

Assignment 1

ICE191 Software Architecture / Cloud Computing

Alumno : Marcos Alberto Moroyoqui Olan **Matricula** : 030934

1. Read the following article <http://martinfowler.com/articles/microservices.html> and explain what "Microservices Architecture" means. 20 points.

La arquitectura de microservicios hace referencia a un modelo de arquitectura para el desarrollo de software. Lo que caracteriza y diferencia a esta arquitectura de microservicios es la modularización y separación de servicios en unidades pequeñas, completamente independientes entre sí.

Una arquitectura de microservicios entonces es el diseño de una sola aplicación, la cual se compone de unidades o aplicaciones (servicios) más pequeños independientes, lo cual permite modularidad y escalabilidad en adición a una gran basta cantidad de beneficios relacionados con este modularidad.

De entre los beneficios que destacan debido a la arquitectura de microservicios es la independencia de los equipos de desarrollo. Al modularizar y hacer independiente cada uno de los componentes de la aplicación a desarrollar, la arquitectura de microservicios nos permite tartar cada componente como un producto distinto, por lo cual, el desarrollador no está atado a usar una tecnología en específico, siempre y cuando los métodos de comunicación con los otros productos o servicios estén estandarizados y bien definidos.

Otro punto de vista, o interpretación de la palabra arquitectura de microservicios, es la de interpretarla como una filosofía de desarrollo de software, en la cual la separación de componentes, y la division especifica y clara de funcionalidades, en productos o componentes rige las disecciones a la hora de la implementación de software.

Bajo esta filosofía de desarrollo, el equipo de desarrollo sacrifica la cohesión de los módulos y la velocidad de comunicación de distintos partes de un programa, pero al mismo tiempo al hacer esto, gana la independencia de los productos, lo que aumenta la escalabilidad, reduce los costos, y facilita la manutención de un software creado, sin embargo a su vez también aumenta la necesidad de planear correctamente la interfaz de comunicación entre los distintos servicios.

2. Install aws cli in your Mac or Linux computer and setup the config and credentials files. 20 points.

```
manmadearc@Manmade-Laptop:~/Jaquez$ aws configure --profile JaquezAws
AWS Access Key ID [None]: AF
AWS Secret Access Key [None]: gU
Default region name [None]: us-east-1
Default output format [None]: json
```

```
manmadearc@Manmade-Laptop:~/Jaquez$ aws sts get-caller-identity --profile JaquezAws
{
  "UserId": "AIDAUIDHMFQXU5TOKF4YY",
  "Account": "292274580527",
  "Arn": "arn:aws:iam::292274580527:user/marcos.moroyoqui@cetys.edu.mx"
}
```

Las credenciales fueron censuradas para mantener la seguridad de las llaves las mismas fueron cubiertas.

3. Install python and boto3 in your Mac or Linux computer. 10 points

```
1
2 import boto3
3
4 session = boto3.Session(profile_name="JaquezAws")
5 client = session.client("sts")
6
7 msg = client.get_caller_identity()
8
9
10
11 print("Received",msg)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
manmadearc@Manmade-Laptop:~/Jaquez/Tareas$ /bin/python3 /home/manmadearc/Jaquez/Tareas/TestingBoto3.py
Received {'UserId': 'AIDAUIDHMFQXU5TOKF4YY', 'Account': '292274580527', 'Arn': 'arn:aws:iam::292274580527:user/marcos.moroyoqui@cetys.edu.mx', 'ResponseMetadata': {'RequestId': 'f361439d-862a-43b3-a908-82c4df3a9eab', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid': 'f361439d-862a-43b3-a908-82c4df3a9eab', 'content-type': 'text/xml', 'content-length': '426', 'date': 'Sat, 28 Jan 2023 02:28:41 GMT'}, 'RetryAttempts': 0}}
```

4. Explain how you can have more than one access key in your computer and how to use each one. Provide code or configuration examples. 10 points.

Para poder hacer el uso de multiples llaves de acceso en la misma computadora, uno de los métodos óptimos para realizar este es mediante la utilización de la amazon cli.

