|  |  |
| --- | --- |
|  |  |
| Challenges with Microservices | 1. Bounded context (How to decide the boundary) 2. Configuration Management (So many environments) 3. Dynamically scale up and down (Dynamic load balancing) 4. Logging, fault tolerance 5. Pack of card , one on the other |
| Microservices - Advantages of Microservices | 1. Microservices Built using different technologies 2. Dynamic scaling (load vary) 3. Faster release cycle. |
| Introduction to Spring Cloud  Provided solution to changes. | 1. Configuration management: Cloud config server 2. Dynamic load balancing : Ribbon Client side load balancing 3. Service Discovery : Eureka 4. REST client for REST calls 5. Monitoring : Zipkin distributed tracing 6. Netflix API Gateway Zuul 7. Fault tolerance: Hystrix |
| Differences between Microservices and SOA | 1. SOA independent services but not independently deployable component. 2. SOA One centralized system with multiple services |
| Monolith Architecture |  |
| Microservice Architecture |  |
|  | @SpringBootApplication (Starting point of Application)  @EnableEurekaServer (Discovery Server)  @EnableEurekaClient  @EnableCircuitBreaker  @HystrixCommand to the method  @EnableHystixDashboard  Eureka default port:8761  //You can see all the services.  REST Template for rest calls. With timeout. Solve when service call is slow. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |