服务注册中心比较

Feature	Consul	zookeeper	etcd	euerka
服务健康检查	服务状态，内存，硬盘等	(弱)长连接，keepalive	连接心跳	可配支持
多数据中心	支持	—	—	—
kv存储服务	支持	支持	支持	—
一致性	raft	paxos	raft	—
cap	ca	cp	cp	ap
使用接口(多语言能力)	支持http和dns	客户端	http/grpc	http（sidecar）
watch支持	全量/支持long polling	支持	支持 long polling	支持 long polling/ 大部分增量
自身监控	metrics	—	metrics	metrics
安全	acl /https	acl	https支持（弱）	—
spring cloud集成	已支持	已支持	已支持	已支持

> 表格来源
> https://blog.csdn.net/u010963948/article/details/71730165

解读一下，CAP中，作为服务发现服务，一定要具备高可用性，所以一定要具有A，才能保证这个重要服务的可用性，因此直接排除就剩Consul和Eureka

由于eureka 2.0闭源，所以这里我们使用consul来作为服务注册发现的服务

部署Consul集群服务

> 官方文档地址
> https://www.consul.io/docs/agent/basics.html

consul的启动方式有两种：

1.直接 consul agent + 配置参数 命令启动；

启动命令如下：

```
consul agent -server -ui -bootstrap-expect=3 -data-dir=/tmp/consul -node=server1 -client=0.0.0.0 -bind=192.168.125.118
```

2.使用配置参数文件启动，json格式。

参数文件配置：

Example Configuration File

```
{
 "datacenter": "east-aws",
 "data_dir": "/opt/consul",
 "log_level": "INFO",
 "node_name": "foobar",
 "server": true,
 "watches": [
 {
 "type": "checks",
  "handler": "/usr/bin/health-check-handler.sh"
  }
  ],
  "telemetry": {
  "statsite_address": "127.0.0.1:2180"
  }
}
```

启动命令如下：

```
consul agent -config-file consul_config.json -ui
```

> 所有配置参数见官网文档：
> https://www.consul.io/docs/agent/options.html

下面是服务运行中可以不中断，就能重新加载的配置参数

Reloadable Configuration

Reloading configuration does not reload all configuration items. The items which are reloaded include:

- Log level
- Checks
- Services
- Watches
- HTTP Client Address
- Node Metadata
- Metric Prefix Filter

- Discard Check Output
- RPC rate limiting

我自己定义的配置文件内容如下：

```
{
 "datacenter": "consul-cluster",
 "node_name": "consul-01",
 "bind_addr": "0.0.0.0",
 "client_addr": "0.0.0.0",
 "advertise_addr": "192.168.100.150",
 "bootstrap_expect": 3,
 "server": true,
 "data_dir": "/home/ryzc/springcloud/consul/data",
  "http_config": {
  "response_headers": {
  "Access-Control-Allow-Origin": "*"
  }
  },
  "log_level": "INFO",
  "enable_syslog": true,
  "ports": {
  "http": 8500,
  "dns": 8600,
  "server": 8300,
  "serf_lan": 8301,
  "serf_wan": 8302
  },
  "enable_script_checks": true
}