Para hacer esto la aws cli maneja “perfiles” lo cuales contienen las credenciales necesarias. Para crear un perfil podemos utilizar el comando aws configure con la bandera —profile Y el nombre del perfil. Por ejemplo:

```
PS C:\Users\marco> aws configure --profile DEVTEST
```

```
PS C:\Users\marco> aws configure --profile DEVTEST
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [None]: us-east-1
Default output format [None]: json
PS C:\Users\marco>
```

Una vez que tengamos esos datos. Podemos llenar los campos que nos indica la cli de aws. Esto creara distintos perfiles los cuales podemos ver mediante el comando

Aws configure list-profiles

```
PS C:\Users\marco> aws configure list-profiles
default
DEVTEST
```

Para el uso en consola de cualquiera de estos servicios, podemos ejecutar cualquier comando de la cli, especificando el perfil a utilizar. Por ejemplo, si queremos configurar un servicio de s3, basta con llamar el comando con el parámetro `--profile` y el nombre del perfil a utilizar. Por ejemplo:

```
PS C:\Users\marco> aws s3 ls --profile DEVTEST

An error occurred (InvalidAccessKeyId) when calling the ListBuckets operation: The AWS Access Key Id you provided does not exist in our records.
```

O por el contrario, para su uso en por ejemplo el cliente de boto3 para python, basta con inicializar la sesión del cliente de boto 3, pasando un parámetro adicional, en este caso `profile name` a su instanciación, para seleccionar el perfil a utilizar

```
import boto3

session = boto3.Session(profile_name="JaquezAws")
client = session.client("sts")

msg = client.get_caller_identity()

print("Received",msg)
```

```

manmadearc@Manmade-Laptop:~/Jaquez/Tareas$ /bin/python3 /home/manmadearc/J
aquez/Tareas/TestingBoto3.py
Received {'UserId': 'AIDAUJDHMFQXU5TOKF4YY', 'Account': '292274580527', 'A
rn': 'arn:aws:iam::292274580527:user/marcos.moroyoqui@cetys.edu.mx', 'Resp
onseMetadata': {'RequestId': 'de579b29-63bb-41a6-b001-908fb8cddbcb', 'HTTP
StatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid': 'de579b29-63bb-41a6-
b001-908fb8cddbcb', 'content-type': 'text/xml', 'content-length': '426', '
date': 'Sat, 28 Jan 2023 02:17:51 GMT'}, 'RetryAttempts': 0}}

```

Con esto podemos utilizar una gran variedad de perfiles cuando se utiliza un servicio de aws en una misma computadora.

5. Explain what S3 is, how it works and how you can use it to deploy a static website. 10 points.

Amazon S3, es un servicio ofrecido por Amazon para el almacenamiento de archivos. El nombre S3 está dado por su nombre en inglés. Amazon Simple Storage Service (S3), de manera simple Amazon S3 es un servicio de almacenamiento de archivos en la nube, los cuales pueden tener múltiple propósito. Por lo que, de la manera más resumida posible, Amazon S3 es un servicio de almacenamiento en la nube.

El almacenamiento de Amazon S3, hace el uso de 3 entidades fundamentales para entender su funcionamiento estos se componen generalmente de Buckets, la unidad de almacenamiento principal la cual almacena objetos el cual puede ser cualquier archivo y los meta datos que describan que es ese archivo. Cada uno de estos objetos tiene un identificados o nombre clave que identifica al objeto dentro del bucket y es usado para definir las políticas de uso de ese archivo. Asi mediante estos 3 elementos, los buckets, los objetos y las claves, nosotros somos capaces de crear medios de almacenamiento , archivos y definir las políticas de acceso en base a ls indentificadores o claves, lo cual nos permitira un acceso seguro en la nube a nuestros archivos.

En un sentido muy amplio el servicio de almacenamiento de amazon, consiste en “buckets” o cubetas, las cuales pueden ser interpretados como un folder o file system que guarda toda la información la cual puedas subir, de esta manera al crear un bucket tu estas creando un espacio o disco de almacenamiento en la nube, a la cual puedes acceder mediante un enlace. Los buckets de amazon s3 tienen una gran cantidad de usos, los puedes usar simplemente para almacenar archivos de alguna aplicación, backups , etc.

De entre estas capacidades que brinda S3, es que el mismo servicio permite hacer hosting de sitios web estáticos.

Para hacer el deploy de un sitio web en Amazon S3, podemos hacer uso de la consola de comandos propia de amazon, AWS CLI.

Para lograr esto, lo primero que necesitamos hacer es crear un “bucket” o unidad de almacenamiento.

