**Microservices design pattern :**

1. Service discovery pattern
2. Circuit breaker pattern
3. External configuration

**Microservices : Advantages :**

1. Scalability of services
2. Faster deployment
3. Ease of understanding the application
4. Eliminate technology lock
5. Choosing vendor is independent
6. Fault tolerance

**Disadvantage :**

1. Communication between services
2. More resource is required since more services & different technology can be involved in each service
3. Testing is tough
4. Maintaining logs is not easier
5. Debugging is also difficult

Q. How to maintain different version in Microservices ?

Ans: - 1. Through API gateway 2. URL based versioning 3. Header driven versioning

Q. Why Kafka ?

Ans :- Kafka is often used in real-time streaming data architecture to provide real-time analytics. Since Kafka is fast & **scalable** fault tolerant messaging system. Kafka is used where JMS, RabbitMQ may not considered due to large volume. Records written to Kafka topics are persisted to disk and replicated to other servers for fault tolerance. Kafka is popular because of it’s exceptional performance. Kafka relies on OS karnel to move data quickly.

Q. system design for table booking app.

Ans :-

Restaurant Service –

Payment Service –

Notification Service –

User – id, name, location, mob

Restaurant – id, name, location, rating, tag, availability, hours of operation

Payment – user Id, name, type

Table – id, restaurant id, seats

Reservation – Id, Restaurant id, user id, date/time, booking size

Seating Arrangement – table id, reservation id

User -> table (1 to many)

User -> Reservation (1 to many)

User -> Restaurant (1-1)

Restaurant -> Table (1-many)

Table -> seats (1-many)