**Task: Build a serverless, scalable, and secure backend system for a file parsing service using the following technologies:**

* Java, Javascript or Typescript for application logic, where Typescript is recommended
* AWS lambda
* S3 bucket for input and DynamoDB for storage
* AWS SQS for asynchronous data exchange
* AWS for infrastructure and deployment

The system should have the following functionality:

* Use any mechanism to place a text document in a S3 bucket. The text file will contain one or many vehicle brands, one on each row in the document.
* Placing the document on S3 will trigger a Lambda function, that will read the document, parse each line and post the content (a brand name) on a SQS queue. If the operation is successful, delete the input file. If the operation is unsuccessful produce a meaningful error output to the log. Empty file and unreadable/corrupt file will produce error logs.
* Another Lambda function will consume the SQS queue. The content of the message (a brand) will be stored in a database. The data should be structured according to DynamoDB best practices and stored in a way that allows for keeping track of how many times a certain type of brand have been registered. An info log entry will be outputted to CloudWatch for each new message, if the operation is successful. Identify potential errors that can occur and output these errors to the log.

To ensure security and scalability, the following requirements should be met:

* Use AWS CloudFormation, CDK or 3rdparty tools like SST to create the infrastructure.
* Use appropriate security best practices.
* Log transactions and any errors to CloudWatch
* Relevant coverage of unit and where applicable integration tests

The candidate should deliver the following:

* The source code of the microservices as github(or other git provider) repository(ies)
* A CloudFormation stack, orchestrating the ecosystem
* A README file with instructions on how to deploy the system
* A detailed explanation of the choices made in terms of design and architecture.
* A reasoning on test coverage

During the interview we will discuss how the candidate managed to:

* Write clean, maintainable, and well-documented code
* Use best practices for security, scalability, and performance
* Use AWS and other technologies effectively
* Provide clear and detailed explanations of the design and architecture decisions.

Example

|  |
| --- |
| Volvo  Volvo  Renault |

The file

= 3 messages on SQS

= 3 inserts/updates in DB

Desired output

* The DB structure should be designed in a way that facilitates queries of available brands (Volvo, Renault) and the quantity for a certain brand (Volvo: 2)
* For extra credit expose rest endpoints for the 2 queries exemplified above using serverless technologies.