Esto Se puede lograr mediante el siguiente comando

```
> aws s3api create-bucket --bucket Static-Website
```

Este comando lo que hara es crear por defecto un bucket en nuestra cuenta de AWS en la region us-east-1. Por otro lado, si queremos cambiar la region en la que se crea el bucket, podemos hacer uso de los parámetros `--region` para especificar la region, en conjunto con `--create-bucket-configuration LocationConstraint=` Nombre de la region”.

Una vez creada nuestro bucket, necesitamos poder habilitar el acceso público a nuestros datos, para eso necesitamos cambiar la política de privacidad o acceso del bucket que hemos creado.

Para esto podemos crear un archivo `policy.json`, que contenga la siguiente estructura:

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

Esta configuración lo que nos permitirá, es dar el acceso de lectura, para obtener o pedir los archivos dentro de nuestro bucket. Esto lo podemos ver en el son anterior en la seccion de interior de Statement, donde El Effect es permitir, principal hace referencia a quien, en este caso el asterisco represente a cualquiera, la acción es el hecho de obtener un objeto, y el resource o recurso, es cualquier elemento que se encuentre dentro de nuestro bucket Static-Website.

Creamos este archivo dentro del directorio en el que estemos ejecutando el aws cli, y lo ejecutamos mediante el siguiente comando.

```
PS C:\Users\marco> aws s3api put-bucket-policy --bucket Static-Website --policy file://policy.json
```

Una vez actualizado esta política, podemos entonces comenzar a subir nuestros archivos a el bucket que hemos creado. Esto lo podemos hacer dirigiéndonos al directorio que contiene nuestros archivos html. Una vez en el directorio solo necesitamos ejecutar el siguiente comando.

```
PS C:\Users\marco> aws s3 sync ./ s3://Static-WebSite/
```

Donde ./ hace referencia al directorio que queremos subir y s3://Static-WebSite, es el nombre del bucket al que queremos subir nuestros archivos.

Una vez creado este directorio, solo nos hace falta notificarle a amazon que el bucket sera destinado a ser un sitio web estático, para eso podemos usar el siguiente comando.

```
PS C:\Users\marco> aws s3 website s3://Static-Website --index-document index.html --error-document error.html
```

Este comando configurara el bucket como un sitio web, donde s3://Static-WebSite es el nombre del bucket a configurar, --index-document especifica cual es el documento de inicio, y error es elk documento que se mostrara cuando la página no pueda ser encontrada.

Con esto solo quedara acceder al nuestro bucket mediante el enlace de acceso, el cual se compone de la siguiente manera
<bucket-name>.s3-website.<AWS-region>.amazonaws.com.

6. Explain what CloudFront is, how it works and how you can use it to deploy a static website. 10 points.

CloudFront es un servicio CDN Proveído por Amazon. Este servicio de CDN (Content Delivery Network), es un servicio el cual se enfoca en la distribución de contenido web estático y

dinámico a lo largo de diversos servidores de internet, con la facilidad de reducir el tiempo el cual se tarda en entregar o acceder al contenido de la web a la cual se busca acceder.

Podemos pensar en el servicio de CloudFront como un servicio de distribución de contenido a lo largo de todo el planeta. Por ejemplo, cuando un usuario intenta acceder a un contenido de internet, en lugar de conectarse a el servidor más cercano, CloudFront crea copias del sitio web a lo largo de todo el planeta, y hace la lógica necesaria para redirigir a el usuario, a aquel servidor que tenga el contenido replicado y que sea más cercano al mismo, o a aquel que tenga menos demanda y lo pueda atender de manera más rápida.

Podemos pensar en una CDN y CloudFront como una analogia a un centro de distribución. Donde cuando tú quieres comprar cualquier producto, no acudes a la fábrica donde maquilan el producto, si no que en su lugar accedes a una tienda o almacén más cercano a tu ciudad, de esta manera si el producto existe en el almacén cercano a tu localidad, lo que haces es ir a ese almacén a solicitar el producto, por el contrario, si vas a ese almacén y el producto no está disponible, el almacén es el que se encarga de pedir en el almacén.

Esta analogia también sirve para describir el funcionamiento de un servicio de CDN, donde en una primera instancia CloudFront se encarga de distribuir su contenido a lo largo de todos sus nodos o servidores de replicación de contenido, y cuando un usuario hace un pedido a el contenido, No suggestions se encarga de seleccionar el servidor que más rápido puede atender la solicitud de contenido, una vez llega al servidor, el servidor revisa en su cache si el contenido que tiene esta actualizado, o si está disponible, de ser así se envían los datos al usuario, de lo contrario el servidor hace un request al recurso original para acceder al contenido y en cuanto este lo recibe, se guarda en la cache del servidor y se envía al usuario que solicito la información.

