

Assignment 1

ICE191 Software Architecture / Cloud Computing

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

Esta es una arquitectura de software que está basada en microservicios. Los microservicios son una forma de distribuir en partes algún proyecto/producto de software. Estos son el antónimo de la forma monolítica de hacer software, donde todo el programa o servicio está en un solo lugar, y por ende cualquier cambio puede causar repercusiones en todo el sistema, lo mismo aplica para fallos, un fallo puede estropear todo el sistema.

Los microservicios en sí son programas pequeños (relativos a un programa monolítico que cumpla la misma funcionalidad) que tienen el objetivo de encargarse de tareas específicas dentro de un proyecto. Por ejemplo, podrías tener un microservicio en una página de ventas web para notificar a los usuarios de los estados de sus pedidos, y otro para procesar configuraciones de los usuarios.

Los microservicios se comunican entre sí por medio de APIs (más información en sus preguntas correspondientes) usando solicitudes HTTP. El beneficio de esta arquitectura es que los componentes se pueden modificar independientemente sin tener que modificar otras partes del servicio entero. Esto también afecta los *deployments*, así que en vez de subir todo el proyecto entero de nuevo, solo se sube el microservicio que se cambió. Con experiencia puedo decir que esto es verdad, porque si donde trabajo la aplicación a la que se dedica mi equipo fuera completamente monolítica, subirla al ambiente de producción tardaría como una película del señor de los anillos.

Otro beneficio es la descentralización, lo primero es de bases de datos. Cada microservicio o conjunto de microservicios cuenta con sus propias bases de datos, por lo que en vez de tener una muy grande, hay varias chiquitas con lo que necesitan, así si una falla no se perjudica otras, también por esto es más fácil de hacerle cambios a una porque es más pequeña que alguna que habrá en una monolítica. La descentralización

Hecho por: Omar Emilio Durán Cortez // 26780 (C8)

también ocurre por el lado del negocio, en vez de un equipo lidiar con el proyecto, múltiples pueden trabajar para este.

Versión 2.9.18 del CLI instalada en mi sistema Debian 11 en WSL 2

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

```
omar@Cuaderno-De-1-KG:~$ python3
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

```
omar@Cuaderno-De-1-KG:~$ python3
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import boto3
>>> boto3
<module 'boto3' from '/home/omar/.local/lib/python3.9/site-packages/boto3/__init__.py'>
>>>
```

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 hacer esto debemos modificar el archivo de credenciales. Este se genera automáticamente después de instalar el CLI de AWS y por lo menos un ID ha sido configurado, normalmente se debería encontrar en la dirección “~/.aws/credentials”.

De acuerdo con documentación oficial de AWS (s.f), lo que debemos hacer es lo siguiente:

```
[default]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY

[user1]
aws_access_key_id=AKIAI44QH8DHBEXAMPLE
aws_secret_access_key=je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY
```

Lo que está ocurriendo en esta imagen es que las últimas tres líneas fueron añadidas, entre *brackets* dice “user1” porque ese es el nombre que se usará en el CLI cuando quieras especificar que usarás ese usuario, las demás son la llave de acceso y llave secreta del otro perfil. Pero eso no es todo, también se debe modificar el archivo de configuración, afortunadamente se encuentra en la misma carpeta (se llama “config”).

```
[default]
region=us-west-2
output=json

[profile user1]
region=us-east-1
output=text
```

A diferencia del archivo previo, no solo se debe poner entre *brackets* “user1” sino también “profile” al inicio.

Si todo eso se hizo bien, ahora en el CLI para especificar un perfil al final de cada comando se debe poner “--profile nombreDeUsuarioAdicional” de no hacerlo se ejecutará el comando con el perfil predeterminado (“default”).

Repetir este proceso por cada usuario que se quiera añadir, solo se debe mantener consistencia en el nombre del perfil nuevo entre archivo de configuración y de credenciales.

En el caso de boto3, de acuerdo con documentación oficial de Boto3 (s.f) se deben añadir estas líneas en tu archivo de python

