Microservices Interview Questions

1)What is Microservices?

- A Services must run Independent & it should be Re-usable, implemented using less coding called as "Microservices";
- Every Re-usable Services can be converted to Microservices.

2)What are the Microservices Advantages ?

- Load Balancing.
- Parallel Coding / Testing / Maintenance.
- Easy to scale as individual component.
- Technology diversity i.e. we can mix libraries, databases, frameworks etc.
- Fault isolation i.e. a process failure should not bring whole system down.
- Independent deployment
- Deployment time reduce

3)What are the Microservices Disadvantages?

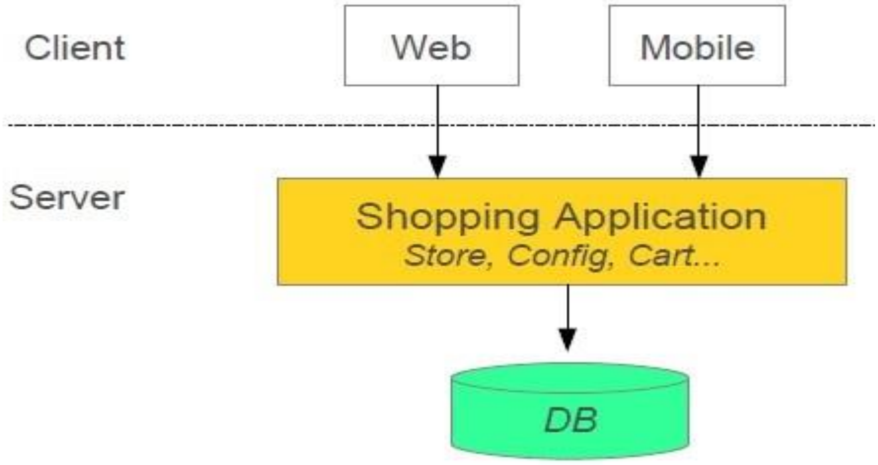
- Difficult to achieve strong consistency across services
- ACID transactions do not span multiple processes.
- Distributed System so hard to debug and trace the issues
- Greater need for end to end testing
- Required cultural changes in across teams like Dev and Ops working together even in same team.

4)What is Microservices Architecture?

- Microservices architecture allows to avoid monolith application for large system. It provide loose coupling between collaborating processes which running independently in different environments with tight cohesion.
- For example imagine an online shop with separate microservices for user-accounts, product-catalog order-processing and shopping carts. So these components are inevitably important for such a large online shopping portal. For online shopping system we could use following architectures.

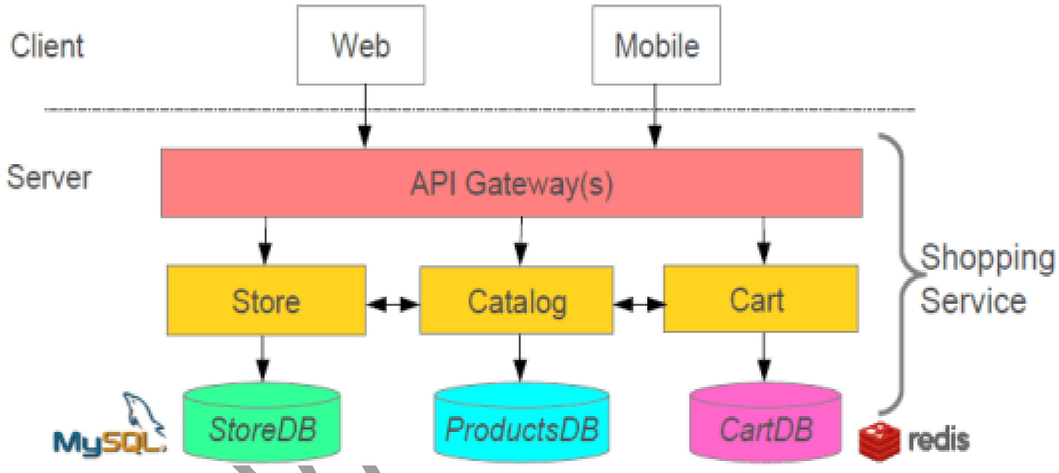
Shopping system without Microservices (Monolith architecture)

In this architecture we are using Monolith architecture i.e. all collaborating components combine all in one application.



Shopping system with Microservices:

In this architecture style the main application divided in a set of sub applications called microservices. One large Application divided into multiple collaborating processes as below.



5)How will you monitor multiple microservices for various indicators like health?

- Spring Boot provides actuator endpoints to monitor metrics of individual microservices. These endpoints are very helpful for getting information about applications like if they are up, if their components like database etc are working good. But a major drawback or difficulty about using actuator endpoints is that we have to individually hit the endpoints for applications to know their status or health. Imagine microservices involving 50 applications, the admin will have to hit the actuator endpoints of all 50 applications. To help us deal with this situation, we will be using open source project i.e

Spring Boot Admin (Admin UI For Monitoring Microservices)
Built on top of Spring Boot Actuator, it provides a web UI to enable us visualize the metrics of multiple applications.

6)What does one mean by Service Registration and Discovery ?

- When we start a project, we usally have all the configurations in the properties file. As more and more services are developed and deployed, adding and modifying these properties become more complex. Some services might go down, while some the location might change. This manual changing of properties may create issues. Eureka Service Registration and Discovery helps in such scenarios. As all services are registered to the Eureka server and lookup done by calling the Eureka Server, any change in service locations need not be handled

7) What does one mean by Load Balancing ?

- In computing, load balancing improves the distribution of workloads across multiple computing resources, such as computers, a computer cluster, network links, central processing units, or disk drives. Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any single resource. Using multiple components with load balancing instead of a single component may increase reliability and availability through redundancy. Load balancing usually involves dedicated software or hardware, such as a multilayer switch or a Domain Name System server process.

8) How to achieve server side load balancing using Spring Cloud?

- Server side load balancingcan be achieved using Netflix Zuul.
  Zuul is a JVM based router and server side load balancer by Netflix.
  It provides a single entry to our system, which allows a browser, mobile app, or other user interface to consume services from multiple hosts without managing cross-origin resource sharing (CORS) and authentication for each one. We can integrate Zuul with other Netflix projects like Hystrix for fault tolerance and Eureka for service discovery, or use it to manage routing rules, filters, and load balancing across your system.

9) What Is Eureka?

- Eureka is the Netflix Service Discovery Server and Client. Eureka Server is using Spring Cloud

Facebook Group: https://www.facebook.com/groups/thejavatemple