### 介绍

Consul是HashiCrop公司所推出的开源工具，官网介绍：

Service discovery and configuration made easy（简化服务发现和配置），Distributed, highly available, and datacenter-aware（分布式，高可用，多数据中心）

## 主要特性

* 服务发现

Consul简化了服务注册和发现，可以使用DNS或者HTTP接口。也可以注册外部服务比如SaaS提供商。

Consul的客户端可以提供服务，例如API或者mysql，而其他客户端可以使用consul来发现给定服务的提供者。通过使用DNS或者HTTP，应用可以轻易的找到他们依赖的服务。

* 失败检测

和服务发现配套的健康检查阻止将请求路由到不健康的主机并使得服务易于提供熔断器。

Consul客户端可以提供任意数量的健康检查，可以和给定服务关联（webserver是否返回200OK），或者和本地节点关联（内存使用率是否低于90%）。这个信息可以被操作者使用来监控集群健康，也用于服务发现组件路由请求远离不健康站点。

* 多数据中心

Consul很容易扩展到多个数据中心，不需要复杂的配置。可以在其他数据中心查找服务，或者保持请求本地。支持多数据中心意味着consul的用户不必担心构建额外的抽象层来扩展到多个地域。

* 键值对存储

灵活的键值对存储，用于动态配置，特性标记，协调，leader选举等。支持Long poll用于配置变更的准实时通知。应用可以将consul的分层键值对存储用于任何目的。简单的HTTP API易于使用。



## Consul的主要架构

Consul是分布式的高可用系统。

每个提供服务给consul的节点都运行有一个consul的agent。运行agent对于发现其他服务或者获取/设置键值数据并非必须。agent负责做节点上的服务的健康检查也包括节点自己。

agent和一个或者多个consul server进行交互。数据存储并同步在consul server上。服务器自行选举一个leader。虽然consul可以单机工作，推荐3到5台consul以避免失败场景导致数据丢失。推荐每个数据中心一个consul服务器集群。

需要发现其他服务或者节点的基础设施的组件可以查询任何consul服务器或者consul agent。agent自动将请求转发给服务器。

每个数据中心运行consul服务器的一个集群。当跨数据中心的服务发现或者配置请求发生时，本地consul服务器转发请求到远程数据中心并返回结果。

## 安装Consul服务端

Consul和Eureka不同，Eureka只需要在项目中加入服务端依赖，就可以作为服务端使用；Consul需要下载并单独安装。

1. 下载地址：[https://www.consul.io/downloads.html](https://links.jianshu.com/go?to=https%3A%2F%2Fwww.consul.io%2Fdownloads.html" \t "_blank)
2. 上传到服务器
3. unzip consul\_1.6.2\_darwin\_amd64.zip
4. consul -v查看版本

## 启动Consul服务端

### 1、开发模式启动单节点

1. consul agent -dev -ui -client 0.0.0.0启动命令，添加红色部分才能被外网访问，并有web页面。
2. <http://localhost:8500>打开管理页面。
3. curl localhost:8500/v1/catalog/nodes查看consul节点。

### 2、集群模式启动

1. 创建4台虚拟机
2. 启动集群，三个service和一个client

consul agent -server -bootstrap-expect 3 -data-dir /tmp/consul –

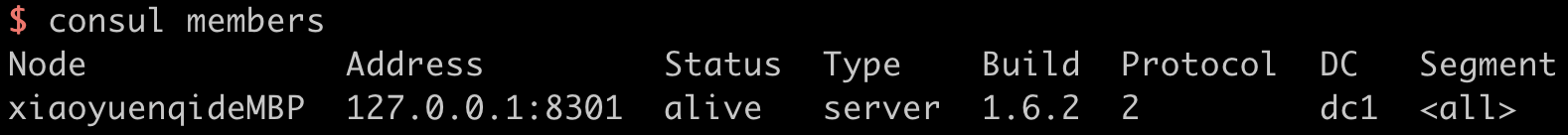
node=c1 -ui -client 0.0.0.0 -bind 192.168.1.149 -join 192.168.1.149

consul agent -server -bootstrap-expect 3 -data-dir /tmp/consul -node=c2 -ui -client 0.0.0.0 -bind 192.168.1.151 -join 192.168.1.149

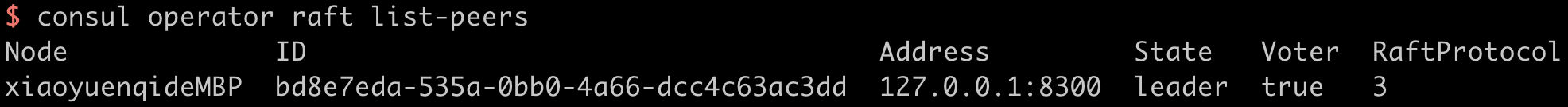
consul agent -server -bootstrap-expect 3 -data-dir /tmp/consul -node=c3 -ui -client 0.0.0.0 -bind 192.168.1.152 -join 192.168.1.149

consul agent -data-dir /tmp/consul -node=c4 -ui -client 0.0.0.0 -bind 192.168.1.116 -join 192.168.1.109

1. 查看集群成员consul members



1. 查看成员角色consul operator raft list-peers



各参数含义：

1. agent：Consul的核心命令,主要作用有维护成员信息、运行状态检测、声明以及处理请求等。
2. -server：代表server模式。
3. -ui：开启web控制台。
4. -bootstrap-expect：代表想要创建的集群数目，官方建议3或5。
5. -data-dir：数据存储目录。
6. -node：当前node的名称。
7. -client：一个客户端服务注册的地址，可以和当前server的一致也可以是其他主机地址，系统默认是127.0.0.1。
8. -bind：集群通讯地址。
9. -join：加入的集群地址。

