1. OBJETIVO

En esta práctica utilizaremos la consola de administración de **Amazon Web Services** (**AWS**), una aplicación web con una interface de usuario para acceder a nuestra infraestructura Cloud en Amazon. Cada servicio (EC2, VPC, S3, …) cuenta con su propia consola de gestión, accesible desde la consola general de administración. En la consola de administración también puede consultarse la información sobre nuestra cuenta y el consumo de recursos / facturación.

Nota importante: hay que prestar especial atención a los costes asociados a los recursos que aprovisionamos. Estos servicios se tarifican por horas, y no es conveniente mantenerlos activos durante demasiado tiempo. Se recomienda consultar el Panel de Facturación (Usuario → *Billing Dashboard*) e incluso **crear una alarma de facturación con CloudWatch** (ver el documento https://docs.aws.amazon.com/es_es/AmazonCloudWatch/latest/monitoring/monitor_estimated_charges_with_cloudwatch.html).

Podemos usar la cuenta de formación de **AWS Academy** para administrar recursos AWS de forma limitada, accediendo al curso *AWS Academy Learner Lab - Foundation Services* → *Modules* → *Learner Lab*. En este caso tendremos que trabajar en la región de **Virginia del Norte** (**us-east-1**), y ya contamos con una cuenta del proveedor *voclabs* para el acceso, **una pareja de claves SSH pública/privada** (que podemos descargar siempre que queramos), y credenciales IAM (en la sección ***AWS Details*)**.

2. EJECUCIÓN DE INSTANCIAS EC2

1. Acceder a la consola de administración web y seleccionar una Región AWS para realizar la práctica. Si accedemos con la cuenta de AWS Academy usaremos la región **us-east-1**. Si lo hacemos con nuestra cuenta particular, se recomienda trabajar en una región europea. Vamos a crear en primer lugar una pareja de claves pública/privada. Si accedemos con la cuenta de AWS Academy las claves ya existen (sección *AWS Details*), y sólo tenemos que descargarlas, en formatos .pem y .ppk. En ese caso no es necesario crear nuevas claves, aunque también podemos hacerlo para reproducir el proceso a seguir con una cuenta AWS.

   En la consola de computación EC2 (*Services* → *Compute* → *EC2*) acceder a la sección de claves (*Network & Security* → *Key Pairs)* y crear una nueva pareja de claves RSA en formato .pem (**ClavesPractica**). Guardar la clave privada descargada en un lugar seguro. Si va a utilizarse el cliente Putty para acceder a las instancias será necesario convertir la clave a formato .ppk (botónes *Load* y *Save Private Key* de la utilidad Puttygen; el archivo de claves se guardará en formato .ppk).

   Nota: para usar la clave privada generada (archivo .pem) en un cliente SSH de Linux o Mac tendremos que llevar una copia del archivo al equipo, asignar derechos exclusivos al archivo para el propietario (chmod **400**) y usar la opción **-i** en la llamada al cliente.

2. Acceder a la sección Network & Security → Security Groups. Crear un nuevo grupo de seguridad (**SSH_HTTP_Any**) para permitir el acceso a las instancias (*inbound*) por los puertos 22/tcp y 80/tcp desde cualquier dirección IP. Comprobar que se acepta todo el tráfico en las conexiones salientes (*outbound*).

3. En la consola VPC (*Services* → *Networking & Content Delivery* → *VPC*) comprobar el direccionamiento IPv4 asignado a nuestra VPC y a las subredes definidas (una por cada zona de disponibilidad de la Región). Consultar también la tabla de rutas existente y el nombre del *router* por defecto en la VPC existente.

4. Nuevamente en la consola EC2 lanzar una primera instancia **PruebasWeb** con una **Amazon Linux AMI** y tipo **t2.micro** (forma parte de la capa gratuita), con almacenamiento EBS de 8 GB (el contenido es persistente, aunque podemos eliminarlo automáticamente al finalizar la instancia). Arrancaremos la instancia en una de las redes “públicas” existentes (son las subredes predeterminadas, y existe una por zona de disponibilidad), y tendrá asociada un IP pública (elástica). Usará el grupo de seguridad **SSH_HTTP_Any** y la clave

pública de **ClavesPractica**. Lanzar la instancia. Tras un par de minutos la máquina estará operativa.

5. Identificar las direcciones IP asignadas (privada y pública), así como los nombres de dominio asociados a la instancia. Usando un cliente SSH y la clave privada descargada abrir una conexión a la instancia. La información de conexión está disponible pulsando en el botón superior **Connect** (el usuario será **ec2-user** para la Amazon Linux, aunque este nombre puede cambiar en otras distribuciones). Realizar algunas tareas de mantenimiento y configuración en la máquina virtual: promover a superusuario (sudo su -), comprobar la dirección IP privada asignada y la tabla de encaminamiento, actualizar el SO (yum -y update), instalar el servidor **apache (httpd)** y **configurar su arranque automático**. Instalar también el intérprete **php**, y crear una página de inicio predeterminada (index.php) que nos presentará la dirección IP privada de la máquina virtual. Comprobar que se visualiza la página de inicio con la dirección IP de la instancia accediendo desde un navegador web con la dirección IP pública y el nombre de dominio externo asignado. Instalar también el programa **stress**, que usaremos en el apartado de auto escalado.
6. Comprobar el efecto sobre las direcciones IP pública y privada de un reinicio de la máquina (*shutdown -r now*) y una detención (*shutdown -h now*, equivalente a *Actions → Instance Settings → Stop* desde la consola EC2) y arranque posterior (*Actions → Instance Settings → Start*). Nota: verificar también que el servidor web se inicia automáticamente.
7. Generar una imagen de la instancia con nombre **ServidorWeb** (*Actions → Image → Create Image*) con los valores por defecto. Podemos comprobar que se genera una nueva imagen privada (sección *Images → AMIs*). Por defecto, para generar la imagen AMI la instancia se reinicia. Cuando necesitemos un servidor Web podremos lanzar una instancia de esta imagen. Nota: esta AMI creada la mantendremos hasta el final de la práctica.
8. Cuando concluya la generación de la imagen, terminar la ejecución de la instancia PruebasWeb (*Actions → Instance Settings → Terminate*). Podemos comprobar que también se elimina el volumen EBS de la instancia, pero no la imagen creada (y el *snapshot* asociado).

3. BALANCEADOR DE CARGA. ELB (*Elastic Load Balancer*)

1. En la consola de computación (*Services → Compute → EC2*) lanzar dos instancias **t2.micro** usando la imagen **ServidorWeb** creada en el apartado anterior (ServidorWeb1 y ServidorWeb2). Las instancias se ejecutarán en dos zonas de disponibilidad distintas (a y b), y sólo tendrán asociada una IP privada (no usaremos direcciones elásticas). Crearemos un nuevo grupo de seguridad, (**SSH_HTTP_Internal**), con acceso a los servicios SSH y HTTP, pero sólo desde la red VPC. Por último, inyectaremos nuestra pareja de claves (**ClavesPractica**) en cada instancia, aunque no vamos a poder alcanzar las instancias desde el exterior.
2. Ir a la sección Load Balancing → *Target Groups.* Crear un nuevo grupo de destinatarios (ServidoresWeb) con el protocolo HTTP (puerto 80). Establecer el *check* que nos propone el asistente (una consulta a la raíz del servidor). Registrar en el grupo de destinatarios las dos instancias de nuestro servidor Web (ServidorWeb1 y ServidorWeb2).
3. Ir a la sección Load Balancing → *Load Balancers.* Crear un balanceador de carga de Aplicación HTTP/HTTPS con nombre **Balanceador1**, con esquema expuesto a Intenet (***Internet-facing***), sólo para tráfico IPv4 y HTTP (puerto 80/TCP). El balanceador trabajará sobre las dos primeras zonas de disponibilidad (a y b), en las que se encuentran las instancias. Usaremos el grupo de seguridad **SSH_HTTP_Any** (aunque sólo necesitamos HTTP). Establecer el grupo de destinatarios (ServidoresWeb) con el protocolo HTTP (puerto 80) y fijar el *check* que nos proponen por defecto (una consulta a la página de inicio). Por último, lanzar el balanceador.
4. Lanzar consultas desde el exterior al nombre de dominio asociado al balanceador creado. Debería mostrarse la página de inicio y el balanceo que se está produciendo (mediante las

