**[consul配置参数大全、详解、总结](https://www.cnblogs.com/sunsky303/p/9209024.html)**

<https://www.cnblogs.com/sunsky303/p/9209024.html>

## 命令行选项

以下选项全部在命令行中指定。

* [-advertise](https://www.consul.io/docs/agent/options.html" \l "_advertise) - 通告地址用于更改我们通告给集群中其他节点的地址。默认情况下，[-bind](https://www.consul.io/docs/agent/options.html" \l "_bind)地址是通告的。但是，在某些情况下，可能存在无法绑定的可路由地址。这个标志使闲聊不同的地址来支持这一点。如果此地址不可路由，则节点将处于持续振荡状态，因为其他节点会将非可路由性视为故障。在Consul 1.0和更高版本中，这可以设置为 [go-sockaddr](https://godoc.org/github.com/hashicorp/go-sockaddr/template) 模板。

* [-advertise-wan](https://www.consul.io/docs/agent/options.html" \l "_advertise-wan) - 广告WAN地址用于更改我们向通过WAN加入的服务器节点发布的地址。这也可以在与[translate\_wan\_addrs](https://www.consul.io/docs/agent/options.html" \l "translate_wan_addrs)配置选项结合使用时在客户端代理上设置。默认情况下，[-advertise](https://www.consul.io/docs/agent/options.html" \l "_advertise)地址是通告的。但是，在某些情况下，所有数据中心的所有成员都不能位于同一个物理或虚拟网络上，尤其是混合云和专用数据中心的混合设置。该标志使服务器节点能够通过WAN的公共网络闲聊，同时使用专用VLAN来相互闲聊以及彼此的客户端代理，并且如果远程数据中心是远程数据中心，则允许客户端代理在从远程数据中心访问时访问此地址配置[translate\_wan\_addrs](https://www.consul.io/docs/agent/options.html" \l "translate_wan_addrs)。在Consul 1.0和更高版本中，这可以设置为 [go-sockaddr](https://godoc.org/github.com/hashicorp/go-sockaddr/template) 模板

* [-bootstrap](https://www.consul.io/docs/agent/options.html" \l "_bootstrap) - 该标志用于控制服务器是否处于“引导”模式。每个数据中心最多只能运行一个服务器，这一点很重要。从技术上讲，一个处于引导模式的服务器可以自我选择为Raft领导者。只有一个节点处于这种模式非常重要; 否则，一致性不能保证，因为多个节点能够自我选择。不建议在引导群集后使用此标志。

* [-bootstrap-expect](https://www.consul.io/docs/agent/options.html" \l "_bootstrap_expect) - 此标志提供数据中心中预期服务器的数量。不应该提供此值，或者该值必须与群集中的其他服务器一致。提供时，Consul会等待指定数量的服务器可用，然后引导群集。这允许初始领导者自动选举。这不能与遗留[-bootstrap](https://www.consul.io/docs/agent/options.html" \l "_bootstrap)标志结合使用。该标志需要[-server](https://www.consul.io/docs/agent/options.html" \l "_server)模式。

* [-bind](https://www.consul.io/docs/agent/options.html" \l "_bind) - 应为内部集群通信绑定的地址。这是集群中所有其他节点都应该可以访问的IP地址。默认情况下，这是“0.0.0.0”，这意味着Consul将绑定到本地计算机上的所有地址，并将 第一个可用的私有IPv4地址[通告](https://www.consul.io/docs/agent/options.html" \l "_advertise)给群集的其余部分。如果有多个私有IPv4地址可用，Consul将在启动时退出并出现错误。如果你指定“[::]”，领事将 [做广告](https://www.consul.io/docs/agent/options.html#_advertise)第一个可用的公共IPv6地址。如果有多个公共IPv6地址可用，则Consul将在启动时退出并出现错误。Consul同时使用TCP和UDP以及相同的端口。如果您有任何防火墙，请确保同时允许这两种协议。在Consul 1.0和更高版本中，可以将其设置为要绑定到的空间分隔的地址列表，或者可能会解析为多个地址的 [go-sockaddr](https://godoc.org/github.com/hashicorp/go-sockaddr/template)模板。

* [-serf-wan-bind](https://www.consul.io/docs/agent/options.html" \l "_serf_wan_bind) - 应该被绑定到Serf WAN八卦通信的地址。默认情况下，该值遵循与[-bind命令行标志](https://www.consul.io/docs/agent/options.html" \l "_bind)相同的规则，如果未指定该值，-bind则使用该选项。这在Consul 0.7.1及更高版本中可用。在Consul 1.0和更高版本中，这可以设置为 [go-sockaddr](https://godoc.org/github.com/hashicorp/go-sockaddr/template) 模板

* [-serf-lan-bind](https://www.consul.io/docs/agent/options.html" \l "_serf_lan_bind) - Serf LAN八卦通信应该绑定的地址。这是群集中所有其他LAN节点应可访问的IP地址。默认情况下，该值遵循与[-bind命令行标志](https://www.consul.io/docs/agent/options.html" \l "_bind)相同的规则，如果未指定该值，-bind则使用该选项。这在Consul 0.7.1及更高版本中可用。在Consul 1.0和更高版本中，这可以设置为 [go-sockaddr](https://godoc.org/github.com/hashicorp/go-sockaddr/template)模板

* [-client](https://www.consul.io/docs/agent/options.html" \l "_client) - Consul将绑定客户端接口的地址，包括HTTP和DNS服务器。默认情况下，这是“127.0.0.1”，只允许回送连接。在Consul 1.0和更高版本中，可以将其设置为要绑定到的空间分隔的地址列表，或者 可能会解析为多个地址的 [go-sockaddr](https://godoc.org/github.com/hashicorp/go-sockaddr/template)模板。

* [-config-file](https://www.consul.io/docs/agent/options.html" \l "_config_file) - 要加载的配置文件。有关此文件格式的更多信息，请阅读[配置文件](https://www.consul.io/docs/agent/options.html" \l "configuration_files)部分。该选项可以多次指定以加载多个配置文件。如果指定了多次，稍后加载的配置文件将与先前加载的配置文件合并。在配置合并期间，单值键（string，int，bool）将简单地将它们的值替换，而列表类型将被附加在一起。

* [-config-dir](https://www.consul.io/docs/agent/options.html" \l "_config_dir) - 要加载的配置文件的目录。Consul将加载后缀为“.json”的所有文件。加载顺序是按字母顺序排列的，并使用与上述[config-file](https://www.consul.io/docs/agent/options.html" \l "_config_file)选项相同的合并例程 。可以多次指定此选项以加载多个目录。不加载config目录的子目录。有关配置文件格式的更多信息，请参阅 [配置文件](https://www.consul.io/docs/agent/options.html#configuration_files)部分。

* [-config-format](https://www.consul.io/docs/agent/options.html" \l "_config_format) - 要加载的配置文件的格式。通常，Consul会从“.json”或“.hcl”扩展名检测配置文件的格式。将此选项设置为“json”或“hcl”强制Consul解释任何带或不带扩展名的文件，以该格式解释。

* [-data-dir](https://www.consul.io/docs/agent/options.html" \l "_data_dir) - 此标志为代理存储状态提供了一个数据目录。这对所有代理都是必需的。该目录在重新启动时应该是持久的。这对于在服务器模式下运行的代理尤其重要，因为它们必须能够保持群集状态。此外，该目录必须支持使用文件系统锁定，这意味着某些类型的已装入文件夹（例如VirtualBox共享文件夹）可能不合适。**注意：**服务器和非服务器代理都可以在此目录中的状态下存储ACL令牌，因此读取访问权限可以授予对服务器上的任何令牌的访问权限，并允许访问非服务器上的服务注册期间使用的任何令牌。在基于Unix的平台上，这些文件使用0600权限编写，因此您应确保只有受信任的进程可以与Consul一样的用户身份执行。在Windows上，您应确保该目录具有适当的权限配置，因为这些权限将被继承。

* [-datacenter](https://www.consul.io/docs/agent/options.html" \l "_datacenter) - 此标志控制运行代理程序的数据中心。如果未提供，则默认为“dc1”。Consul对多个数据中心拥有一流的支持，但它依赖于正确的配置。同一个数据中心内的节点应该位于单个局域网中。

* [-dev](https://www.consul.io/docs/agent/options.html" \l "_dev) - 启用开发服务器模式。这对于在关闭所有持久性选项的情况下快速启动Consul代理非常有用，从而启用可用于快​​速原型开发或针对API进行开发的内存服务器。此模式**不**适合生产使用，因为它不会将任何数据写入磁盘。

* [-disable-host-node-id](https://www.consul.io/docs/agent/options.html" \l "_disable_host_node_id) - 将此设置为true将阻止Consul使用来自主机的信息生成确定性节点标识，并将生成随机节点标识，该标识将保留在数据目录中。在同一台主机上运行多个Consul代理进行测试时，这非常有用。Consul在版本0.8.5和0.8.5之前缺省为false，因此您必须选择加入基于主机的ID。基于主机的ID是使用<https://github.com/shirou/gopsutil/tree/master/host>生成的，与HashiCorp的[Nomad](https://www.nomadproject.io/)共享 ，因此如果您选择加入基于主机的ID，则Consul和Nomad将使用信息在主机上在两个系统中自动分配相同的ID。

* [-disable-keyring-file](https://www.consul.io/docs/agent/options.html" \l "_disable_keyring_file) - 如果设置，密钥环不会被保存到文件中。任何已安装的密钥在关机时将丢失，只有在给定的 -encrypt密钥在启动时可用。这默认为false。

* [-dns-port](https://www.consul.io/docs/agent/options.html" \l "_dns_port) - 侦听的DNS端口。这将覆盖默认端口8600.这在Consul 0.7和更高版本中可用。

* [-domain](https://www.consul.io/docs/agent/options.html" \l "_domain) - 默认情况下，Consul响应“consul”中的DNS查询。域。该标志可用于更改该域。该域中的所有查询都假定由Consul处理，不会递归解决。

* [-enable-script-checks](https://www.consul.io/docs/agent/options.html" \l "_enable_script_checks)这将控制是否在此代理上启用[执行脚本的运行状况检查](https://www.consul.io/docs/agent/checks.html)，并且默认为false运营商必须选择允许这些[脚本](https://www.consul.io/docs/agent/checks.html)。如果启用，建议[启用ACL](https://www.consul.io/docs/guides/acl.html)以控制允许哪些用户注册新的检查以执行脚本。这是在Consul 0.9.0中添加的。

* [-encrypt](https://www.consul.io/docs/agent/options.html" \l "_encrypt) - 指定用于加密Consul网络流量的密钥。该密钥必须是Base64编码的16字节。创建加密密钥的最简单方法是使用 [consul keygen](https://www.consul.io/docs/commands/keygen.html)。群集中的所有节点必须共享相同的加密密钥才能进行通信。提供的密钥会自动保留到数据目录并在代理程序重新启动时自动加载。这意味着为了加密Consul的闲话协议，这个选项只需要在每个代理的初始启动序列中提供一次。如果Consul在使用加密密钥初始化后提供，则忽略提供的密钥并显示警告。

* [-hcl](https://www.consul.io/docs/agent/options.html" \l "_hcl) - HCL配置片段。此HCL配置片段将附加到配置中，并允许在命令行上指定配置文件的全部选项。该选项可以多次指定。这是在Consul 1.0中添加的。

* [-http-port](https://www.consul.io/docs/agent/options.html" \l "_http_port) - 要监听的HTTP API端口。这覆盖了默认端口8500.当将Consul部署到通过环境传递HTTP端口的环境时，此选项非常有用，例如像CloudFoundry这样的PaaS，允许您通过Procfile直接设置端口。

* [-join](https://www.consul.io/docs/agent/options.html" \l "_join) - 启动时加入的另一位代理的地址。这可以指定多次以指定多个代理加入。如果Consul无法加入任何指定的地址，代理启动将失败。默认情况下，代理在启动时不会加入任何节点。请注意，[retry\_join](https://www.consul.io/docs/agent/options.html" \l "retry_join)在自动执行Consul集群部署时，使用 可能更适合帮助缓解节点启动竞争条件。

在Consul 1.1.0和更高版本中，这可以设置为 [go-sockaddr](https://godoc.org/github.com/hashicorp/go-sockaddr/template) 模板

* [-retry-join](https://www.consul.io/docs/agent/options.html" \l "retry-join)- 类似于[-join](https://www.consul.io/docs/agent/options.html" \l "_join)第一次尝试失败时允许重试连接。这对于知道地址最终可用的情况很有用。该列表可以包含IPv4，IPv6或DNS地址。在Consul 1.1.0和更高版本中，这可以设置为 [go-sockaddr](https://godoc.org/github.com/hashicorp/go-sockaddr/template) 模板。如果Consul正在非默认的Serf LAN端口上运行，则必须指定。IPv6必须使用“括号”语法。如果给出多个值，则按照列出的顺序尝试并重试它们，直到第一个成功为止。这里有些例子：
* # Using a DNS entry
* $ consul agent -retry-join "consul.domain.internal"
* # Using IPv4
* $ consul agent -retry-join "10.0.4.67"
* # Using IPv6
* $ consul agent -retry-join "[::1]:8301"

### [»](https://www.consul.io/docs/agent/options.html" \l "cloud-auto-joining)云端自动加入

从Consul 0.9.1开始，retry-join使用[go-discover](https://github.com/hashicorp/go-discover)库接受使用云元数据进行自动集群加入的统一接口 。有关更多信息，请参阅[云端自动加入页面](https://www.consul.io/docs/agent/cloud-auto-join.html)。

# Using Cloud Auto-Joining

$ consul agent -retry-join "provider=aws tag\_key=..."

* [-retry-interval](https://www.consul.io/docs/agent/options.html" \l "_retry_interval) - 加入尝试之间的等待时间。默认为30秒。

* [-retry-max](https://www.consul.io/docs/agent/options.html" \l "_retry_max)- [-join](https://www.consul.io/docs/agent/options.html#_join)在退出代码1之前尝试执行的最大尝试次数。默认情况下，它设置为0，将其解释为无限次重试。

* [-join-wan](https://www.consul.io/docs/agent/options.html" \l "_join_wan) - 启动时加入的另一个WAN代理的地址。可以指定多次以指定要加入的多个WAN代理。如果Consul无法加入任何指定的地址，代理启动将失败。默认情况下，代理[-join-wan](https://www.consul.io/docs/agent/options.html#_join_wan)启动时不会有任何节点。

在Consul 1.1.0和更高版本中，这可以设置为 [go-sockaddr](https://godoc.org/github.com/hashicorp/go-sockaddr/template) 模板。

* [-retry-join-wan](https://www.consul.io/docs/agent/options.html" \l "_retry_join_wan)- 与[retry-join](https://www.consul.io/docs/agent/options.html" \l "_retry_join)第一次尝试失败时允许重试wan连接类似。这对于我们知道地址最终可用的情况很有用。截至领事0.9.3 [云](https://www.consul.io/docs/agent/options.html#cloud-auto-joining)支持[自动加入](https://www.consul.io/docs/agent/options.html#cloud-auto-joining)。

在Consul 1.1.0和更高版本中，这可以设置为 [go-sockaddr](https://godoc.org/github.com/hashicorp/go-sockaddr/template) 模板

* [-retry-interval-wan](https://www.consul.io/docs/agent/options.html" \l "_retry_interval_wan)- 两次[-join-wan](https://www.consul.io/docs/agent/options.html" \l "_join_wan)尝试之间的等待时间。默认为30秒。

* [-retry-max-wan](https://www.consul.io/docs/agent/options.html" \l "_retry_max_wan)- [-join-wan](https://www.consul.io/docs/agent/options.html#_join_wan)在退出代码1之前尝试执行的最大尝试次数。默认情况下，它设置为0，将其解释为无限次重试。

* [-log-level](https://www.consul.io/docs/agent/options.html" \l "_log_level) - Consul代理启动后显示的日志级别。这默认为“信息”。可用的日志级别是“跟踪”，“调试”，“信息”，“警告”和“错误”。您始终可以通过[consul monitor](https://www.consul.io/docs/commands/monitor.html)并使用任何日志级别连接到代理。另外，日志级别可以在配置重载期间更改。

* [-node](https://www.consul.io/docs/agent/options.html" \l "_node) - 集群中此节点的名称。这在集群内必须是唯一的。默认情况下，这是机器的主机名。

* [-node-id](https://www.consul.io/docs/agent/options.html" \l "_node_id) - 在Consul 0.7.3及更高版本中可用，即使节点或地址的名称发生更改，该节点仍然是该节点的唯一标识符。这必须采用十六进制字符串的形式，长度为36个字符，例如 adf4238a-882b-9ddc-4a9d-5b6758e4159e。如果未提供（最常见的情况），那么代理将在启动时生成一个标识符，并将其保存在[数据目录中，](https://www.consul.io/docs/agent/options.html" \l "_data_dir) 以便在代理重新启动时保持相同。如果可能，主机的信息将用于生成确定性节点ID，除非[-disable-host-node-id](https://www.consul.io/docs/agent/options.html" \l "_disable_host_node_id)设置为true。

* [-node-meta](https://www.consul.io/docs/agent/options.html" \l "_node_meta)- 在Consul 0.7.3及更高版本中可用，这指定了一个任意的元数据键/值对，与表单的节点相关联key:value。这可以指定多次。节点元数据对具有以下限制：
  + 每个节点最多可注册64个键/值对。
  + 元数据密钥的长度必须介于1到128个字符（含）之间
  + 元数据键只能包含字母数字-，和\_字符。
  + 元数据密钥不能以consul-前缀开头; 这是保留供内部使用的领事。
  + 元数据值的长度必须介于0到512（含）之间。
  + 开头的密钥的元数据值rfc1035-在DNS TXT请求中逐字编码，否则元数据kv对将根据[RFC1464进行](https://www.ietf.org/rfc/rfc1464.txt)编码。

* [-pid-file](https://www.consul.io/docs/agent/options.html" \l "_pid_file) - 此标志为代理存储其PID提供文件路径。这对发送信号很有用（例如，SIGINT 关闭代理或SIGHUP更新检查确定

* [-protocol](https://www.consul.io/docs/agent/options.html" \l "_protocol) - 要使用的Consul协议版本。这默认为最新版本。这应该只在[升级](https://www.consul.io/docs/upgrading.html)时设置。您可以通过运行查看Consul支持的协议版本consul -v。

* [-raft-protocol](https://www.consul.io/docs/agent/options.html" \l "_raft_protocol) - 它控制用于服务器通信的内部版本的Raft一致性协议。必须将其设置为3才能访问自动驾驶仪功能，但不包括[cleanup\_dead\_servers](https://www.consul.io/docs/agent/options.html" \l "cleanup_dead_servers)。Consul 1.0.0及更高版本默认为3（以前默认为2）。有关 详细信息，请参阅 [Raft协议版本兼容性](https://www.consul.io/docs/upgrade-specific.html#raft-protocol-version-compatibility)。

* [-raft-snapshot-threshold](https://www.consul.io/docs/agent/options.html" \l "_raft_snapshot_threshold) - 这将控制保存到磁盘的快照之间的最小数量的木筏提交条目。这是一个很少需要更改的低级参数。遇到磁盘IO过多的非常繁忙的群集可能会增加此值以减少磁盘IO，并最大限度地减少所有服务器同时进行快照的机会。由于日志会变得更大并且raft.db文件中的空间直到下一个快照才能被回收，所以增加这一点会使磁盘空间与磁盘IO之间的交易关闭。如果由于需要重播更多日志而导致服务器崩溃或故障切换时间延长，服务器可能需要更长时间才能恢复。在Consul 1.1.0和更高版本中，这个默认值为16384，在之前的版本中它被设置为8192。

* [-raft-snapshot-interval](https://www.consul.io/docs/agent/options.html" \l "_raft_snapshot_interval) - 控制服务器检查是否需要将快照保存到磁盘的频率。他是一个很少需要改变的低级参数。遇到磁盘IO过多的非常繁忙的群集可能会增加此值以减少磁盘IO，并最大限度地减少所有服务器同时进行快照的机会。由于日志会变得更大并且raft.db文件中的空间直到下一个快照才能被回收，所以增加这一点会使磁盘空间与磁盘IO之间的交易关闭。如果由于需要重播更多日志而导致服务器崩溃或故障切换时间延长，服务器可能需要更长时间才能恢复。在Consul 1.1.0及更高版本中，这个默认设置为30s，并且在之前的版本中设置为5s。

* [-recursor](https://www.consul.io/docs/agent/options.html" \l "_recursor) - 指定上游DNS服务器的地址。该选项可以提供多次，功能上与[recursors配置选项](https://www.consul.io/docs/agent/options.html" \l "recursors)等效。

* [-rejoin](https://www.consul.io/docs/agent/options.html" \l "_rejoin) - 提供时，领事将忽略先前的休假，并在开始时尝试重新加入集群。默认情况下，Consul将休假视为永久意图，并且在启动时不会再尝试加入集群。该标志允许先前的状态用于重新加入群集。

* [-segment](https://www.consul.io/docs/agent/options.html" \l "_segment) - （仅限企业）此标志用于设置代理所属网段的名称。代理只能加入其网段内的其他代理并与其通信。有关更多详细信息，请参阅[网络细分指南](https://www.consul.io/docs/guides/segments.html)。默认情况下，这是一个空字符串，它是默认的网段。

* [-server](https://www.consul.io/docs/agent/options.html" \l "_server) - 此标志用于控制代理是否处于服务器或客户端模式。提供时，代理将充当领事服务器。每个Consul集群必须至少有一个服务器，理想情况下每个数据中心不超过5个。所有服务器都参与Raft一致性算法，以确保事务以一致的，可线性化的方式进行。事务修改所有服务器节点上维护的集群状态，以确保节点发生故障时的可用性。服务器节点还参与其他数据中心中服务器节点的WAN八卦池。服务器充当其他数据中心的网关，并根据需要转发流量。

* [-non-voting-server](https://www.consul.io/docs/agent/options.html" \l "_non_voting_server) - （仅限企业）此标志用于使服务器不参与Raft仲裁，并使其仅接收数据复制流。在需要大量读取服务器的情况下，这可用于将读取可伸缩性添加到群集。

* [-syslog](https://www.consul.io/docs/agent/options.html" \l "_syslog) - 该标志启用记录到系统日志。这仅在Linux和OSX上受支持。如果在Windows上提供，将会导致错误。

* [-ui](https://www.consul.io/docs/agent/options.html" \l "_ui) - 启用内置的Web UI服务器和所需的HTTP路由。这消除了将Consul Web UI文件与二进制文件分开维护的需要。

* [-ui-dir](https://www.consul.io/docs/agent/options.html" \l "_ui_dir) - 此标志提供包含Consul的Web UI资源的目录。这将自动启用Web UI。目录必须对代理可读。从Consul版本0.7.0及更高版本开始，Web UI资产包含在二进制文件中，因此不再需要此标志; 仅指定-ui标志就足以启用Web UI。指定'-ui'和'-ui-dir'标志将导致错误。

## [»](https://www.consul.io/docs/agent/options.html" \l "configuration-files)配置文件

除了命令行选项之外，配置还可以放入文件中。在某些情况下，这可能更容易，例如使用配置管理系统配置Consul时。

配置文件是JSON格式的，使得它们易于被人类和计算机读取和编辑。该配置被格式化为一个单独的JSON对象，并在其中进行配置。

配置文件不仅用于设置代理，还用于提供检查和服务定义。这些用于向其他群集宣布系统服务器的可用性。它们分别在[检查配置](https://www.consul.io/docs/agent/checks.html)和 [服务配置下](https://www.consul.io/docs/agent/services.html)分别记录。服务和检查定义支持在重新加载期间进行更新。

#### [»](https://www.consul.io/docs/agent/options.html" \l "example-configuration-file)示例配置文件

{

"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"

}

}

#### [»](https://www.consul.io/docs/agent/options.html" \l "example-configuration-file-with-tls)示例配置文件，带有TLS

{

"datacenter": "east-aws",

"data\_dir": "/opt/consul",

"log\_level": "INFO",

"node\_name": "foobar",

"server": true,

"addresses": {

"https": "0.0.0.0"

},

"ports": {

"https": 8080

},

"key\_file": "/etc/pki/tls/private/my.key",

"cert\_file": "/etc/pki/tls/certs/my.crt",

"ca\_file": "/etc/pki/tls/certs/ca-bundle.crt"

}

尤其请参阅ports设置的使用：

"ports": {

"https": 8080

}

除非https已为端口分配了端口号，否则Consul将不会为HTTP API启用TLS > 0。

#### [»](https://www.consul.io/docs/agent/options.html" \l "configuration-key-reference)配置密钥参考

* [acl\_datacenter](https://www.consul.io/docs/agent/options.html" \l "acl_datacenter) - 这指定了对ACL信息具有权威性的数据中心。必须提供它才能启用ACL。所有服务器和数据中心必须就ACL数据中心达成一致。将它设置在服务器上是集群级别强制执行所需的全部功能，但是为了使API正确地从客户端转发，它必须在其上进行设置。在Consul 0.8和更高版本中，这也可以实现ACL的代理级执行。有关更多详细信息，请参阅[ACL指南](https://www.consul.io/docs/guides/acl.html)。

* [acl\_default\_policy](https://www.consul.io/docs/agent/options.html" \l "acl_default_policy) - “允许”或“否认”; 默认为“允许”。默认策略在没有匹配规则时控制令牌的行为。在“允许”模式下，ACL是一个黑名单：允许任何未被明确禁止的操作。在“拒绝”模式下，ACL是白名单：任何未明确允许的操作都会被阻止。注意：在您设置acl\_datacenter 为启用ACL支持之前，这不会生效。

* [acl\_down\_policy](https://www.consul.io/docs/agent/options.html" \l "acl_down_policy) - “允许”，“拒绝”或“扩展缓存”; “扩展缓存”是默认值。如果无法从令牌[acl\_datacenter](https://www.consul.io/docs/agent/options.html" \l "acl_datacenter)或领导者节点读取令牌策略，则应用停机策略。在“允许”模式下，允许所有操作，“拒绝”限制所有操作，“扩展缓存”允许使用任何缓存ACL，忽略其TTL值。如果使用非缓存ACL，“extend-cache”就像“拒绝”一样。

* [acl\_agent\_master\_token](https://www.consul.io/docs/agent/options.html" \l "acl_agent_master_token)- 用于访问需要代理读取或写入权限的[代理端点](https://www.consul.io/api/agent.html)或节点读取权限，即使Consul服务器不存在以验证任何令牌。这应该只在运行中断时使用，应用程序通常会使用常规ACL令牌。这是在Consul 0.7.2中添加的，只有在[acl\_enforce\_version\_8](https://www.consul.io/docs/agent/options.html" \l "acl_enforce_version_8)设置为true 时才会使用 。有关更多详细信息，请参阅 [ACL Agent Master Token](https://www.consul.io/docs/guides/acl.html#acl-agent-master-token)。

* [acl\_agent\_token](https://www.consul.io/docs/agent/options.html" \l "acl_agent_token) - 用于客户端和服务器执行内部操作。如果没有指定，那么 [acl\_token](https://www.consul.io/docs/agent/options.html" \l "acl_token)将被使用。这是在领事0.7.2中添加的。

该令牌至少必须具有对其将注册的节点名称的写入访问权限，以便设置目录中的任何节点级别信息，例如元数据或节点的标记地址。还有其他地方使用了这个令牌，请参阅[ACL代理令牌](https://www.consul.io/docs/guides/acl.html#acl-agent-token) 了解更多详情。

* [acl\_enforce\_version\_8](https://www.consul.io/docs/agent/options.html" \l "acl_enforce_version_8) - 用于客户端和服务器，以确定在Consul 0.8之前预览新ACL策略是否应该执行。在Consul 0.7.2中添加，Consul版本在0.8之前默认为false，在Consul 0.8和更高版本中默认为true。这有助于在执行开始前允许策略就位，从而轻松过渡到新的ACL功能。有关更多详细信息，请参阅[ACL指南](https://www.consul.io/docs/guides/acl.html#version_8_acls)。

* [acl\_master\_token](https://www.consul.io/docs/agent/options.html" \l "acl_master_token)- 仅用于服务器[acl\_datacenter](https://www.consul.io/docs/agent/options.html" \l "acl_datacenter)。如果该令牌不存在，将使用管理级权限创建该令牌。它允许运营商使用众所周知的令牌ID引导ACL系统。

在acl\_master\_token当服务器获取集群领导只安装。如果您想要安装或更改acl\_master\_token，请acl\_master\_token 在所有服务器的配置中设置新值。一旦完成，重新启动当前领导者以强制领导人选举。如果acl\_master\_token未提供，则服务器不会创建主令牌。当你提供一个值时，它可以是任何字符串值。使用UUID将确保它看起来与其他标记相同，但并非绝对必要。

* [acl\_replication\_token](https://www.consul.io/docs/agent/options.html" \l "acl_replication_token)- 仅用于[acl\_datacenter](https://www.consul.io/docs/agent/options.html" \l "acl_datacenter)运行Consul 0.7或更高版本以外的服务器。如果提供，这将启用使用此令牌的[ACL复制](https://www.consul.io/docs/guides/acl.html" \l "replication)来检索ACL并将其复制到非权威本地数据中心。在Consul 0.9.1及更高版本中，您可以启用ACL复制[enable\_acl\_replication](https://www.consul.io/docs/agent/options.html" \l "enable_acl_replication) ，然后使用每台服务器上的[代理令牌API](https://www.consul.io/api/agent.html" \l "update-acl-tokens)设置令牌。如果acl\_replication\_token在配置中设置，它将自动设置[enable\_acl\_replication](https://www.consul.io/docs/agent/options.html#enable_acl_replication)为true以实现向后兼容。

如果存在影响授权数据中心的分区或其他中断，并且 [acl\_down\_policy](https://www.consul.io/docs/agent/options.html" \l "acl_down_policy)设置为“extend-cache”，则可以使用复制的ACL集在中断期间解析不在缓存中的令牌。有关更多详细信息，请参阅 [ACL指南](https://www.consul.io/docs/guides/acl.html#replication)复制部分。