## 服务注册

### 1、通过HTTP API注册服务

1. 注册一个ID为“test001”，name为“test”的服务：

curl -X PUT -d '{"id": "test001","name": "test","address": "127.0.0.1","port": 8080,"tags": ["dev"]}' <http://127.0.0.1:8500/v1/agent/service/register>

1. 查看服务是否注册成功

[curl http://localhost:8500/v1/catalog/service/test](curl%20http:/localhost:8500/v1/catalog/service/test) 查看服务信息

[curl http://localhost:8500/v1/health/service/test?passing](curl%20http:/localhost:8500/v1/health/service/test?passing) 健康检查

[http://localhost:8500](http://localhost:8500/)打开管理页面查看已注册的服务

### 2、通过项目注册服务

1）添加依赖：

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-consul-discovery</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-actuator-autoconfigure</artifactId>

</dependency>

2）配置文件：

spring:

application:

name: consul-client

cloud:

consul:

host: 127.0.0.0

port: 8500

discovery:

register: true

service-name: ${spring.application.name}

tags: dev

health-check-path: /actuator/health

ip-address: 192.168.1.107

prefer-ip-address: true

management:

endpoints:

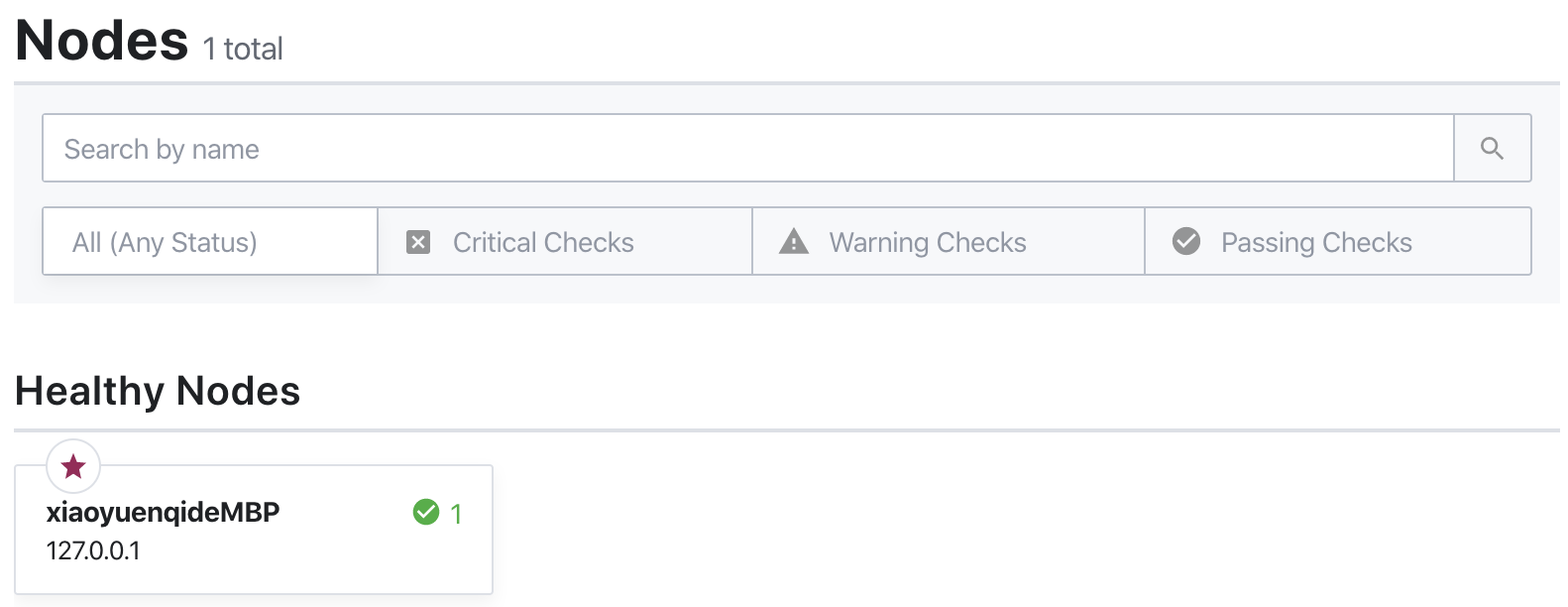
web:

exposure:

include: "\*"

3）启动项目

4）<http://127.0.0.1:8500>打开管理页面查看已注册的服务



5）防火墙需要关闭，否则service check访问不通。

6）ip-address：127.0.0.1一定要有，为本机地址，否则找不到本机。

7）prefer-ip-address:true一定要开启

## 停止Agent

使用Ctrl+C关闭Agent，中断Agent之后可以看到他离开集群并关闭。

在退出中，Consul提醒其他集群成员，这个节点离开了。如果你强行杀掉进程。集群的其他成员应该能检测到这个节点失效了。当一个成员离开，它的服务和检测也会从目录中移除。当一个成员失效了，它的健康状况被简单的标记为危险，但是不会从目录中移除。Consul会自动尝试对失效的节点进行重连。允许他从某些网络条件下回复过来。离开的节点则不会再继续联系。

## 更新服务

服务定义可以通过配置文件并发送SIGHUP会给agent来进行更新，这样可以让你在不关闭服务或者保持服务请求可用的情况下进行更新。

consul reload

另外HTTP API可以用来动态的添加，移除和修改服务。