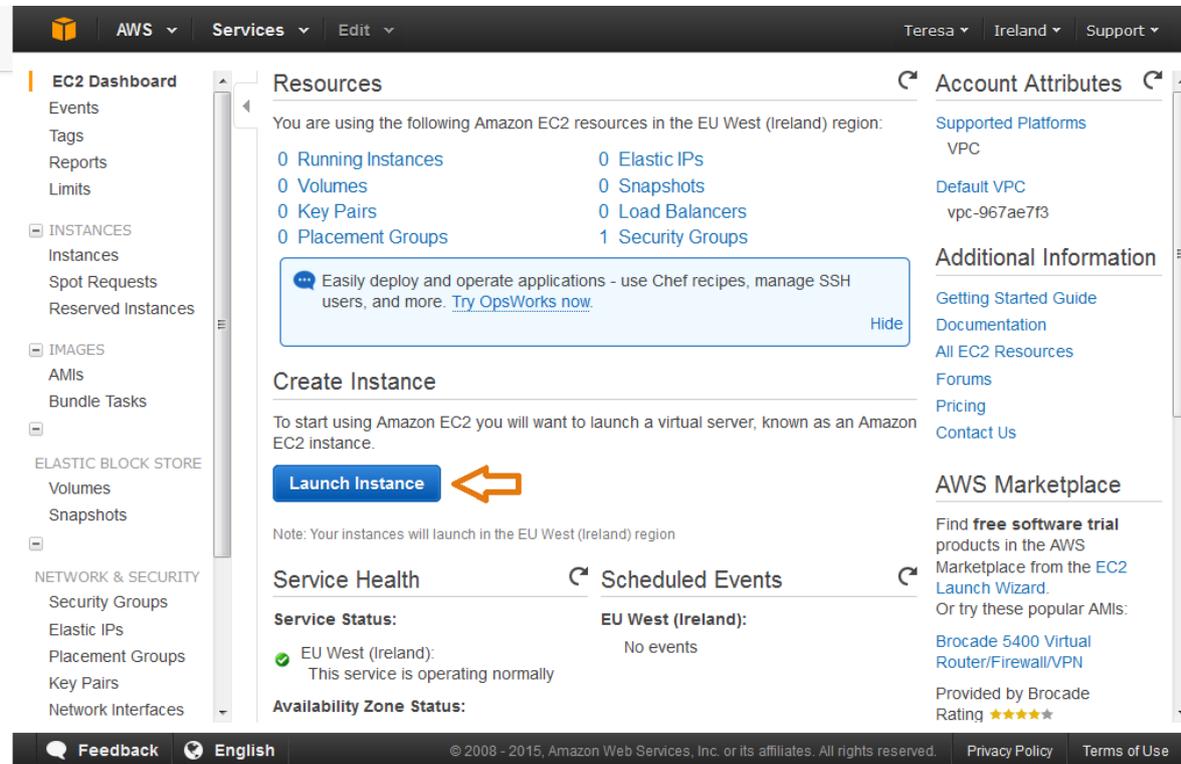


Figura 14: Lanzamiento de la Instancia

En este momento, nos aparecerá en la pantalla una selección de distintas AMI's en las que se muestran datos como el sistema operativo, tipo de procesador y aplicaciones. Hay multitud de AMI's por elegir, si pulsamos en la izquierda de la pantalla "Free Tier Only" nos mostrará sólo las AMI's que se pueden escoger de coste gratuito. Elegimos "Amazon Linux AMI" y pulsamos en "Select":

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 1: Choose an Amazon Machine Image (AMI) Cancel and Exit

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Quick Start 1 to 22 of 22 AMIs

- My AMIs**
- AWS Marketplace**
- Community AMIs**
- Free tier only** ⓘ

Amazon Linux Free tier eligible	Amazon Linux AMI 2015.03 (HVM), SSD Volume Type - ami-a10897d6 The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages. Root device type: ebs Virtualization type: hvm	Select 64-bit
Red Hat Free tier eligible	Red Hat Enterprise Linux 7.1 (HVM), SSD Volume Type - ami-25158352 Red Hat Enterprise Linux version 7.1 (HVM), EBS General Purpose (SSD) Volume Type Root device type: ebs Virtualization type: hvm	Select 64-bit
SUSE Linux Free tier eligible	SUSE Linux Enterprise Server 12 (HVM), SSD Volume Type - ami-e801af9f SUSE Linux Enterprise Server 12 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud. Advanced Systems Management. Web and Scripting. and	Select 64-bit

Feedback English © 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Figura 15: Selección de AMI

Como se puede apreciar, la selección de parámetros para nuestra instancia es bastante intuitiva y sencilla de seguir. Una vez seleccionado la AMI “Amazon Linux”, pasaremos al paso número dos de este proceso, que es elegir el tipo de instancia que vamos a utilizar, en este caso, escogemos la única que está disponible de manera gratuita, una instancia t2.micro con las características que se pueden ver en la siguiente imagen:

Step 2: Choose an Instance Type
 Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance types Current generation [Show/Hide Columns](#)

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	m4.large	2	8	EBS only	Yes	Moderate

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Instance Details](#)

Figura 16: Tipo de Instancia

Tras este paso, podemos revisar y lanzar ya nuestra instancia o continuar configurando detalles de ésta, si es la primera vez que se maneja Amazon Web Services, lo ideal es continuar configurando los detalles de nuestra instancia. Así pues, pulsamos hacia el siguiente paso del proceso de creación.

El tercer paso a seguir será como se ha mencionado antes, la configuración detallada de la instancia, es muy importante establecer como mínimo una instancia, aunque se pueden establecer todas las que se quieran para una misma AMI, además este paso posee diferentes opciones como la solicitud al contado de las instancias para tener ventajas a la hora de aprovecharse en los precios que serán más bajos, asignar gestión en la instancia, entre otros.

The screenshot displays the 'Step 3: Configure Instance Details' page in the AWS Management Console. The page is divided into several sections, each with a configuration option and a help icon (i). The options are:

- Number of instances:** A text input field containing the number '1'.
- Purchasing option:** A checkbox labeled 'Request Spot Instances' which is currently unchecked.
- Network:** A dropdown menu showing 'vpc-967ae7f3 (172.31.0.0/16) (default)' with a 'Create new VPC' link.
- Subnet:** A dropdown menu showing 'No preference (default subnet in any Availability Zone)' with a 'Create new subnet' link.
- Auto-assign Public IP:** A dropdown menu showing 'Use subnet setting (Enable)'.
- Domain join directory:** A dropdown menu showing 'None' with a 'Create new directory' link.
- IAM role:** A dropdown menu showing 'None' with a 'Create new IAM role' link.
- Shutdown behavior:** A dropdown menu showing 'Stop'.
- Enable termination protection:** A checkbox labeled 'Protect against accidental termination' which is currently unchecked.

At the bottom of the configuration area, there are four buttons: 'Cancel', 'Previous', 'Review and Launch' (highlighted in blue), and 'Next: Add Storage'.

The footer of the console shows 'Feedback', 'English', '© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.', 'Privacy Policy', and 'Terms of Use'.

Figure 17: Detalles en la configuración de la instancia

En el paso siguiente, nos encargaremos de abastecer a nuestra instancia del tamaño de almacenamiento que queremos, el usuario podrá adjuntar volúmenes adicionales EBS y volúmenes de almacenen de instancias para las instancias o editar los ajustes de en el volumen raíz (root volume).

The screenshot displays the 'Step 4: Add Storage' configuration page in the AWS Management Console. At the top, a navigation bar shows the current step '4. Add Storage' highlighted among seven steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Tag Instance, 6. Configure Security Group, and 7. Review. Below the navigation bar, the page title 'Step 4: Add Storage' is followed by a descriptive paragraph: 'Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.'

The main configuration area contains a table with the following columns: Type, Device, Snapshot, Size (GiB), Volume Type, IOPS, Delete on Termination, and Encrypted. The table lists one volume: 'Root' with device '/dev/sda1', snapshot 'snap-4f91fdb8', size '30', volume type 'General Purpose (SSD)', IOPS '90 / 3000', 'Delete on Termination' checked, and 'Encrypted' set to 'Not Encrypted'. Below the table is an 'Add New Volume' button. A light blue information box states: 'Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.'

At the bottom of the configuration area, there are navigation buttons: 'Cancel', 'Previous', 'Review and Launch' (highlighted in blue), and 'Next: Tag Instance'. The footer of the console includes 'Feedback', 'English', copyright information '© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.', 'Privacy Policy', and 'Terms of Use'.

Figura 18: Agregar Almacenamiento

Para usuarios con cuenta de prueba el máximo almacenamiento que podrán escoger será de 30 GiB, en este caso escogeremos un tamaño de 8GiB.

A continuación, procedemos al quinto paso de la creación de nuestra instancia, en el que se añaden etiquetas al servidor para poder distinguirlo en el futuro. Si tenemos una sola instancia no existirá problema pero en el caso de que se manejen 50 es importante poder distinguir qué hace cada una de ellas. Cada etiqueta, está formada por una clave (key) y por un valor (value) que queramos:

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 5: Tag Instance

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. [Learn more](#) about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)
Teresa	TeresaServer

Create Tag (Up to 10 tags maximum)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Security Group](#)

[Feedback](#) [English](#) © 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Figura 19: Etiquetas de la Instancia

El número máximo de etiquetas a crear para una instancia será de 10, si el usuario no sabe qué nombres poner en dichas etiquetas, se puede observar como en la descripción de este paso nos dan un ejemplo para la clave y el valor de ésta.

El sexto paso de este proceso será la configuración de un grupo de seguridad, en el caso de que no se cree un grupo de seguridad, siempre se podrá configurar más adelante. La seguridad en la nube es crucial para que los usuarios puedan confiar en dejar su información en cualquier proveedor de Cloud Computing, por ello, AWS nos da a elegir si crear un nuevo grupo de seguridad o escoger uno de los ya existentes.

The screenshot displays the AWS Management Console interface for configuring a security group. At the top, there are navigation tabs for '1. Choose AMI', '2. Choose Instance Type', '3. Configure Instance', '4. Add Storage', '5. Tag Instance', '6. Configure Security Group', and '7. Review'. The main heading is 'Step 6: Configure Security Group'. Below this, there is a description of a security group and instructions on how to assign one. The 'Assign a security group' section has two radio buttons: 'Create a new security group' (selected) and 'Select an existing security group'. The 'Security group name' field contains 'launch-wizard-1' and the 'Description' field contains 'launch-wizard-1 created 2015-08-24T12:16:28.073+02:00'. A table for adding rules is shown with columns for 'Type', 'Protocol', 'Port Range', and 'Source'. The first rule is for 'RDP' using 'TCP' on port '3389' with a source of 'Anywhere' and '0.0.0.0/0'. An 'Add Rule' button is below the table. A yellow warning box contains a warning icon and text: 'Warning: Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Review and Launch'.

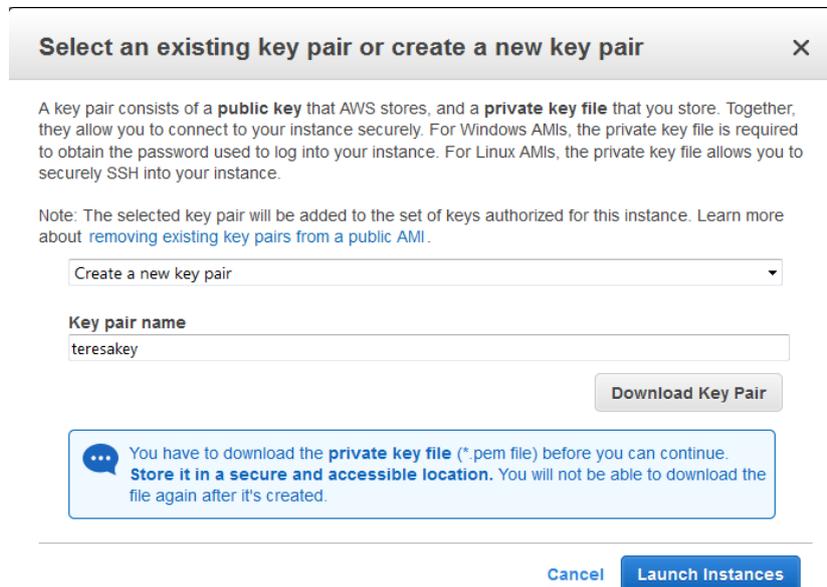
Figura 20: Configuración del Grupo de Seguridad

En la configuración del grupo de seguridad se pueden agregar reglas para permitir que el tráfico que el usuario quiera llegue a la instancia. Si un usuario desea configurar un servidor web y permitir que el tráfico de Internet para llegar a su instancia, deberá añadir reglas que permiten el acceso sin restricciones a los puertos HTTP y HTTPS.

Como se observa en la figura anterior, si ponemos como regla una source de 0.0.0.0 quiere decir que se permite que todas las direcciones IP podrán acceder a la instancia.

Uno de los últimos pasos de la creación de nuestra instancia en EC2 será el resumen de todas las opciones que hemos seleccionado anteriormente. Una vez revisado todo, sólo bastará con pulsar en "Launch" para lanzar nuestra instancia.

Una vez pulsado "Launch", Amazon Web Service muestra una pequeña ventana en la que tendremos que crear un par de claves RSA para poder acceder al servidor una vez lo hayamos arrancado. Después de crear el par de claves nos la descargamos ya que las usaremos para acceder al terminal Linux por SSH. La clave pública se sitúa dentro del servidor en /root/.ssh/authorized_keys.



The screenshot shows a dialog box titled "Select an existing key pair or create a new key pair" with a close button (X) in the top right corner. Below the title, there is explanatory text: "A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance." Below this is a note: "Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#)." There is a dropdown menu currently set to "Create a new key pair". Below that is a text input field for "Key pair name" containing the text "teresakey". To the right of the input field is a "Download Key Pair" button. At the bottom of the dialog, there is a blue informational box with a speech bubble icon containing the text: "You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created." At the very bottom of the dialog, there are two buttons: "Cancel" and "Launch Instances".

Figura 21: Creación del par de claves RSA

3.1.3 Configuración de red del servidor creado

Una vez lanzada nuestra instancia, pulsamos en el panel de navegación sobre “Instancias”, se puede observar que en la lista de máquinas virtuales que se está configurando la nueva instancia que hemos creado. El tiempo en el que será accesible poder conectarse a nuestra instancia será de un par de minutos.

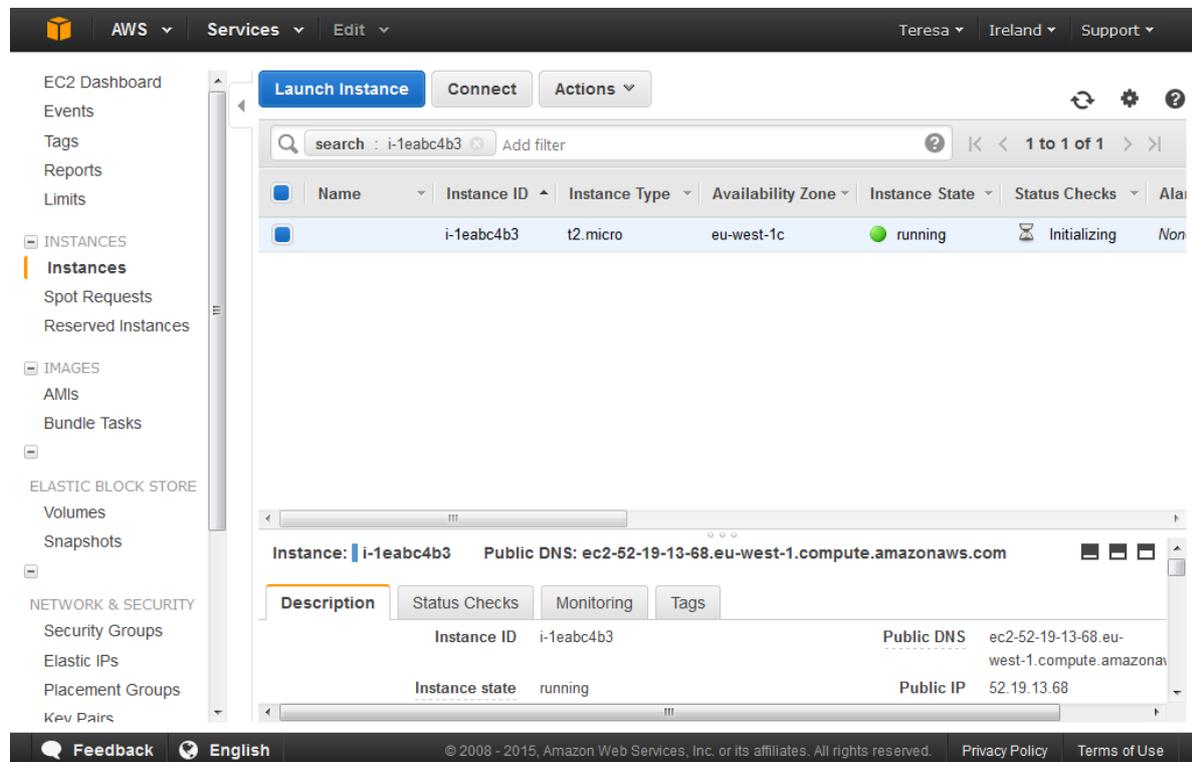


Figura 22: Aspecto de la “Lista de instancias”

Cuando nuestra instancia ya está en el estado “running” podremos conectarnos a nuestra instancia. Además podemos ver los detalles de nuestra instancia situados en la parte inferior de la lista seleccionando la instancia deseada. Existen múltiples maneras para conectarse a la instancia, que se verán en los próximos apartados.

3.1.4 Eliminación de instancias

Tenemos una instancia pero queremos eliminar y en nuestra lista de instancias no tenemos ningún botón que elimina dicha instancia. Al eliminar una instancia es imposible volver a conectarse a ella, si lo que queremos es detener dicha instancia, ésta se apagará y no se cargarán costes por uso, aunque sí por los volúmenes EBS si el usuario se encuentra ya fuera del periodo de prueba.

El proceso para finalizar una instancia es el siguiente, se accede a la lista de instancias EC2 que tenemos en nuestra consola de Amazon Web Services, a continuación seleccionamos la instancia a eliminar y pulsamos botón derecho, una vez hecho esto, seleccionamos “Terminate” que se encuentra en “Instance State”. Amazon EC2 procederá entonces a la eliminación de dicha instancia y una vez que la instancia esté en el estado “Terminate” sólo tendremos que actualizar la lista de nuestras instancias para que no aparezca.