* [acl\_token](https://www.consul.io/docs/agent/options.html" \l "acl_token) - 提供时，代理向Consul服务器发出请求时将使用此令牌。通过提供“？token”查询参数，客户端可以基于每个请求重写此令牌。如果未提供，则会使用映射到“匿名”ACL策略的空令牌。

* [acl\_ttl](https://www.consul.io/docs/agent/options.html" \l "acl_ttl) - 用于控制ACL的生存时间缓存。默认情况下，这是30秒。此设置会对性能产生重大影响：减少刷新次数会增加刷新次数，同时减少刷新次数。但是，由于缓存不会主动失效，所以ACL策略可能会过时到TTL值。

* [addresses](https://www.consul.io/docs/agent/options.html" \l "addresses) - 这是一个允许设置绑定地址的嵌套对象。在Consul 1.0和更高版本中，这些可以设置为要绑定的空间分隔的地址列表 ，也可以将可以解析为多个地址的[go-sockaddr](https://godoc.org/github.com/hashicorp/go-sockaddr/template)模板设置为空格分隔列表。

http支持绑定到Unix域套接字。套接字可以在表单中指定unix:///path/to/socket。一个新的域套接字将在给定的路径上创建。如果指定的文件路径已经存在，Consul将尝试清除该文件并在其位置创建域套接字。套接字文件的权限可以通过[unix\_socketsconfig结构调整](https://www.consul.io/docs/agent/options.html" \l "unix_sockets)。

在Unix套接字接口上运行Consul agent命令时，使用 -http-addr参数指定套接字的路径。您也可以将所需的值放在CONSUL\_HTTP\_ADDR环境变量中。

对于TCP地址，变量值应该是端口的IP地址。例如：10.0.0.1:8500而不是10.0.0.1。但是，[ports](https://www.consul.io/docs/agent/options.html" \l "ports)在配置文件中定义端口时，端口将在结构中单独设置 。

以下键有效：

* + [dns](https://www.consul.io/docs/agent/options.html" \l "dns) - DNS服务器。默认为client\_addr

* + [http](https://www.consul.io/docs/agent/options.html" \l "http) - HTTP API。默认为client\_addr

* + [https](https://www.consul.io/docs/agent/options.html" \l "https) - HTTPS API。默认为client\_addr

* [advertise\_addr](https://www.consul.io/docs/agent/options.html" \l "advertise_addr)等同于[-advertise命令行标志](https://www.consul.io/docs/agent/options.html" \l "_advertise)。

* [serf\_wan](https://www.consul.io/docs/agent/options.html" \l "serf_wan_bind)等同于[-serf-wan-bind命令行标志](https://www.consul.io/docs/agent/options.html" \l "_serf_wan_bind)。

* [serf\_lan](https://www.consul.io/docs/agent/options.html" \l "serf_lan_bind)等同于[-serf-lan-bind命令行标志](https://www.consul.io/docs/agent/options.html" \l "_serf_lan_bind)。

* [advertise\_addr\_wan](https://www.consul.io/docs/agent/options.html" \l "advertise_addr_wan)等同于[-advertise-wan命令行标志](https://www.consul.io/docs/agent/options.html" \l "_advertise-wan)。

* [autopilot](https://www.consul.io/docs/agent/options.html" \l "autopilot)在Consul 0.8中增加的这个对象允许设置多个子键，这些子键可以为Consul服务器配置操作友好的设置。有关自动驾驶仪的更多信息，请参阅[自动驾驶仪指南](https://www.consul.io/docs/guides/autopilot.html)。

以下子键可用：

* + [cleanup\_dead\_servers](https://www.consul.io/docs/agent/options.html" \l "cleanup_dead_servers) - 这可以控制定期和每当将新服务器添加到群集时自动删除已死的服务器节点。默认为true。

* + [last\_contact\_threshold](https://www.consul.io/docs/agent/options.html" \l "last_contact_threshold) - 在被认为不健康之前，控制服务器在没有与领导联系的情况下可以走的最长时间。必须是持续时间值，例如10s。默认为200ms。

* + [max\_trailing\_logs](https://www.consul.io/docs/agent/options.html" \l "max_trailing_logs) - 控制服务器在被认为不健康之前可以跟踪领导者的最大日志条目数。默认为250。

* + [server\_stabilization\_time](https://www.consul.io/docs/agent/options.html" \l "server_stabilization_time) - 在添加到集群之前，控制服务器在“健康”状态下必须稳定的最短时间。只有当所有服务器运行Raft协议版本3或更高时才会生效。必须是持续时间值，例如30s。默认为10s。

* + [redundancy\_zone\_tag](https://www.consul.io/docs/agent/options.html" \l "redundancy_zone_tag)- （仅限企业）[-node-meta](https://www.consul.io/docs/agent/options.html" \l "_node_meta)当Autopilot将服务器分为多个区域进行冗余时，这将控制使用的密钥。每个区域中只有一台服务器可以同时成为投票成员。如果留空（默认），则此功能将被禁用。

* + [disable\_upgrade\_migration](https://www.consul.io/docs/agent/options.html" \l "disable_upgrade_migration)- （仅限企业）如果设置为true，此设置将禁用Consul Enterprise中的Autopilot升级迁移策略，等待足够的新版本服务器添加到群集，然后再将其中的任何一个升级为选民。默认为false。

* [bootstrap](https://www.consul.io/docs/agent/options.html" \l "bootstrap)等同于 [-bootstrap命令行标志](https://www.consul.io/docs/agent/options.html#_bootstrap)。

* [bootstrap\_expect](https://www.consul.io/docs/agent/options.html" \l "bootstrap_expect)等同于[-bootstrap-expect命令行标志](https://www.consul.io/docs/agent/options.html" \l "_bootstrap_expect)。

* [bind\_addr](https://www.consul.io/docs/agent/options.html" \l "bind_addr)等同于 [-bind命令行标志](https://www.consul.io/docs/agent/options.html#_bind)。

* [ca\_file](https://www.consul.io/docs/agent/options.html" \l "ca_file)这为PEM编码的证书颁发机构提供了一个文件路径。证书颁发机构用于使用适当的[verify\_incoming](https://www.consul.io/docs/agent/options.html" \l "verify_incoming)或 [verify\_outgoing](https://www.consul.io/docs/agent/options.html" \l "verify_outgoing)标志检查客户端和服务器连接的真实性。

* [ca\_path](https://www.consul.io/docs/agent/options.html" \l "ca_path)这提供了PEM编码证书颁发机构文件目录的路径。这些证书颁发机构用于检查具有适当[verify\_incoming](https://www.consul.io/docs/agent/options.html" \l "verify_incoming)或 [verify\_outgoing](https://www.consul.io/docs/agent/options.html" \l "verify_outgoing)标志的客户端和服务器连接的真实性。

* [cert\_file](https://www.consul.io/docs/agent/options.html" \l "cert_file)这提供了一个PEM编码证书的文件路径。证书提供给客户或服务器来验证代理的真实性。它必须随同提供[key\_file](https://www.consul.io/docs/agent/options.html" \l "key_file)。

* [check\_update\_interval](https://www.consul.io/docs/agent/options.html" \l "check_update_interval) 此间隔控制检查稳定状态检查的输出与服务器同步的频率。默认情况下，它被设置为5分钟（“5米”）。许多处于稳定状态的检查会导致每次运行的输出略有不同（时间戳等），从而导致不断的写入。该配置允许推迟检查输出的同步，以减少给定时间间隔的写入压力。如果支票更改状态，则新状态和相关输出立即同步。要禁用此行为，请将该值设置为“0s”。

* [client\_addr](https://www.consul.io/docs/agent/options.html" \l "client_addr)等同于 [-client命令行标志](https://www.consul.io/docs/agent/options.html#_client)。

* [datacenter](https://www.consul.io/docs/agent/options.html" \l "datacenter)等同于 [-datacenter命令行标志](https://www.consul.io/docs/agent/options.html#_datacenter)。

* [data\_dir](https://www.consul.io/docs/agent/options.html" \l "data_dir)等同于 [-data-dir命令行标志](https://www.consul.io/docs/agent/options.html#_data_dir)。

* [disable\_anonymous\_signature](https://www.consul.io/docs/agent/options.html" \l "disable_anonymous_signature)禁止使用更新检查提供匿名签名以进行重复数据删除。看[disable\_update\_check](https://www.consul.io/docs/agent/options.html" \l "disable_update_check)。

* [disable\_host\_node\_id](https://www.consul.io/docs/agent/options.html" \l "disable_host_node_id) 等同于[-disable-host-node-id命令行标志](https://www.consul.io/docs/agent/options.html" \l "_disable_host_node_id)。

* [disable\_remote\_exec](https://www.consul.io/docs/agent/options.html" \l "disable_remote_exec) 禁用对远程执行的支持。设置为true时，代理将忽略任何传入的远程exec请求。在0.8版之前的Consul版本中，这个默认为false。在Consul 0.8中，默认值更改为true，以使远程exec选择加入而不是选择退出。

* [disable\_update\_check](https://www.consul.io/docs/agent/options.html" \l "disable_update_check) 禁用自动检查安全公告和新版本发布。这在Consul Enterprise中被禁用。

* [discard\_check\_output](https://www.consul.io/docs/agent/options.html" \l "discard_check_output) 在存储之前丢弃健康检查的输出。这减少了健康检查具有易失性输出（如时间戳，进程ID，...）的环境中Consul raft日志的写入次数。

* + [discovery\_max\_stale](https://www.consul.io/docs/agent/options.html" \l "discovery_max_stale) - 为所有服务发现HTTP端点启用陈旧请求。这相当于[max\_stale](https://www.consul.io/docs/agent/options.html" \l "max_stale)DNS请求的 配置。如果此值为零（默认值），则将所有服务发现HTTP端点转发给领导者。如果此值大于零，则任何Consul服务器都可以处理服务发现请求。如果领队服务器超过领导者discovery\_max\_stale，则将对领导者重新评估该查询以获得更多最新结果。Consul代理还会添加一个新的 X-Consul-Effective-Consistency响应标头，用于指示代理是否执行了陈旧的读取。discover-max-stale 在Consul 1.0.7中引入，作为Consul操作员在代理级别强制来自客户端的陈旧请求的方式，默认值为0，与先前Consul版本中的默认一致性行为相匹配。

