1. OBJETIVO

En esta práctica utilizaremos la interfaz de línea de comandos de AWS (AWS CLI) para administrar distintos servicios de AWS (EC2 y VPC). También se aborda la interacción con AWS a través de *scripts* desarrollados en Python (SDK Boto3).

Nota importante: hay que prestar especial atención a los costes asociados a los recursos que aprovisionamos. Como sabemos, estos servicios se tarifican por horas, y no es conveniente mantenerlos activos durante demasiado tiempo.

2. OPCIONAL. AMAZON IAM (*Identity and Access Management*).

Nota: Este apartado sólo puede realizarse si contamos con una cuenta personal de AWS.

a. Utilizando la consola de administración de AWS y el servicio IAM, dar de alta un nuevo usuario (**UsuarioPruebas**) para acceso programático. Guardar sus credenciales (*access key ID* y *secret access key*) para su uso posterior con AWS CLI. El usuario no tendrá permisos asociados directamente.

b. Crear un grupo de usuarios (**Administradores**), al que pertenecerá el usuario anterior. El nuevo grupo tendrá todos los permisos sobre los servicios EC2 y VPC (políticas "FullAccess").

3. INSTALACIÓN DEL CLIENTE AWS CLI.

a. Instalar la última versión de la interfaz de línea de comandos (CLI) de AWS en un sistema Linux.

b. Asociar las credenciales de un usuario programático a la configuración de AWS CLI. Establecer la región AWS de trabajo como predeterminada y **json** como formato de salida (archivos credentials y config).

c. Verificar que se tiene acceso a la infraestructura AWS con el cliente configurado (por ejemplo, listar las zonas de disponibilidad de la región configurada.

   Nota: En caso de detectar problemas de autentificación, comprobar en primer lugar que la fecha y hora del equipo cliente están sincronizadas (para la sincronización puede usar el protocolo *ntp*) o el comando date -s "aaaa-mm-dd hh:mm".

Nota: Si se utiliza una cuenta de AWS Academy, es necesario añadir el **aws_session_token** al archivo **credentials** (las credenciales pueden obtenerse en *AWS Details*).

4. AMAZON EC2 (*Elastic Compute Cloud*)

Realizar este apartado usando exclusivamente la interface **AWS CLI** (no la consola Web).

a. Crear un nuevo grupo de seguridad (**ssh-http-sg**) que asociaremos a nuestras instancias EC2. Permitirá las conexiones entrantes por los puertos TCP 22 y 80. El grupo permitirá cualquier conexión saliente (es la configuración predeterminada).

b. Crear una pareja de claves pública / privada con nombre **CEU-Keys** y guardar la clave privada (en formato .pem) en el directorio .ssh de nuestro usuario Linux. Cambiar los permisos de este archivo para que pueda ser utilizado por el cliente ssh.

c. Obtener el ID (ami-xxxxxx) de la última imagen disponible para **x86_64** cuya descripción comienza con la cadena **Amazon Linux 2023 AMI.** Lanzar una primera instancia EC2 usando el ID de esta imagen. Será del tipo **t2.micro**, usará el grupo de seguridad y las claves obtenidas en los apartados anteriores, y se ubicará en la subred por defecto **de la primera zona de disponibilidad (1a)**.

   Nota: Indicar el proceso seguido con la consola CLI para obtener el ID de la imagen, y verificar que ese ID coincide con el que nos proponen como primera opción a través de la consola web. Si no se obtiene el ID esperado con AWS CLI, usar el propuesto por la consola web.

d. Verificar que se cuenta con una nueva instancia iniciada y obtener su dirección IP pública (elástica) asociada. A continuación, abrir una sesión SSH usando la clave privada y el usuario predeterminado de definido en la AMI (ec2-user en esta imagen).

e. Detener la instancia EC2, verificar su nuevo estado y a continuación volver a arrancarla.

5. AMAZON VPC (*Virtual Private Cloud*)

Realizar este apartado usando exclusivamente la interface **AWS CLI** (no la consola Web).

a. Identificar las características de la red VPC de nuestra cuenta y de las subredes presentes en las zonas de disponibilidad de la región en la que nos encontramos (identificador de red y subredes, y bloque de direcciones IP asociadas).

b. Crear una nueva subred en la **tercera zona de disponibilidad (1c)** de nuestra región. Tendrá asociado un bloque de direcciones /24 válido (dentro del espacio de direcciones de la VPC, y aún no asignado). Visualizar las propiedades de la subred creada ¿Qué valor tiene la propiedad *map-public-ip-on-launch*? ¿Qué implica ese valor?

