**What is MicroServices ?**

Microservices are an architectural approach to building applications that are distributed and loosely coupled so that a change in one microservice won't break the entire app.

Speciality of this architecture is that polyglot architecture is supported. For example, if a team is working on one of the microservice using Java, Spring Boot, and MySQL, another team can work on another microservice using Python, Node JS, and NoSQL.

Different microservices can use a different version of the same programming language.

Different microservices can use different programming languages.

Different microservices can use different architectures as well.

**Q)Why Microservices?**

* In the case of monolith applications, there are several problems like
* Same code base for presentation, business layer, and data access layer. Application is deployed as a single unit.
* Complex to maintain and scalability is an issue.
* Microservice solves the above problems.
* Microservices are ideal when a monolith or a legacy application needs to be modernized.
* For new software development, if the key business drivers are to reduce time to market, scalable better software, lower costs, faster development, or cloud-native development, microservices are ideal.
* Each service is independent and gives the flexibility to choose the programming language, database, and/or architecture.
* Distinct services can be developed, deployed, and maintained independently.

**Q) What are the pros and cons of Microservice Architecture?**

**Pros of Microservice Architecture**

1) Freedom to use different technologies

2) Each microservices focuses on single capability

3) Supports individual deployable units

4) Allow frequent software releases

5) Ensures security of each service

6) Multiple services are parallelly developed and deployed

**Cons of Microservice Architecture**

1) Management of a large number of services is difficult.

2) Communication between microservices is complex.

3) Increased efforts for configuration and other operations

4) Difficult to maintain transaction safety and data boundaries

5) Due to the decentralized nature of microservices, more microservices will mean more resources hence high Investment

6) Debugging of problems is harder unless the right instrumentation is followed during design and development.

7) Microservices will need a large team size with the right mix of experience in design, development, automation, deployments, tools, and testing.