direcciones IP de la página de inicio). Podemos comprobar el estado de las instancias en la sección de Load Balancing → *Target Groups*.

5. Los servidores web ofrecen sus contenidos a través del balanceador, pero no están accesibles desde el exterior para su administración. Para acceder por SSH a estos servidores podemos usar distintas técnicas. Por ejemplo, adquirir una dirección IP elástica (pública) y a continuación asociarla a una de las instancias internas (igualmente será necesario modificar la política de seguridad para permitir el acceso SSH desde cualquier dirección IP). También es posible levantar una nueva instancia auxiliar con una IP pública y, una vez conectados por SSH a esta nueva instancia, abrir nuevas conexiones SSH desde ella a las máquinas internas (será necesario copiar la clave privada a esa nueva instancia). Otras opciones: podemos usar la instancia auxiliar anterior y crear túneles SSH, o configurar en esa instancia la función de *routing* y reglas de *port-forwarding* hacia las máquinas internas.

   Probar al menos el primer modo: Reservar una IP elástica (Network & Security → *Elastic IPs)* y asociarla a una de las máquinas internas (ServidorWeb1). También hay que cambiar el grupo de seguridad para poder acceder por SSH desde el exterior. Una vez sea posible acceder a la instancia mediante SSH, detener el servicio Apache y comprobar su efecto al lanzar nuevas peticiones al balanceador.

6. Terminar la ejecución de las instancias (ServidorWeb1, ServidorWeb2) y eliminar la dirección IP elástica, el balanceador y el grupo de destinatarios.

4. SUBREDES PÚBLICAS / PRIVADAS. NAT Gateway.

1. Normalmente las instancias privadas se ubican en subredes privadas, aisladas de las subredes "publicas", cuyos recursos pueden estar accesibles desde el exterior mediante direcciones elásticas. Todas las subredes definidas en la VPC, públicas y privadas, tendrán asociado un bloque de direcciones IP libres dentro del rango asociado a la VPC.

   Crear en primer lugar una nueva subred con direccionamiento /24 (**el primero disponible**) en una de las zonas de disponibilidad de la región. Para ello es necesario revisar en primer lugar la asignación de direcciones a las subredes ya definidas, accediendo a la sección *Subnets* del panel de control de VPC. A continuación, dentro de esta sección, añadir la nueva subred en una zona de disponibilidad, con nombre **Red Interna** y el bloque /24 seleccionado previamente.

2. En la consola de administración de EC2 lanzar dos nuevas instancias Amazon Linux AMI de tipo t2.micro (**ServidorPublico** y **ServidorPrivado**), la primera ubicada en una de las subredes públicas y con dirección IP elástica asociada, y la segunda en la nueva subred y sin dirección elástica (sólo tendrá IP privada). En el grupo de seguridad permitir el acceso a los servicios SSH y HTTP. Acceder a la instancia pública por SSH y conectar desde ella a la instancia privada (es necesario contar con la clave privada ClavesPractica.pem en el cliente ssh). Comprobar que no es posible acceder al exterior desde la red privada.

3. Configurar un ***NAT Gateway*** en la subred pública de la misma zona de disponibilidad donde se encuentra la subred privada (es necesario configurar este servicio en cada Zona de Disponibilidad donde existan subredes privadas). Para ello, en la consola de VPC (*Services → Compute → VPC*) vamos a la sección *NAT Gateways* y creamos uno nuevo en la subred pública de la zona de disponibilidad deseada. Se le asociará una IP pública.

