

© Germán Moltó, 2013-2021. Se prohíbe la divulgación, utilización, transmisión, distribución, reproducción y modificación total o parcial de este documento y de cualquier otro material educativo del Curso Online de Cloud Computing con Amazon Web Services por cualquier medio sin el previo y expreso consentimiento del autor, ni siquiera para ámbito académico y/o educativo. Este material es de uso estricta y exclusivamente personal.

26/04/2021

Práctica

*Despliegue de Instancias de Máquinas Virtuales con Amazon EC2
Usando AWS CLI*

Práctica

Despliegue de Instancias de Máquinas Virtuales con Amazon EC2 Usando AWS CLI

Contenido

Introducción.....	2
Resultados de Aprendizaje.....	2
Advertencias Previas	3
Entorno de Trabajo	3
Configuración del Entorno de Trabajo (Opcional)	3
Verificación del Entorno de Trabajo	5
Despliegue de Máquinas Virtuales en AWS	7
Creación del Par de Claves, Grupo de Seguridad y Elección de AMI.....	7
Despliegue de Instancias de Máquinas Virtuales	13
Sobre las Características de las Instancias basadas en EBS.....	24
Usando la AWS Management Console	28
Despliegue de Instancias Basadas en Windows	29
Creación del Grupo de Seguridad y Selección de la AMI	29
Despliegue, Datos de Acceso y Conexión a la Instancia	30
Conclusiones	34
Información Adicional	34
Referencias.....	34

Introducción

Amazon Web Services (AWS) [1] es un proveedor de Cloud público pionero en el campo de las tecnologías Cloud (ofreciendo servicio desde 2006). De entre todos los servicios que ofrece, en esta práctica se utilizarán principalmente los siguientes:

- Amazon Elastic Compute Cloud (EC2).

Esta práctica permite que el alumno realice el despliegue de máquinas virtuales sobre la infraestructura de Cloud público de Amazon, usando EC2. Esto permitirá ofrecer una visión de la forma de trabajar desde el punto de vista del IaaS (*Infrastructure as a Service*), donde se realiza un aprovisionamiento de máquinas virtuales, que posteriormente debe gestionar de forma conveniente el usuario. Existen múltiples cuentas de usuario para los alumnos creadas mediante el servicio IAM (*Identity and Access Management*) bajo una cuenta de usuario de AWS con cargo a la tarjeta de crédito VISA del profesor. Esto permite agrupar el coste de todos los usuarios bajo una misma cuenta.

En esta práctica se utilizará fundamentalmente la línea de comandos para interactuar con el Cloud público. No obstante, AWS ofrece la AWS Management Console que permite la realización de la mayor parte de la funcionalidad aquí explicada de forma gráfica a través de un navegador web. El uso de línea de comandos facilita el *scripting* o la posibilidad de integrar la funcionalidad de AWS en aplicaciones propias del usuario.

En esta práctica se utilizará la interfaz de línea de comandos de AWS (AWS CLI) [18], que es la herramienta oficial de acceso a AWS desde un terminal. Existen, no obstante, otras formas de acceso desde línea de comandos a AWS, como por ejemplo las “Amazon EC2 API Tools” [3], que requiere que el cliente tenga Java y las credenciales de usuario basadas en certificados X.509. También es posible utilizar la herramienta “aws” [6], desarrollada en Python y que permite utilizar diferentes servicios de AWS desde línea de comandos usando las credenciales de usuario basadas en *Access Key* y *Secret Key*. Esta herramienta consta de un único fichero Perl y que tiene como dependencias la librería *curl*, instalable de forma sencilla desde cualquier distribución Linux.

Resultados de Aprendizaje

Se espera que, una vez finalizada la práctica, el alumno sea capaz de:

- Conocer el esquema de funcionamiento a nivel de usuario de Amazon EC2.
- Entender el concepto de máquina virtual y su mecanismo de despliegue a través EC2.



1. Tu profesor te habrá asignado un número. Los comandos que aparecen en el boletín hacen referencia al usuario alucloudoo. Por favor, **substituye en los comandos que veas oo por tu número** (por ejemplo, si tu número es 06, entonces el usuario que has de utilizar es alucloudoo6). Recuerda esta regla para facilitar tanto tu trabajo como el del resto de compañeros. Para facilitar tu trabajo, en los comandos se utiliza una variable de entorno llamada ID que se debe resolver a tu número.
2. En esta práctica haremos ejecuciones reales sobre un proveedor de Cloud público, que generan un coste económico. Asegúrate de **liberar apropiadamente los recursos** (terminar instancias, eliminar ficheros de S3 si ya no los vas a gastar, etc.) para **no incurrir en costes adicionales**. Si tienes alguna duda al respecto consulta con tu profesor.
3. Cualquier recurso creado en AWS que no cumpla con la nomenclatura indicada en este boletín podrá ser automáticamente eliminado por parte del instructor.
4. Por defecto trabajarás en la región us-east-1 (N. Virginia) de AWS. El acceso al resto de regiones de AWS ha sido restringido por el instructor.

5. Recuerda que hay un sistema de recompensas, definido en la Guía de Prácticas, por el que podrás aumentar la duración del curso reportando las discrepancias que encuentres en este boletín frente a posibles cambios introducidos por AWS. Cuento con tu colaboración.

Esta práctica se realiza fundamentalmente mediante línea de comandos. Sin embargo, puede realizarse casi de forma completa también usando la AWS Management Console. Si no te sientes cómodo usando la línea de comandos, revisa al final del boletín las instrucciones para usar la AWS Management Console.

Advertencias Previas

En primer lugar, tienes a tu disposición en PoliformaT una guía para la realización de las prácticas que explica el uso de un cliente SSH para conectarse al entorno de realización de prácticas, así como la forma de acceder a la AWS Management Console mediante un navegador web.

En segundo lugar, a lo largo de este boletín verás comandos que hay que ejecutar en una ventana de terminal conectada por SSH a una máquina (virtual) Linux remota. Para facilitarte el trabajo, puedes copiar y pegar los comandos de este boletín en la ventana de terminal. No obstante, deberás tener cuidado con aquellos comandos que utilicen comillas simples (o dobles). A veces, al copiar el comando del boletín y pegarlo en la ventana de terminal, el carácter asociado a las comillas no es el apropiado, por lo que deberás modificar el comando en la ventana de terminal para que lleve las comillas apropiadas. De lo contrario, te puedes encontrar con algún mensaje de este estilo:

```
SignatureDoesNotMatch | The request signature we calculated does not match the  
signature you provided. Check your AWS Secret Access Key and signing method.  
Consult the service documentation for details
```



Este boletín está diseñado para responder a la problemática común que te puedes encontrar durante la resolución de la práctica. Si te surge algún problema, lo más probable es que el boletín te explique a continuación una posible solución al respecto. Revisa la propuesta antes de contactar con el instructor. Si encuentras un problema que no está explicado en el boletín contacta con el instructor explicándolo con detalle para así aislarlo y resolverlo.

Entorno de Trabajo

En primer lugar, inicia sesión mediante SSH con la cuenta de usuario y contraseña que te haya proporcionado tu profesor y en la máquina que te haya indicado. El entorno de trabajo ya debería estar configurado por defecto, pero por si acaso asegúrate de que:

- Puedes ejecutar el comando “aws”.

Configuración del Entorno de Trabajo (Opcional)

A continuación, se resumen las instrucciones de instalación que deberías realizar en tu propio equipo para acceder a AWS **si no utilizases el entorno pre-configurado** que te ofrecemos durante el curso (o si quieras usar AWS desde tu casa y desde cualquier otra máquina que no sea la que te ofrecemos para realizar las prácticas). Recuerda por tanto que **no es necesario que ejecutes los siguientes comandos en la máquina que os ofrecemos para realizar las prácticas**.

1. Instalación de AWS CLI

La herramienta AWS CLI se instala de forma sencilla con los siguientes comandos (ejemplo para GNU/Linux Ubuntu 12.04):

```
:~$ sudo apt-get install python-pip  
:~$ pip install awscli
```

2. Configuración de la herramienta

Es preciso crear el fichero `$HOME/.aws/credentials` para que contenga la “Access Key ID” (Nº de clave de acceso) y la “Secret Access Key” (Clave de acceso secreto). Estas claves se obtienen al crear un usuario IAM. Verifica si dicho fichero existe y, en ese caso, no deberás realizar ninguna configuración adicional. De lo contrario, dicho fichero tiene que tener permisos de lectura y escritura solo para el usuario (puedes cambiar los permisos mediante el comando `chmod 0600 $HOME/.aws/credentials`). También es posible utilizar el comando `aws configure` para especificar los valores de las credenciales.

```
:~$ ls -l $HOME/.aws/credentials  
-rw----- 1 alucloud00 alucloud00 137 Feb 15 21:41  
/home/alucloud00/.aws/credentials  
:~$ cat $HOME/.aws/credentials  
[default]  
aws_access_key_id = AKIAJANQMN62O7ADSC5A  
aws_secret_access_key = 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  
region = us-east-1
```

3. Definición de variables de entorno

Es posible modificar el formato de salida de la herramienta usando el parámetro `--output` (tal y como se comenta en la guía de uso de AWS CLI). Para utilizar por defecto la salida en formato texto, lo más apropiado es modificar el fichero de configuración de AWS CLI añadiendo al final la línea que se muestra a continuación en negrita:

```
:~$ cat $HOME/.aws/credentials  
[default]  
aws_access_key_id = AKIAJBIQMN72O7ADSC5A  
aws_secret_access_key = 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  
region = us-east-1  
output = text
```

Además, para agilizar los comandos que utilizaremos te recomendamos que utilices la variable de entorno `ID`, que debe resolverse a tu código de alumno (por ejemplo `$ID` vale `06` para el usuario `alucloud06`). Puedes verificar si dicha variable de entorno está definida con el comando:

```
echo $ID
```

Si obtienes un número, no es necesario que hagas ninguna configuración adicional puesto que la variable de entorno ya está establecida. De lo contrario puedes darle valor a la variable `ID` con la siguiente instrucción (deberás sustituir `XX` por tu código de alumno):

```
export ID=XX
```

Verificación del Entorno de Trabajo

El entorno estará correctamente configurado si la ejecución del siguiente comando muestra un resultado similar al que se muestra a continuación:

```
:~$ aws ec2 describe-regions
REGIONS      ec2.eu-north-1.amazonaws.com      opt-in-not-required eu-north-1
REGIONS      ec2.ap-south-1.amazonaws.com      opt-in-not-required ap-south-1
REGIONS      ec2.eu-west-3.amazonaws.com      opt-in-not-required eu-west-3
REGIONS      ec2.eu-west-2.amazonaws.com      opt-in-not-required eu-west-2
REGIONS      ec2.eu-west-1.amazonaws.com      opt-in-not-required eu-west-1
REGIONS      ec2.ap-northeast-2.amazonaws.com    opt-in-not-required ap-northeast-2
REGIONS      ec2.ap-northeast-1.amazonaws.com    opt-in-not-required ap-northeast-1
REGIONS      ec2.sa-east-1.amazonaws.com        opt-in-not-required sa-east-1
REGIONS      ec2.ca-central-1.amazonaws.com     opt-in-not-required ca-central-1
REGIONS      ec2.ap-southeast-1.amazonaws.com   opt-in-not-required ap-southeast-1
REGIONS      ec2.ap-southeast-2.amazonaws.com   opt-in-not-required ap-southeast-2
REGIONS      ec2.eu-central-1.amazonaws.com     opt-in-not-required eu-central-1
REGIONS      ec2.us-east-1.amazonaws.com        opt-in-not-required us-east-1
REGIONS      ec2.us-east-2.amazonaws.com        opt-in-not-required us-east-2
REGIONS      ec2.us-west-1.amazonaws.com       opt-in-not-required us-west-1
REGIONS      ec2.us-west-2.amazonaws.com       opt-in-not-required us-west-2
```

Estas son las diferentes regiones que actualmente están disponibles para ser utilizadas. Para las prácticas utilizaremos la región por defecto que utiliza Amazon, us-east-1. Esta región está localizada en la costa este de Estados Unidos, en Virginia (EE.UU.). Cada región tiene diferentes zonas de disponibilidad (*availability zones*), que son diferentes centros de datos dentro de la misma región, con el objetivo de ofrecer alta disponibilidad dentro una región. Los fallos de una zona de disponibilidad no deben afectar a otra puesto que involucran máquinas y redes diferentes.

