14-12-2021

Miguel Angel Povea Ramirez

13/12/2021

PROYECTO ASO

AMAZON WEB SERVICES

Contenido

[INSTANCIA WINDOWS 1](#_Toc90395552)

[INSTANCIA LINUX 7](#_Toc90395553)

[PARTICIONES EN WINDOWS Y RAID 14](#_Toc90395554)

[AUTOESCALING 20](#_Toc90395555)

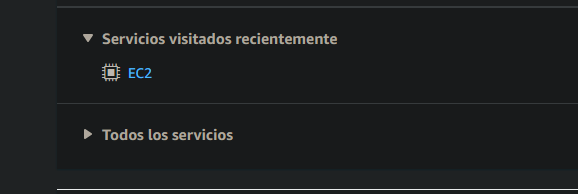
[BALANCEADOR DE CARGA 23](#_Toc90395556)

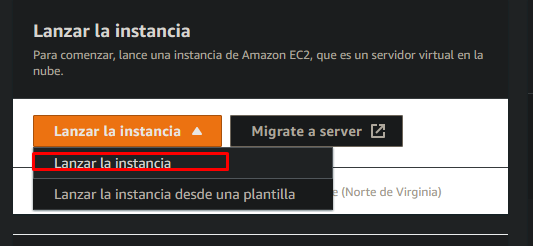
[CONCLUSION DE AMAZON WEB SERVICES 27](#_Toc90395557)

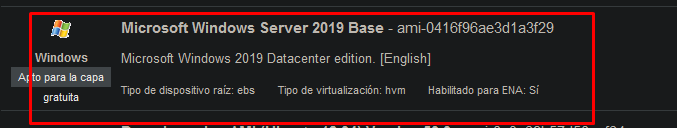
Hola, este es el proyecto de AMAZON WEB SERVICES en el vamos a usar la herramienta de amazon para gestionar un servidor remotamente, concretamente vamos a ver la creación de instancias, creación de RAID en discos duros y particiones. El autoescaling y por último el balanceador de carga.

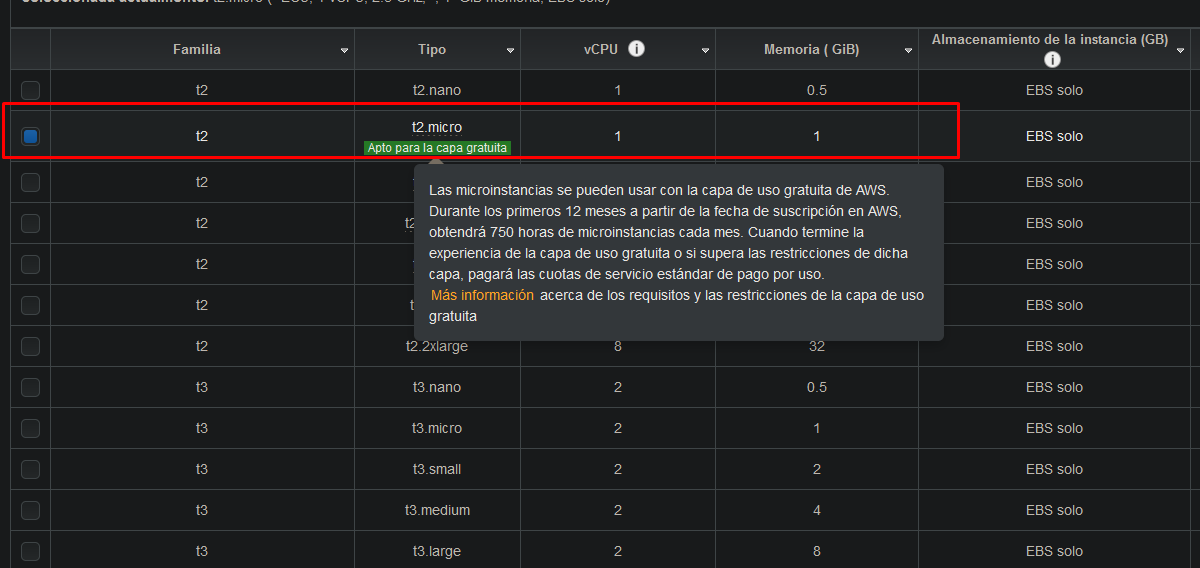
# INSTANCIA WINDOWS

Creación de instancia win2019 server.





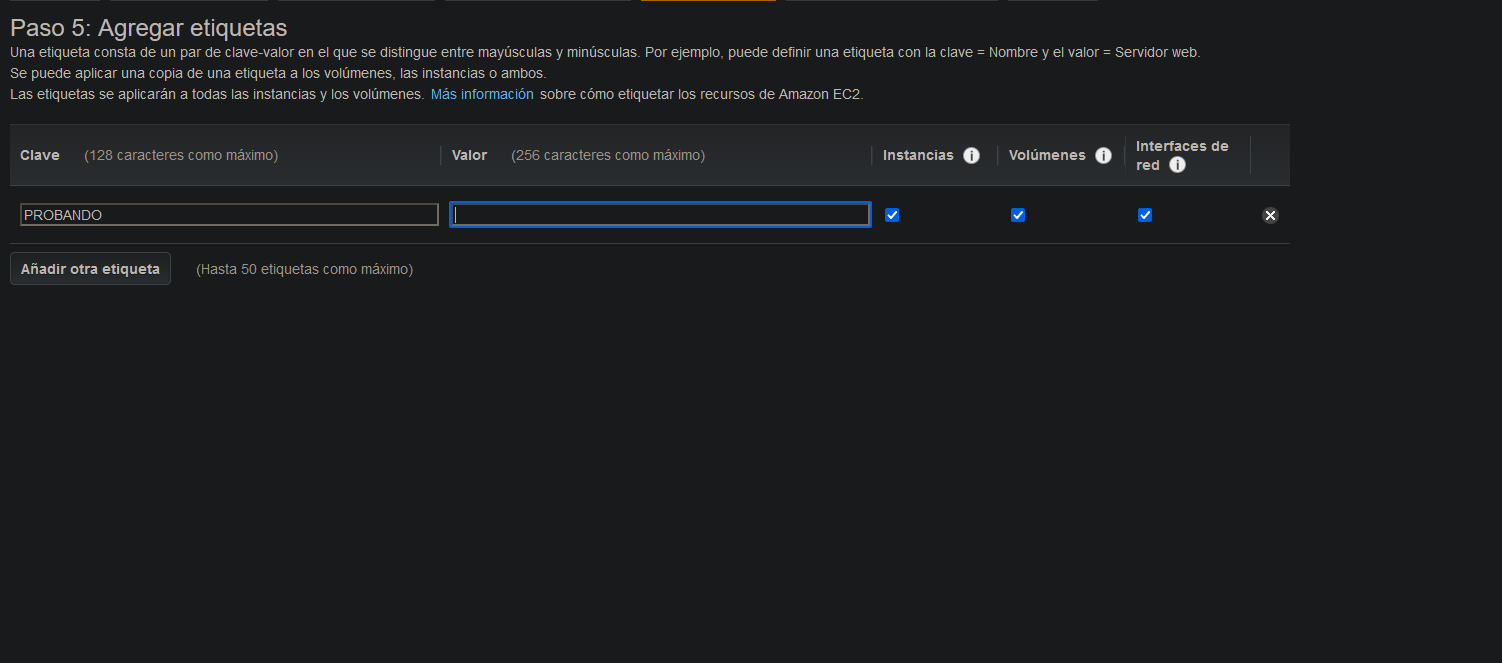




A continuación, la plataforma nos brinda la posibilidad de adición en el espacio de almacenamiento, pero vamos a rechazar dicha proposición.

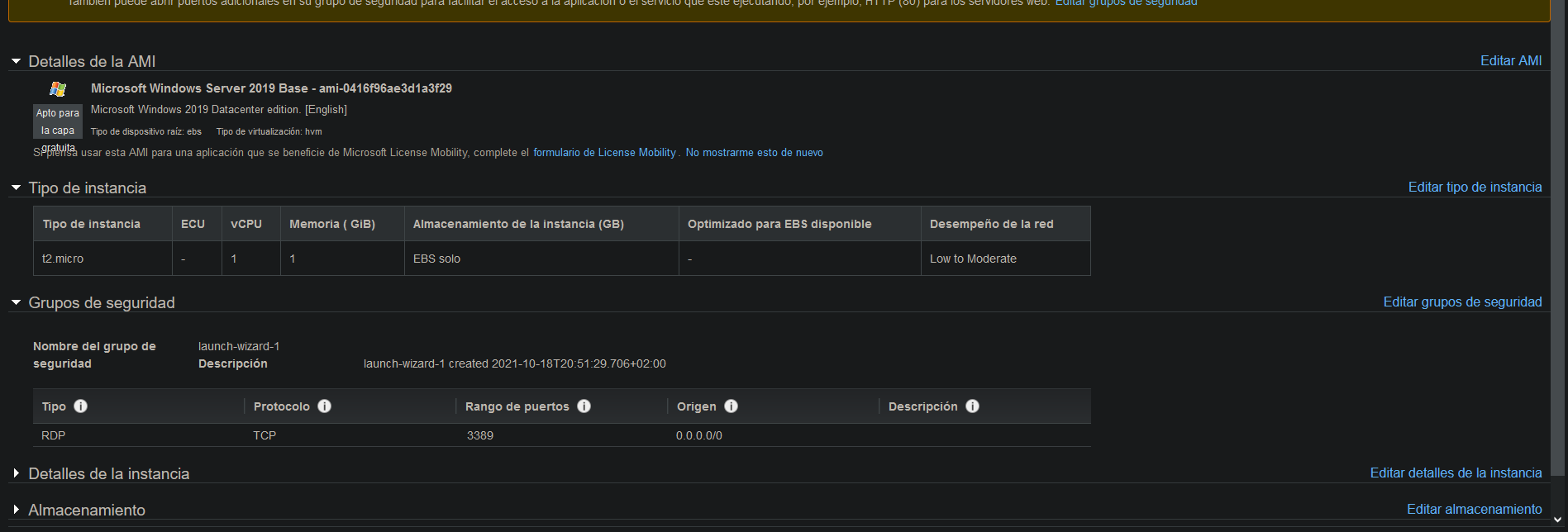


Añadimos un identificador del par de claves.

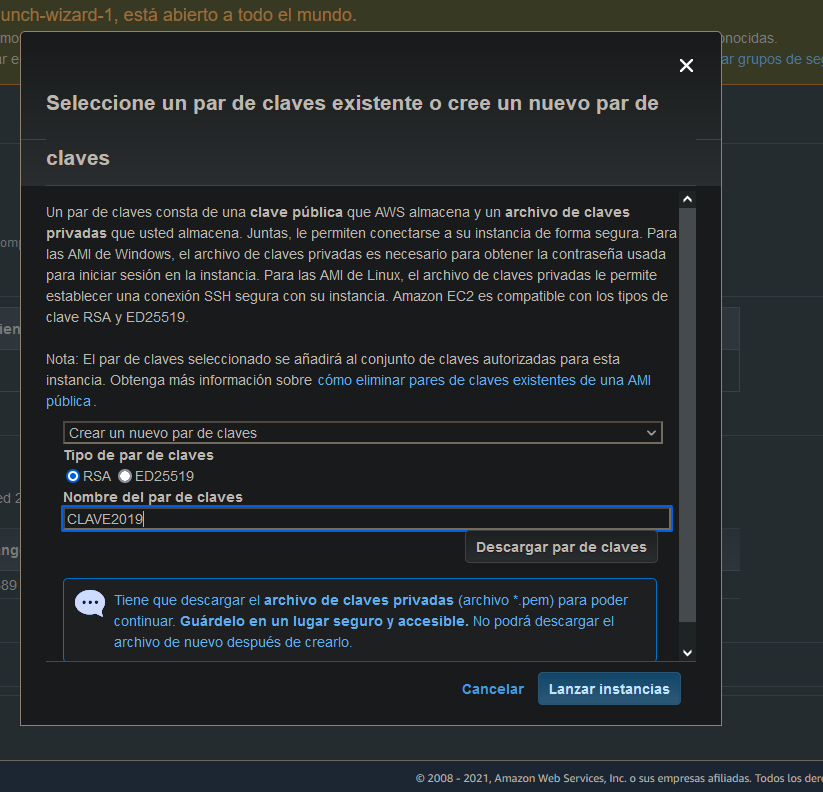


Durante el proceso de creación, la plataforma nos notifica de un posible error en la seguridad del cortafuegos, no obstante, vamos a obviar esta recomendación aunque se agradece deleitosa recomendación.

