

Assignment 2

ICE191 Software Architecture / Cloud Computing

1. Create a static website using AWS S3 and explain in great technical detail what you did. **30 points.**

Paso 1) Corre este comando para crear la cubeta

```
aws s3 mb s3://nombreDeCubeta
```

Dado que S3 se comporta como sistema de archivos, usaremos comandos similares a los que se usan para manejar sistemas de archivos, en este caso "mb" es el equivalente a "mkdir" de sistemas unix, "s3://" indica que la dirección está en S3 (es obligatorio), y después va el nombre de la cubeta, yo le puse "omar-duran-cetys.cetystijuana.com", poner "cetystijuana.com" era obligatorio y diré porque en la pregunta 2.

Paso 2) Corre este otro comando para convertir la cubeta en sitio web

```
aws s3 website s3://nombreDeCubeta --index-document index.html --error-document error.html
```

Aquí configuramos nuestra cubeta de S3 como un sitio web estático, hay dos parámetros importantes, el "index-document" que es la página principal de nuestro sitio web, y el otro es "error-document" que es el que se muestra cuando ha habido un error en nuestro sitio web (como un error 404 por acceder a algún recurso inexistente)

Paso 2b) Crea archivos para tu sitio

Crear dos archivos, uno llamado "index.html" y otro llamado "error.html", deben estar en la misma carpeta para este proceso que estoy indicando. Dentro de cada uno solo les puse un tag <h1> con texto de "Hola" y "rompiste algo" para cada sitio correspondientemente

Paso 3) Corre este comando para subir los archivos a tu sitio

```
aws s3 sync direccionFuente s3://nombreDeCubeta
```

Este comando sincroniza cubeta y computadora, a la izquierda va el origen, y a la derecha el destino, es este caso el origen es mi computadora, yo puse un punto "." para indicar que quiero que suba los archivos en el folder donde corrí el comando (porque ahí tenía mis archivos de index y error)

Paso 4) Sigue estos pasos para hacer un archivo que hará tu cubeta pública

```
touch policyBucket.json
```

```
Nano policyBucket.json
```

En este archivo puse la política de acceso para mi cubeta, la cual necesita tener acceso público, en la imagen de abajo está la política para hacer eso en mi *bucket*

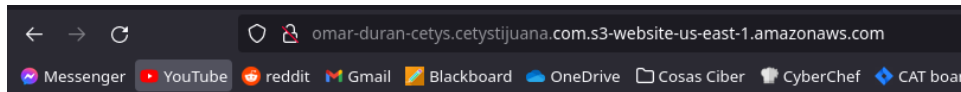
```
{
  "Policy": "{\n\"Version\": \"2012-10-17\", \"Statement\": [\n{\n\"Sid\": \"PublicReadGetObject\", \"Effect\": \"Allow\", \"Principal\": \"*\", \"Action\": \"s3:GetObject\", \"Resource\": \"arn:aws:s3:::omar-duran-cetys.cetystijuana.com/*\"}]\n}"
```

La política debe tener los siguientes datos. "Version" es el valor de versión o forma de hacer la política, el valor especificado en clase es de esa fecha. En "Statement" va una lista de cosas que se deben hacer a la cubeta, "Sid" es el nombre de lo que se hará (es libre), "Effect" es si permitimos y denegamos, "Action" es la acción que se permite o niega (en este caso siendo leer objetos de S3), "Resource" es el recurso al que se le aplica la política siendo en este caso toda mi cubeta, por eso el asterisco después de acabar el ARN de mi cubeta.

Paso 5) Corre este comando para hacer tu cubeta pública con la política del paso 4

```
aws s3api put-bucket-policy --bucket nombreDeBucket --policy \
file://direccionATuArchivoDePoliticaDeBucket
```

En este comando "put-bucket-policy" se subió la política a mi cubeta, "file://" es para indicar que en vez de escribir la política en el comando quiero usar un archivo de mi sistema.



Hola este es el sitio de Omar Duran

Éxito total, con estos pasos, el *bucket* "omar-duran-cetys.cetystijuana.com" ya es accesible para todo público.

