1.Consul(https://www.consul.io/downloads.html)是HashiCorp公司推出的开源工具, 用于实现分布式系统的服务发现与配置共享。

与其他分布式服务注册与发现的方案，Consul的方案更"一站式"，内置了服务注册与发现框架、分布一致性协议实现（Raft算法）、健康检查、Key/Value存储、多数据中心方案，

不再需要依赖其他工具（比如ZooKeeper等）。

Consul用Golang实现，因此具有天然可移植性(支持Linux、windows和Mac OS X)；

安装包仅包含一个可执行文件，方便部署，与Docker等轻量级容器可无缝配合。

2.Consul的角色:

Agent - 运行在consul集群的每个node上的daemon，agent可以run在server或者是client上，每个agent都可以提供DNS或者是HTTP的接口，负责health check和service发现等。用consul agent命令可以启动，具体可run "consul agent --help"查看参数。

Client - client把所有的RPCs转发到server端，是相对无状态的。唯一在后台运行的时client端执行了LAN gossip pool，只消耗极少的资源和网络带宽。

server: 负责维护cluster state， RPC quiries， 和其他数据中心交换WAN gossip，转发queries给leader或者其他数据中心。 每个数据中心的 server 数量推荐为 3 个或是 5 个.

Datacenter：指的是网络环境是私有，低延迟，高带宽的，排除和公网的通信。

Serf：consul是基于Serf上实现的，Serf是一个服务发现，编配（应用集群管理等）工具，它去中心化，高可用并且能故障恢复（容忍), Serf使用Gossip协议，采用Go语言编写。

Serf提供成员关系，纠错检查，广播的功能。gossip包含的任意node-to-node间的通信，主要是基于UDP。

WAN Gossip： 指 WAN gossip pool 包含哪些都在同一局域网或者是datacenter的node。

RPC： Remote Procedure Call. 客户端和服务器端的请求/应答机制。

3.consul命令格式:

consul [--version] [--help] <command> [<args>]

Available commands are:

agent          Runs a Consul agent

force-leave    Forces a member of the cluster to enter the "left" state

info           Provides debugging information for operators

join           Tell Consul agent to join cluster

keygen         Generates a new encryption key

leave          Gracefully leaves the Consul cluster and shuts down

members        Lists the members of a Consul cluster

monitor        Stream logs from a Consul agent

reload         Triggers the agent to reload configuration files

version        Prints the Consul version

4.命令行参数:

-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

-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。

-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

-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

-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

-config-file：明确的指定要加载哪个配置文件

-config-dir：配置文件目录，里面所有以.json结尾的文件都会被加载

-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

-dc：该标记控制agent允许的datacenter的名称，默认是dc1

-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

-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。

-retry-join：和join类似，但是允许你在第一次失败后进行尝试。

-retry-interval：两次join之间的时间间隔，默认是30s

-retry-max：尝试重复join的次数，默认是0，也就是无限次尝试

-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。

-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

-protocol：consul使用的协议版本

-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。

-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

-syslog：开启系统日志功能，只在linux/osx上生效

-ui-dir:提供存放web ui资源的路径，该目录必须是可读的

-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

5.配置文件参数：

acl\_datacenter：只用于server，指定的datacenter的权威ACL信息，所有的servers和datacenter必须同意ACL datacenter

acl\_default\_policy：默认是allow

acl\_down\_policy：

acl\_master\_token：

acl\_token：agent会使用这个token和consul server进行请求

acl\_ttl：控制TTL的cache，默认是30s

addresses：一个嵌套对象，可以设置以下key：dns、http、rpc

advertise\_addr：等同于-advertise

bootstrap：等同于-bootstrap

bootstrap\_expect：等同于-bootstrap-expect

bind\_addr：等同于-bind

ca\_file：提供CA文件路径，用来检查客户端或者服务端的链接

cert\_file：必须和key\_file一起

check\_update\_interval：

client\_addr：等同于-client

datacenter：等同于-dc

data\_dir：等同于-data-dir

disable\_anonymous\_signature：在进行更新检查时禁止匿名签名

disable\_remote\_exec：禁止支持远程执行，设置为true，agent会忽视所有进入的远程执行请求

disable\_update\_check：禁止自动检查安全公告和新版本信息

dns\_config：是一个嵌套对象，可以设置以下参数：allow\_stale、max\_stale、node\_ttl 、service\_ttl、enable\_truncate

domain：默认情况下consul在进行DNS查询时，查询的是consul域，可以通过该参数进行修改

enable\_debug：开启debug模式

enable\_syslog：等同于-syslog

encrypt：等同于-encrypt

key\_file：提供私钥的路径

leave\_on\_terminate：默认是false，如果为true，当agent收到一个TERM信号的时候，它会发送leave信息到集群中的其他节点上。

log\_level：等同于-log-level

node\_name:等同于-node

ports：这是一个嵌套对象，可以设置以下key：dns(dns地址：8600)、http(http api地址：8500)、rpc(rpc:8400)、serf\_lan(lan port:8301)、serf\_wan(wan port:8302)、server(server rpc:8300)

protocol：等同于-protocol

recursor：

rejoin\_after\_leave：等同于-rejoin

retry\_join：等同于-retry-join

retry\_interval：等同于-retry-interval

server：等同于-server

server\_name：会覆盖TLS CA的node\_name，可以用来确认CA name和hostname相匹配

skip\_leave\_on\_interrupt：和leave\_on\_terminate比较类似，不过只影响当前句柄

start\_join：一个字符数组提供的节点地址会在启动时被加入

statsd\_addr：

statsite\_addr：

syslog\_facility：当enable\_syslog被提供后，该参数控制哪个级别的信息被发送，默认Local0

ui\_dir：等同于-ui-dir

verify\_incoming：默认false，如果为true，则所有进入链接都需要使用TLS，需要客户端使用ca\_file提供ca文件，只用于consul server端，因为client从来没有进入的链接

verify\_outgoing：默认false，如果为true，则所有出去链接都需要使用TLS，需要服务端使用ca\_file提供ca文件，consul server和client都需要使用，因为两者都有出去的链接

watches：watch一个详细名单

6.

7.

8.

9.

10.