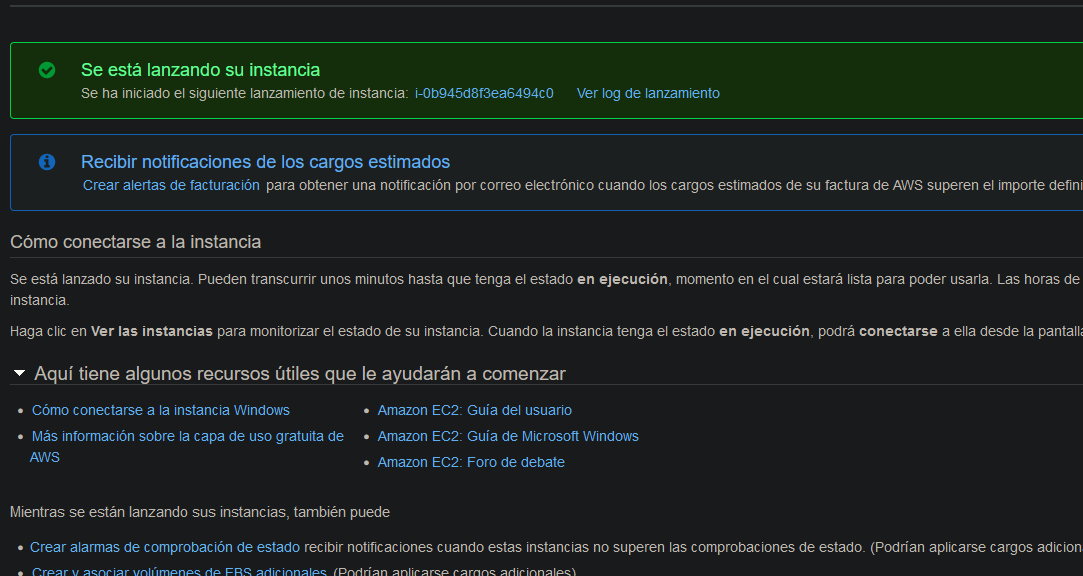




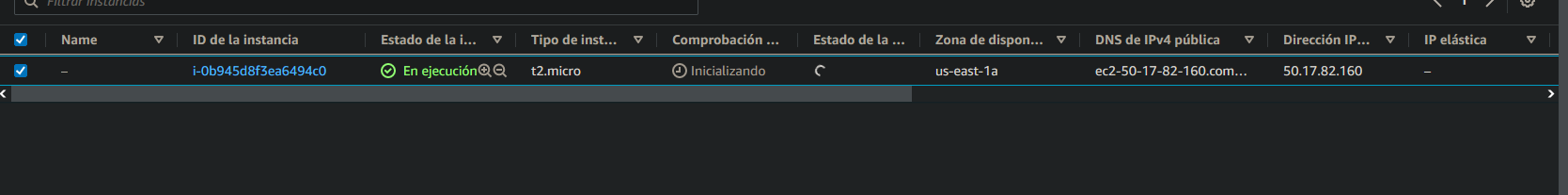
A continuación, generamos un par de claves RSA (Rivest–Shamir–Adleman). (Con lo que estamos dotando a la instancia de un buen nivel de seguridad). Las cuales podremos usar más adelante para acceder a dicha instancia.



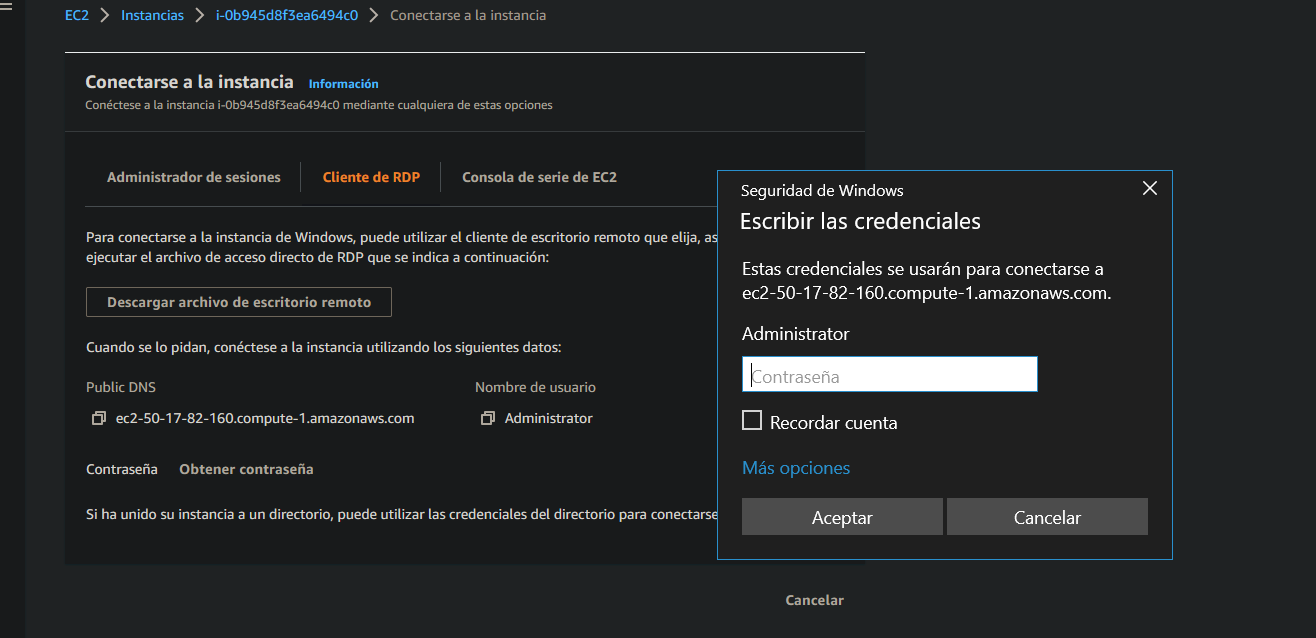
La página web nos devuelve un mensaje de que la instancia ha sido lanzada exitosamente.



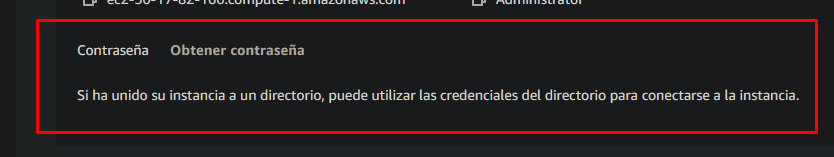
La instancia se encuentra en ejecución, tal y como podemos ver en el gestor de la misma.

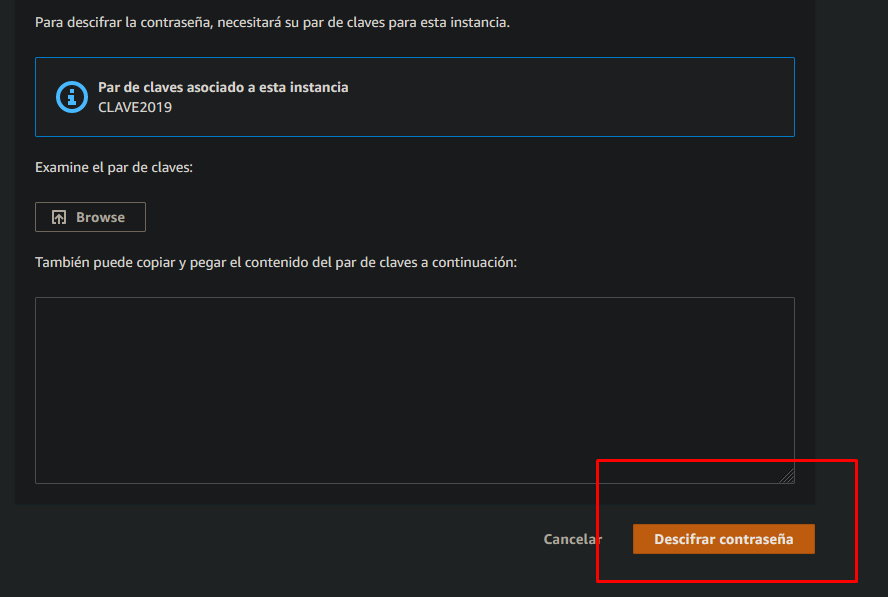


Accedemos a la instancia mediante su ip, nombre de usuario y desencriptando la contraseña gracias a la clave que generamos con anterioridad.

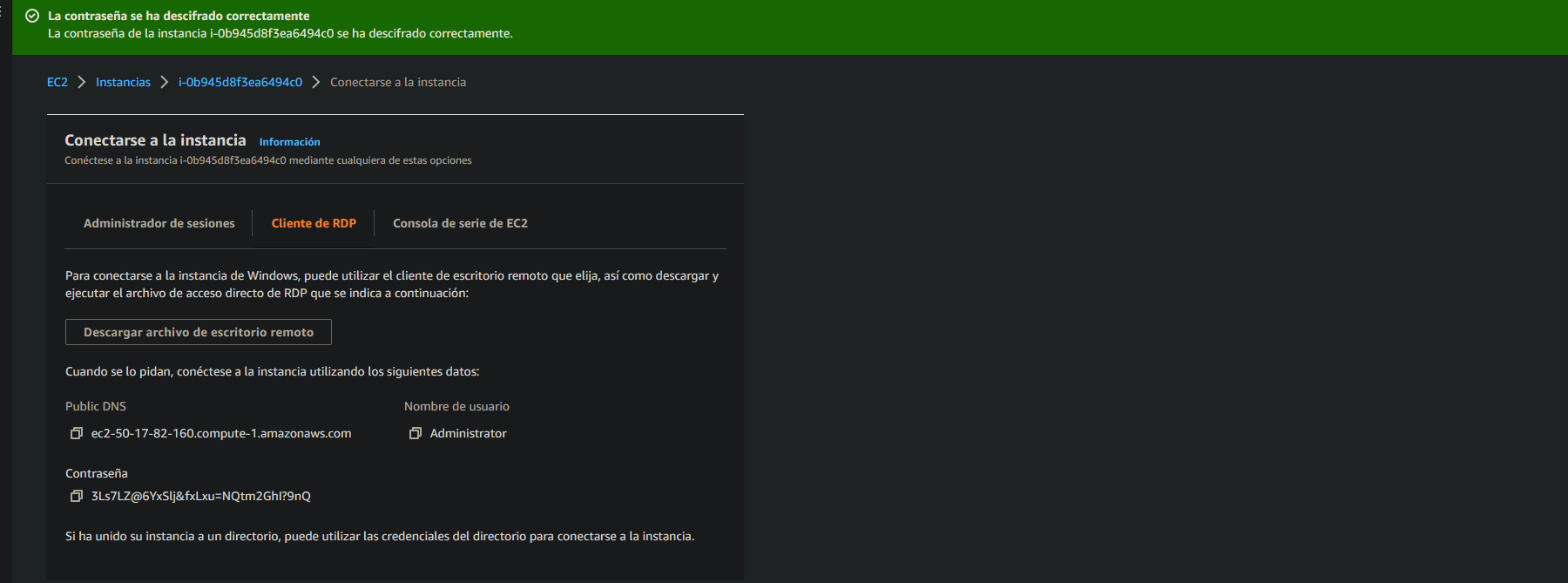


Para revelar la contraseña tras ser desencriptada, pulsamos en la opción “Mostrar contraseña”.

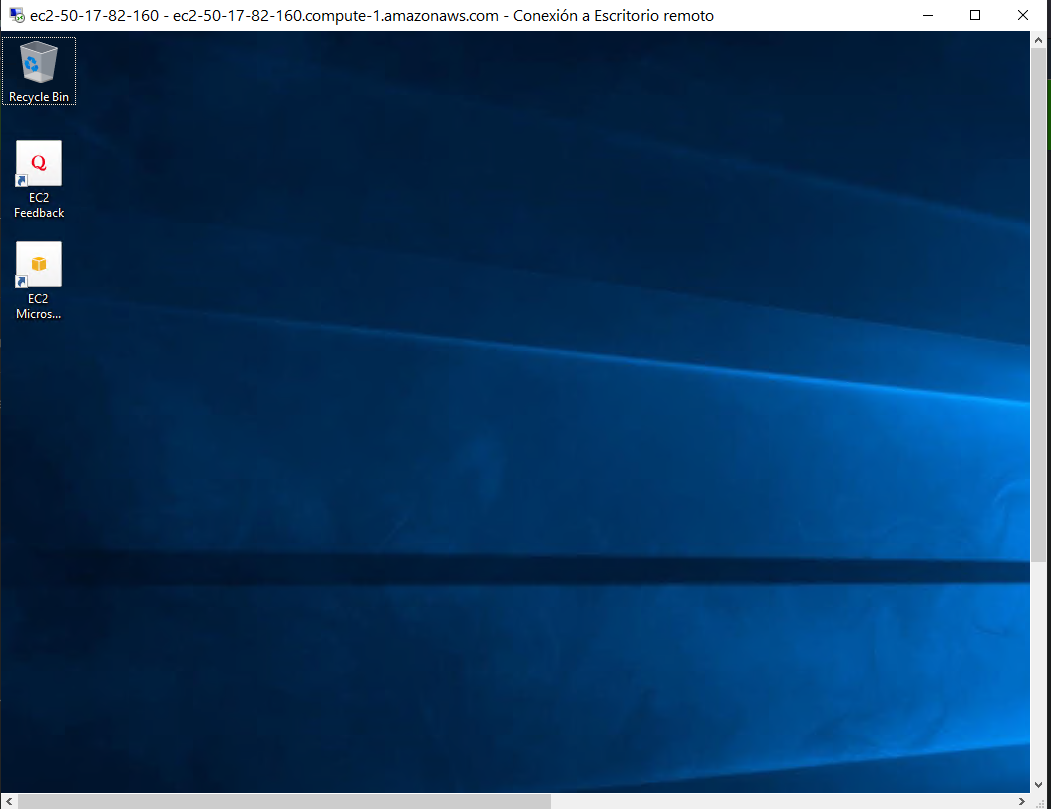




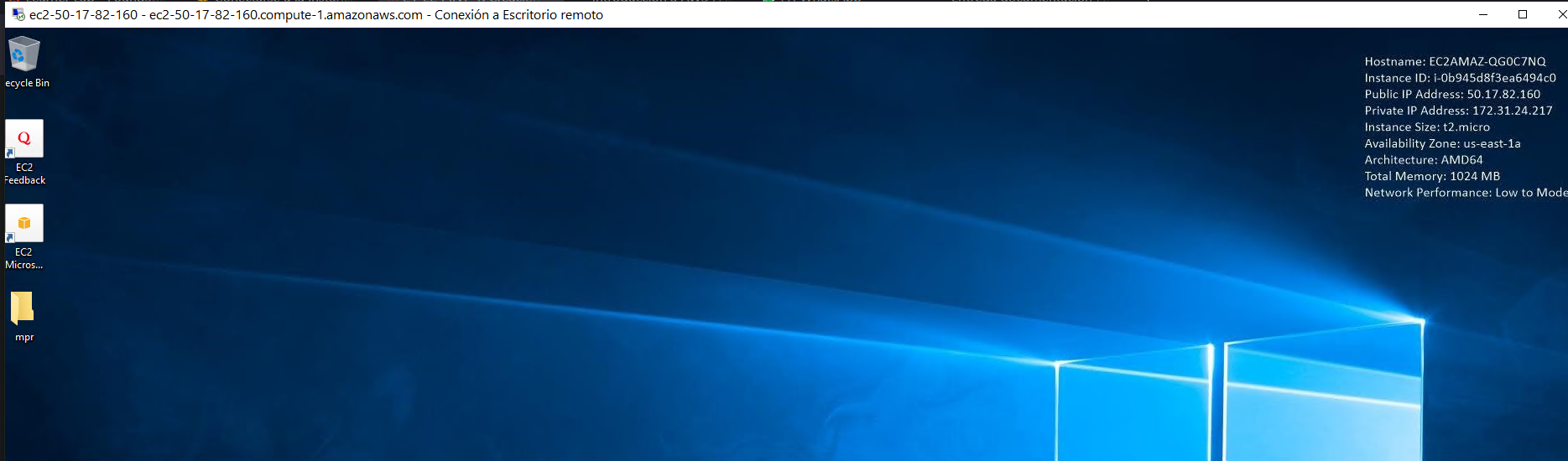
Introducimos la clave RSA, y pulsamos en “Descifrar contraseña”.