2. Link your website to a subdomain of cetystijuana.com and explain in great technical detail what you did. **30 points.**

El primer paso fue ir a la documentación oficial de Route53 para ver cómo hacerlo con la consola de AWS

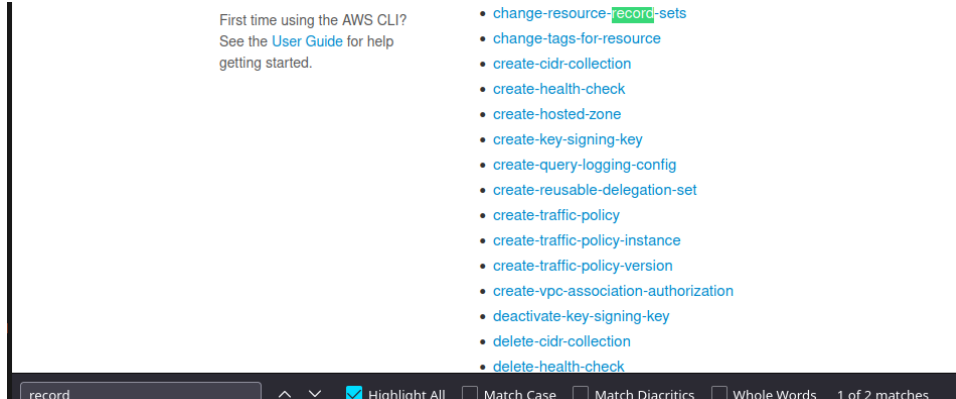
Configuring Amazon Route 53 to route traffic to an S3 Bucket

To configure Amazon Route 53 to route traffic to an S3 bucket that is configured to host a static website, perform the following procedure.

To route traffic to an S3 bucket

1. Sign in to the AWS Management Console and open the Route 53 console at <https://console.aws.amazon.com/route53/>.
2. In the navigation pane, choose **Hosted zones**.
3. Choose the name of the hosted zone that has the domain name that you want to use to route traffic to your S3 bucket.
4. Choose **Create record**.

Aquí dice que debo crear un récord. Aquí es cuando me moví a la documentación del CLI para buscar comando relacionados a récords



Las únicas dos menciones eran creación y listado de récords, por lo tanto, el de creación debía de ser el correcto

change-resource-record-sets

Description ¶

Creates, changes, or deletes a resource record set, which contains authoritative DNS information for a specified domain name or subdomain name. For example, you can use `ChangeResourceRecordSets` to create a resource record set that routes traffic for test.example.com to a web server that has an IP address of 192.0.2.44.

■ **Related Resource Record Sets**

Sí es el comando correcto.

```
{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "TTL": time to live in seconds,
        "ResourceRecords": [
          {
            "Value": "applicable value for the record type"
          },
          {...}
        ]
      }
    },
    {...}
  ]
}
```

Este *string* que acabo de copiar y pegar de la documentación del CLI (s.f) es un "resource record set" básico, usaré este porque solamente quiero redirigir el tráfico de mi cubeta a un subdominio, nada de cosas sofisticadas.

En "comment" va un comentario que describa el récord, es libre la decisión, en "Changes" va una lista de cambios, cada cambio va dentro de *brackets*, en "Action" va si se crea, borra o actualiza un récord, en "ResourceRecordSet" van unos *brackets* con datos respecto al récord.

En "ResourceRecordSet" en "Name" va el nombre del récord, o sea el URL **completo** al que se redirigirá el tráfico. En "Type" va el tipo de récord, aquí depende de

lo que quieras hacer, si vas a redirigir a un recurso de S3 debe de ser A según la documentación del CLI, pero sí es un sitio web entonces esto no funcionará porque ya lo probe y dice que la dirección IPV4 es invalida (el tipo A redirige a direcciones IP), dado que será un subdominio deberemos usar CNAME. En TTL es el tiempo que vive el récord en cache antes de que los DNSs lo deban leer denuevo.

