•Agent，Agent是长期运行在每个consul集群成员节点上守护进程。通过命令consul agent启动。Agent有client和server两种模式。由于每个节点都必须运行agent，所有节点要么是client要么是server。所有的Agent都可以可以调用DNS或HTTP API，并负责检查和维护服务同步。

•client 运行client模式的Agent，将所有的RPCs转发到Server。Client是相对无状态的。Client唯一所做的是在后台参与LAN gossip pool。只消耗少量的资源，少量的网络带宽。

•Server 运行Server模式的Agent，参与Raft quorum，维护集群的状态，响应RPC查询，与其他数据中心交互WAN gossip，转发查询到Leader或远程数据中心。

•datacenter  数据中心的定义似乎是显而易见的，有一些细节是必须考虑的。例如，在EC2，多个可用性区域是否被认为组成了单一的数据中心？我们定义数据中心是在同一个网络环境中——私有的，低延迟，高带宽。这不包括基于公共互联网环境，但是对于我们而言，在同一个EC2的多个可用性区域会被认为是一个的数据中心。

•Consensus - 当本系列文档中，consensus，意味着leader election协议，以及事务的顺序。由于这些事务是基于一个有限状态机，consensus的定义意味着复制状态机的一致性。

•Gossip – consul是建立在Serf之上，提供了完成的Gossip协议，用于成员维护故障检测、事件广播。详细细节参见[*gossip documentation*](https://www.consul.io/docs/internals/gossip.html)。这足以知道gossip是基于UDP协议实现随机的节点到节点的通信，主要是在UDP。

•LAN Gossip，指的是LAN gossip pool，包含位于同一个局域网或者数据中心的节点。

•WAN Gossip，指的是WAN gossip pool，只包含server节点，这些server主要分布在不同的数据中心或者通信是基于互联网或广域网的。

•RPC，远程过程调用。是允许client请求服务器的请求/响应机制。

1. -advertise：通知展现地址用来改变我们给集群中的其他节点展现的地址，一般情况下-bind地址就是展现地址
2. -bootstrap：用来控制一个server是否在bootstrap模式，在一个datacenter中只能有一个server处于bootstrap模式，当一个server处于bootstrap模式时，可以自己选举为raft leader。
3. -bootstrap-expect：在一个datacenter中期望提供的server节点数目，当该值提供的时候，consul一直等到达到指定sever数目的时候才会引导整个集群，该标记不能和bootstrap公用
4. -bind：该地址用来在集群内部的通讯，集群内的所有节点到地址都必须是可达的，默认是0.0.0.0
5. -client：consul绑定在哪个client地址上，这个地址提供HTTP、DNS、RPC等服务，默认是127.0.0.1
6. -config-file：明确的指定要加载哪个配置文件
7. -config-dir：配置文件目录，里面所有以.json结尾的文件都会被加载
8. -data-dir：提供一个目录用来存放agent的状态，所有的agent允许都需要该目录，该目录必须是稳定的，系统重启后都继续存在
9. -dc：该标记控制agent允许的datacenter的名称，默认是dc1
10. -encrypt：指定secret key，使consul在通讯时进行加密，key可以通过consul keygen生成，同一个集群中的节点必须使用相同的key
11. -join：加入一个已经启动的agent的ip地址，可以多次指定多个agent的地址。如果consul不能加入任何指定的地址中，则agent会启动失败，默认agent启动时不会加入任何节点。
12. -retry-join：和join类似，但是允许你在第一次失败后进行尝试。
13. -retry-interval：两次join之间的时间间隔，默认是30s
14. -retry-max：尝试重复join的次数，默认是0，也就是无限次尝试
15. -log-level：consul agent启动后显示的日志信息级别。默认是info，可选：trace、debug、info、warn、err。
16. -node：节点在集群中的名称，在一个集群中必须是唯一的，默认是该节点的主机名
17. -protocol：consul使用的协议版本
18. -rejoin：使consul忽略先前的离开，在再次启动后仍旧尝试加入集群中。
19. -server：定义agent运行在server模式，每个集群至少有一个server，建议每个集群的server不要超过5个
20. -syslog：开启系统日志功能，只在linux/osx上生效
21. -ui-dir: 提供存放web ui资源的路径，该目录必须是可读的
22. -pid-file: 提供一个路径来存放pid文件，可以使用该文件进行SIGINT/SIGHUP(关闭/更新)agent

Agent有各种各样的配置选项，这些配置选项可以通过**命令行参数**的方式设定，也可用通过**配置文件**的方式设定——所有的配置选项都是可选的，当然也是有默认值的。

当加载配置选项时，consul是按照词典顺序从所有配置文件或目录中加载。比如，basic\_config.json会先于extra\_config.json处理。后面设定的配置选项会合并到前面的配置集合中，如果存在重复的配置选项则会覆盖。当然，在某些情况下，比如事件处理程序，后面处理程序会追加到现有的配置选项中，形成事件处理程序列表。

Consul支持**重新加载配置**(在收到SIGHUP信号时)，但也不是所有的改变都可以重新加载，只有某些ReLoadable Configuration可以重新加载。也可用通过命令重新加载，后面会介绍。

**1.1.1 命令行参数**

如下都是可以在命令行指定的参数：

Ø  -config-file

需要加载的配置文件，在文件中都是json格式的信息，该参数可以多次配置，后面文件中加载的参数会覆盖前面加载文件中的参数。

Ø  -config-dir

需要加载的配置文件目录，consul将加载目录下所有后缀为“.json”的文件，加载顺序为字母顺序，文件中配置选项合并方式如config-file。该参数可以多次配置。目录中的子目录是不会加载的。

Ø  -data-dir

此目录是为Agent存放state数据的。是所有Agent需要的，该目录应该存放在持久存储中（reboot不会丢失），对于server角色的Agent是很关键的——需要记录集群状态。并且该目录是支持文件锁。

Ø  -log-level

日志级别。默认为info。有如下级别："trace","debug", "info", "warn", and "err"。可以使用consul monitor来连接agent查看日志。日志级别可以在reload时进行改变。

Ø  -syslog

将日志记录进syslog（仅支持Linux和OSX平台）

Ø  -pid-file

记录pid的文件

Ø  -server

设置Agent是server模式还是client模式。Consul agent有两种运行模式：Server和Client。这里的Server和Client只是Consul集群层面的区分，与搭建在Cluster之上 的应用服务无关。Consule Server模式agent节点用于采用raft算法维护Consul集群的状态，官方建议每个Consul Cluster至少有3个或以上的运行在Server mode的Agent，Client节点不限。

Ø  -client

将绑定到client接口的地址，可以是HTTP、DNS、RPC服务器。默认为“127.0.0.1”——只允许回路连接。RPC地址会被其他的consul命令使用，比如consul members——查询agent列表

Ø  -node

节点在集群的名字，在集群中必须是唯一的。默认为节点的Hostname

Ø  -bootstrap

设置服务是否为“bootstrap”模式。如果数据中心只有1个server agent，那么需要设置该参数。从技术上来讲，处于bootstrap模式的服务器是可以选择自己作为Raft Leader的。在consul集群中，只有一个节点可以配置该参数，如果有多个参数配置该参数，那么难以保证一致性。

Ø  -bootstrap-expect

数据中心期望的server agent数目，不能与bootstrap一起使用。此参数要么不配置，要么consul集群中所有的server均配置相同的值。如果配置了该参数，只想当consul集群中**有效server agent的数目**达到此数量时才会引导集群启动。

Ø  -dc

数据中心的名字

Ø  -bind

用于集群内部通信的IP地址，与集群中其他节点互连可通。默认为“0.0.0.0”——consul将使用第一个有效的私有IPv4地址。如果指定“[::]”，consul将使用第一个有效的公共IPv6地址。使用TCP和UDP通信。注意防火墙，避免无法通信。

启动服务

consul agent -server -datacenter=dc1 -bootstrap -data-dir /tmp/consul -config-file ./conf -ui-dir ./dist -node=n1 -bind 127.0.0.1

consul agent -server -bootstrap -data-dir /tmp/consul -ui-dir ./dist -config-file=./conf