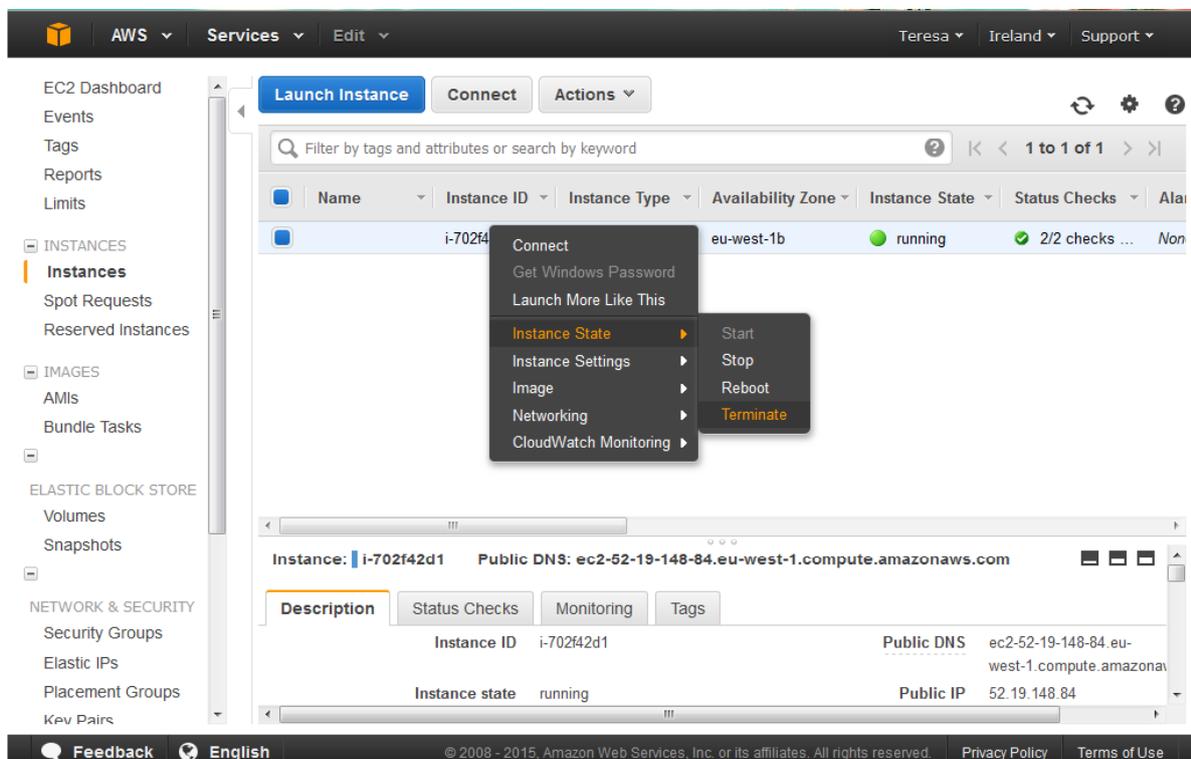


Figura 23: Eliminación de una instancia

El hecho de que se elimine una instancia no hace que se elimine el volumen de Amazon EBS que se tenía de esa instancia. Para eliminar el volumen de almacenamiento simplemente tendremos que pulsar “Volumes” en el panel de navegación, escoger el volumen de la instancia que hemos eliminar y pulsar con el botón derecho y a continuación en “Delete” y confirmar que se desea eliminar.

3.2 CONEXIÓN A UNA INSTANCIA EN AMAZON EC2

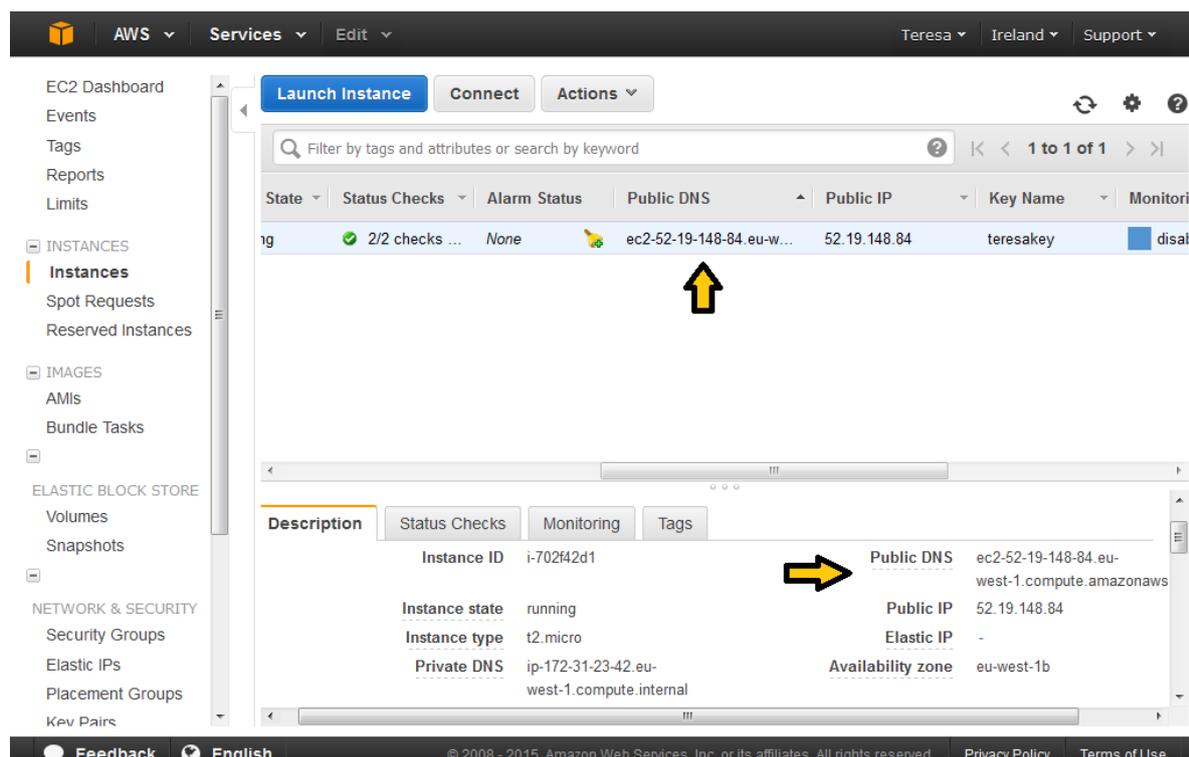
Una vez creada la instancia EC2 que ayudará a cualquier usuario a ofrecer sus propios servicios, llega el momento de conectarse a esta instancia. En esta parte del proyecto se explicarán los pasos que hay que realizar para conectarse a nuestra instancia, así como las diferentes formas que existen para realizar esta conexión.

3.2.1 Pasos Previos a la conexión de la Instancia

Antes de conectarse a la instancia hay que realizar varias tareas que vienen explicadas en el presente apartado.

- Obtener la DNS pública de la instancia:

Con la DNS pública de la instancia podremos acceder a nuestro servidor desde cualquier navegador web. Para obtener la DNS de nuestra instancia, solo tendremos que seleccionar dicha instancia estando en la consola AWS y buscar la DNS en la columna de la instancia, o en caso de no encontrarla buscarla en el panel de “descripción” que se encuentra en la parte inferior de la consola.



The screenshot shows the AWS Management Console interface. At the top, there are navigation tabs for 'Launch Instance', 'Connect', and 'Actions'. Below this is a search bar and a table of instances. The table has columns for State, Status Checks, Alarm Status, Public DNS, Public IP, Key Name, and Monitoring. One instance is listed with the following details:

State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Monitoring
Running	2/2 checks ...	None	ec2-52-19-148-84.eu-w...	52.19.148.84	teresakey	disabled

Below the table, the 'Description' tab is selected, showing the following details for instance ID 'i-702f42d1':

Instance ID	i-702f42d1	Public DNS	ec2-52-19-148-84.eu-west-1.compute.amazonaws.com
Instance state	running	Public IP	52.19.148.84
Instance type	t2.micro	Elastic IP	-
Private DNS	ip-172-31-23-42.eu-west-1.compute.internal	Availability zone	eu-west-1b

Figura 24: DNS Pública de la Instancia

- Obtención de la clave privada:

En el apartado de creación de la instancia, tuvimos que crear un par de claves RSA para acceder al servidor una vez arrancado. Estas claves se descargan en un archivo .pem que tendremos que tener localizado para cuando vayamos a conectarnos a la instancia.

- Habilitar tráfico SSH entrante desde la IP del usuario para la instancia:

Para poder realizar este paso, tendremos que asegurarnos de que el grupo de seguridad que se creó en el momento en el que se creó la instancia permita este tipo de tráfico, como por defecto esta opción en el grupo de seguridad está desactivada, procederemos a activarla añadiendo una regla al grupo de seguridad. Para añadir esta regla deberemos realizar los siguientes pasos:

En primer lugar y estando en la lista de instancias, seleccionaremos nuestra instancia y en la parte inferior en la que sale la descripción de nuestra instancia, buscaremos los grupos de seguridad que tiene ésta instancia. Si pulsamos en “view rules” nos mostrarán las reglas que tiene nuestra instancia.

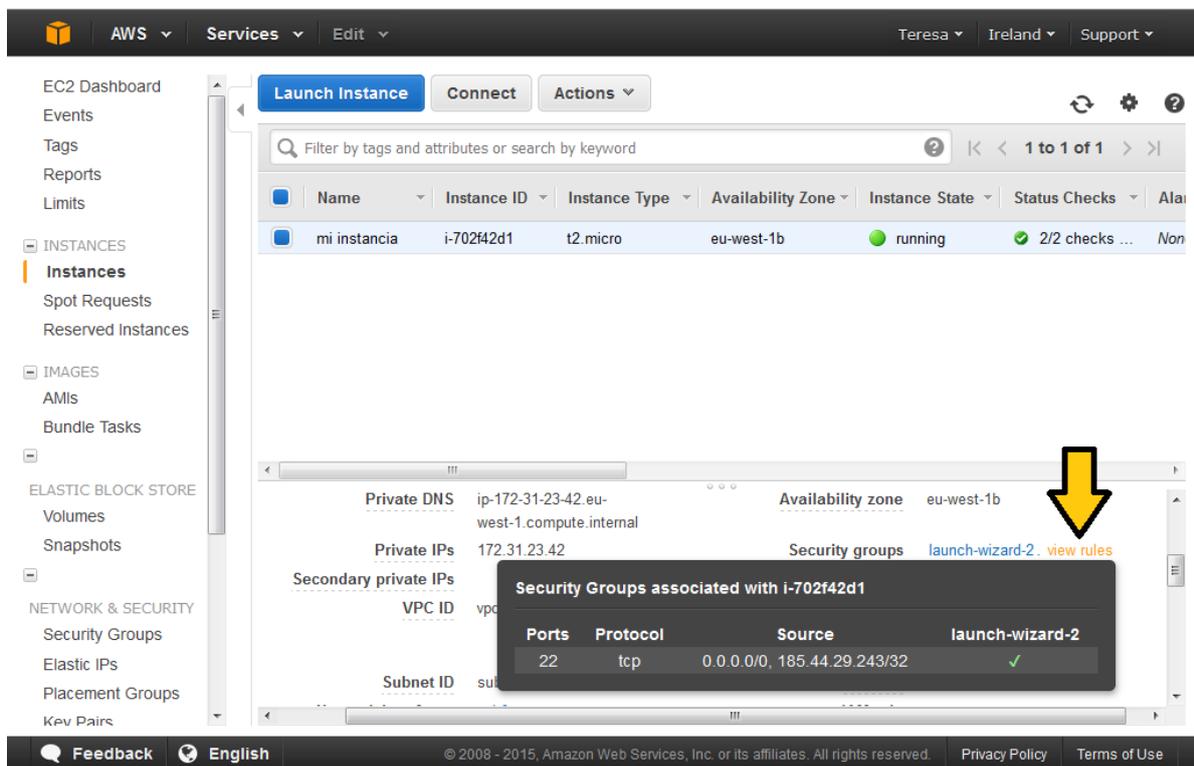


Figura 25 : Vista de las reglas de nuestra instancia

A continuación, pulsaremos en el panel de navegación “Security Groups” para ver los grupos de seguridad creados, y seleccione el grupo de seguridad de la instancia que hemos creado.

Una vez escogido el grupo de seguridad, pulsaremos en la pestaña “Inbound” situado en el panel inferior donde se muestran los detalles del grupo de seguridad. En esta pestaña podemos ver las reglas que el grupo de seguridad permite, pulsaremos en “Edit” para añadir la nueva regla.

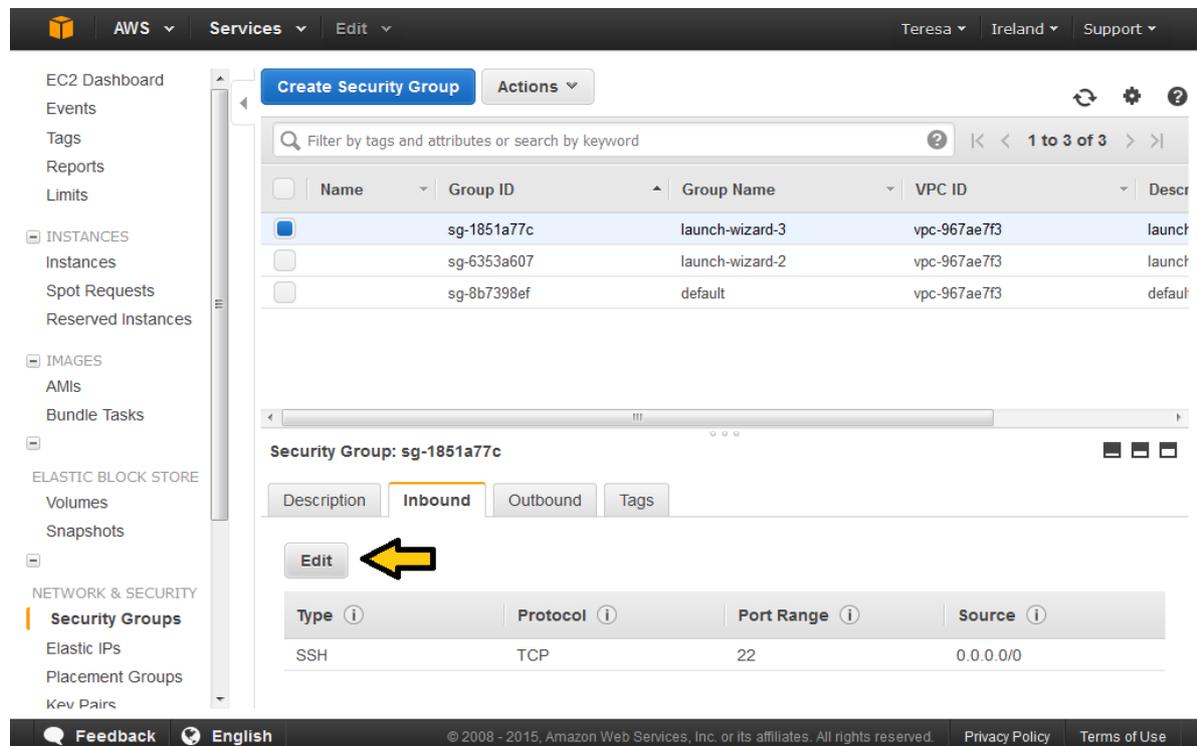


Figura 26: Grupos de seguridad

En el cuadro de diálogo que aparecerá tras pulsar Edit, pulsaremos “Add rule” y a continuación, seleccionaremos como tipo de tráfico SSH y en el campo origen seleccionaremos “My IP”, la IP que se carga en el cuadro anexo será la IP pública del equipo del usuario, también existe la posibilidad de poner cualquier IP pública ya sea de otro dispositivo o bien un rango de IP’s públicas (lo cual es bastante útil para empresas).

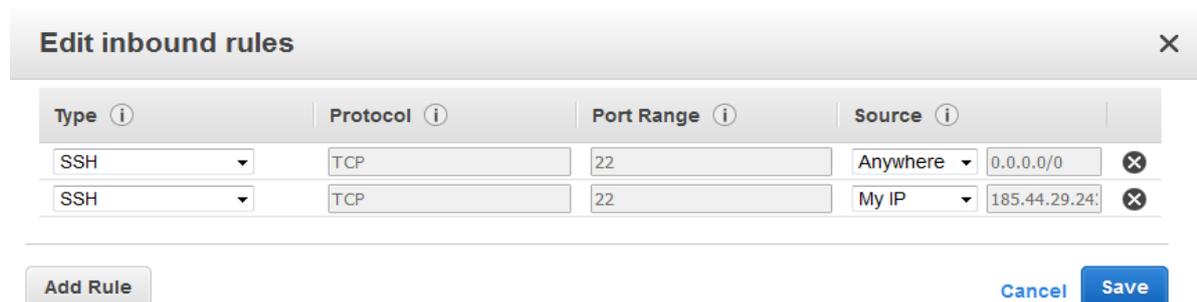


Figura 27: Agregar nueva regla en un grupo de seguridad

Sólo queda pulsar “Save” para agregar esta nueva regla. Las modificaciones realizadas en los grupos de seguridad se aplican automáticamente en las instancias en las que tiene asignado este grupo de seguridad.

3.2.2 Métodos de conexión a una Instancia

Existen diferentes maneras de conectarse a una instancia Linux, en este apartado se explicará cómo acceder a la instancia que se ha creado utilizando cualquiera de estos métodos y se escogerá el método que más convenga para conectar nuestra instancia.

- **Conexión desde un terminal MAC o Linux utilizando un cliente SSH**

Los computadores MAC o Linux suelen incluir por defecto un cliente SSH. Para comprobarlo sólo habrá que escribir en la línea de comandos “ssh”, en caso de que el ordenador no reconozca el comando introducido, “OpenSSH Project” proporciona una implementación libre y completa de herramientas SSH.

Para empezar a conectarse a la instancia de esta forma abriremos la línea de comandos, iremos al directorio donde se ubica el archivo de la clave privada que se creó al lanzar la instancia y ejecutaremos el comando “chmod” para asegurar que la clave privada sólo puede ser vista por el dueño de la clave:

```
$ chmod 400 my-key-pair.pem
```

Donde my-key-pair será el nombre que le pusimos al par de claves RSA que nos descargamos en pasos anteriores. Ahora, ejecutaremos el siguiente comando:

```
$ ssh -i /path/my-key-pair.pem ec2-user@ public_dns_name
```

Lo que estamos haciendo al ejecutar esta línea de comando es conectarnos a nuestra instancia, especificando nuestra clave privada con el archivo .pem y el nombre de usuario con el nombre de la DNS pública de la instancia. El nombre de usuario cambia dependiendo del Sistema Operativo que utilicemos:

Sistema Operativo	Nombre de usuario que utilizaremos
Amazon Linux	ec2-user
RHEL5	root ó ec2-user
Ubuntu	ubuntu
Fedora	fedora
SUSE Linux	root ó ec2-user

Tabla 1: Nombres de usuarios por Sistema Operativo

El sistema responderá preguntando si deseamos continuar la conexión (yes/no), Introduciremos “yes” en la línea de comandos y recibiremos respuesta de que el terminal ya ha sido añadido a la lista de host conocidos, y ya podremos trabajar con nuestra instancia.

- **Conexión de Windows a la Instancia utilizando PuTTY**

Para conectarse a la instancia de Linux desde Windows necesitaremos la herramienta PuTTY, el proceso para realizar dicha tarea, será el que viene a continuación:

En primer lugar el usuario debe descargar e instalar PuTTY en su página web en el apartado de descargas (Download), aquí se observa que dependiendo de las especificaciones del Sistema Operativo y de las necesidades que tenga el usuario puede descargarse múltiples ejecutables y archivos comprimidos. Escogemos el conjunto de PuTTY que contiene todas las herramientas que vamos a necesitar.

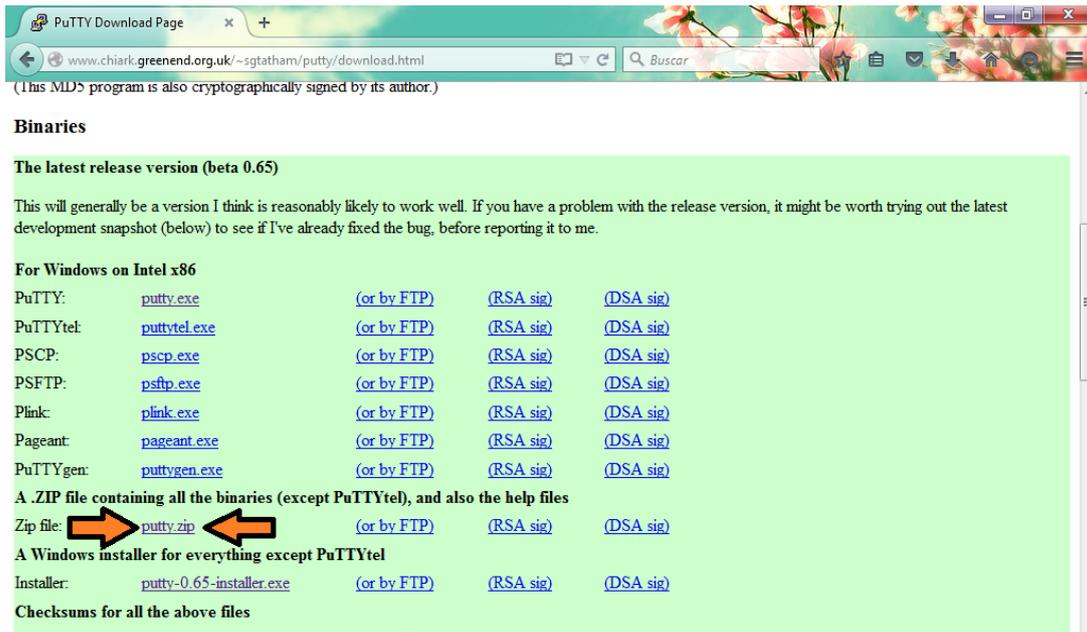


Figura 28: Descarga de PuTTY

Para poder conectarse con PuTTY la clave privada tendrá que ir en el formato .ppk, una vez instalado el conjunto de herramientas que nos hemos descargado, procedemos a convertir el archivo de nuestra clave privada .pem a .ppk, para ello, abriremos la herramienta PuTTYgen. Nos saldrá el siguiente cuadro de diálogo en el que seleccionaremos la opción SSH-2 RSA y pulsamos en "Load" para cargar nuestro archivo .pem.

