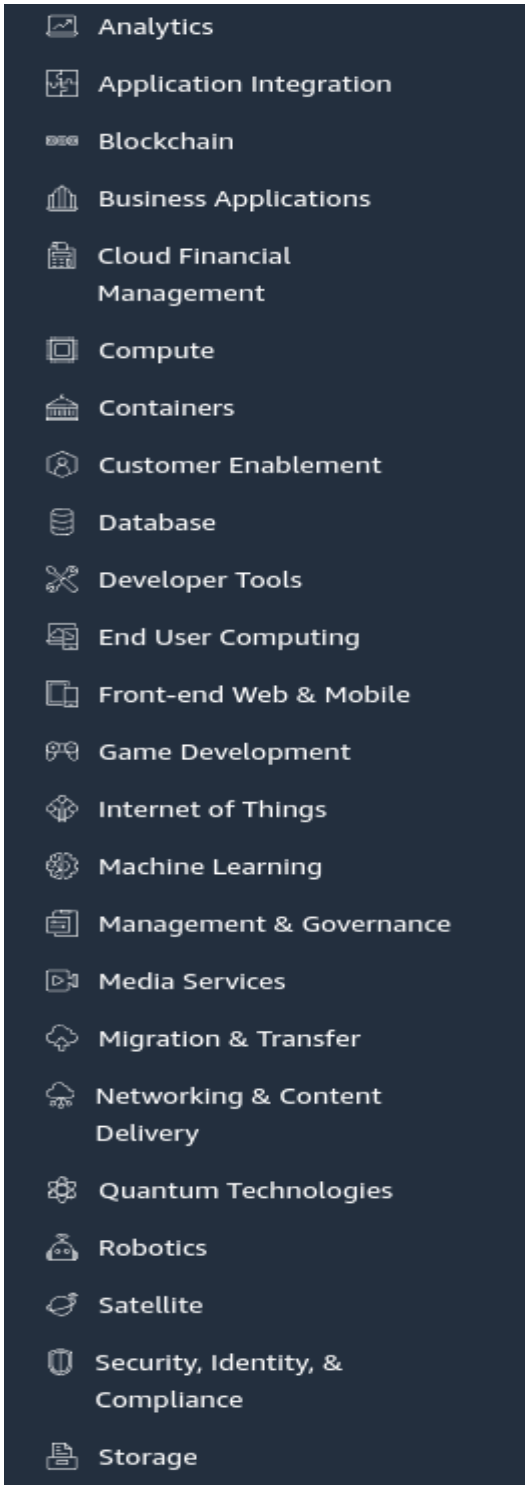
What I did with AWS and DynamoDB?

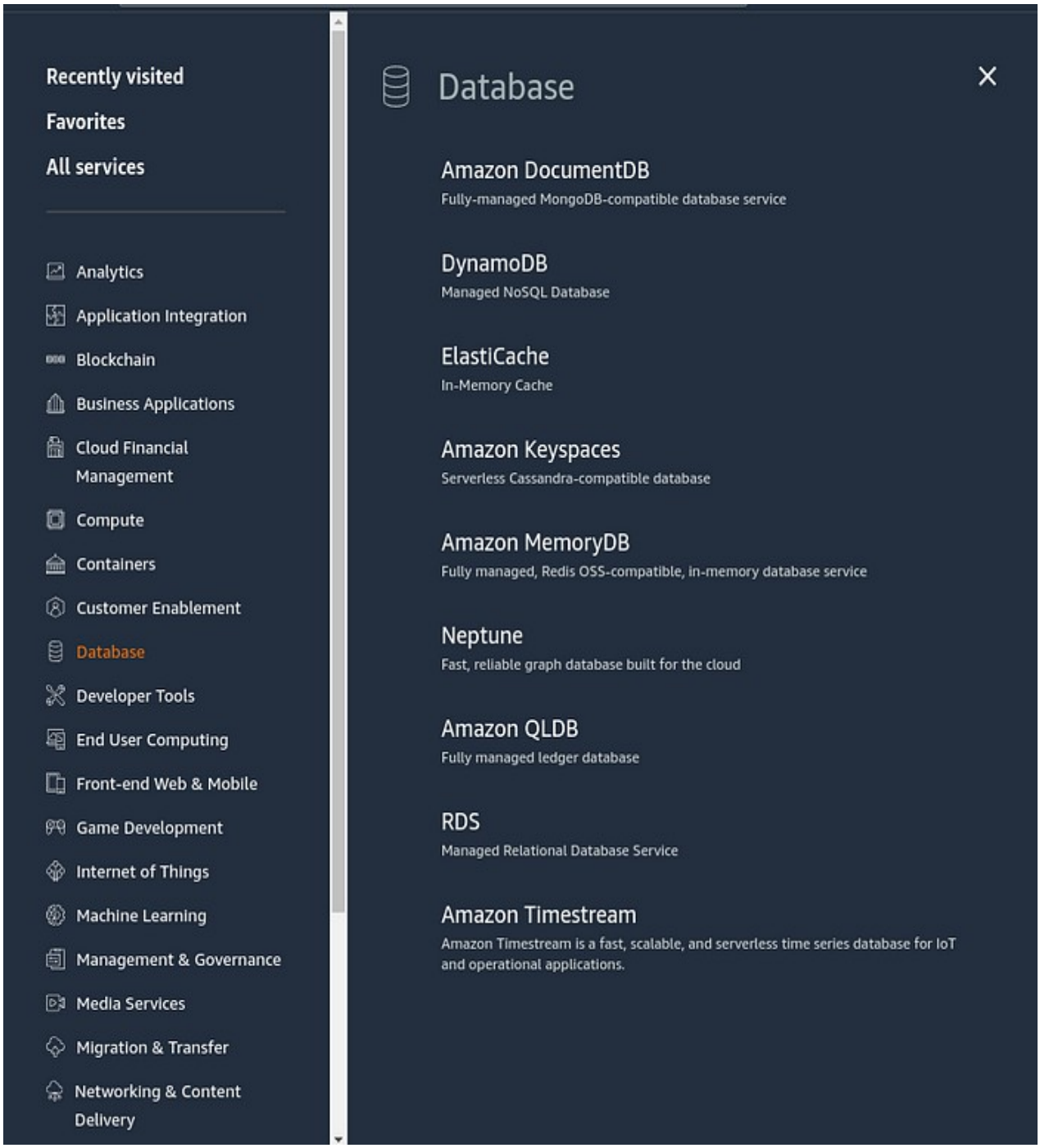
hi! I made some additions to a desktop application I developed with **AWS**. With these, I want to explain how I entered the **Cloud System** and what I did.