Dentro de "ResourceRecordSet" van unos *brackets* con un solo para llave valor, siendo el del lugar de origen del tráfico, el opuesto al "Name" de "ResourceRecordSet".

```
{
  "Comment": "Esta política debería hacer que el sitio de Omar sea subdominio de cetystijuana.com",
  "Changes": [
    {
      "Action": "CREATE",
      "ResourceRecordSet": {
        "Name": "omar-duran-cetys.cetystijuana.com",
        "Type": "CNAME",
        "TTL": 100,
        "ResourceRecords": [
          {
            "Value": "omar-duran-cetys.cetystijuana.com.s3-website-us-east-1.amazonaws.com"
          }
        ]
      }
    ]
  ]
}
```

Así es como quedó mi política. En el caso de "Name" en "ResourceRecordSet" es importante destacar que debe coincidir con el nombre del dominio el final del nombre, o sea que si quiero un subdominio para lol.com, mi cubeta de S3 debe llamarse "NombreDeSubdominio.lol.com".

Ahora explicaré como es que lo subí.

```
okladec@Oktaworkstation ~/Documents/Clases8vo/Sitio % aws route53 list-hosted-zones
{
  "HostedZones": [
    {
      "Id": "/hostedzone/Z03346142C3RKH191036Y",
      "Name": "cetystijuana.com.",
      "CallerReference": "RISWorkflow-RD:fc4e3528-b645-4b45-8c60-ea43b3e4363f",
      "Config": {
        "Comment": "HostedZone created by Route53 Registrar",
        "PrivateZone": false
      },
      "ResourceRecordSetCount": 15
    }
  ]
}
```

El comando para hacer cambios de récord requiere como parámetro el "hosted-zone-ID" este es un ID que según documentación de Route53 (s.f) aloja información respecto a récords de un dominio padre (en este caso el padre sería cetystijuana.com). Para conseguir este ID corrí el comando en la imagen, el cual lista en el servicio de route53, todas las "hosted zones" a disposición del usuario. Ahí, dentro de la única entrada presente estaba el ID que necesitaba.

Una vez con esta información pude correr el siguiente comando:

```
oktadeco@oktaWorkstation /media/oktadeco/Linux_Tri % aws route53 change-resource-record-sets --hosted-zone-id Z03346142C3RKH191036Y --change-batch file:///home/oktadeco/Documents/Clases8vo/recordPolicy.json
```

De aquí lo único que me falta explicar de este comando es el "--change-batch", es cual es el parámetro que contiene los cambios a hacer (el json mostrado)

```
{
  "ChangeInfo": {
    "Id": "/change/C0543584212VMLS5HMMCC9",
    "Status": "PENDING",
    "SubmittedAt": "2023-02-04T07:50:01.609000+00:00",
    "Comment": "Esta polituica deberia hacer que el sitio de Omar sea subdominio ce cetystijuana.com"
  }
}
(END)
```

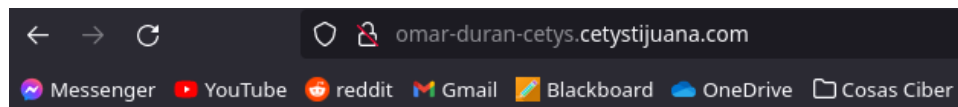
Al correrlo esto fue lo que me salió, dice "PENDING" porque es una solicitud y hay que esperar a que se hagan los cambios, para este sitio web todo ocurría al instante, o más rápido que yo volviendo a mi explorador web.

```
oktadeco@oktaWorkstation ~/Documents/Clases8vo/Sitio % aws route53 list-resource-record-sets --hosted-zone-id Z03346142C3RKH191036Y
```

Este comando nuevo ("list-resource-record-sets") muestra los récords de un hosted Zone ID indicado, en este caso del del dominio de cetystijuana.com.

```
{
  "Name": "omar-duran-cetys.cetystijuana.com.",
  "Type": "CNAME",
  "TTL": 100,
  "ResourceRecords": [
    {
      "Value": "omar-duran-cetys.cetystijuana.com.s3-website-us-east-1.amazonaws.com"
    }
  ]
}
```

