

## 1. Background

- a. List some of your favourite software/tools/SaaS to be used during development (e.g. Mac/Window/Ubuntu, VS Code/Vim/IntelliJ IDEA, GitHub/Bitbucket, Vercel, etc.).
- b. Describe a workflow for deploying code from development into production system.
- c. What would be the top 3 things you would check when doing a code review for a junior developer?

## 2. Design

Suppose you have an API service. The single source of truth for this system is a relational database (MySQL). For certain queries, you want to optimise query performance using a full-text search database, i.e. Elasticsearch.

The API server and MySQL are located in the same datacentre, but Elasticsearch we uses a hosted service (elastic.co), so network failures across datacenters need to be considered.

The current implementation is as follows: For some fields update in the MySQL (e.g. `title` of a `posts` table), the API would double the write to the Elasticsearch synchronously, so when there is a network issue the data between the MySQL and Elasticsearch will become inconsistent.

```
try:
    # Update MySQL
    cursor.execute("UPDATE posts SET title = 'test' WHERE id = 1")
    db.commit()

    # Update ES, when network issue happen the data will become inconsistent
    res = es.index(index="posts-index", id=1, document={'title': 'test'})

except Exception as err:
    print("Error when updating title")
```

If we allow the data update to Elasticsearch to become async with max delay = 1 to 2 mins, but we want the recovery process (re-sync) to happen fully automatically when network connectivity between data centres is restored. Can you propose a design support this requirement and illustrate with a diagram (no code required)?