

服务发现 Consul

服务发现

由于线上生产环境尤其容器部署情况下服务实例地址都是 **动态分配** 的，因此不能配置文件写死地址。
为了解决服务地址查询的问题，只能在运行时通过服务发现组件解析 **服务名/psm** 来获取服务实例地址和端口。
常见的服务发现框架有 `Zookeeper`、`Etcd`、`Consul`。公司的服务发现是基于 **Consul** 实现的。

关于 Consul 介绍可以阅读

- 📄 [Node.js 服务发现 \(Consul\) 使用指南](#)
- 📄 [服务发现 \(Consul\) 快速入门 for 前端](#)

使用介绍

📌 Node.js 端 Consul 相关操作都是由 [@byted-service/consul](#) 进行实现的，包括了**服务查询**、**服务监听**等功能。

Node.js 端一般只有服务查询的需求，而需要查询的服务，通过容器部署脚本，已经实现服务注册。

开发环境（Devbox/BOE）使用 Consul 查询服务，需要配置 `CONSUL_HTTP_HOST/ROUTE_IP`，实际就是指定 `Consul Agent` 的地址，向它发送服务查询请求。默认地址 `127.0.0.1:2280`

```
1  "scripts": {
2    "dev": "ROUTE_IP=10.225.130.44 ...",
3  }
```

具体实现

Consul 抽象类

通过 Consul 类生成实例，通过实例发起服务查询请求。

```
1  // consul.ts
2  export class Consul {
3    public address: string;
4    private cache = {} as {
5      [key: string]:
6        | {
7          updatedAt: number;
8          endpoints: Endpoint[];
9          consulHash: string;
10         }
11      | undefined;
12   };
13   private cacheTime: number;
14
15   private logger: ILogger;
16   private enableWarning: boolean;
```

```
17
18     private request: ReturnType<typeof createRequest>;
19
20     constructor(options: ConsulOptions = {}) {
21         ...
22     }
23
24     public async lookup(psm: string, options: LookupOptions = {}) {
25         ...
26     }
27 }
28
29 export interface ConsulOptions {
30     /**
31      * Consul Agent Host
32      */
33     host?: string;
34     /**
35      * Consul Agent Port
36      */
37     port?: string | number;
38     /**
39      * Consul Agent UDS
40      */
41     socketPath?: string;
42     /**
43      * Logger 实例
44      */
45     logger?: ILogger;
46     /**
47      * 是否输出 warning, 本地默认 false
48      */
49     enableWarning?: boolean;
50     /**
51      * 缓存时间, 单位 ms, 默认 15s
52      */
53     cacheTime?: number;
54     /**
55      * 是否开启 debug ( 调试用 ), 默认 false
56      */
57     debug?: boolean;
58 }
59
60 export interface LookupOptions {
61     /**
62      * 服务 Cluster
63      */
64     cluster?: string;
65     /**
66      * 服务 IDC
67      */
68     idc?: string;
69     /**
70      * 服务 address family 过滤条件
71      * - dual-stack: 返回 IPv4 + IPv6 地址
72      * - v4: 返回 IPv4 地址
73      * - v6: 返回 IPv6 地址
74      */
75     addrfam?: string;
76     /**
77      * 服务地址合并条件, 只有当 options.addrfam 是 dual-stack 时, 才会生效
78      * - v4: 以 IPv4 为主
```

```

79      * - v6: 以 IPv6 为主
80      */
81      unique?: string;
82      /**
83       * 返回条数
84       */
85      limit?: number;
86      /**
87       * 是否不使用缓存
88       */
89      nocache?: boolean;
90      /**
91       * 请求 Consul Agent 时使用的 x-consul-result-hash 值
92       */
93      consulHash?: string;
94      /**
95       * 超时时间, 单位 ms, 默认 3s
96       */
97      timeout?: number;
98  }
99

```

Consul 内部只有一个 `lookup` 方法, 通过 `psm` 获取服务地址。

💡 由于 `@byted-service/consul` 在 `byted-service` 中被广泛集成, 因此在大多数情况下, 无需手动创建并调用 `consul.lookup()`

Http 协议

具体实现是基于 `http` 协议, 通过向 `Consul Agent` 发起服务查询请求:

```

1  curl 'http://${ROUTE_IP}:2280/v1/lookup/name?...'
2
3  url query:
4      name // psm
5      limit
6      cluster
7      addr-family
8      unique
9

```

也可以指定 `socket` 文件, 默认路径: `/opt/tmp/sock/consul.sock`

Endpoint 节点

通过 Consul 查询返回的服务 `Endpoint` 可能有多个, 可以通过方法 `random` 实现负载均衡, 基于服务的权重 `weight`。

```

1  export interface Endpoint {
2      host: string;
3      port: number;
4      /** @deprecated use addr instead */
5      uri: string;
6      addr: string;

```

```
7     cluster: string;
8     env: string;
9     weight: number;
10    tags: { [key: string]: string };
11  }
12
13  export const random = async (
14    psm: string,
15    options: LookupOptions = {},
16    consul = new Consul()
17  ) => {
18    const endpoints = await lookupName(psm, options, consul);
19    let total = 0;
20    for (let i = 0; i < endpoints.length; i++) {
21      total += endpoints[i].weight;
22    }
23
24    let rand = Math.floor(total * Math.random());
25    for (let i = 0; i < endpoints.length; i++) {
26      rand -= endpoints[i].weight;
27
28      if (rand < 0) {
29        return endpoints[i];
30      }
31    }
32
33    /* istanbul ignore next */
34    throw new ConsulError('should never happen');
35  };
```

缓存

默认缓存时间

超时时间

默认请求超时时间

其他查询方式



如果你发现服务发现结果不符合预期，可以先按照下面的方法自行排查

命令行工具 sd

官网地址: https://site.bytedance.net/docs/4145/5743/sd_cmd/

在大多数环境下都已经预装了这个软件 (目前已知 faas 环境是没有预装的)

```
1 sd lookup p.s.m
```

Curl

直接用 curl 访问 Consul agent

```
1 curl 'http://127.0.0.1:2280/v1/lookup/name?name=p.s.m'
2
3 # 需要验证 ipv6 的用这个 url
4 curl 'http://127.0.0.1:2280/v1/lookup/name?name=p.s.m&addr-family=dual-stack&unique=v6'
```

指定 Socket 文件

```
1 # 查看 unix domain socket 是否存在
2 ls /opt/tmp/sock/consul.sock
3
4 # curl 版本在 7.40 以上
5 curl --unix-socket /opt/tmp/sock/consul.sock 'http://127.0.0.1:2280/v1/lookup/name?name=p.s.m'
6
7 # curl 不支持 --unix-socket 参数时, 可以先将 unix domain socket 转换为 port
8 socat -d -d TCP-LISTEN:2280,fork UNIX:/opt/tmp/sock/consul.sock
9 curl 'http://127.0.0.1:2280/v1/lookup/name?name=p.s.m&addr-family=dual-stack&unique=v6'
```