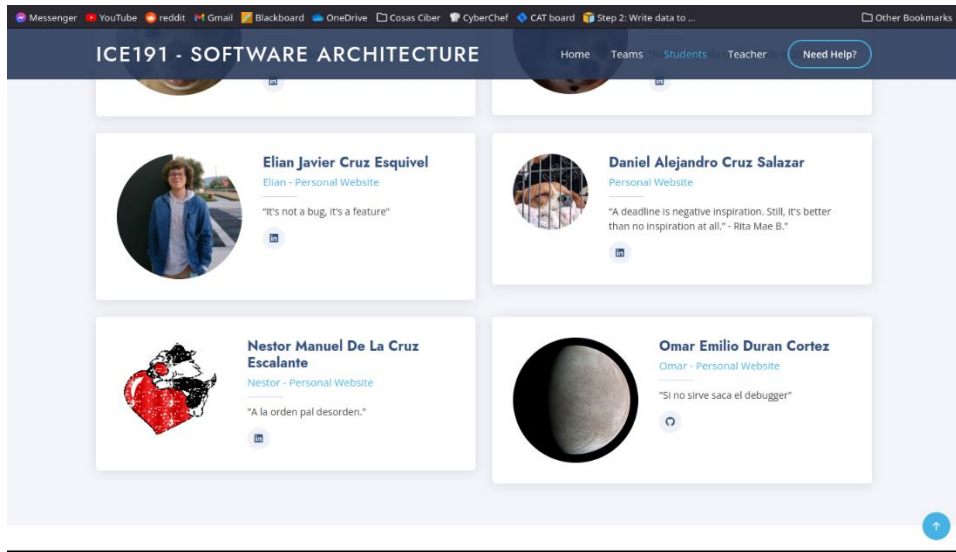
Y como se puede apreciar ahí sale mi récord con mi cubeta de S3 siendo apuntada por el subdominio mío de cetystijuana.com



Hola este es el sitio de Omar Duran

Éxito total, lo probe justo después de correr el comando y mi sitio ya era un subdominio del sitio padre.

3. Add yourself to the list of students in cetystijuana.com. 15 points



El proceso para hacer ello fue primero descargar todo el *bucket*

```
" aws s3 sync s3://cetystijuana.com . "
```

Hacer esto sincronizaba todo el *bucket* en mi computadora, o sea que descargaba sus contenidos que no tenía yo (en este caso todos). Al acabar eso entré al documento "index.html" en el *root* de la carpeta y fui a la sección de estudiantes. Ahí hice una copia del div de alguien más, y le cambie los datos para que se vea como el mío en la imagen.

Lo distinto es que puse el enlace de mi sitio, una foto distinta, y como no tengo actualmente un perfil de LinkedIn puse mi perfil de GitHub, hasta usé el icono de GitHub que había en el CSS.

Para finalizar, corrí el siguiente comando para subir los cambios:

```
" aws s3 sync . s3://cetystijuana.com "
```

Lo distinto aquí es que en vez del origen ser la cubeta, fue mi computadora, y el destino fue la cubeta.

Por si había curiosidad, la foto es de una de las lunas de Júpiter, Europa.

4. Create an entry in the Students DynamoDB table using the cli with the following model:

```
{
  "id": "matricula",
  "full_name": "YYYYYY",
  "personal_webstie": "s3_url"
}
```

a.

b. 15 points.

Corrí este comando con estos parametros:

```
oktadecc@ktaWorkstation ~/Documents/ClasesDvo % aws dynamodb put-item \
--table-name Students \
--item file:///itemOmar.json

Parameter validation failed:
Invalid type for parameter Item.id, value: 26780, type: <class 'str'>, valid types: <class 'dict'>
Invalid type for parameter Item.full_name, value: Omar Emilio Duran Cortez, type: <class 'str'>, valid types: <class 'dict'>
Invalid type for parameter Item.personal_website, value: omar-duran-cetys.cetystijuana.com, type: <class 'str'>, valid types: <class 'dict'>
```

El comando "put-item" para dynamoDB es utilizado para poner datos en bases de datos, uno de sus parámetros es el nombre de la tabla, y el otro es el "item" (la data), puede ser puesto este en la terminal como texto o apuntar a algún archivo con el prefijo "file://".

