Tech Assessment - 5 - Next for AWS service test strategy

Monday, August 18, 2025

4:21 PM

**Goals**

Prove the E2E end-to-end application and service setup works in AWS:

1. API /validate to check CSV file
2. API /upload saves a CSV to **S3**
   1. triggers **Glue** crawler to catalog the data
   2. Populate data and store into AWS **Athena** DB (DB: bills\_db, table: custom\_csv) and can be query

**Setup on AWS (I need more ramp up on it)**

* S3 bucket setup
* Glue crawler setup
* ATHENA DB setup
  + DB name: bills\_db, TABLE name: custom\_csv
* API server setup on AWS

**Account, Credentials, permissions**

* IAM, account, keys, secret access keys
* Region

**Test Approach Considerations:**

1. **Test types** 
   * **Automation regression test: /validate + CSV => /upload + CSV => /query SQL check**
   * **Smoke test, API calls + manual check: /validate + CSV => /upload + CSV => AWS check S3 bucket + ATHENA DB check data from CSV**
   * **Performance Test -** 
     + **Create automation performance test scripts based on E2E integration automation test cases**
     + **Verify system performance under load, multiple users**
     + **Performance KPIs: total calls, channels, pass/fail rate, throughput, cpu/memory usage, response time, latency, …**
   * **Security test - vulnerability scan if needed**
   * **Scalability test - optional, if system support large number of users required**

1. **Test target:** 
   1. AWS related service
      1. S3 Bucket, glue crawler, Athena DB
   2. Integrate AWS check with automation test suite
   3. Using boto3 AWS SDK for automation within QA Python test framework to interact and validation with related AWS service
      1. Add new automation test cases for AWS service validation

1. **Positive scenarios** 
   * **Basic API calls and validation:** 
     + **'/validate' -> '/upload' -> stores the CSV -> triggers Glue marker -> loads rows into DB -> SQL query, example: SELECT \* FROM custom\_csv;**
     + **S3 Bucket, glue crawler, Athena DB check to confirm results**
   * **Test CSV with empty, one row or multiple row data records**

1. **Negative scenarios** 
   1. Using similar API negative test scenarios:
      1. Missing column, missing data, wrong data type
      2. Empty CSV, non-exist CSV
   2. Verify focus on AWS service:
      1. AWS:
         1. S3 bucket, Athenna DB check (nothing break or related error message)
         2. No service crash or interrupted during negative tests
      2. API side:
         1. Verify API returned status code, such as: 422
         2. Verify return error type, such as: invaiidSchema invalidType, ..
2. **Test summary report**