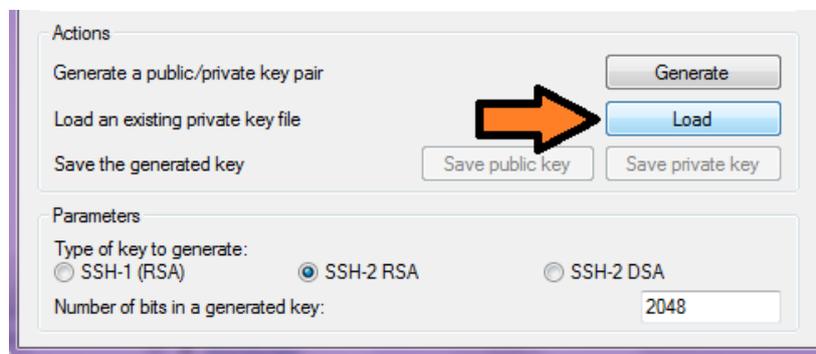


Figura 29: Carga archivo .pem en PuTTYgen

Para buscar nuestra clave privada .pem, seleccionaremos ver todos los tipos de archivos ya que por defecto PuTTYgen muestra sólo los archivos .ppk, una vez encontrado el archivo .pem haremos clic en abrir, aparecerá entonces un cuadro de diálogo en el que nos dirá que se ha generado el archivo .ppk con éxito y tendremos que dar a aceptar.

Ahora sólo hace falta guardar nuestra clave privada .pem convertida en .ppk para que cuando vayamos a utilizar PuTTY pueda utilizarla sin ningún problema.

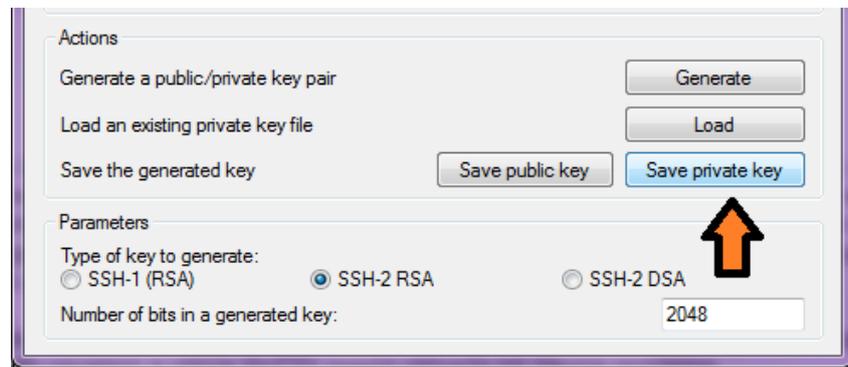


Figura 30: Clave privada .ppk generada

Iniciamos ya la herramienta PuTTY, y el usuario comenzará a configurar la conexión a la instancia. En el panel “Category”, seleccionaremos “Session” y se completan los siguientes campos: En “Host Name” introduciremos el nombre de usuario que es “ec2-user” seguido del nombre de la DNS pública de la instancia, en el tipo de conexión (Connection Type), seleccionaremos SSH y por último, observar que el número de puerto es el 22.

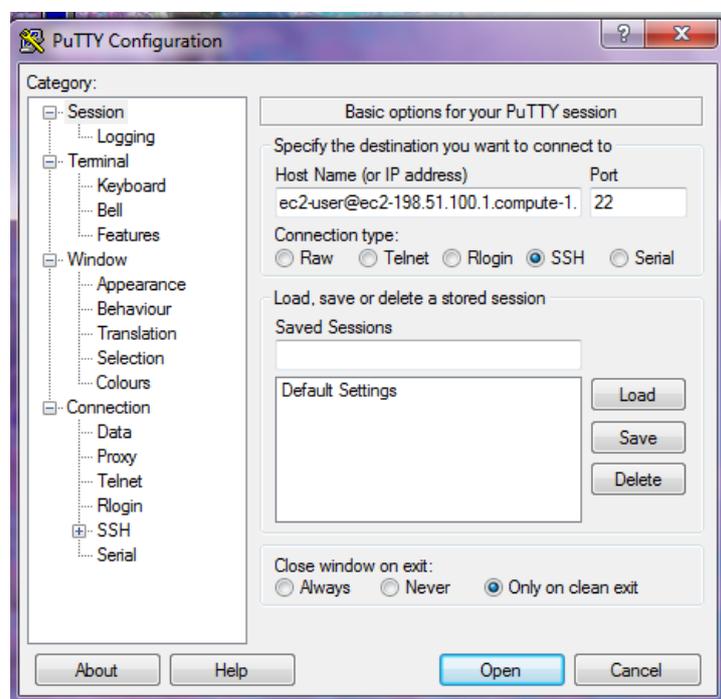


Figura 31: Configuración de la Sesión PuTTY

Una vez realizadas estas modificaciones, expandimos “SSH” que se encuentra en el apartado “Connection” del panel de categorías y pulsamos en “Auth”, en este cuadro de diálogo hacemos clic en “Browse” para seleccionar el archivo .ppk que generamos y a continuación, pulsamos en “Open” para iniciar sesión en PuTTY.

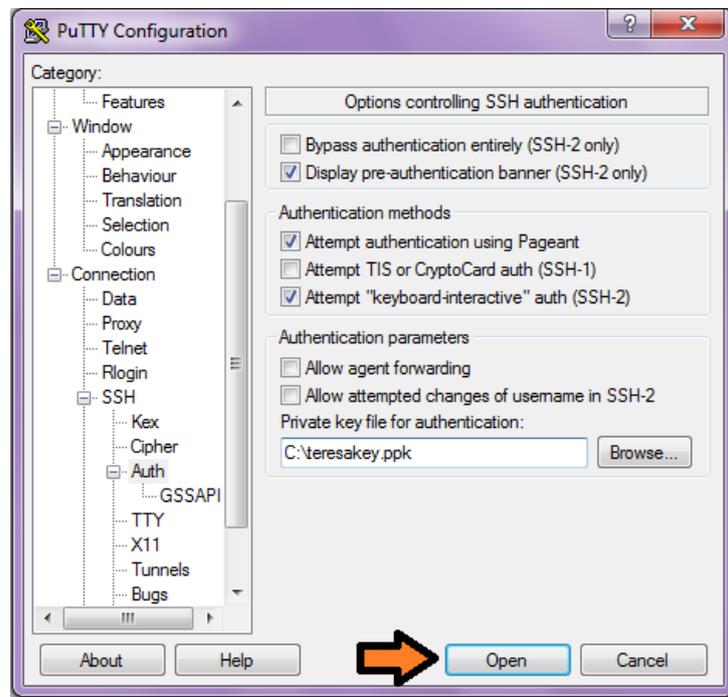


Figura 32: Inicio de conexión en PuTTY

Si es la primera vez que se conecta a esta instancia, PuTTY muestra un cuadro de diálogo de alerta de seguridad que pregunta si confía en el host que se está conectando. Haga clic en “Yes”, se abrirá una ventana y tras todo este proceso el usuario ya está conectado a la instancia.

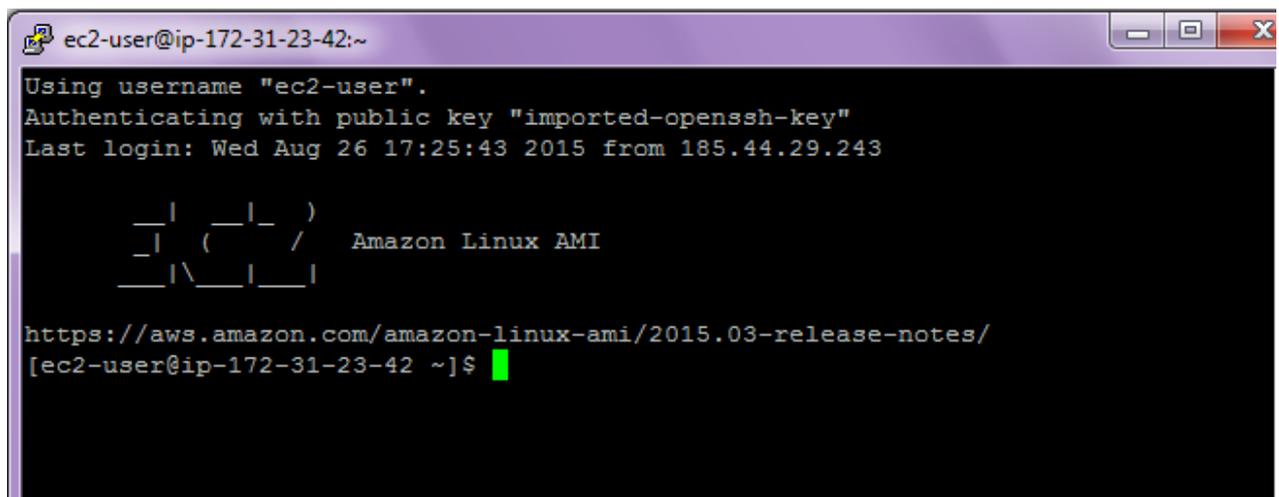


Figura 33: Instancia con PuTTY

- **Conexión a la instancia utilizando el navegador web**

Para la conexión desde el navegador web a una instancia, será necesario tener Java instalado y habilitado en el navegador, los pasos a realizar para poder conectarse a una instancia de esta manera serán los siguientes:

Situados en la consola Amazon EC2, pulsaremos Instancias en el panel de navegación y seleccionaremos la instancia deseada y hacemos clic en “Connect”. Nos aparecerá un cuadro de diálogo en el que escogeremos “A Java SSH Client directly from my browser (Java required)” (Un cliente Java SSH directamente desde mi navegador), en los campos “User Name” escribiremos “ec2-user” y en “Private Key Path” la localización del archivo .pem incluyendo el nombre de la clave par..

Connect To Your Instance [X]

I would like to connect with

- A standalone SSH client
- A Java SSH Client directly from my browser (Java required)

Enter the required information in the fields below to connect to your instance. AWS automatically detects the key pair name, and Public IP for your instance. You need to enter the location and name of the .pem file containing your private key.

Public IP 52.19.148.84

User name

Key name teresakey.pem

Private key path

Save key location Store in browser cache

Launch SSH Client

Close

Figura 34: Conexión a la instancia desde navegador

Haga clic en “Save key location” para almacenar la ubicación de la clave privada en la caché del navegador. Esta opción permite a Amazon EC2 detectar la ubicación de la clave privada en las sesiones del explorador subsiguientes, hasta que se borre la caché de su navegador.

Hacemos clic en Launch SSH Client para conectarnos, y nos aparecerá una ventana de MindTerm, si es la primera vez que nos conectamos, nos pedirá que aceptemos el acuerdo de licencia, la configuración del directorio de host y la del directorio personal.

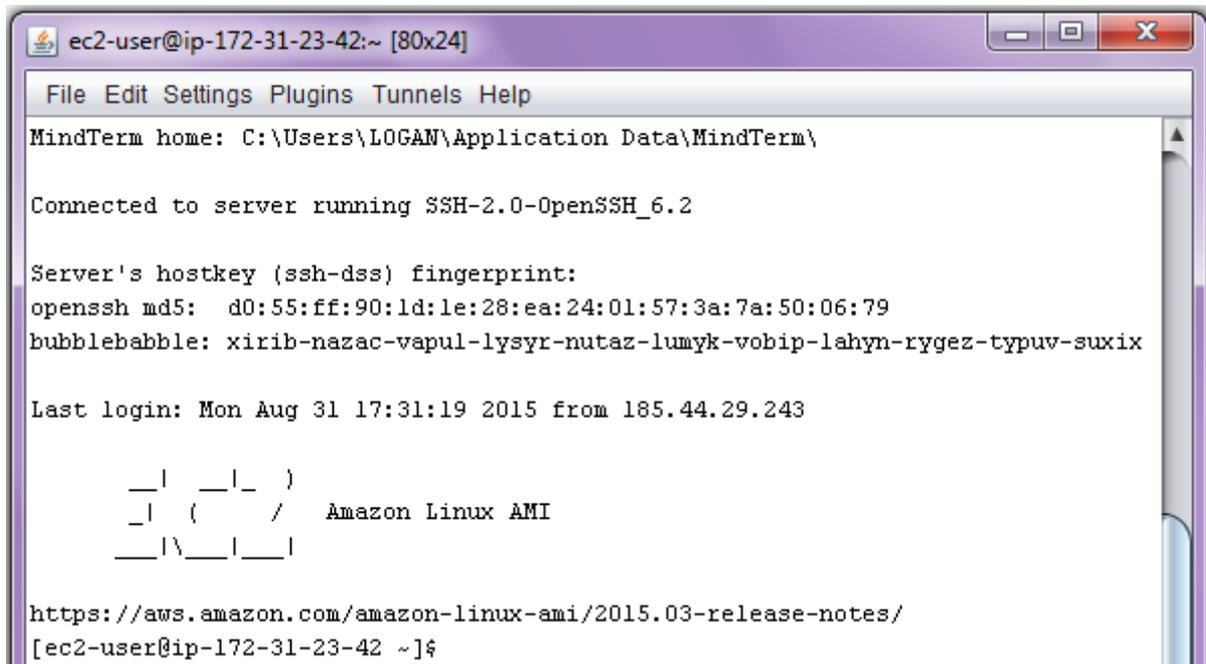


Figura 35: Instancia con MindTerm

Para cualquier usuario, lo más viable sería conectarse a la instancia desde el navegador web, ya que no debe descargar ni instalar nada en el terminal que esté utilizando, y esto permite que si el usuario decide ejecutar la instancia desde cualquier otro lugar pueda realizarlo de forma rápida y sencilla. No obstante, con las últimas actualizaciones de ciertos navegadores no permiten esta última opción y por tanto el mejor método para conectarse a la instancia será utilizando PuTTY si se utiliza Windows.

3.2.3 Volúmenes de almacenamiento de una Instancia

A diferencia de los datos almacenados en un almacén de instancias local (que solo persiste mientras dura esa instancia), los volúmenes de Amazon EBS pueden almacenar los datos independientemente de la vida útil de la instancia. Por lo tanto, se recomienda utilizar el almacén de instancias local solo para los datos temporales y para los datos que requieran un mayor nivel de durabilidad, se recomienda utilizar volúmenes de Amazon EBS.

Tras conectarnos a nuestra instancia, veremos a fondo el almacenamiento que tiene escribiendo la siguiente línea de comandos:

```
$ df -h
```

El resultado que se obtiene es la lista de volúmenes de almacenamientos montados que tiene la instancia y el porcentaje que está utilizado sobre el tamaño total del volumen. Es probable que sea necesario espacio adicional para los datos, por ello una solución sencilla será agregar volúmenes de Amazon EBS a la instancia. Éstos volúmenes sirven como almacenamiento en red para la instancia. Pero primero será necesario crear el volumen en la consola EC2 y adjuntarlo a la instancia antes de montarlo.

Creación de un volumen de almacenamiento

Abriremos la consola Amazon EC2 y en el panel de navegación seleccionaremos Instancias y en la instancia que nos interesa observamos en la descripción de dicha instancia situada en la parte inferior de la pantalla la zona de disponibilidad de ésta.

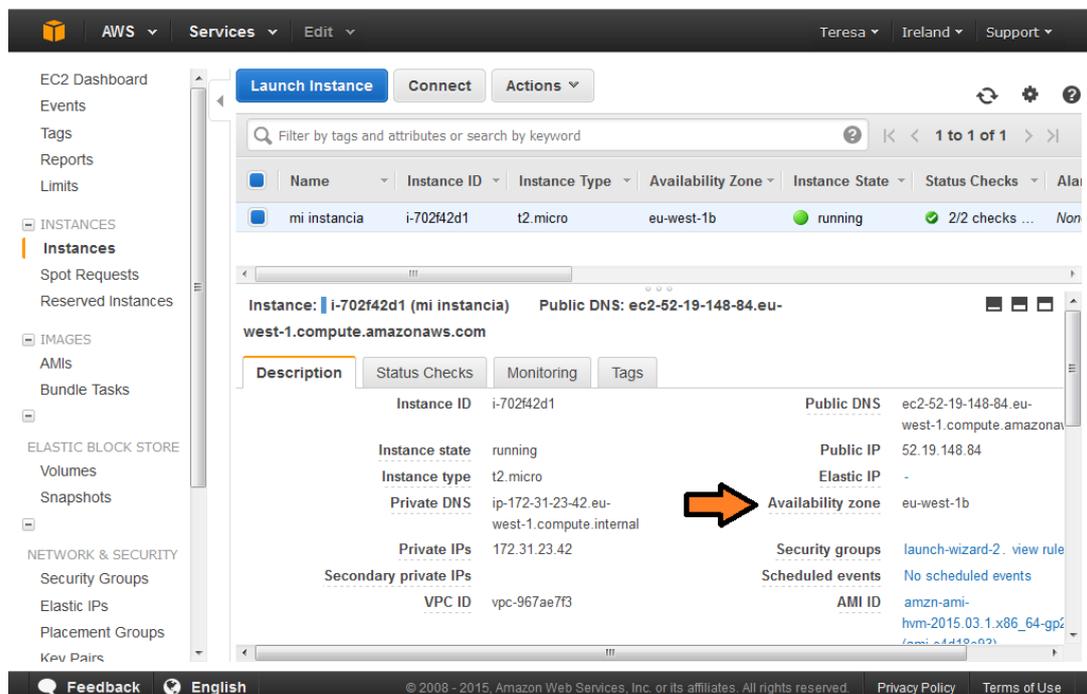
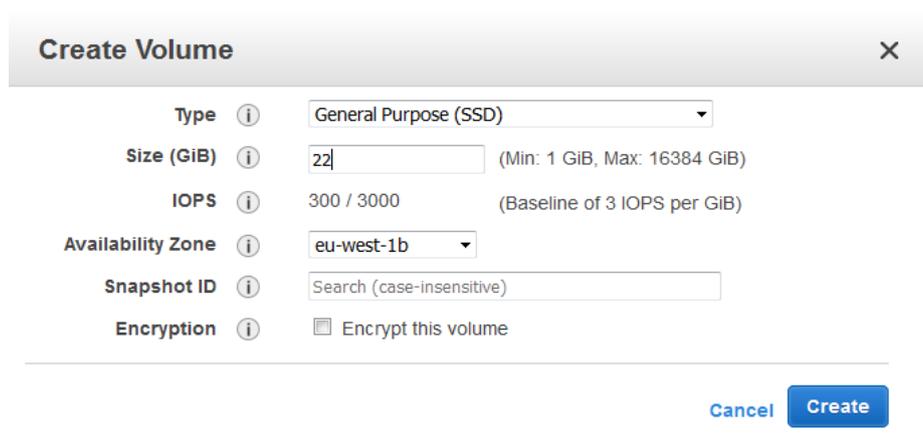


Figura 36: Zona de disponibilidad de la Instancia

Una vez tenemos localizado la zona de disponibilidad de nuestra instancia, hacemos clic en “Volumes”, en el panel de navegación. A continuación, pulsaremos en “Create Volumes”. Aparece un cuadro de diálogo en el que se completará una serie de campos necesarios para dicha creación.