Como podemos observar en la captura que se muestra a continuación, la conexión con el servidor ha sido exitosa, puesto que ya nos encontramos en el escritorio de éste.



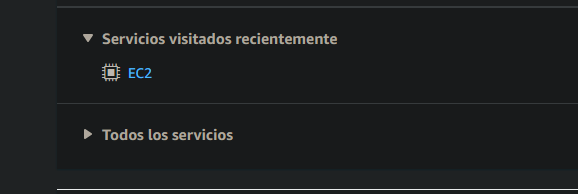
En la siguiente captura, podemos ver la carpeta creada, tal y como se nos pide en el enunciado del ejercicio.



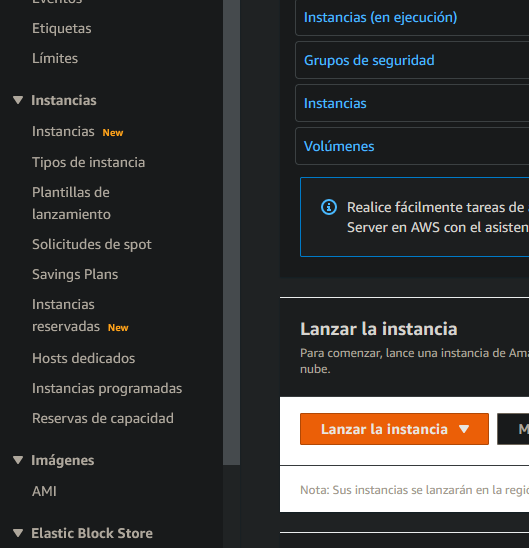
# INSTANCIA LINUX

Una vez accedemos a vocareum, debemos de hacer click en el botón que dice “AWS”.

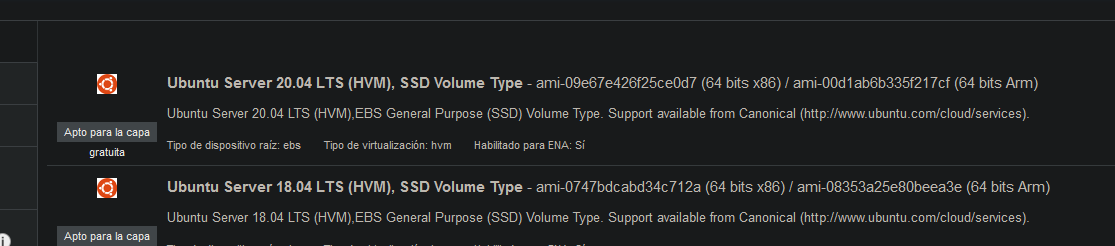
A continuación nos dirigimos a servicios, instancias y por último, EC2.

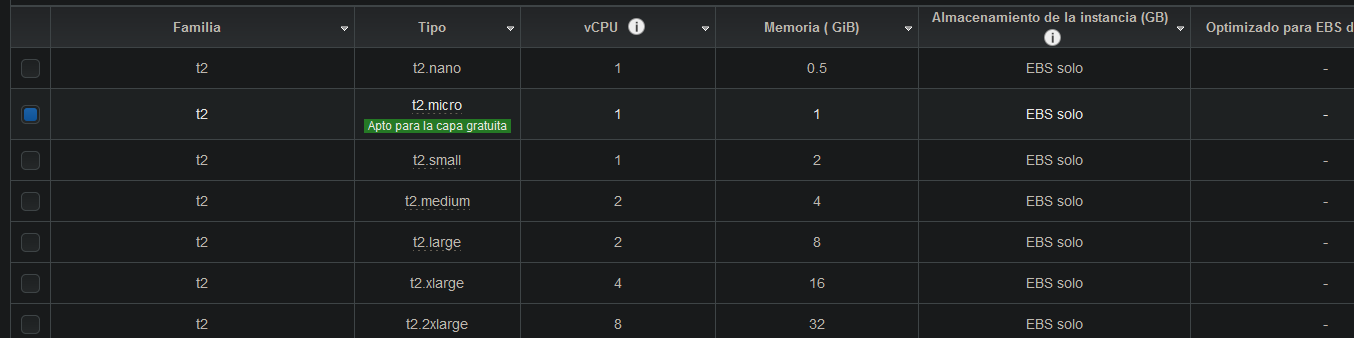


Dentro del apartado de instancias, hacemos click en donde dice: “Lanzar instancia”.



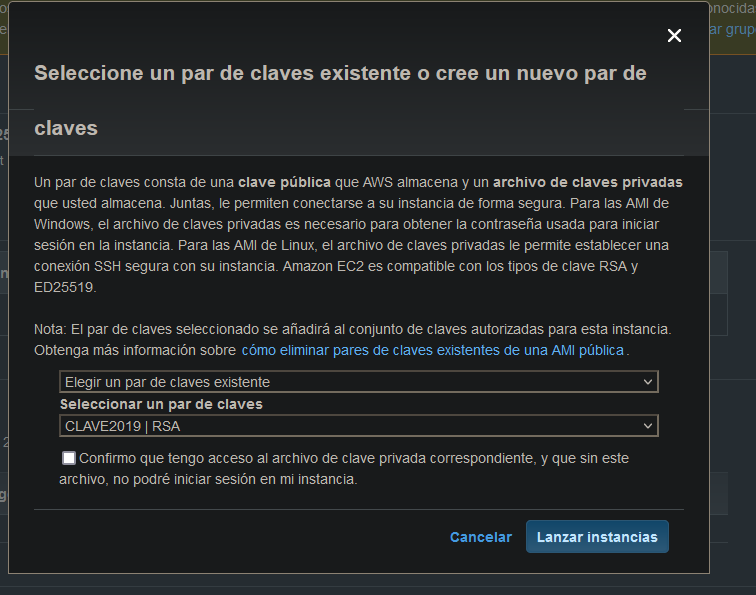
Debemos de encontrar una AMI de Ubuntu (free tier), a ser posible.





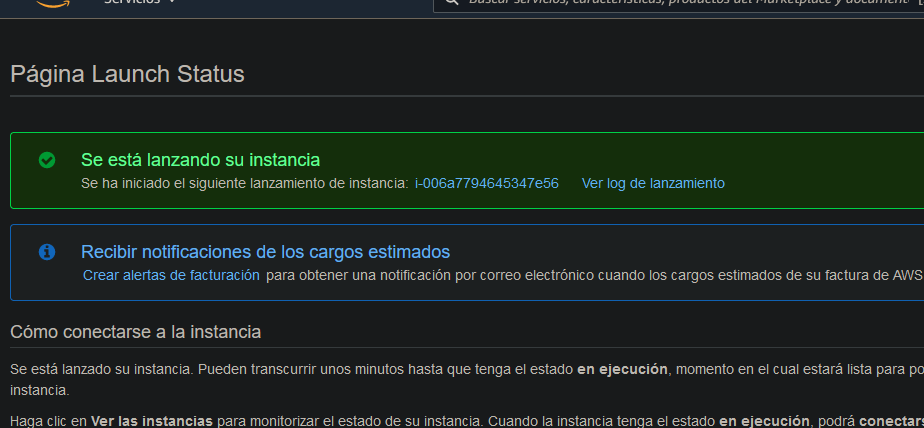
Puesto que en el ejercicio anterior hicimos lo mismo que se requiere en éste (cambiando únicamente el tipo de AMI a seleccionar), vamos a omitir cada uno de los pasos que no requieran de acción por nuestra parte.

En este paso, si que debemos actuar, generando el par de claves que vamos a utilizar. Tal y como hicimos en la instancia Windows.



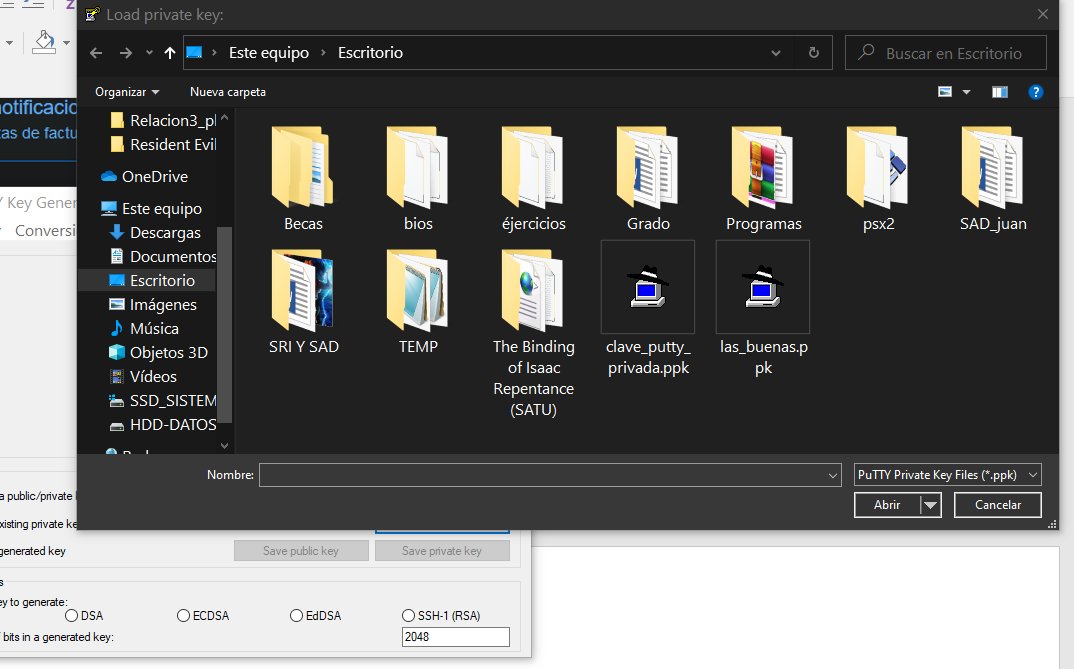


Desastrosamente, por alguna razón la AMI Ubuntu Free Tier nos da un error inesperado por lo que me veo en la obligación de usar una AMI propia de Amazon.



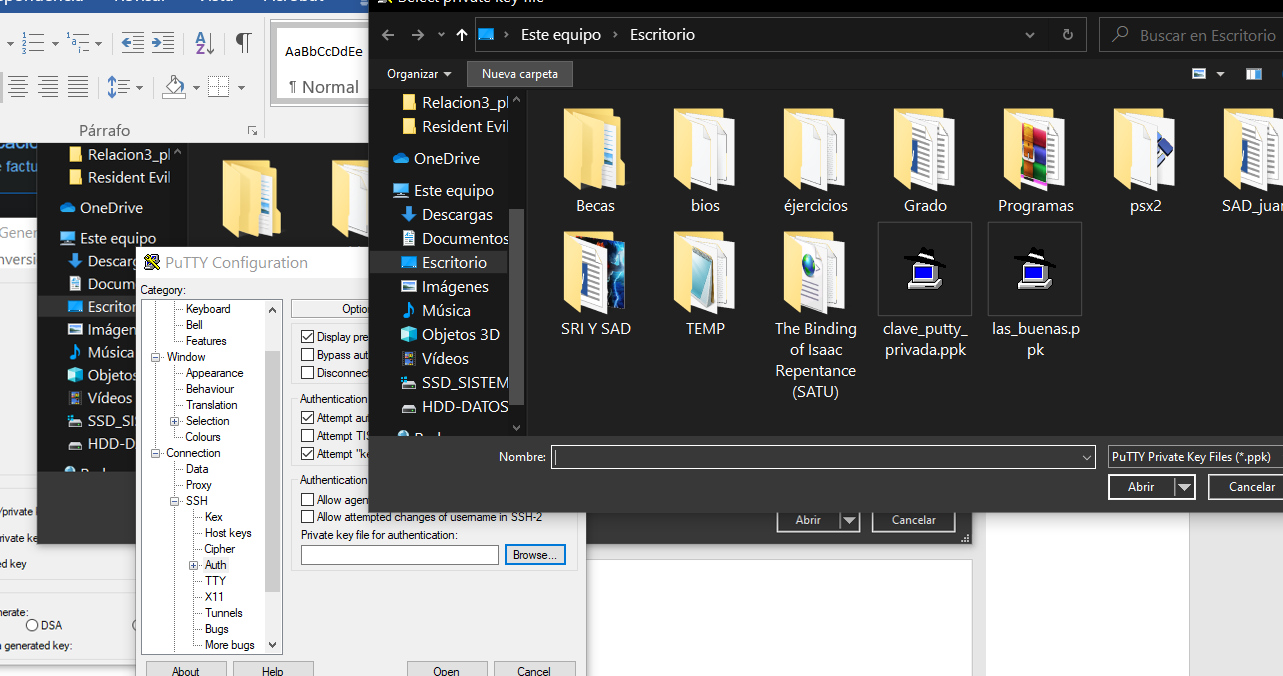
A continuación, debemos insertar la clave privada en el cliente SSH que vamos a utilizar para que asi podamos conectar con la instancia, con total seguridad.

