

## 目 标

1. 了解Zookeeper的作用与基本原理
2. 掌握Zookeeper的应用方式
3. 掌握Zookeeper 在Kafka 集群中的作用及工作原理

## 一、什么是Zookeeper

- ◆ 分布式**协调**服务
- ◆ 为分布式应用而生

某科的词条

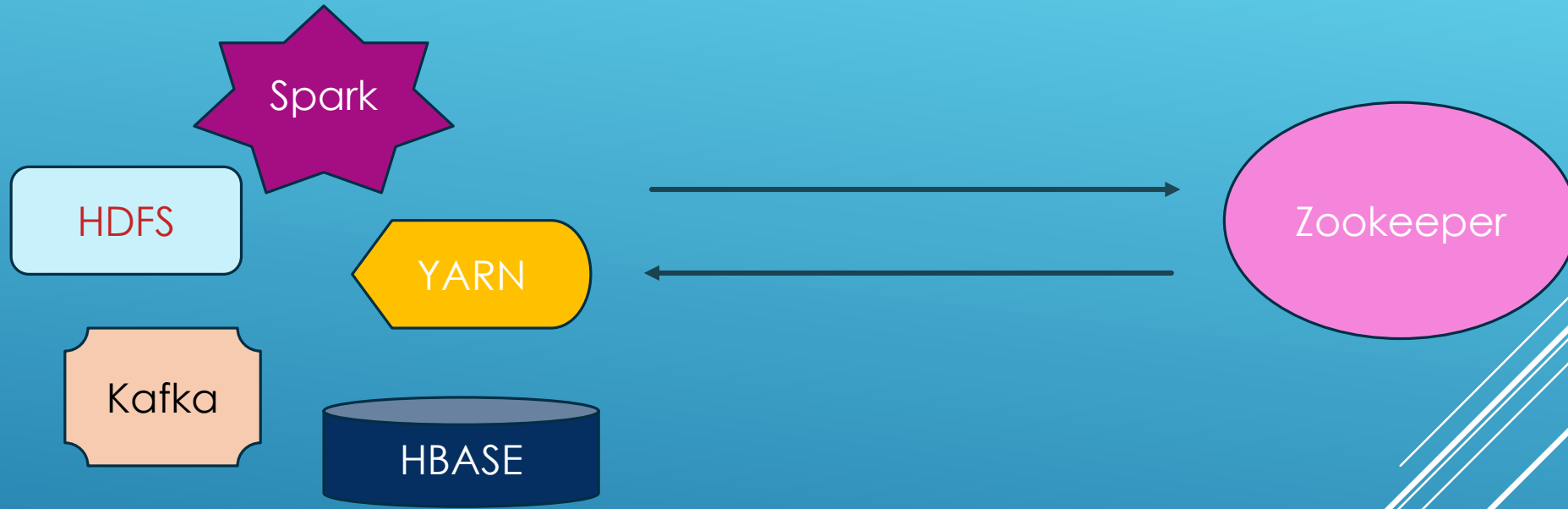
zookeeper

 编辑

ZooKeeper是一个**分布式的**，开放源码的**分布式应用程序协调服务**，是Google的Chubby一个**开源**的实现，是Hadoop和Hbase的重要组成部分。它是一个为分布式应用提供一致性服务的软件，提供的功能包括：**配置维护、域名服务、分布式同步、组服务等。**

  收藏  386  21

## 一、很多地方都可以有它的身影

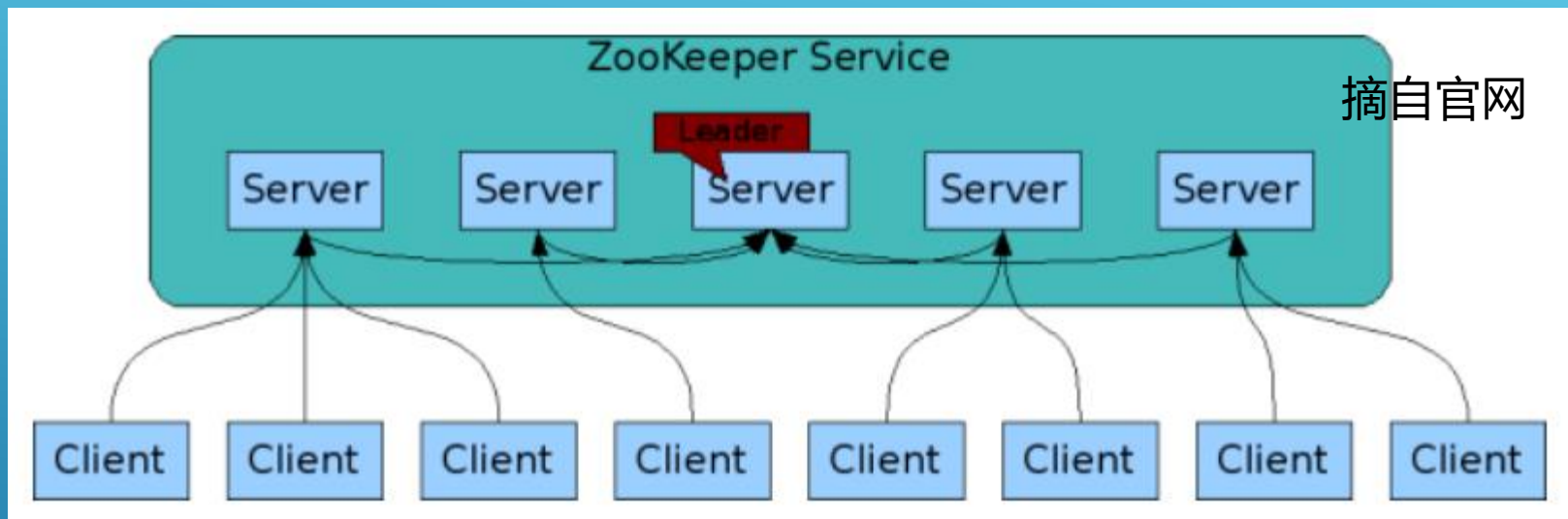


## 二、它是用来干嘛的？