En la opción “Type”, seleccionaremos General Purpose (SSD) para crear un volumen de uso general EBS. Después, se escribe el tamaño del volumen que se desea, con un máximo de 30 GiB de almacenamiento para usuarios de prueba gratuita. Crearemos un volumen de máximo 22 GiB ya que el volumen de almacenamiento que creamos junto con la instancia era de 8GiB.



The screenshot shows a 'Create Volume' dialog box with the following fields and values:

- Type: General Purpose (SSD)
- Size (GiB): 22 (Min: 1 GiB, Max: 16384 GiB)
- IOPS: 300 / 3000 (Baseline of 3 IOPS per GiB)
- Availability Zone: eu-west-1b
- Snapshot ID: Search (case-insensitive)
- Encryption: Encrypt this volume

Buttons: Cancel, Create

Figura 37: Creación de un Volumen de Almacenamiento

A continuación, escogeremos la zona de disponibilidad del volumen de almacenamiento, la zona deberá ser la misma que la de la instancia, si no, no se podrá fijar el volumen a la instancia. Una vez completados todos los campos, pulsamos en “Create”.

Tras crear nuestro volumen de almacenamiento, volveremos a hacer clic en “Volumes” situado en la categoría “Elastic Block Store” del panel de navegación. El volumen que acabamos de crear se encuentra ya en la lista de volúmenes y si el estado de éste es “available” quiere decir que el volumen se encuentra listo para ser conectado a una instancia.

Pulsaremos en el volumen con el botón derecho y seleccionaremos la opción “Attach Volumen” para adjuntar dicho volumen a la instancia. Nos aparecerá un cuadro de diálogo en el que se debe configurar el nombre o ID de la instancia y un nombre de dispositivo no utilizado para la instancia, en este caso dejaremos el que nos aparece ya “/dev/sdf” y una vez configurado los datos necesarios pulsaremos en “Attach”.

Attach Volume [X]

Volume ⓘ
vol-0d3153fd in eu-west-1b

Instance ⓘ
 in eu-west-1b

Device ⓘ

Linux Devices: /dev/sdf through /dev/sdp

Note: Newer Linux kernels may rename your devices to /dev/xvdf through /dev/xvdp internally, even when the device name entered here (and shown in the details) is /dev/sdf through /dev/sdp.

Cancel **Attach**

Figura 38: Adjuntar volumen de almacenamiento

Ahora vemos que el estado del volumen creado ha pasado de “available” a “in-use” y que está adjuntado a la instancia con el nombre del dispositivo que le hemos asignado. No obstante, la instancia no reconoce aún el nuevo volumen, esto es debido a que es necesario montarlo y así al ejecutar el comando **df -h**. Si queremos ver todos los volúmenes de almacenamiento adjuntados, tendremos que utilizar el comando **lsblk**, en caso de no funcionar este comando usaremos la siguiente línea de comandos **sudo fdisk -l | grep disco**.

```
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda 202:0 0 8G 0 disk
  xvda1 202:1 0 8G 0 part /
xvdf 202:80 0 22G 0 disk
```

Para montar el volumen en nuestra instancia, vemos en primer lugar que utilizando el comando **lsblk** tenemos el volumen xvdf adjuntado. Debido a que se ha creado un volumen vacío es necesario formatear el volumen antes de poder montarlo, para ello, utilizaremos el comando **mkfs**. Con este comando crearemos un sistema de archivos ext4 en el volumen y sustituya el nombre del dispositivo en caso de haber utilizado el nombre /dev/xvdf cuando se adjuntó el volumen. Si el volumen no está vacío no hay que utilizar este comando, de lo contrario eliminará los datos existentes.

```
$ sudo mkfs -t ext4 /dev/xvdf
```

La respuesta que obtenemos tras esta línea de comandos es la creación del sistema de archivos con sus especificaciones. Una vez tenemos preparada la instancia procedemos a montar el dispositivo. Ejecutaremos un comando para crear el directorio en el que se encontrará el dispositivo de almacenamiento y a continuación, ejecutaremos el comando para montar dicho dispositivo:

```
$ sudo mkdir /mnt/my-data  
$ sudo mount /dev/xvdf /mnt/my-data
```

El volumen de almacenamiento ya se encuentra montado y adjuntado a nuestra instancia, para comprobarlo sólo tendremos que escribir de nuevo el comando **df -h**:

```
Filesystem      Size  Used Avail Use% Mounted on  
/dev/xvda1     7.8G  1.2G  6.6G  15% /  
devtmpfs       490M   60K  490M   1% /dev  
tmpfs          499M    0   499M   0% /dev/shm  
/dev/xvdf      20G   44M   19G   1% /mnt/my-data
```

Por último, si deseamos ver el contenido del nuevo volumen, sólo tendremos que ejecutar el siguiente comando:

```
$ ls /mnt/my-data
```

Una vez realizado todos estos pasos de nuestra instancia, se puede decir que tenemos completada una arquitectura ejemplo que puede ser utilizada para cualquier propósito del usuario y que debe ser personalizada de acuerdo con las aplicaciones que se van a desarrollar.

Estos volúmenes pueden ser reutilizados cuando se acaba el ciclo de vida una instancia por una nueva si el volumen existente se adjunta a ésta, siendo conveniente el formateo de dicho volumen para este nuevo uso.

3.3 FUNCIONAMIENTO DE SERVICIOS EN LA CONSOLA AWS

En el capítulo anterior se describen de forma general algunos de los servicios que ofrece Amazon Web Services y se especifica un poco más en cuanto a los servicios Amazon EC2, Amazon SQS, Amazon S3 y por último pero no menos importante Amazon DynamoDB. En este capítulo ya se ha visto cómo funciona Amazon EC2 creando una instancia y comprobando que ésta funciona. Por ello, en el presente apartado se explicará el funcionamiento de los servicios Amazon SQS, S3 y DynamoDB.

3.3.1 Funcionamiento de Amazon SQS

Como ya se sabe, Amazon SQS es un sistema de cola distribuida que permite a las aplicaciones web crear de forma sencilla y fiable una cola de mensajes, esta cola es un bucket temporal de datos que están a la espera de ser procesados. Además Amazon SQS está diseñado para estar siempre disponible y entregar mensajes.

Para entender un poco más cómo funciona Amazon SQS, se describe a continuación el ciclo de vida que tiene un mensaje de Amazon SQS al que llamaremos A (Mensaje A). Suponiendo que ya existe una cola, un componente 1 envía un mensaje A a la cola y este mensaje se expande de forma redundante a través de los servidores de SQS, además se activa un tiempo de visibilidad para que este mensaje pueda ser visto por otro componente que pueda procesarlo cuando esté disponible. A continuación, un componente 2 recupera el mensaje A de la cola para procesarlo. Mientras se está procesando el mensaje A, el tiempo de visibilidad de este mensaje sigue activo. Cuando el mensaje A ha sido procesado, el componente 2 elimina el mensaje para evitar que sea de nuevo recibido y procesado una vez expire el tiempo de espera de visibilidad.

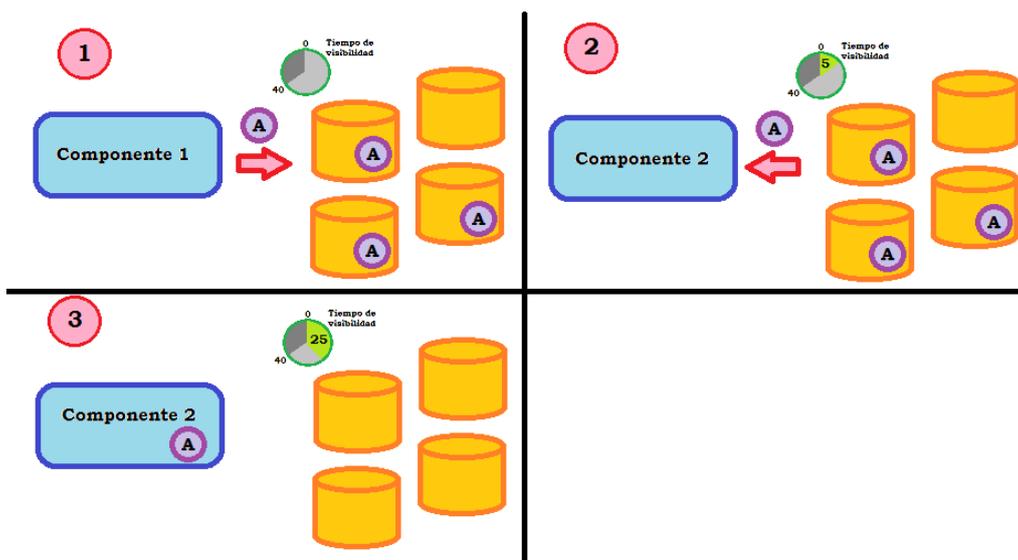


Figura 39: Ciclo de vida de un mensaje SQS

Una vez sabemos cómo funciona una cola SQS, es momento de ver cómo se crea dicha cola, para ello hay múltiples maneras de realizarlo. El usuario tiene la opción tanto de escribir el código de la cola SQS como de utilizar la consola de AWS.

En el caso de querer escribir el código, el proceso a realizar para crear nuestra cola SQS (en PHP) serán los siguientes:

Para crear la cola SQS, primero se requiere una librería PHP (sqs.client.php) que se obtiene del paquete SDK de PHP en AWS, a continuación se especifican las credenciales AWS, es decir, se ponen los valores de la clave de acceso y la clave privada de AWS (“aws_access_key_id” y “aws_secret_access_key”) obtenidas tras el registro de la cuenta AWS y también se pone el “ENDPOINT” que sirve para acceder al servicio:

```
require_once('sqs.client.php');
define('AWS_ACCESS_KEY_ID', 'YOUR-KEY');
define('AWS_SECRET_ACCESS_KEY', 'YOUR-KEY');
define('SQS_ENDPOINT', 'http://queue.amazonaws.com');

$q = new SQSClient(AWS_ACCESS_KEY_ID, AWS_SECRET_ACCESS_KEY,
SQS_ENDPOINT);
try{
    $result = $q->CreateQueue(SQS_Q); //SQS_Q es el nombre de la cola
    //echo 'Queue Created: ', $result, "\n<br />\n";
}catch(Exception $e){ //Lanzamos excepción si no se ha creado la cola
    throw($e);
}
```

Una vez creada la cola SQS, se pueden enviar mensajes a la cola, recibir o eliminar dichos mensajes, para poder realizar estas operaciones sólo tendremos que escribir las siguientes líneas de código:

```
//Envío de un mensaje a la cola, este mensaje va codificado como URL
$messageId = $q->SendMessage(urlencode(SQS_TEST_MESSAGE));
//Recepción de un mensaje en la cola, se decodifica el mensaje con urldecode()
$messages = $q->ReceiveMessage();
    foreach($messages as $message)
    {
        echo 'Message received', "\n<br />\n";
        echo 'message id: ', $message->MessageId, "\n<br />\n";
        echo 'receipt handle: ', $message->ReceiptHandle, "\n<br />\n";
        echo 'message: ', urldecode($message->Body), "\n<br />\n";
    }
//Eliminación de un mensaje en la cola
$q->DeleteMessage($message->ReceiptHandle)
```

Para recibir el mensaje en cola, hay que saber que el "ReceiptHandle" es una cadena de 1024 caracteres asociados con el acto de recibir el mensaje de la cola SQS, no con el mensaje en sí. Para eliminar el mensaje, se debe proporcionar el ReceiptHandle, esto quiere decir que se deberá recibir un mensaje de la cola antes de poder eliminarlo.

Desde la consola de Amazon Web Services, la creación de una cola SQS es bastante más sencilla, sólo tendremos que pinchar en "Console Home" y elegir el servicio SQS en "Application Services". A continuación pulsaremos en Create Queue:



Figura 40: Crear nueva cola SQS

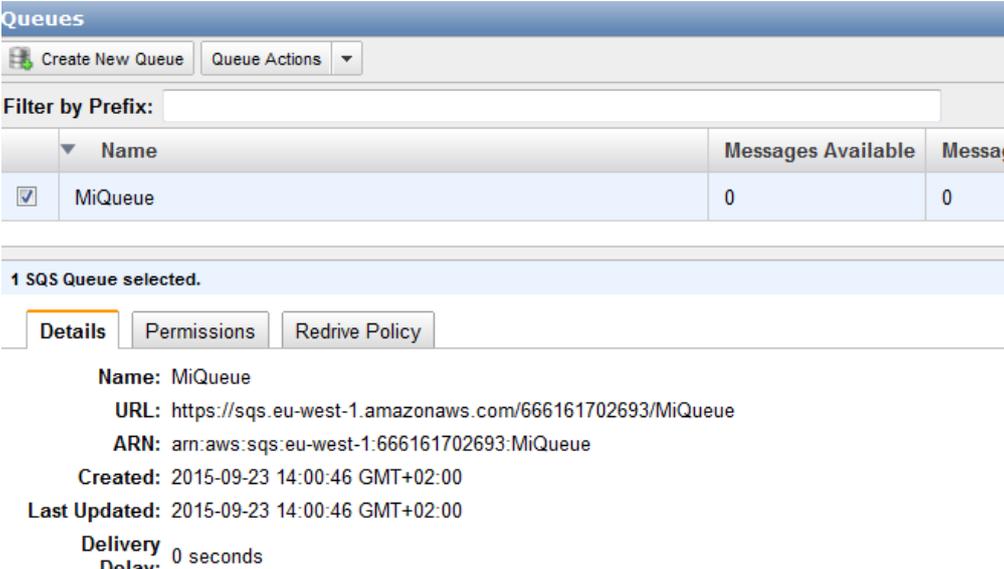
Ahora nos aparecerá un cuadro de diálogo en el que podemos configurar la nueva cola SQS, se deberá poner obligatoriamente el nombre de la cola para poder crearla, las especificaciones que aparecen más abajo pueden modificarse dependiendo de las necesidades del usuario al crearla.

Estas especificaciones serán:

Default Visibility Timeout (longitud de tiempo que un mensaje recibido desde una cola será visible para otros componentes de recepción), Message Retention Period (cantidad de tiempo que Amazon SQS retendrá un mensaje si no se borra), Maximum Message Size (Tamaño máximo del mensaje), Delivery Delay (cantidad de tiempo en demorar la primera entrega de todos los mensajes añadidos a esta cola), Receive Message Wait Time (tiempo de espera máximo de recepción de un mensaje que esté disponible) y por último Use Redrive Policy (enviará un mensaje después de exceder el tiempo de espera máximo de recepción.)

Figura 41: Especificaciones de la cola SQS

Una vez creada la cola, nos aparecerá la lista de colas SQS creadas donde aparecerá la cola creada y más abajo, los detalles de ésta:



The screenshot displays the AWS Management Console interface for SQS Queues. At the top, there is a header 'Queues' and a navigation bar with 'Create New Queue' and 'Queue Actions'. Below this is a 'Filter by Prefix:' search box. A table lists the queues, with one entry 'MiQueue' selected, showing 0 messages available and 0 messages sent. Below the table, a message states '1 SQS Queue selected.' and three tabs are visible: 'Details' (selected), 'Permissions', and 'Redrive Policy'. The 'Details' tab shows the following information:

- Name:** MiQueue
- URL:** <https://sqs.eu-west-1.amazonaws.com/666161702693/MiQueue>
- ARN:** `arn:aws:sqs:eu-west-1:666161702693:MiQueue`
- Created:** 2015-09-23 14:00:46 GMT+02:00
- Last Updated:** 2015-09-23 14:00:46 GMT+02:00
- Delivery Delay:** 0 seconds

Figura 42: Aspecto lista de cola SQS

3.3.2 Funcionamiento de Amazon S3

Amazon S3 almacena datos como objetos dentro de recursos conocidos como "buckets". Además de almacenar todos los objetos que el usuario desee dentro de un bucket, podrá realizar operaciones de escritura, lectura y eliminación de los objetos almacenados en el contenedor.

Los datos cuando se almacenan en el bucket de Amazon S3 se suprime la necesidad de planificar sobre la capacidad, cuando los datos envejecen, Amazon S3 se encarga de migrarlos de manera transparente y automática a un nuevo hardware. Cuando los datos alcanzan su fecha de vencimiento, Amazon S3 brinda opciones programáticas para realizar eliminaciones recurrentes y de gran volumen.

Al igual que en Amazon SQS, el usuario puede crear el bucket de Amazon S3 tanto por código como por la consola de AWS. Utilizando el código PHP, crear un bucket se realiza de la siguiente manera:

Se creará un archivo s3 class.s3.php donde se incluyen las credenciales de AWS como variables, además de todos los métodos necesarios que se vayan a utilizar, y en el archivo de la aplicación, lo único que se tendrá que hacer es definir una variable "BUCKET" con un valor (en el caso de la aplicación una ID aleatoria guardada en \$random) y más tarde crear el bucket llamando al método y pasándole la variable BUCKET:

```
define('BUCKET', random);

$s3 = new S3();
$s3->createBucket(BUCKET);

-----class.s3.php-----

class S3
{
    var $_key    = "YOUR KEY";
    var $_secret = "YOUR SECRET KEY";

function createBucket($bucket)
{
    $req = array( "verb" => "PUT",
                 "md5"  => null,
                 "type" => null,
                 "headers" => null,
                 "resource" => "/"$bucket",
                 );
    $result = $this->sendRequest($req);
    return $this->isOk($result);
}
```

Para colocar, cargar, descargar, modificar o eliminar datos dentro de nuestro nuevo bucket, sólo tendremos que llamar al método correspondiente:

```
$var=$s3->putObject( BUCKET, $object, $filename, $public = null, $disposition = null)
```

Donde BUCKET será el bucket donde se subirá el objeto, \$object el archivo que se va a cargar dentro del bucket, \$filename es la ruta donde se encuentra el archivo a cargar, \$public es una variable que indica si el archivo es público y \$disposition guarda el archivo en una ubicación específica dentro del bucket. El resto de métodos que se utilizan con los buckets se podrá visualizar en el apartado de creación de la aplicación.

Para crear el bucket en la consola AWS se realizará siguiendo este proceso:

Desde “Console Home” hacemos clic en S3 ubicado en “Storage & Content Delivery” y a continuación pulsaremos en “Create Bucket”.

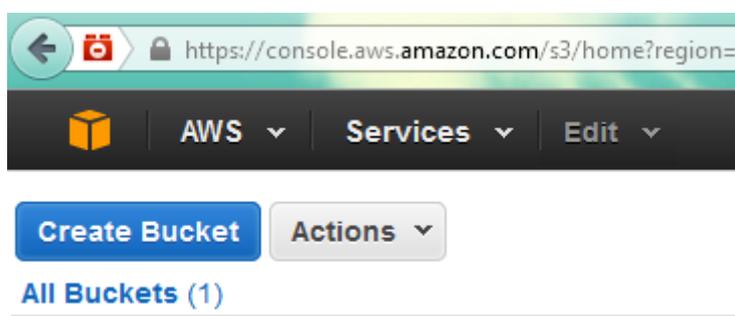


Figura 43: Crear Bucket

Aparecerá un cuadro de diálogo en el que se debe poner nombre al bucket y la región donde se va a utilizar para optimizar la latencia, minimizar costes o abordar requisitos normativos. Los objetos almacenados en una región nunca dejarán dicha región salvo que el usuario los transfiera de forma específica a otra región. Cuando se realice esto, sólo habrá que pulsar sobre “create” para tener nuestro bucket S3.

