**1. What is Microservices Architecture?**

Microservices architecture allows avoiding monolith application for the large system. It provides loose coupling between collaborating processes which running independently in different environments with tight cohesion.

* ***Loose Coupling***– Effect of changes isolated
* ***Tight Cohesion***– Code perform a single well-defined task
* ***Loose Coupling***– Application build from collaboration services or processes, so any process change without affecting another process.
* ***Tight Cohesion***-An individual service or process that deals with a single view of data.

## **Microservices Benefits**

* The smaller code base is easy to maintain.
* Easy to scale as an individual component.
* Technology diversity i.e. we can mix libraries, databases, frameworks etc.
* Fault isolation i.e. a process failure should not bring the whole system down.
* Better support for smaller and parallel team.
* Independent deployment
* Deployment time reduce

## **Principles of Microservices**

There are the following principles of Microservices:

* Single Responsibility principle
* Modelled around business domain
* Isolate Failure
* Infrastructure automation
* Deploy independently

## **Disadvantages of Microservices**

* Microservices has all the associated complexities of the distributed system.
* There is a higher chance of failure during communication between different services.
* Difficult to manage a large number of services.
* The developer needs to solve the problem, such as network latency and load balancing.
* Complex testing over a distributed environment.