



Ejercicios Python 12/5

Alumnos: 👤👤

- Franco Pignanelli
- Ian Porra
- Fabricio Moreira
- Candela Serra
- Franco Rolon



Repositorio en la nube: [https://github.com/francopig/aws-python/tree/main/Ejercicios de clase/04. Ejercicios 12-5](https://github.com/francopig/aws-python/tree/main/Ejercicios%20de%20clase/04.%20Ejercicios%2012-5)

Ejercicio 1

Consigna

Definir una función que reciba una cierta cantidad de números y los retorne ordenados. Si recibe letras las borra. Los números repetidos no se eliminan.

Solución

```
def ordenar(cadena):

    #quitamos caracteres
    limpio = ""
    for caracter in cadena:
        if caracter.isdigit():
            limpio += caracter

    #ordenamos
    ordenado = ""
    for h in range(len(cadena)):
        for i in range(len(cadena)):
            if cadena[i] == str(h):
                ordenado += str(h)

    return(ordenado)

print("hola")
print(ordenar("01010101"))
print(ordenar("01242151235"))
print(ordenar("01q+242aa15123r"))

"""
Salida:
hola
00001111
01112223455
0111222345
"""
```

Ejercicio 2

Consigna

Función que recibe 3 strings, se busca el segundo string dentro del primero y se reemplaza por el tercero.

Tratamos de resolverlo sin utilizar replace() para que no esté tan regalado 🧐👉

Solución 1

Si str 2 y 3 solo tienen un carácter

```
# Función que recibe 3 strings, se busca el segundo string dentro del primero y se reemplaza por el tercero.

def reemplazador(str1,str2,str3):

    respuesta = ""
    for i in range(len(str1)):
        if str1[i] == str2:
            respuesta += str3
        else:
            respuesta += str1[i]
    return respuesta

print(reemplazador("FrancoPig","a","x")) #salida: FrxncoPig
```

Solución 2

Solución más robusta, funciona incluso si str2 y str3 tiene más de 1 carácter.

(Acá tuve que usar ChatGPT :(💔)

```
def reemplazador(str1, str2, str3):
    respuesta = ""
    i = 0
    while i < len(str1):
        if str1[i:i+len(str2)] == str2:
            respuesta += str3
            i += len(str2)
        else:
            respuesta += str1[i]
            i += 1
    return respuesta

print(reemplazador("FrancoPig","an","xx"))
```



```
if str1[i:i+len(str2)] == str2
```

Aquí se está comparando si la subcadena de `str1` que comienza en la posición `i` y tiene una longitud de `len(str2)` es igual a la cadena `str2`.

El operador de rebanado (slice) `[i:i+len(str2)]` se utiliza para extraer una subcadena de `str1` que comienza en la posición `i` y tiene una longitud de `len(str2)`. Luego, se compara si esta subcadena es igual a `str2`.

Si la subcadena es igual a `str2`, se ejecuta el código dentro del bloque del `if`, lo que significa que se agrega `str3` a `respuesta` y se aumenta `i` en `len(str2)`. De esta forma, se salta la subcadena coincidente en la siguiente iteración. Si la subcadena no es igual a `str2`, se ejecuta el código dentro del bloque del `else`, lo que significa que se agrega el carácter actual de `str1` a `respuesta` y se aumenta `i` en 1.

