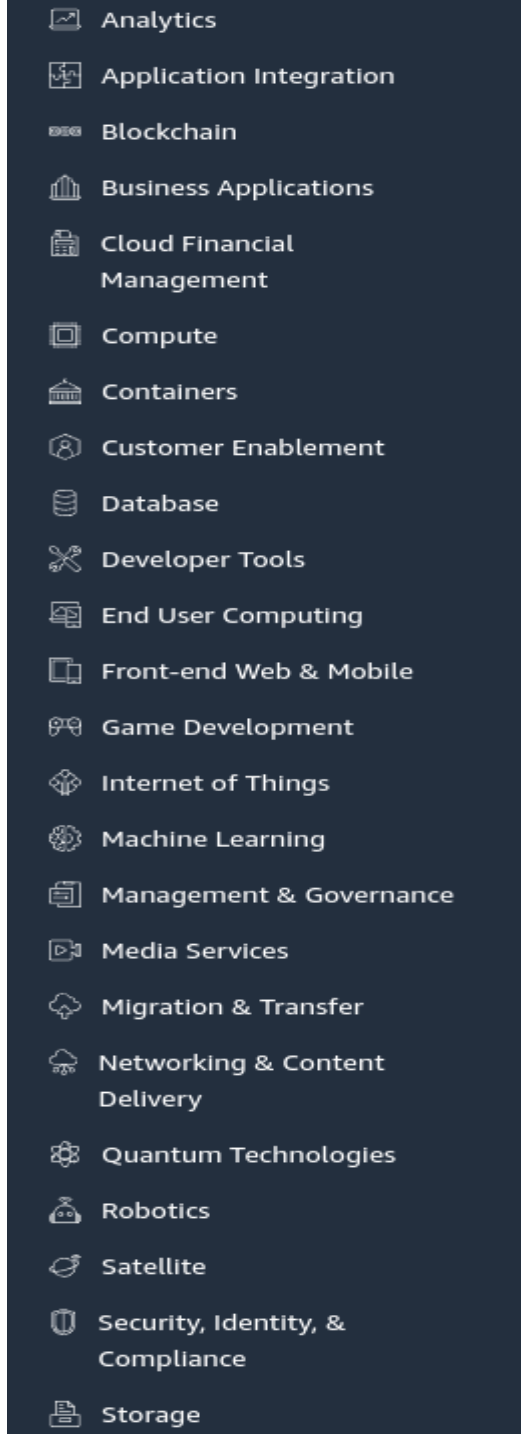


AWS ve DynamoDB ile Neler Yaptım?

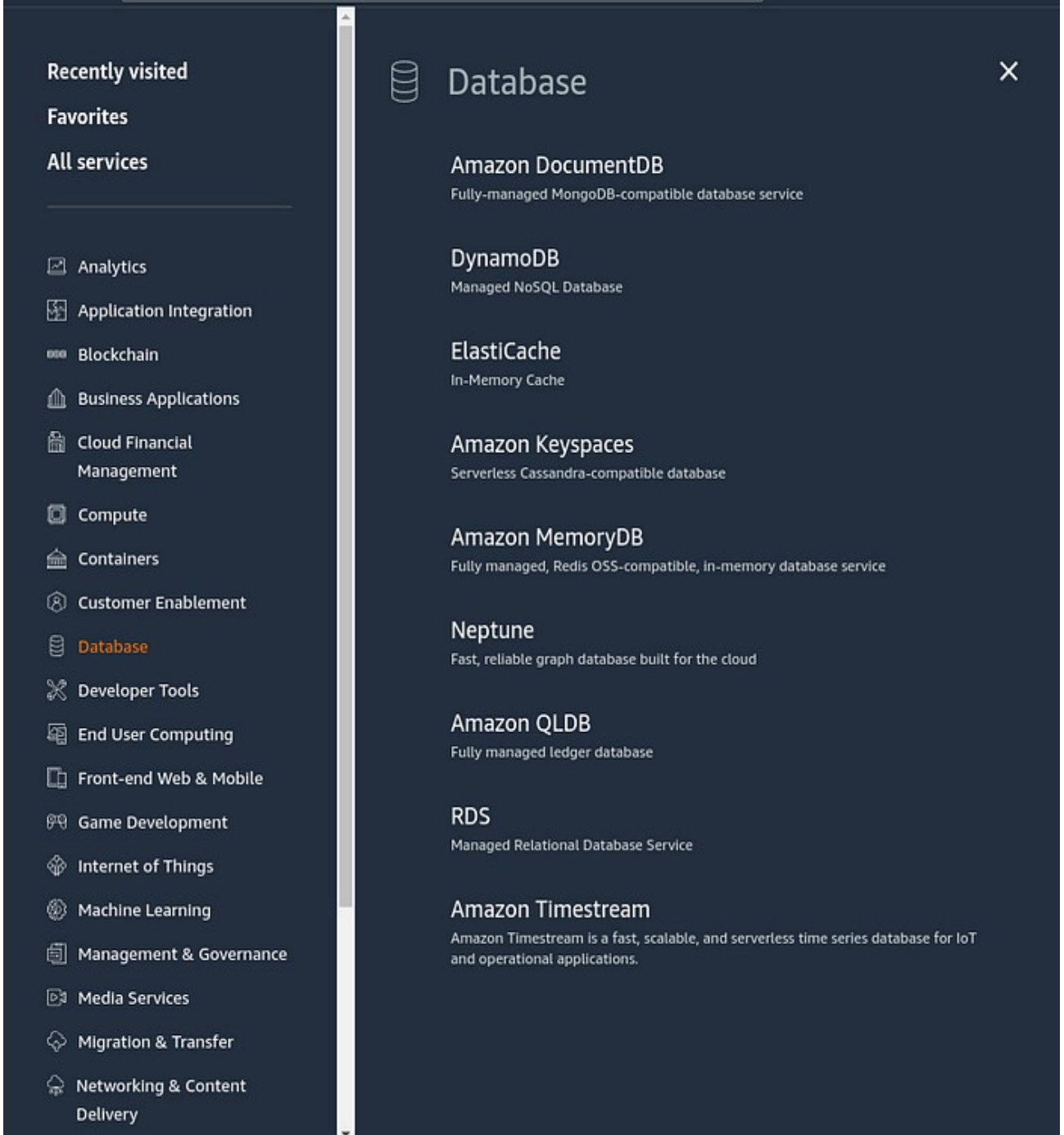
hi! Geliştirdiğim bir masaüstü uygulamasına **AWS** ile bazı eklemeler yaptım. Bunlarla birlikte **Bulut Sistemine** nasıl girdiğimi ve neler yaptığımı anlatmak istiyorum.