```

bind_addr一定要设置为0.0.0.0，这样虚拟机之间才能通信

同时， 必须设置 "advertise_addr": "192.168.100.150"，来标记本地IP

否则会报错，bind_addr和advertise_addr必须有一个要标记本地IP的

3.执行命令启动第一个consul服务

```
➜ consul consul agent -config-file config/consul_config_01.json -ui
bootstrap_expect > 0: expecting 3 servers
==> Starting Consul agent...
```

```
==> Consul agent running!
 Version: 'v1.4.0'
 Node ID: '2f7ea62d-a773-e094-ce09-062038ab6e97'
 Node name: 'consul-01'
 Datacenter: 'consul-cluster' (Segment: '<all>')
 Server: true (Bootstrap: false)
  Client Addr: [0.0.0.0] (HTTP: 8500, HTTPS: -1, gRPC: -1, DNS: 8600)
  Cluster Addr: 192.168.100.150 (LAN: 8301, WAN: 8302)
  Encrypt: Gossip: false, TLS-Outgoing: false, TLS-Incoming: false

==> Log data will now stream in as it occurs:

  2019/01/15 10:57:47 [INFO] raft: Initial configuration (index=0): []
  2019/01/15 10:57:47 [INFO] raft: Node at 192.168.100.150:8300 [Follower] entering Follower state (Leader: "")
  2019/01/15 10:57:47 [INFO] serf: EventMemberJoin: consul-01.consul-cluster 192.168.100.150
  2019/01/15 10:57:47 [INFO] serf: EventMemberJoin: consul-01 192.168.100.150
  2019/01/15 10:57:47 [INFO] consul: Adding LAN server consul-01 (Addr: tcp/192.168.100.150:8300) (DC: consul-cluster)
  2019/01/15 10:57:47 [INFO] consul: Handled member-join event for server "consul-01.consul-cluster" in area "wan"
  2019/01/15 10:57:47 [INFO] agent: Started DNS server 0.0.0.0:8600 (tcp)
  2019/01/15 10:57:47 [INFO] agent: Started DNS server 0.0.0.0:8600 (udp)
  2019/01/15 10:57:47 [INFO] agent: Started HTTP server on [::]:8500 (tcp)
  2019/01/15 10:57:47 [INFO] agent: started state syncer
  2019/01/15 10:57:53 [WARN] raft: no known peers, aborting election
  2019/01/15 10:57:54 [ERR] agent: failed to sync remote state: No cluster leader
```

然后尝试只修改端口号，启动2、3两个服务，但是启动不了

所以搭建集群，就需要使用不同机器、不同IP，即需要启动虚拟机

4.搭建虚拟机

> 虚拟机构建参考：
> 虚拟机.note

然后与第一个consul服务相同操作，在两个虚拟机上分别启动两个consul服务

5.启动成功后，把s2、s3服务添加到consul服务1上

执行命令：

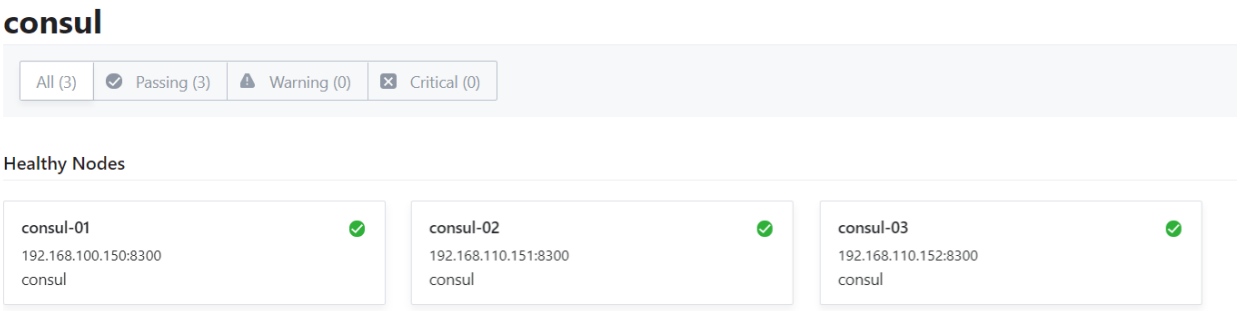
```
./consul join -http-addr http://192.168.110.151:8500 192.168.100.150
Successfully joined cluster by contacting 1 nodes.

./consul join -http-addr http://192.168.110.152:8500 192.168.100.150
Successfully joined cluster by contacting 1 nodes.
```

6.检查集群

```
ubuntu@s2:~/springcloud/consul$ ./consul members
Node Address Status Type Build Protocol DC Segment
consul-01 192.168.100.150:8301 alive server 1.4.0 2 consul-cluster <all>
consul-02 192.168.110.151:8301 alive server 1.4.0 2 consul-cluster <all>
consul-03 192.168.110.152:8301 alive server 1.4.0 2 consul-cluster <all>
```