En mi caso, utilizé PuTTy.



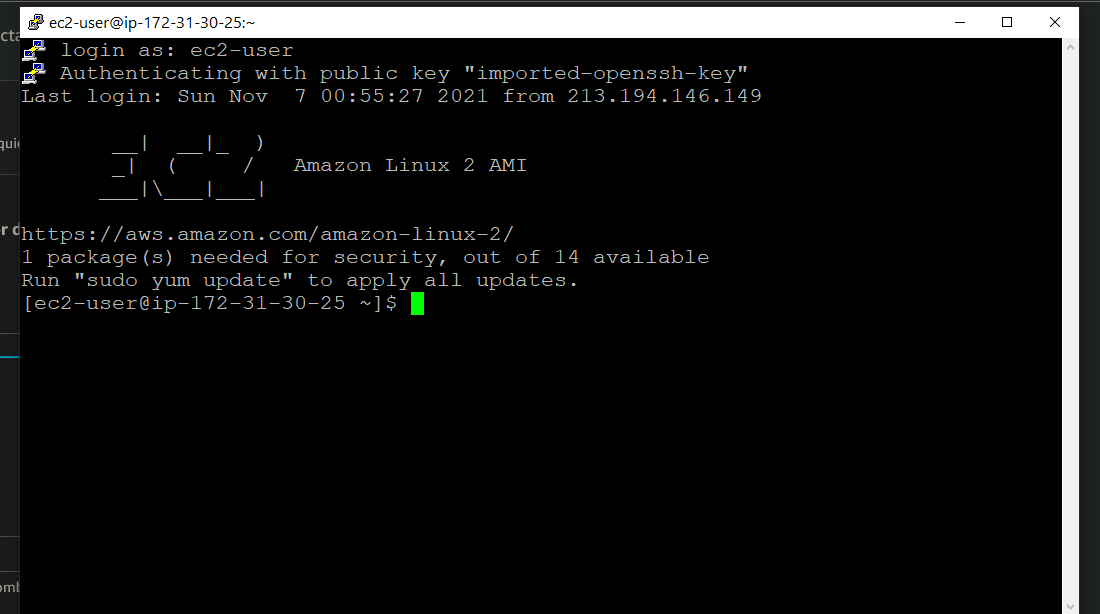
Debemos pulsar en “Save Key” y hecho esto, volvemos a PuTTy.

En la siguiente captura, procedo a buscar el archivo que genera PuTTy, y que es reconocido por el programa en cuestión.



Luego, guardamos la clave como un perfil en PuTTy y conectamos con las instrucciones que brinda la plataforma de AWS.

Por último, lo tendríamos todo listo, teniendo en cuenta que usaremos el usuario “ec2-user” conectamos con el servidor.



Puesto que el servidor, va a actuar como un servidor de archivos, debemos de indicar una serie de puertos, los cuales vamos a estar utilizando para dicha acción. Además de ello, abriremos el puerto 22 para hacer uso de la utilidad SSH. Haremos uso de los puertos 80 y 443, también conocidos como “HTTP” y “HTTPS” respectivamente.



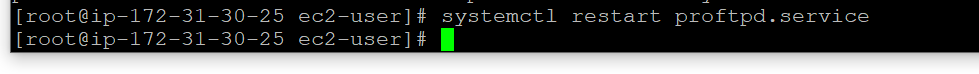
En el siguiente paso, instalamos el paquete que vamos a utilizar para la creación del servidor de archivos o FTP.

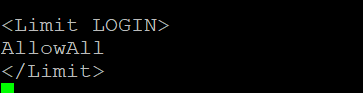


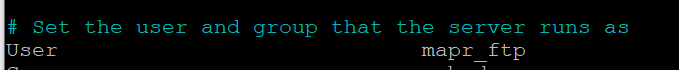
Debido a que la distribución de Linux no es Ubuntu, sino Debian, la sintaxis de algunos comandos varía, entre ellos, el de instalación de paquetes, por lo que, en este caso, la instalación quedaría de la siguiente forma.

Yum install proftpd

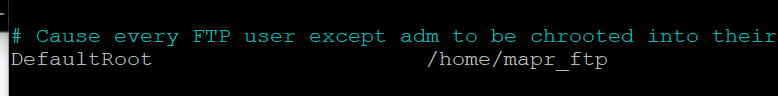
Reiniciamos el servicio.



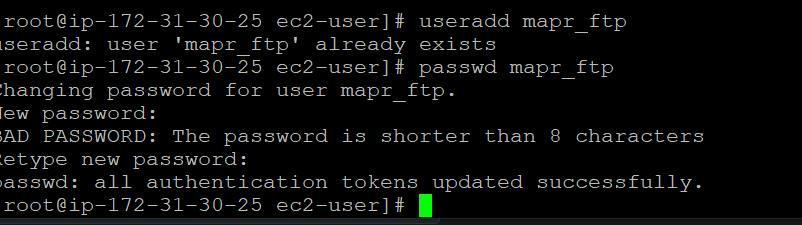




Para la correcta funcionalidad del servicio, algunas de las directivas del fichero de configuración debe ser modificadas, en las capturas de pantalla, muestro como debería de verse la correcta instalación del mismo.



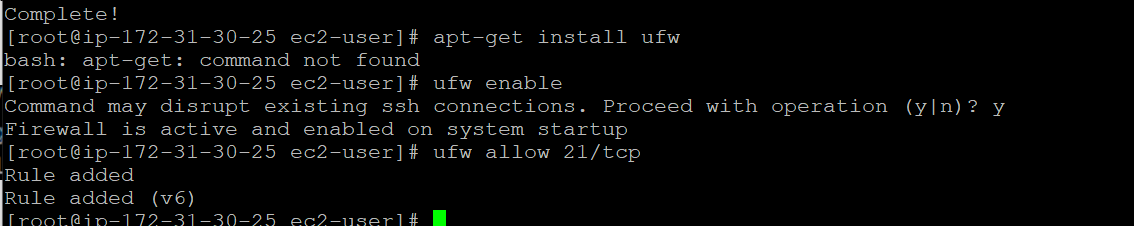
Crearemos un usuario.



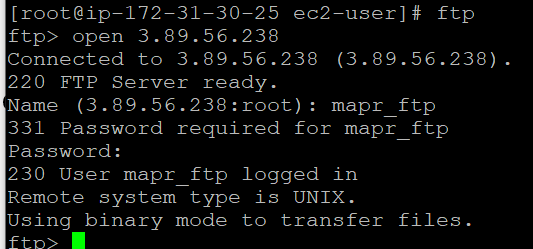
Este fichero de configuración que se muestra



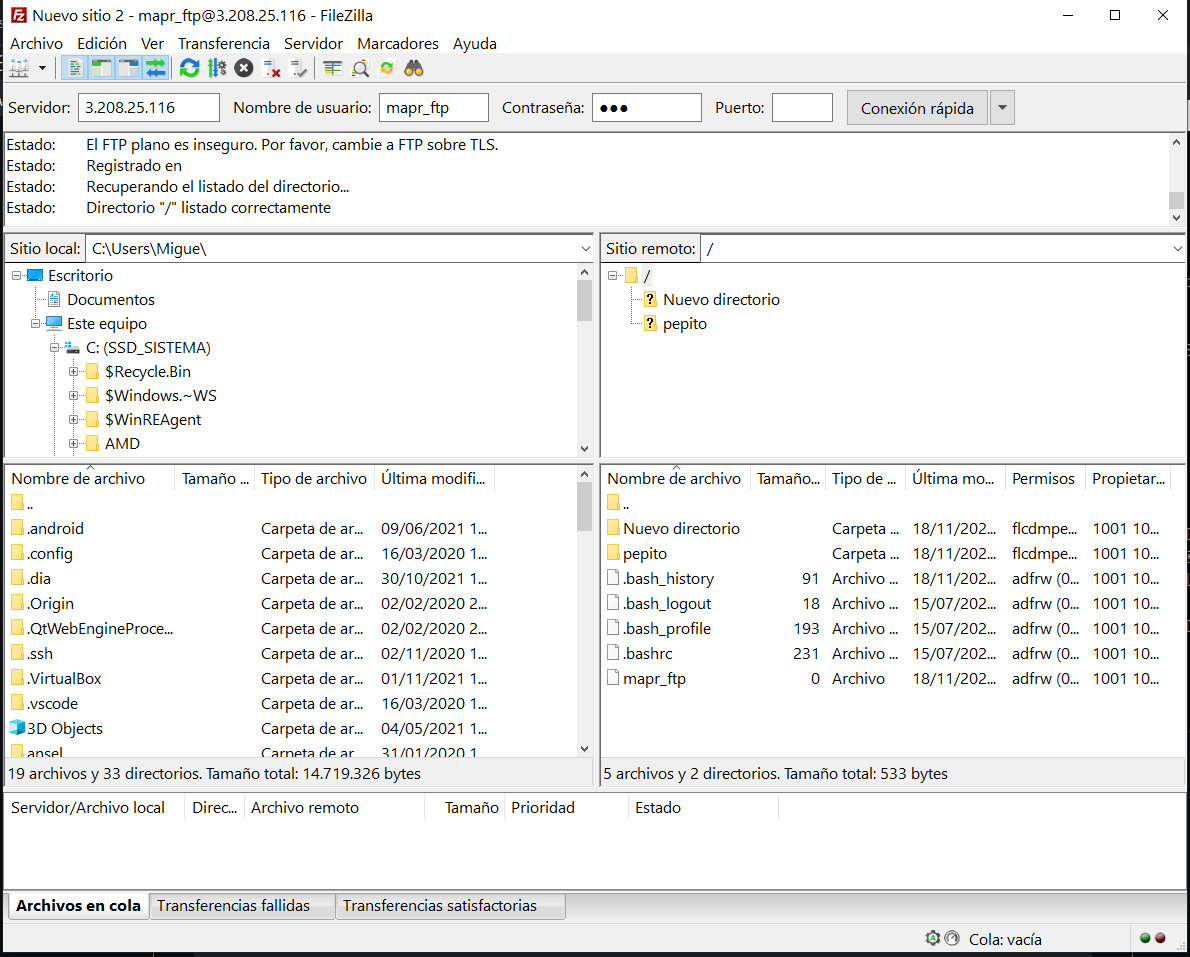
Para asegurarme de la apertura de puertos, además de haberlo hecho ya una vez en la configuración de la instancia, procedo a hacerlo por segunda vez, aunque esta desde el propio servidor.



Al introducir las credenciales, la conexión es exitosa, por lo que sin duda el servicio funciona con normalidad.

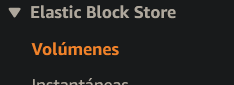


Además de esto, vamos a probar con la aplicación “Filezilla” que nos debe permitir la conexión con el FTP de la máquina de forma remota.

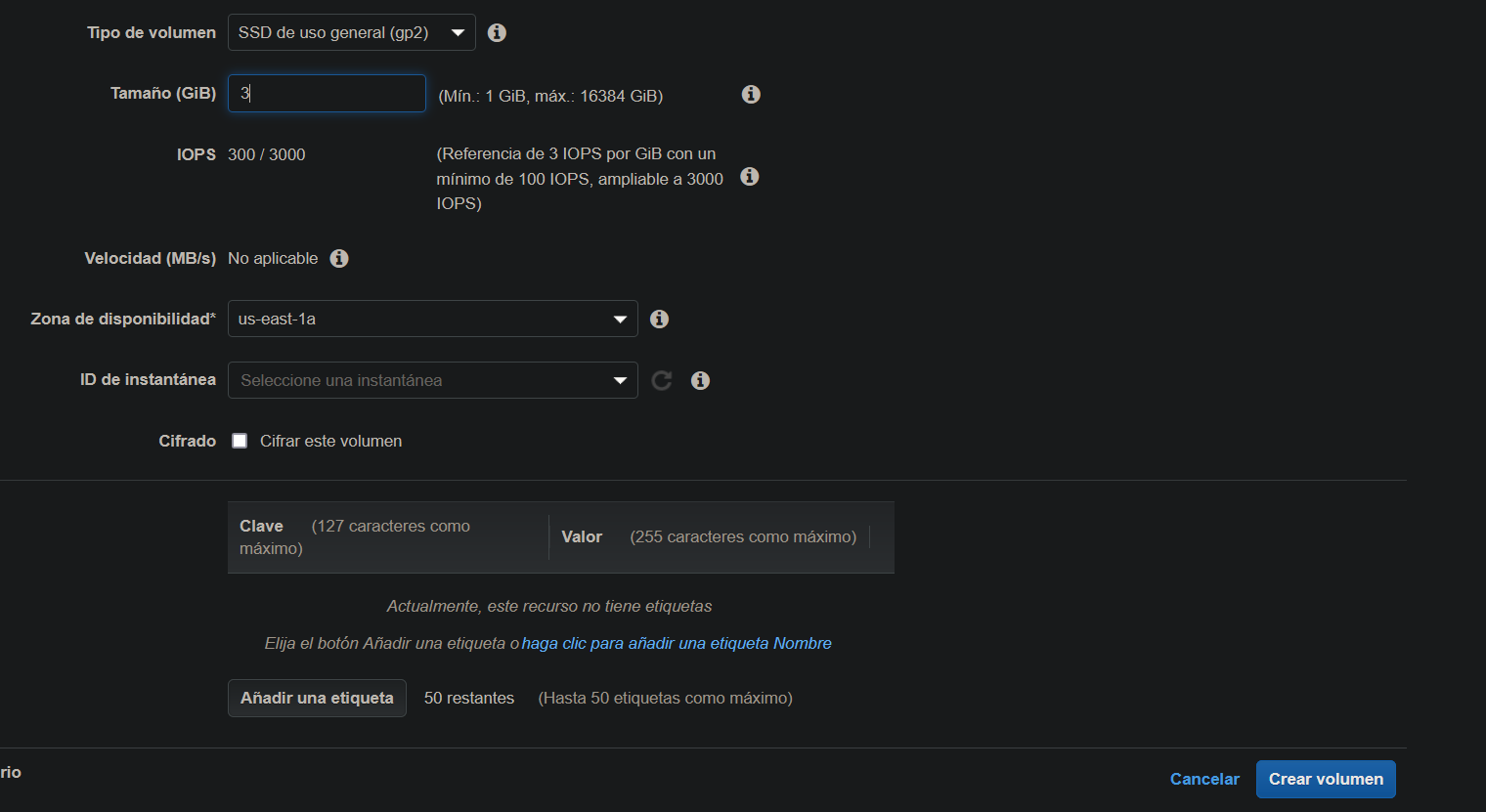


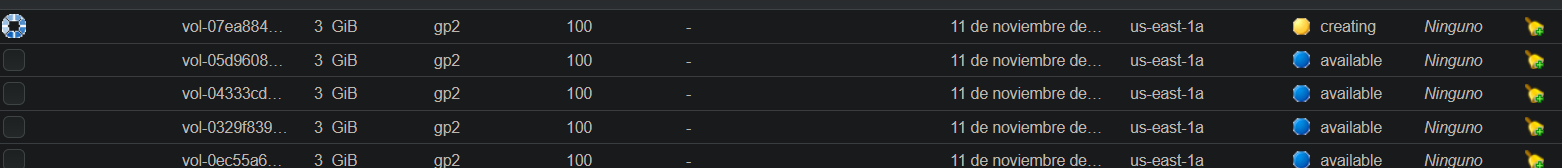
# PARTICIONES EN WINDOWS Y RAID

En el enunciado de la actividad de creación de particiones en Windows y sistema de discos RAID, se nos pide crear 5 volúmenes de 3 GB, en una instancia Windows, para ello, aprovecharemos la instancia creada con anterioridad.



