Assignment 3 7 de Febrero del 2023

Daniel Agraz Vallejo Cloud Computing

1. Create a CloudFront distribution for your website and explain each step with great technical detail. **50 points.**

Para acceder a los comandos que ofrece AWS CLI sobre el servicio de Cloudfront es importante poner primero **aws cloudfront,** para hacer referencia al recurso de cloudfront y así tener acceso sobre comandos.

El comando **create-distribution** del AWS CLI, te permite crear una distribución de CloudFront para tu página, tal que:

a)

```
aws cloudfront create-distribution
--distribution-config file:///path/to/config.json
```

O también se puede emplear:

b)

```
aws cloudfront create-distribution
--origin-domain-name agraz.cetystijuana.com.s3-website.us-
east-1.amazonaws.com
--default-root-object index.html
```

Para ejecutar este comando se tienen dos opciones **a)** y **b).** Con **a)** solo se requiere **distribution-config** que pide la ruta del archivo de configuración de la distribución en formato JSON. La estructura del archivo de configuración es el siguiente:


```
{
    "CallerReference": "cli-1676081849-223482",
    "Aliases": {
        "Quantity": 1,
        "Items": [
            "agraz.cetystijuana.com"
        ]
    },
    "DefaultRootObject": "index.html",
    "Origins": {
        "Quantity": 1,
        "Items": [
            {
                "Id": "agraz.cetystijuana.com.s3-website-us-east-1.amazonaws.com-1676081849-696824",
                "DomainName": "agraz.cetystijuana.com.s3-website.us-east-1.amazonaws.com",
                "OriginPath": "",
                "CustomHeaders": {
                    "Quantity": 0
                },
                "CustomOriginConfig": {
                    "HTTPPort": 80,
                    "HTTPSPort": 443,
                    "OriginProtocolPolicy": "http-only",
                    "OriginSslProtocols": {
                        "Quantity": 3,
                        "Items": [
                            "TLSv1",
                            "TLSv1.1",
                            "TLSv1.2"
                        ]
                    },
```


A continuación se explicará cada uno de las llaves del archivo de configuración, por cuestiones de brevedad solo se mencionaran las llaves más relevantes.

- **CallerReference:** Es un valor único que asegura que la solicitud no puede ser repetida. Es un String al que le puedes asignar un nombre para identificar tu distribución.

- **Aliases:** es la estructura que contiene información sobre el CNAME, que es el registro al que va apuntar al dominio donde se encuentra nuestro bucket. Dentro de

Alisases, se tiene que especificar **Quantity**, que es el número de aliases de registros que quieres asociar con la distribución, y **Items**, que es directamente el nombre del dominio al que quieres que tu distribución haga referencia.

- **DefaultRootObject:** Es el objeto que quieres que Cloudfront solicite desde su origen, cuando un usuario entra a tu página bajo el alias de tu dominio.

- **Origins:** Contiene aquella información sobre los orígenes de la distribución. Un **Origin** es la locación donde tu contenido es guardado y de donde cloudfront obtiene el contenido para hacerlo visible al usuario, aquí igual es necesario especificar **Quantity**, que indica el número de orígenes que tendrá la distribución y **Items**, que es una lista que contiene:
    - **Id:** es un string que funge como identificador único para un origen.
    - **DomainName:** Es el nombre del dominio al que apunta la distribución, en mi caso es: *agraz.cetystijuana.com.s3-website.us-east-1.amazonaws.*

Como nuestro Bucket S3 fue configurado para hostear un sitio web estático, también se tiene que es pacificar el **CustomOriginConfig**, que contiene:

• **HTTPPort:** es el puerto HTTP que Cloudfront usara para conectarse desde el origen establecido.

• **HTTPSPort:** es el puerto HTTPS que cloudfront usa para conectarse al origen agregando la “S” de Secure a las transacciones de la página.

• **OriginProtocolPolicy:** especifica el protocolo sea HTTP o HTTPS que cloudfront empleara para conectase al origen. En mi archivo de configuración este apartado lo tengo como: ‘*http-only’* , que indica que Cloudfront siempre usara el HTTP para conectarse a la locación donde los archivos de la página

• **OriginSslProtocols:** pide que selecciones el versión mínimo del protocolo SSL/TLS que cloudfront use para conectarse al origen con HTTPS. Aquí se tiene especificado 3 versiones: TLSv1, TLSv1.1 y TLS1.2.

• **AllowedMethods:** es la estructura dentro del JSON que controla que métodos HTTP va a procesar Cloudfront y reenvía al bucket S3. Por predeterminado cloudfront indica que solo reenvía información a solicitudes **GET** y **HEAD**.

Aprovechando el espacio de esta pregunta quiero incluir, de que manera se puede tener un certificado y configurarlo para la distribución de Cloudfront, con el servicio ACM de aws.