Es posible consultar las zonas de disponibilidad para una determinada región con la operación `describe-availability-zones`. A continuación, se listan las zonas de disponibilidad de la región us-east-1 (situada en en el Norte de Virginia, EE.UU).

```
:~$ aws ec2 describe-availability-zones --region us-east-1
AVAILABILITYZONES      us-east-1      us-east-1      opt-in-not-required      us-
east-1      available      usel-az1      us-east-1a
AVAILABILITYZONES      us-east-1      us-east-1      opt-in-not-required      us-
east-1      available      usel-az3      us-east-1b
AVAILABILITYZONES      us-east-1      us-east-1      opt-in-not-required      us-
east-1      available      usel-az4      us-east-1c
AVAILABILITYZONES      us-east-1      us-east-1      opt-in-not-required      us-
east-1      available      usel-az6      us-east-1d
AVAILABILITYZONES      us-east-1      us-east-1      opt-in-not-required      us-
east-1      available      usel-az2      us-east-1e
AVAILABILITYZONES      us-east-1      us-east-1      opt-in-not-required      us-
east-1      available      usel-az5      us-east-1f
```

Si tratas de realizar una operación que no sea de consulta sobre otra región obtendrás un mensaje de error de autorización (únicamente tienes permisos para utilizar la región us-east-1).

Si no se especifica ningún modificador, los comandos realizan las operaciones sobre la región por defecto, que se haya indicado en el fichero de configuración de AWS CLI (disponible en \$HOME/.aws/credentials). Si deseas realizar las operaciones sobre otra región deberás usar el modificador `--region` en cada comando.

Despliegue de Máquinas Virtuales en AWS

En esta sección se plantea el despliegue de una máquina virtual de la forma más sencilla posible, a partir de una AMI (*Amazon Machine Image*) [19] ya existente. Para ello, se seguirá el proceso descrito en la Figura 1.



Figura 1. Acciones a realizar para el despliegue de instancias en Amazon EC2.

Creación del Par de Claves, Grupo de Seguridad y Elección de AMI

1. Construcción del par de claves.

Inicialmente será necesario construir un par de claves (*keypair*). Se trata de un par (clave pública, clave privada) utilizado para que el usuario pueda conectarse a la máquina virtual mediante SSH sin tener que especificar la contraseña. La clave privada la almacenará el usuario en su equipo. La clave pública se guarda en Amazon y se injectará en la máquina virtual en el momento del arranque. También es posible importar un par de claves que el usuario ya tuviera, para usarlas en EC2. Ten en cuenta que los pares de clave creados van ligados a una región concreta y no pueden ser compartidos entre diferentes regiones. El par de claves podrá ser compartido para acceder a otras instancias. Por lo tanto, puede crearse una única vez.

Usamos la opción `create-key-pair` dándole un nombre al par de claves (no es mala idea que el nombre incluya la región para la que ha sido creada, dado que el par de claves es específico para una región y no puede ser reutilizado para otras regiones, aunque por simplicidad le daremos un nombre básico). Fíjate como al usar la variable de entorno ID, esta se substituye por tu código de alumno.

```
:~$ aws ec2 create-key-pair --key-name alucloud$ID-keypair --query
'KeyMaterial' > alucloud$ID-priv.pem
:~$ cat alucloud$ID-priv.pem
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAjbPRQ5dpoAj4kMd9c+W/+yGJk81CwIDuLTbvbYDH78eTmlThcXcsoj4QOIT
5h8Hpj24d1JIojomcNwXjkHbnliD7SyECDQQu4Y1/05RAMWsgnhrw9TDOBdRDwcxySnVVU75664
G
...
-----END RSA PRIVATE KEY-----
```

Si obtienes un mensaje de error indicando que no hay permisos para escribir en el fichero alucloud\$ID-priv.pem, asegúrate de estar situado en el directorio \$HOME del usuario (al que puedes acudir ejecutando el comando cd).

La clave pública se habrá guardado automáticamente en Amazon EC2, mientras que la clave privada se ha guardado en el fichero alucloud\$ID-priv.pem (donde \$ID se substituye por tu código de usuario). Es importante que el fichero donde se almacena la clave privada solo tenga permisos de lectura y escritura para el usuario. Para ello, se puede cambiar los permisos con la siguiente operación:

```
:~$ chmod 0600 alucloud$ID-priv.pem
```

No pierdas la clave privada. No hay forma de recuperarla, ni siquiera desde la AWS Management Console. Afortunadamente, siempre es posible borrar un par de claves y volver a crearlo nuevamente. Antes de continuar, asegúrate que el contenido del fichero alucloud\$ID-priv.pem contiene efectivamente una clave privada (con el formato que se muestra en el ejemplo anterior).

Si en el momento de la creación del par de claves obtienes un mensaje indicando que el par de claves ya existe, lo más probable es que algún otro alumno se haya confundido y haya creado el par de claves usando tu identificador (por ejemplo que el alumno con identificador o6 haya creado el par de claves alucloudo3-keypair). No podrás acceder a las instancias desplegadas con dicho par de claves al no disponer de la clave privada. Por ello, en ese caso, y tras asegurarte de que estás utilizando tu identificador de alumno correcto, puedes proceder a borrar tu mismo el par de claves con el comando:

```
:~$ aws ec2 delete-key-pair --key-name alucloud$ID-keypair  
true
```

Luego puedes proceder a la creación de un nuevo par de claves. Es posible que el instructor haya restringido la eliminación de los pares de claves para limitar que unos alumnos interfieran con otros. Por ello, si recibes un mensaje de error indicando que no es posible eliminarlo, podrás proceder a crear uno nuevo utilizando un nombre (por ejemplo alucloud\$ID-keypair2). Alternativamente puedes avisar al instructor para que proceda a la eliminación del par de claves.

Recuerda que la clave pública del par de claves se injectará en la instancia únicamente en el momento del despliegue de la misma. Por ello, si eliminas el par de claves y vuelves a crear uno nuevo con el mismo nombre, esto no provoca que se actualice la clave pública de las instancias desplegadas con dicho par de claves.

2. Construcción del grupo de seguridad

A continuación, es preciso definir el grupo de seguridad (*security group*) al que la instancia (máquina virtual) pertenecerá. Un grupo de seguridad define un conjunto de reglas de cortafuegos y permisos de conexión especificando qué tipo de conexiones y hacia qué puertos de la instancia se permiten las conexiones. Múltiples instancias pueden utilizar el mismo grupo de seguridad. Por ejemplo, si se pretende desplegar un servidor web que pretende ser administrado de forma remota vía SSH habrá que incluir la instancia en un grupo de seguridad que permita el acceso a los puertos 22 (SSH) y 80 (HTTP) desde Internet. El grupo de seguridad podrá ser compartido por parte de otras instancias. Por lo tanto, puede crearse una única vez.

Creamos un grupo de seguridad llamado gs-aws-\$ID que permita acceso al puerto TCP 80 (HTTP) desde cualquier punto de Internet y al puerto TCP 22 (SSH) únicamente desde el entorno de prácticas (la máquina lab.cursocloudaws.net). En esta práctica desplearemos las instancias sobre un VPC llamado default, con identificador - vpc-83a213fb, que es el comportamiento habitual para las nuevas

cuentas de AWS. Aunque más adelante estudiarás con detalle el concepto de VPC y crearás uno tú mismo, por el momento, es importante que sepas que un VPC es una sección aislada de AWS para tu usuario con diferentes subredes, cada una de ellas en una zona de disponibilidad diferente. Crearemos el grupo de seguridad dentro del VPC.

Fíjate en el uso de CIDR (*Classless Inter-Domain Routing*) para definir los rangos de direcciones que pueden tener acceso a la instancia. En este caso se permite desde cualquier punto de Internet.

```
:~$ aws ec2 create-security-group --vpc-id vpc-83a213fb --group-name gs-aws-$ID --description 'Puertos 80 y 22 abiertos'  
sg-cc8b8dbe
```

Fíjate como el comando anterior te devuelve un identificador que tendrás que utilizar a continuación para establecer las reglas de autorización de entrada de tráfico TCP. Ten a mano dicho identificador porque te hará falta más adelante en varias ocasiones. Por tanto, recuerda cambiar el valor sg-c3afaeb1 por el identificador de grupo de seguridad que hayas obtenido.

Averiguamos en primer lugar la IP del entorno de prácticas con el comando (en este ejemplo usamos lab.cursocloudaws.net pero asegúrate de indicar el nombre de máquina que te haya proporcionado el instructor en el correo de bienvenida¹):

```
:~$ nslookup lab.cursocloudaws.net  
  
Server:      10.0.0.2  
Address:     10.0.0.2#53  
  
Non-authoritative answer:  
Name: lab.cursocloudaws.net  
Address: 54.157.106.156 ← Esta es la IP pública de la máquina lab.cursocloudaws.net
```

Creamos las reglas:

```
:~$ aws ec2 authorize-security-group-ingress --group-id sg-c3afaeb1 --  
protocol tcp --port 22 --cidr 54.157.106.156/32  
  
:~$ aws ec2 authorize-security-group-ingress --group-id sg-c3afaeb1 --  
protocol tcp --port 80 --cidr 0.0.0.0/0
```

¹ Ten en cuenta que es posible que tu instructor te haya asignado la máquina [lab2.cursocloudaws.net](#), por lo que no te equivoques en este punto. En el correo de bienvenida tienes indicada la máquina asignada.

Es posible obtener un listado de todos los grupos de seguridad creados en el VPC “default” (cuyo identificador es vpc-83a213fb” mediante la opción describe-groups.

```
:~$ aws ec2 describe-security-groups --filters "Name=vpc-id,Values=vpc-83a213fb"

SECURITYGROUPS Puerto 80 y 22 abiertos      sg-e9161f82      gs-aws-04      974349055189
IPPERMISSIONS 22      tcp      22
IPRANGES      52.202.41.247 /32
IPPERMISSIONS 80      tcp      80
IPRANGES      0.0.0.0/0
SECURITYGROUPS Puerto 80 y 22 abiertos      sg-751e1f1e      gs-aws-03      974349055189
IPPERMISSIONS 22      tcp      22
IPRANGES      0.0.0.0/0
IPPERMISSIONS 80      tcp      80
IPRANGES      0.0.0.0/0
```

Ten en cuenta que te aparecerán listados los grupos de seguridad creados tanto por ti como por el resto de alumnos del curso, debido al mecanismo de gestión de usuarios empleados.

En realidad, no hubiera sido necesario que cada alumno cree su propio grupo de seguridad, ya que, al representar un conjunto de reglas, puede reaprovecharse para diferentes usuarios. Sin embargo, esto permite conocer el proceso completo de despliegue de una instancia.

3. Selección de la Imagen de Máquina Virtual

Una imagen de máquina virtual incluye toda la configuración para desplegar una instancia en Amazon EC2. En el contexto de Amazon, éstas se denominan *Amazon Machine Image* (AMI), definidas como un tipo especial de sistema operativo pre-configurado y software de aplicación usado para crear una máquina virtual (instancia) en Amazon EC2. Sirve como una unidad básica de despliegue para los servicios desplegados en EC2.