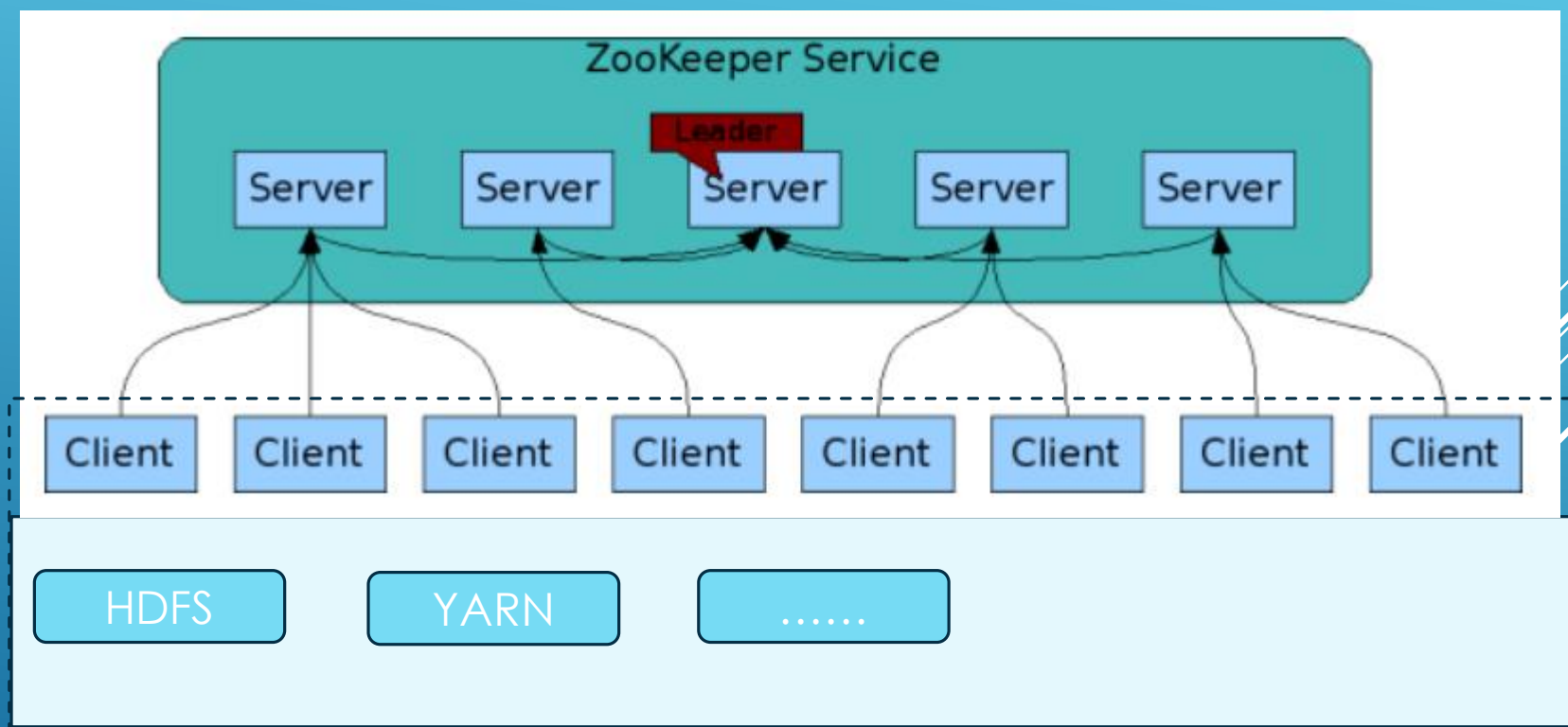
- ◆ 配置管理——提供配置文件的集中访问
- ◆ 命名服务——类似DNS
- ◆ 分布式锁——确保分布式环境下资源共享安全问题
- ◆ 集群管理——Master的产生

