



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

## 第三版：SpringCloud 25 道

### 什么是 Spring Cloud?

Spring cloud 流应用程序启动器是基于 SpringBoot 的 Spring 集成应用程序，提供与外部系统的集成。Spring cloud Task，一个生命周期短暂的微服务框架，用于快速构建执行有限数据处理的应用程序。

### 使用 Spring Cloud 有什么优势?

使用 SpringBoot 开发分布式微服务时，我们面临以下问题

- 1、与分布式系统相关的复杂性-这种开销包括网络问题，延迟开销，带宽问题，安全问题。
- 2、服务发现-服务发现工具管理群集中的流程和服务如何查找和互相交谈。它涉及一个服务目录，在该目录中注册服务，然后能够查找并连接到该目录中的服务。
- 3、冗余-分布式系统中的冗余问题。
- 4、负载均衡 --负载均衡改善跨多个计算资源的工作负荷，诸如计算机，计算机集群，网络链路，中央处理单元，或磁盘驱动器的分布。
- 5、性能-问题 由于各种运营开销导致的性能问题。
- 6、部署复杂性-Devops 技能的要求。

### 服务注册和发现是什么意思？Spring Cloud 如何实现？

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

当我们开始一个项目时，我们通常在属性文件中进行所有的配置。随着越来越多的服务开发和部署，添加和修改这些属性变得更加复杂。有些服务可能会下降，而某些位置可能会发生变化。手动更改属性可能会产生问题。Eureka 服务注册和发现可以在这种情况下提供帮助。由于所有服务都在 Eureka 服务器上注册并通过调用 Eureka 服务器完成查找，因此无需处理服务地点的任何更改和处理。

## 负载均衡的意义什么？

在计算中，负载均衡可以改善跨计算机，计算机集群，网络链接，中央处理单元或磁盘驱动器等多种计算资源的工作负载分布。负载均衡旨在优化资源使用，最大化吞吐量，最小化响应时间并避免任何单一资源的过载。使用多个组件进行负载均衡而不是单个组件可能会通过冗余来提高可靠性和可用性。负载均衡通常涉及专用软件或硬件，例如多层交换机或域名系统服务器进程。

## 什么是 Hystrix？它如何实现容错？

Hystrix 是一个延迟和容错库，旨在隔离远程系统，服务和第三方库的访问点，当出现故障是不可避免的故障时，停止级联故障并在复杂的分布式系统中实现弹性。

通常对于使用微服务架构开发的系统，涉及到许多微服务。这些微服务彼此协作。

思考以下微服务

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



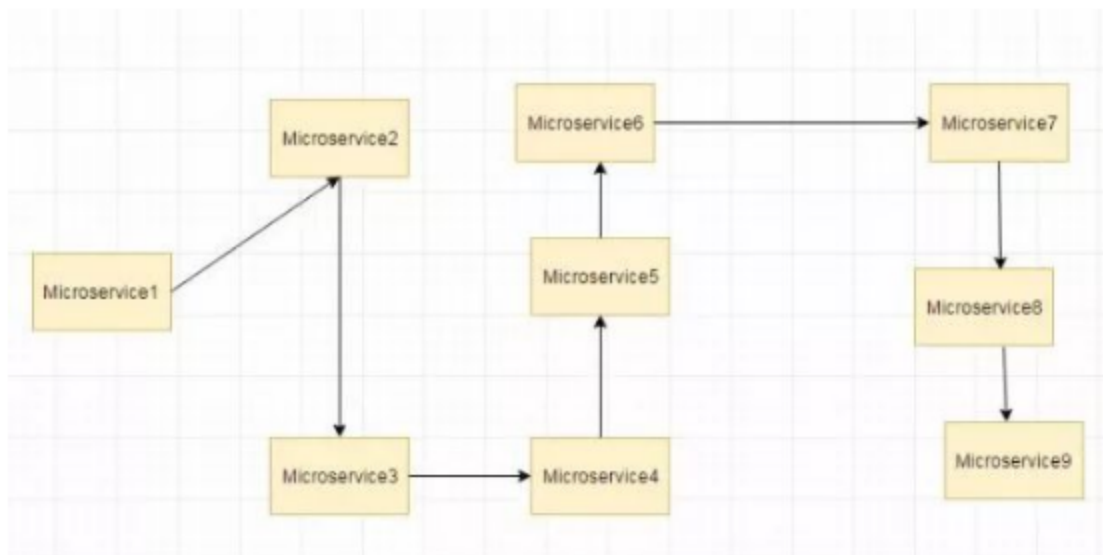
微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题