7.UI界面查看



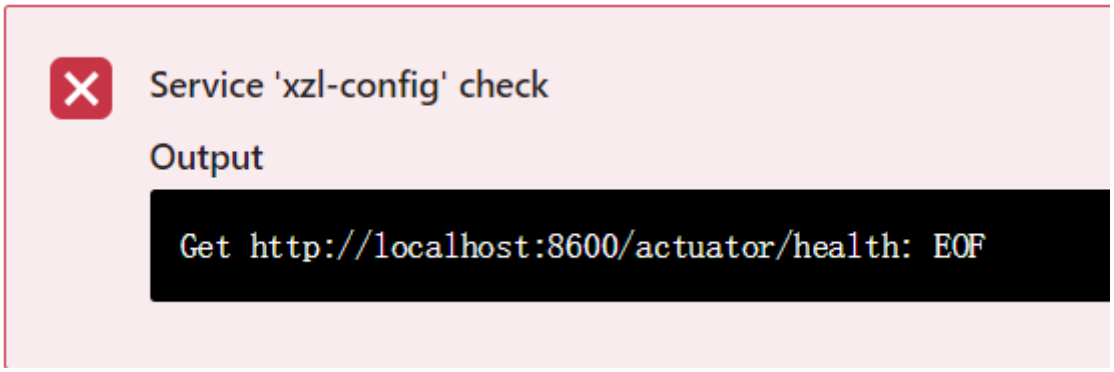
集群部署完成！

注册在consul上的服务需要手动删除

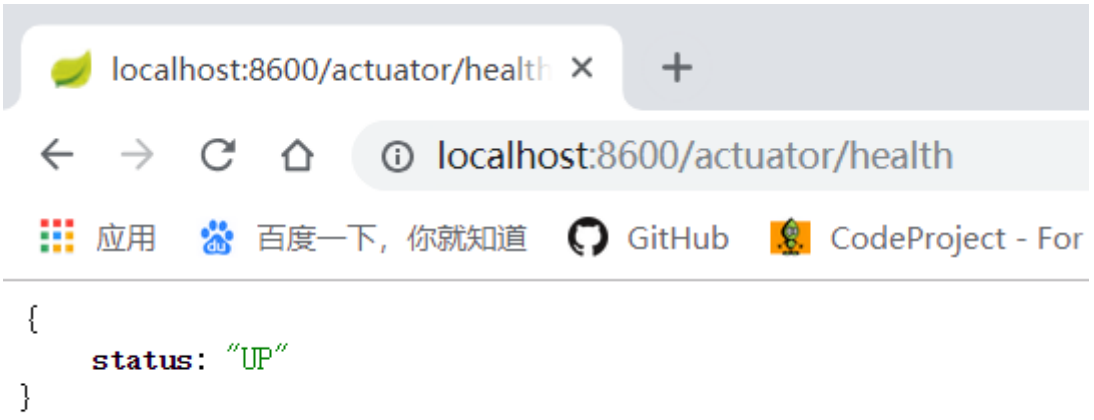
执行下面：

```
PUT http://{IP}:{PORT}/v1/agent/service/deregister/{SERVICE-ID}
```

服务注册到consul上面，总是失败，显示



直接方位这个health链接也没问题啊



临时解决方案：

暂时先关闭所有注册服务的健康检查开关，在配置文件中配置如下：

```
spring:
 cloud:
 consul:
 discovery:
 register-health-check: false
```

Consul端口占用情况

```
Ports Used
Consul requires up to 6 different ports to work properly, some on TCP, UDP,
  or both protocols. Below we document the requirements for each port.
Server RPC (Default 8300). This is used by servers to handle incoming reque
  sts from other agents. TCP only.
Serf LAN (Default 8301). This is used to handle gossip in the LAN. Required
  by all agents. TCP and UDP.
Serf WAN (Default 8302). This is used by servers to gossip over the WAN, to
  other servers. TCP and UDP. As of Consul 0.8 the WAN join flooding featur
```

```
  e requires the Serf WAN port (TCP/UDP) to be listening on both WAN and LA
  N interfaces. See also: Consul 0.8.0 CHANGELOG and GH-3058
HTTP API (Default 8500). This is used by clients to talk to the HTTP API. T
  CP only.
DNS Interface (Default 8600). Used to resolve DNS queries. TCP and UDP.
```

Ports Used

Consul requires up to 6 different ports to work properly, some on TCP, UDP, or both protocols. Below we document the requirements for each port.

- Server RPC (Default 8300). This is used by servers to handle incoming requests from other agents. TCP only.
- Serf LAN (Default 8301). This is used to handle gossip in the LAN. Required by all agents. TCP and UDP.
- Serf WAN (Default 8302). This is used by servers to gossip over the WAN, to other servers. TCP and UDP. As of Consul 0.8 the WAN join flooding feature requires the Serf WAN port (TCP/UDP) to be listening on both WAN and LAN interfaces. See also: Consul 0.8.0 CHANGELOG and GH-3058
- HTTP API (Default 8500). This is used by clients to talk to the HTTP API. TCP only.
- DNS Interface (Default 8600). Used to resolve DNS queries. TCP and UDP.