* [dns\_config](https://www.consul.io/docs/agent/options.html" \l "dns_config)此对象允许设置多个可以调节DNS查询服务的子密钥。有关更多详细信息，请参阅[DNS缓存](https://www.consul.io/docs/guides/dns-cache.html)指南 。

以下子键可用：

* + [allow\_stale](https://www.consul.io/docs/agent/options.html" \l "allow_stale) - 启用DNS信息的陈旧查询。这允许任何Consul服务器而不仅仅是领导者来服务请求。这样做的好处是您可以通过Consul服务器获得线性读取可扩展性。在0.7之前的Consul版本中，默认为false，意味着所有请求都由领导者提供服务，从而提供更强的一致性，但吞吐量更低，延迟更高。在Consul 0.7及更高版本中，为了更好地利用可用服务器，默认为true。

* + [max\_stale](https://www.consul.io/docs/agent/options.html" \l "max_stale)- 什么时候[allow\_stale](https://www.consul.io/docs/agent/options.html" \l "allow_stale) 被指定，这是用来限制陈旧结果被允许的。如果领队服务器超过领导者max\_stale，则将对领导者重新评估该查询以获得更多最新结果。在领事0.7.1之前，这默认为5秒; 在Consul 0.7.1和更高版本中，默认为10年（“87600h”），这有效地允许任何服务器回答DNS查询，不管它多么陈旧。实际上，服务器通常只比领导者短几毫秒，所以这可以让Consul在没有领导者可以选举的长时间停工场景中继续提供请求。

* + [node\_ttl](https://www.consul.io/docs/agent/options.html" \l "node_ttl) - 默认情况下，这是“0”，因此所有节点查找均以0 TTL值提供服务。通过设置此值可以启用节点查找的DNS缓存。这应该用“s”后缀表示第二个或“m”表示分钟。

* + [service\_ttl](https://www.consul.io/docs/agent/options.html" \l "service_ttl) - 这是一个允许使用每项服务策略设置TTL服务查找的子对象。当没有特定的服务可用于服务时，可以使用“\*”通配符服务。默认情况下，所有服务均以0 TTL值提供服务。通过设置此值可启用服务查找的DNS缓存。

* + [enable\_truncate](https://www.consul.io/docs/agent/options.html" \l "enable_truncate) - 如果设置为true，则将返回超过3条记录或超过适合有效UDP响应的UDP DNS查询将设置截断标志，指示客户端应使用TCP重新查询以获得满载记录集。

* + [only\_passing](https://www.consul.io/docs/agent/options.html" \l "only_passing) - 如果设置为true，任何健康检查警告或严重的节点将被排除在DNS结果之外。如果为false，则默认情况下，只有健康检查失败的节点将被排除。对于服务查找，会考虑节点自身的运行状况检查以及特定于服务的检查。例如，如果某个节点的健康状况检查非常重要，则该节点上的所有服务都将被排除，因为它们也被视为关键。

* + [recursor\_timeout](https://www.consul.io/docs/agent/options.html" \l "recursor_timeout) - Consul在递归查询上游DNS服务器时使用的超时。查看[recursors](https://www.consul.io/docs/agent/options.html#recursors) 更多细节。缺省值是2s。这在Consul 0.7和更高版本中可用。

* + [disable\_compression](https://www.consul.io/docs/agent/options.html" \l "disable_compression) - 如果设置为true，则不会压缩DNS响应。Consul 0.7中默认添加并启用了压缩。

* + [udp\_answer\_limit](https://www.consul.io/docs/agent/options.html" \l "udp_answer_limit) - 限制包含在基于UDP的DNS响应的答案部分中的资源记录数。此参数仅适用于小于512字节的UDP DNS查询。此设置已弃用，并由Consul 1.0.7替换[a\_record\_limit](https://www.consul.io/docs/agent/options.html" \l "a_record_limit)。

* + [a\_record\_limit](https://www.consul.io/docs/agent/options.html" \l "a_record_limit) - 限制A，AAAA或ANY DNS响应（包括TCP和UDP）答案部分中包含的资源记录数。在回答问题时，Consul将使用匹配主机的完整列表，随机随机洗牌，然后限制答案的数量a\_record\_limit（默认：无限制）。此限制不适用于SRV记录。

在实施和实施[RFC 3484第6节](https://tools.ietf.org/html/rfc3484#section-6)规则9的环境中（即DNS答案总是被排序并因此决不是随机的），客户端可能需要设置该值1以保留预期的随机分配行为（注意： [RFC 3484](https://tools.ietf.org/html/rfc3484)已被过时 [RFC 6724](https://tools.ietf.org/html/rfc6724)，因此它应该越来越不常见，需要用现代的解析器来改变这个值）。

* [domain](https://www.consul.io/docs/agent/options.html" \l "domain)等同于 [-domain命令行标志](https://www.consul.io/docs/agent/options.html#_domain)。

* [enable\_acl\_replication](https://www.consul.io/docs/agent/options.html" \l "enable_acl_replication)在Consul服务器上设置时，启用[ACL复制](https://www.consul.io/docs/guides/acl.html" \l "replication)而不必通过设置复制令牌[acl\_replication\_token](https://www.consul.io/docs/agent/options.html#acl_replication_token)。相反，启用ACL复制，然后在每台服务器上使用[代理令牌API](https://www.consul.io/api/agent.html" \l "update-acl-tokens)引入令牌。查看[acl\_replication\_token](https://www.consul.io/docs/agent/options.html" \l "acl_replication_token)更多细节。

* [enable\_agent\_tls\_for\_checks](https://www.consul.io/docs/agent/options.html" \l "enable_agent_tls_for_checks) 当设置时，使用代理人的TLS配置的一个子集（key\_file，cert\_file，ca\_file，ca\_path，和 server\_name），以建立HTTP客户端的HTTP健康检查。这允许使用代理的凭证检查需要双向TLS的服务。这是在Consul 1.0.1中添加的，默认为false。

* [enable\_debug](https://www.consul.io/docs/agent/options.html" \l "enable_debug)设置后，启用一些额外的调试功能。目前，这仅用于设置运行时概要分析HTTP端点。

* [enable\_script\_checks](https://www.consul.io/docs/agent/options.html" \l "enable_script_checks)等同于 [-enable-script-checks命令行标志](https://www.consul.io/docs/agent/options.html#_enable_script_checks)。

* [enable\_syslog](https://www.consul.io/docs/agent/options.html" \l "enable_syslog)等同于[-syslog命令行标志](https://www.consul.io/docs/agent/options.html" \l "_syslog)。

* [encrypt](https://www.consul.io/docs/agent/options.html" \l "encrypt)等同于 [-encrypt命令行标志](https://www.consul.io/docs/agent/options.html#_encrypt)。

* [encrypt\_verify\_incoming](https://www.consul.io/docs/agent/options.html" \l "encrypt_verify_incoming) - 这是一个可选参数，可用于禁用对输入八卦执行加密，以便在正在运行的群集上从未加密的文件升级到加密的八卦。有关更多信息，请参阅[此部分](https://www.consul.io/docs/agent/encryption.html#configuring-gossip-encryption-on-an-existing-cluster)。默认为true。

* [encrypt\_verify\_outgoing](https://www.consul.io/docs/agent/options.html" \l "encrypt_verify_outgoing) - 这是一个可选参数，可用于禁用强制执行传出八卦的加密，以便在正在运行的群集上从未加密的文件转换为加密的八卦文件。有关更多信息，请参阅[此部分](https://www.consul.io/docs/agent/encryption.html" \l "configuring-gossip-encryption-on-an-existing-cluster)。默认为true。

* [disable\_keyring\_file](https://www.consul.io/docs/agent/options.html" \l "disable_keyring_file)- 相当于 [-disable-keyring-file命令行标志](https://www.consul.io/docs/agent/options.html#_disable_keyring_file)。

* [key\_file](https://www.consul.io/docs/agent/options.html" \l "key_file)这提供了一个PEM编码私钥的文件路径。密钥与证书一起用于验证代理的真实性。这必须随同提供[cert\_file](https://www.consul.io/docs/agent/options.html" \l "cert_file)。

* [http\_config](https://www.consul.io/docs/agent/options.html" \l "http_config) 该对象允许为HTTP API设置选项。

以下子键可用：

* + [block\_endpoints](https://www.consul.io/docs/agent/options.html" \l "block_endpoints) 此对象是要在代理程序上阻止的HTTP API端点前缀的列表，默认为空列表，表示所有端点都已启用。与此列表中的一个条目具有共同前缀的任何端点将被阻止，并且在访问时将返回403响应代码。例如，为了阻断所有V1 ACL端点，此设定为 ["/v1/acl"]，这将阻止/v1/acl/create，/v1/acl/update以及与开始其它ACL端点/v1/acl。这只适用于API端点，而不是，/ui或者 /debug必须禁用它们各自的配置选项。任何使用禁用端点的CLI命令都将不再起作用。对于更通用的访问控制，Consul的[ACL系统](https://www.consul.io/docs/guides/acl.html)应该被使用，但是这个选项对于完全去除对HTTP API端点的访问是有用的，或者对特定的代理来说是非常有用的。这在Consul 0.9.0及更高版本中可用。

* + [response\_headers](https://www.consul.io/docs/agent/options.html" \l "response_headers) 该对象允许向HTTP API响应添加标题。例如，可以使用以下配置在HTTP API端点上启用 [CORS](https://en.wikipedia.org/wiki/Cross-origin_resource_sharing)：
  + {
  + "http\_config": {
  + "response\_headers": {
  + "Access-Control-Allow-Origin": "\*"
  + }
  + }
  + }

* [leave\_on\_terminate](https://www.consul.io/docs/agent/options.html" \l "leave_on_terminate)如果启用，当代理收到TERM信号时，它将向Leave群集的其余部分发送消息并正常离开。此功能的默认行为根据代理是否作为客户端或服务器运行而不同（在Consul 0.7之前默认值被无条件设置为false）。在客户端模式下的代理程序中，默认为true 服务器模式的代理程序，对于服务器模式中的代理程序，缺省为false。