假设如果上图中的微服务 9 失败了，那么使用传统方法我们将传播一个异常。但这仍然会导致整个系统崩溃。

随着微服务数量的增加，这个问题变得更加复杂。微服务的数量可以高达 1000。这是 hystrix 出现的地方 我们将使用 Hystrix 在这种情况下的 Fallback 方法功能。我们有两个服务 employee-consumer 使用由 employee-producer 公开的服务。

简化图如下所示



现在假设由于某种原因，employee-producer 公开的服务会抛出异常。我们在这种情况下使用 Hystrix 定义了一个回退方法。这种后备方法应该具有与公开服务相同的返回类型。如果暴露服务中出现异常，则回退方法将返回一些值。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

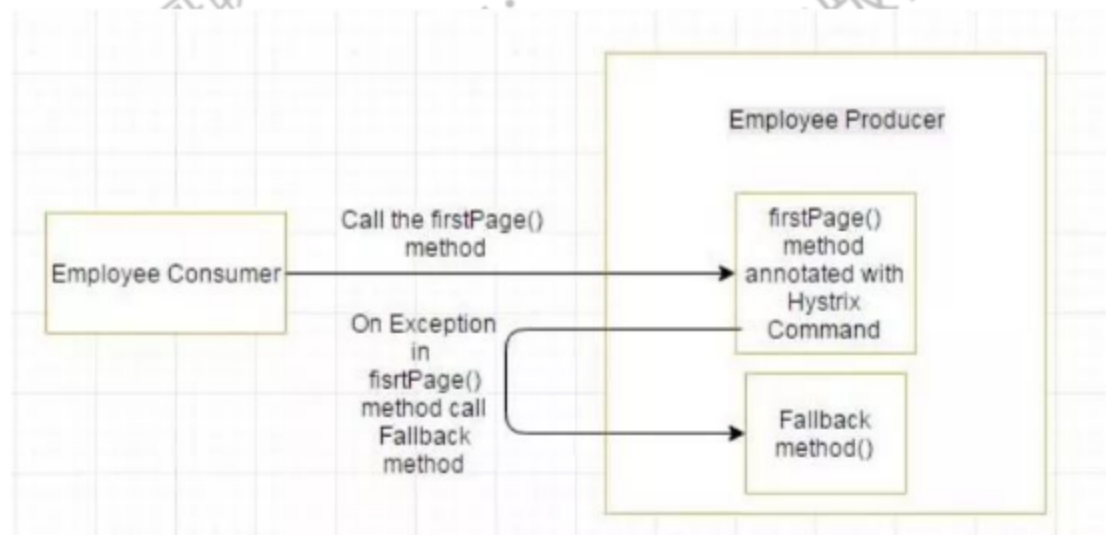
扫码关注



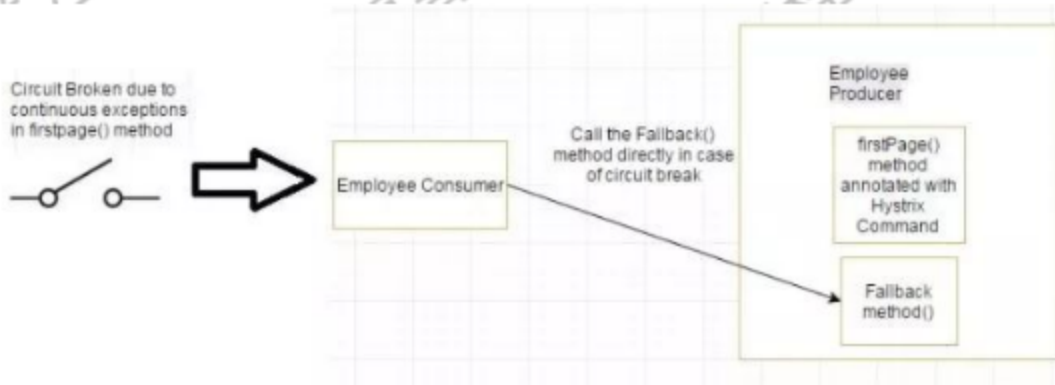
回复：面试题 获取最新版面试题

## 什么是 Hystrix 断路器？我们需要它吗？

由于某些原因，employee-consumer 公开服务会引发异常。在这种情况下使用 Hystrix 我们定义了一个回退方法。如果在公开服务中发生异常，则回退方法返回一些默认值。