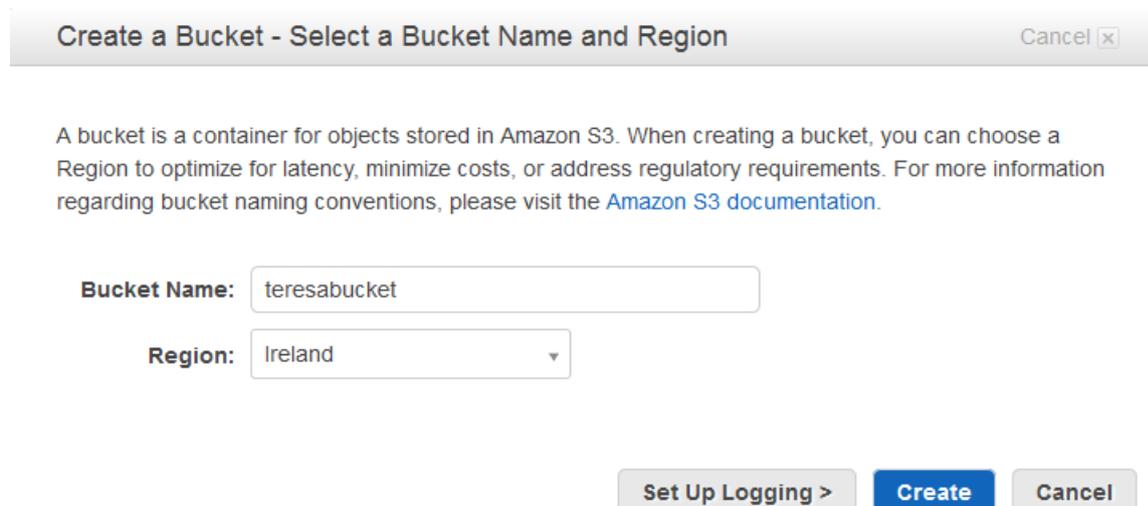


Figura 44: Nombre del bucket

Una vez se ha creado se ha creado el bucket, aparecerá la lista de buckets que sean creados y en el lateral aparecerán los detalles de dicho bucket. y ahora que ha creado un bucket, el usuario está listo para agregarle un objeto. Un objeto puede ser cualquier clase de archivo: un archivo de texto, una fotografía, un vídeo, etc. Cuando añade un archivo a Amazon S3, tiene la opción de incluir metadatos como la fecha de creación del objeto o la clase de almacenamiento, entre otros, además de establecer permisos para controlar el acceso al archivo.

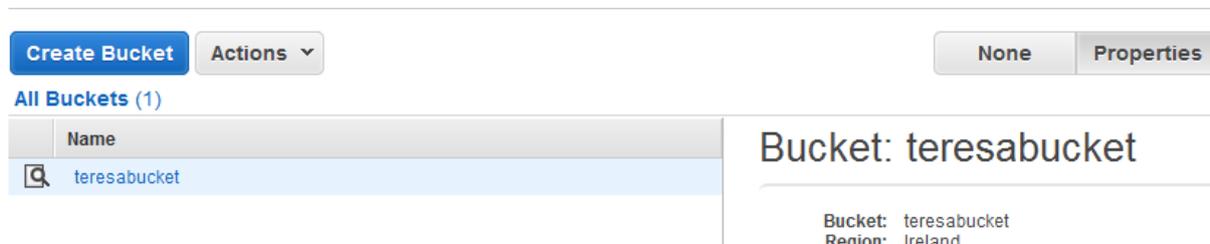


Figura 45: Aspecto Lista de Buckets

Para añadir un objeto, pulsamos en el bucket creado, y observamos que tenemos la opción de crear directorios y de subir objetos, pulsaremos entonces en “Upload” para subir un archivo.



Figura 46: Subida de objetos al bucket

En el asistente Upload–Select Files, si se desea cargar una carpeta entera, debe hacer clic en Enable Enhanced Uploader para instalar el applet necesario de Java. Sólo es necesario hacer esto una vez por cada sesión de la consola.

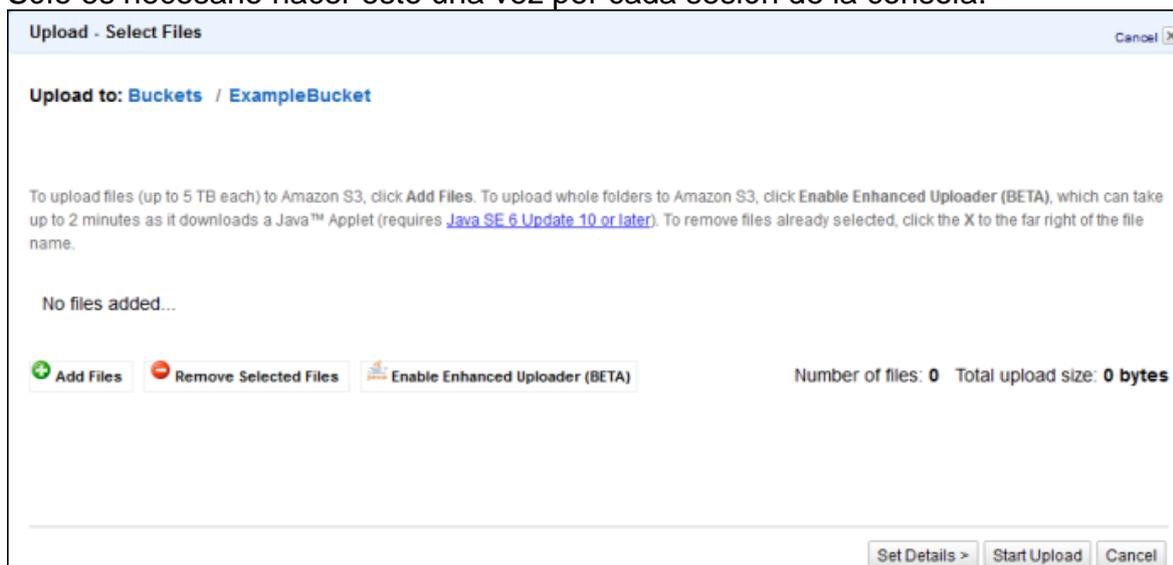


Figura 47: Asistente de subida de objetos

A continuación, se hace clic en “Add Files” para subir los archivos y se abrirá un cuadro de diálogo de selección de archivos. ahora se escoge el archivo a cargar y pulsamos en “Start Upload”.

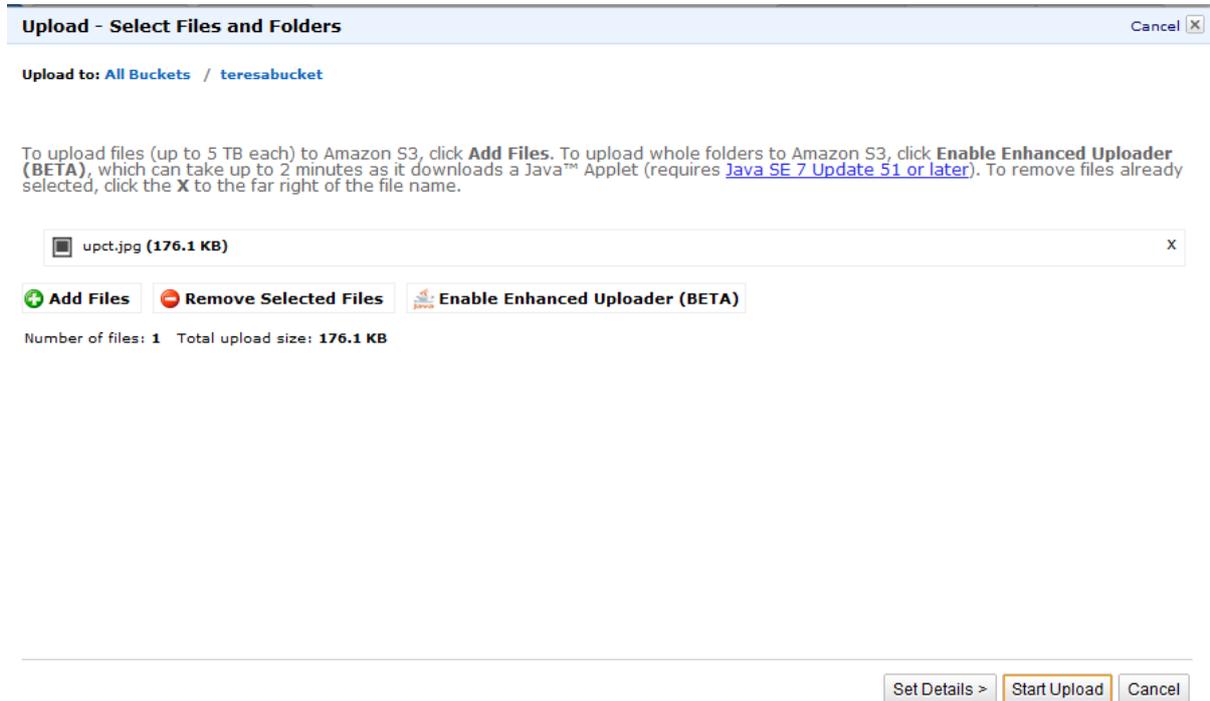


Figura 48: Subida del Archivo al bucket

Una vez subido el archivo, se verá dentro del bucket y se podrá abrir para verse en el navegador, descargarse, cambiar el nombre o mover el archivo.

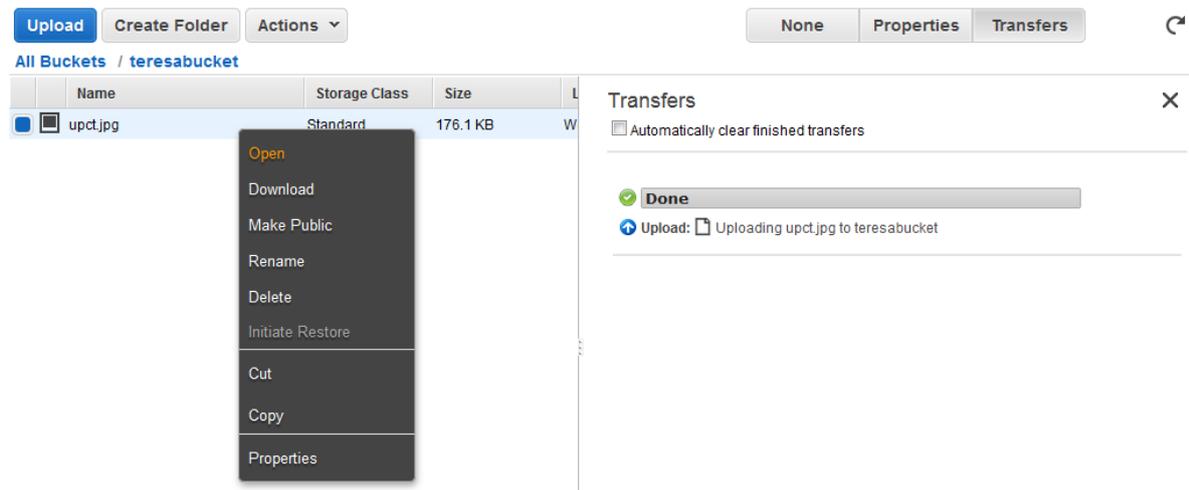


Figura 49: Objeto Subido

3.3.3 Funcionamiento de Amazon DynamoDB

Como anteriormente se ha mencionado anteriormente, Amazon DynamoDB es un servicio de base de datos totalmente gestionada en la nube compatible con modelos de almacenamiento de datos de valor de clave y de documentos. Su modelo de datos flexible y su rendimiento es fiable.

Para crear y utilizar Amazon DynamoDB existen muchos métodos, como utilizar la consola de AWS, Java, Python o Node.js, entre otros. Se procede a la creación de una base de datos DynamoDB para clientes PHP.

Asumiendo que tenemos descargado el paquete SDK de AWS para PHP, la forma más fácil de comenzar y trabajar de manera rápida es utilizar el método "Aws\DynamoDb\DynamoDbClient::factory()" y proporcionar las credenciales AWS al perfil en el archivo ~/.aws/credentials. El parámetro de la región en la que nos encontramos es esencial para la creación de la base de datos:

```
Use Aws\DynamoDb\DynamoDbClient;

$client = DynamoDbClient::factory(array(
    'profile' => '<profile in your aws credentials file>',
    'region' => '<region name>'
));
```

Una vez realizado este paso, se realiza un constructor para poder conectarse a Amazon DynamoDB con un archivo de configuración (es un archivo que proporciona ajustes de configuración predeterminados para asociar clases de cliente con nombres de servicio):

```
use Aws\Common\Aws;

// Create a service builder using a configuration file
$saws = Aws::factory('/path/to/my_config.json');

// Get the client from the builder by namespace
$client = $saws->get('DynamoDb');
```

Ahora se pueden crear tablas, para almacenar artículos con sus elementos, para crear una tabla es necesario crear un esquema para las claves de la tabla y también tendrá que especificar la cantidad de rendimiento aprovisionado que debe ponerse a disposición de la tabla.

```
// Create an "errors" table
$client->createTable(array(
    'TableName' => 'errors',
    'AttributeDefinitions' => array(
        array(
            'AttributeName' => 'id',
            'AttributeType' => 'N'
        ),
        array(
            'AttributeName' => 'time',
            'AttributeType' => 'N'
        )
    ),
    'KeySchema' => array(
        array(
            'AttributeName' => 'id',
            'KeyType' => 'HASH'
        ),
        array(
            'AttributeName' => 'time',
            'KeyType' => 'RANGE'
        )
    ),
    'ProvisionedThroughput' => array(
        'ReadCapacityUnits' => 10,
        'WriteCapacityUnits' => 20
    )
));
```

La tabla tendrá ahora un estado de CREATING mientras que está siendo aprovisionada, y se puede sondear la tabla hasta que se convierte en ACTIVE.

```
// Wait until the table is created and active
$client->waitUntil('TableExists', array(
    'TableName' => 'errors'
));
```

Las tablas tienen múltiples funciones utilizando métodos como describeTable() para obtener información, updateTable() para actualizar la tabla, ver en una lista el número de tablas creadas, etc.

Para agregar elementos a la lista se utilizará el método putItem() en el cliente:

```
$result = $client->putItem(array(
    'TableName' => 'errors',
    'Item' => array(
        'id' => array('N' => '1201'),
        'time' => array('N' => $time),
        'error' => array('S' => 'Executive overflow'),
        'message' => array('S' => 'no vacant areas')
    )
));
```

Y para comprobar si los artículos están subidos, se utilizará el método getItem() para el cliente. Debido a Amazon DynamoDB trabaja bajo un modelo 'consistencia eventual', tenemos que especificar que estamos llevando a cabo una operación de lectura consistente:

```
$result = $client->getItem(array(
    'ConsistentRead' => true,
    'TableName' => 'errors',
    'Key' => array(
        'id' => array('N' => '1201'),
        'time' => array('N' => $time)
    )
));

// Grab value from the result object like an array
echo $result['Item']['id']['N'] . "\n";
//> 1201
echo $result->getPath('Item/id/N') . "\n";
//> 1201
echo $result['Item']['error']['S'] . "\n";
//> Executive overflow
echo $result['Item']['message']['S'] . "\n";
//> no vacant areas
```

Para crear la base de datos Amazon DynamoDB con la consola de AWS como se ha hecho con los servicios de Amazon SQS y Amazon S3, desde “Console Home” pulsaremos sobre DynamoDB en “DataBase” y haremos clic en “Create table” para crear la tabla.

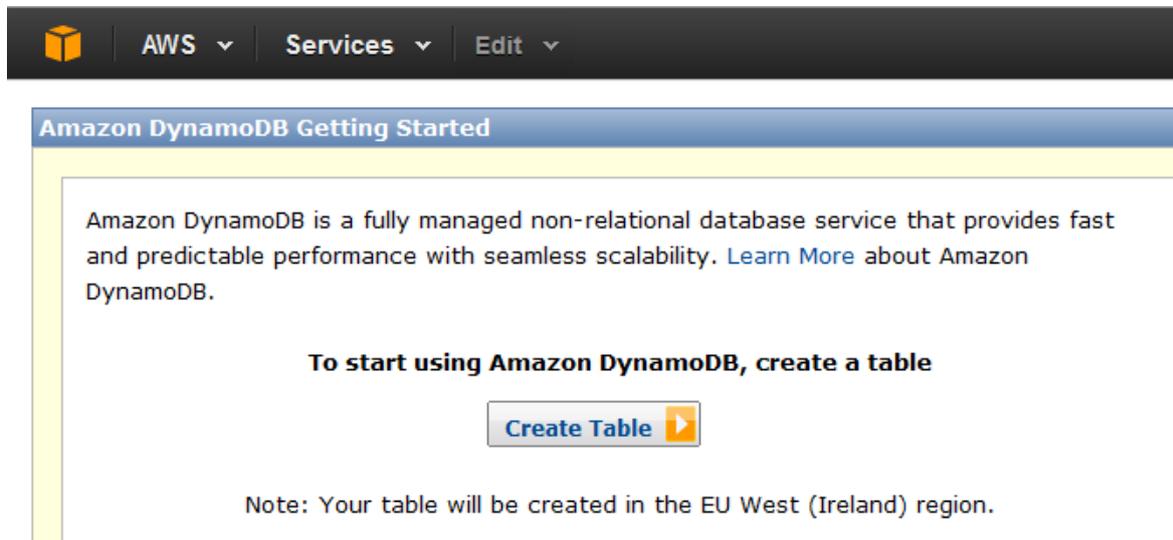


Figura 50: Crear Tabla

Se abrirá un asistente para crear tablas, y estableceremos un nombre y la clave primaria, escogemos clave de tipo Hash, rellenamos el campo y pulsamos en continuar.

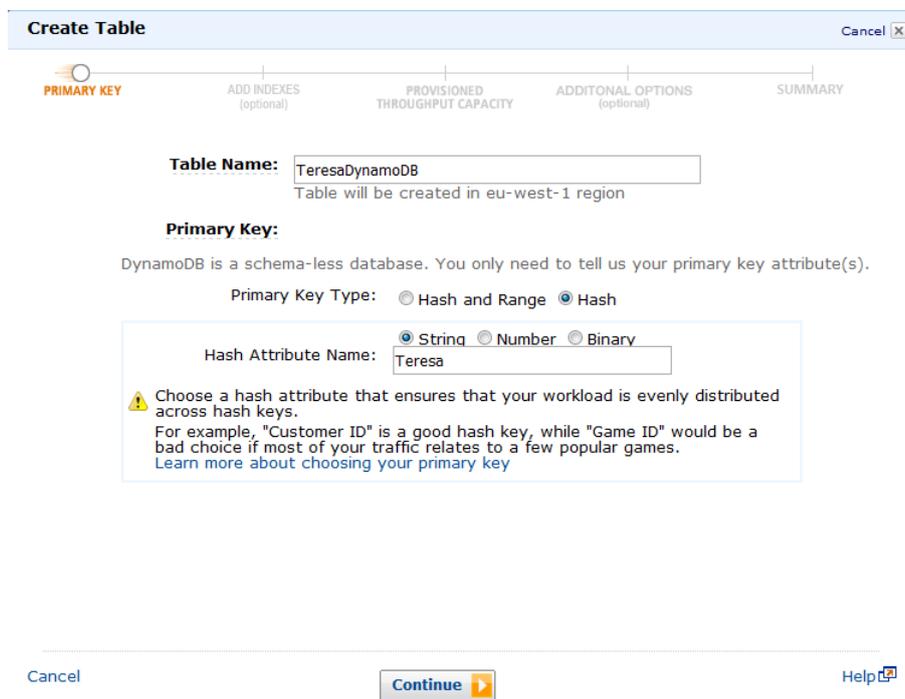


Figura 51: Clave Primaria

A continuación se procede a añadir índices, esta parte es opcional, Un índice es una estructura de datos que mantiene un hash alternativo y una clave range. Se puede utilizar para consultar un elemento de la misma forma en que usa el hash y la clave range. Rellenamos los campos y pulsamos en “Add Index to Table“. Cuando se hayan añadido todos los que se quieran, pulsamos en continuar.

Figura 52: Añadir índices

El tercer paso de la creación de la tabla será especificar la capacidad de aprovisionado. Dejamos sin marcar la casilla de “ayuda a calcular la capacidad estimada“. Es importante configurar la capacidad de aprovisionado adecuada en función de su tamaño del artículo esperada y sus tasas de solicitud de espera de lectura y escritura.

Figura 53: Capacidad de aprovisionado

El último paso antes de la realización de nuestra tabla, será seleccionar la casilla de verificación de uso básico de alarmas. Esto configura automáticamente las alarmas CloudWatch para que le notifique cuando su consumo alcanza el 80% de rendimiento provisionado de la tabla. Sólo hay que rellenar el campo “Notification to” con el correo electrónico para que nos avise cuando esto ocurra.