4. Crear una nueva tabla de rutas para encaminar el tráfico saliente (0.0.0.0/0) hacia el nuevo NAT Gateway. Contaremos con dos tablas de encaminamiento: una para las redes "publicas" (salida a través del igw), y otra para la red privada creada (a través del nuevo *NAT Gateway*).

5. Asignar la nueva tabla de encaminamiento a la subred privada (sección *Subnets* de VPC). A continuación, acceder a la instancia lanzada en la subred privada y comprobar que es posible alcanzar recursos externos. ¿Cómo podemos conocer desde la instancia qué

dirección IP pública se está usando para las conexiones exteriores? Comprobar que dicha dirección coincide con la IP elástica asociada al NAT Gateway.

6. Es posible limitar el tráfico que entra o sale de las subredes mediante **Network ACLs**. Verificar en primer lugar la estructura de la ACL predeterminada asociada a las subredes, y sus reglas *Inbound* y *Outbound*. ¿Se limita de alguna manera el tráfico de red con esta ACL? A continuación, crear una ACL de prueba para la red privada que impida el acceso por HTTP desde la red "publica" donde se encuentra el ServidorPublico, pero que sí permita el acceso por SSH desde esa red (será el único servicio permitido). A continuación, activar la ACL en la red privada y verificar su funcionamiento. Indicar las reglas usadas.
7. Terminar la ejecución de las instancias usadas, el NAT Gateway, la subred privada, la Network ACL y la tabla de encaminamiento. Indicar **el orden seguido en la eliminación** de estos recursos Nota: hay que tener en cuenta las dependencias. Por ejemplo, para eliminar una tabla de encaminamiento no puede estar asignada a ninguna subred.

5. ALMACENAMIENTO. S3

1. Acceder a la consola de administración de S3 (*Storage → S3*) y crear un nuevo *bucket*. Utilizar un nombre significativo (por ejemplo, ***mi_nombre_y_apellido***) y ubicarlo en la región de trabajo. Usar las opciones por defecto (ACLs disabled, bloquear todo el acceso público, encriptado, …) salvo **el control de versiones**, que lo activaremos. Identificar la clase de almacenamiento S3 usada en este nuevo *bucket* (*Storage class*).

   Nota: Los nombres de bucket **son únicos en una región AWS**, y siguen el convenio del DNS (caracteres alfanuméricos, puntos y guiones, y sólo minúsculas).
2. Subir algún archivo al nuevo *bucket* creado (botón *Upload*). Crear alguna carpeta para organizar los archivos (es posible usar *Cut / Paste* para mover los archivos entre las carpetas). Estudiar las propiedades de los archivos subidos al *bucket* (pulsando sobre la casilla de selección del archivo). ¿Qué ocurre si intentamos descargar el archivo pulsando sobre su URL?
3. Vamos a usar el *bucket* para alojar contenidos web estáticos de acceso público a través de un URL. Para ello seleccionamos el *bucket* y editamos su configuración. Habrá que deshabilitar la configuración de bloqueo de acceso público y escribir una política de *bucket*. En Permisos, desactivar la opción **Bloquear acceso público**. En Propiedades, editar la sección *Static website hosting*, y activar esta opción, indicando en las propiedades las páginas index.html y error.html propuestas.
4. Añadir una página de inicio (index.html) en la raíz del *bucket* con un texto significativo. Si queremos acceder vía web a todos los contenidos del *bucket* será necesario establecer una política de acceso adecuada (en la misma pestaña de Permisos, añadir una *Bucket policy*). Podría ser algo similar a esto (sustituir el nombre del *bucket* por el que corresponda):

