Microservices with Node JS and React

# Fundamentals

## Monolithic server

A monolithic server contains all routing, middleware’s, business logic and database access requires to implement all features of the app.

|  |
| --- |
|  |

## Microservice

A single microservice contains all routing, middleware’s, business logic and database access required to implement one feature of the app.

|  |
| --- |
|  |

|  |
| --- |
|  |

**Advantages of microservices is that if some features/service inside of the application crashes a portion of the app is still going to work properly.**

## Challenges in Microservices.

Data Management between services:-

* Each service gets its own database (if it needs one).
* Services will never, ever reach into another services database

|  |
| --- |
|  |

**We want each service to run independently of other services because: -**

1. If we use single database, then if anything bad happened to this database all our services will crash immediately, and scaling is very challenging.

|  |
| --- |
|  |

### Communication between service

|  |
| --- |
|  |

#### Sync – Synchronization Communication

Services communicate with each other using direct requests.

|  |
| --- |
|  |

|  |
| --- |
|  |

#### Async – Asynchronous communication between services.

Services communicate with each other using events.

##### Method 1:- Event Bus without creating any new database for the

##### service-D

Here basically we have Event Bus where all the events flow from different services. Generally, we don’t use this in real world application and, also pros and cons is almost same as synchronous communication.

|  |
| --- |
|  |

##### Method 2: - Event Bus by creating any new database for the service D

Here we basically create a new database for the service which wants data from different services.

Every time the other service stores any data in database it will create a event which will pass to event Bus and then event BUS will send that event to that service to update the database.

|  |
| --- |
|  |

##### Question after Async Events

|  |
| --- |
|  |

##### Event Bus

|  |
| --- |
|  |

### Dealing with Missing Events

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

Option 3 is we are going to use.

## Running services with Docker

### Why Docker?

In Docker it helps to create a series of things called container. A Container is an isolated computing environment (it contains everything it required to run a single program).

|  |
| --- |
|  |

|  |
| --- |
|  |

### Why Kubernetes

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

Reason we use Kubernetes around microservices, it helps in communication between services in very easy and straight forward.