MICROSERVICES

What is Microservice?

Robert C. Martin coined the term [single responsibility principle](https://en.wikipedia.org/wiki/Single_responsibility_principle) which states “gather together those things that change for the same reason, and separate those things that change for different reasons.”

A microservices architecture takes this same approach and extends it to the loosely coupled services which can be developed, deployed, and maintained independently. Each of these services is responsible for discrete task and can communicate with other services through simple APIs to solve a larger complex business problem.

Key Benefits :

* By the help of microservices, it is easy to create an application in to groups dividing discrete parts into groups so that they can be built by small teams. Services boundaries make it easier to scale up the development effort if need be.
* Once the whole application application is developed its easy to deployed them each other. And easy to identify the hot services and deploy them independently.
* In the case of error the whole application doesn’t crash as they are not tightly coupled and only that service affects and when the error fixed the service start functioning from where it lefts.
* One of the best benefit of microservice structure is we can use different languages and databases in different services/parts such they are loosely coupled so it doesn’t effect and make it more flexible.

Drawbacks of Monolithic Architecture:

1. All the services and parts of monolithic architecture are tightly coupled so it is very difficult to make or update it separately.
2. In case of error the whole application crashed due to high dependency of services over each other.
3. We can’t use different languages and databases for monolithic approach.
4. In case of any error rises the person who is solving has to understand whole code with each due to high dependency and then try to solve it.

Microservices VS Monolithic approaches:

