

Assignment 2

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

- a. Primeramente, antes de iniciar debemos abrir nuestra consola
- b. En la terminal debemos escribir:

```
aws s3api create-bucket --bucket  
"david.rolan.cetystijuana.com" --region us-east-1
```

“aws” porque siempre tenemos que usarlo,

“s3api” porque nos permite trabajar mejor con json ademas que tiene mas funciones precisas para trabajar con los buckets, “create-bucket” comando que nos ayuda crear el bucket, “bucket” consecuente al nombre que se le asignara al bucket mismo, “--region” con esto determinamos la region que se utilizara.

- c. Una ves dimos *enter* deberia de mostrar la ubicación donde se encuentra
- d. Una de las formas para probar que esta bien creado el bucket, es introducir en la terminal `aws s3api list-buckets`

Nuevamente utilizamos comando “aws” ya se explico anteriormente, “s3api” porque queremos que nos arroje un formato json de output, “list-buckets” de esta forma llamamos los buckets que estan creados en ese momento.

- e. Si esto surgio de manera exitosa aparece nuestro bucket creado asi junto al nombre y la fecha con la que se creo.

- f. Ahora debemos agregar la policy para esto usamos el comando

- g.

```
aws s3api put-bucket-policy --bucket  
david.rolan.cetystijuana.com --policy  
///Users/davidroldanmachado/Desktop/bucket_policy-1.json
```

“put-bucket-policy” comando para agregar la poliza a nuestro bucket, “--bucket” a donde sera despues seguiria el nombre del bucket, “--policy” aquí agregaremos la ruta exacta donde se encuentra nuestro poliza en formato json el cual debera estar en el siguiente formato:

```
{
```

```

    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "PublicReadGetObject",
        "Effect": "Allow",
        "Principal": "*",
        "Action": "s3:GetObject",
        "Resource":
"arn:aws:s3:::david.rolдан.cetystijuana.com/*"
      }
    ]
  }

```

Los valores de arriba significan lo siguiente Sid: id, la forma que lo identificamos, Effect: que accion quieres tomar, la cual nosotros escogimos “Allow”, Principal: Quien puede hacer algo, en nuestro caso se escogio “*” lo cual significa que todos pueden seguir la publicReadGetObject, Action: Como se vna a comunicar quien con quien, Resource: arn significa (Amazon Resource Name) es el unico id de todos los recursos de AWS.

Para ver si nuestra policy se realizo correctamente ingresamos el comando:

```
aws s3api get-bucket-policy --bucket
david.rolдан.cetystijuana.com
```

Ahora sera “get-bucket-policy”, agregamos “bucket” y el nombre. Si esta correcto damos enter y debera de regresarnos algo asi:

```

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

```

- h. Ahora finalmente creamos nuestros website de pruebas con que tengan algo de informacion esta perfecto, por ejemplo yo utilice 2 archivos llamados index.html y error.html.

Para esto debemos utilizar el siguiente formato para asignarlos a nuestro website de prueba.

```
aws s3 website s3://david.rolan.cetystijuana.com --index-document index.html --error-document error.html
```

- i. Ahora hace falta crear estos archivos se pueden crear por la terminal mediante nano y el nombre que le asignamos anteriormente.
- j. Toca ahora hacer sync mediante el siguiente comando: `aws s3 sync .`

```
s3://david.rolan.cetystijuana.com
```

Este comando nos sincronizara todo lo de la carpeta con nuestra domino ya antes realizado

Para comprobar nuestros html que esten correctos podemos usar el comando

```
aws s3api get-bucket-website --bucket david.rolan.cetystijuana.com
```

- k. Nos debera mostrar una informacion como este ejemplo:

```
{
  "IndexDocument": {
    "Suffix": "index.html"
  },
  "ErrorDocument": {
    "Key": "error.html"
  }
}
```

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

1. Para poder unir nuestro website con el subdominio tenemos que hacer un CNAME record, lo que se busca es unir nuestro bucket con el subdominio hacia el dominio principal, de esta forma lo reedireccionaremos, la siguiente información es un ejemplo

de como iría acomodada la información del JSON, básicamente lo que hacemos es convertir nuestro url enorme que usábamos en el url simple que queremos.

```
{
  "Comment": "Creating Alias resource record sets in Route
53",
  "Changes": [
    {
      "Action": "CREATE",
      "ResourceRecordSet": {
        "Name": "david.roldan.cetystijuana.com",
        "Type": "CNAME",
        "TTL": 600,
        "ResourceRecords": [
          {
            "Value": "david.roldan.cetystijuana.com.s3-
website-us-east-1.amazonaws.com"
          }
        ]
      }
    }
  ]
}
```

2. Después de tener nuestro JSON ahora toca enviarlo esto se solicita hacia el host por medio de Route53 con el siguiente comando

```
aws route53 change-resource-record-sets --hosted-zone-id
Z03346142C3RKH191036Y --change-batch "Ruta exacta de tu
archivo"
```

3. En el output nos enfocaremos en la opcion que dice Pending, esta tarda un poco pero puede tomar su tiempo la forma de comprobar cuando ya esta sincronizada o aprobada es mediante el mismo comando o viendo si en la lista de pagians ya te encuentras visible, esto se realiza mediante este comando.

```
aws route53 list-resource-record-sets --hosted-zone-id
Z03346142C3RKH191036Y
```