* [limits](https://www.consul.io/docs/agent/options.html" \l "limits)在Consul 0.9.3及更高版本中可用，这是一个嵌套对象，用于配置代理执行的限制。目前，这只适用于客户端模式的代理，而不是Consul服务器。以下参数可用：

* + [rpc\_rate](https://www.consul.io/docs/agent/options.html" \l "rpc_rate) - 通过将此代理允许为Consul服务器发出的RPC请求的最大请求速率设置为每秒请求数，配置RPC速率限制器。默认为无限，这会禁用速率限制。
  + [rpc\_max\_burst](https://www.consul.io/docs/agent/options.html" \l "rpc_max_burst) - 用于对RPC速率限制器进行再充电的令牌桶的大小。默认为1000个令牌，并且每个令牌都适用于对Consul服务器的单个RPC调用。有关 令牌桶速率限制器如何操作的更多详细信息，请参阅<https://en.wikipedia.org/wiki/Token_bucket>。

* [log\_level](https://www.consul.io/docs/agent/options.html" \l "log_level)等同于 [-log-level命令行标志](https://www.consul.io/docs/agent/options.html#_log_level)。

* [node\_id](https://www.consul.io/docs/agent/options.html" \l "node_id)等同于 [-node-id命令行标志](https://www.consul.io/docs/agent/options.html#_node_id)。

* [node\_name](https://www.consul.io/docs/agent/options.html" \l "node_name)等同于 [-node命令行标志](https://www.consul.io/docs/agent/options.html#_node)。

* [node\_meta](https://www.consul.io/docs/agent/options.html" \l "node_meta)可用于Consul 0.7.3及更高版本，此对象允许将任意元数据键/值对与本地节点相关联，然后可用于过滤某些目录端点的结果。有关更多信息，请参阅 [-node-meta命令行标志](https://www.consul.io/docs/agent/options.html#_node_meta)。
* {
* "node\_meta": {
* "instance\_type": "t2.medium"
* }
* }

* [performance](https://www.consul.io/docs/agent/options.html" \l "performance)在Consul 0.7和更高版本中可用，这是一个嵌套对象，允许调整Consul中不同子系统的性能。请参阅[服务器性能](https://www.consul.io/docs/guides/performance.html)指南获取更多详细信息 以下参数可用：

* + [leave\_drain\_time](https://www.consul.io/docs/agent/options.html" \l "leave_drain_time) - 服务器在优雅休假期间居住的时间，以便允许对其他Consul服务器重试请求。在正常情况下，这可以防止客户在执行Consul服务器滚动更新时遇到“无领导者”错误。这是在Consul 1.0中添加的。必须是持续时间值，例如10秒。默认为5秒。

* + [raft\_multiplier](https://www.consul.io/docs/agent/options.html" \l "raft_multiplier) - Consul服务器用于缩放关键Raft时间参数的整数乘法器。忽略该值或将其设置为0将使用下面描述的默认时间。较低的值用于收紧时间并提高灵敏度，而较高的值用于放松时间并降低灵敏度。调整这会影响Consul检测领导者失败并执行领导者选举所花的时间，但需要更多的网络和CPU资源才能获得更好的性能。

默认情况下，Consul将使用适用于[最小Consul服务器](https://www.consul.io/docs/guides/performance.html" \l "minimum)的较低性能时序，当前相当于将此值设置为5（此默认值可能会在未来版本的Consul中进行更改，具体取决于目标最小服务器配置文件是否更改）。将此值设置为1会将Raft配置为其最高性能模式，相当于Consul在0.7之前的默认时间，并且建议用于[生产Consul服务器](https://www.consul.io/docs/guides/performance.html" \l "production)。有关调整此参数的更多详细信息，请参阅[上次接触](https://www.consul.io/docs/guides/performance.html#last-contact)时间的说明。最大允许值是10。

* + [rpc\_hold\_timeout](https://www.consul.io/docs/agent/options.html" \l "rpc_hold_timeout) - 客户或服务器在领导者选举期间将重试内部RPC请求的持续时间。在正常情况下，这可以防止客户遇到“无领导者”的错误。这是在Consul 1.0中添加的。必须是持续时间值，例如10秒。默认为7秒。

* [ports](https://www.consul.io/docs/agent/options.html" \l "ports) 这是一个嵌套对象，允许为以下键设置绑定端口：

* + [dns](https://www.consul.io/docs/agent/options.html" \l "dns_port) - DNS服务器，-1禁用。默认8600。

* + [http](https://www.consul.io/docs/agent/options.html" \l "http_port) - HTTP API，-1禁用。默认8500。

* + [https](https://www.consul.io/docs/agent/options.html" \l "https_port) - HTTPS API，-1禁用。默认-1（禁用）。

* + [serf\_lan](https://www.consul.io/docs/agent/options.html" \l "serf_lan_port) - Serf LAN端口。默认8301。

* + [serf\_wan](https://www.consul.io/docs/agent/options.html" \l "serf_wan_port) - Serf WAN端口。默认8302.设置为-1以禁用。**注意**：这将禁用不推荐的WAN联合。各种目录和广域网相关端点将返回错误或空的结果。

* + [server](https://www.consul.io/docs/agent/options.html" \l "server_rpc_port) - 服务器RPC地址。默认8300。

* [protocol](https://www.consul.io/docs/agent/options.html" \l "protocol)等同于 [-protocol命令行标志](https://www.consul.io/docs/agent/options.html#_protocol)。

* [raft\_protocol](https://www.consul.io/docs/agent/options.html" \l "raft_protocol)等同于 [-raft-protocol命令行标志](https://www.consul.io/docs/agent/options.html#_raft_protocol)。

* [raft\_snapshot\_threshold](https://www.consul.io/docs/agent/options.html" \l "raft_snapshot_threshold)等同于 [-raft-snapshot-threshold命令行标志](https://www.consul.io/docs/agent/options.html#_raft_snapshot_threshold)。

* [raft\_snapshot\_interval](https://www.consul.io/docs/agent/options.html" \l "raft_snapshot_interval)等同于 [-raft-snapshot-interval命令行标志](https://www.consul.io/docs/agent/options.html#_raft_snapshot_interval)。

* [reap](https://www.consul.io/docs/agent/options.html" \l "reap)这将控制Consul的子进程自动收集，如果Consul在Docker容器中以PID 1的形式运行，这将非常有用。如果没有指定，则Consul会自动收集子进程，如果它检测到它正在以PID 1运行。如果设置为true或false，则无论Consul的PID如何，它都会控制收割（强制分别开启或关闭） 。Consul 0.7.1中删除了该选项。对于Consul的更高版本，您将需要使用包装器收获流程，请参阅 [Consul Docker图像入口点脚本](https://github.com/hashicorp/docker-consul/blob/master/0.X/docker-entrypoint.sh) 以获取示例。如果您使用的是Docker 1.13.0或更高版本，则可以使用该命令的新--init选项，docker run并且docker将启用PID 1的初始化进程，以便为容器收集子进程。有关[Docker文档的](https://docs.docker.com/engine/reference/commandline/run/" \l "options)更多信息。

* [reconnect\_timeout](https://www.consul.io/docs/agent/options.html" \l "reconnect_timeout)这将控制从集群中彻底删除发生故障的节点需要多长时间。默认值为72小时，建议将其设置为至少为节点或网络分区的预期可恢复的最大停机时间的两倍。警告：将此时间设置得太低可能会导致Consul服务器在扩展节点故障或分区过程中从法定数中删除，这可能会使群集恢复复杂化。该值是一个带单位后缀的时间，可以是秒，分钟或小时的“s”，“m”，“h”。该值必须> = 8小时。

* [reconnect\_timeout\_wan](https://www.consul.io/docs/agent/options.html" \l "reconnect_timeout_wan)这是[reconnect\_timeout](https://www.consul.io/docs/agent/options.html" \l "reconnect_timeout)参数的WAN等效项，用于控制从WAN池中完全删除发生故障的服务器所需的时间。这也默认为72小时，并且必须> 8小时。

* [recursors](https://www.consul.io/docs/agent/options.html" \l "recursors)此标志提供用于递归解析查询（如果它们不在Consul的服务域内）的上游DNS服务器的地址。例如，节点可以直接使用Consul作为DNS服务器，并且如果该记录不在“领事”范围内。域，查询将在上游解决。从Consul 1.0.1开始，递归可以作为IP地址或go-sockaddr模板提供。IP地址按顺序解析，重复项被忽略。

* [rejoin\_after\_leave](https://www.consul.io/docs/agent/options.html" \l "rejoin_after_leave)等同于[-rejoin命令行标志](https://www.consul.io/docs/agent/options.html" \l "_rejoin)。

* [retry\_join](https://www.consul.io/docs/agent/options.html" \l "retry_join)- 相当于[-retry-join](https://www.consul.io/docs/agent/options.html" \l "retry-join)命令行标志。

* [retry\_interval](https://www.consul.io/docs/agent/options.html" \l "retry_interval)等同于 [-retry-interval命令行标志](https://www.consul.io/docs/agent/options.html#_retry_interval)。

* [retry\_join\_wan](https://www.consul.io/docs/agent/options.html" \l "retry_join_wan)等同于 [-retry-join-wan命令行标志](https://www.consul.io/docs/agent/options.html#_retry_join_wan)。每次尝试加入广域网地址列表，[retry\_interval\_wan](https://www.consul.io/docs/agent/options.html#_retry_interval_wan)直到至少有一个加入工作。

* [retry\_interval\_wan](https://www.consul.io/docs/agent/options.html" \l "retry_interval_wan)等同于 [-retry-interval-wan命令行标志](https://www.consul.io/docs/agent/options.html#_retry_interval_wan)。

* [segment](https://www.consul.io/docs/agent/options.html" \l "segment)（仅限企业）等同于 [-segment命令行标志](https://www.consul.io/docs/agent/options.html#_segment)。

* [segments](https://www.consul.io/docs/agent/options.html" \l "segments)（仅限企业）这是一个嵌套对象列表，它允许设置网段的绑定/通告信息。这只能在服务器上设置。有关更多详细信息，请参阅 [网络细分指南](https://www.consul.io/docs/guides/segments.html)。

