# consul 文档

## 安装

## 更新

## 词汇表

此页面收集了Consul和Consul Enterprise文档中使用的一些技术术语的简要定义，以及Consul社区中交流中经常出现的一些术语。

### Agent

代理是Consul集群的每个成员上的长时间运行守护程序。 它是通过运行consul agent来启动的。 代理能够以客户端或服务器模式运行。 由于所有节点都必须运行代理，因此将节点称为客户端或服务器更简单，但代理还有其他实例。 所有代理都可以运行DNS或HTTP接口，并负责运行检查并保持服务同步。

### Client

客户端是将所有RPC转发到服务器的代理。客户端相对来说是无状态的。客户端执行的唯一后台活动是参与局域网（LAN）gossip池。这具有最小的资源开销，并且只消耗少量的网络带宽。

### Server

服务器是一种具有扩展职责集的代理，包括参与Raft 仲裁、维护集群状态、响应RPC查询、与其他数据中心（datacenters）交换WAN流言（gossip）以及将查询转发给leader或远程数据中心。

### Datacenter

我们将数据中心定义为一个私有、低延迟和高带宽的网络环境。这不包括穿越公共互联网的通信，但就我们的目的而言，一个EC2区域内的多个可用区域将被视为单个数据中心的一部分。

**注释：**亚马逊EC2

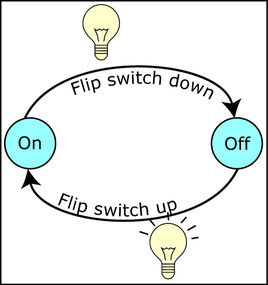
*亚马逊弹性计算云（EC2，Elastic Compute Cloud）是一个让使用者可以租用云端电脑运行所需应用的系统。EC2借由提供Web服务的方式让使用者可以弹性地运行自己的Amazon机器映像档，使用者将可以在这个虚拟机器上运行任何自己想要的软件或应用程式。提供可调整的云计算能力。它旨在使开发者的网络规模计算变得更为容易。*

*--来自百度百科*

### Consensus（协商一致/共识）

在我们的文档中使用时，我们使用共识来表示**就选举的领导者（elected leader）以及对交易的顺序**达成一致。 由于这些事务应用于有限状态机（[finite-state machine](https://en.wikipedia.org/wiki/Finite-state_machine)），因此我们对共识的定义意味着复制状态机的一致性。 维基百科（[Wikipedia](https://en.wikipedia.org/wiki/Consensus_(computer_science))）上对共识进行了更加详细的描述，我们的实现描述在[此处](https://www.consul.io/docs/internals/consensus.html)。

**注释：**有限状态机

*有限状态机，（英语：Finite-state machine, FSM），又称有限状态自动机，简称状态机，是表示有限个状态以及在这些状态之间的转移和动作等行为的数学模型。*

*--来自百度百科*

### Gossip

Consul建立在[Serf](https://www.serf.io/)之上，它提供了一种用于多种用途的完整的gossip协议（[gossip protocol](https://en.wikipedia.org/wiki/Gossip_protocol)）。 Serf提供会员资格，故障检测和事件广播的功能。 在gossip文档（ [gossip documentation](https://www.consul.io/docs/internals/gossip.html)）种我们有更加详尽的描述。 在这里，只要知道gossip协议用于节点到节点（node-to-node）的通讯就已经足够，其主要是使用UDP协议。