Existen muchas AMIs de acceso público, y el usuario puede crear una AMI específica, bien desde cero o a partir de otra AMI existente. Existen AMIs con diferentes variantes de Linux, Windows y OpenSolaris. Existen dos tipos de AMIs en AWS:

- AMIs basadas en S3 (denominadas *Instance-Store Images*). En las instancias de EC2 desplegadas a partir de estas AMIs, cualquier fichero nuevo creado o modificación de los existentes, desaparecerá cuando la instancia termine. En realidad, cuando se arranca la instancia es como si se clonase la imagen temporalmente, por lo que, al desplegar una nueva instancia de la misma imagen, ésta siempre mantiene la misma configuración base. Este tipo de instancias no pueden detenerse (stop) e iniciarse (start). Por el contrario, tan solo pueden ser reiniciadas o terminadas. Si se reinicia, los datos del sistema de archivos no se pierden. Tan solo cuando se termine la instancia.
- AMIs basadas en EBS (denominadas *EBS Images*). En las instancias de EC2 desplegadas a partir de estas AMIs, se crea volumen EBS que mantiene los datos de la misma y que se destruye cuando se termina la instancia pero que se mantiene al detenerla. Por tanto, los cambios realizados en estas instancias sí que se mantienen si ésta se detiene (stop) y luego se inicia (start).

Además, las instancias pueden ser de 32 o de 64 bits, al igual que las AMIs. Esto es independiente del mecanismo de almacenamiento (basado en S3 o en EBS). Las AMIs basadas en EBS son las más modernas y tienen muchas ventajas con respecto a las de tipo *instance-store* (tamaño de partición más grande, posibilidad de detenerlas y volverlas a iniciar, etc.).

Es posible obtener un listado actualizado de las AMIs disponibles en AWS a través del AWS Marketplace [4], que es la opción más cómoda. Alternativamente, se puede utilizar la línea de comandos para obtener un listado de AMIs disponibles. Por ejemplo, para obtener un listado de las AMIs creadas por Amazon se utilizaría el comando:

```
:~$ aws ec2 describe-images --owners amazon

IMAGES x86_64 Microsoft Windows Server 2012 RTM 64-bit Locale Italian Base AMI
provided by Amazon xen ami-ff2c0896 amazon/Windows_Server-2012-RTM-Italian-
64Bit-Base-2013.11.13 amazon machine Windows_Server-2012-RTM-
Italian-64Bit-Base-2013.11.13 801119661308 windows True /dev/sda1 ebs
available hvm
BLOCKDEVICEMAPPINGS /dev/sda1
EBS True snap-4455db5d30 standard
...
```

Ten en cuenta que la ejecución del comando anterior tardará unos segundos. La información ofrecida por este comando es bastante básica y únicamente permite conocer, entre otros, el identificador de las AMIs, su arquitectura y si son de tipo instance-store (basada en S3) o de tipo EBS. No es posible conocer más detalles sobre el software instalado en dicha AMI.

Si deseamos ampliar la información hay que acudir al AWS Marketplace (disponible en <https://aws.amazon.com/marketplace>). Ten en cuenta que necesitas credenciales de acceso a la AWS Management Console para realizar ciertas operaciones con el AWS MarketPlace, como la obtención de información detallada sobre las AMIs y tus credenciales de alumno tienen limitados algunos permisos (para evitar suscripciones a AMIs de pago, por ejemplo). Por ello, la opción más cómoda es acudir a la consola de EC2 y pulsar sobre el botón “Launch Instance”:

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Launch Instance

Luego, en la barra lateral elige la opción “Community AMIs” (ten en cuenta que si eliges la opción AWS Marketplace no podrás ver el identificador de AMI)

Step 1: Choose an Amazon Machine Image (AMI)

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

My AMIs

AWS Marketplace

Community AMIs

Search community AMIs

amzn2-ami-hvm-2.0.20180622.1-x86_64-gp2 - ami-b70554c8

Amazon Linux 2 AMI 2.0.20180622.1 x86_64 HVM gp2

Select

64-bit

Root device type: ebs Virtualization type: hvm ENI Enabled: Yes

Ahora, en dicho catálogo podemos buscar una AMI que incluya una configuración de LAMP (Linux, Apache, MySQL y PHP) [16] para disponer de un servidor web ya preinstalado. Para ello introduce en la barra de búsqueda texto “bitnami-lamp”² (sin las comillas dobles) y pulsa Enter.

Step 1: Choose an Amazon Machine Image (AMI)

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.

bitnami-lamp

AMI Name	AMI ID	Type
bitnami-lampstack-7.1.18-1-linux-debian-9-x86_64-hvm-ebs	ami-00232ad584ddcf6a4	64-bit (x86)
bitnami-lampstack-5.5.13-0-dev-linux-ubuntu-12.04.4-x86_64-ebs-ami-dcb041b4-3-ami-f8189d90	ami-00266568	64-bit (x86)
bitnami-lampstack-5.6.35-0-r11-linux-debian-9-x86_64-hvm-ebs	ami-0026bdb66251e88e	64-bit (x86)
bitnami-lampstack-5.5.10-2-dev-linux-ubuntu-12.04.3-x86_64-ebs-ami-6b687402-3-ami-d020a5b8	ami-003b7868	64-bit (x86)

La AMI que hemos elegido incluye el soporte de virtualización de tipo HVM (Hardware Virtualization Machine), puesto que lo hemos especificado en la búsqueda, mientras que otras soportan el tipo de virtualización PV (Paravirtual). Tienes una descripción de los distintos tipos de virtualización soportados por AWS en [25]. Asegúrate de que eliges la versión HVM de la AMI puesto que, de lo contrario, no te dejará utilizar los tipos de instancia que se indican en este boletín de prácticas. El

² Bitnami es una empresa que, entre otras funciones, produce AMIs para AWS: <https://bitnami.com/>

identificador de AMI, que depende de cada región, aparece justo al lado del nombre (por ejemplo: ami-0026bdbe66251e88e). En este caso es el identificado para la región us-east-1 (N. Virginia).

Ten en cuenta que es posible que el identificador de AMI para dicha región haya cambiado, puesto que el creador de la misma, al actualizar la versión del software incluido, produce una nueva AMI con diferente identificador (incluso puede llegar a de-registrar una AMI). Si ese es el caso, deberás elegir el identificador de AMI actual para la región us-east-1 para desplegar la instancia.

Despliegue de Instancias de Máquinas Virtuales

1. Despliegue de la instancia en EC2

Desplegamos una instancia de tipo `t2.micro` en una subred del VPC especificando como grupo de seguridad el que hemos creado anteriormente (`gs-aws-$ID`).

En primer lugar, averiguamos las subredes disponibles en el VPC:

```
:~$ aws ec2 describe-subnets --filters "Name=vpc-id,Values=vpc-83a213fb"
SUBNETS      False us-east-1c 4088 172.31.32.0/20      False True available
            subnet-432a9408 vpc-83a213fb
TAGS  Name subnet-default-1c-public
TAGS  owner grycap
SUBNETS      False us-east-1a 4085 172.31.0.0/20      False True available
            subnet-2bfb6c4f vpc-83a213fb
TAGS  Name subnet-default-1a-public
TAGS  owner grycap
SUBNETS      False us-east-1f 4091 172.31.80.0/20      False False available
            subnet-251a7f2a vpc-83a213fb
TAGS  Name subnet-default-1f-public
TAGS  owner grycap
SUBNETS      False us-east-1d 4091 172.31.48.0/20      False True available
            subnet-13c9384e vpc-83a213fb
TAGS  Name subnet-default-1d-public
TAGS  owner grycap
SUBNETS      False us-east-1e 4090 172.31.64.0/20      False True available
            subnet-c0c33fef vpc-83a213fb
TAGS  owner grycap
TAGS  Name subnet-default-1e-public
SUBNETS      False us-east-1b 4091 172.31.16.0/20      False True available
            subnet-c2f25afed vpc-83a213fb
TAGS  owner grycap
TAGS  Name subnet-default-1b-public
```



Los valores de atributos `Name` y `Values` indicados en `--filters` son sensibles a mayúsculas. Si utilizas `name` y/o `values` (con la primera letra en minúscula) obtendrás el siguiente fallo:

```
$ aws ec2 describe-instances --filters "Name=key-name,values=alucloud$ID-keypair"
```

```
Error parsing parameter '--filters': Unknown key 'values', valid choices are: Values, Name
```

Verás que existe una subred por cada zona de disponibilidad de la region us-east-1. Esta es la configuración habitual de un VPC por defecto. Por tanto, si queremos desplegar en la zona de disponibilidad us-east-1d entonces habrá que desplegar la instancia en la subred `subnet-13c9384e`.

Desplegamos la instancia en dicha subred. Para ello usamos el comando:

```
:~$ aws ec2 run-instances --image-id ami-4b43a431 --key-name alucloud$ID-
  keypair --security-group-ids sg-cc8b8dbe --instance-type t2.micro --
  subnet-id subnet-13c9384e

974349055189      r-0acee30936780cf3
INSTANCES 0      x86_64      False xen    ami-4b43a431      i-
0919b839b29d10964 t3.micro    alucloud00-keypair    2017-11-
06T16:23:36.000Z ip-172-31-84-184.ec2.internal 172.31.84.184
               /dev/sda1 ebs   True      subnet-13c9384e hvm    vpc-83a213fb
MONITORING disabled
NETWORKINTERFACES      16:c9:fe:86:d7:a4 eni-f6e46154      974349055189
               ip-172-31-84-184.ec2.internal 172.31.84.184      True  in-use
               subnet-13c9384e vpc-83a213fb
ATTACHMENT 2017-11-06T16:23:36.000Z      eni-attach-3ea9792e      True  0
               attaching
GROUPS      sg-cc8b8dbe gs-aws-00
PRIVATEIPADDRESSES      True  ip-172-31-84-184.ec2.internal 172.31.84.184
PLACEMENT us-east-1f      default
SECURITYGROUPS      sg-cc8b8dbe gs-aws-00
STATE 0      pending
STATEREASON pending      pending
```

Cada instancia lleva asociado un identificador único, que en este caso es i-0919b839b29d10964. Es importante que anotes los identificadores de las instancias que vas desplegando para posteriormente terminarlas cuando ya no sean necesarias.

El tipo de instancia elegido (*t2.micro*) determina las características del hardware virtual y, por lo tanto, las prestaciones de la instancia. Puedes encontrar la información más actualizada sobre los precios de las instancias, así como el resto de tipos de instancias en [13] aunque una buena página para consultarlos de forma fácil es <http://ec2instances.info/>

Ten en cuenta si eliges otro tipo de instancia (como por ejemplo, *t3.micro*), es posible que obtengas un mensaje de error, pues las versiones de las AMIs elegidas deben estar adaptadas para los tipos de instancia más modernos.

An error occurred (InvalidParameterCombination) when calling the RunInstances operation: Enhanced networking with the Elastic Network Adapter (ENA) is required for the 't3.micro' instance type. Ensure that you are using an AMI that is enabled for ENA

El precio de los tipos de instancia depende de la región en la que se desplieguen. Los precios mostrados en la tabla se corresponden con la región *us-east-1*. Aunque se muestre el precio por hora, la facturación se realiza por segundos.

La instancia se habrá desplegado en la región *us-east-1* (Virginia del Norte, USA) puesto que así lo indica la configuración de la herramienta AWS CLI. Al haber elegido una subred concreta creada en una zona de disponibilidad concreta, esta instancia se ha desplegado en la zona de disponibilidad *us-east-1f*. Obviamente, es posible cambiar la región y la zona de disponibilidad en la que se despliega una instancia mediante las opciones de la AWS CLI.

Es posible conocer el estado de todas las instancias desplegadas en Amazon EC2 por el mismo usuario, mediante el comando `describe-instances`.

```
:~$ aws ec2 describe-instances

RESERVATIONS      974349055189          r-42beae23
GROUPS           sg-a2912fcb default
INSTANCES        0       x86_64    VXrev1369304998014      False xen ami-
0781e96e        i-960d58b8   m1.small   aki-825ea7eb      micafer 2014-01-
27T09:16:54.000Z None None /dev/sda1    ebs     User initiated (2014-01-28
16:09:57 GMT) paravirtual
BLOCKDEVICEMAPPINGS /dev/sda1
EBS   2013-05-23T10:30:02.000Z      True attached vol-f97d73a1
MONITORING disabled
PLACEMENT us-east-1d None default
SECURITYGROUPS sg-a2912fcb default
STATE 80 stopped
STATEREASON Client.UserInitiatedShutdown Client.UserInitiatedShutdown:
User initiated shutdown
TAGS Name cDMI-FTP
RESERVATIONS      974349055189          r-958061ec
GROUPS           sg-a2912fcb default
INSTANCES        0       x86_64    None False xen ami-e50e888c      i-
736d3b0a        t1.micro   aki-825ea7eb      alucloud00-keypair 2014-01-
16T13:59:21.000Z None None /dev/sda1    ebs     User initiated (2014-02-09
07:19:39 GMT) paravirtual
BLOCKDEVICEMAPPINGS /dev/sda1
EBS   2013-12-05T12:00:50.000Z      True attached vol-f7560980
MONITORING disabled
PLACEMENT us-east-1c None default
SECURITYGROUPS sg-a2912fcb default
STATE 80 stopped
STATEREASON Client.UserInitiatedShutdown Client.UserInitiatedShutdown:
User initiated shutdown
TAGS Name iACAWS
```

En el resultado se muestran todas las instancias que también han lanzado tus compañeros por lo que te resultará difícil encontrar de forma sencilla la información de la instancia que acabas de desplegar. Por ello, es posible solicitar la información de descripción de una única instancia (o varias de ellas) con el siguiente comando (para que veas un formato de información alternativo forzamos a obtener el resultado en formato JSON mediante la opción --output):

```
:~$ aws ec2 describe-instances --output json --instance-ids i-960d58b8

{
  "Reservations": [
    {
      "Instances": [
        {
          "Monitoring": {
            "State": "disabled"
          },
          "PublicDnsName": "ec2-34-238-192-217.compute-1.amazonaws.com",
          "State": {
            "Code": 16,
            "Name": "running"
          },
          "EbsOptimized": false,
          "LaunchTime": "2017-11-06T16:23:36.000Z",
          "PublicIpAddress": "34.238.192.217",
```

```

    "PrivateIpAddress": "172.31.84.184",
    "ProductCodes": [
        {
            "ProductCodeId": "8w7jjeix980u6s4dbijmm2u24",
            "ProductCodeType": "marketplace"
        }
    ],
    "VpcId": "vpc-83a213fb",
    "StateTransitionReason": "",
    "InstanceId": "i-0919b839b29d10964",
    "EnaSupport": false,
    "ImageId": "ami-4b43a431",
    "PrivateDnsName": "ip-172-31-84-184.ec2.internal",
    "KeyName": "alucloud00-keypair",
    "SecurityGroups": [
        {
            "GroupName": "gs-aws-00",
            "GroupId": "sg-cc8b8dbe"
        }
    ],
    "ClientToken": "",
    "SubnetId": "subnet-13c9384e",
    "InstanceType": "t2.micro",
    "NetworkInterfaces": [
        {
            "Status": "in-use",
            "MacAddress": "16:c9:fe:86:d7:a4",
            "SourceDestCheck": true,
            "VpcId": "vpc-83a213fb",
            "Description": "",
            "NetworkInterfaceId": "eni-f6e46154",
            "PrivateIpAddresses": [
                {
                    "PrivateDnsName": "ip-172-31-84-
184.ec2.internal",
                    "PrivateIpAddress": "172.31.84.184",
                    "Primary": true,
                    "Association": {
                        "PublicIp": "34.238.192.217",
                        "PublicDnsName": "ec2-34-238-192-
217.compute-1.amazonaws.com",
                        "IpOwnerId": "amazon"
                    }
                }
            ],
            "PrivateDnsName": "ip-172-31-84-184.ec2.internal",
            "Attachment": {
                "Status": "attached",
                "DeviceIndex": 0,
                "DeleteOnTermination": true,
                "AttachmentId": "eni-attach-3ea9792e",
                "AttachTime": "2017-11-06T16:23:36.000Z"
            },
            "Groups": [
                {
                    "GroupName": "gs-aws-00",
                    "GroupId": "sg-cc8b8dbe"
                }
            ],
            "Ipv6Addresses": [],
            "OwnerId": "974349055189",
            "PrivateIpAddress": "172.31.84.184",
            "SubnetId": "subnet-13c9384e",
            "Association": {

```

```

        "PublicIp": "34.238.192.217",
        "PublicDnsName": "ec2-34-238-192-217.compute-
1.amazonaws.com",
        "IpOwnerId": "amazon"
    }
}
],
"SourceDestCheck": true,
"Placement": {
    "Tenancy": "default",
    "GroupName": "",
    "AvailabilityZone": "us-east-1f"
},
"Hypervisor": "xen",
"BlockDeviceMappings": [
{
    "DeviceName": "/dev/sda1",
    "Ebs": {
        "Status": "attached",
        "DeleteOnTermination": true,
        "VolumeId": "vol-0b6101c56a72320ab",
        "AttachTime": "2017-11-06T16:23:37.000Z"
    }
}
],
"Architecture": "x86_64",
"RootDeviceType": "ebs",
"RootDeviceName": "/dev/sda1",
"VirtualizationType": "hvm",
"Tags": [
{
    "Value": "alucloud00",
    "Key": "owner"
}
],
"AmiLaunchIndex": 0
}
],
"ReservationId": "r-0acee30936780cfcc3",
"Groups": [],
"OwnerId": "974349055189"
}
]
}
}

```

Se observa que el identificador de la instancia es, en este caso particular es i-0919b839b29d10964 (indicado por el atributo *InstanceId*). El nombre DNS público de la instancia, a la que podrás conectarte por SSH, es ec2-34-238-192-217.compute-1.amazonaws.com (indicando por el atributo *PublicDnsName*).

Si lo que deseas es obtener un listado de las instancias que has desplegado tú entonces puedes especificar un filtro de búsqueda que sólo muestre aquellas instancias que han sido lanzadas con un determinado par de claves. Como el par de claves de cada alumno tiene un nombre único, sirve como discriminante adecuado. El comando sería:

```
:~$ aws ec2 describe-instances --filters "Name=key-
name,Values=alucloud$ID-keypair"
```

Si únicamente te interesa conocer el identificador de la instancia y el nombre publico (FQDN), siempre puedes filtrar más la información anterior usando el comando `grep` para quedarte únicamente con las filas donde aparece la palabra INSTANCES.

```
:~$ aws ec2 describe-instances --filters "Name=key-name,Values=alucloud$ID-keypair" | grep INSTANCES
INSTANCES      0      x86_64 None  False  xen    ami-e50e888c i-736d3b0a  t1.micro
                aki-825ea7eb alucloud00-keypair 2014-01-16T13:59:21.000Z  None   None
                /dev/sda1   ebs    User initiated (2014-02-09 07:19:39 GMT)  paravirtual
INSTANCES      0      x86_64 None  False  xen    ami-e50e888c i-849458aa  t1.micro
                aki-825ea7eb alucloud00-keypair 2014-01-08T15:30:08.000Z  ip-10-225-18-
20.ec2.internal 10.225.18.20 ec2-107-22-226-77.compute-1.amazonaws.com
                107.22.226.77/dev/sda1   ebs    None   paravirtual
```

Alternativamente, quizá es más aconsejable utilizar la opción `--query` de AWS CLI para parsear directamente la salida y obtener únicamente el identificador y el nombre DNS de las instancias desplegadas con un determinado par de claves, de la siguiente manera:

```
:~$ aws ec2 describe-instances --filters "Name=key-name,Values=alucloud$ID-keypair" --query
'Reservations[*].Instances[*].[InstanceId,PublicDnsName]'
i-736d3b0a
i-849458aa  ec2-107-22-226-77.compute-1.amazonaws.com
i-73279b5d
i-fcc5bed2  ec2-23-20-121-92.compute-1.amazonaws.com
i-960d58b8  ec2-54-226-177-26.compute-1.amazonaws.com
i-88113da8  ec2-54-221-241-27.compute-1.amazonaws.com
```

Ten en cuenta que hay que respetar las mayúsculas en los nombres de atributos (Name, Values, Reservations, Instances, InstanceId, PublicDnsName) para que funcione correctamente el comando.

Quizá te sorprenda ver en el listado instancias sin nombre DNS publico asociado (aparece una línea en blanco en su lugar). Eso es debido a que dichas instancias no están en ejecución, sino que están detenidas (temporalmente para evitar gastos).

Es posible asignarle un nombre a la instancia estableciendo una etiqueta a la misma. De esta manera, en la AWS Console se mostrará con un nombre concreto. Para ello se utiliza la opción `--tag` :

```
:~$ aws ec2 create-tags --resources i-960d58b8 --tags
Key=Name,Value="nombre de la instancia"
```

A veces es posible que, por cuestiones de mantenimiento, alguna zona de disponibilidad de Amazon no esté disponible. En ese caso, se obtendría el siguiente mensaje:

```
| Unsupported | The requested Availability Zone is currently constrained
and we are no longer accepting new customer requests for t1/m1/c1/m2
instance types. Please retry your request by not specifying an
Availability Zone or choosing us-east-1c, us-east-1d, us-east-1e. |
```

En ese caso, hay que desplegar la instancia en cualquier otra zona de disponibilidad de la misma región. Por ejemplo, para desplegar la instancia en la zona de disponibilidad us-east-1e de la región us-east-1, hay que buscar la subred del VPC que ha sido creada en dicha zona de disponibilidad:

```
:~$ aws ec2 describe-subnets --filters "Name=vpc-id,Values=vpc-83a213fb" | grep us-east-1e
SUBNETS      False us-east-1e 4091 172.31.64.0/20      False True available
subnet-c0c33fef      vpc-83a213fb
```

A la vista del resultado, la subred subnet-c0c33fef ha sido creada en la zona de disponibilidad us-east-1e. Ahora, se puede desplegar la instancia en dicha subred con el siguiente comando (aunque no es necesario que lo ejecutes si no quieres tener una nueva instancia):

```
:~$ aws ec2 run-instances --image-id ami-4b43a431 --key-name alucloud$ID-keypair --security-group-ids sg-cc8b8dbe --instance-type t2.micro --subnet-id subnet-c0c33fef
```

Además, ten en cuenta que es posible configurar la herramienta AWS CLI para que muestre la información de salida en diferentes formatos. Concretamente, mediante un formato de texto (--output text), que es el que está configurado por defecto en el entorno de prácticas, o mediante un formato tabular (--output table) o mediante un formato JSON (--output json). Prueba las diferentes opciones y utiliza aquella con la que te sientas más cómodo.

2. Conexión a la instancia

Dado que la máquina virtual dispone de servidor web preinstalado y servidor SSH, y que hemos habilitado los correspondientes puertos en el grupo de seguridad, debe ser posible acceder a la misma tanto por SSH (puerto 22), desde un cliente SSH, como por HTTP (puerto 80) desde un navegador web. Hasta que la instancia no esté en estado *running* y haya pasado el suficiente tiempo para que arranque el SO, el servidor SSH y se copie la clave pública a la instancia, no será posible acceder a ella. En poco más de 2 minutos debería de estar disponible.

