Platform Engineering Role Code Challenge

Description

We would like to build a store of movie data that is accessible via a Restful API. The data is sourced from multiple data providers. All data providers are expected to use the same data format. Data cannot be retrieved again from provider and must be stored perpetually.

Data providers will send data as updates occur. We have chosen to use AWS S3 for providers to send their data. An update event will be for a single movie. Updates can be sent any time and there is no limit to how many updates can be sent.

Requirements

Candidates are asked to implement a solution as follows

- Build an API for querying data as workload in a Kubernetes cluster; data can be queried by year, movie name, cast member or genre
- API must be available 24/7 with zero downtime
- API must return response time within 5ms
- Choose a data storage solution that is fit for purpose
- Prepare a diagram describing the chosen architecture
- Implementation can be in any programming language
- Assume data format is that of the provided data set; see resources below.

The following is not required for implementation, but candidates must be ready to discuss a possible solution

- Implementing API in multiple regions
- Identifying the source of incoming API requests
- Audit changes to data
- Different providers accessing the S3 bucket to write data updates
- CICD of the full stack: infrastructure resources and application code
- Choice of data storage layer and factors that went into the decision
- Encrypting data at rest and in transit

Candidates have one week to provide a solution. Once ready, candidates should send a link to a GitHub project with all materials (code, manifest files, images … etc.) for review. If extra time is needed, please reach out to summer.hasani@jpmorgan.com or ross.w.millar@jpmorgan.com

Resources

Minikube for running K8 cluster locally https://minikube.sigs.k8s.io/docs/start/

Eksctl for creating K8 in AWS https://eksctl.io/

Movie data set https://github.com/prust/wikipedia-movie-data

For Java solutions, https://start.spring.io/