# Consul服务注册与发现

# 搭建consul集群

## 准备三台服务器，

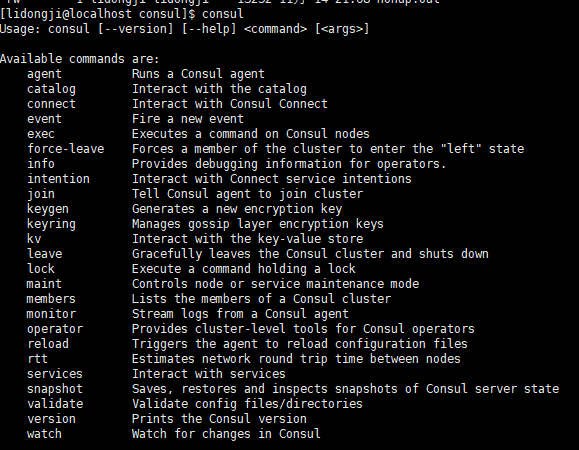
192.168.75.229 ，192.168.75.219，192.168.75.198

## 2.下载安装consul



(1).解压缩后执行以下命令：sudo scp consul /usr/bin/。

(2).然后测试启动执行 consul 命令出现如下画面说明安装成功：



1. .将consul部署在上面三台服务器上。
2. .启动集群步骤和命令

在IP为node0节点上执行命令nohup consul agent -data-dir /tmp/node0 -node=node0 -bind=192.168.75.229 -datacenter=dc1 -ui -client=192.168.75.229 -server -bootstrap-expect 1 & 启动node0节点。以服务端模式启动，开启远程访问

在IP为node1节点上执行命令nohup consul agent -data-dir /tmp/node1 -node=node1 -bind=192.168.75.219 -datacenter=dc1 -ui & 启动node1节点。以客户端模式启动，没有开启远程访问

在IP为node2节点上执行命令nohup consul agent -data-dir /tmp/node2 -node=node2 -bind=192.168.75.198 -datacenter=dc1 -ui -client=192.168.75.198 & 启动node2节点。以客户端模式启动，开启远程访问

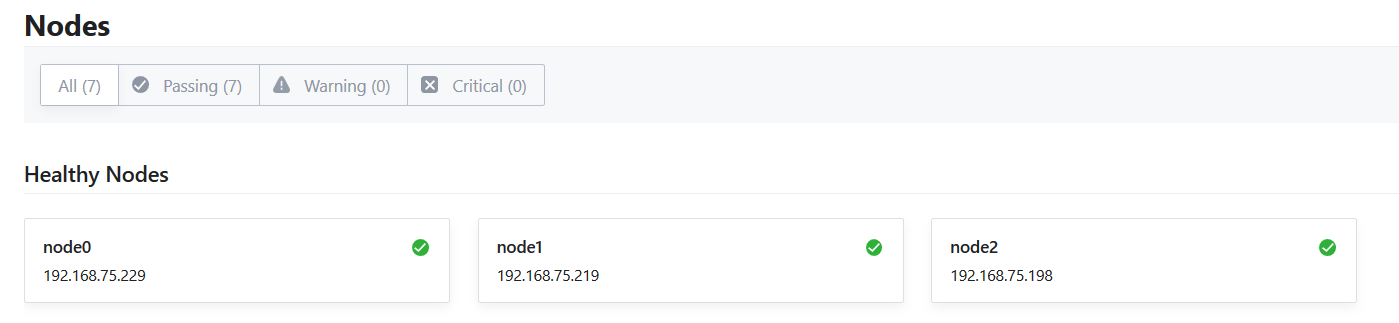
在node1节点上执行命令：consul join 192.168.75.229将node1节点加入到node0上。



在node2节点上执行命令：consul join -http-addr=192.168.75.198:8500 192.168.75.229 将node2节点加入node0上



这样集群就搭建完成。访问http://192.168.75.229:8500/ui/



# 创建一个微服务注册到consul集群中

## 1.consul服务发现特性

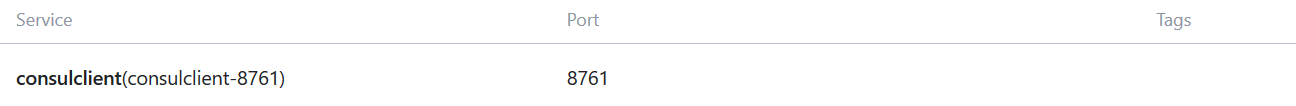
1. 创建一个springboot工程
2. 添加如下maven依赖

<parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>1.5.9.RELEASE</version>  
 <relativePath/> <!-- lookup parent from repository -->  
</parent>  
<dependencyManagement>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-dependencies</artifactId>  
 <version>Dalston.SR5</version>  
 <type>pom</type>  
 <scope>import</scope>  
 <exclusions>  
 </exclusions>  
 </dependency>  
 </dependencies>  
</dependencyManagement>  
  
