



链滴

基本概念 (2) : 节点, 集群, 分片及副本

作者: [GentLeLe](#)

原文链接: <https://ld246.com/article/1677590506766>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

分布式系统的可用性和扩展性

- 高可用性
 - 服务可用性：允许有节点停止服务
 - 数据可用性：部分节点丢失，不会丢失数据
- 可扩展性
 - 请求量提升/数据的不断增长（将数据分布到所有节点上，实现水平扩展）

ES分布式特性

- ES的分布式架构带来的好处
 - 存储的水平扩容
 - 提供系统的可用性，部分节点停止服务，整个集群的服务不受影响
- ES的分布式架构
 - 不通集群通过不同的名字来区分，默认名称为：elasticsearch
 - 通过配置文件修改，或者在命令行中 `-E cluster.name=panshaller` 进行设定
 - 一个集群可以有一个或者多个节点

节点

- 节点是一个ES的实例
 - 本质上就是==一个JAVA进程==
 - ==一台机器上可以运行多个ES进程==，但是生产环境一般建议一台机器上只运行一个ES实例
- 每一个节点都有名字，通过配置文件配置，或者启动时候，`-E node.name=node1`
- 每一个节点在启动之后，==会分配一个UID，保存在data目录==下

Master-eligible nodes 和Master Node

- 每个节点启动后，默认就是一个Master eligible 节点
 - 可以设置`node.master: false` 禁止
- Master-eligible 节点可以参加选主流程，成为Master 节点
- 当第一个节点启动时候， 它会将自己选举成Master 节点
- 每个节点上都保存了集群的状态，只有Master 节点才能修改集群的状态信息
 - 集群状态 (Cluster State) ，维护了一个集群中必要的信息
 - 所有的节点信息
 - 所有的索引和其相关的Mapping 与Setting 信息

- 分片的路由信息
 - 任意节点都能修改信息会导致数据的不一致性

Data Node 和 Coordinating Node

- Data Node
 - 可以保存数据的节点，叫做Data Node。负责保存分片数据。==在数据扩展上起到了至关重要的作用==。
- Coordinating (协调) Node
 - 负责接收Client的请求，将请求分发到合适的节点，最终把结果汇聚到一起
 - ==每个节点默认==都起到了Coordinating Node的职责

其他类型节点

- ingest Node
 - [Elasticsearch的ETL利器——Ingest节点](#)
- Hot Node 和 Warm Node
 - 主要==从硬件成本==出发，不同硬件配置的Data Node, 用来实现Hot & Warm 架构，降低群部署的成本
 - 比如比较【旧】的日志数据可以存储在warm Node（硬件配置较低）中，节约成本
- Machine Learning Node
 - 负责跑机器学习的Job, 用来做异常检测
- Tribe Node
 - （ 5.3 开始使用Cross Cluster Search) Tribe Node 连接到不同的Elasticsearch 集群，并且将这些集群当成一个单独的集群处理

配置节点的类型

- 开发环境中一个节点可以承担多种角色
- 生产环境中，应该设置单一的角色节点

| 节点类型 | 配置参数 | 默认值 |
|---|-------------|------|
| master eligible | node.master | true |
| data | node.data | true |
| ingest | node.ingest | true |
| coordinating only | no | 每个节点 |
| 认都是coordinating节点，只要设置其他类型全部为false，自动达到效果 | | |
| machine learning | node.ml | true |

(需要enable x-pack)

分片

- 主分片 (==Primary Shared==)
 - 用于解决数据水平扩展的问题。通过主分片，可以将数据分布到集群内的所有节点之上
 - 一个分片是一个运行的Lucene的实例
 - 主分片数在索引创建时指定，后续不允许修改，除非进行Reindex操作
- 副本分片 (==Replica Shared==)
 - 用于解决数据高可用的问题。副本分片是主分片的考本
 - 副本分片数，可以动态调整
 - 增加副本数，还可以在在一定程度上提高服务的可用性（读取的吞吐）

示例：

```
PUT /cars
{
  "setting":{
    "number_of_shards":3
    "number_of_replicas":1
  }
}
```

```
<iframe src="/widgets/Daily Views" data-src="/widgets/Daily Views" data-subtype="widget"
order="0" frameborder="no" framespacing="0" allowfullscreen="true" style="width: 863px;
eight: 629px;"></iframe>
```

```
<iframe src="/widgets/custom.json" data-src="/widgets/custom.json" data-subtype="widget"
border="0" frameborder="no" framespacing="0" allowfullscreen="true"></iframe>
```

```
<iframe src="/widgets/drawio" data-src="/widgets/drawio" data-subtype="widget" border=
0" frameborder="no" framespacing="0" allowfullscreen="true" style="width: 861px; height: 5
1px;"></iframe>
```