> https://zhuanlan.zhihu.com/p/44061870
> https://zhuanlan.zhihu.com/p/34830403

选举过程

1. 发起选举

master选举当然是由master-eligible节点发起，当一个master-eligible节点发现满足以下条件时发起选举：

- 该master-eligible节点的当前状态不是master。
- 该master-eligible节点通过ZenDiscovery模块的ping操作询问其已知的集群其他节点，没有任何节点连接到master。
- 包括本节点在内，当前已有超过minimum_master_nodes个节点没有连接到master。

总结一句话，即当一个节点发现包括自己在内的多数派的master-eligible节点认为集群没有master时，就可以发起master选举。

2. 选举过程

- 当前node获取集群中其他节点的ZenPing.PingResponse列表。
- pingResponses中筛选出其他节点中标记的master node集合：activeMasters
- 构造候选master列表：masterCandidates。（通过配置选项node.master（true）来指定的）
- 先根据节点的clusterStateVersion比较，clusterStateVersion越大，优先级越高。clusterStateVersion相同时，进入compareNodes，其内部按照节点的Id比较(Id为节点第一次启动时随机生成)。

3. 选举成功

得票最多的 node 成为 master，其他节点申请 join 集群，全部 join 成功则选举完成。

选举策略源码

```
1  public MasterCandidate electMaster(Collection<MasterCandidate> candidate
s) {
2  List<MasterCandidate> sortedCandidates = new ArrayList<>(candidates);
3  sortedCandidates.sort(MasterCandidate::compare);
4  return sortedCandidates.get(0);
```

```
}

public static int compare(MasterCandidate c1, MasterCandidate c2) {
int ret = Long.compare(c2.clusterStateVersion, c1.clusterStateVersion);
if (ret == 0) {
 ret = compareNodes(c1.getNode(), c2.getNode());
 }
 return ret;
 }
```

总结一下：

1. 当clusterStateVersion越大，优先级越高。这是为了保证新Master拥有最新的clusterState(即集群的meta)，避免已经commit的meta变更丢失。因为Master当选后，就会以这个版本的clusterState为基础进行更新。(一个例外是集群全部重启，所有节点都没有meta，需要先选出一个master，然后master再通过持久化的数据进行meta恢复，再进行meta同步)。
2. 当clusterStateVersion相同时，节点的Id越小，优先级越高。即总是倾向于选择Id小的Node，这个Id是节点第一次启动时生成的一个随机字符串。之所以这么设计，应该是为了让选举结果尽可能稳定，不要出现都想当master而选不出来的情况。