<properties>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>  
 <java.version>1.8</java.version>  
</properties>  
  
<dependencies>  
 <!--consul依赖-->  
 <dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-consul-discovery</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
</dependencies>  
  
<build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
</build>

1. 配置文件application.yml

**spring:  
 cloud:  
 consul:  
 host:** 192.168.75.198 *#consul集群的node2节点* **port:** 8500 *# 节点远程访问端口号* **discovery:***# hostname: localhost  
# instance-id: ${spring.cloud.consul.discovery.hostname}:${server.port}* **health-check-path:** /health *#健康检查路径* **health-check-interval:** 15s *#健康检查频率* **application:  
 name:** consulclient  
**server:  
 port:** 8762

启动微服务就会将其注册到consul中



## 支持Ribbon实现负载均衡特性

1. 上述微服务修改端口号server.port=8762,启动微服务将其注册到consul集群中
2. 这两个微服务中添加如下代码

@Value("${server.port}")  
private String port;  
@RequestMapping("index")  
public String home() {  
 return "Hello world" + port;  
}

1. 创建一个微服务工程名字应用名字feign消费者
2. 添加pom依赖

<parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>1.5.9.RELEASE</version>  
 <relativePath/> <!-- lookup parent from repository -->  
</parent>  
<dependencyManagement>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-dependencies</artifactId>  
 <version>Edgware.SR3</version>  
 <type>pom</type>  
 <scope>import</scope>  
 <exclusions>  
 </exclusions>  
 </dependency>  
 </dependencies>  
</dependencyManagement>  
  
<properties>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>  
 <java.version>1.8</java.version>  
</properties>  
  
<dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
 <!--consul依赖-->  
 <dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-consul-discovery</artifactId>  
 </dependency>  
 <!--熔断器依赖-->  
 <dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-hystrix</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-hystrix-dashboard</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-actuator</artifactId>  
 </dependency>  
 <!--消费者依赖-->  
 <dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-feign</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
</dependencies>

1. 配置文件

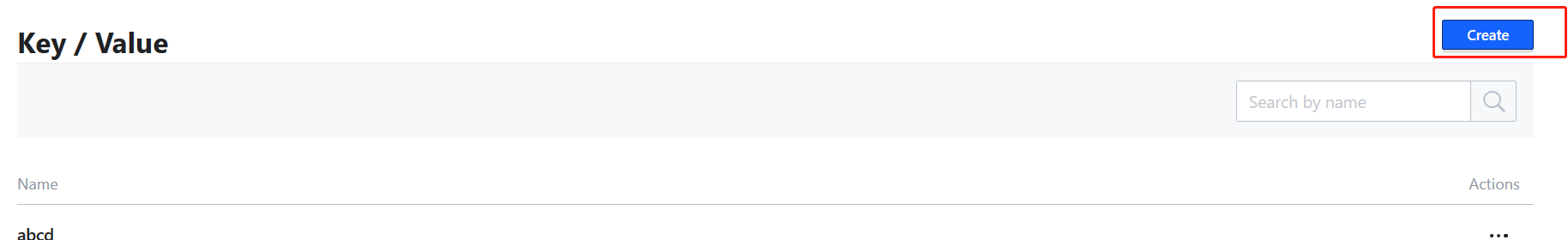
**server:  
 port:** 8769  
**spring:  
 application:  
 name:** feign2  
 **cloud:  
 consul:  
 host:** 192.168.75.198  
 **port:** 8500  
 **discovery:  
 health-check-path:** /health  
 **health-check-interval:** 15s  
*#开启熔断器***feign:  
 hystrix:  
 enabled:** true  
**logging.level.com.dongji.consulserver.service.ApiService:** DEBUG

1. 添加如下代码消费者调用两个名字相同（consulclient）的微服务

@FeignClient(value = "consulclient",fallback = ApiServerError.class,configuration = FooConfiguration.class)  
public interface ApiService {  
 @RequestMapping(value = "/index",method = RequestMethod.*GET*)  
 String index();  
}

启动服务访问消费者接口http://192.168.75.219/index发现它轮训调用了consulclient服务。

# 支持存储key/value



可以在界面上创建，也可以通过命令创建。