First of all, let’s talk about **AWS**. **AWS, Amazon Web Service** is a platform that offers cloud-based services. **AWS** provides services in many areas such as data storage, data processing, data analysis, artificial intelligence and machine learning. **AWS** services make it easier to process and analyze large data sets, develop and deploy applications.

There are dozens of services you can access through **AWS**. I have added some titles below.



Since I will use **AWS** for data storage in my own project, I used **DynamoDB** service. **DynamoDB** is a high-performance and scalable **NoSQL** database. This database is used to efficiently store data and provide fast access.



AWS Access Key ID and AWS Secret Access Key are required to establish a connection with **AWS**. This information can be generated from **Security Credentials > Access Key.**

You need to enter this information from the terminal to establish the connection on your computer. Or, if you use **Python** in your project, you need to connect with the **boto3** library. **boto3** is an SDK for interacting with **AWS** services via **Python**. **boto3** allows you to work programmatically with **AWS** services and manage various **AWS** resources. Here is the part of my connection to send data to the **AWS** database:

```
import boto3

# AWS baglanti bilgileri
aws_access_key_id = 'hellomynameiskevin'
aws_secret_access_key = 'ihaveachangnesia'
region_name = 'us-east-1'

# AWS session olustur
session = boto3.Session(
    aws_access_key_id=aws_access_key_id,
    aws_secret_access_key=aws_secret_access_key,
    region_name=region_name
)

# DynamoDB kaynagi oluşturma
dynamodb = session.resource('dynamodb')
client = session.client('dynamodb')
```

The *boto3.Session* function creates a session with the given access keys and zone name. The session is used for all interactions with *AWS.session.resource(‘dynamodb’)* function returns the **DynamoDB** resource *and session.client(‘dynamodb’)* function returns the **DynamoDB** client. These objects are used to perform operations on **DynamoDB**.

After making the connection and accessing the database, I started throwing the data I had into the database. For this part, I created a text file with random names. I wrote a code to assign each data in this file as columns and to assign values between 0–100 to these columns every 5 seconds. The main part where this code was executed was as follows:

```
def main():
    # Read parameters
    current_directory = os.path.dirname(os.path.abspath(__file__))
    file_path = os.path.join(current_directory, 'items_parameters.txt')
    parameters = read_parameters_from_file(file_path)
    # Get next table name
    table_name = get_next_table_name()
    # Create table
    table = create_table(table_name)
```

```
    print(f"Table {table_name} created successfully")
    # Add data
    data_id = 1
    while True:
        add_data_to_table(table, data_id, parameters)
        print(f"Data added to table {table_name}")
        data_id += 1
        time.sleep(5)  # add data every 5 second
```

First of all, I started by reading my text file and making the names in it as parameters. Then I created the tables by adding 1 to the table if there is a table with this name in the database (*for example, if there is a table with this name, the new table name should be given as Veriler2*). Finally, I assigned an id and grew my tables by giving new values to these items every 5 seconds with a loop. By continuing this process as long as the code was running, I was able to continue uploading my data to the **AWS** database. Also, my *add_data_to_table* function I wrote for the values I added between 0–100 was as follows:

```
# Adding data
def add_data_to_table(table, data_id, parameters):
    data = {'dataId': data_id}
    for param in parameters:
        data[param] = random.randint(0, 100)
    table.put_item(Item=data)
```

Up to this point, I have created a connection between my application and **AWS** and managed to upload data to the **Cloud System** with a code. In the next step, I will try to pull this data and work on real-time data display.

See you in my next article.