Ejecutando el comando **aws acm request-certificate**, te permitirá obtener un ARN de **acm** que contiene el identificador del certificado:

```
aws acm request-certificate
--domain-name  *agraz.cetystijuana.com
--validation-method DNS
```

Este comando en cuestión, requiere de dos parámetros:

- **--domain-name:** es el FQDN (Fully Qualified Domain Name) o dominio de la página que quieres asegurar con el certificado ACM. En el ejemplo, se antepone un asterisco (*) al nombre del dominio, esto se debe a que se quiere generar un certificado Wildcard, el cual sirve para proteger varios sitios sobre el mismo dominio.

- **--validation-method:** te pide seleccionar el método de validación que se va a utilizar para solicitar un certificado público, para validar que eres dueño del dominio.

Una vez obtenido el ARN que te arroja el comando, se tiene que poner en el archivo de configuración de la distribución de Cloudfront, de la siguiente manera:

```
"ViewerCertificate": {
    "CloudFrontDefaultCertificate": false,
    "ACMCertificateArn": "arn:aws:acm:us-east-1:292274580527:certificate/e961006c-fa16-48ce-aeae-1f093f83db26",
    "Certificate": "arn:aws:acm:us-east-1:292274580527:certificate/e961006c-fa16-48ce-aeae-1f093f83db26",
    "SSLSupportMethod": "vip",
    "MinimumProtocolVersion": "TLSv1",
    "CertificateSource": "acm"
},
```

- **ViewerCertificate:** es la estructura que determina la configuración SSL/TLS de la distribución cuando un usuario solicita los recursos de la página.
    - **CloudFrontDefaultCertificate:** este es un booleano que define si actualmente la distribución emplea el certificado predeterminado que viene al crear la distribución de Cloudfront o si usa uno externo. Como se obtuvo un certificado externos por medio de **acm** esta llave esta en *false.*

    - **ACMCertificateArn:** en esta llave es donde se pone el Arn del certificado acm que solicitamos, este proceso es el mismo para la llave de **Certificate.**

    - **SSLSupportMethod:** Como nuestra distribución utiliza un alias, se tiene que especificar que conexiones HTTPS acepta la distribución. En el ejemplo esta como, **vip**, que indica que la distribución acepta conexiones HTTPS de todos los usuarios incluyendo a los que no soportan SNI.

    - **MinimumProtocolVersion:** De la misma manera, como la distribución emplea Aliases se tiene especificar la política de seguridad que Cloudfront utilice para conexiones HTTPS.

    - **CertificateSource:** Es campo solo indica la provenienza del certificado, en este caso, el certificado wildcard que se obtuvo es de **acm**.

Una vez llenado los campos del certificado, se tiene que actualizar la distribución con el nuevo archivo de configuración, eso se puede lograr con el comando: **aws cloudfront update-distribution.** Donde tienes que especificar los siguientes argumentos:

- **--id:** es el identificador de la distribución de Cloudfront que se busca actualizar.
- **--distribution-config:** es la ruta donde se encuentra el archivo de configuración en JSON.

- **--if-match:** es el **Etag** que maneja el control de versiones de la distribución. Este argumento indica que cuando se busca la distribución por ese **Etag**, omita la configuración de esa distribución, para introducir los nuevos cambios del archivo de configuración de la distribución.

Un ejemplo de este comando es el siguiente:

```
aws cloudfront update-distribution
--id E2M06YB83ZALM1
--distribution-config file:///Users/kekaz16/Documents/Semestre6/
CloudComputing/UploadUpdate.json
--if-match E1O17JSJWOQO06
```

Para la variante **b)**, no es necesario especificar en primera instancia el archivo de configuración de la distribución, con solo dar el **origin-domain-name**, que es el nombre del origen al que la distribución de Cloudfront va a apuntar y va a hacer la solicitud de archivos cuando un usuario entre a la página, y finalmente necesita el **default-root-object**, que como explicamos en el archivo de configuración es el objeto que cloudfront va a cargar primero entrando a la página, en este caso, es un archivo de html.

Esta segunda alternativa de creación de Cloudfront es posible ya que, los parámetros del comando **--default-root-object** y **--distribution-config** son argumentos mutuamente excluyentes. Sin embargo, al terminar la ejecución de este método, se obtiene de output el archivo de configuración con respecto a la distribución creada.