La conexión por ssh se realiza mediante el siguiente comando (recuerda utilizar la IP correspondiente o el nombre DNS. Es necesaria la clave privada para conectarte a una instancia de EC2. De hecho, si quisieras conectarte mediante usuario y contraseña deberás cambiar la configuración del servidor SSH [27]. Si no consigues conectar sigue leyendo un poco más adelante donde se explican posibles causas, antes de contactar con el instructor:

```
:~$ ssh -i alucloud$ID-priv.pem bitnami@ec2-54-226-177-26.compute-1.amazonaws.com
The authenticity of host ec2-54-226-177-26.compute-1.amazonaws.com
(10.31.224.79)' can't be established.
ECDSA key fingerprint is cb:43:89:75:17:ad:99:bc:0b:7b:86:c7:fc:8d:15:e6.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-54-81-30-200.compute-1.amazonaws.com,10.31.224.79' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 12.04.4 LTS (GNU/Linux 3.2.0-58-virtual x86_64)
```

En primer lugar, se te muestra la huella (*fingerprint*) de la clave ECDSA [23] del *host* (la instancia de EC2). Debes saber que cuando una instancia de EC2 arranca, se crean varias claves de host (una de tipo RSA, otra DSA y otra de tipo ECDSA) que permiten identificar la instancia. Al conectarte por SSH se te muestra la huella de la clave del host para que puedas verificar que efectivamente te estás conectando a dicha instancia y no a cualquier otra máquina que pueda estar efectuando algún ataque de tipo man-in-the-middle [24]. Puedes obtener la información sobre la clave ECDSA del host con el siguiente comando:

```
:~$ aws ec2 get-console-output --instance i-f4468d15 | grep ECDSA
+-[ECDSA 256]---+
ec2: 256 cb:43:89:75:17:ad:99:bc:0b:7b:86:c7:fc:8d:15:e6 root@ip-10-158-
72-71 (ECDSA)
```

Verás que en un punto del log se indica cual es el fingerprint de dicha clave ECDSA recién creada, que debe coincidir con la que te indica el comando ssh. Durante las prácticas, no es necesario que realices este procedimiento, pero no está de más conocerlo.

Fíjate que al desplegar la instancia en EC2, la clave pública del usuario (especificada en el *keypair*) se habrá injectado en la cuenta de usuario *bitnami* de la instancia (máquina virtual). Por ello, le especificamos al cliente SSH que use la clave privada para conectarse como usuario *bitnami* a la instancia (fichero alucloud\$ID-priv.pem). Otro tipo de imágenes, basadas en otras distribuciones de Linux pueden requerir otra cuenta de usuario diferente para conectarse a la instancia (a menudo *ec2-user*). En concreto, las basadas en Red Hat (Fedora, CentOS) requieren conectarse como usuario *root*, mientras que las basadas en Ubuntu suelen elegir el usuario *ubuntu*. En este caso concreto fíjate que el usuario elegido es *bitnami*. Esto es debido a la configuración específica de dicha AMI.

Generalmente, el creador de la AMI debe aportar la información de uso de la misma a través del AWS MarketPlace. Si seleccionas la AMI en el MarketPlace, pulsas sobre el botón “Continue” y eliges la pestaña “Manual Launch” verás que en “Usage Instructions” hay un enlace con información sobre la cuenta de usuario por defecto (bitnami), así como información sobre el software instalado en la misma.

Elegir una Amazon Machine Image (AMI)

Una AMI es una plantilla que contiene la configuración de software (sistema operativo, servidor de aplicaciones y aplicaciones) necesaria para lanzar la instancia. Puede seleccionar una AMI proporcionada por AWS o nuestra comunidad de usuarios, o bien a través de AWS Marketplace.

LAMP

LAMP packaged by Bitnami
Bitnami by VMware [\[\]](#)
4★ 38 revisiones de AWS [\[\]](#)
Free Tier

Información general | Detalles del producto | Precios | **Uso** | Soporte

64-bit (x86) Amazon Machine Image (AMI)

Instrucciones de uso

Once the instance is running, enter the public DNS provided by Amazon into your browser. You will then see the Bitnami Welcome Page. Click the 'Access my application' link to visit the LAMP application. Please check our documentation at <https://docs.bitnami.com/aws/faq/get-started/find-credentials/> to learn how to get your password. You may change this username and password within the application settings. You can also access your instance via SSH using the username 'bitnami' and your Amazon private key. For additional setup instructions and frequently asked questions please go to <https://docs.bitnami.com/aws/infrastructure/lamp/>

Recursos adicionales

LAMP packaged by Bitnami [\[\]](#)
User Guide [\[\]](#)
Changelog [\[\]](#)

Continuar

Si al conectarte vía SSH se te solicita una contraseña asegúrate de que: i) has esperado el tiempo suficiente desde que la instancia está en estado RUNNING antes de realizar la conexión SSH (no más de 2-3 minutos), dado que es necesario que se inyecte la clave pública en la instancia; ii) estás especificando el fichero alucloud\$ID-priv.pem como parámetro al comando ssh, dado que ese fichero contiene la clave privada necesaria para identificar al usuario y autorizar la conexión vía SSH sin contraseña a la instancia; iii) el contenido del fichero alucloud\$ID-priv.pem contiene efectivamente una clave privada. Si lo editas verás que comienza con la línea -----BEGIN RSA PRIVATE KEY---- y termina con la línea -----END RSA PRIVATE KEY----- (asegúrate de que no te olvidas ninguno de los cinco guiones delanteros ni traseros tanto en la primera como en la última línea y de que no hay líneas en blanco entre medias de la clave privada ni ningún carácter adicional) y iv) estás utilizando el nombre de cuenta apropiado para dicha AMI (en nuestro caso concreto es *bitnami*) en la conexión vía SSH.

Si por el contrario obtienes el mensaje de error “*Warning: Identity file alucloud\$ID-priv.pem not accessible: No such file or directory*”, es porque el comando ssh no puede encontrar el fichero de clave privada que le estás indicando (mediante el parámetro *-i*). Para resolverlo, asegúrate de que: i) estás indicando la ruta correcta al fichero de clave privada; ii) estás indicando el nombre correcto de la clave privada y iii) estás situado en el directorio correcto para acceder a ese fichero con la ruta que has indicado). Si obtienes el mensaje de error “*WARNING: UNPROTECTED PRIVATE KEY FILE*”, es porque te olvidaste de cambiar los permisos al fichero alucloud\$ID-priv.pem mediante el comando *chmod*, tal y como está explicado en la sección anterior.

Si obtienes algún error al tratar de conectar por SSH de tipo “*Connection timed out*”, asegúrate de que la máquina lleve en estado running al menos un par de minutos, que estés utilizando la IP pública (en lugar de la IP privada) o el nombre DNS correcto de tu instancia y que el grupo de seguridad con el que desplegarla la instancia permita la conexión al puerto 22 desde la máquina de prácticas (asegúrate de que la IP indicada es correcta). Por último, aunque muy poco probable, es posible que haya ocurrido algún problema al desplegarla. Para ello, deberás proceder a terminar la instancia y desplegar una nueva. Si el problema persiste puedes tratar de forzar a desplegar en una zona de disponibilidad diferente. Por ejemplo, desde la AWS Management Console es posible ver si una instancia ha superado los tests de accesibilidad (*System Status Check*, que verifica que el hardware sobre el que se ejecuta la instancia funciona correctamente e *Instance Status Check*, que verifica que el sistema operativo de la instancia está recibiendo tráfico y, por tanto, se ejecuta correctamente). Por ejemplo, en la figura

siguiente, el triángulo amarillo indica que uno de los dos tests no ha sido superado por lo que existe un problema con dicha instancia. En ese caso, se recomienda terminar la instancia y desplegarla nuevamente (quizá en otra zona de disponibilidad). Si el problema persiste puede ser problema de la AMI, por lo que habría que elegir una AMI diferente.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name
	i-00bf1464	t1.micro	us-east-1d	stopped	None				im-138600674815
	i-2f5a3401	m1.small	us-east-1c	stopped	None				alucloud12-keypair
<input checked="" type="checkbox"/>	i-55754275	m1.small	us-east-1e	running	⚠ 1/2 checks ...	None	ec2-54-80-...	54.80.89.189	alucloud18-keypair

Una vez conectado a la instancia, es posible ver algunas características de la misma, como el estado del sistema de archivos, la versión del kernel de Linux o la clave pública correspondiente al par de claves con el que ha sido desplegada la instancia.:

```
bitnami@ip-10-31-224-79:~$ df
Filesystem      1K-blocks    Used   Available Use% Mounted on
/dev/xvda1        10321208 1726920     8070004  18% /
udev              838012       8    838004   1% /dev
tmpfs             338512      172    338340   1% /run
none               5120       0      5120   0% /run/lock
none               846276       0    846276   0% /run/shm
/dev/xvdb        153899044 192068 145889352   1% /mnt

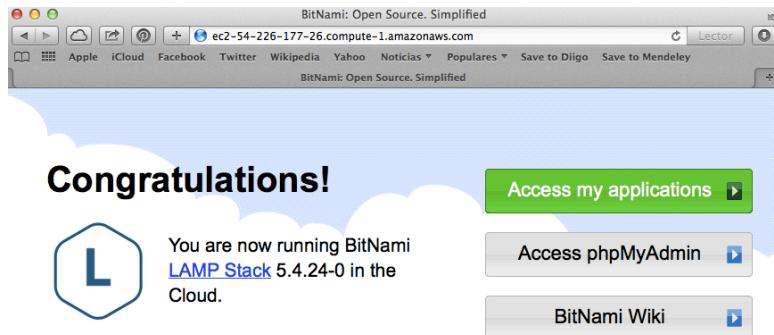
bitnami@ip-10-31-224-79:~$ uname -a
Linux ip-10-31-224-79 3.2.0-58-virtual #88-Ubuntu SMP Tue Dec 3 17:58:13
UTC 2013 x86_64 x86_64 x86_64 GNU/Linux

bitnami@ip-10-31-224-79:~$ cat $HOME/.ssh/authorized_keys
ssh-rsa
AAAAB3NzaC1yc2EAAAQABAAQChJPBWkwpFWaTp2kAtnJSmFZWhpLeX6ewZy5S0ZkaOJu
C1WahjqHhNXsifyczXh0RLe4/jkvTSTGPPlgLVESG6RZendhpmsog1UKI2GPemt9BFejy+hdO
e35M9Zd6Wv4AoWPtjHvP2IwuxDGu81xdf4yBCwlWC/nxWBEsAfsnfsguX1EcMRO0ULoLE6GeRKk
eP5saQnD8MEFTOyIAz1bPv9ORfOrC9975wMnEvABxY7jhfrnn3D72eb8yGYYAY+PmDXrSQmR1I
+f1Nwr77N5wi+7KRJ8a49ijCU6npDj1kAsAVUoMSe3thBbWBkJbf3w8nFRiywkVgZSA8SIDfh
alucloud00-keypair
```

Para salir de la sesión SSH con la instancia desplegada en Amazon EC2 y volver al *prompt* del entorno de realización de prácticas utiliza el comando exit:

```
bitnami@ip-10-31-224-79:~$ exit
```

Para verificar que el servidor web está activo, es posible abrir un navegador y conectarse a la IP (o nombre DNS) que identifica la instancia. En nuestro caso: <http://ec2-54-226-177-26.compute-1.amazonaws.com>

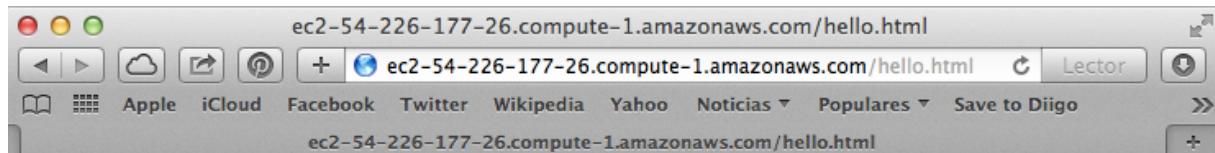


Si no recibes contestación del servidor web es posible que hayas olvidado desplegar la instancia de Amazon EC2 con un grupo de seguridad que permita el tráfico al puerto 80 (puerto en el que escucha el servidor web). En ese caso, deberás modificar el grupo de seguridad, de manera que todas las instancias lanzadas con dicho grupo de seguridad reflejarán automáticamente el cambio. La modificación del grupo de seguridad se puede hacer simplemente añadiendo una nueva regla a tu grupo de seguridad ya existente, para permitir la conexión al puerto 80. Revisa el apartado correspondiente a la creación de grupos de seguridad para asegurarte de que incluye las reglas apropiadas. Los cambios se aplicarán a todas las instancias de manera automática e instantánea.