```
import boto3  
  
session = boto3.Session(profile name='dev')  
dev_s3_client = session.client('s3')
```

En este caso, el nombre del perfil debe coincidir con lo puesto en archivos de credenciales y configuración

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

Amazon S3 es uno de los servicios que provee Amazon Web Services. Dejando los requisitos no funcionales que provee a un lado como su seguridad, fácil acceso, optimización, etc. S3 permite el almacenamiento de datos, sea videos, fotos, archivos de texto, logs, configuraciones, etc.

Esto ocurre por medio de almacenamiento a base de objetos, por lo que me saldré de tema un momento. Por definición formal, este tipo de almacenamiento según Google Cloud (s.f) es data almacenada de forma no estructurada la cual se acomoda en bloques (conjuntos, uniones, ...) con *metadata* para identificar sus contenidos. Es utilizado porque ciertos datos no son apropiados para bases de datos relacionales, ¿Cómo metes un archivo de audio en una tabla de SQL? De que se puede se puede si lo abstraes, pero es mejor darle un alcance en base a documentos.

Con la tangente por el lado. S3 funciona creando una cubeta (*bucket*) la cual es su forma de referirse a sus recursos de almacenamiento. Tu metes objetos en tus cubetas como si fuese algún almacenamiento de archivos en la nube, y una vez que son subidos a las cubetas son ahora parte del Internet, pero por defecto no accesibles para el público general.

Dado que permite poner archivos en Internet, y por defecto un explorador de Internet intentará correr archivos HTML como páginas web, podemos alojar en una cubeta de S3 un sitio web estático nada más subiendo nuestros archivos del sitio y darle el hipervínculo al usuario al archivo correcto. Obviamente hay que hacerlos de acceso público.

Sitio estático significa que no hay *backend* de servidor, así que nada de código dinámico de *frameworks* como ASP.NET, Angular, entre otros, solo habrá html+css+js vainilla.

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

CloudFront es otro servicio de AWS, este se autoproclama (s.f) un servicio de *Content Delivery Network*. Así que me iré en tangente rápidamente de nuevo, una CDN es una red encargada según Cloudfare (s.f) dedicada para el cacheo de contenidos web, no es una alternativa al alojamiento, sino que es usado para dar contenido a múltiples pates del mundo sin sobrecargar al servidor alojador, aunque esto solo ayuda en lecturas, no escrituras, para esto necesitas más servidores.

En el caso de CloudFront, este cachea en sitios eje (*edge*) contenidos tanto estáticos como dinámicos para distribuir a los usuarios contenido de S3, instancias de EC2, entre otros. Los sitios eje son sitios físicos en donde hay servidores de AWS para este almacenamiento de cache, lo cual provoca que en vez que el usuario acceda al archivo original en algún centro de datos lejano, accederá a una copia en un lugar cercano a su ubicación.

También es compatible con el servicio de AWS Lambda, el cual permite correr código *serverless* en tu red de contenido, *serverless* siendo que todo corre en la nube y no debes preocuparte ni por hardware ni dependencias, ni infraestructura solo tienes que programar y ya.

La forma en que nos puede servir para hacer un sitio web estático sería usándolo en conjunción con S3, no tiene chiste tener una CDN sin sitio web. Mientras que en S3 crearemos una cubeta con archivos, en CloudFront haremos una distribución web. Se crearía una distribución (S3 genera cubetas de datos, CloudFront distribuciones de contenido) la cual apuntará a nuestro *endpoint* de la cubeta de S3 (URI del recurso).

7. Explain what an API is. 10 points.

Una API es algo que se le refiere en inglés como *Application Programming Interface*, es algo en un programa que le permite comunicarse con otros programas de forma más sencilla (en comparación a escrituras en disco, sockets web, y otros medios a mano).