如果 firstPage method() 中的异常继续发生，则 Hystrix 电路将中断，并且员工使用者将一起跳过 firstPage 方法，并直接调用回退方法。断路器的目的是给第一页方法或第一页方法可能调用的其他方法留出时间，并导致异常恢复。可能发生的情况是，在负载较小的情况下，导致异常的问题有更好的恢复机会。



关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

## 什么是 Netflix Feign? 它的优点是什么?

Feign 是受到 Retrofit, JAXRS-2.0 和 WebSocket 启发的 java 客户端联编程序。Feign 的第一个目标是将约束分母的复杂性统一到 http apis, 而不考虑其稳定性。在 employee-consumer 的例子中, 我们使用了 employee-producer 使用 REST 模板公开的 REST 服务。

但是我们必须编写大量代码才能执行以下步骤

- 1、使用功能区进行负载均衡。
- 2、获取服务实例, 然后获取基本 URL。
- 3、利用 REST 模板来使用服务。 前面的代码如下

```
@Controller
public class ConsumerControllerClient {

    @Autowired
    private LoadBalancerClient loadBalancer;

    public void getEmployee() throws RestClientException, IOException {

        ServiceInstance
        serviceInstance=loadBalancer.choose("employee-producer");

        System.out.println(serviceInstance.getUri());

        String baseUrl=serviceInstance.getUri().toString();
```

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

```
baseUrl=baseUrl+"/employee";

RestTemplate restTemplate = new RestTemplate();
ResponseEntity<String> response=null;
try{
response=restTemplate.exchange(baseUrl,
        HttpMethod.GET, getHeaders(),String.class);
}catch (Exception ex)
{
    System.out.println(ex);
}
System.out.println(response.getBody());
```

之前的代码，有像 NullPointer 这样的例外的机会，并不是最优的。我们将看到如何使用 Netflix Feign 使呼叫变得更加轻松和清洁。如果 Netflix Ribbon 依赖关系也在类路径中，那么 Feign 默认也会负责负载均衡。

## 什么是 Spring Cloud Bus? 我们需要它吗?

考虑以下情况：我们多个应用程序使用 Spring Cloud Config 读取属性，而 Spring Cloud Config 从 GIT 读取这些属性。

下面的例子中多个员工生产者模块从 Employee Config Module 获取 Eureka 注册的财产。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



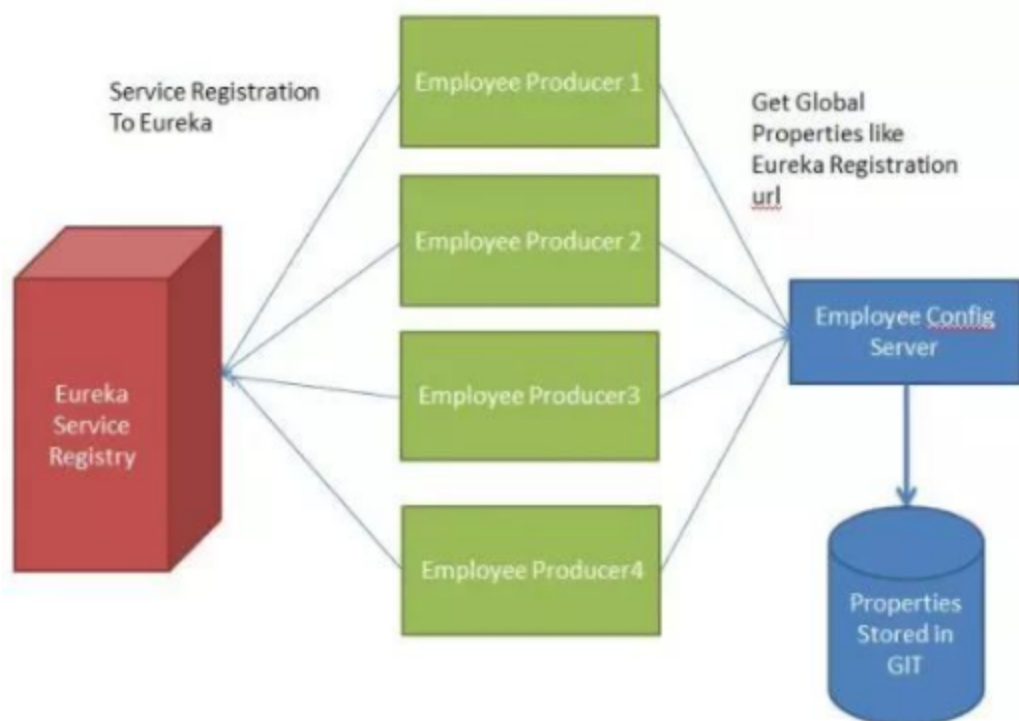
微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题