Podemos probar a realizar un cambio en la página web por defecto del servidor web de la instancia para verificar que se actualiza desde el navegador web. La localización de los ficheros que muestra el servidor web Apache cambia de una distribución a otra. Por ejemplo, en Ubuntu suelen estar en /var/www/html. Sin embargo, en esta AMI dichos ficheros están en /opt/bitnami/apache2/htdocs.

```
bitnami@ip-10-31-224-79:~$ echo "<h1> Hello from the Cloud </h1>" > /opt/bitnami/apache2/htdocs/hello.html
```

Al acceder a la URL <http://ec2-54-226-177-26.compute-1.amazonaws.com/hello.html> se verá el nuevo mensaje:



Hello from the Cloud

Si experimentas algún problema en el arranque de una instancia en Amazon EC2 es posible obtener los mensajes que mostraría por pantalla dicha máquina durante el arranque. Para ello, es posible usar la AWS Management Console (explicado más adelante), seleccionar una instancia y elegir la opción “Get System Log”. Tienes más información sobre como determinar qué problemas tiene una instancia que no consigue arrancar en [15].

3. Terminación de la instancia

Dado que la instancia ha sido desplegada a partir de una AMI basada en EBS (en lugar de una AMI basada en S3), es posible detenerla para iniciarla nuevamente más adelante (incluso cambiando el tipo

de instancia, por ejemplo de t2.micro a t2.small) y seguir teniendo el mismo estado, es decir, el mismo contenido de los ficheros dentro de la máquina virtual (instancia). Esto es lo que se conoce como escalado vertical y permite ajustar la capacidad y prestaciones de la máquina virtual a las necesidades en cada momento (si bien hay un pequeño *downtime* [21] mientras se detiene, se cambia el tipo de instancia y se vuelve a iniciar).

Si hubiéramos desplegado una instancia basada en S3 (de tipo instance-store) y tratásemos de detenerla obtendríamos el siguiente mensaje de error:

```
Client.UnsupportedOperation: The instance 'i-db4544bd' does not have an
'ebs' root device type and cannot be stopped.
```

Terminamos la instancia:

```
:~$ aws ec2 terminate-instances --instance-ids i-0919b839b29d10964

TERMINATINGINSTANCES      i-0919b839b29d10964
CURRENTSTATE            32    shutting-down
PREVIOUSSTATE           16    running
```

Si consultamos a continuación el estado, se observa que está en estado *shutting-down*:

```
:~$ aws ec2 describe-instances --instance-ids i-0919b839b29d10964

RESERVATIONS      974349055189      r-0acee30936780cfcc3
INSTANCES        0      x86_64          False False xen    ami-4b43a431      i-
0919b839b29d10964 t2.micro       alucloud00-keypair      2017-11-
06T16:23:36.000Z          /dev/sda1    ebs    User initiated (2017-11-06
16:41:07 GMT)    hvm
MONITORING        disabled
PLACEMENT         us-east-1f      default
STATE             48    terminated
STATEREASON Client.UserInitiatedShutdown Client.UserInitiatedShutdown:
User initiated shutdown
TAGS      Name nombre de la instancia
TAGS      owner alucloud00
```

y finalmente pasa a estado *terminated*. En este momento se han liberado los recursos asociados y se deja de incurrir en gastos. Ten en cuenta que una instancia terminada no puede volver a ser iniciada. El volumen raíz desaparece y lo único que se conserva son los volúmenes EBS adicionales (no el raíz) que pudieran estar conectados a dicha instancia. La única opción posible de deshacer la terminación de una instancia es desplegar una nueva instancia a partir de la misma AMI.

Sobre las Características de las Instancias basadas en EBS

Estas instancias tienen carácter persistente ya que el despliegue de la instancia va ligado al despliegue de un volumen EBS que alojará el dispositivo raíz de la instancia [22]. Por ello, todos los cambios que se realicen en la instancia se verán reflejados al parar y volver a arrancar la instancia (a diferencia de lo que ocurre con una instancia basada en S3). Mientras una instancia basada en EBS está parada no genera gasto derivado de segundos de cómputo. Sin embargo, el volumen EBS asociado sí que lleva un coste asociado, incluso aunque la instancia esté parada. La terminación de la instancia basada en EBS provoca la destrucción del volumen EBS asociado.

A continuación, vamos a proceder lanzar otra instancia basada en EBS para ver cómo se crea un volumen EBS asociado, dónde se guardarán los datos de la instancia. Concretamente, se utilizará la AMI ami-of9fc25dd2506cf6, basada en Amazon Linux 2 con plataforma de 64 bits:

```
:~$ aws ec2 run-instances --image-id ami-0f9fc25dd2506cf6 --key-name alucloud$ID-keypair --security-group-ids sg-cc8b8dbe --instance-type t3.micro --subnet-id subnet-13c9384e

974349055189 r-0c46ffef3146c9ca2
INSTANCES 0 x86_64 False xen ami-8c1be5f6 i-023623b4c558f46a0
           t3.micro alucloud00-keypair 2017-11-06T16:45:10.000Z ip-172-31-95-
244.ec2.internal 172.31.95.244 /dev/xvda ebs True subnet-
13c9384e hvm vpc-83a213fb
MONITORING disabled
NETWORKINTERFACES 16:cc:9d:a7:95:50 eni-08e366aa 974349055189 ip-172-31-
95-244.ec2.internal 172.31.95.244True in-use subnet-13c9384e vpc-83a213fb
ATTACHMENT 2017-11-06T16:45:10.000Z eni-attach-f7ab7be7 True 0 attaching
GROUPS sg-cc8b8dbe gs-aws-00
PRIVATEIPADDRESSES True ip-172-31-95-244.ec2.internal 172.31.95.244
PLACEMENT us-east-1f default
SECURITYGROUPS sg-cc8b8dbe gs-aws-00
STATE 0 pending
STATEREASON pending pending
```

El listado de volúmenes refleja que al crear una instancia a partir de una imagen basada en EBS se procede automáticamente a la creación de un volumen en EBS (en este caso de 8 GBytes) para alojar los datos de la instancia. La información obtenida indica que el volumen está conectado a la instancia en /dev/xvda.

```
:~$ aws ec2 describe-volumes | grep i-023623b4c558f46a0

ATTACHMENTS 2022-04-26T16:41:04.000Z True /dev/xvda i-
0c8252f4b8d6f95b0 attached vol-0dc41effd00a4d9b8

:~$ aws ec2 describe-volumes --volume-ids vol-000af11c9bf2ddaf3
VOLUMES us-east-1d 2022-04-26T16:41:04.269Z False 100 False 8
           snap-0b25d4444bcce9352 in-use vol-0dc41effd00a4d9b8 gp2
ATTACHMENTS 2022-04-26T16:41:04.000Z True /dev/xvda i-
0c8252f4b8d6f95b0 attached vol-0dc41effd00a4d9b8
```

Averiguamos la IP de la nueva instancia con la opción `describe-instances`:

```
:~ $ aws ec2 describe-instances --instance-ids i-023623b4c558f46a0 --query 'Reservations[*].Instances[*].[InstanceId,PublicDnsName]'  
i-023623b4c558f46a0      ec2-34-236-187-67.compute-1.amazonaws.com
```

Recuerda que no es necesario usar la opción `--query` para obtener el nombre DNS de la instancia, pero su uso permite reducir la verbosidad del comando y mostrar únicamente los datos pedidos (identificador de instancia y nombre DNS público).

A continuación, nos conectamos por SSH a esta instancia (en este caso se requiere conectar como usuario `ec2-user` al ser una AMI basada en Amazon Linux) y a verificar algunos datos sobre la misma:

```
:~ $ ssh -i alucloud$ID-priv.pem ec2-user@ec2-54-197-109-185.compute-1.amazonaws.com  
The authenticity of host 'ec2-54-197-109-185.compute-1.amazonaws.com (54.197.109.185)' can't be established.  
RSA key fingerprint is b9:5e:4b:9b:78:0b:30:a5:02:08:00:d0:c0:fa:ad.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'ec2-54-197-109-185.compute-1.amazonaws.com,54.197.109.185' (RSA) to the list of known hosts.  
  
_ _ | _ _ )  
_ | ( _ _ /     Amazon Linux 2 AMI  
_ _ \| _ _ |  
  
https://aws.amazon.com/amazon-linux-2/  
[ec2-user@ip-10-191-82-23 ~]$ df -h  
Filesystem           Size  Used Avail Use% Mounted on  
devtmpfs            462M    0  462M   0% /dev  
tmpfs              470M    0  470M   0% /dev/shm  
tmpfs              470M  400K  470M   1% /run  
tmpfs              470M    0  470M   0% /sys/fs/cgroup  
/dev/nvme0n1p1     8.0G  1.6G  6.5G  20% /  
tmpfs              94M    0   94M   0% /run/user/1000
```

Es posible que obtengas temporalmente un mensaje de “Connection refused” aunque el estado de la instancia sea RUNNING. Efectivamente, hasta que el servidor SSH no esté activo no será posible conectar. En pocos minutos debería estar disponible la instancia para ser accedida mediante SSH. Una vez dentro de la máquina virtual, se observa que el disco principal es de 8 GBytes (correspondiente al volumen EBS que se creó automáticamente, de los cuales 1.6 GBytes están en uso para el Sistema Operativo).

Para verificar que efectivamente los cambios realizados en la instancia se mantienen, aunque esta se detenga, creamos un nuevo fichero en la misma y a detenerla posteriormente. Fíjate en el uso de `stop-instances` en lugar de `terminate-instances`. Esta última opción sí que procede a eliminar tanto la instancia como el volumen EBS asociado, provocando la pérdida de todos los datos que hubieran podido ser generados en la instancia.

```
[ec2-user@ip-10-191-82-23 ~]$ echo "Hola" > hola.txt  
[ec2-user@ip-10-191-82-23 ~]$ exit
```

```
(El siguiente comando se ejecuta desde el entorno de prácticas).
:~$ aws ec2 stop-instances --instance-ids i-023623b4c558f46a0
STOPPINGINSTANCES i-023623b4c558f46a0
CURRENTSTATE      64      stopping
PREVIOUSSTATE     16      running
```

La instancia parada no genera gasto adicional por segundos de instancia, pero sí cuesta el volumen EBS que, obviamente, no ha sido destruido. Se puede observar que la instancia no se destruye (tan solo ha sido parada) y que el volumen asociado aparece como todavía conectado a la instancia.

```
:~ $ aws ec2 describe-volumes --output json --volume-ids vol-000af11c9bf2ddaf3
{
    "Volumes": [
        {
            "AvailabilityZone": "us-east-1f",
            "Attachments": [
                {
                    "AttachTime": "2017-11-06T16:45:11.000Z",
                    "InstanceId": "i-023623b4c558f46a0",
                    "VolumeId": "vol-000af11c9bf2ddaf3",
                    "State": "attached",
                    "DeleteOnTermination": true,
                    "Device": "/dev/xvda"
                }
            ],
            "Encrypted": false,
            "VolumeType": "gp2",
            "VolumeId": "vol-000af11c9bf2ddaf3",
            "State": "in-use",
            "Iops": 100,
            "SnapshotId": "snap-080eb3cb2eda29974",
            "CreateTime": "2017-11-06T16:45:11.328Z",
            "Size": 8
        }
    ]
}
```

Iniciamos nuevamente la máquina y a verificar que el fichero sigue existiendo.

```
:~$ aws ec2 start-instances --instance-ids i-023623b4c558f46a0
STARTINGINSTANCES i-023623b4c558f46a0
CURRENTSTATE      0      pending
PREVIOUSSTATE     80      stopped

:~ $ aws ec2 describe-instances --instance-ids i-023623b4c558f46a0 --query 'Reservations[*].Instances[*].[InstanceId,PublicDnsName]'
i-023623b4c558f46a0      ec2-174-129-150-68.compute-1.amazonaws.com

:~$ ssh -i alucloud$ID-priv.pem ec2-user@ec2-174-129-150-68.compute-1.amazonaws.com
[ec2-user@ip-10-140-14-47 ~]$ ls -l
total 4
```

```
-rw-rw-r-- 1 ec2-user ec2-user 5 May 28 13:48 hola.txt
```

Fíjate que al detener y posteriormente iniciar la nueva instancia, la IP ha cambiado. En realidad, al detener la instancia se libera su dirección pública por lo que queda temporalmente sin dirección pública hasta que vuelva a ser iniciada, momento en el que recibe una nueva dirección pública y, por tanto, un nuevo nombre DNS público.