Öncelikle **AWS** yani Amazon Web Service'den bahsedelim. **AWS**, bulut tabanlı hizmetler sunan bir platformdur. **AWS**, veri depolama, veri işleme, veri analizi, yapay zeka ve makine öğrenimi gibi birçok alanda hizmet sağlar. **AWS**'nin sunduğu hizmetler sayesinde, büyük veri setlerini işlemek ve analiz etmek, uygulama geliştirmek ve dağıtmak kolaylaşır.

AWS üzerinden ulaşabileceğiniz onlarca servis mevcut. Bazı başlıkları aşağıya ekledim.



AWS'yi kendi projemde veri depolama amacıyla kullanacağım için **DynamoDB** servisinden yararlandım. **DynamoDB**, yüksek performanslı ve ölçeklenebilir bir **NoSQL** veritabanıdır. Bu veritabanı, verileri verimli bir şekilde depolamak ve hızlı erişim sağlamak için kullanılır.



AWS ile bağlantı kurmak için gerekli olan **AWS** erişim anahtarı (**AWS Access Key ID**) ve gizli anahtar (**AWS Secret Access Key**) bilgileri kullanmak gerekir. Bu bilgiler **Security Credentials** > **Access Key** kısmından oluşturulabilir.

Bağlantının bilgisayarınızda kurulması için terminalden bu bilgileri girmeniz gerekir. Ya da projenizde **Python** kullanırsanız **boto3** kütüphanesiyle bağlantı sağlamanız gerekir. **boto3**, **AWS** servisleri ile **Python** üzerinden etkileşim kurmak için kullanılan bir SDK'dır. **boto3**, **AWS** servisleri ile programatik olarak çalışmanıza ve çeşitli **AWS** kaynaklarını yönetmenize olanak tanır. Benim **AWS** veritabanına veri göndermek için bağlantı yaptığım kısmı şöyleydi:

```
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 kaynaklari olusturma
dynamodb = session.resource('dynamodb')
client = session.client('dynamodb')
```

`boto3.Session` fonksiyonu, verilen erişim anahtarları ve bölge adı ile bir oturum oluşturur. Oturum, **AWS** ile yapılacak tüm etkileşimlerde kullanılır. `session.resource('dynamodb')` fonksiyonu, **DynamoDB** kaynağını döner ve `session.client('dynamodb')` fonksiyonu ise **DynamoDB** istemcisini döner. Bu nesneler, **DynamoDB** üzerinde işlem yapmak için kullanılır.

Bağlantı yaptıktan ve veritabanına ulaştıktan sonra elimdeki verileri veritabanına atmaya başladım. Bu kısım için rastgele isimler olan bir text dosyası oluşturdum. Bu dosyadaki her bir veriyi sütun olarak ve her 5 saniyede bir bu sütunlara 0–100 arası değerler atacak şekilde bir kod yazdım. Bu kodun çalıştırıldığı ana kısım şöyleydi:

```
def main():
    # Parametreleri oku
    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)
    # Sonraki tablo adini al
    table_name = get_next_table_name()
    # Tablo olusturma
    table = create_table(table_name)
```

```
print(f"Table {table_name} created successfully")
# Verileri ekleme
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) # Her 5 saniyede bir veri ekle
```

Öncelikle text dosyamı okumak ve buradaki isimleri parametre yapmakla başladım. Sonrasında tabloları, eğer veritabanında bu isimde tablo varsa ona 1 ekleyerek (*örneğin Veriler1 tablosu varsa Veriler2 olarak yeni tablo ismi verilmeli*) yeni tabloyu oluşturdum. En sonunda da id ataması yaptım ve bir döngüyle her 5 saniyede bu itemlere yeni değerler vererek tablolarımı büyüttüm. Kod çalıştığı sürece bu işleme devam ederek **AWS** veritabanına verilerimi yüklemeye devam edebildim. Ayrıca 0–100 arası eklediğim değerler için yazdığım **add_data_to_table** fonksiyonum da şöyleydi:

```
# Veri ekleme
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)
```

Bu aşamaya kadar uygulamam ile **AWS** arasında bir bağlantı oluşturdum ve bir kod ile **Bulut Sistemine** veri yüklemeyi başardım. Sonraki aşamada bu verileri çekmeye çalışarak gerçek zamanlı veri gösterimi üzerinde çalışacağım.

Sonraki yazımda görüşmek üzere.