如果假设 GIT 中的 Eureka 注册属性更改为指向另一台 Eureka 服务器，会发生什么情况。在这种情况下，我们将不得不重新启动服务以获取更新的属性。

还有另一种使用执行器端点/刷新的方式。但是我们将不得不为每个模块单独调用这个 url。例如，如果 Employee Producer1 部署在端口 8080 上，则调用 `http://localhost:8080/refresh`。同样对于 Employee Producer2 `http://localhost:8081/refresh` 等等。这又很麻烦。这就是 Spring Cloud Bus 发挥作用的地方。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



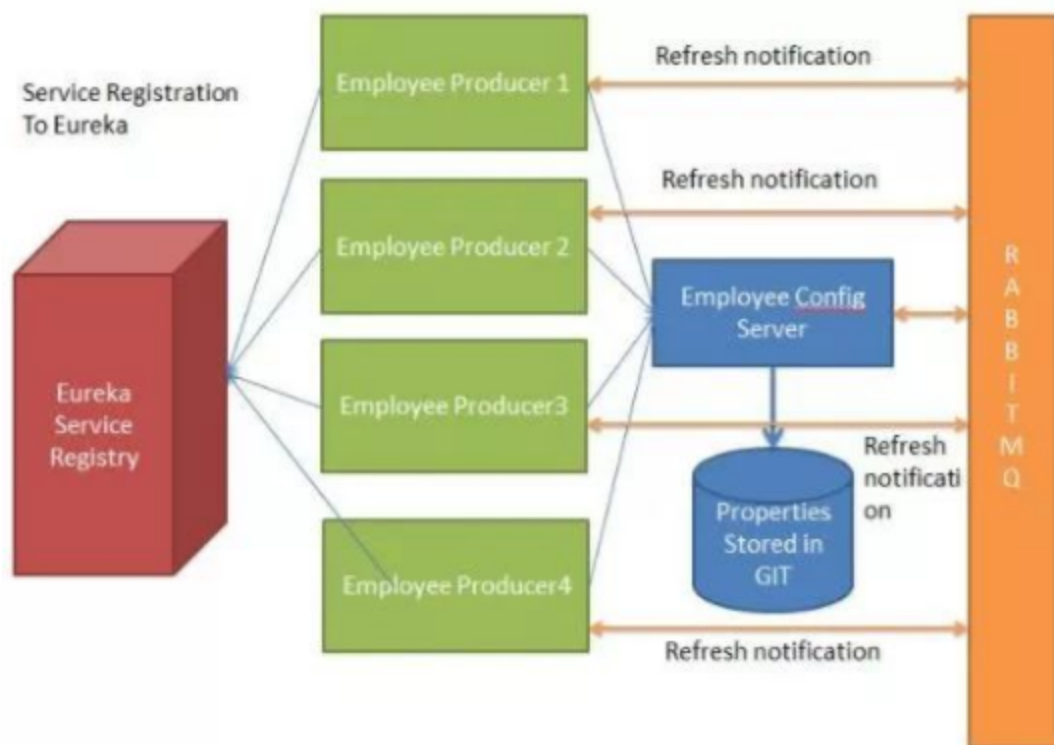
微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题



Spring Cloud Bus 提供了跨多个实例刷新配置的功能。因此，在上面的示例中，如果我们刷新 Employee Producer 1，则会自动刷新所有其他必需的模块。如果我们有多个微服务启动并运行，这特别有用。这是通过将所有微服务连接到单个消息代理来实现的。无论何时刷新实例，此事件都会订阅到侦听此代理的所有微服务，并且它们也会刷新。可以通过使用端点/总线/刷新来实现对任何单个实例的刷新。

## Spring Cloud Config

集中配置管理工具，分布式系统中统一的外部配置管理，默认使用 Git 来存储配置，可以支持客户端配置的刷新及加密、解密操作。

## Spring Cloud Netflix

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题





微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

Netflix OSS 开源组件集成，包括 Eureka、Hystrix、Ribbon、Feign、Zuul 等核心组件。

- 1、 Eureka：服务治理组件，包括服务端的注册中心和客户端的服务发现机制；
- 2、 Ribbon：负载均衡的服务调用组件，具有多种负载均衡调用策略；
- 3、 Hystrix：服务容错组件，实现了断路器模式，为依赖服务的出错和延迟提供了容错能力；
- 4、 Feign：基于 Ribbon 和 Hystrix 的声明式服务调用组件；
- 5、 Zuul：API 网关组件，对请求提供路由及过滤功能。