Se les dice interfaz porque le permite interactuar con la aplicación dado que muestra los comportamientos que se pueden hacer con tal programa. Por definición formal, según Amazon Web Services (s.f) una API es un mecanismo que permite a dos *softwares* comunicarse en base a protocolos ya predefinidos.

Unas formas de operaciones comunes en APIs se les conoce como CRUD, siendo *Create*, *Read*, *Update*, y, *Delete*, las cuales están relacionadas al acceso o modificación de elementos en bases de datos.

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

Las APIs RESTful son APIs de tipo "*representational state transfer*" ¿Ok pero que significa eso? Por definición formal según RedHat (s.f) una API se considera que es REST cuando: hay data "cacheada", las solicitudes se manejan por HTTP, se hace la jerarquía de todo el servicio invisible al usuario (servidores encargados de seguridad, replicación, manejo de carga), y lo más importante: que sea *stateless*, sin estados. *Stateless* es que una solicitud es suficiente para conseguir algún recurso siempre y cuando se dé la información adecuada, y que no se mantiene contexto de las interacciones en los *queries*.

Una forma de explicarlo más a fondo es compararlo con una de sus contrapartes, las APIs SOAP. De acuerdo con Geeks for Geeks (2022) APIs REST permiten múltiples tipos de formatos mientras que SOAP solo maneja formato XML, SOAP no usa HTTPS, mientras que REST sí. Y, SOAP es muy rígido, con muchos detalles y especificaciones que ni han sido tocados desde 2009 según los documentos de W3C, mientras que REST que solo debe basarse en los principios establecidos.

Las APIs REST sí soportan operaciones CRUD.

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

No es por decir que mi semestre pasado fue perfecto (no lo fue), pero no hay muchos espacios para mejorar mi acercamiento a lo que se me pone enfrente. Trabajar en las mañanas y tener escuela en las tardes/noches me da tan poco tiempo para hacer cosas que las pocas sobras que tengo apenas me sirven para mantener "momentum" y trabajar de forma concentrada.

Tampoco podría decir que estudiar más para los exámenes, y no porque no quiera/pueda, sino porque no los tendré este semestre (al menos en modalidad de libro cerrado en hora de clase).

Lo más "razonable" que podría hacer es, estudiar más los fines de semana, repasar los temas, leer capítulos de libros al respecto (en especial para la materia de redes y esta misma), ver videos bien explicados. Aunque, tampoco libros completos ni dedicar todo el fin de semana, tampoco voy a disponerme tanto, por mi propio bien.

Recursos utilizados:

Amazon Web Services. (Junio 2020). *Getting started with Amazon S3*. [en línea].

Recuperado de: <https://www.youtube.com/watch?v=e6w9LwZJFIA>

Amazon S3. (s.f) En *AWS*. [en línea]. Recuperado de:

<https://aws.amazon.com/s3/?nc=sn&loc=1>

Credentials. (s.f). En *Boto3 Docs*. [en línea]. Recuperado de:

<https://boto3.amazonaws.com/v1/documentation/api/latest/guide/credentials.html>

Difference between REST API and SOAP API. (2022). En *Geeks for geeks*. [en línea].

Recuperado de: <https://www.geeksforgeeks.org/difference-between-rest-api-and-soap-api/>

Named profiles for the AWS CLI. (s.f) En *AWS*. [en línea]. Disponible en:

<https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-profiles.html>

SOAP Version 1.2 Part 0: Primer (Second Edition). (2007). En *W3C*. [en línea].

Recuperado de: <https://www.w3.org/TR/2007/REC-soap12-part0-20070427/>

Speeding up your website with Amazon CloudFront (s.f) En *AWS*. [en línea]. Recuperado

de: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/website-hosting-cloudfront-walkthrough.html>

What is an API. (s.f). En *AWS*. [en línea]. Recuperado de:

<https://aws.amazon.com/what-is/api/>

What is Object storage? (s.f) En *Google Cloud*. [en línea]. Recuperado de:

<https://cloud.google.com/learn/what-is-object-storage>

What is a CDN? | How do CDNs work? (s.f). En *CloudFlare*. [en línea]. Recuperado de:

<https://www.cloudflare.com/learning/cdn/what-is-a-cdn/>

Hecho por: Omar Emilio Durán Cortez // 26780 (C8)