

Clusters and High Availability - Part 2

This lesson provides insight into fault tolerance.

We'll cover the following ^

- What is fault tolerance?

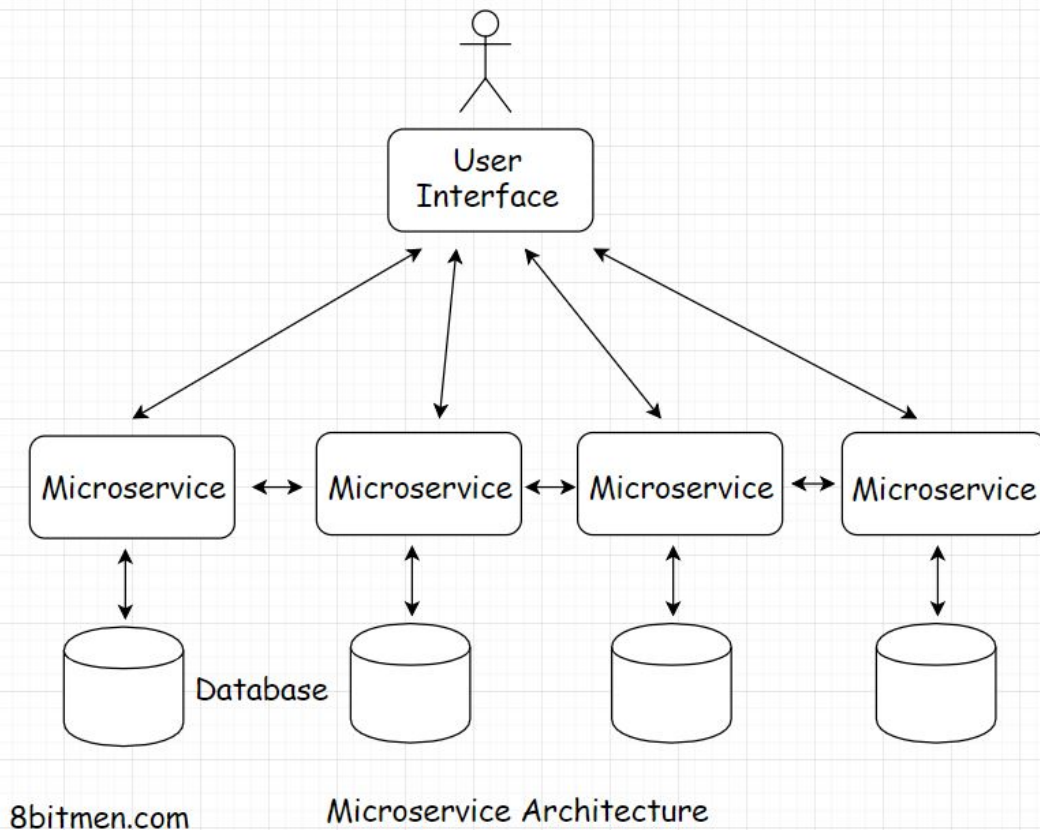
What is fault tolerance?

Fault tolerance is the ability of the system to stay up despite taking hits. A fault-tolerant system is equipped to handle faults. Being fault-tolerant is an essential element in designing life-critical systems.

A few instances, out of several, running the service go offline and bounce back all the time. In case of these internal failures, the system could work at a reduced level, but it will not go down entirely.

A very basic example of a system being fault-tolerant is a social networking application. In the case of backend node failures, a few services of the app such as image upload, post likes, etc. may stop working. However, the application as a whole will still be up. This approach is also known as *fail soft*.

At the application level, to achieve high availability and fault-tolerance, a big monolith service is architecturally broken down into smaller, loosely coupled services called microservices.



The upsides of splitting a big monolith into several microservices are:

- Easier management
- Easier development
- Ease of adding new features
- Ease of maintenance
- High availability and fault tolerance

Every microservice takes the onus of running different features of an application such as image upload, comment, instant messaging, user tagging, and so on. So, even if a few services go down, the application as a whole is still up.

In the next couple of lessons, we will discuss the *redundancy and replication* HA mechanisms.