4. Si realizaste los pasos correctamente ya pudiste rediriccionar una pagina utilizando Route53, ahora lo puedes comprobar ingresando tu pagina en el navegador sin "www", esta ya deberia de estar visible para el publico, por ejmplo en mi caso seria[david.roldan.cetystijuana.com]

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

1. Primeramente debemos realizar una carpeta donde guardaremos los archivos que descargaremos de la pagina.
2. Para realizar esta accion debemos utilizar el siguiente comando

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

3. Con este se empezaran a descargar los archivos a la carpeta seleccionada
4. Una vez que tengamos necesitamos modificar la parte exacta del html en este caso debnemos de fijarnos en el orden de los div
5. El mio quedo de la siguiente manera

```
<div class="row">
  <div class="col-lg-6">
    <div class="member d-flex align-items-start"
data-aos="zoom-in" data-aos-delay="100">
      <div class="pic"></div>
```

```

        <div class="member-info">
            <h4>David Roldan</h4>
            <span><a
href="//david.roldan.cetystijuana.com">Roldan's Website
</a></span>

            <p>"Todos nacemos inteligentes, unos
descubren en que lo son y otros mueren sin saberlo" -
Roldan</p>

            <div class="social">
                <a href="www.linkedin.com/in/david-roldan-
4125ab248"> <i class="ri-linkedin-box-fill"></i> </a>
            </div>
        </div>
    </div>
</div>

```

6. Posterior a esto agregamos nuestra imagen en la carpeta indicada la cual es
assets/img/team
Ahí agregaremos nuestra imagen y de ahí la extraremos
7. Ya que tengamos todo esto comprobamos que nadie haya subido una nueva version
para que de esta manera no eliminemos contenido, cuando estemos seguro que
tenemos la nueva version, procedemos a subirlo con el siguiente comando.

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

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

1. Primeramente debemos crear un json con nuestros datos lo que debemos agregar en
el, es lo siguiente:

```

{
    "id" : {"S":"27292"},

```

```
"full_name" : {"S" : "David Eliseo Roldan Machado"},  
"personal_website":{"S" : "david.roldan.cetystijuana.com"}  
}
```

2. Ya una vez que lo tengamos procedemos a subirlo.

Para proceder a subirlo tendremos que utilizar el siguiente comando

```
aws dynamodb put-item --table-name Students --item  
file:///ubicacion del archive.json]
```

En este comando ahora estamos utilizado “dynamodb” este comando se utiliza cuando se trabaja con bases de datos, “put-item” porque vamos a agregar un nuevo objeto, “—table-name” Aquí iría el nombre de la tabla a donde queremos agregar, “--item” el artículo que se insertara el cual será el archivo json.

3. Si todo sale correctamente no tendrás nada de output, para checarlo, tendremos que agregar este comando:

```
aws dynamodb scan --table-name Students
```

Con este comando nuevamente de tipo dynamodb, “scan” para escanear y mostrar todos los atributos de las tablas que seleccionemos, “—table-name” De que tabla queremos en nuestro caso sería Students

4. Por último nos debería mostrar todos los registros que se tienen en Studets
5. Watch the Not Just Code Monkeys by Martin Fowler and write your comments in yourpersonal static website. **10 points.**

La presentación empieza diciendo que muchas veces los desarrolladores crean o hacen ideas que son dadas por alguien más, porque así están acostumbrados, cuando tú también tienes ideas que puedes compartir con los demás, y nos aclara esta idea con la idea de Amazon del desarrollador que propuso agregar los si compraste esto puede que te guste esto otro, al empresario no le gustó la idea porque pensó que iba a distraer al desarrollador lo terminó agregando y por números le demostró que estaba correcto.

Otra idea que atrajo mi atención es que el programador debe interesarse y a su vez aprender más del proyecto en el que se encuentra, porque ese es el camino para desarrollar mejor software, en cuestión nos ayuda a poder colaborar y hablar de una mejor manera con todos los compañeros del equipo.

También como usuarios debemos de entender al usuario y no jugar con él, utilizando malos principios de engaños como son los conocidos “black patterns”, este era un concepto que desconocía es que es una interfaz la cual busca engañar al usuario para conseguir que compre un producto o servicio, en pocas palabras es una práctica un poco ética con el fin de solo conseguir vender más cosas. El punto como mencione debemos de poner en los zapatos del usuario y nos gustaría que usen malas prácticas o información engañosa, así que de esta forma deberíamos de hacer las cosas entendiendo y comprendiendo mejor al usuario.

Otro punto importante es entender el software que hagamos va a tener un impacto en lo que nos rodea, por lo que hagamos seremos juzgados y vistos, porque es lo que los demás verán de nuestro trabajo, para esto debemos pensar que impacto tendrá será bueno o malo para los demás.

Me encantó como terminé esto dándole más valor al área porque dijo como las profesiones como médicos que tienen status en su área, con responsabilidad mencionando las cualidades que se tiene, podrían ser líderes del mundo pero más que todo se enfocan en llevar esto con responsabilidad porque también tienen la responsabilidad de formar un mejor mundo que nos rodea. Porque esto es algo que nos suele pasar mucho en nuestra área, que se puede decir que es un poco menospreciada alrededor de las demás carreras, quien sabe porque pero se formó un estereotipo muy marcada y fuerte sobre las personas de programación o del área de la computación, pero lo que nos quiere decir este documental es que somos igual de importantes que las demás carreras, capaz en Estados Unidos actualmente ya se encuentran de una forma igualada, pero en México si falta camino por recorrer el punto es que debemos pensar y sentirnos igual de importantes que todas las demás carreras o profesiones del mundo.