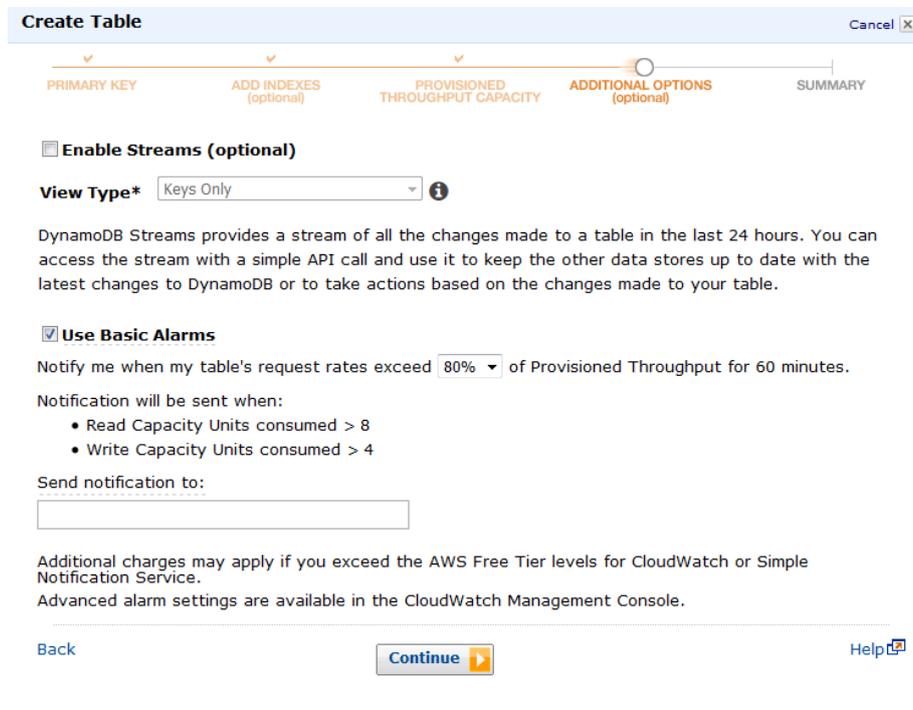


Figura 54: Configuración del uso básico de alarmas

Una vez realizado este paso y vista la revisión de todos los pasos realizados hasta el momento, pulsaremos en crear la tabla. Automáticamente se recibirá un correo de Amazon Web Services confirmando que las notificaciones llegarán a ese correo, por último se verá la lista de tablas creadas en DynamoDB.

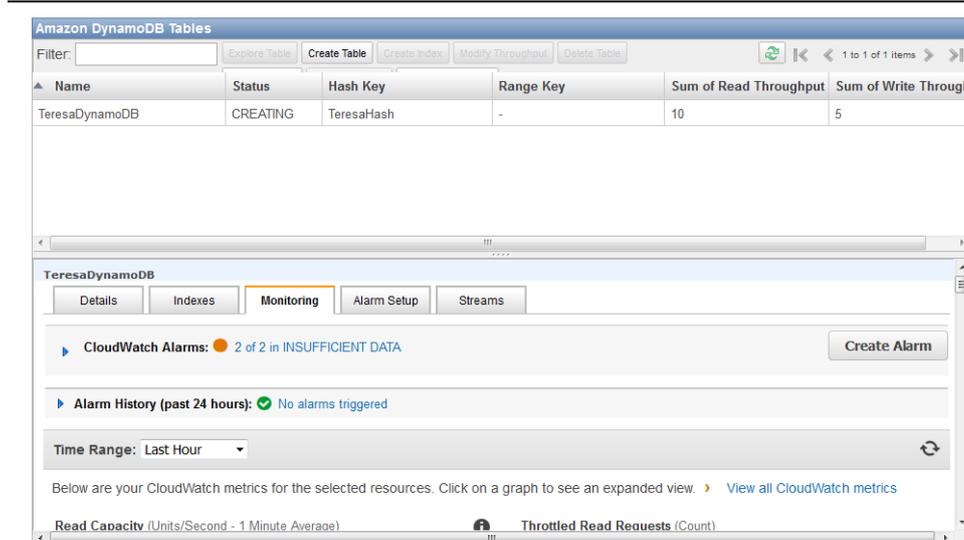


Figura 55: Aspecto Lista de tablas en DynamoDB

DESARROLLO DE UNA APLICACIÓN CON AMAZON WEB SERVICES

El desarrollo de aplicaciones en Cloud Computing ya es una realidad, y por ello y utilizando el proveedor Amazon Web Services se mostrará en los siguientes apartados un sencillo ejemplo en el que se desarrolla una aplicación de procesamiento de imágenes, el código que se ha utilizado para el desarrollo de la aplicación es uno ya existente creado por John Fronckowiak y Tom Myer, con modificaciones, ya que el propósito es ver cómo funciona cualquier aplicación utilizando Amazon AWS.

4.1 PROCESADOR DE IMÁGENES

4.1.1 Introducción

El desarrollo de una aplicación de procesamiento de imágenes, donde los usuarios de Internet proporcionan sus imágenes, se puede atascar si no se tiene espacio suficiente en la CPU o en el disco duro, además el procesamiento de imágenes puede ser propenso a errores, laborioso y repetitivo.

Un procesador de imágenes utilizando los servicios en “La Nube” es una solución sencilla en la que los usuarios suben una imagen y ésta se procesa en Amazon Web Services. Hay múltiples librerías que permiten realizar distintas tareas en el procesamiento de imágenes como poner la imagen en blanco y negro, voltear la imagen, en este caso haremos una miniatura de la imagen que el usuario quiera proceder, aunque se pueden añadir todas las tareas que se crean posibles.

4.1.2 Requisitos de la aplicación

Para que nuestra aplicación funcione correctamente, implica disponer de:

- Una instancia Amazon Elastic Cloud Computing (EC2) en ejecución de Apache y PHP (PHP debe incluir PEAR y GD para el procesamiento de las imágenes).
- Un almacenamiento Amazon S3 (Amazon Simple Storage Service) para mantener las imágenes cargadas.
- Amazon SimpleDB para el seguimiento de los metadatos de las imágenes.
- Un servicio Amazon SQS (Simple Query Service) que envía y recibe mensajes sobre las imágenes.

4.1.3 Funcionamiento de la Aplicación

El funcionamiento del procesador de imágenes utilizando los servicios necesarios de Amazon Web Services es de la siguiente manera:

En primer lugar, un cliente externo se conecta desde el navegador web y sube una imagen para procesar. Esa imagen, se recibe en el servidor web y éste realiza varias tareas, la primera es guardar la imagen temporalmente en Amazon S3, la segunda es introducir una solicitud de procesamiento en la cola Amazon SQS y la tercera es que la información sobre la solicitud de procesamiento se guarda en la base de datos de Amazon SimpleDB.

Amazon EC2 tiene la aplicación web PHP y tiene la característica de poder interactuar con los otros servicios ya nombrados. Cuando el cliente externo envía su imagen para procesarla, el servicio de Amazon EC2 se encuentra ocupado procesando imágenes de otros clientes externos que mandaron con anterioridad su imagen para procesar. Siempre que se procesa una imagen, la aplicación web se ocupa de comprobar la cola de Amazon SQS para continuar procesando nuevas imágenes.

Cuando llega el momento de procesar la imagen, se ve la solicitud pendiente del procesamiento de la imagen en la cola y se recupera la imagen de Amazon S3 y la información de la solicitud de la base de datos y se procede al procesamiento de la imagen. Cuando la imagen es procesada, la aplicación se encarga de enviar dicha imagen por correo electrónico al cliente externo.

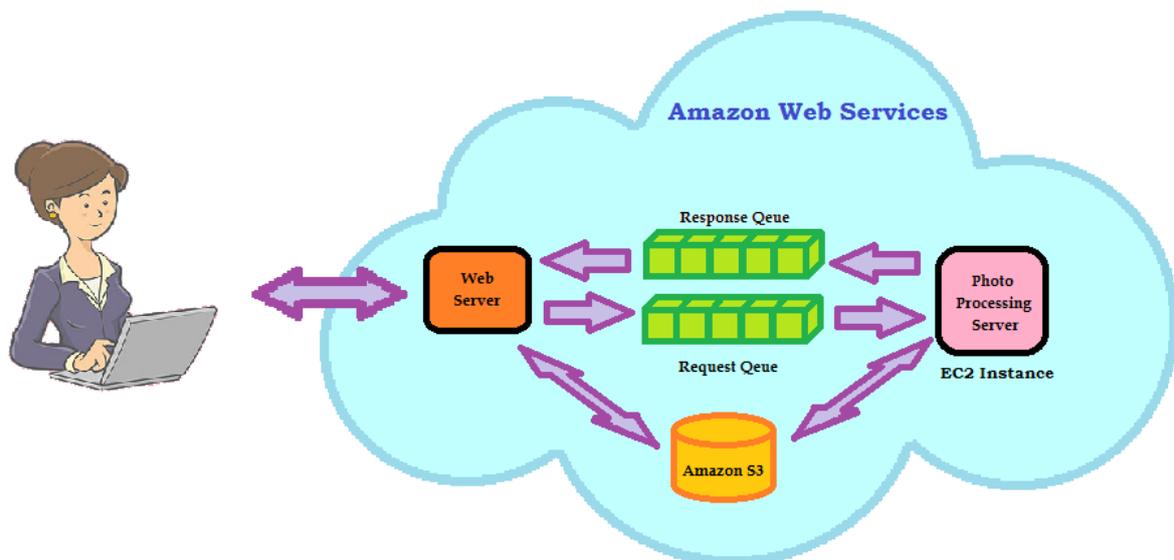


Figura 56: Procesador de Imágenes

4.2 CREACIÓN DE LA APLICACIÓN

Para la creación de la aplicación, como se ha mencionado anteriormente, está construida sobre una instancia Amazon EC2, esta instancia necesitará de herramientas como Apache, PHP, PEAR y GD para que pueda funcionar. Cuenta con Amazon S3 para el almacenamiento, Amazon SimpleDB para el seguimiento de los metadatos y por último Amazon SQS para la mensajería entre los componentes.

En este apartado del proyecto se explicarán algunos de los pasos que se han seguido para crear el código de la aplicación, hay muchas maneras de realizar este tipo de aplicación pero nos decantamos por ésta que es sencilla.

4.2.1 Subida de la imagen a la aplicación

Para la subida de la imagen se crea un archivo PHP que contiene un formulario de HTML sencillo en el que el usuario especifica qué archivo quiere subir, el nombre de usuario y el correo electrónico. Este archivo se coloca en la web raíz de Amazon EC2.

```
<html>
<head>
<title>Procesador de imágenes</title>
</head>
<body>
<h1>Crear Miniaturas</h1>
<form action="upload.php" method="post" enctype="multipart/form-data">
<input type="hidden" name="MAX_FILE_SIZE" value="1000000" />
<p><label for="myfile">Elija imagen para subir</label>
<input name="userfile" id="myfile" type="file" /></p>

<p><label for="name">Nombre :</label>
<input type='text' name='fullname' id='name'/>
</p>

<p><label for="email">Correo electrónico: </label>
<input type='text' name='emailaddress' id='email'/>
</p>

<p><label for='crop'><input type='checkbox' name='task' id='thumbnail'
value='thumbnail' checked='checked'/><b>Crear Miniatura</b></label></p>

<p><input type="submit" value="Subir Archivo" /></p>

</form>
</body>
</html>
```

Como podemos observar, upload.php será el lugar en el que se establece la acción forma del formulario, y en el que se realizarán diversas tareas que veremos a continuación.

4.2.2 Pasos previos al procesado del archivo

Una vez tenemos la imagen cargada, se deben realizar los siguientes pasos antes de procesarla:

Antes de comenzar con los pasos que vamos a describir, es necesario definir las constantes que tendrán la clave de acceso de AWS y la clave privada, además se utiliza require_once() para vincular en algunos archivos que servirán de ayuda para realizar unas tareas.

```
require_once 'Crypt/HMAC.php';

require_once 'HTTP/Request.php';

require_once('class.s3.php');

require_once('simpledb.class.php');

require_once('sqs.client.php');

define('AWS_ACCESS_KEY_ID', 'change-this');

define('AWS_SECRET_ACCESS_KEY', 'change-this-too');
```

A continuación, y como se ha explicado en el funcionamiento de la aplicación, utilizaremos tres clases para trabajar con Amazon S3, SQS y SimpleDB. Para ello se han reconstruido las clases sobre unas ya existentes, que son class.s3.php, simpledb.class.php y sqs.client.php creadas por John Fronckowiak, Alex Bosworth y Amazon, respectivamente.

El primer paso que realizaremos es crear una cadena aleatoria, pues el usuario va a utilizarla como un nombre de bucket de Amazon S3, la clave para la entrada de Amazon SimpleDB y el cuerpo del mensaje enviado a través de Amazon SQS. La cadena aleatoria se guardará como una variable aleatoria (\$random) y tras esto, se pueden crear nuevas constantes.

```
$random = md5(uniqid(rand(), true));

define('BUCKET',$random);
define('DOMAIN','photo_jobs');
define('SQS_Q', 'photo_q');
define('SQS_ENDPOINT', 'http://queue.amazonaws.com');
```

Tras definir el valor del Bucket, es momento de conectarse a Amazon S3, crear el bucket y subir el archivo a Amazon S3, aunque no sea necesario, el archivo es mejor guardarlo en el directorio /tmp de la instancia Amazon EC2.

```
$s3 = new S3();  
$s3->createBucket(BUCKET);  
move_uploaded_file(  
$_FILES['userfile']['tmp_name'], "/tmp/" . $_FILES['userfile']['name'] );  
chmod( "/tmp/" . $_FILES['userfile']['name'], 0777 );  
$attempt = $s3->putObject( BUCKET,  
$_FILES['userfile']['name'], "/tmp/" . $_FILES['userfile']['name'], true);
```

Tras realizar este paso, es el momento de conectarse a Amazon SimpleDB y crear el dominio que contendrá los datos, en este caso se utilizarán las constantes que contienen la clave de acceso y la contraseña de Amazon.

```
$sd = new SimpleDb(AWS_ACCESS_KEY_ID, AWS_SECRET_ACCESS_KEY);  
$sd->createDomain(DOMAIN);
```

Una vez realizado este paso, se agregan los datos como atributos, entonces se encapsulan con el nombre del dominio y la cadena aleatoria como clave única para la tarea. Este paso da como resultado un registro de la ruta de archivo que se almacena en Amazon S3, el nombre y el correo electrónico del cliente externo, la tarea a realizar (en este caso sólo se realiza la miniatura) y el estado de la tarea (el estado inicial es de “no iniciado”).

```
$data["fullname"] = array($_POST['fullname']);  
$sd->putAttributes(DOMAIN,$random,$data);  
$data["email"] = array($_POST['emailaddress']);  
$sd->putAttributes(DOMAIN,$random,$data);  
  
$data["status"] = array('not started');  
$sd->putAttributes(DOMAIN,$random,$data);  
  
$data["path"] = array($_FILES['userfile']['name']);  
$sd->putAttributes(DOMAIN,$random,$data);  
  
$data["task"] = array($_POST['task']);  
$sd->putAttributes(DOMAIN,$random,$data);
```

En el siguiente paso se procede a crear una cola de Amazon SQS y el mensaje, este mensaje es muy sencillo, ya que contiene la cadena aleatoria que se generó anteriormente. Crear la cola y enviar un mensaje se realiza con el siguiente código:

```
$q = new SQSClient(AWS_ACCESS_KEY_ID, AWS_SECRET_ACCESS_KEY,
SQS_ENDPOINT);
try{
    $result = $q->CreateQueue(SQS_Q);
    //echo 'Queue Created: ', $result, "\n<br />\n";
}catch(Exception $e){
    throw($e);
}

$messageId = $q->SendMessage(urlencode($random));
```

Una vez realizada la cola en Amazon SQS, se envía un correo electrónico al cliente externo confirmando que se ha recibido la foto con la información de la imagen que va a ser procesada.

```
$to = $_POST['emailaddress'];
$from = 'test@example.com';
$subj = "Tu imagen se ha cargado correctamente!";
$msg = "Usted recibirá un correo electrónico una vez que el archivo haya sido
procesado.\r\n";
$msg .= "Nombre Introducido: " . $_POST['fullname'] . "\r\n"; $msg .= "MessageID: " .
$messageId . "\r\n";
$msg .= "MessageID: " . $messageId . "\r\n";
$msg .= "Bucket: " . $random;
mail($to, $subj, $msg, "From:$from\r\n");

header('Location:process.php');
```

4.2.3 Procesado del archivo

Una vez mandado el primer correo al cliente, la aplicación realizará las siguientes tareas: se coge un mensaje de la cola de Amazon SQS y utiliza el cuerpo de dicho mensaje para buscar el archivo que se va a procesar de la base de datos Amazon SimpleDB, se procesa dicho archivo y por último se envía el archivo ya procesado al correo electrónico al cliente.

En este archivo primero se inicia en Amazon SQS y se quita el mensaje de la cola. Cuando tenemos ese mensaje, se saca el cuerpo y el Receipt Handle para eliminar el mensaje.

```
$q = new SQSClient(AWS_ACCESS_KEY_ID, AWS_SECRET_ACCESS_KEY,
SQS_ENDPOINT, SQS_Q);
$nextMessage = $q->ReceiveMessage(1);
if (count($nextMessage)){
foreach($nextMessage as $message){
    $BUCKET = urldecode($message->Body);
    $handle = $message->ReceiptHandle;
}
}
```

Una vez hecho esto, pasamos a iniciar sesión en Amazon SimpleDB, y a continuación se recorren los dominios disponibles, para encontrar la clave del elemento que coincide con el cuerpo que se retiró del mensaje de la cola. Tras encontrar las coincidencias, se guardan los atributos en el objeto \$attr.

```
$domains = $sd->listDomains();
foreach ($domains->ListDomainsResult as $domains){
    foreach ($domains as $id => $d_name){
        if ($d_name == DOMAIN){
            $mydomain = $sd->query($d_name);
            foreach ($mydomain->QueryResult as $items){
                foreach ($items as $itemid => $_name){
                    if ($_name == $BUCKET){
                        $attr = $sd->getAttributes($d_name,$_name);
                    }
                }
            }
        }
    }
}
}
```

Ahora mediante un bucle se va sacando la información del atributo \$attr necesaria para el procesado de la imagen, se obtiene la dirección de correo electrónico, una tarea y una ruta de archivo, con esta información ya se sabe con qué bucket se va a trabajar, coincidiendo con el valor que se sacó del cuerpo de Amazon SQS.

```
foreach ($attr->GetAttributesResult as $attribute){
    foreach ($attribute as $array){

        if ($array->Name == "email"){
            $EMAIL = $array->Value;
        }
        if ($array->Name == "path"){
            $OBJECT = $array->Value;
        }

        if ($array->Name == "task"){
            $TASK = $array->Value;
        }
    }
}
```

A continuación se procede a iniciar sesión en Amazon S3, a descargar el archivo mediante \$OBJECT y \$CUBE, creando la imagen en miniatura y con la posibilidad de subir la imagen de nuevo a Amazon S3, sobrescribiendo la imagen subida anteriormente.