2. Update your DNS Record to route to the CloudFront end point and explain each step with great technical detail. **10 points.**

Para actualizar el registro del DNS del Bucket S3 al dominio de Cloudfront de la distribución que se creo, se tiene hacer uso del servicio de aws **route53**, con el que se tiene que ejecutar el siguiente comando:

```
aws route53 change-resource-record-sets
--hosted-zone-id Z03346142C3RKH191036Y
--change-batch file:///Users/kekaz16/Documents/Semestre6/
CloudComputing/subdomainLink.json
```

change-resource-record-sets, te permite crear, actualizar o borrar el registro que contiene la información del DNS que dirige el tráfico de la página a un subdominio. Este comando toma dos argumentos:

- **--hosted-zone-id:** indica el identificador de la zona donde quieres hacer la operación.
- **--change-batch:** es la ruta del archivo que contiene la estructura JSON con las especificaciones del registro.

Este último parámetro requiere de un archivo JSON con el siguiente modelo:

- **Action:** Hace referencia a la acción que se va a realizar con el recurso especificado, en este caso, se utiliza **UPSERT**, para indicar que se va actualizar un recurso de Route53 que ya existe.

- **ResourceRecordSet:** Es la estructura que contiene los datos a detalles de los cambios que se le van hacer al recurso de Route53. Dentro de esta estructura, las llaves que incluye son:
  - **Name:** Es el nombre del subdominio con el que quieres que route53 se refiera a tu página**.**
  - **Type:** Especifica el tipo de DNS que busca que maneje el registro. **CNAME,** hace referencia al registro que redirige a un dominio en especifico.
  - **TTL (Time To Live):** Son los segundos que permites que los datos de tu origen se mantengan almacenados en otras “edge locations”.

  - **ResourceRecords:** Es una lista que contiene la información de los registros de recursos sobre los que se tomara acción.
    - **Value:** Es el valor nuevo del registro del DNS. Esta es la llave clave con la que se logra que al acceder a la página por el alias “*agraz.cetystijuana.com*” te redirige al dominio del cloudfront por medio del DNS, que este a su vez, apunta a las instancia S3 que almacena el sitio web. Por ello, esta llave esta con el valor del dominio de Cloudfront que arrojo la distribución: *d1lyw144nm3ues.cloudfront.net*

Con esto último, se logra cambiar el DNS que previamente se había configurado con el origen de la instancia S3 al dominio de la distribución de Cloudfront que se creo.

3. Explain what it means to minify website resources (HTML, JS, CSS) and the advantages and disadvantages of this process. **10 points.**

Minify o Minification se refiere al proceso de remover información innecesaria y redundante tanto en archivos de Markup como de Código esto incluye comentarios, espacios en blanco o código no utilizado. También busca implementar técnicas como tener nombres de funciones y variables más cortos, compactar bloques de código para no generar tantas líneas. El propósito de Minification es reducir el tamaño de los

archivos HTML, CSS y JS, ya que esto no afecta ni la funcionalidad ni la forma en la que los recursos son procesados por el Browser.

Este proceso tiene las siguientes ventajas:

- Reduce los tiempos de carga de la página
- Mejora el desempeño del código, al compactar su tamaño.
- Tiene la capacidad de procesar varias solicitudes de la red.

Así como sus desventajas:

- Incrementa la complejidad de depurar el código
- Se vuelve más difícil de entender y mantener el código
- Puede causar problemas de compatibilidad por dependencias, plugins y variables del servidor.

4. Write a python script (and explain each step with great technical detail) to:

a. Create or Update a record in the Students DynamoDB table based on the id.

b. Delete a record in the Students DynamoDB table based on the id.

c. Find a record in the Students DynamoDB table by id.

d. **20 points**

Para usar **Boto3,** se tiene que importar e indicar que servicio se va usar.

```
import boto3
```

Para empezar a utilizar los recursos de AWS, con la función **boto3.resource( )** puedes indicar el nombre del servicio que vas a usar. En este caso, será '*dynamodb'*

```
dynamodb = boto3.resource('dynamodb')
```

Para seleccionar la tabla a la que se van a realizar las operaciones **CRUD,** se tiene que almacenar en una variable (*table*) y con el recurso **dynamodb,** se puede acceder a las tablas del esquema con el "dot notation" de **dynamodb.Table( )**, siendo que dentro de los paréntesis se tiene que poner el nombre concreto de la tabla que se quiere manipular.

```
table = dynamodb.Table('Students')
```

a. Para actualizar un registro en Dynamo utilice el método **update_item ( )**, que permite cambiar los atributos de un registro. Dentro de los parámetros del método se pueden encontrar:

- **Key :** La Llave primaria del registro que se quiere actualizar
- **UpdateExpression :** Indica la operación que se va a realizar en los atributos a actualizar, los nuevos valores que van a tener y define el número de atributos que serán actualizados.
- **ExpressionAttributeValues :** Será el diccionario que contiene los valores con los que se va actualizar el registro.

```
import boto3
dynamodb = boto3.resource('dynamodb')
table = dynamodb.Table('Students')

table.update_item(
    Key={'id' : '33293'},
    UpdateExpression = 'SET full_name = :f, personal_website = :p',
    ExpressionAttributeValues = {
        ':f' : 'D4N13L 4GR4Z V4LL3J0',
        ':p' : 'DAgraz.cetystijuana.com'
    }
)
```

```
item = response['Item']
print(item)
```

```
{'full_name': 'Daniel Agraz Vallejo', 'id': '33293', 'personal_website': 'agraz.cetystijuana.com'}
```

Antes el registro estaba así:

```
ents/Semestre6/CloudComputing/Q4-DynamoDBwPython.py
{'full_name': 'D4N13L 4GR4Z V4LL3J0', 'id': '33293', 'personal_website': 'DAgraz.cetystijuana.com'}
kekaz16@daniels-air CloudComputing %
```

Después de ejecutar **update_item( ):**

b. Ahora para remover un registro de una tabla con DynamoDB, con el método

```
table.delete_item(
    Key={
        'id' : '33293'
    }
)
```

delete_item( ), tienes que indicar la llave primaria del registro que se busca remover.

c. Para extraer un registro de una tabla se puede utilizar el método **get_item( )**, siendo

```
response = table.get_item(
    Key={
        'id': '33293'
    }
)
```

que solo se necesita especificar la llave primaria del registro que se quiere obtener.

Para imprimir el item obtenido por la función, almacena la función en una variable, en este caso, ***response*** , luego para acceder al registro se tiene que indicar la índice donde se encuentra los datos del registro y finalmente con un **print( )** puedes desplegar la información del registro.

```
{'full_name': 'Daniel Agraz Vallejo', 'id': '33293', 'personal_website': 'agraz.cetystijuana.com'}
```

Asegúrate que el registro exista en la tabla, ya que de lo contrario al ejecutar el código te saltara un error.

5. Explain the difference between Growth mindset and Fixed mindset according to Carol Dweck. **10 points.**

Dweck considera que el Growth mindset es aquella mentalidad que visualiza los errores como una oportunidad de aprendizaje, que impulsa a ser cada vez más curioso con los retos a los que nos enfrentamos, por otro lado, se refiere al Fixed mindset a ese estado de autoconciencia en el que se tiene la convicción de que tus cualidades son irreversibles y no tienen espacio para mejora.

Esta Mentalidad de Crecimiento que propone Carol Dweck busca confirmar que la inteligencia se puede desarrollar, y que esto lo provoca aquel deseo de aprender y mejorar. Este concepto desemboca a que el individuo sea más perseverante cuando se encuentra con algún obstáculo en su camino, confíe en el proceso de aprendizaje y reconozca que con esfuerzo y práctica se logra el dominio de una habilidad. Otros factor que distingue esta mentalidad es la capacidad de tomar el éxito de los demás para su desarrollo personal, así como, tiene una mente abierta ante las críticas para moldear su proceso a su conveniencia.

De lado del Growth Mindset, un ejemplo es cuando a un niño se le recompensa por su buen esfuerzo por lo que hizó o participó, no necesariamente por haber salido bien en un examen.

La Mentalidad Fija, se manifiesta en el ahora y se encuentra en aquellas personas que buscan justificar su inteligencia una y otra vez. Solo tienen en mente la situación de que su habilidad se vea juzgada. Esta mentalidad tiende a tener una reacción de frustración o desesperación al enfrentarse a un reto, por lo que suelen evadirlos. No

son capaces de recibir retroalimentación útil, y constantemente se comparan ante el éxito de los demás.

Un ejemplo de como una persona puede adquirir el Fixed Mindset sería los niños que se concentran en sacar buenas calificaciones, solo para que sus papas los recompensen o halaguen su inteligencia. Otro ejemplo sería cuando un niño dice que es malo en matemáticas y se encasilla en la idea de que no puede mejorar porque no es lo suyo.

En conclusión, la percepción de ti mismo y tus alrededores, es principalmente un factor fundamental que define si tienes una mentalidad que te permita desarrollar tu potencial no solo en el ámbito académico sino también en cuestión de tu calidad como persona, dando como resultado una mayor oportunidad para cumplir con tus metas de vida y objetivos personales, o existe el caso de que no conozcas tu verdadero potencial por tener miedo a salir de tu zona de comfort y explorar nuevos retos.

Referencias Bibliográficas

Farnam Street (2023). *Carol Dweck: A summary of Growth and Fixed Mindsets.* https://fs.blog/carol-dweck-mindset/

Imperva (2022). *Minfication.* https://www.imperva.com/learn/performance/minification/

TED (2014). *The power of believing that you can improve* | *Carol Dweck.* https://www.youtube.com/watch?v=_X0mgOOSpLU&ab_channel=TED