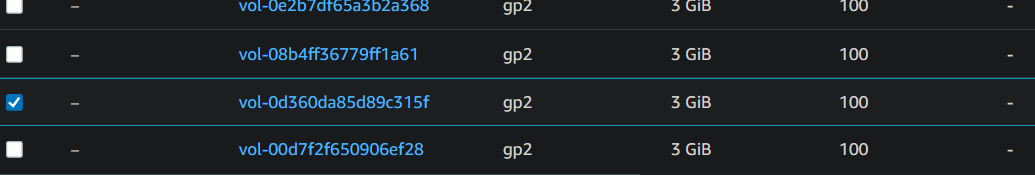




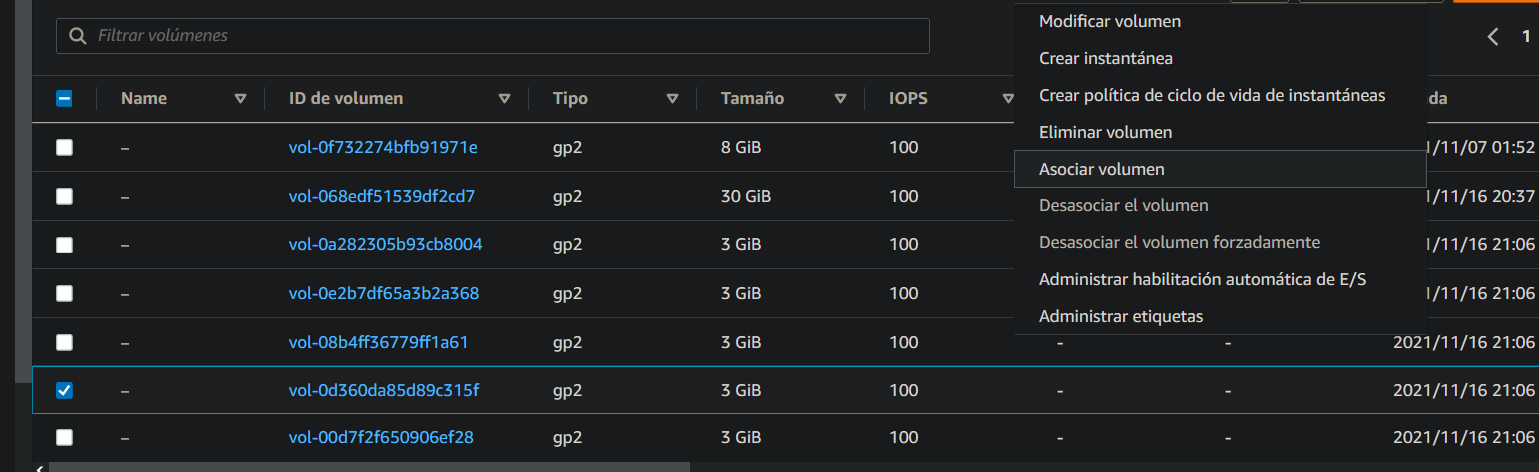


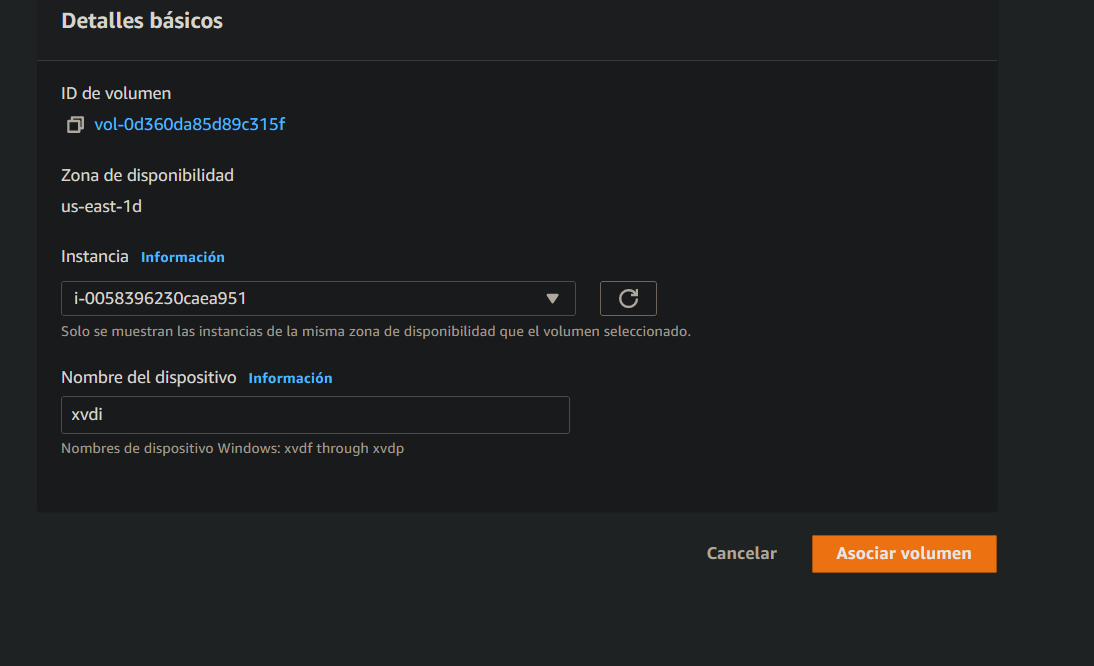
Hemos creado los 5 discos que usaremos durante el transcurso de esta actividad.

Luego, debemos de seleccionar la instancia la cual será asignada a todos y cada uno de los discos creados.

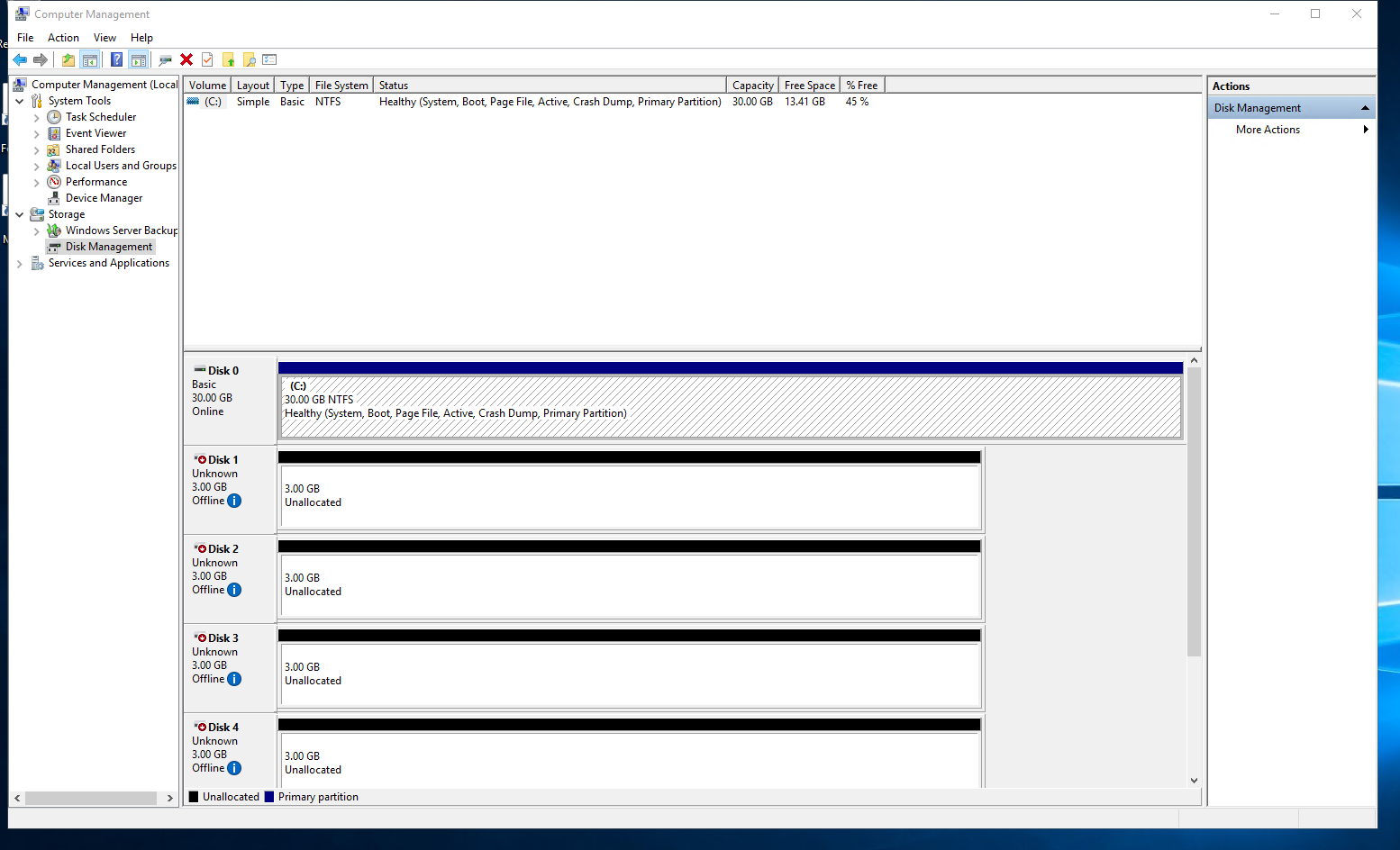


Luego, hacemos click en asociar volumen y seleccionamos sobre la instancia que queremos que sea asignada al éste.





Debemos hacer lo mismo con todos los volúmenes.

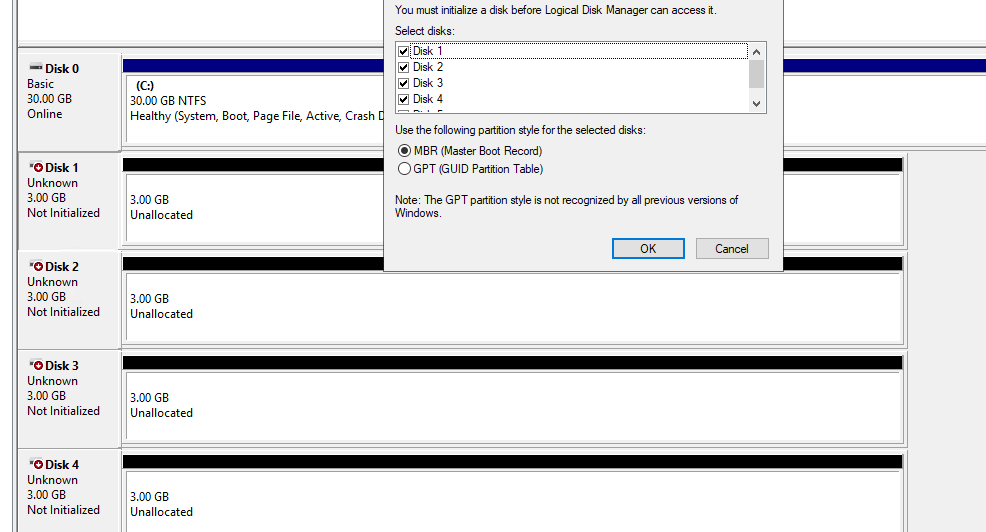


Dentro de la máquina Windows, vamos a abrir la herramienta de administración de discos, desde la cual se pueden inicializar discos, crear particiones y crear los discos independientes de array redundante, mejor conocido como RAID.