. Para hacer deploy a un sitio web estático mediante CloudFront, el procedimiento es bastante sencillo.

Asumiendo que realizamos al proceso anteriormente descrito para crear un bucket en amazon s3. Una vez se ha realizado este proceso, únicamente hace falta el crear una distribución del contenido de nuestro bucket, para obtener acceso a esos archivos, mediante el siguiente comando.

```
PS C:\Users\marco> aws cloudfront create-distribution --origin-domain-name Static-Website.com --default-root-object index.html
```

Donde --origin-domain-name es la url donde se encuentra almacenado el contenido original en nuestro caso la url provida por amazon y --default-root-object es aquel archivo dentro del bucket el cual sera distribuido o replicado.

Una vez que se ha realizado lo anterior, el servicio nos proveerá con una URL de nuestra distribución la cual se mirará similar a la siguiente forma <distribution-id>.cloudfront.net

Finalmente, para terminar de configurar un dominio, solo quedara el crear un routing en nuestro servicio de DNS de referencia a la URL proveída por nuestra distribución de CloudFront.

7. Explain what an API is. 10 points.

API por sus siglas en ingles es una Application Programing Interface. De manera general se puede pensar que un api, es un mecanismo de comunicación entre dos softwares distintos, es decir es una herramienta dentro del software la cual nos permite coordinar o controlar diversas funciones del código sin tener la necesidad de conocer o utilizar los métodos del software o aplicación a controlar.

Para entender un poco mejor el concepto de API imaginemos el caso de una aplicación de banco, la cual da las opciones de consultar datos, hacer pagos, registrar pagares, entre otras. Si en algún momento, algún desarrollador quisiera utilizar o crear una nueva función para hacer un pagare, sería muy complicado que el desarrollador fuera proveído del código fuente de los servicios de banco, ya que el mismo tendría que aprender la arquitectura de la aplicación y la lógica para generar todos estas acciones o transacciones.

Para evita esto, es precisamente que las API, fueron creadas. Mediante las API, los creadores de la aplicación de banco pueden definir y crear interfaces o métodos que les permitan a estos programadores crear otra aplicación que se aprovechen de los servicios de la banca ya desarrollada. Para esto los desarrolladores del banco exponen una serie de métodos de comunicación con estructuras definidas para realizar procesos enteros dentro de la aplicación sin tener que conocer su lógica.

Para esto el desarrollador debe establecer una serie de requerimientos o un estándar de comunicación que permita saber a el desarrollador externo, como comunicarse con la aplicación y hacer una llamada o una solicitud con la información necesaria para realizar cualquier tipo de movimiento dentro de la banca en línea, ya sea un pagare, una consulta de saldo etc.

De esta manera las API resultan ser simples mecanismos de comunicaciones entre distintas tecnologías los cuales permiten el funcionamiento conjunto de dos sistemas diferentes.

8. Explain what a Restful API is. 10 points.

Como mencionamos anteriormente, las API's son mecanismos de comunicación o interfaces de comunicación que permiten intercambiar mensajes entre distintas tecnologías o aplicaciones. Para poder lograr esto, las API deben establecer una estructura o un medio por el cual comunicarse para que ambas partes puedan entender la información que se quiera transmitir.

Bajo este concepto es que se desarrollaron diversos métodos de comunicación, entre las API's, entre ellas podemos encontrar estándares de comunicación como lo pueden ser, API SOAP

(Protocolo de Acceso A objetos Simples),, o las API RPC(Procedimientos remotos), las API de WebSocket las cuales permiten comunicación bidireccional o las API REST, las cuales son comunicaciones De Transferencias de estado Representacional. Cada una de estas establece una manera distinta de hacer la comunicación y de funcionar a la hora de consumir o hacer uso de estas API.

Cuando hablamos de API's Restful a menudo nos referimos en lugar a un protocolo o estándar particular de comunicación, a una guía de diseño de arquitectura, por lo cual no existe realmente un estándar a la hora de llevar a cabo la comunicación, sin embargo, este modelo de API's si surge o crea una serie de limitaciones a la hora de llevar a cabo el desarrollo de un api bajo este concepto de arquitectura.