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "PublicReadGetObject",
         "Effect": "Allow",
         "Principal": "*",
         "Action": "s3:GetObject",
         "Resource": "arn:aws:s3:::nombre-del-bucket/*"
      }
   ]
}
```

   Definir y activar dicha política. A continuación, acceder a uno de los contenidos del *bucket* a través de su URL.

5. Desde el panel de control de S3 eliminar el *bucket* de pruebas.

6. AUTOESCALADO PARA APLICACIONES WEB

Vamos a definir un grupo de auto escalado horizontal para un servicio web. Estará formado por un balanceador ELB y entre 2 y 5 instancias EC2 que ejecutarán la imagen ServidorWeb creada en el apartado 2 y que tiene instalado el servidor Apache. Las instancias se irán levantando o deteniendo en función de la demanda del servicio. La configuración del auto escalado requiere definir el balanceador al que se asociarán las instancias, la imagen AMI y el tipo de máquina que usaremos (en la sección *Launch Templates*) y las características del escalado (en la sección *Auto Scaling Group*): número inicial y máximo de instancias, zona(s) en las que se lanzarán y condiciones de arranque/detención.

1. Creamos un grupo de destinatarios (**ServicioWeb-tg**) de tipo Instancia, con protocolo y *check* HTTP, e inicialmente sin destinatarios. A continuación, creamos un balanceador de carga de aplicación (**ServicioWeb-elb)**, de tipo *Internet-facing HTTP* para IPv4. Trabajará en las zonas a y b, donde se lanzarán las instancias, con el grupo de seguridad **SSH_HTTP_Any** y el grupo de destinatarios creado (ServicioWeb-tg).
2. Creamos una Plantilla de Lanzamiento (**ServicioWeb-lt**). Nota: Anteriormente se usaban Configuraciones de Lanzamiento, pero se recomienda migrar a las nuevas *Plantillas de Lanzamiento*. Seleccionamos ***Launch Templates*** (en la sección ***Instances***) y a continuación ***Create Launch Instance***. Damos el nombre a la plantilla (ServicioWeb-lt). Usaremos nuestra AMI (**ServidorWeb**) con tipo de instancia **t2.micro**, y nuestra pareja de claves SSH. **No** incluimos la subred en la plantilla, y sí el grupo de seguridad (**SSH_HTTP_Any**: usamos este para poder conectarnos a las instancias también desde el exterior por SSH, aunque realmente no sería necesario, sería suficiente con el servicio HTTP accesible sólo desde la VPC**)**. Salvamos la plantilla. En este punto podemos seleccionar la creación del grupo de autoescalado, o no activar la opción y hacerlo a continuación en un nuevo paso.
3. Creamos un nuevo grupo de *Auto Scaling* **(ServicioWeb-as)**. Seleccionamos la plantilla de lanzamiento creada previamente (ServicioWeb-lt). En la sección de Network, establecemos la VPC y las subredes donde se crearán todas las instancias (subredes públicas de las zonas a y b). A continuación, asociamos el balanceador de carga ya definido, a través del *Target Group*. En el chequeo dejamos seleccionado el tipo de comprobación de estado EC2, y el periodo de comprobación de gracia de 300 segundos (no hace falta activar la monitorización por parte del balanceador). Activamos la monitorización de *CloudWatch*. En cuanto al tamaño del grupo, establecemos inicialmente las capacidades **deseada, mínima y máxima en 2**, y **NO activamos** la política de escalado. No añadimos notificaciones ni etiquetas. Aparecerá un resumen de configuración. **Creamos** el grupo de autoescalado y se inicia.
4. Comprobamos que se lanzan automáticamente las dos primeras instancias (las únicas que arrancarán). También podemos comprobar que las instancias se han registrado en el balanceador y que se reparte el tráfico web entre las mismas (accedemos al servicio a través del nombre de DNS del Balanceador ServicioWeb-elb). Todavía no hay escalado y sólo se lanzarán nuevas instancias si dejan de funcionar las actuales (podemos comprobarlo si terminamos alguna de ellas o detenemos su servicio HTTP; la pestaña de *Activity* nos informa del comportamiento del grupo de AutoEscalado).