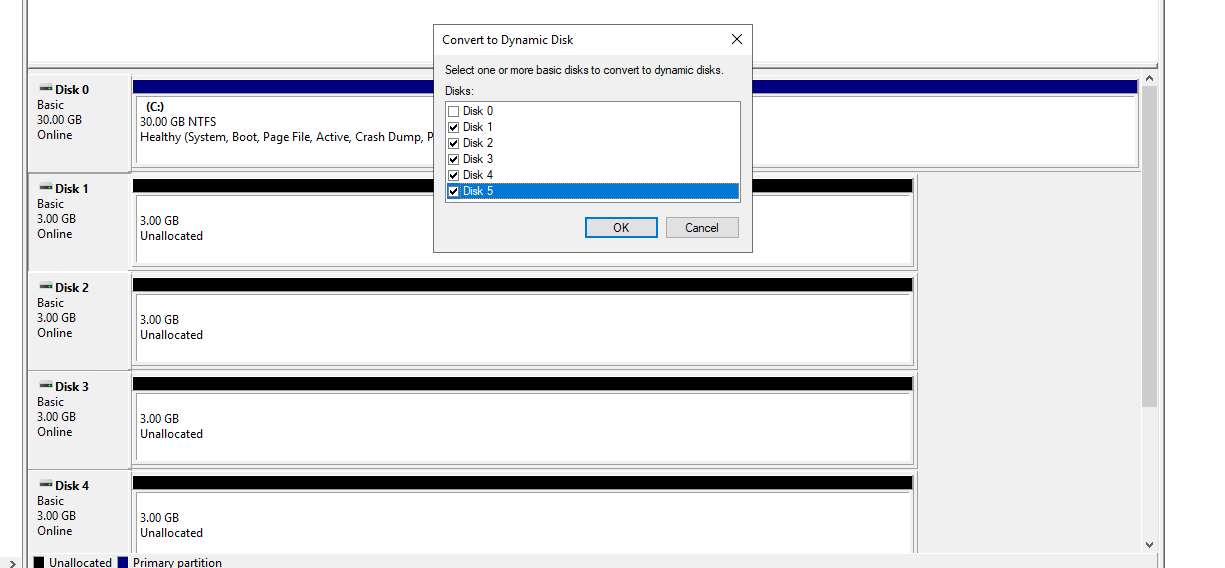
Seguidamente los inicializamos.



Debemos asegurarnos de que todos los discos han sido inicializados para así poder trabajar con ellos.

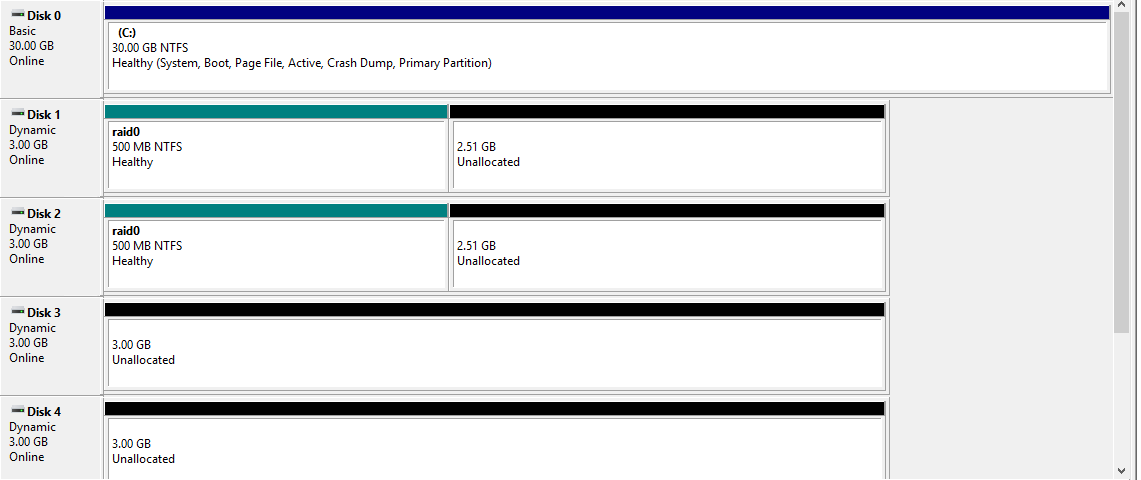


Una vez hemos inicializado los discos, debemos convertirlo en discos dinámicos.



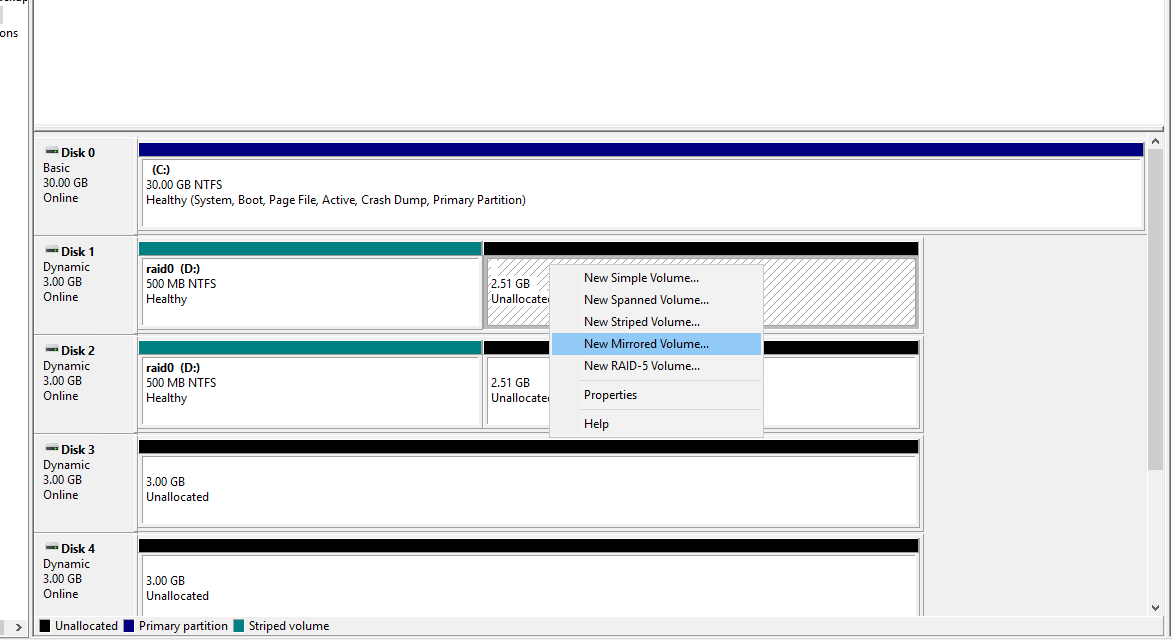
Asignamos 500mb a los primeros dos discos para hacer el primero de los RAID, el RAID0

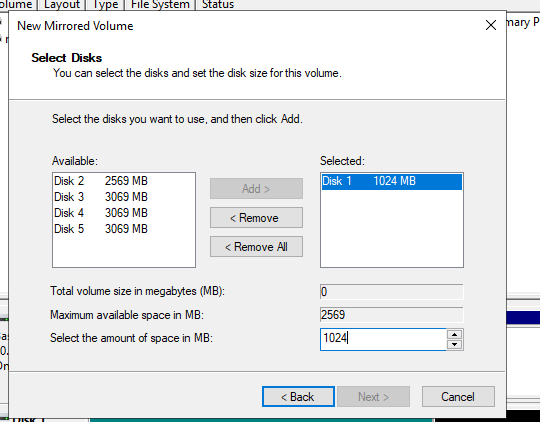
los RAID0. Este tipo de RAID consiste en almacenar dos discos en dos “bandas”. Este método se emplea para aumentar el rendimiento de las actividades de lectura/escritura.



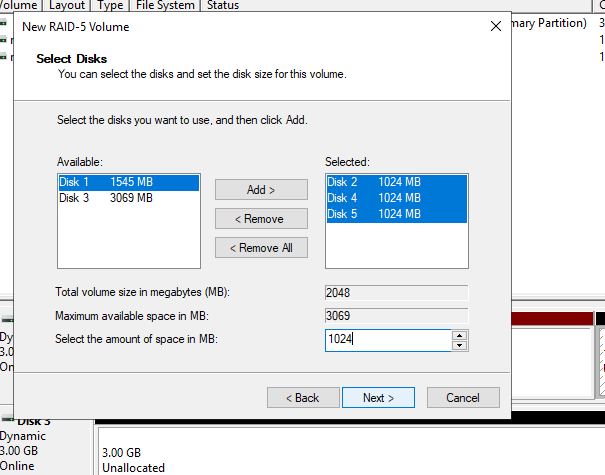
Una vez creados los RAID0

Creamos el volumen espejo o reflejado de 1gb. Dicha forma de almacenamiento consiste en crear una copia exacta de los datos de otro volumen, lo cual nos ofrece un mayor desempeño frente a errores, puesto que uno de los dispositivos falla, el otro podrá respaldarle.

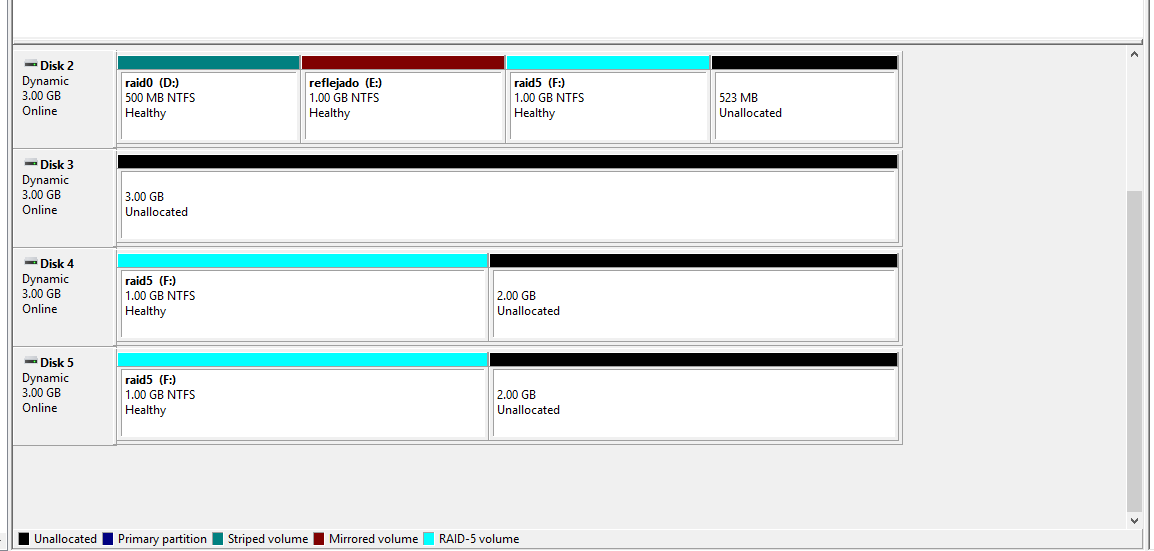




Por último, el RAID5. Este sistema es el más común y el que más nos vamos a encontrar. ¿Cuáles son las ventajas de este tipo de RAID?. Pues bien, entre sus ventajas podemos encontrar que las operaciones de lectura son muy rápidas, aunque por su contra, los procesos de escritura si son mas lentos. A pesar de ello es muy buena alternativa a la hora de obtener un buen nivel de tolerancia a fallos en nuestros dispositivos.



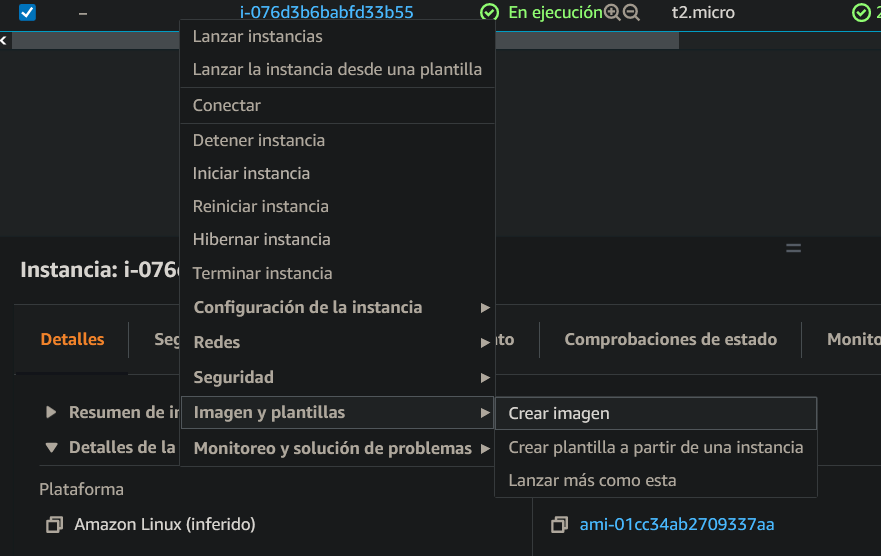
De esta forma, concluye la actividad, en la siguiente captura se muestra, a modo de resumen y de forma gráfica, la creación de los RAID.



# AUTOESCALING

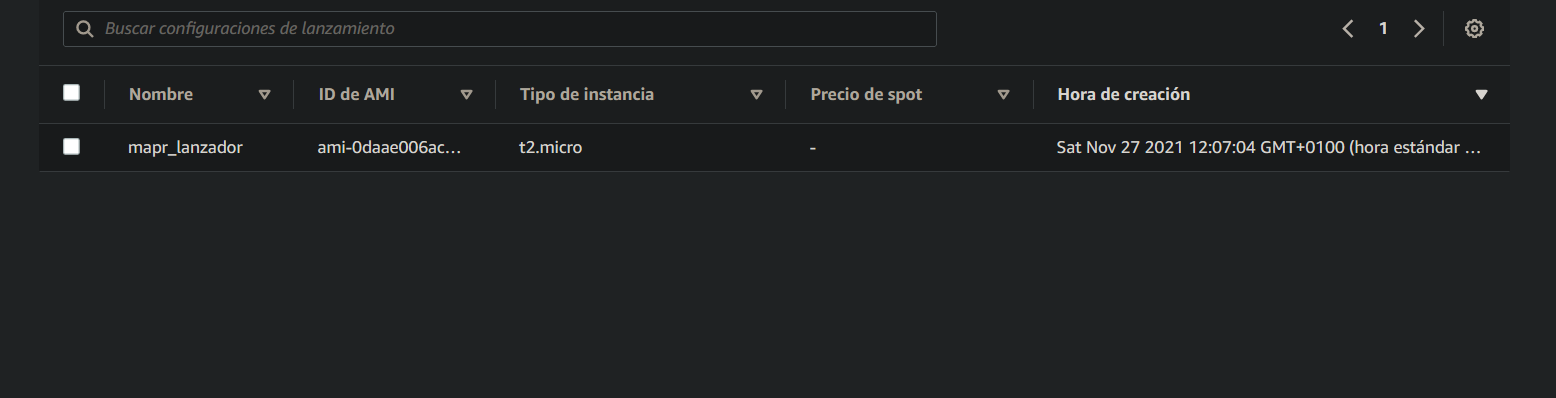
El autoescaling, consiste en repartir la carga de trabajo de los servidores, para ello la utilidad levanta determinados servidores dependiendo del volumen de trabajo.