La función redimensionar() es una función simple que crea la miniatura de una imagen y que toma tres argumentos: el archivo para procesar, el ancho de archivo de salida, y la altura del archivo de salida. La función hace uso de herramientas GD que se encargan de la manipulación de imágenes.

```
$s3 = new S3();
$s3->downloadObject($BUCKET,$OBJECT,"/tmp/".$OBJECT);
$miruta = '/tmp/'. $OBJECT;
redimensionar($miruta, 100,100);
$upload = $s3->putObject( $BUCKET, $OBJECT, '/tmp/'. $OBJECT, true);
```

La función redimensionar() tendrá el siguiente código:

```
function redimensionar($ruta,$ancho,$alto){
//Obtenemos las dimensiones, la ruta es absoluta.
    $dim = getimagesize($ruta);

    if($dim[1]){
        //Para asegurarnos de que dim[1] es diferente de cero
        $cociente = $dim[0] / $dim[1];
    }
    if($alto){
        //Para asegurarnos de que alto es diferente de cero
        $coc_max = $ancho / $alto;
    }

    if(($dim[0]<=$ancho)&&($dim[1]<=$alto)){

        $ancho = $dim[0];
        $alto = $dim[1];
    }else{
        if($cociente>=$coc_max){
            echo "imagen igual\n\t";
            $alto = $ancho / $cociente;
        }else{
            echo "imagen nueva\n\t";
            $ancho = $alto * $cociente;
        }
    }
    $sim= imagecreatetruecolor($ancho,$alto);
    $sim = imagescale($ruta, $ancho, $alto, IMG_BICUBIC_FIXED);
    imagejpeg($sim,$ruta,100);
    imagedestroy($sim);
    imagedestroy($ruta);
}
```

Una vez creada la miniatura, se envía por correo electrónico el archivo de imagen al cliente. Dentro del correo se incluye una dirección URL que contiene el enlace a un archivo llamado retrieve.php, cuando se envía el correo, se procede a eliminar el mensaje Amazon SQS:

```
$URL = "http://your-ec2-  
address.amazonaws.com/retrieve.php?b=".$BUCKET."&o=".urlencode($OBJECT);  
$to = $EMAIL;  
$from = 'test@example.com';  
$subj = "Imagen Procesada!";  
$msg = "Su archivo ($OBJECT) está listo. Por favor, diríjase a:\r\n $URL\r\n para  
recibir su imagen.\r\n";  
mail($to, $subj,$msg, "From:$from\r\n");  
$q->DeleteMessage($handle);  
  
header('Location:thanks.php');  
} //end if(count($nextMessage))
```

Si se desea se puede redirigir el usuario a una página de agradecimiento, pues es un archivo HTML muy sencillo. El archivo "thanks.php" podría ser de esta manera:

```
<h1> Gracias! </ h1>  
<p> Recibirá un correo electrónico en breve confirmando que hemos recibido su  
archivo y su información. En unos momentos, usted recibirá otro email cuando su  
archivo haya sido procesado. </ p>
```

4.2.4 Recuperación del archivo procesado

En el apartado anterior, se enviaba en un correo electrónico la URL con un enlace a un archivo llamado retrieve.php, pues bien, este archivo se encarga de tomar los parámetros GET entrantes del bucket y objeto para la visualización de la imagen en el navegador.

El bucket y la información de los objetos se van a guardar en una sesión de PHP para que se pueda reutilizar de forma segura para su correcta eliminación y por último se podrá quitar el bucket y objetos de Amazon S3 y el elemento de Amazon SimpleDB.

```
session_start();
require_once 'Crypt/HMAC.php';
require_once 'HTTP/Request.php';
require_once('class.s3.php');
require_once('simpledb.class.php');

define('DOMAIN','photo_jobs'); //sdb
define('AWS_ACCESS_KEY_ID', 'change-this');
define('AWS_SECRET_ACCESS_KEY', 'change-this-too');
$BUCKET = $_GET['b'];
$OBJECT = $_GET['o'];

//guardar en sesión para su posterior limpieza
$_SESSION['o'] = $OBJECT;
$_SESSION['b'] = $BUCKET;
$s3 = new S3();
$file = $s3->getObject($BUCKET,$OBJECT);

echo '';
echo "<p>Descarga la imagen en tu dispositivo y
<a href='./cleanup.php'>finaliza el proceso elimin&ndola</a>.";

$sd = new SimpleDb(AWS_ACCESS_KEY_ID, AWS_SECRET_ACCESS_KEY);
$names = array('fullname', 'email', 'status', 'path', 'task');
$sd->deleteAttributes(DOMAIN,$BUCKET, $names);
```

4.2.5 Finalización del procesado de imágenes

Cuando el usuario pulsa en el enlace “Finalizar el proceso de limpieza”, se abre el archivo cleanup.php, este archivo se encarga de iniciar dos sesiones en PHP, coger las dos variables que tienen los valores del bucket y objeto y pasarlo a las funciones que se encargan de eliminarlos.

```
session_start();
require_once 'Crypt/HMAC.php';
require_once 'HTTP/Request.php';
require_once('class.s3.php');define('AWS_ACCESS_KEY_ID', 'change-this');
define('AWS_SECRET_ACCESS_KEY', 'change-this-too');

//1.Proceso GET de las variables
$BUCKET = $_SESSION['b'];
$OBJECT = $_SESSION['o'];

//2. Eliminar el cubo y objetos de S3
$s3 = new S3();
$s3->deleteObject($BUCKET, $OBJECT);
$s3->deleteBucket($BUCKET);

echo "<h1>Gracias</h1>";
echo "<p>Tus archivos han sido eliminados.</p>";
```

4.3 CONFIGURACIÓN DE LA INSTANCIA EC2

Para que la aplicación funcione correctamente, hay que realizar algunas configuraciones de la instancia Amazon EC2, se da por hecho que el usuario que va a desarrollar la aplicación está inscrito en Amazon EC2 y en los servicios S3, junto con las claves tanto públicas como privadas para poder manejar los servicios.

Necesitaremos un servidor LAMP (Linux, Apache, MySQL, PHP) para que funcione la aplicación en la instancia además de que PHP pueda permitir las librerías GD y PEAR para el funcionamiento las funciones de procesado de imagen.

4.3.1 Configuración del servidor LAMP

La instalación del servidor web Apache con PHP y MySQL en la instancia Amazon EC2 puede ser útil para alojar una página web estática o desplegar una aplicación PHP. El procedimiento para realizar esta instalación es el siguiente:

En primer lugar, la instancia EC2 debe estar en marcha con la DNS pública desde Internet, además debe estar configurado el grupo de seguridad para permitir tráfico SSH (puerto 22), HTTP (puerto 80) y HTTPS (puerto 443).

Las actualizaciones del software deben estar al día, para realizar este paso, hay que escribir esta línea de comandos:

```
$ sudo yum update -y
```

Se procede a instalar a continuación el servidor web Apache, MySQL y los paquetes de software PHP, para ello escribiremos la siguiente línea de comandos:

```
$ sudo yum install -y httpd24 php56 mysql55-server php56-mysqlnd
```

Ahora se inicia el servidor web Apache con el siguiente comando y a continuación introducimos el comando `chkconfig` para configurar el servidor web Apache y después ponemos en estado on los niveles de ejecución 2, 3, 4 y 5 de `httpd`:

```
$ sudo service httpd start  
$ sudo chkconfig httpd on  
$ chkconfig --list httpd httpd 0:off 1:off 2:on 3:on 4:on 5:on 6:off
```

Ya se puede poner a prueba el servidor web, para ello se introduce la DNS pública de la instancia, y se debería poder ver la página de prueba de Apache.

La página de prueba de Apache aparecerá cuando el contenido del directorio raíz del servidor esté vacío, la raíz de Apache en Amazon Linux es /var/www/html, que es propiedad de root por defecto.

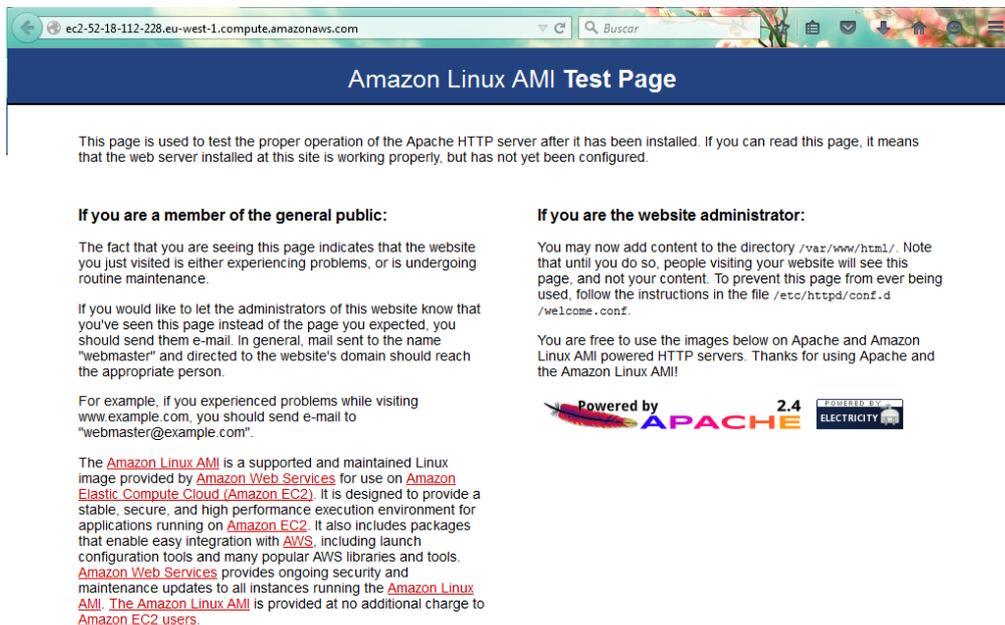


Figura 57: Servidor Web Apache en Amazon Linux

Si queremos manipular archivos en este directorio, es necesario modificar la propiedad y los permisos de éste, existen muchas maneras de poder realizar esta tarea, pero en este caso se agrega un grupo www a la instancia y se añaden permisos para el usuario, sólo hay que escribir las siguientes líneas de comandos en la instancia, para que los cambios sean efectuados, hay que cerrar la sesión y volver a entrar para poder ver el nuevo grupo:

```
$ sudo groupadd www  
$ sudo usermod -a -G www ec2-user
```

Una vez vuelto a iniciar la sesión, escribiremos las siguientes líneas de comandos, que nos sirven para cambiar la propiedad del grupo de /var/www y su contenido en el grupo, y cambiar los permisos de escritura y establecer el ID del grupo del directorio y subdirectorios. Así, el usuario de la instancia ec2-user puede añadir, eliminar y editar archivos en el directorio raíz de Apache.

```
$ sudo chown -R root:www /var/www  
$ sudo chmod 2775 /var/www  
$ find /var/www -type d -exec sudo chmod 2775 {} +  
$ find /var/www -type f -exec sudo chmod 0664 {} +
```

Para probar que el servidor Apache funciona, se va a crear un archivo PHP sencillo en el directorio raíz de Apache:

```
$ echo "<?php echo 'Hola'; ?>" > /var/www/html/hola.php
```

A continuación, y desde el navegador web, introducimos la URL correspondiente a este archivo, este enlace está formado por la DNS pública de nuestra instancia y el documento PHP.

```
http:// my.public.dns.amazonaws.com/hola.php
```

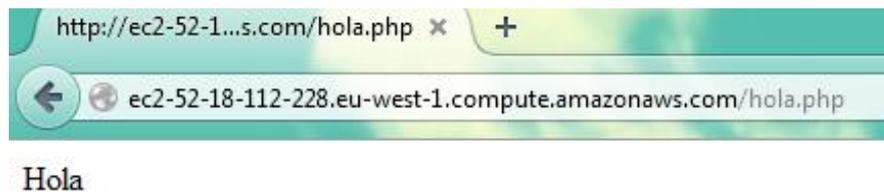


Figura 58: Prueba del servidor Apache

Por último, la instalación por defecto del servidor MySQL tiene varias características que son ideales para las pruebas y el desarrollo, pero se debe desactivar o quitar para los servidores de producción. Aunque no se vaya a usar el servidor MySQL, la realización de este procedimiento no es mala idea.

Primero iniciamos el servidor MySQL y después ejecutamos `Mysql_secure_installation`:

```
$ sudo service mysqld start  
$ sudo mysql_secure_installation
```

Cuando ejecutamos la segunda línea de comandos, nos pedirá la contraseña root, por defecto no tiene, así que pulsaremos intro, y automáticamente pulsaremos Y para establecer una contraseña segura, escribiéndola dos veces. Tras crear la nueva contraseña pulsaremos cuatro veces Y para eliminar las cuentas de usuario anónimo, para desactivar a distancia root de inicio de sesión para eliminar la base de datos de prueba y para recargar las tablas de privilegios y guardar los datos.

Como tarea opcional, podemos arrancar MySQL si se desea con el comando:

```
$ sudo chkconfig mysqld on
```

Con este último paso ya hemos creado un servidor LAMP web que funciona correctamente. Si se agrega contenido al directorio raíz de Apache, el usuario debería ser capaz de ver el contenido en la dirección pública DNS para la instancia.

4.3.2 Transmisión de archivos a una instancia Amazon EC2

Ya tenemos listo todo el sistema para añadir nuestra aplicación y probar su funcionamiento, el código de dicha aplicación en primer lugar se tiene en el computador del usuario y para que comience a estar disponible es necesario transmitir los archivos a la instancia ya creada.

Hay múltiples maneras de transmisión de los archivos a la instancia. La forma que se ha escogido para este proyecto es la transferencia de archivos desde Windows a la instancia Amazon EC2 utilizando la herramienta WinSCP.

WinSCP es una herramienta de gestión de archivos basado en GUI (Interfaz Gráfica de Usuario) para el sistema operativo Windows que permite la carga y transferencia de archivos a un terminal remoto a través de los protocolos SFTP, SCP, FTP y FTPS. Esta herramienta permite arrastrar y soltar los archivos que se desean desde el equipo local para la instancia de Linux o sincronizar estructuras completas de directorios entre los dos sistemas.

Los únicos requisitos que se necesitarán para poner en funcionamiento WinSCP será tener la dirección pública DNS de la instancia de Linux y la clave privada en el formato .pkk (en el apartado 3.2 se menciona cómo convertir la clave privada .pem a .pkk utilizando PuTTYgen).

Los pasos a realizar son también muy sencillos e intuitivos, en primer lugar descargamos, instalamos e iniciamos la herramienta WinSCP. La descarga es gratuita y contiene múltiples idiomas a elegir para su funcionamiento si se desea.

Al iniciar sesión en WinSCP, el usuario debe completar el nombre del host, es decir, la dirección pública DNS para la instancia. En el nombre de usuario, pondremos ec2-user ya que estamos utilizando una instancia de Amazon Linux.

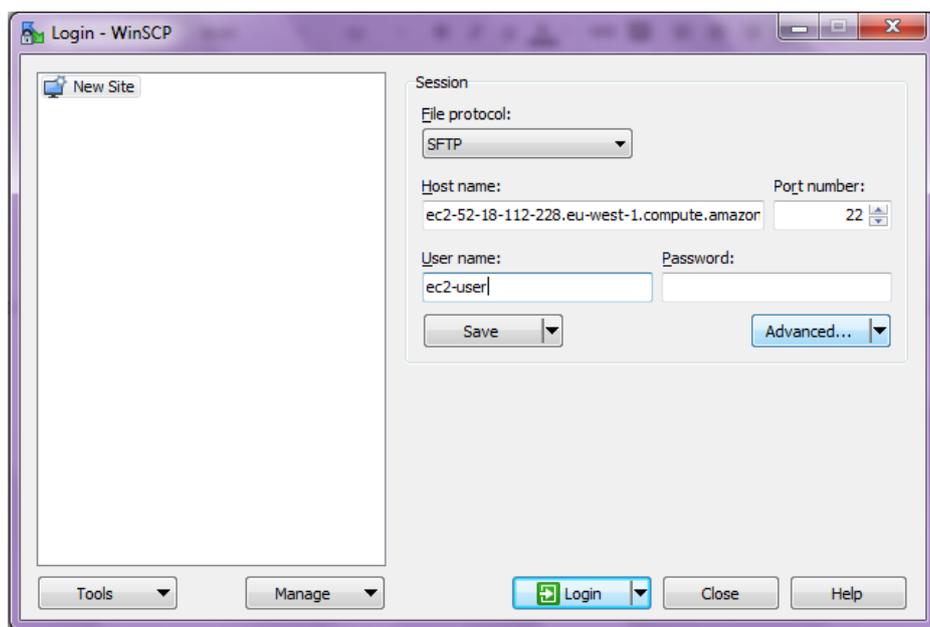


Figura 59: WinSCP

Una vez rellenos estos campos, pulsamos en “Advanced”. El cuadro de diálogo nos muestra las configuraciones avanzadas para la conexión entre la equipo local y el terminal remoto. Pulsamos en el panel de navegación en SSH e inmediatamente en Authentication, en la pantalla que nos aparece entonces se puede observar que hay un apartado en el que pide la clave privada y pulsamos en “...” para seleccionar el directorio en el que se encuentra nuestra clave privada en el formato .ppk.

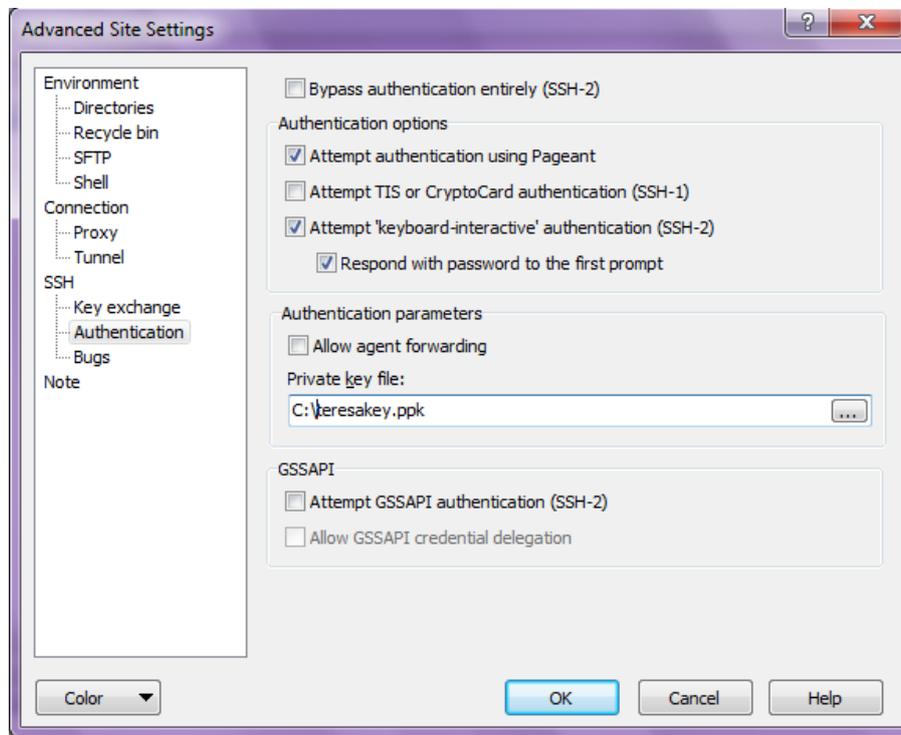


Figura 60: Clave privada en WinSCP

Desde el lugar en el que se encuentra, pulsamos en el panel de navegación en “Directories” y en el campo “remote directory” escribimos si lo deseamos el directorio raíz del servidor Apache (/var/www/html) para que nos resulte más cómoda la transferencia de archivos.

Una vez realizadas todas estas configuraciones pulsamos “Ok” para salir de las configuraciones avanzadas y hacemos clic en “Login” para establecer conexión entre las dos máquinas y poder transferir datos de un lugar a otro de forma sencilla.

