

1、consistency, one (primary shard) , all (all shard) , quorum (default)

我们在发送任何一个增删改操作的时候，比如说 `put /index/type/id`, 都可以带上一个 consistency 参数，指明我们想要的写一致性是什么？

比如：`put /index/type/id?consistency=quorum`

one：要求我们这个写操作，只有一个 primary shard 是 active 活跃可用的，就可以执行

all：要求我们这个写操作，必须所有的 primary shard 和 replica shard 都是活跃的，才可以执行写这个操作

quorum：默认的值，要求所有的 shard 中，必须大部分的 shard 都是活跃可用的，才可以执行这个操作

2、quorum机制，写之前必须确保大多数 shard 都可用， $(\text{int})((\text{primary} + \text{number_of_replicas})/2 + 1)$ ，当 $\text{number_of_replicas} > 1$ 时才生效

$\text{quorum} = \text{int}((\text{primary} + \text{number_of_replicas}) / 2 + 1)$

举个例子：3个 primary shard , $\text{number_of_replicas} = 2$, 总共就是 $3 + 6 = 9$ 个 shard

$\text{quorum} = (\text{int})((3 + 2) / 2 + 1) = 4$

所以，要求9个shard中至少有4个shard是active状态的时候，才能执行写这个操作。

(即：大多数 shard 是活跃状态的)

3、如果节点数少于 quorum 数量，可能导致 quorum 不齐全，进而导致无法执行任何写操作

3个 primary shard , $\text{replica} = 1$ 要求至少是3个shard是active，按照之前学习的shard & replica机制，必须在不同的节点上，如果说只有1台机器的话，是不是有可能出现说，3个 shard 都没法分配齐全，此时就可能会出现写操作无法执行的情况

1个 primary shard , $\text{replica}=3$, $\text{quorum} = ((1+3)/2) + 1 = 3$, 要求1个 primary shard + 3个 replica shard = 4个shard , 其中必须有3个 shard 是要处于 active状态的。如果这个时候只有两台机器的话，会出现怎样的情况呢？

es 提供了一种特殊的处理场景，就是说当 $\text{number_of_replicas} > 1$ 时候才生效，因为假如

说，你就一个 primary shard， replica = 1， 此时就 2 个 shard， $((1+1)/2) + 1 = 2$ ， 要求必须有 2 个 shard 是活跃的， 但是可能就只有 1 个 node， 此时就是 1 个 shard 是活跃的， 如果你不特殊处理的话， 导致我们的单点集群就无法工作

4、 quorum 不齐全时候， wait， 默认 1 分钟， timeout， 100， 30s

等待期间， 期待活跃的 shard 数量可以增加， 最后实在不行就会 timeout

我们其实可以在写操作的时候， 加一个 timeout 参数， 比如： put /index/type/id?

timeout=30, 这个就是自己去设定 quorum 不齐全的时候， es 的 timeout 时长， 可以缩短， 也可以增长