Para realizar el autoescaling, debemos acceder a: “EC2”, luego de ello, buscamos la AMI que habíamos creado con anterioridad para nuestro servidor ftp. Luego seleccionamos click derecho e imagen y plantilla.

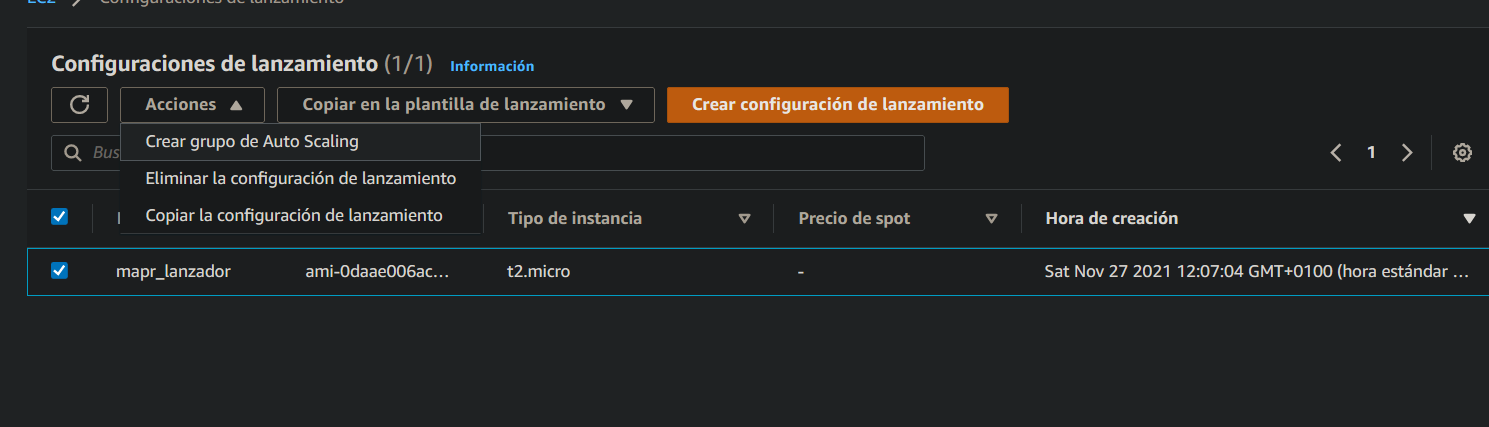


Seguidamente, en este tipo de instancia se incluirá la de tipo t2.micro puesto que es la que hemos estado usando hasta ahora.

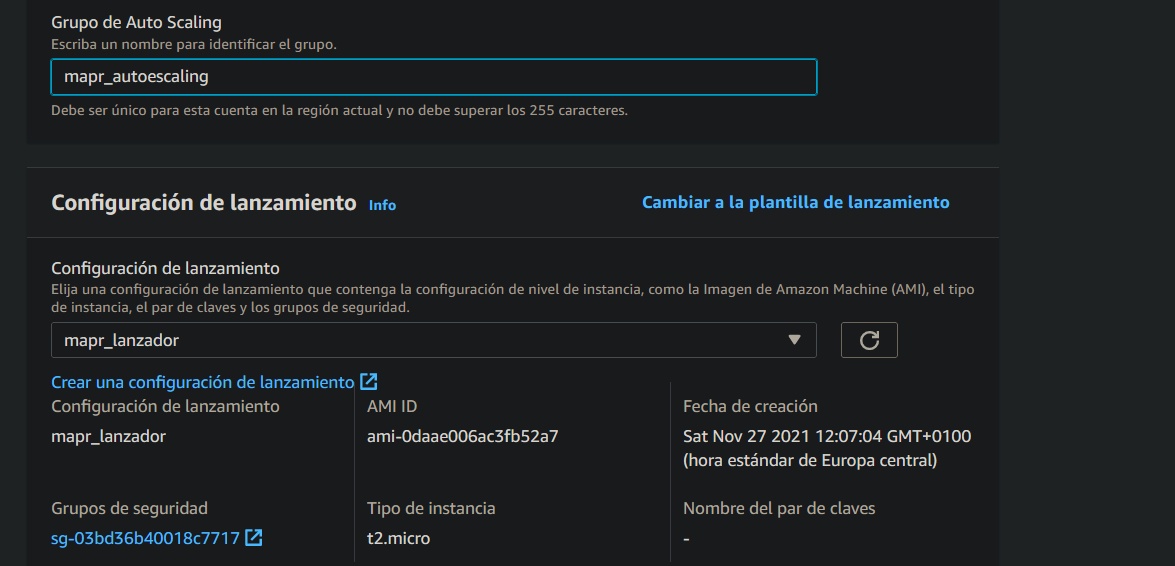
Como podemos ver ya se ha creado, este proceso es muy sencillo.



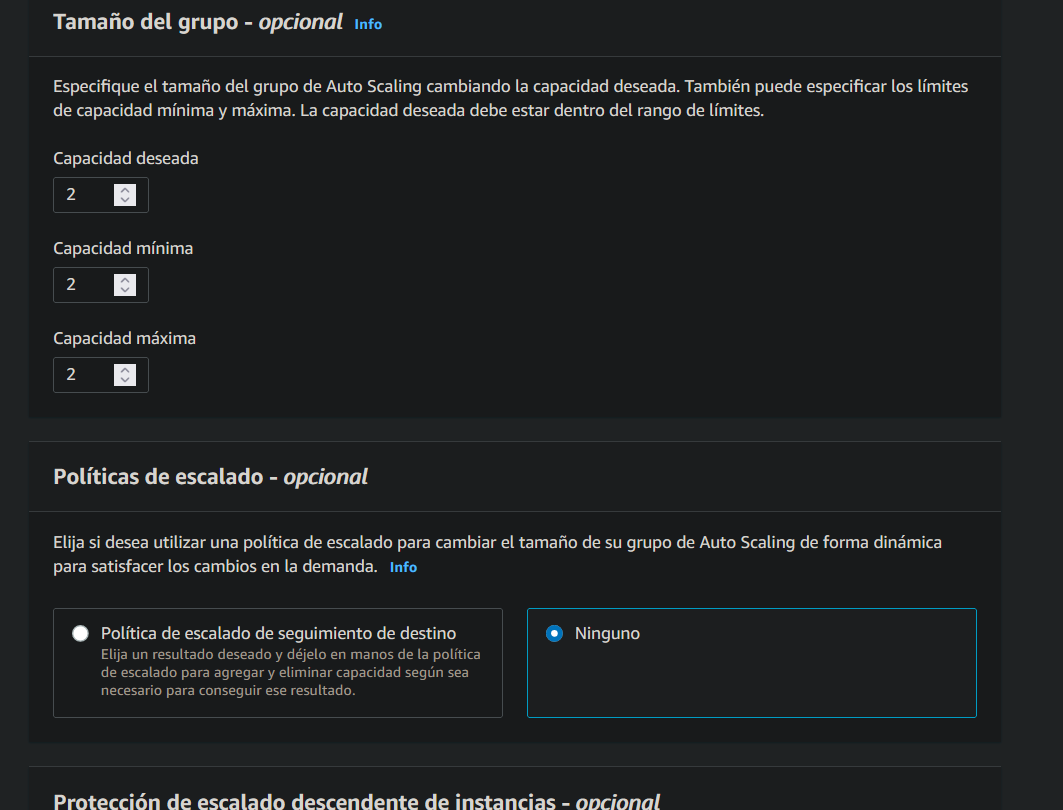
Seguido de esto, debemos acceder a Acciones y a continuación, Crear Grupo.

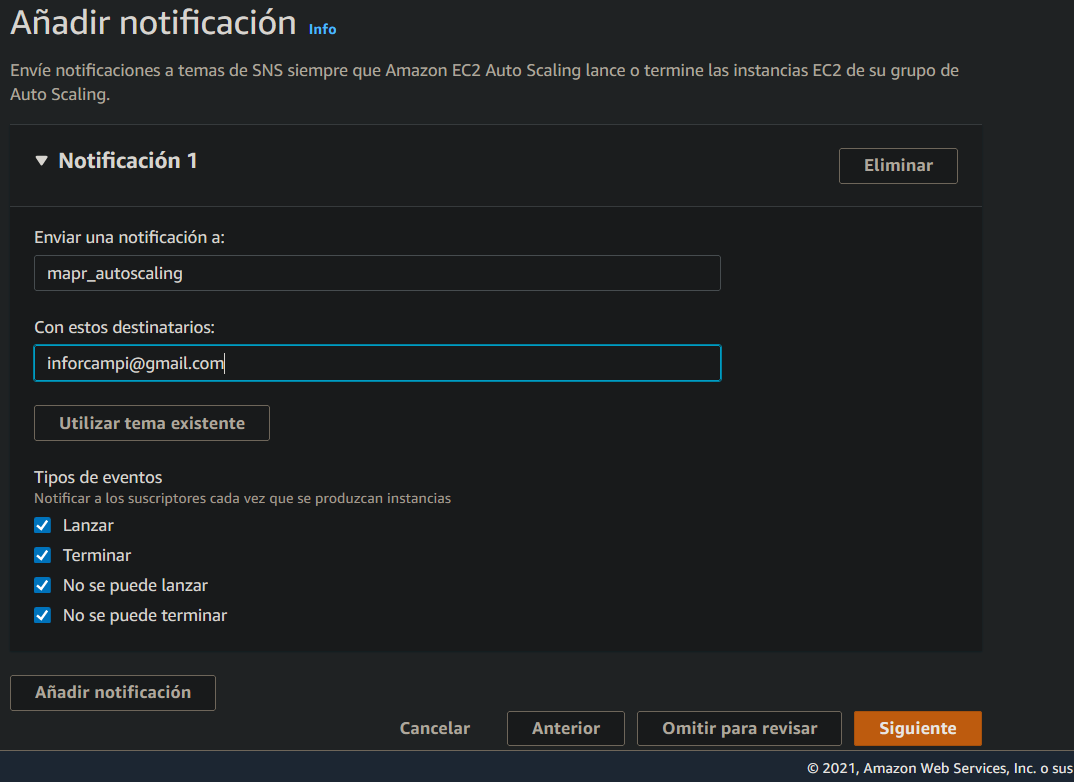


Debemos de darle un nombre, para así identificarlo.



En este menú de la captura que aparece mas abajo, es donde se le va a indicar, la cantidad de estancias que debe haber, así como los límites de éstas.



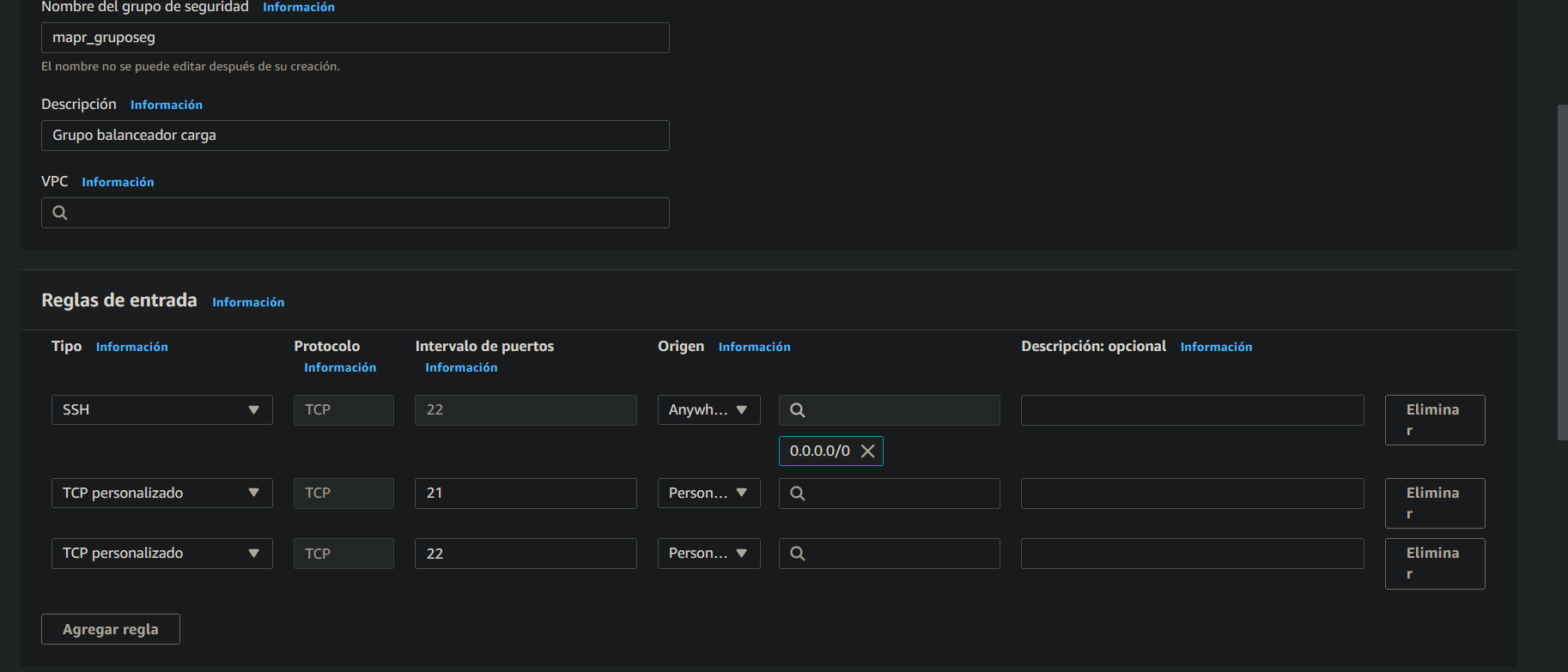
A continuación, debemos indicar la dirección de correo electrónico a la cual se va a enviar un reporte en el momento que alguna de las instancias falle, y por lo tanto, sea necesario iniciar otra.