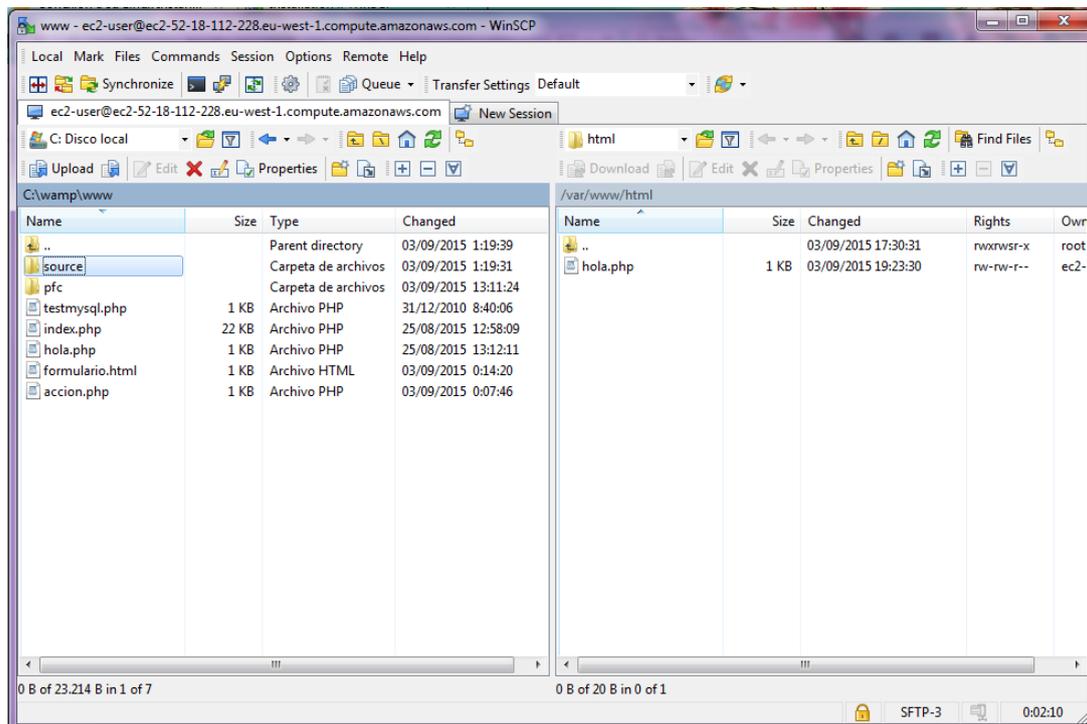


Figura 61: Transferencia de archivos con WinSCP

4.3.3 Prueba de la configuración realizada de la instancia

Para probar que funciona tanto la transmisión de archivos como nuestro servidor, hemos creado el archivo prueba.php en Windows, que pasaremos por WinSCP a la instancia Amazon Linux y probaremos el servidor para ver que dicho código funciona perfectamente.



Figura 62 : Prueba PHP en Amazon Linux

4.3.4 Instalación de librerías en la instancia

Para nuestra instancia EC2, y como se ha dicho al inicio de este apartado, es necesario la utilización de las librerías GD y PEAR para el correcto funcionamiento de la aplicación de procesamiento de imágenes, para ello, se han realizado los siguientes pasos:

En primer lugar, vamos a utilizar el sistema de administración e instalación de paquetes de software (RPM) de Remi. Para nuestra instancia instalaremos el Enterprise LINUX 5. Los comandos que se escriben serán los siguientes:

```
$ wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-5.noarch.rpm
$ wget http://rpms.remirepo.net/enterprise/remi-release-5.rpm
$ sudo rpm -Uvh remi-release-5.rpm
$ repomd http://rpms.remirepo.net/ fc$(VERSION).$(ARCH)
```

A continuación con el comando YUM, instalamos y habilitamos las librerías GD y PEAR:

```
$ sudo yum --enablerepo= remi php-gd
$ sudo yum --enablerepo= remi php-pear
```

Por último, y si repasamos el código de la aplicación, observamos que es necesario dos componentes de PEAR para el funcionamiento de varios documentos PHP que tenemos (process.php, upload.php). Para la instalación de estos dos componentes sólo basta con escribir las siguientes líneas de comandos y el reinicio del servidor web Apache:

```
$ sudo pear install Crypt_HMAC
$ sudo pear install HTTP_Request
$ sudo apachectl -k restart
```

4.4 RESULTADOS OBTENIDOS

Una vez preparado la instancia y con el código de la aplicación, sólo falta ver el resultado que se ha obtenido de este gran proceso. Accederemos desde nuestro navegador al formulario HTML de la aplicación para que se procese una imagen deseada:



http://ec2-52-17-96-19.eu-west-1.compute.amazonaws.com/index.php

Creador de Miniaturas

Elija una imagen para cargar upct.jpg

Su Nombre:

Su Email:

Crear miniatura

Figura 63: Procesador de Miniaturas

Una vez completados los campos del formulario, pulsamos sobre la opción de crear miniatura y cargamos el archivo. A partir de este momento, comienza todo el proceso que realiza la aplicación para procesar la imagen, aunque al cliente se le redireccionará a una página de agradecimiento e información sobre el proceso de su imagen:



http://ec2-52-17-96-19.eu-west-1.compute.amazonaws.com/thanks.php

Gracias!

Usted recibirá un correo electrónico en breve con la información de confirmación.

En breves, Usted va a recibir otro correo cuando su archivo haya sido procesado.

Figura 64: Thanks.php

A partir de este momento, el cliente debe esperar a la llegada de dos Emails en la bandeja de entrada de su correo electrónico, y en caso de no recibir dichos correos, es posible que se encuentren en la carpeta de "correo no deseado". En el primer correo se visualizará la información que la cola SQS le ha asignado a la imagen, es decir, el identificador que lleva la imagen y el cubo ("Bucket") en el que se guarda.

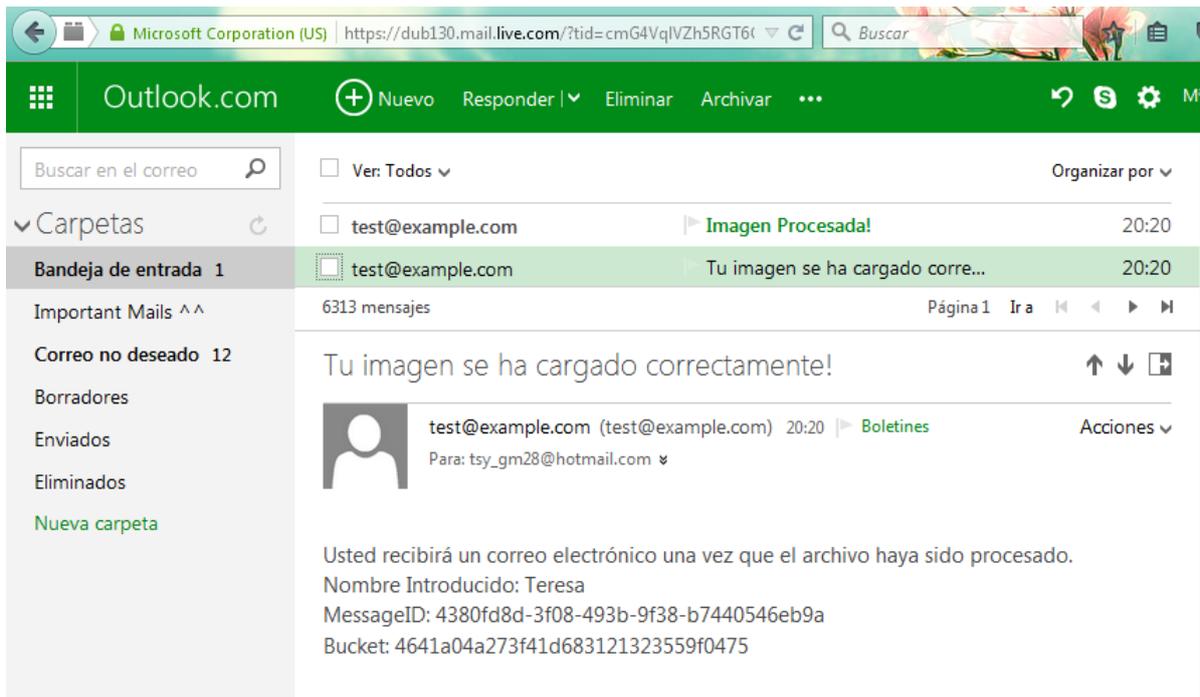


Figura 65: Imagen Cargada en la aplicación

El segundo correo sin embargo, se envía cuando la imagen es procesada y en este correo se incluye el enlace a la URL del archivo que se encuentra ubicado en una página php llamada Retrieve.php:

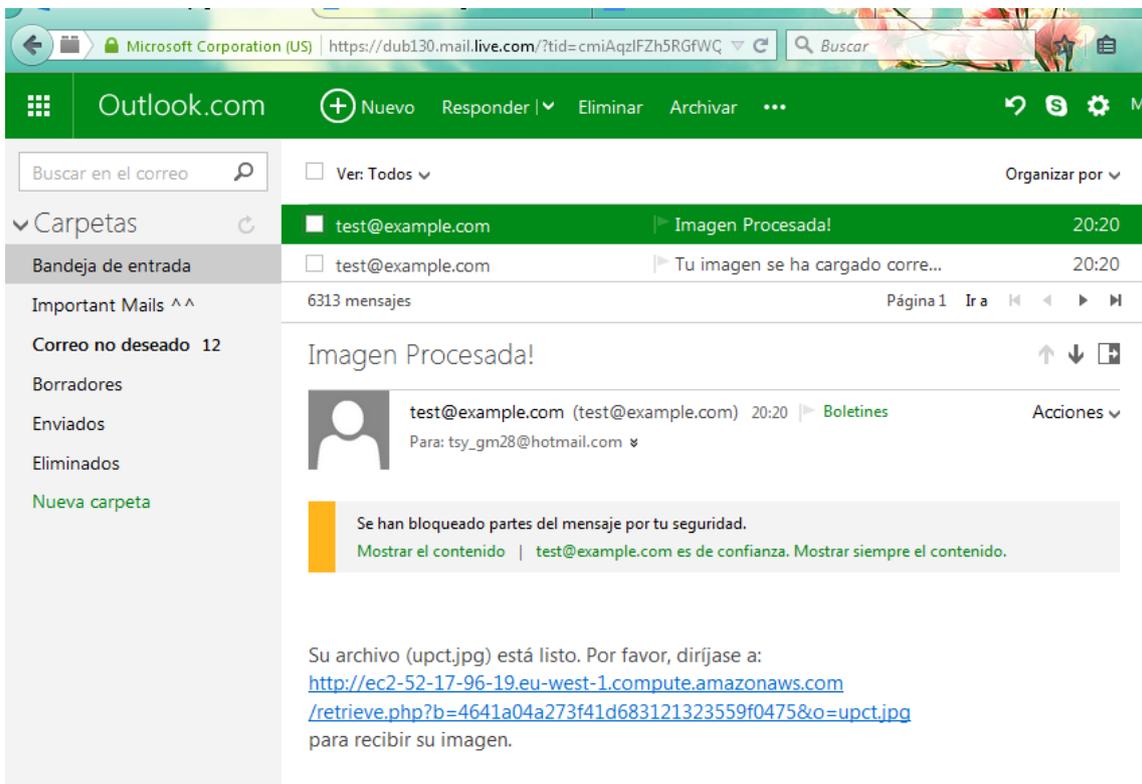


Figura 66: Imagen Procesada

El cliente sólo tendrá que pulsar en la dirección del enlace que aparece en el mensaje para ir a la página y descargarse su archivo ya procesado, una vez hecho esto, el propio usuario elimina el archivo pulsando sobre el texto de “finalizar el proceso” y se direccionará a una página que mostrará en texto que todos los datos han sido eliminados.



Figura 67: Imagen para descargar

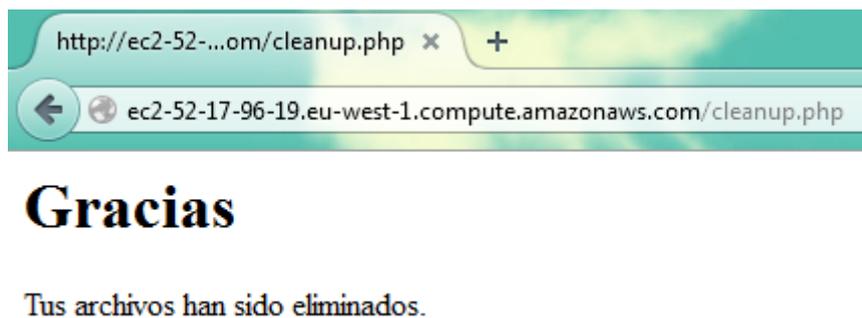


Figura 68: Archivo eliminado

PRESUPUESTO

Para la realización de este proyecto final de carrera no ha sido necesario el uso de dinero ya que se ha utilizado la versión gratuita que ofrece Amazon Web Services con un periodo de prueba de un año.

No obstante, el código de la aplicación que se ha utilizado requería el Sistema Operativo Fedora y una instancia de tipo M1, requisitos que no están incluidos en dicha versión de prueba.

Para controlar los gastos de elementos como las instancias, volúmenes y servicios que Amazon Web Services ofrece, existe una herramienta de calculadora que ayuda a calcular el precio mensual de los servicios y herramientas de AWS que vamos a utilizar. Esta calculadora se llama “Amazon Web Simple Monthly Calculator” y su uso es bastante intuitivo y sencillo, sólo hay que rellenar la información de los elementos que vamos a utilizar y en la pestaña adyacente encontraremos la factura que se genera automáticamente. Además si el usuario se encuentra en el periodo de prueba cuando realiza estos gastos, se realizan descuentos.

Description	Instances	Usage	Type	Billing Option	Mon Cost
Linux on t2.micro	1	100 % Utilized/Mo	Linux on t2.micro	On-Demand (No Co	

Figura 69: Amazon Web Simple Monthly Calculator

Si queremos estimar el coste de una instancia de tipo micro que funcione sin parar y sin la versión de prueba gratuita que ofrece Amazon Web Services (las instancias de tipo micro son gratuitas durante este periodo de prueba) sólo tendremos que seleccionar el tipo de instancia micro a utilizar, y observando el coste por hora bajo demanda de cada instancia se calculará dicho coste. Para una instancia t1.micro el coste mensual es de 14.64\$ ya que el coste por horas bajo demanda es de 0.020\$, un coste 13.07€ en Europa. Si elegimos una instancia t2.micro, con coste por horas bajo demanda de 0.014\$, el coste mensual será de 10.25\$, 9.15€.

CONCLUSIONES Y LÍNEAS FUTURAS

6.1 CONCLUSIONES

La utilización del Cloud Computing y de los servicios web para el desarrollo de aplicaciones web ha marcado un gran cambio en todos los sentidos. Cualquier persona u organización puede desarrollar sus aplicaciones desde cualquier lugar del mundo pagando sólo por lo que se utiliza, ahorrando gastos tanto en compra de hardware e instalación de software como el mantenimiento del mismo pudiendo abastecer así, a clientes de cualquier zona del planeta.

El proceso para la realización del entorno web que se va a utilizar es bastante intuitivo gracias a las herramientas que nos proporciona el proveedor web de Cloud Computing que se vaya a utilizar y que cuenta con múltiples opciones para seleccionar, como el sistema operativo o las herramientas para el desarrollo de la aplicación web, entre otros.

Con el uso de Amazon Web Services, queda demostrado que cualquier usuario utilizando algunos servicios como Amazon EC2, S3 y SQS puede construir aplicaciones web integradas y altamente escalables, utilizando técnicas tradicionales de desarrollo de aplicaciones web. La aplicación de procesado de imágenes es uno de los ejemplos que muestra una de las infinitas posibilidades que se pueden llegar a realizar.

Este proyecto sirve como guía de desarrollo de aplicaciones en Cloud Computing utilizando Amazon Web Services para cualquier persona, incluyendo aquellas sin conocimientos previos.

6.2 LÍNEAS FUTURAS

Una vez utilizado Amazon Web Services, podemos ver que para este proyecto final de carrera se han utilizado los servicios más destacados. La aplicación que se ha utilizado (procesador de imágenes) era bastante simple ya que sólo realiza la función de crear una miniatura de la imagen que cargamos, pero esta aplicación puede mejorarse y ampliarse, en el sentido de agregar nuevas funciones como volteo de imagen, poner filtro en escala de grises o recortar, etc, y poder realizar tanto una función individual como varias al mismo tiempo. También puede ser posible la mejora en la interfaz gráfica para que la aplicación web se vea más llamativa con el uso de Java o PHP.

El desarrollo de aplicaciones web en “La Nube” está siendo toda una revolución tecnológica, y evoluciona constantemente. Es posible que en un futuro no muy lejano cuando las personas enciendan cualquiera de los dispositivos que tenga al alcance (ordenador, tablet, smartphone, etc...) desde cualquier lugar del planeta puedan tener acceso a todos sus datos y desarrollar aplicaciones que se encontrarán en servidores de “La Nube” de forma muy segura, flexible, con tiempos de respuesta muy cortos, más escalables y con todo tipo de servicios disponibles, aumentando así la confianza por parte de los usuarios, y pasando a formar parte tanto de las personas como de las empresas.

GLOSARIO DE TÉRMINOS

A continuación se describe un glosario de términos específicos que se encontrarán a lo largo de este proyecto:

API: “Application Programming Interface” constituye el mecanismo más utilizado de comunicación entre aplicaciones. Es una interfaz que permite las aplicaciones de terceros, solicitar datos y tenerlos de vuelta con un formato predefinido y de acuerdo a las normas específicas.

AMI: “Amazon Machine Image” es un tipo especial de aplicación virtual que se utiliza para crear una máquina virtual dentro de Amazon EC2 (Amazon Elastic Compute Cloud). Además es la unidad básica de despliegue para los servicios prestados utilizando EC2.

Clúster: Conjunto de servidores que funcionan como una única máquina mejorando las operaciones y transacciones implantadas en un sistema.

CPU: “Unidad Central de Procesamiento” es el hardware dentro de una computadora u otros dispositivos programables, que interpreta las instrucciones de un programa informático mediante la realización de las operaciones básicas aritméticas, lógicas y de entrada/salida del sistema.

Data Center: centro de almacenamiento de datos y que provee servicios de negocio que entrega de forma segura aplicaciones y datos a usuarios remotos a través de Internet.

GPU: “Unidad de Procesamiento Gráfico” que sirve para aligerar la carga de trabajo del procesador central de aplicaciones como los videojuegos o aplicaciones 3D interactivas.

Hypervisor: o también llamado “Monitor de máquina virtual” (VMM) es una plataforma que permite aplicar distintas técnicas de control de virtualización para utilizar al mismo tiempo diferentes sistemas operativos en un mismo dispositivo.

Instancia EC2: Servidor virtual en Amazon Elastic Compute Cloud (EC2) para ejecutar aplicaciones en la infraestructura de Amazon Web Services. En Amazon Web Services hay múltiples tipos de instancias, las más utilizadas son las T2, que proporcionan un rendimiento base de la CPU con capacidad de alcanzar ráfagas por encima del nivel base.

LAMP: Describe un sistema de infraestructura de Internet que usa Linux (SO), Apache (Servidor web), MySQL (gestor de base de datos) y PHP, Perl o Python (lenguajes de programación). Este conjunto define una infraestructura de un servidor web.

Máquina Virtual: Software que simula un computador y que puede ejecutar programas como si fuese un ordenador físico, de modo que el Sistema Operativo o aplicaciones que corren sobre él no detectan la diferencia.

Multi-tenancy: Uso común entre todos los clientes y usuarios de los servicios de computación en la nube desde la misma plataforma tecnológica del proveedor contratado.

RPM: (Red Hat Package Manager) es una herramienta de administración de paquetes para GNU/Linux. Instala, actualiza, verifica y solicita programas.

RSA: Sistema criptográfico asimétrico de claves públicas, es el algoritmo de cifrado más utilizado.

Virtualización: Describe cómo coordinar el uso de los recursos para que varios sistemas operativos puedan funcionar al mismo tiempo de forma independiente y sin que haya conocimiento de que los recursos están siendo compartidos entre SO en un solo computador físico.

REFERENCIAS Y BIBLIOGRAFÍA

Amazon Web Services:

<http://aws.amazon.com>

Descarga de SDK PHP de Amazon Web Services:

<https://github.com/aws/aws-sdk-php/releases>

Programming Amazon EC2 de Jurg van Vliet y Flavia Paganelli.

Cloud Computing:

<http://www.cloudcomputingtoday.info/cloud-computing-hosting/>

<http://www.revistacloudcomputing.com/glosario-cloud-computing/>

<http://vmartinezdelacruz.com/algo-sobre-nubes/>

<http://nubeblog.com/2008/10/15/saas-iaas-y-paas-las-tres-clases-de-cloud-computing/>

Virtualización:

<http://www.virtualitze.com/es/virtualizacion.php>

<http://www.welivesecurity.com/la-es/2014/07/14/migrando-nube-bondades-peligros-virtualizacion/>

REMI RPM'S REPOSITORY:

<http://blog.famillecollet.com/pages/Config-en>

<http://www.ipackage.org/>

LAMP server:

<http://lamphowto.com/>

Google.com