## 二、架构

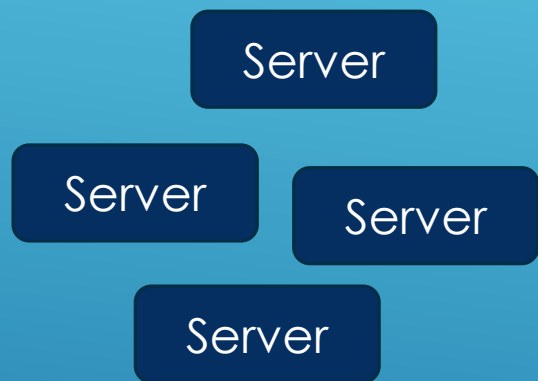
与大多数分布式应用一样，Server集群 与Client 方式



## 与其它应用的集成方式



### 三、集群角色



ZK集群

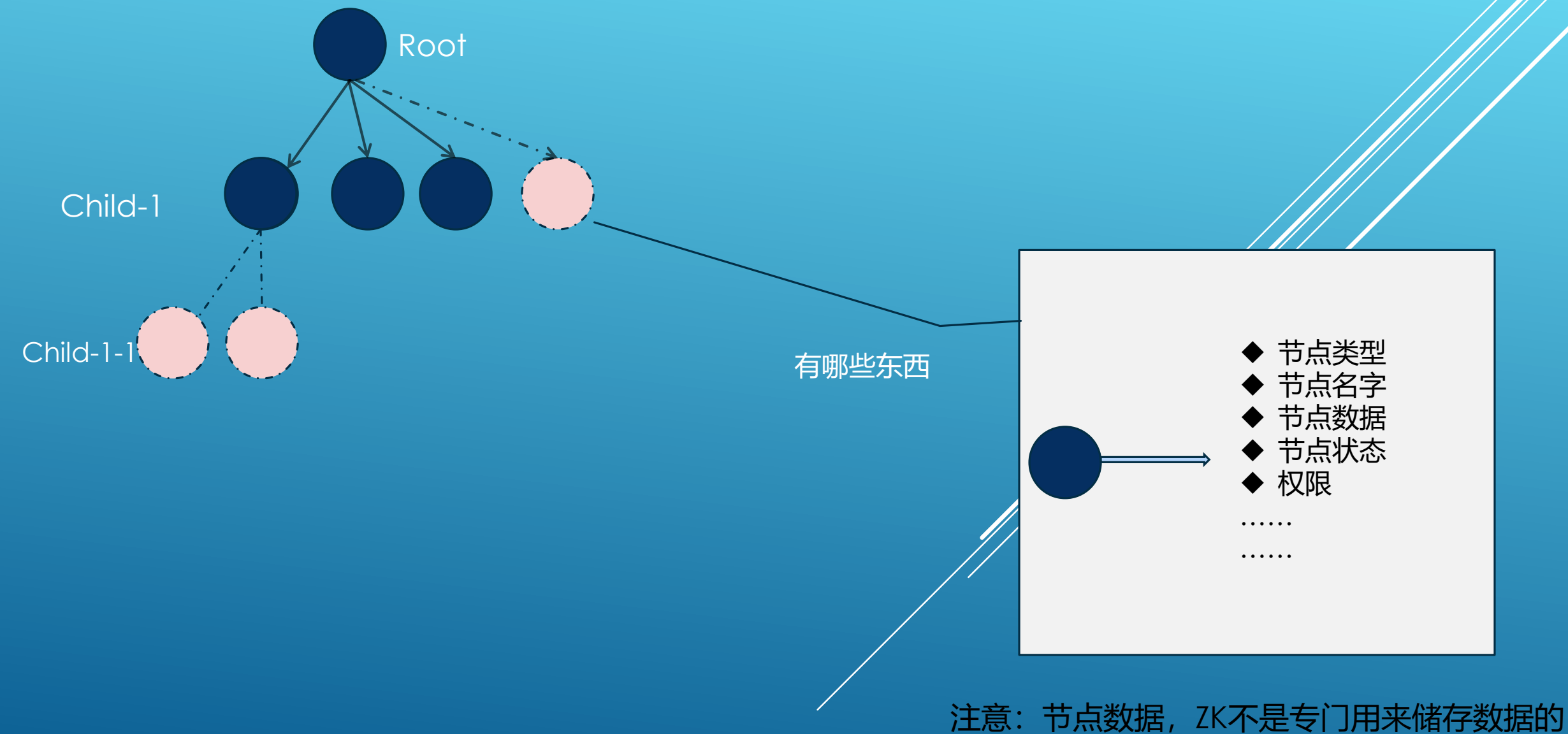


Leader -- 读写保障

Follower – 读、有机会上位成Leader

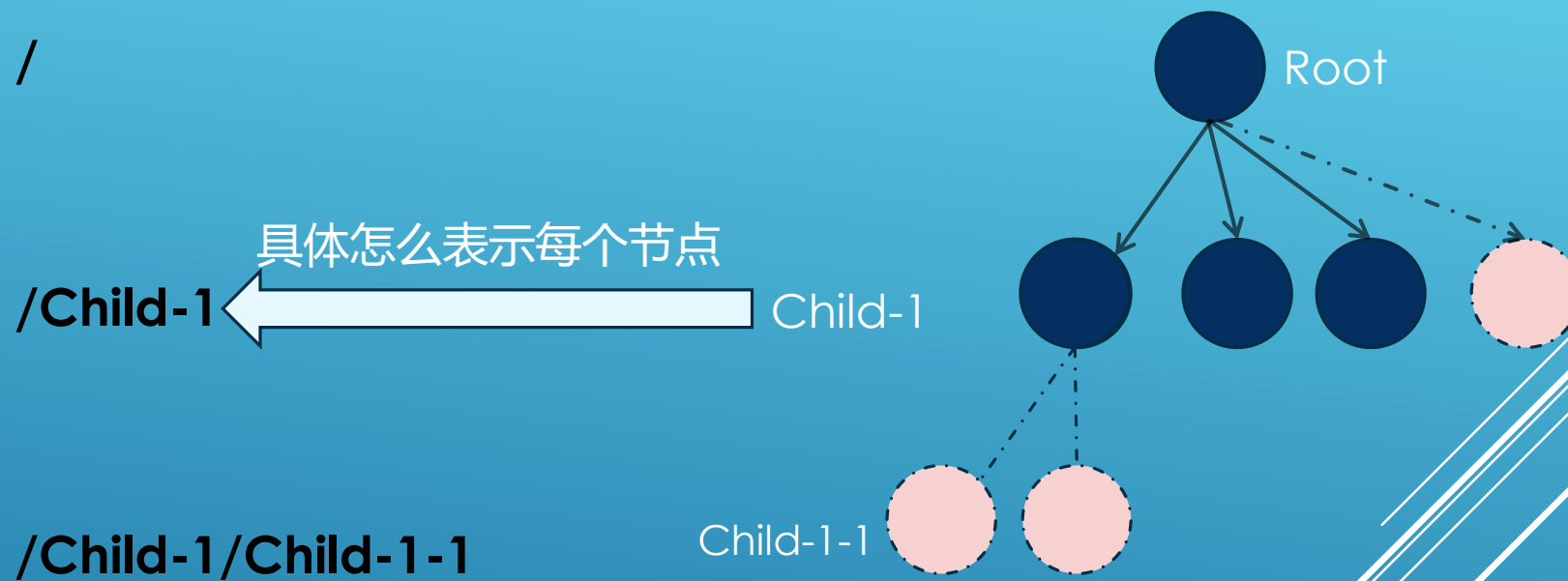