c. Lanzar una nueva instancia con las mismas características de la anterior pero ahora ubicada en la nueva subred. La instancia no tendrá dirección IP pública (elástica) asociada.

d. Obtener la dirección IP privada de la nueva instancia creada. Comprobar que es posible abrir una conexión SSH a la nueva instancia desde la anterior (será necesario contar con la clave SSH privada en la primera instancia).

e. La nueva instancia creada no tiene salida a Internet. Para facilitar la salida de esta máquina reservar en primer lugar una IP pública (elástica) y a continuación crear un **nat-gateway** que tendrá asociada la IP pública reservada y estará ubicado en la **subred pública de la tercera zona de disponibilidad**. Crear una tabla de rutas para la subred privada y una nueva ruta que usará el nat-gateway para las conexiones externas desde la subred privada. Verificar que la nueva instancia tiene conectividad exterior.

6. Boto3. AWS SDK PARA PYTHON

En este apartado se crearán pequeños *scripts* en Python que harán uso de la librería boto3. En algunos casos, en la llamada al *script* se indicarán los parámetros requeridos para la ejecución.

a. Instalar el SDK Boto3 para AWS. Utilizar las credenciales de usuario creadas al comienzo de la práctica. Verificar su funcionamiento con el intérprete de Python.

b. Crear un script en Python para listar todas las instancias EC2 no terminadas presentes en la región predeterminada. Devolverá su ID, estado (solo el “Name”), ID de la subred donde se encuentra, dirección IP privada y, en caso de tenerla, dirección IP pública (elástica).

c. Crear un script en Python para listar todas las subredes presentes en las VPC existentes en la región. Para cada subred devolverá su ID, el ID de la VPC donde se encuentra, el bloque de direcciones IP asociado a la subred, y si está activada la asignación automática de dirección IP pública (yes|no).

d. Crear un script en Python para lanzar una nueva instancia EC2 en la región predeterminada. Recibirá como parámetros el ID de la imagen (AMI), el tipo de instancia, el ID de la subred a la que se conectará, si tendrá asociada una IP pública o elástica (parámetro yes|no), id del grupo de seguridad, y nombre de la clave pública a inyectar. Devolverá el ID de la nueva instancia y su estado.

e. Crear un script en Python que nos indique el número de volúmenes, instantáneas, grupos de seguridad, balanceadores de carga y parejas de claves asignados a nuestra cuenta en la región en la que nos encontramos.

f. Crear un script en Python para eliminar (*Terminate*) la instancia o instancias cuyos ID se pasan como parámetro.

g. Crear un script en Python para arrancar todas las instancias de la región que se encuentren detenidas (*Stopped*) y tengan asociada una etiqueta (*Tag*) que se pasará como parámetro.

h. **OPCIONAL**. Crear un script Python para recolectar información de los recursos EC2 y VPC asociados a la cuenta **en cualquier región**. Concretamente, en cada región al menos deberán indicarse los siguientes parámetros:

   a. En EC2, número de instancias en ejecución, volúmenes, parejas de claves, instantáneas y grupos de seguridad.

   b. En VPC, número de subredes, direcciones IP elásticas y NAT gateways.

   Importante: para este apartado, necesitará una cuenta programática AWS con acceso a todas las regiones.

Una vez completados los apartados termine la ejecución de las instancias EC2 lanzadas y elimine también los componentes adicionales utilizados (subred privada, direcciones IP elásticas, tabla de rutas, nat-gateway, imágenes, instantáneas, …)

Nota: los aspectos significativos de la configuración realizada tendrán que reflejarse en la memoria de la práctica.

7. REFERENCIAS

- Interfaz de línea de comandos de AWS
  https://aws.amazon.com/es/cli/
- Documentación de la interfaz de línea de comandos de AWS
  https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-welcome.html
- Referencia de la CLI
  https://awscli.amazonaws.com/v2/documentation/api/latest/reference/index.html
- Boto3. Ejemplos de código
  https://boto3.amazonaws.com/v1/documentation/api/latest/guide/examples.html
- Boto3. Documentación
  https://boto3.amazonaws.com/v1/documentation/api/latest/index.html
- Uso de Boto3. *10 Boto3 scripts to simplify your AWS journey*
  https://devopslearning.medium.com/10-boto3-scripts-to-simplify-your-aws-journey-ad6ce6f99852
- Introducción a Boto3. 101 DaysofDevops
  https://www.101daysofdevops.com/courses/101-days-of-devops/lessons/day-15-introduction-to-boto3/