## Spring Cloud Bus

用于传播集群状态变化的消息总线，使用轻量级消息代理链接分布式系统中的节点，可以用来动态刷新集群中的服务配置。

## Spring Cloud Consul

基于 Hashicorp Consul 的服务治理组件。

## Spring Cloud Security

安全工具包，对 Zuul 代理中的负载均衡 OAuth2 客户端及登录认证进行支持。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

## Spring Cloud Sleuth

Spring Cloud 应用程序的分布式请求链路跟踪，支持使用 Zipkin、HTrace 和基于日志（例如 ELK）的跟踪。

## Spring Cloud Stream

轻量级事件驱动微服务框架，可以使用简单的声明式模型来发送及接收消息，主要实现为 Apache Kafka 及 RabbitMQ。

## Spring Cloud Task

用于快速构建短暂、有限数据处理任务的微服务框架，用于向应用中添加功能性和非功能性的特性。

## Spring Cloud Zookeeper

基于 Apache Zookeeper 的服务治理组件。

## Spring Cloud Gateway

API 网关组件，对请求提供路由及过滤功能。

## Spring Cloud OpenFeign

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

基于 Ribbon 和 Hystrix 的声明式服务调用组件，可以动态创建基于 Spring MVC 注解的接口实现用于服务调用，在 Spring Cloud 2.0 中已经取代 Feign 成为了一等公民。

Spring Cloud 的版本关系

Spring Cloud 是一个由许多子项目组成的综合项目，各子项目有不同的发布节奏。为了管理 Spring Cloud 与各子项目的版本依赖关系，发布了一个清单，其中包括了某个 Spring Cloud 版本对应的子项目版本。

为了避免 Spring Cloud 版本号与子项目版本号混淆，Spring Cloud 版本采用了名称而非版本号的命名，这些版本的名字采用了伦敦地铁站的名字，根据字母表的顺序来对应版本时间顺序，例如 Angel 是第一个版本，Brixton 是第二个版本。

当 Spring Cloud 的发布内容积累到临界点或者一个重大 BUG 被解决后，会发布一个“service releases”版本，简称 SRX 版本，比如 Greenwich.SR2 就是 Spring Cloud 发布的 Greenwich 版本的第 2 个 SRX 版本。目前 Spring Cloud 的最新版本是 Hoxton。

## 微服务之间如何独立通讯的？

同步通信：dobbo 通过 RPC 远程过程调用、springcloud 通过 REST 接口 json 调用 等。

异步：消息队列，如：RabbitMq、ActiveM、Kafka 等。

**eureka 和 zookeeper 都可以提供服务注册与发现的功能，请**

**说说两个的区别？**

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

- 1、 zookeeper 是 CP 原则，强一致性和分区容错性。
- 2、 eureka 是 AP 原则 可用性和分区容错性。
- 3、 zookeeper 当主节点故障时，zk 会在剩余节点重新选择主节点，耗时过长，虽然最终能够恢复，但是选取主节点期间会导致服务不可用，这是不能容忍的。
- 4、 eureka 各个节点是平等的，一个节点挂掉，其他节点仍会正常保证服务。

## SpringCloud 和 Dubbo 有哪些区别?

首先，他们都是分布式管理框架。

dubbo 是二进制传输，占用带宽会少一点。SpringCloud 是 http 传输，带宽会多一点，同时使用 http 协议一般会使用 JSON 报文，消耗会更大。

dubbo 开发难度较大，所依赖的 jar 包有很多问题大型工程无法解决。

SpringCloud 对第三方的继承可以一键式生成，天然集成。

SpringCloud 接口协议约定比较松散，需要强有力的行政措施来限制接口无序升级。

最大的区别：

**Spring Cloud 抛弃了 Dubbo 的 RPC 通信，采用的是基于**

**HTTP 的 REST 方式。**

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

严格来说，这两种方式各有优劣。虽然在一定程度上来说，后者牺牲了服务调用的性能，但也避免了上面提到的原生 RPC 带来的问题。而且 REST 相比 RPC 更为灵活，服务提供方和调用方的依赖只依靠一纸契约，不存在代码级别的强依赖，这在强调快速演化的微服务环境下，显得更为合适。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题