### LAN Gossip

指局域网（LAN）gossip池，其中包含全部位于同一局域网或数据中心的节点。

**注意：**既包含server也包含client

### WAN Gossip

指仅包含服务器（server）的WAN gossip池。 这些服务器主要位于不同的数据中心，通常通过互联网或广域网进行通信。

### RPC

远程过程调用（**R**emote **P**rocedure **C**all）。 这是一种允许客户端向服务器发送请求的请求/响应机制。

## Consul Internals（内部结构）

本节介绍了consul 的一些内部细节。 理解Consul的内部结构对于在生产中成功使用它是必要的。

请查看以下文档以了解Consul的工作原理。

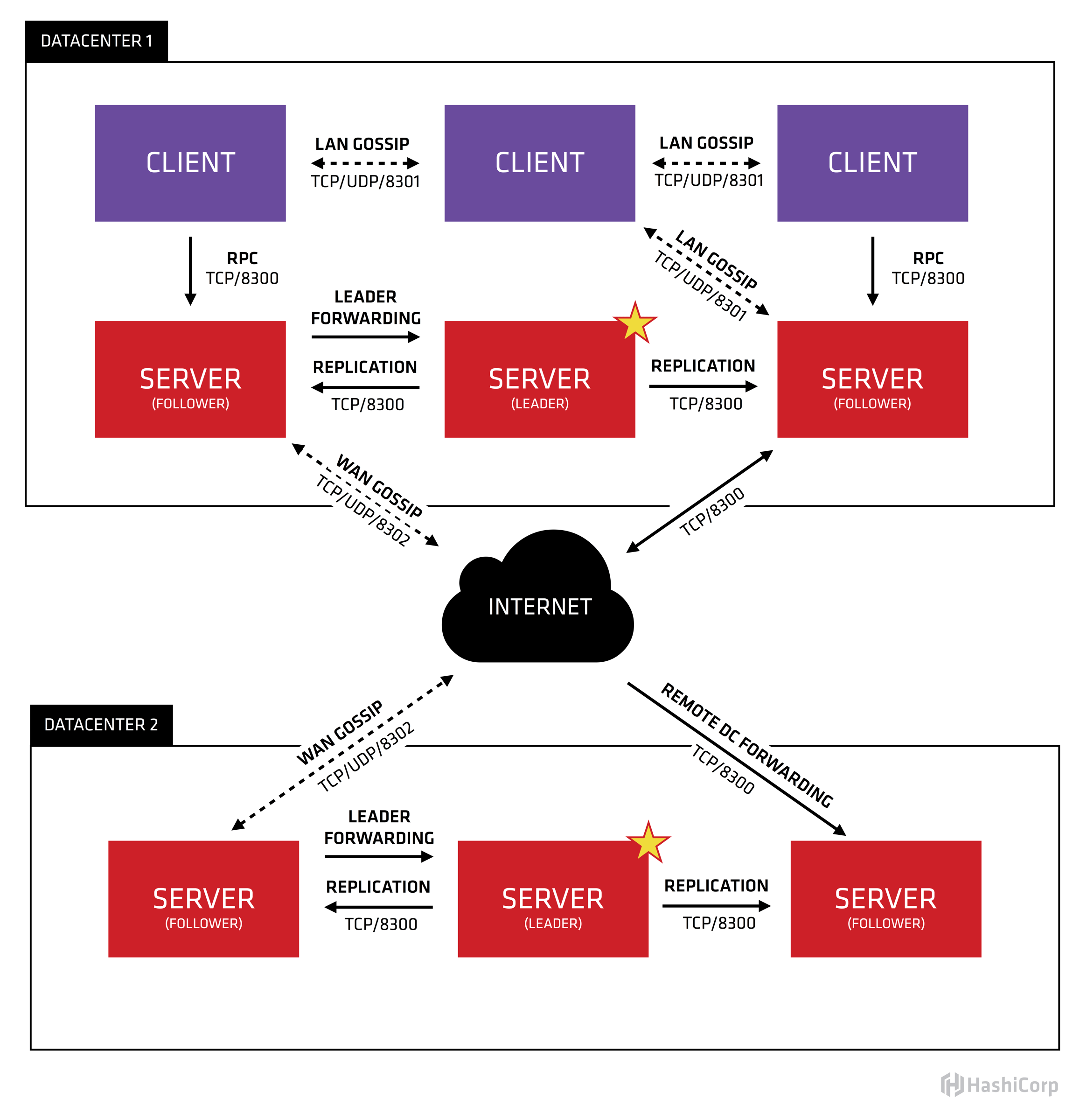
### [Architecture](https://www.consul.io/docs/internals/architecture.html)

Consul是一个复杂的系统，有许多不同的移动（moving）部件。 为了帮助Consul的用户和开发人员形成其工作原理的心理模型，该文档描述其系统架构。

**在讨论这个架构之前，我们建议阅读词汇表部分，这样对所讨论的内容将更有帮助。**

#### 10,000 foot view（总体预览）

从10,000英尺的高度，Consul的架构看起来像这样：

[](https://www.consul.io/assets/images/consul-arch-420ce04a.png)

让我们分解这个图像并描述每一块。 首先，我们可以看到有两个数据中心，标记为“数据中心1”和“数据中心2”。 Consul为多个数据中心提供一流的支持，并期望这是常见的情况。