* + [name](https://www.consul.io/docs/agent/options.html" \l "segment_name) - 细分受众群的名称。必须是长度介于1到64个字符之间的字符串。

* + [bind](https://www.consul.io/docs/agent/options.html" \l "segment_bind) - 用于分组的八卦图层的绑定地址。[-bind](https://www.consul.io/docs/agent/options.html" \l "_bind)如果未提供，则缺省为该值。

* + [port](https://www.consul.io/docs/agent/options.html" \l "segment_port) - 用于细分的八卦图层的端口（必需）。

* + [advertise](https://www.consul.io/docs/agent/options.html" \l "segment_advertise) - 用于分组的八卦图层的广告地址。[-advertise](https://www.consul.io/docs/agent/options.html" \l "_advertise)如果未提供，则缺省为该值。

* + [rpc\_listener](https://www.consul.io/docs/agent/options.html" \l "segment_rpc_listener)- 如果为true，则会[-bind](https://www.consul.io/docs/agent/options.html" \l "_bind)在rpc端口上的该段地址上启动单独的RPC侦听器。只有段的绑定地址与地址不同时才有效 [-bind](https://www.consul.io/docs/agent/options.html#_bind)。默认为false。

* [server](https://www.consul.io/docs/agent/options.html" \l "server)等同于 [-server命令行标志](https://www.consul.io/docs/agent/options.html#_server)。

* [non\_voting\_server](https://www.consul.io/docs/agent/options.html" \l "non_voting_server)- 相当于 [-non-voting-server命令行标志](https://www.consul.io/docs/agent/options.html#_non_voting_server)。

* [server\_name](https://www.consul.io/docs/agent/options.html" \l "server_name)提供时，将覆盖[node\_name](https://www.consul.io/docs/agent/options.html" \l "_node)TLS证书。它可以用来确保证书名称与我们声明的主机名相匹配。

* [session\_ttl\_min](https://www.consul.io/docs/agent/options.html" \l "session_ttl_min) 允许的最小会话TTL。这确保会话不会在TTL小于指定的限制时创建。建议将此限制保持在默认值以上，以鼓励客户发送频繁的心跳。默认为10秒。

* [skip\_leave\_on\_interrupt](https://www.consul.io/docs/agent/options.html" \l "skip_leave_on_interrupt)这类似于[leave\_on\_terminate](https://www.consul.io/docs/agent/options.html" \l "leave_on_terminate)但仅影响中断处理。当Consul收到一个中断信号（比如在终端上打Control-C）时，Consul会优雅地离开集群。将其设置为true禁用该行为。此功能的默认行为根据代理是否作为客户端或服务器运行而不同（在Consul 0.7之前默认值被无条件设置为false）。在客户端模式下的代理上，默认为false服务器模式下的代理，并且默认为true （即服务器上的Ctrl-C将服务器保留在群集中，因此是仲裁，并且客户端上的Ctrl-C将优雅地离开）。

* [start\_join](https://www.consul.io/docs/agent/options.html" \l "start_join)[-join](https://www.consul.io/docs/agent/options.html#_join)启动时指定节点地址的字符串数组。请注意，[retry\_join](https://www.consul.io/docs/agent/options.html" \l "retry_join)在自动执行Consul集群部署时，使用 可能更适合帮助缓解节点启动竞争条件。

* [start\_join\_wan](https://www.consul.io/docs/agent/options.html" \l "start_join_wan)[-join-wan](https://www.consul.io/docs/agent/options.html#_join_wan)启动时指定WAN节点地址的字符串数组。

* [telemetry](https://www.consul.io/docs/agent/options.html" \l "telemetry) 这是一个嵌套对象，用于配置Consul发送其运行时遥测的位置，并包含以下键：

* + [circonus\_api\_token](https://www.consul.io/docs/agent/options.html" \l "telemetry-circonus_api_token) 用于创建/管理支票的有效API令牌。如果提供，则启用度量标准管理。

* + [circonus\_api\_app](https://www.consul.io/docs/agent/options.html" \l "telemetry-circonus_api_app) 与API令牌关联的有效应用名称。默认情况下，它被设置为“consul”。

* + [circonus\_api\_url](https://www.consul.io/docs/agent/options.html" \l "telemetry-circonus_api_url) 用于联系Circonus API的基本URL。默认情况下，它被设置为“ <https://api.circonus.com/v2> ”。

* + [circonus\_submission\_interval](https://www.consul.io/docs/agent/options.html" \l "telemetry-circonus_submission_interval) 指标提交给Circonus的时间间隔。默认情况下，它被设置为“10s”（十秒）。

* + [circonus\_submission\_url](https://www.consul.io/docs/agent/options.html" \l "telemetry-circonus_submission_url)check.config.submission\_url来自先前创建的HTTPTRAP检查的Check API对象 的字段。

* + [circonus\_check\_id](https://www.consul.io/docs/agent/options.html" \l "telemetry-circonus_check_id)从先前创建的HTTPTRAP检查中 检查ID（不**检查包**）。check.\_cidCheck API对象中字段的数字部分。

* + [circonus\_check\_force\_metric\_activation](https://www.consul.io/docs/agent/options.html" \l "telemetry-circonus_check_force_metric_activation) 强制激活已存在且当前未激活的度量标准。如果启用了支票管理，则默认行为是在遇到新的指标时添加新指标。如果该指标已经存在于支票中，则**不会**被激活。此设置将覆盖该行为。默认情况下，它被设置为false。

* + [circonus\_check\_instance\_id](https://www.consul.io/docs/agent/options.html" \l "telemetry-circonus_check_instance_id) 唯一标识来自此实例的度量标准。当它们在基础架构内移动时，它可用于维护度量连续性，即瞬态或短暂实例。默认情况下，它被设置为主机名：应用程序名称（例如“host123：consul”）。

* + [circonus\_check\_search\_tag](https://www.consul.io/docs/agent/options.html" \l "telemetry-circonus_check_search_tag) 一个特殊的标签，当与实例ID结合使用时，有助于在未提供提交URL或检查ID时缩小搜索结果的范围。默认情况下，它被设置为service：application name（例如“service：consul”）。

* + [circonus\_check\_display\_name](https://www.consul.io/docs/agent/options.html" \l "telemetry-circonus_check_display_name) 指定一个名称以在创建时进行检查。该名称显示在Circonus UI Checks列表中。可用于Consul 0.7.2及更高版本。

* + [circonus\_check\_tags](https://www.consul.io/docs/agent/options.html" \l "telemetry-circonus_check_tags) 用逗号分隔的附加标签列表在创建时添加到支票中。可用于Consul 0.7.2及更高版本。

* + [circonus\_broker\_id](https://www.consul.io/docs/agent/options.html" \l "telemetry-circonus_broker_id) 创建新支票时使用的特定Circonus Broker的ID。broker.\_cidBroker API对象中字段的数字部分。如果启用指标管理并且未提供提交URL和检查ID，则将尝试使用实例ID和搜索标记搜索现有检查。如果找不到，则会创建一个新的HTTPTRAP检查。默认情况下，不会使用此选项，并选择随机企业代理或默认的Circonus Public Broker。

* + [circonus\_broker\_select\_tag](https://www.consul.io/docs/agent/options.html" \l "telemetry-circonus_broker_select_tag) 当未提供经纪人代码时，将使用特殊标签选择Circonus经纪人。这个最好的用途是作为代理应该基于针对所使用的提示，其中该特定的实例正在运行（例如一个特定的地理位置或数据中心，DC：SFO）。默认情况下，这是留空，不使用。

* + [disable\_hostname](https://www.consul.io/docs/agent/options.html" \l "telemetry-disable_hostname) 这将控制是否在计算机主机名的前面加上运行时间遥测，默认为false。

* + [dogstatsd\_addr](https://www.consul.io/docs/agent/options.html" \l "telemetry-dogstatsd_addr)这提供了格式中DogStatsD实例的地址host:port。DogStatsD是statsd协议兼容的风格，增加了用标签和事件信息修饰指标的功能。如果提供，领事将发送各种遥测信息到该实例进行聚合。这可以用来捕获运行时信息。

* + [dogstatsd\_tags](https://www.consul.io/docs/agent/options.html" \l "telemetry-dogstatsd_tags)这提供了将被添加到发送到DogStatsD的所有遥测包的全局标签列表。它是一个字符串列表，其中每个字符串看起来像“my\_tag\_name：my\_tag\_value”。

* + [filter\_default](https://www.consul.io/docs/agent/options.html" \l "telemetry-filter_default) 这将控制是否允许过滤器未指定的度量标准。默认为true，这将允许在没有提供过滤器时的所有指标。如果设置为false不使用过滤器，则不会发送指标。

* + [metrics\_prefix](https://www.consul.io/docs/agent/options.html" \l "telemetry-metrics_prefix) 写入所有遥测数据时使用的前缀。默认情况下，它被设置为“consul”。这是在Consul 1.0中添加的。对于之前版本的Consul，使用statsite\_prefix相同结构中的配置选项。由于此前缀适用于所有遥测提供商，因此它已重新命名为Consul 1.0，而不仅仅是statsite。

* + [prefix\_filter](https://www.consul.io/docs/agent/options.html" \l "telemetry-prefix_filter) 这是一个过滤规则列表，适用于通过前缀允许/屏蔽指标，格式如下：
  + [
  + "+consul.raft.apply",
  + "-consul.http",
  + "+consul.http.GET"
  + ]

前导的“ **+** ”将使用给定前缀的任何度量标准，并且前导“ **-** ”将阻止它们。如果两个规则之间有重叠，则更具体的规则优先。如果多次列出相同的前缀，则阻塞将优先。

* + [prometheus\_retention\_time](https://www.consul.io/docs/agent/options.html" \l "telemetry-prometheus_retention_time) 如果该值大于0s（缺省值），则可以使[Prometheus](https://prometheus.io/)导出度量标准。持续时间可以使用持续时间语义来表示，并将在指定的时间内汇总所有计数器（这可能会影响Consul的内存使用情况）。此参数的价值至少是普罗米修斯刮擦间隔的2倍，但您也可能需要很长的保留时间，例如几天（例如744h才能保留至31天）。使用prometheus获取指标然后可以使用/v1/agent/metrics?format=prometheusURL 执行，或者通过发送值为Accept的Accept头来text/plain; version=0.0.4; charset=utf-8 执行/v1/agent/metrics（如普罗米修斯所做的那样）。格式与普罗米修斯本身兼容。在此模式下运行时，建议启用此选项[disable\_hostname](https://www.consul.io/docs/agent/options.html#telemetry-disable_hostname)以避免使用主机名的前缀度量标准。

