#19: Data Stores and Cloud Architectures

March 30, 2023 · Wade Fagen-Ulmschneider

Common Data Storage Options:

Scope	Data Storage	Interface	Technology
Local	**Variables**	Lang. Feature	All Programs!
	Files	File or File-like	open / fopen
Remote (Cloud)	**Object Storage**		S3 / MinIO
	Key-Value DB	Dictionary API set / get	redis
	Document DB	JSON find / update / insert	mongodb
	Relational DB	SQL	mysql / postgres
	Special Purpose	Objects	neo4j

The Illini "Coin Flip" Game Architecture:

Using Local Variable Storage for the Wallet Service:

19/wallet-service-local/app.py

```
 5 d = {}
 …
 8 def createUser(sessionID):
 9   d[sessionID] = 100
 …
30   d[sessionID] = userData["amount"]
```

Using a Key-Value Store (redits) for the Wallet Service:

19/wallet-service-kvstore/app.py

```
 5 kvStore = redis.Redis()
 …
 9 def createUser(sessionID):
10   kvStore.set(sessionID, 100)
 …
32   kvStore.set(sessionID, userData["amount"])
```

Using a Document Database (mongo) for the Wallet Service:

19/wallet-service-documentdb/app.py

```
 5 mongo = MongoClient(port=27017)
 6 db = mongo["IlliniCoin"]["users"]
 …
10 def createUser(sessionID):
11   userData = {"amount": 100, "sessionID": sessionID}
12   r = db.insert_one(userData)
 …
32   db.update_one({"sessionID": sessionID}, { "$set":
   {"amount": userData["amount"]}})
```

Cloud Architectures

Q: What are cloud architectures?

Three Primary Design Patterns for Cloud Architectures:

[1]:

[2]:

[3]:

Monolithic Software Architecture:
Characteristics:

A (Small) Monolithic Example: Illinois Open Source Queue
Endpoint: https://queue.illinois.edu/
Source Code: https://github.com/illinois/queue

Microservice Software Architecture:
Characteristics:

Microservice Example:
PiggyMetrics by Alexander Lukyanchikov
GitHub: https://github.com/sqshq/PiggyMetrics
via: https://github.com/davidetaibi/Microservices_Project_List

---

Serverless Software Architecture:
Characteristics:

Serverless Examples:

adoptable-pet-bot/serverless.yml
https://github.com/lynnaloo/adoptable-pet-bot/blob/master/serverless.yml

```
 6 provider:
 7   name: aws
 8   runtime: nodejs4.3
 9   region: us-east-1
 …
22 functions:
23   tweetPet:
24     handler: handlers/tweetPet.tweetPet
25     description: Tweets Adoptable Pets on a Schedule
26     memorySize: 512
27     timeout: 10
28     events:
29       - schedule: rate(6 hours)
```

emojibot/serverless.yaml
https://github.com/markhobson/emojibot/blob/master/serverless.yml

```
14 provider:
15   name: aws
16   region: eu-west-1
17   stage: dev
18   runtime: nodejs14.x
19   memorySize: 128
 …
24 functions:
25   event:
26     handler: src/handler.event
27     events:
28       - http: POST /event
29   explain:
30     handler: src/handler.explain
31     events:
32       - http: POST /explain
```

serverless-image-labeller/serverless.yml
https://github.com/nileshprasad137/serverless-image-labeller/blob/master/serverless.yml

```
 7 provider:
 8   name: aws
 9   logs:
10     restApi: true
11   runtime: python3.7
 …
48 functions:
49   labelOnS3Upload:
50     handler: handlers/S3UploadHandler.labelOnS3Upload
51     events:
52       - s3:
53           bucket: ${self:provider.environment.SERVERLESS_IMAGE_LABELLING_BUCKET}
54           event: s3:ObjectCreated:*
55           existing: true
56   getImagesByLabel:
57     handler: handlers/getImagesByLabelHandler.getImagesByLabel
58     events:
59       - http:
60           path: getImagesByLabel
61           method: post
62           cors: true
```