在每个数据中心，我们拥有一个客户端和服务器和混合体。预计将有3至5个服务器（server）。这是在失败和可用性之间取得平衡，因为随着增加更多的机器，协商一致逐渐变慢。然而，客户端的数量是没有限制的，他们可以很容易地扩展到成千上万的客户端（client）。

数据中心中的所有代理（agent）都参与了gossip协议。这意味着有一个gossip池，它包含一个给定数据中心的所有代理（agent）。这有几个目的：首先，不需要使用服务器地址配置客户端；发现是自动完成的。第二，检测代理故障的工作不是放在server上，而是分布式的。这使得故障检测比幼稚的心跳方案更具可伸缩性。它还为节点提供故障检测；如果agent不可达，则该节点可能已经发生故障。第三，它被用作一个消息层，用于在发生领导人选举（leader election）等重要事件时发出通知。

**注意：**连接类型与连接个数

每个数据中心中的server都是单个Raft对等集的一部分。 这意味着他们共同选举一个领导者（leader），这个选择的server具有额外职责。领导者（leader）负责处理所有查询和交易。 作为共识协议（[consensus protocol](https://www.consul.io/docs/internals/consensus.html)）的一部分，还必须将事务复制到所有对等体。 由于此要求，当非领导者（non-leader）服务器收到RPC请求时，它会将其转发给群集leader。

服务器代理（server agent）也作为WAN gossip池的一部分运行。 此池与LAN池不同，因为它针对较高的Internet延迟进行了优化，并且预计仅包含其他Consul服务器代理（server agent）。 此池的目的是允许数据中心以低触摸方式发现彼此。 在线创建新的数据中心就像加入现有的WAN gossip 池一样简单。 由于服务器（server）都在此池中运行，因此它还支持跨数据中心请求。 当服务器（server）收到对不同数据中心的请求时，它会将其转发到正确数据中心的随机服务器（server）。 然后该服务器（server）可以转发给本地领导者（leader）。

这导致数据中心之间的耦合非常低，但由于故障检测，连接缓存和多路复用，跨数据中心请求相对快速且可靠。

通常，不会在不同的Consul数据中心之间复制数据。 当对另一个数据中心中的资源发出请求时，本地Consul服务器（server）会将RPC请求转发给该资源的远程Consul服务器（server）并返回结果。 **如果远程数据中心不可用，那么这些资源也将不可用**，但这不会影响本地数据中心。 在某些特殊情况下，可以复制有限的数据子集，例如使用Consul的内置ACL复制（[ACL replication](https://learn.hashicorp.com/consul/day-2-operations/acl-replication)）功能，或者像[consul-replicate](https://github.com/hashicorp/consul-replicate)这样的外部工具。

在某些地方，客户端代理可能会缓存来自服务器的数据，以便在本地提供数据以提高性能和可靠性。示例包括连接证书和意图（intentions），它们允许客户端代理在不需要往返服务器的情况下对入站连接请求进行本地决策。一些API端点还支持可选的结果缓存。这有助于可靠性，因为本地代理可以继续响应某些查询，如服务发现或从缓存连接授权，即使与服务器的连接中断或服务器暂时不可用。

#### Getting in depth（深入）

在这一点上，我们已经介绍了Consul的高级架构，但每个子系统还有更多细节。 共识协议（[consensus protocol](https://www.consul.io/docs/internals/consensus.html)）与gossip协议（[gossip protocol](https://www.consul.io/docs/internals/gossip.html)）同样有详细的文档说明。 此外还有其使用的安全模型和协议的文档说明。

有关其他详细信息，请查阅源代码，在IRC中询问或联系邮件列表。

* [Consensus Protocol](https://www.consul.io/docs/internals/consensus.html)
* [Gossip Protocol](https://www.consul.io/docs/internals/gossip.html)
* [Network Coordinates](https://www.consul.io/docs/internals/coordinates.html)
* [Sessions](https://www.consul.io/docs/internals/sessions.html)
* [Anti-Entropy](https://www.consul.io/docs/internals/anti-entropy.html)
* [Security Model](https://www.consul.io/docs/internals/security.html)

在部署生产数据中心之前，您还应该熟悉Jepsen测试（[Jepsen testing](https://www.consul.io/docs/internals/jepsen.html)）。

**注释：**Jepsen测试（吉布森测试）

[jepsen](https://link.jianshu.com/?t=https://github.com/aphyr/jepsen)是一个分布式测试库，我们可以使用它对某个分布式系统执行一系列操作，并最终验证这些操作是否正确执行