查看集群状态

**consul operator raft list-peers**

启动客户端

consul agent -datacenter=dc1 -data-dir /tmp/consul -config-file=./conf -ui-dir ./dist -bind 192.168.1.126

启动DNSAgent

查看53端口netstat -aon|findstr "53"

# 服务发现系统consul-HTTP API

<http://localhost:8500/>

服务发现：<http://localhost:8500/v1/catalog/node/node>名称

服务健康：<http://localhost:8500/v1/health/node/node>名称

consul的主要接口是RESTful HTTP API，该API可以用来增删查改nodes、services、checks、configguration。所有的endpoints主要分为以下类别：

kv - Key/Value存储

agent - Agent控制

catalog - 管理nodes和services

health - 管理健康监测

session - Session操作

acl - ACL创建和管理

event - 用户Events

status - Consul系统状态

下面我们就单独看看每个模块的具体内容。  
  
**agent endpoints**：agent endpoints用来和本地agent进行交互，一般用来服务注册和检查注册，支持以下接口

/v1/agent/checks : 返回本地agent注册的所有检查(包括配置文件和HTTP接口)

/v1/agent/services : 返回本地agent注册的所有 服务

/v1/agent/members : 返回agent在集群的gossip pool中看到的成员

/v1/agent/self : 返回本地agent的配置和成员信息

/v1/agent/join/<address> : 触发本地agent加入node

/v1/agent/force-leave/<node>>: 强制删除node

/v1/agent/check/register : 在本地agent增加一个检查项，使用PUT方法传输一个json格式的数据

/v1/agent/check/deregister/<checkID> : 注销一个本地agent的检查项

/v1/agent/check/pass/<checkID> : 设置一个本地检查项的状态为passing

/v1/agent/check/warn/<checkID> : 设置一个本地检查项的状态为warning

/v1/agent/check/fail/<checkID> : 设置一个本地检查项的状态为critical

/v1/agent/service/register : 在本地agent增加一个新的服务项，使用PUT方法传输一个json格式的数据

/v1/agent/service/deregister/<serviceID> : 注销一个本地agent的服务项

**catalog endpoints**：catalog endpoints用来注册/注销nodes、services、checks

/v1/catalog/register : Registers a new node, service, or check

/v1/catalog/deregister : Deregisters a node, service, or check

/v1/catalog/datacenters : Lists known datacenters

/v1/catalog/nodes : Lists nodes in a given DC

/v1/catalog/services : Lists services in a given DC

/v1/catalog/service/<service> : Lists the nodes in a given service

/v1/catalog/node/<node> : Lists the services provided by a node

**health endpoints**：health endpoints用来查询健康状况相关信息，该功能从catalog中单独分离出来

/v1/healt/node/<node>: 返回node所定义的检查，可用参数?dc=

/v1/health/checks/<service>: 返回和服务相关联的检查，可用参数?dc=

/v1/health/service/<service>: 返回给定datacenter中给定node中service

/v1/health/state/<state>: 返回给定datacenter中指定状态的服务，state可以是"any", "unknown", "passing", "warning", or "critical"，可用参数?dc=

**session endpoints**：session endpoints用来create、update、destory、query sessions

/v1/session/create: Creates a new session

/v1/session/destroy/<session>: Destroys a given session

/v1/session/info/<session>: Queries a given session

/v1/session/node/<node>: Lists sessions belonging to a node

/v1/session/list: Lists all the active sessions

**acl endpoints**：acl endpoints用来create、update、destory、query acl

/v1/acl/create: Creates a new token with policy

/v1/acl/update: Update the policy of a token

/v1/acl/destroy/<id>: Destroys a given token

/v1/acl/info/<id>: Queries the policy of a given token

/v1/acl/clone/<id>: Creates a new token by cloning an existing token

/v1/acl/list: Lists all the active tokens

**event endpoints**：event endpoints用来fire新的events、查询已有的events

/v1/event/fire/<name>: 触发一个新的event，用户event需要name和其他可选的参数，使用PUT方法

/v1/event/list: 返回agent知道的events

**status endpoints**：status endpoints用来或者consul 集群的信息

/v1/status/leader : 返回当前集群的Raft leader

/v1/status/peers : 返回当前集群中同事