Finalmente, una vez terminadas las tareas, es momento de destruir la instancia:

```
:~$ aws ec2 terminate-instances --instance-ids i-023623b4c558f46a0
TERMINATINGINSTANCES      i-023623b4c558f46a0
CURRENTSTATE            32    shutting-down
PREVIOUSSTATE           16    running
```

Tal y como se ha comentado anteriormente, la destrucción de la instancia basada en EBS provoca la destrucción del volumen EBS asociado.

¡Enhorabuena! Ya sabes cómo aprovisionar infraestructura de máquinas virtuales bajo demanda, de forma sencilla y mediante un modelo de pago por uso, sin necesidad de invertir en hardware y sin levantarte de la silla.

Usando la AWS Management Console

A lo largo de esta práctica se ha usado principalmente la línea de comandos (CLI) para interactuar con AWS. La línea de comandos facilita la composición y orquestación de despliegues, permitiendo crear scripts que automaticen tareas. Sin embargo, todos los servicios que se utilizan en esta práctica pueden ser empleados desde la AWS Management Console, accesible desde un navegador web.

Conéctate con las credenciales de IAM (Access Keys) que te suministró tu profesor a la siguiente dirección.

<https://grycap-aws.signin.aws.amazon.com/console>

Tendrás acceso al EC2 Dashboard que te permite tener una visión global de los recursos de cómputo desplegados sobre Amazon EC2, tal y como se muestra en la Figura 2

Desde la interfaz web podrás crear pares de clave, grupos de seguridad, gestionar el ciclo de vida de las instancias, etc. Además, podrás revisar el resto de servicios que conforman el catálogo de productos de AWS.

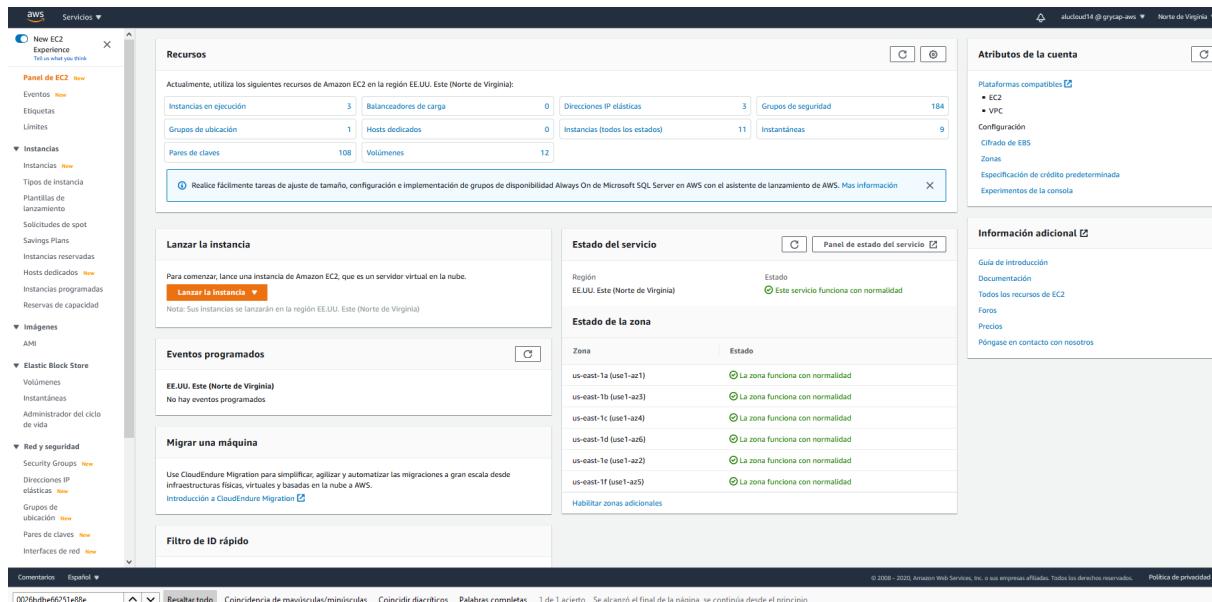


Figura 2. Aspecto del EC2 Dashboard.

Despliegue de Instancias Basadas en Windows

En esta sección se describe cómo desplegar instancias Windows en Amazon EC2. La conexión a la instancia será mediante el protocolo de escritorio remoto (*Remote Desktop Protocol*, RDP). Para ello, será necesario desplegar una instancia de una AMI basada en Windows, utilizando un grupo de seguridad que permita la conexión al puerto usado por RDP (que es el 3389), y posteriormente conectarse a la AWS Management Console para obtener la contraseña necesaria para iniciar sesión en la instancia Windows vía el cliente RDP. A continuación, se detalla paso a paso las acciones necesarias utilizando la línea de comandos. Recuerda que todas estas acciones pueden realizarse también usando la AWS Management Console.

Creación del Grupo de Seguridad y Selección de la AMI

Creamos un grupo de seguridad que permita el tráfico dirigido al puerto 3389 de las instancias de Amazon EC2 que se desplieguen con dicho grupo de seguridad:

```
:~$ aws ec2 create-security-group --vpc-id vpc-83a213fb --group-name gs-aws-$ID-win --description 'Puerto 3389 (RDP)'
sg-ee76769c
:~$ aws ec2 authorize-security-group-ingress --group-id sg-ee76769c --protocol tcp --port 3389 --cidr 0.0.0.0/0
```

Recuerda que si quisieras soportar algún servicio específico en dicha instancia deberías habilitar los puertos requeridos por dichos servicios. Por ejemplo, el puerto 80 para el servidor web Internet Information Server (IIS), el puerto 1433 para acceder a MS SQL Server, etc.

A continuación, elegimos una AMI. Para ello es posible acudir al listado de AMIs pulsando el botón “Launch Instance” de la consola de EC2, apartado “Community AMIs” y buscar alguna basada en Windows Server. Una sencilla búsqueda por “Windows Server 2019 Base” nos indica que, por ejemplo, en el momento de escribir estas líneas la AMI ami-09f2114fecbe506e2 (Microsoft Windows Server

2019 Base), basada en EBS, está disponible para la región us-east-1 (US East (Northern Virginia)), tal y como muestra la Figura 3.

The screenshot shows the AWS Lambda search interface. The search bar at the top contains the query "windows server2019 base". Below the search bar, there are four tabs: "Quickstart AMIs (0)", "My AMIs (0)", "AWS Marketplace AMIs (2879)", and "Community AMIs (2)". The "Community AMIs" tab is selected. On the left, a sidebar titled "Refine results" includes filters for "Operating system" (Linux/Unix and Windows), "Publish date range", and "Clear all filters". The main results area displays two items under the heading "Community AMIs":

- Microsoft packer-base-winserver2019-master**
ami-07c6a09b7ba342265
Platform: Windows Architecture: x86_64 Owner: 867686887310
Publish date: 2022-03-09 Root device type: ebs Virtualization: hvm ENA enabled: Yes
- Microsoft packer-base-winserver2019-v0.0.1**
ami-0dbdf36ae5f30d8d1
Platform: Windows Architecture: x86_64 Owner: 867686887310
Publish date: 2020-09-29 Root device type: ebs Virtualization: hvm ENA enabled: Yes

At the bottom of the results area, it says: "The following results for 'windows server2019 base' were found in other categories • 2879 results in AWS Marketplace AMIs AWS Marketplace AMIs are AMIs that are published by AWS & trusted third-parties".

Figura 3. Interfaz de búsqueda de AMIs

Si al tratar de desplegar una instancia de dicha AMI obtienes el mensaje de error “Not authorized for images: [ami-XXXX]” es porque el propietario ha cambiado los permisos de dicha AMI o la ha eliminado directamente y ya no puede ser instanciada. Esto ocurre con las AMIs basadas en Windows, donde al sacar una nueva versión eliminan las versiones anteriores. La solución pasa por buscar el identificador de AMI más reciente realizando una nueva búsqueda con el procedimiento anterior. Este identificador de AMI seguirá siendo válido hasta que saquen una nueva versión de la AMI, con un nuevo identificador.

Despliegue, Datos de Acceso y Conexión a la Instancia

Desplegamos una instancia de la AMI anterior con el grupo de seguridad creado en la fase anterior. Utilizaremos un tipo t2.small para tener memoria suficiente para ejecutar Windows.

```
:~$ aws ec2 run-instances --image-id ami-07c6a09b7ba342265
--key-name alucloud$ID-keypair --security-group-ids sg-9f2223ed --
instance-type t2.small --subnet-id subnet-13c9384e
974349055189      r-0d8ec05acac95e757
INSTANCES   0      x86_64          False xen    ami-56438a2c      i-
01587b7cf01f8c41e t2.small      alucloud00-keypair  2017-11-
06T17:11:17.000Z windows     ip-172-31-91-167.ec2.internal 172.31.91.167
                  /dev/sda1    ebs    True        subnet-13c9384e    hvm    vpc-
83a213fb
MONITORING disabled
NETWORKINTERFACES      16:b2:07:48:57:58 eni-6eed68cc      974349055189
```

```

ip-172-31-91-167.ec2.internal 172.31.91.167      True  in-use
subnet-13c9384e    vpc-83a213fb
ATTACHMENT 2017-11-06T17:11:17.000Z      eni-attach-aeaf7fbe      True  0
attaching
GROUPS      sg-9f2223ed gs-aws-00-win
PRIVATEIPADDRESSES  True  ip-172-31-91-167.ec2.internal 172.31.91.167
PLACEMENT  us-east-1f      default
SECURITYGROUPS  sg-9f2223ed gs-aws-00-win
STATE 0      pending
STATEREASON pending      pending

```

Antes de conectarte mediante escritorio remoto a la instancia hay que averiguar los datos de acceso (usuario y contraseña) a la instancia.

Para ello, utilizaremos el siguiente comando:

```

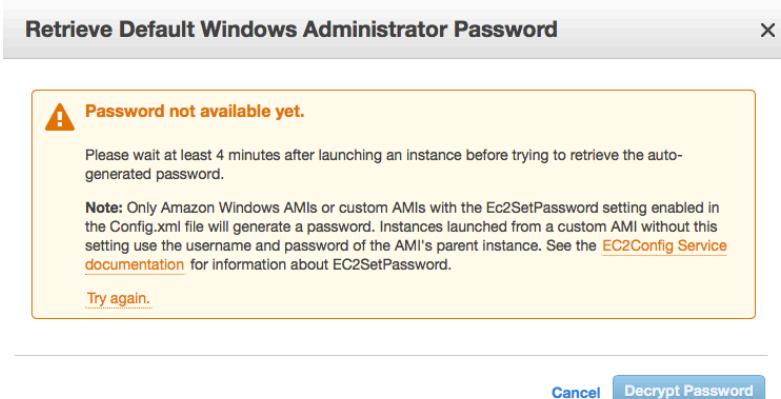
:~$ aws ec2 get-password-data --instance-id i-0b0ad5816f3f52010 --priv-
1aunch-key alucloud$ID-priv.pem

i-0b0ad5816f3f52010      V6jG(TH$zS9%.Pp3YLAeD*-Ecl2EonLR      2020-02-
24T09:53:57.000Z

```

El valor de la segunda columna del resultado es la contraseña del usuario *Administrator* para conectarte mediante VPC a la instancia Windows. Ten en cuenta que si ejecutas dicho comando y te aparece la contraseña vacía es que debes esperar todavía algún minuto más hasta obtener la contraseña.