5. Vamos a modificar el grupo de auto escalado creado (ServicioWeb-as). Editamos los *Group Details* y mantenemos la capacidad mínima y deseada a 2 instancias, y **aumentamos a 5 su capacidad máxima**. Salvamos con *Update*. En la pestaña ***Automatic Scaling*** creamos una **política dinámica de escalado**. Seleccionamos su tipo en *Target tracking scaling*, le damos el nombre **PoliticaEscalado**, métrica de CPU y umbral de 50 (se activará cuando el consumo total de todas las instancias pase del 50%).

6. Podemos comprobar el funcionamiento del escalado abriendo una sesión SSH en las dos instancias iniciales y forzando un uso intensivo de la CPU con el programa **stress**, instalado previamente (ej: stress --cpu 1 --timeout 60m). Al aumentar el consumo de CPU el número de instancias debería aumentar a 4 o 5 (hay que esperar unos minutos: consultar el historial de actividad). Posteriormente, si desactivamos el programa de stress el número de instancias debería volver a bajar a 2 (paciencia, tarda bastante…).
7. Una vez completado el apartado eliminar el Grupo de auto escalado creado (se terminarán todas las instancias del grupo, aunque el proceso es lento). Eliminar también el balanceador, el grupo de destinatarios, y la plantilla de lanzamiento.

7. OPCIONAL. CLOUDFORMATION. INFRAESTRUCTURA COMO CÓDIGO

1. Vamos a construir una sencilla infraestructura CMS (*Content Management System*) basada en WordPress. Usaremos una plantilla predefinida de CloudFormation: Wordpress Blog. Seleccionamos el servicio **CloudFormation**, que se encuentra dentro de la sección *Management & Governance*. Pulsamos sobre *Create Stack*. Seleccionamos la opción *Use a sample template* y la plantilla "*Wordpress Blog.* (ojo: la que está dentro del grupo **Simple**)

   Nota: si pulsamos en el botón *"View in Designer"* podemos ver la estructura de la plantilla. También es posible utilizar un editor gráfico de las plantillas para facilitar su creación/modificación.
2. Pulsamos en *Next*. Especificamos los datos del configuración: Nombre de la pila (**MiWordPress**), nombre de la base de datos, contraseña del administrador y el usuario de la base de datos, tipo de las instancias (usar **t2.micro**), clave SSH, dirección de acceso. Asignamos al resto de parámetros sus valores por defecto. OJO: las claves tienen que ser robustas.

   Nota: las plantillas de la sección Multi-AZ generan una arquitectura más compleja, con balanceador, múltiples instancias, servicio RDS, … Es un proceso más lento y consume muchos más recursos. Por ello no lo usaremos en esta práctica, aunque se recomienda seleccionar una plantilla Multi-AZ y visualizarla en el editor gráfico (*View in Designer*).
3. Lanzamos la ejecución de la nueva pila (*Create Stack*) y visualizamos los eventos generados (pestaña *Events*). Después de un par de minutos el proceso habrá concluido.
4. Usar el URL proporcionado para conectar con el CMS y terminar de configurarlo (nombre del sitio, contraseña del administrador). Acceder como administrador para editar algún contenido. Verificar los cambios con un acceso como usuario no registrado.
5. Eliminar todos los recursos activados. Es posible hacerlo desde la consola de CloudFormation seleccionando el *stack* previamente creado.

Nota: los aspectos significativos de la configuración realizada en todos los apartados de las secciones 2 a 6 tendrán que reflejarse en la memoria de la práctica. Opcionalmente se incluirá todo lo relativo al apartado 7.

8. REFERENCIAS

- https://aws.amazon.com
- https://aws.amazon.com/es/products/
- https://aws.amazon.com/es/pricing/

- Amazon Web Services in Action.
  Michael and Andreas Wittig. Manning Pub, 2015
- AWS For Admins for dummies.
  John Paul Mueller. John Wiley & Sons, Inc. 2017