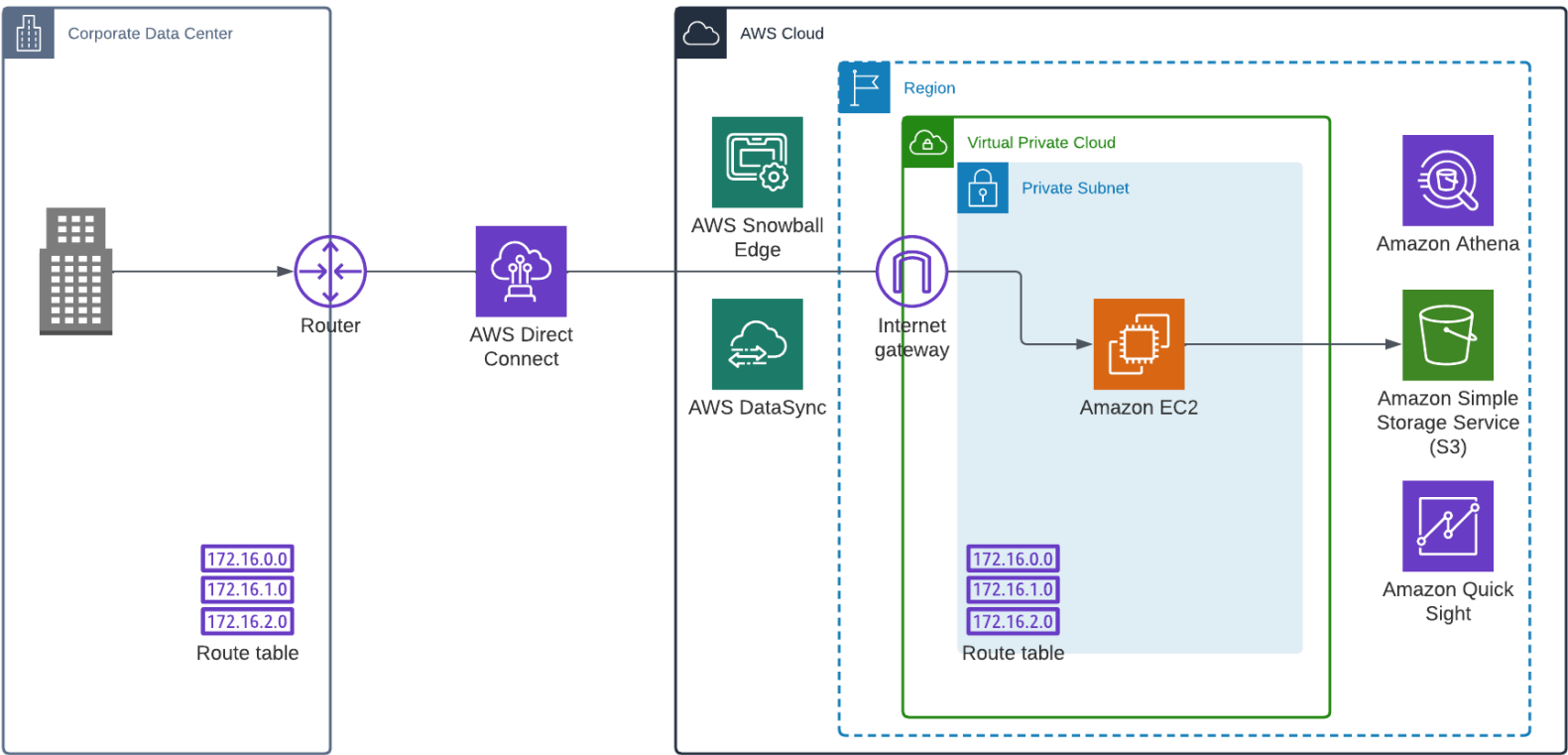
Ejercicio 3 - Arquitectura

Consigna

Una empresa tiene un volumen de datos de 80 TB en un servidor local, con internet de transferencia de datos de 10 megas, el cliente quiere llevar su datos a AWS con la finalidad de que algún servicio le ayude a borrar las columnas duplicadas y después elaborar gráficos.

Diseñar la arquitectura.

Solución



El cliente se le asigna una IP con AWS Direct Connect incrementamos el ancho de banda y reducimos el costo de los grandes volúmenes de tráfico. Entrando a la red privada a través del internet Getaway nos encontramos con la EC2 para procesar los datos los cuales se almacenan en un S3, Snowball Edge y DataSync en conjunto nos ayudan para una transferencia de datos continua. Con Athena tenemos el filtrado de los datos usando SQL y por el lado de QuickSight tenemos las visualizaciones. La S3 y la EC2 están dentro de la misma región para mejorar la latencia.