CSE-4095 Cloud Computing
HW10 - Step Functions

Objective

You will create an orchestration that processes simple banking transactions. Users can submit deposits and withdrawals via http POST operations. You will create a single DynamoDB table that contains the current balance for all accounts. Each incoming transaction will potentially update that table.

Instructions

- Create a DynamoDB table names HW10-Account-List.
  - account (PK) - account number (string)
  - balance - current balance in the account (numeric)
  - date-opened - the date the account was initially opened (string)
  - name - owner of the account (string)
- Populate the table with several records. This gives you a set of accounts to start with. (You do not have to write code for adding new accounts to the banking ledger.)
- Create a workflow to process incoming deposits and withdrawals. Each of these must be separate steps in the workflow. It is your decision when to perform these actions in your workflow.
  - Process the deposits
  - Process the withdrawals
  - Validate that the account number on the transaction is an existing account number that already exists in the DB. If not, then the transaction should fail.
  - Check that the account has enough available funds to process the withdrawal. If not, then the transaction should fail.
- Create an API Gateway that allows the user to submit POST operations that triggers your workflow. You must have at least one Method in your api. Your api must trigger your step function workflow. Use postman or curl to test your api.
- Create lambda functions for each step in the workflow.
- Use CloudWatch to log from as many of your AWS services as possible.

Summary of AWS Services

- DynamoDB table: HW10-Account-List
  - account (PK), balance, data-opened, name
- API Gateway - POST method to trigger your workflow
- Step Functions state machine
- Lambda functions for each step in the workflow

Other Details

- Git
  - Even if you do not use GitHub actions to deploy, you must place the python code for each lambda function in your github repo.
  - Create folder named "hw10" in your repo and store all your files in this folder.
  - Create branches for all your work. (Do not commit directly to "main".) Submit and complete pull requests when milestone progress is completed.
  - Use git add/commit/push to develop frequently. (This is a requirement, not an option.)

- Manual Testing
    - Use the following json examples to test your application. You can create others like these to fully test your application:
      {"account": "12345", "amount": 500, "type": "dep", "desc": "My paycheck" }
      {"account": "12345", "amount": 500, "type": "wtd", "desc": "Store purchase" }
    - You do not need to submit any proof of manual testing
- Automated Testing
    - You have been given a python testing routine in the starter code. Use it to test your workflow. This is how your TA will test to assess if your workflow is functional.
    - You do not need to submit any proof of testing,
- CICD
    - You are not required to create CICD pipeline for this assignment, but you may.

Tutorials

You may find these tutorials useful. But be careful! Tutorials do not always show you exactly what to do for your homework. But they are giving you pieces of what you will need to do. You will need to use discretion on which pieces of these tutorials you will follow.

- Step Functions Tutorial
- Reading from DynamoDB
- Data Flow Simulator (you must be logged into AWS before clicking on this link)

Debugging Tips

- If you receive the error "Missing Authentication Token" while testing your API Gateway endpoint, make sure you using the correct URL. For instance, your Stage URL will look something like this: "https://xxxxxxxxx.execute-api.us-east-1.amazonaws.com/prod". But if you add a Resource/Route for "/file" then you need to test the full URL like this: " https://xxxxxxxxx.execute-api.us-east-1.amazonaws.com/prod**/file**"

Extra credit opportunities

- 20 pts: Create a separate DynamoDB table that records all successful withdrawal and deposit transactions. Expand the starter code provided for testing to test that your transaction logging logic is functioning.

Deliverables

- Make sure to "TAG" all AWS resources you create for this assignment with "hw=10". Do not modify these resources on AWS in ANY way after the due date!
- Hand the following into Gradescope:
    - All code and files in your hw10 folder of your repo.
    - PDF containing:
        - Direct link to your git repo for HW10
        - List of all websites referenced for code

    - Design diagram of your solution. Show all the AWS components you created and how they interact with one another.
- Source file for creating your design diagram.

Grading Rubric

Step Function workflow	50
Lambda functions	20
DynamoDB Table	10
API Gateway	10
Design Diagram	10