Alternativamente, también es posible obtener dicha contraseña mediante la AWS Management Console. Deberás seleccionar la instancia y, con el botón derecho elegir la opción “Get Windows Password” (ver Figura 4). Es posible que tengas que esperar un tiempo desde que lanzas la instancia hasta que puedas obtener la contraseña, por lo que quizás obtengas un mensaje como el que se muestra en la siguiente figura. Se recomienda esperar al menos 4 minutos.



En el panel para obtener la contraseña de Windows necesaria para conectarte de forma remota a la instancia es necesario indicar la clave privada (disponible en el fichero alucloud\$ID-priv.pem) para descifrar dicha contraseña. Para ello puedes o bien elegir el fichero o copiar todo su contenido en el campo de texto, incluyendo las líneas de BEGIN RSA PRIVATE KEY y de END RSA PRIVATE KEY. Si obtienes el siguiente mensaje error “*There was an error decrypting your password. Please ensure*

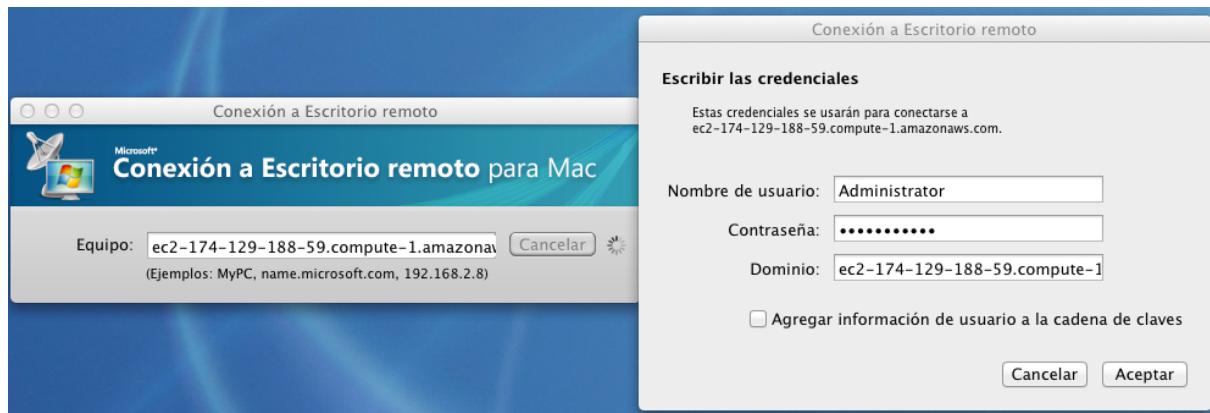
that you have entered your private key correctly” deberás asegurarte que: i) has indicado la clave privada asociada al par de claves con el que has desplegado la instancia previamente y que ii) estás indicando todo el contenido del fichero alucloud\$ID-priv.pem, incluyendo las líneas indicadas anteriormente.

Finalmente, obtendrás tanto la información del usuario (*Administrator*) como la contraseña (en este caso es =gB.g7Zz7ry).

The screenshot shows the AWS Management Console interface. On the left, the navigation pane is open, showing the 'Instances' section with several options like 'Instances', 'Instance Types', and 'Launch Templates'. The main area displays a table of instances with columns for Name, Instance ID, Instance state, and Instance type. One instance, 'alucloud253', is selected and shown in more detail on the right. A modal window titled 'Get Windows password' is open, providing the Windows administrator password for the selected instance. The password is: WXOTISDx&(wukq-Nq\$G+MLS-fnpTSH5G.

Figura 4. Obtención de la contraseña de acceso a una instancia Windows desde la AWS Management Console.

Iniciamos sesión de escritorio remoto con un cliente RDP. Windows suele incluir un cliente RDP, también conocido como Terminal Services, por defecto (prueba a ejecutar el comando *mstsc* para saber si lo tienes instalado). Para OS X es posible utilizar *Microsoft's Remote Desktop Client* [9] y para GNU/Linux tienes disponible herramientas como *rdesktop* o *Remmina*. En nuestro caso, utilizamos el cliente para OS X, especificando tanto el FQDN de la instancia como el usuario y la contraseña de acceso.



Se obtiene un mensaje de advertencia de que el nombre del servidor en el certificado no es correcto, pero elegimos conectarnos igualmente. Una vez realizada la conexión, encontrarás un escritorio como el que se muestra en la Figura 5.

Hay algunos aspectos a tener en cuenta. La instancia dispone de un disco de arranque (unidad C:) de 30 GB formateado como NTFS.

Ten en cuenta que es posible detener la instancia (al estar basada en un volumen EBS) sin que esto provoque la desaparición de los datos almacenados en C:. Una vez vuelta a iniciar la instancia detenida, será posible volver a conectarse mediante el cliente RDP (aunque la IP de la instancia cambia).

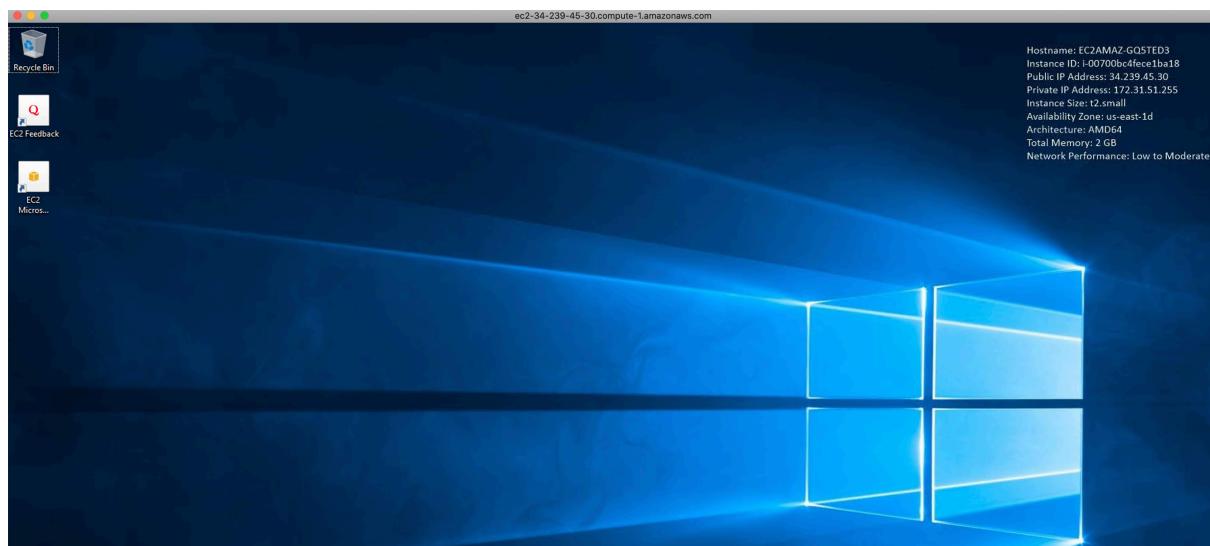


Figura 5. Aspecto del escritorio de una instancia Windows desplegada en Amazon EC2.

Ten en cuenta las siguientes consideraciones:

- Las instancias Windows están limitadas a dos conexiones remotas simultáneas [11], aunque es posible modificar el número de conexiones remotas concurrentes [12].
- Si necesitas crear cuentas de usuario adicionales deberás utilizar las herramientas de administración de Windows para hacerlo.

Una vez finalizada las actividades, procede a terminar la instancia de Windows:

```
:~$ aws ec2 terminate-instances --instance-ids i-0b0ad5816f3f52010

TERMINATINGINSTANCES      i-0b0ad5816f3f52010
CURRENTSTATE              32    shutting-down
PREVIOUSSTATE              16    running
```

Conclusiones

En esta práctica se han utilizado los servicios más relevantes de Amazon Web Services para conseguir el despliegue de infraestructuras virtuales bajo demanda en la forma de instancias. Has podido comprobar la ventaja de desplegar instancias de AMIs basadas en EBS. Has aprendido la creación de grupos de seguridad, pares de clave y la conexión remota a las instancias basadas en GNU/Linux mediante SSH. Esta práctica ha usado fundamentalmente la interfaz de línea de comandos (AWS CLI). Si quieres conocer cómo se interacciona con Amazon EC2 desde la AWS Management Console tienes a tu disposición otra versión de esta misma práctica.

Información Adicional

Esta práctica se realiza en el marco del “Curso Online de Cloud Computing con Amazon Web Services”, ofertado por el Instituto de Instrumentación para Imagen Molecular de la Universitat Politècnica de València. Tienes más información sobre este curso de formación en Cloud Computing en la siguiente dirección: <http://www.grycap.upv.es/cursocloudaws>

Referencias

- [1] Jurg van Vliet, Flavia Paganelli. “Programming Amazon EC2”, O'Reilly, 2011.
- [2] Amazon Web Services (AWS). <http://aws.amazon.com/es/>
- [3] Amazon EC2 API Tools. <http://aws.amazon.com/developertools/351>
- [4] AWS Marketplace. <https://aws.amazon.com/marketplace>
- [5] Amazon Web Services Documentation. <http://aws.amazon.com/es/documentation/>
- [6] Aws: Simple Command-Line Access to Amazon EC2 and Amazon S3. <http://timkay.com/aws/>
- [7] cURL. <http://curl.haxx.se/>
- [8] <http://bitnami.org/stack/tomcatstack#cloudImage>
- [9] Microsoft's Remote Desktop Client. <http://www.microsoft.com/mac/remote-desktop-client>
- [10] rdesktop. <http://www.rdesktop.org>
- [11] Getting Started with Amazon EC2 Windows Instances.
http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/EC2Win_GetStarted.html
- [12] Configure the Number of Simultaneous Remote Connections Allowed for a Connection.
<http://technet.microsoft.com/en-us/library/cc753380.aspx>
- [13] Amazon EC2 pricing. <http://aws.amazon.com/es/ec2/pricing>
- [14] Marvel Universal Social Graph. <http://aws.amazon.com/datasets/5621954952932508>
- [15] Troubleshooting instances. <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-troubleshoot.html>
- [16] LAMP. <http://es.wikipedia.org/wiki/LAMP>
- [17] Bitnami. <http://bitnami.com>
- [18] Interfaz de línea de comandos de AWS. <http://aws.amazon.com/es/cli/>
- [19] Amazon Machine Images (AMI).
<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html>
- [20] Amazon Linux AMI. <http://aws.amazon.com/amazon-linux-ami/>

- [21] Downtime. <http://en.wikipedia.org/wiki/Downtime>
- [22] Amazon EC2 Root Device Volume.
<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/RootDeviceStorage.html>
- [23] ECDSA. <http://es.wikipedia.org/wiki/ECDSA>
- [24] Ataque Man-in-the-Middle. http://es.wikipedia.org/wiki/Ataque_Man-in-the-middle
- [25] Linux AMI Virtualization Types.
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/virtualization_types.html
- [26] They're Here – Longer EC2 Resource IDs Now Available.
<https://aws.amazon.com/es/blogs/aws/theyre-here-longer-ec2-resource-ids-now-available/>
- [27] How do I enable a password login instead of a key pair when logging into my EC2 instance using SSH? <https://aws.amazon.com/es/premiumsupport/knowledge-center/ec2-password-login/>