* + [statsd\_address](https://www.consul.io/docs/agent/options.html" \l "telemetry-statsd_address)这以格式提供statsd实例的地址host:port。如果提供，领事将发送各种遥测信息到该实例进行聚合。这可以用来捕获运行时信息。这仅发送UDP数据包，可以与statsd或statsite一起使用。

* + [statsite\_address](https://www.consul.io/docs/agent/options.html" \l "telemetry-statsite_address)这提供了格式中的一个statsite实例的地址host:port。如果提供，领事将汇集各种遥测信息到该实例。这可以用来捕获运行时信息。这通过TCP流，只能用于statsite。

* [syslog\_facility](https://www.consul.io/docs/agent/options.html" \l "syslog_facility)何时 [enable\_syslog](https://www.consul.io/docs/agent/options.html" \l "enable_syslog)提供，这将控制向哪个设施发送消息。默认情况下，LOCAL0将被使用。

* [tls\_min\_version](https://www.consul.io/docs/agent/options.html" \l "tls_min_version)在Consul 0.7.4中添加，它指定了TLS的最低支持版本。接受的值是“tls10”，“tls11”或“tls12”。这默认为“tls10”。警告：TLS 1.1及更低版本通常被认为不太安全; 避免使用这些如果可能。这将在Consul 0.8.0中更改为默认值“tls12”。

* [tls\_cipher\_suites](https://www.consul.io/docs/agent/options.html" \l "tls_cipher_suites)在Consul 0.8.2中添加，它将支持的密码组列表指定为逗号分隔列表。[源代码中](https://github.com/hashicorp/consul/blob/master/tlsutil/config.go" \l "L363)提供了所有支持的密码套件列表。

* [tls\_prefer\_server\_cipher\_suites](https://www.consul.io/docs/agent/options.html" \l "tls_prefer_server_cipher_suites) 在Consul 0.8.2中添加，这将导致Consul更喜欢服务器的密码套件而不是客户端密码套件。

* [translate\_wan\_addrs](https://www.consul.io/docs/agent/options.html" \l "translate_wan_addrs)如果设置为true，Consul 在为远程数据中心中的节点提供DNS和HTTP请求时，会优先使用配置的[WAN地址](https://www.consul.io/docs/agent/options.html" \l "_advertise-wan)。这允许使用其本地地址在其自己的数据中心内访问该节点，并使用其WAN地址从其他数据中心到达该节点，这在混合网络的混合设置中很有用。这是默认禁用的。

从Consul 0.7和更高版本开始，响应HTTP请求的节点[地址](https://www.consul.io/docs/agent/options.html" \l "_advertise-wan)在查询远程数据中心中的节点时也将优选节点配置的[WAN地址](https://www.consul.io/docs/agent/options.html#_advertise-wan)。一个[X-Consul-Translate-Addresses](https://www.consul.io/api/index.html#translated-addresses)当翻译被启用，以帮助客户知道地址可以被翻译标题将出现在所有响应。在TaggedAddresses响应中域也有一个lan地址，需要该地址的知识，无论翻译的客户。

以下端点转换地址：

* + [/v1/catalog/nodes](https://www.consul.io/api/catalog.html#catalog_nodes)
  + [/v1/catalog/node/<node>](https://www.consul.io/api/catalog.html#catalog_node)
  + [/v1/catalog/service/<service>](https://www.consul.io/api/catalog.html#catalog_service)
  + [/v1/health/service/<service>](https://www.consul.io/api/health.html#health_service)
  + [/v1/query/<query or name>/execute](https://www.consul.io/api/query.html#execute)

* [ui](https://www.consul.io/docs/agent/options.html" \l "ui)- 相当于[-ui](https://www.consul.io/docs/agent/options.html" \l "_ui) 命令行标志。

* [ui\_dir](https://www.consul.io/docs/agent/options.html" \l "ui_dir)- 相当于 [-ui-dir](https://www.consul.io/docs/agent/options.html#_ui_dir)命令行标志。从Consul版本0.7.0及更高版本开始，此配置密钥不是必需的。指定此配置键将启用Web UI。没有必要指定ui-dir和ui。指定两者都会导致错误。

* [unix\_sockets](https://www.consul.io/docs/agent/options.html" \l "unix_sockets) - 这可以调整Consul创建的Unix域套接字文件的所有权和权限。只有在HTTP地址配置了unix://前缀时才使用域套接字。

需要注意的是，这个选项可能对不同的操作系统有不同的影响。Linux通常会观察套接字文件权限，而许多BSD变体会忽略套接字文件本身的权限。在特定的发行版上测试此功能非常重要。此功能目前在Windows主机上无法使用。

以下选项在此构造内有效，并全面应用于Consul创建的所有套接字：

* + [user](https://www.consul.io/docs/agent/options.html" \l "user) - 将拥有套接字文件的用户的名称或ID。

* + [group](https://www.consul.io/docs/agent/options.html" \l "group) - 套接字文件的组ID标识。该选项目前仅支持数字ID。

* + [mode](https://www.consul.io/docs/agent/options.html" \l "mode) - 在文件上设置的权限位。

* [verify\_incoming](https://www.consul.io/docs/agent/options.html" \l "verify_incoming)- 如果设置为true，Consul要求所有传入连接都使用TLS，并且客户端提供证书颁发机构从[ca\_file](https://www.consul.io/docs/agent/options.html" \l "ca_file)or中签名的证书[ca\_path](https://www.consul.io/docs/agent/options.html" \l "ca_path)。这适用于服务器RPC和HTTPS API。默认情况下，这是错误的，Consul不会强制使用TLS或验证客户的真实性。

* [verify\_incoming\_rpc](https://www.consul.io/docs/agent/options.html" \l "verify_incoming_rpc)- 如果设置为true，Consul要求所有传入的RPC连接都使用TLS，并且客户端提供由证书颁发机构从[ca\_file](https://www.consul.io/docs/agent/options.html" \l "ca_file)or中签名的证书[ca\_path](https://www.consul.io/docs/agent/options.html" \l "ca_path)。默认情况下，这是错误的，Consul不会强制使用TLS或验证客户的真实性。

* [verify\_incoming\_https](https://www.consul.io/docs/agent/options.html" \l "verify_incoming_https)- 如果设置为true，则Consul要求所有传入的HTTPS连接都使用TLS，并且客户端提供由证书颁发机构从[ca\_file](https://www.consul.io/docs/agent/options.html" \l "ca_file)or中签名的证书[ca\_path](https://www.consul.io/docs/agent/options.html" \l "ca_path)。默认情况下，这是错误的，Consul不会强制使用TLS或验证客户的真实性。要启用HTTPS API，您必须通过[ports](https://www.consul.io/docs/agent/options.html" \l "ports)配置定义HTTPS端口。默认情况下，HTTPS被禁用。

* [verify\_outgoing](https://www.consul.io/docs/agent/options.html" \l "verify_outgoing)- 如果设置为true，则Consul要求所有传出连接都使用TLS，并且服务器提供由证书颁发机构从[ca\_file](https://www.consul.io/docs/agent/options.html" \l "ca_file)or中签名的证书[ca\_path](https://www.consul.io/docs/agent/options.html" \l "ca_path)。默认情况下，这是错误的，Consul不会使用TLS进行传出连接。这适用于客户端和服务器，因为两者都会建立传出连接。

* [verify\_server\_hostname](https://www.consul.io/docs/agent/options.html" \l "verify_server_hostname) - 如果设置为true，则Consul会验证所有传出连接，即服务器提供的TLS证书与“server。<datacenter>。<domain>”主机名匹配。这意味着verify\_outgoing。默认情况下，这是错误的，并且Consul不验证证书的主机名，只验证它是由受信任的CA签署的。此设置对于防止受损客户端作为服务器重新启动很重要，从而能够执行MITM攻击或添加为Raft对等设备。这在0.5.1中是新的。

* [watches](https://www.consul.io/docs/agent/options.html" \l "watches) - Watches是手表规范的列表，允许在更新特定数据视图时自动调用外部进程。有关更多详情，请参阅 [手表文档](https://www.consul.io/docs/agent/watches.html)。手表可以在配置重新加载时修改。

## [»](https://www.consul.io/docs/agent/options.html" \l "ports-used)使用的端口

Consul最多需要6个不同的端口才能正常工作，有些使用TCP，UDP或两种协议。下面我们记录每个端口的要求。

* 服务器RPC（默认8300）。这由服务器用来处理来自其他代理的传入请求。仅限TCP。
* Serf LAN（默认8301）。这是用来处理局域网中的八卦。所有代理都需要。TCP和UDP。
* Serf WAN（默认8302）。这被服务器用来在WAN上闲聊到其他服务器。TCP和UDP。从Consul 0.8开始，建议通过端口8302在LAN接口上为TCP和UDP启用服务器之间的连接，以及WAN加入泛滥功能。另见： [Consul 0.8.0 CHANGELOG](https://github.com/hashicorp/consul/blob/master/CHANGELOG.md#080-april-5-2017)和[GH-3058](https://github.com/hashicorp/consul/issues/3058)
* HTTP API（默认8500）。这被客户用来与HTTP API交谈。仅限TCP。
* DNS接口（默认8600）。用于解析DNS查询。TCP和UDP。

## [»](https://www.consul.io/docs/agent/options.html" \l "reloadable-configuration)可重新加载配置

重新加载配置不会重新加载所有配置项目。重新加载的项目包括：

* Log level
* Checks
* Services
* Watches
* HTTP Client Address
* [Node Metadata](https://www.consul.io/docs/agent/options.html#node_meta)
* [Metric Prefix Filter](https://www.consul.io/docs/agent/options.html#telemetry-prefix_filter)
* [Discard Check Output](https://www.consul.io/docs/agent/options.html#discard_check_output)