Con esto concluye la creación del grupo de Autoescaling.

# BALANCEADOR DE CARGA

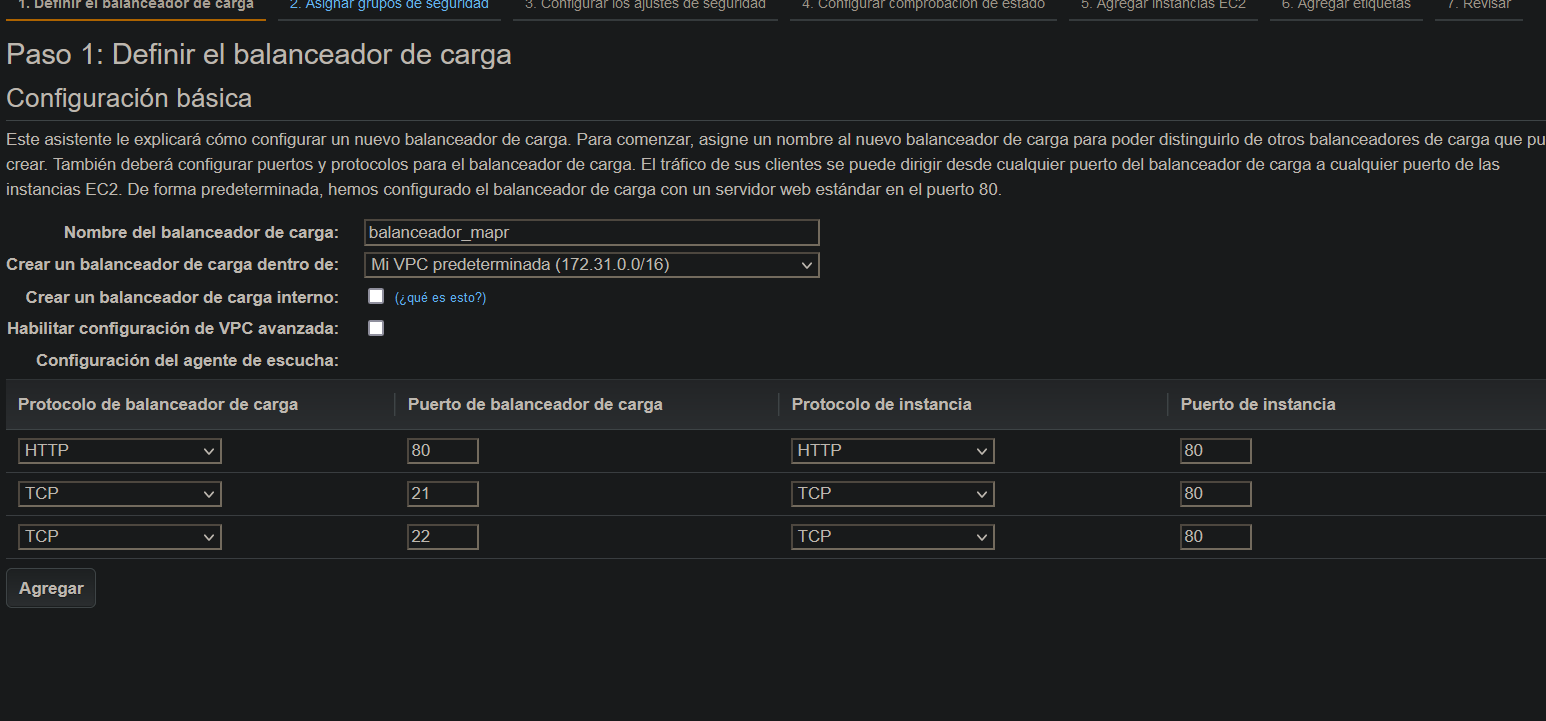
El balanceador de carga es una utilidad que nos permite que nuestro sitio web, aplicación u otro servicio que tengamos en nuestro servidor pueda funcionar con total normalidad, sin que existan retrasos en las peticiones de los clientes o usuarios.

Para la tarea del balanceador de carga lo primero que vamos a tener que hacer es un grupo de seguridad con las opciones que nos da el ejemplo.

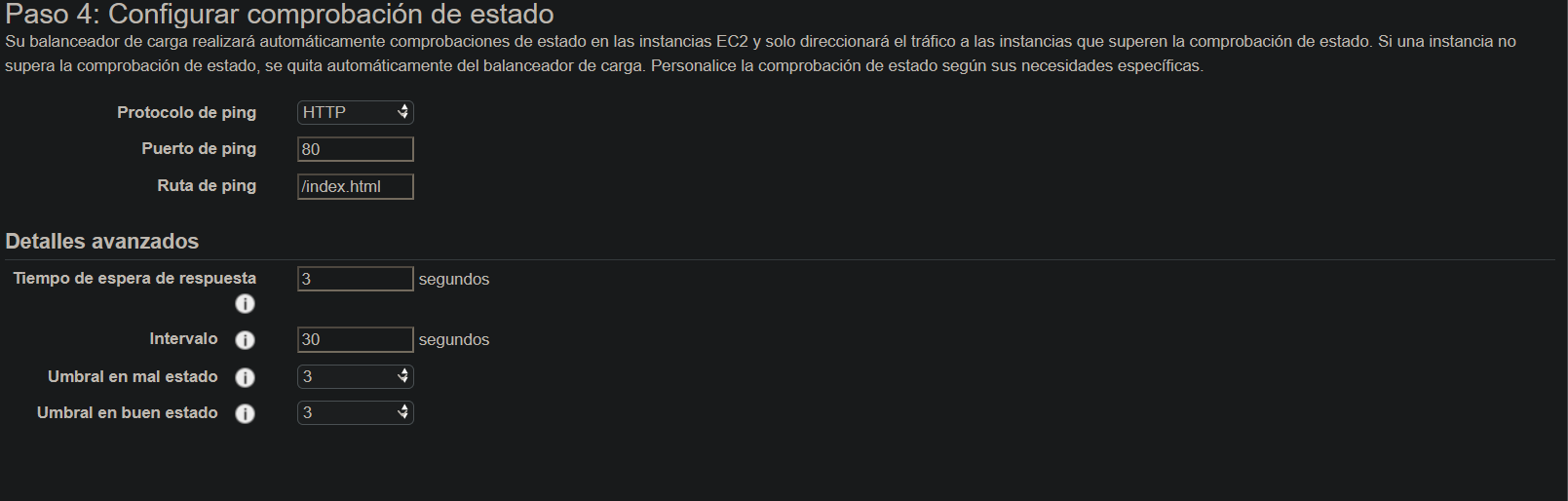


Luego nos vamos a equilibrio de carga dentro de balanceador de carga

Y creamos uno tal que asi.



Aquí configuramos las distintas opciones, las cuales indican una serie de comprobaciones que el balanceador de carga hará antes de usar una determinada instancia.



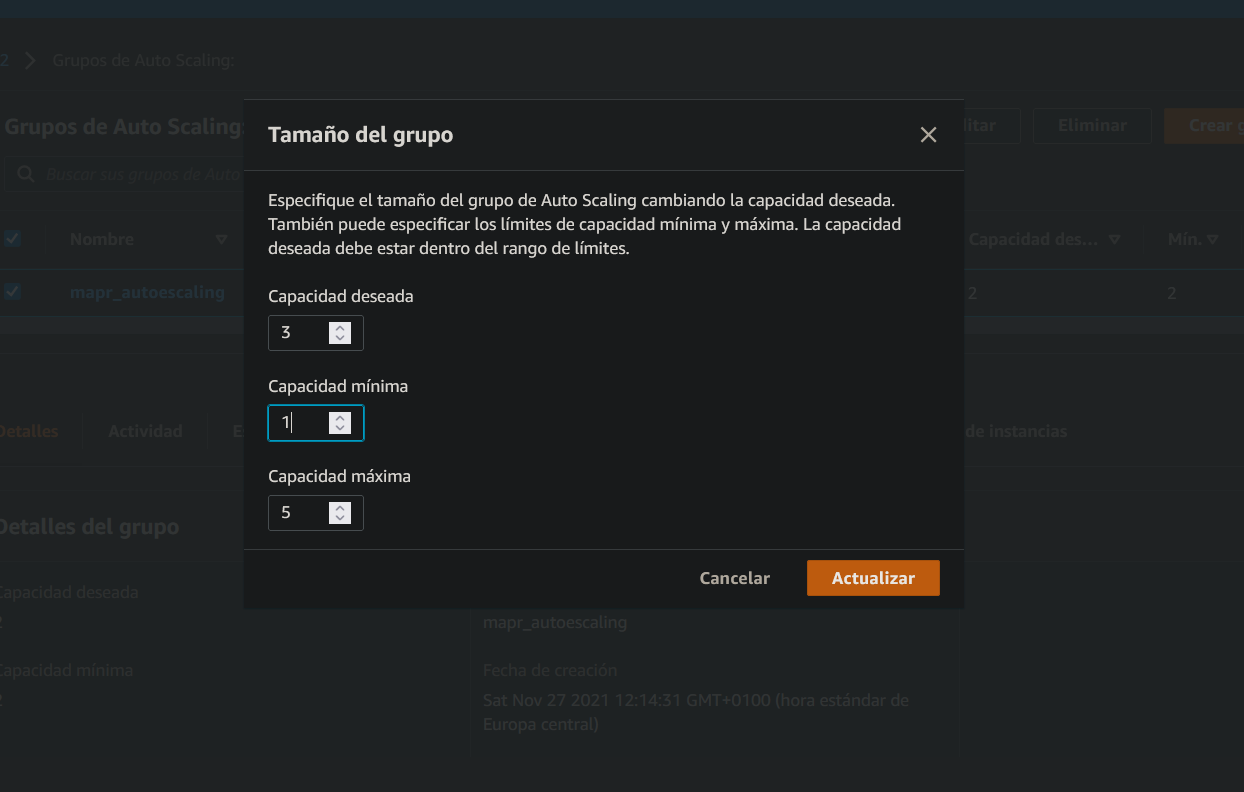
Entre todas las instancias, seleccionaremos donde se encuentra el servidor ftp.



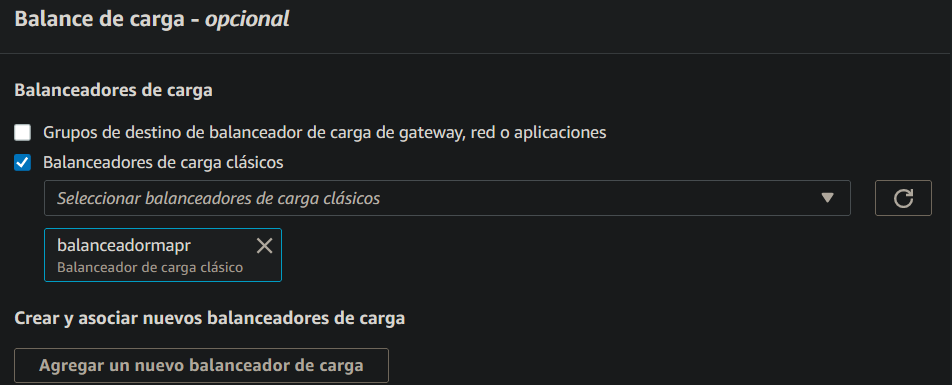
Y listo, ahora sólo quedaría asociarlo a la instancia Linux.



En autoescaling group, le damos a editar para forzar a que podamos comprobar si realmente está funcionando.



Seguido de ello, hacemos click en editar.





Y ya nos aparece funcionado.

# CONCLUSION DE AMAZON WEB SERVICES

Como resumen del proyecto me gustaría decir que durante el transcurso del mismo, hemos conocido la herramienta de Amazon, Amazon Web Services. Consiste en una especie de hosting para tener máquinas virtuales en la nube. Es muy útil por si queremos tener una aplicación web, servidor con base de datos o servicios en la red al cual podamos acceder desde cualquier sitio. Lo bueno de AWS es que al ser de Amazon es muy completo y que también es muy demandado en las empresas del sector.

Además, un buen punto a favor es que a la hora de realizar el curso se adquieren conocimientos de tecnologías muy actuales y que nos pueden servir de cara al mercado laboral.

Como adición a lo anterior, este proyecto puede servir como base para poder hacer una certificación de AWS que es muy demandada. Que, aunque bien no es gratuita, siempre tendremos la oportunidad de afrontarla en un futuro con mayor facilidad.

Como punto negativo a lo que es la plataforma, podríamos decir que durante el desarrollo del proyecto he encontrado una serie de inconvenientes que han dificultado la realización del mismo y que han sido independientes a mi como los siguientes:

* La migración de la aplicación web de Amazon, ha repercutido en que en algunos momentos, durante el transcurso de actividades, hayan desaparecido todas las instancias que habían sido creadas
* En un momento concreto del desarrollo del proyecto necesitaba una AMI concreta para poder realizar una actividad, a la cual no podía acceder por falta de permisos pero que pude fácilmente solucionar debido a la existencia de una instancia Amazon, similar a ésta.

Respecto al uso de Amazon Web Services, la aplicación es fácil de utilizar, siempre y cuando tengamos claro lo que debemos hacer y los requisitos para cada una de las herramientas que nos brinda la plataforma.

FIN