Entre las reglas que encontramos para el desarrollo de un api rest, podemos encontrar las siguientes:

- El api rest debe componerse de 3 elementos. Un Cliente, un servidor y recursos. El cliente es el encargado de solicitar la información, el servidor es el encargado de procesar la información necesaria y devolverla al cliente según se necesario y los recursos son aquellos datos que son necesarios y solicitados por el cliente y procesados por el servidor.
- La comunicación entre el cliente y el servidor debe ser sin estado. Este concepto hace referencia a que el servidor debe completar las solicitudes de manera independiente, es decir una solicitud no debería afectar el comportamiento de otra solicitud, por lo que ambas solicitudes deberían ser atendidas de la misma manera.
- Se debe proveer una interfaz uniforme la cual permite la transición de información de manera estandarizada, para esto se proponen 4 condiciones fundamentales
 - Los recursos deben poder ser identificables e independientes de la representación enviada al cliente
 - El cliente debe poder manipular los recursos con base a la información que recibe
 - El servidor debe de poder enviar metadatos al cliente los cuales le permitan utilizar o modificar la información del recurso
 - El cliente deben poder recibir información o enlaces que le permitan describir dinámicamente más recursos o funcionalidades relacionados al objeto que manipulan.
- El sistema en capas. La API rest debe estar estructurada en capas jerárquicas, las cuales tengan capas intermediarias que ofrezcan funcionalidades adicionales como balancear cargas, entre otros procesos y que además sean invisibles para el cliente.
- El api debe poder proveer de datos que puedan ser almacenados en cache y agilicen el proceso de comunicación entre el cliente y el servidor.
- Código bajo demanda. Bajo la arquitectura rest un api debe poder ser capaz de enviar código ejecutable que le permita al cliente aumentar las funcionalidades de su aplicación.

En general las API RESTful , son una arquitectura de API's las cuales se enfocan en proveer al usuario de recursos similares e identificables, para que la comunicación entre los distritos servicios fluya de manera dinámica.

9. Explain what you are going to do different this semester based on last semester experience. 10 points.

Con base a las experiencias del semestre pasado, lo que hare diferente durante este semestre es utilizar o dirigir mis técnicas de programación mediante el uso de TDD (Test Driven Development), lo cual me permitirá crear casos de prueba, de manera que aseguren la calidad de mi código a la hora de crear cualquier tipo de software.

De la misma manera, buscara el implementar la estrategia de desarrollo de proyecto que hemos visto durante los semestres anteriores para poder llevar acabo un desarrollo más organizado de la materia.

Además de organizar mejor mis tiempos entre el trabajo y la escuela para poder entregar productos de calidad en ambos campos, mediante una mayor planeación del software a realizar para que de esta manera, se pueda documentar de manera más sencilla y comprender más fácil el flujo de las necesidades del software, así como los de la clase.

Referencias:

- ¿Qué es una API y cómo funciona?* (2022). Redhat.com.
<https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces#diferencias-entre-soap-y-rest>
- ¿Qué es una API de REST?* (2022). Redhat.com.
<https://www.redhat.com/es/topics/api/what-is-a-rest-api>
- ¿Qué es API RESTful? | Guía sobre API RESTful para principiantes | AWS.* (2023). Amazon Web Services, Inc.
<https://aws.amazon.com/es/what-is/restful-api/>
- ¿Qué es Amazon S3? - Amazon Simple Storage Service.* (2023). Amazon.com.
https://docs.aws.amazon.com/es_es/AmazonS3/latest/userguide/Welcome.html#CoreConcepts
- Usar CloudFront para actuar como servidor de un sitio web estático alojado en Amazon S3.* (2022). Amazon Web Services, Inc.
<https://aws.amazon.com/es/premiumsupport/knowledge-center/cloudfront-serve-static-website/>
- Lewis, J. (2014). *Microservices.* Martinfowler.com.
<https://martinfowler.com/articles/microservices.html>

Khandelwal, R. (2023). *Setting up CloudFront and S3 Bucket using AWS CLI*.
Linkedin.com.

[https://www.linkedin.com/pulse/setting-up-cloudfront-s3-bucket-using-aws-cli-
-raghav-khandelwal/](https://www.linkedin.com/pulse/setting-up-cloudfront-s3-bucket-using-aws-cli-raghav-khandelwal/)

Create AWS S3 bucket as a static website with AWS CLI. (2019, November 18).
Blog.eq8.Eu.

<https://blog.eq8.eu/til/create-aws-s3-bucket-as-static-website-with-cli.html>