Mi JSON originalmente era de relación llaves a *string*, pero de acuerdo con documentación oficial (s.f), y los errores con los que me topé, se debe especificar el tipo de valor que se contiene en el valor, o sea que, si el valor de la llave "alumno" es "Omar", en realidad es '{"S": "Omar"}' donde "S" significa *String*. Así aplica para cada uno de los valores, por lo que a fin de cuentas los valores son diccionarios de relación llave valor donde la llave es el tipo de valor, y el valor ... sigue siendo el valor.

```
{
  "full_name": {
    "S": "Omar Emilio Duran Cortez"
  },
  "id": {
    "S": "26780"
  },
  "personal_website": {
    "S": "omar-duran-cetys.cetystijuana.com"
  }
},
```

Al haber hecho las correcciones pertinentes este fue el resultado de cómo se veía mi JSON, el comando de inserción no tuvo que cambiarlo.

Eso sí, para verificar que sí se subió bien mi información corrí el siguiente comando "
`aws dynamodb scan --table-name Students`"

De acuerdo con documentación del CLI (s.f). El comando "scan" procesa toda la tabla indicada a la vez, y retorna sus datos. En lo más básico es la forma de leer toda la tabla, y así lo hice.

5. Watch the [Not Just Code Monkeys by Martin Fowler](#) and write your comments on your personal static website. **10 points.**

Una conferencia muy interesante la verdad, el desarrollo de software es una profesión muy versátil, podemos hacer muchas cosas, pero para ello hay que ser proactivos. El software requiere de muchas personas para seguir creciendo y estoy de acuerdo con lo de la anécdota del carrito de compras de Amazon y sus sugerencias, la comunicación entre personas permite el crecimiento, no solo se trata de echar código y de seguir órdenes.

Eso me lleva a otro de sus puntos, el de los patrones oscuros. Esos siempre me molestan cuando los veo, aunque nunca les daba un nombre en peculiar porque no lo sabía. El ejemplo que más me molesta en mi día a día es como es que en Windows se la pasa insistiendo con múltiples formas de hacerte comprar Office, usar Edge, y múltiples otras cosas de Microsoft, estas prácticas son irritantes, pero parte de este mundo desafortunadamente.

Otra parte que se me hizo interesante fue que somos una profesión que todavía la falta diversidad, y estoy de acuerdo con sus métricas, si aquí los que son buenos en esto lucen, ¿porque hay pocas mujeres? Siento que esto debe de cambiar, siempre vienen bien desarrolladores profesionales, y que se diversifique y aumente la cantidad de profesionales debe de ser bueno para la profesión.

Definitivamente una plática para pensar. Me quedo con que el desarrollo de software es una profesión la cual se puede usar mucho para el bien, pero también se necesita pensar bien si lo que harás puede ser correcto para todos. Y, que, para crecer en esta profesión, se requiere mucha socialización no solo para entender el producto en el que trabajas, sino para también hacerlo crecer.

Materiales utilizados:

- <https://docs.aws.amazon.com/cli/latest/reference/route53/change-resource-record-sets.html>
 - Documentación de route 53 para crear record sets usando la consola de AWS
- <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/getting-started-step-2.html>
 - Documentación de dynamoDB para subir datos a una tabla
- <https://awscli.amazonaws.com/v2/documentation/api/latest/reference/dynamodb/scan.html>
 - Documentación oficial del CLI de dynamoDB y su comando scan
- <https://awscli.amazonaws.com/v2/documentation/api/latest/reference/route53/change-resource-record-sets.html>
 - Documentación oficial del CLI de route 53 para cambiar (crear) los "record sets"
- <https://aws.amazon.com/route53/faqs/>
 - Documentación oficial de route 53 para obtener la información de que es un "hosted zone"