Observer – 读，就是读

## 四、Zookeeper的数据模型





## 四、Zookeeper的数据模型



## 五、一些概念

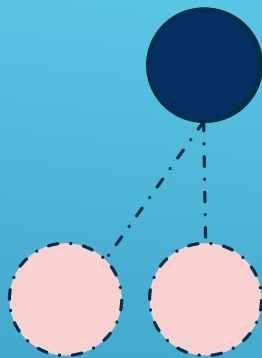
ZK中的节点叫 Znode

节点类型：

按生命周期，分为永久和临时

按是否具备顺序编号，分为有和无

所以可以组合 种节点



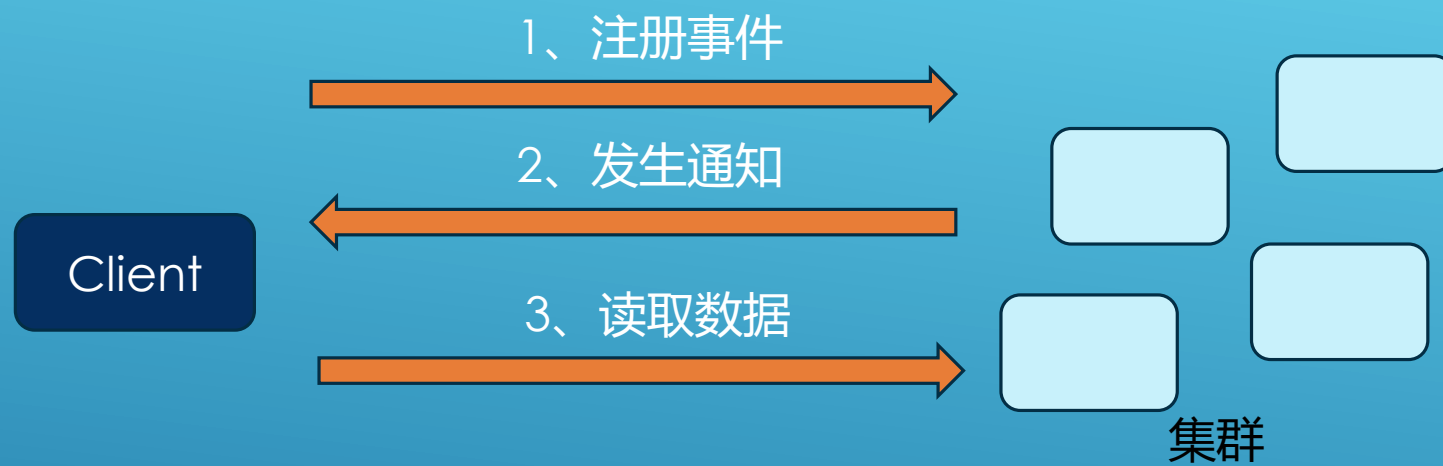
会话

Client与Server集群的一次连接过程，有超时控制

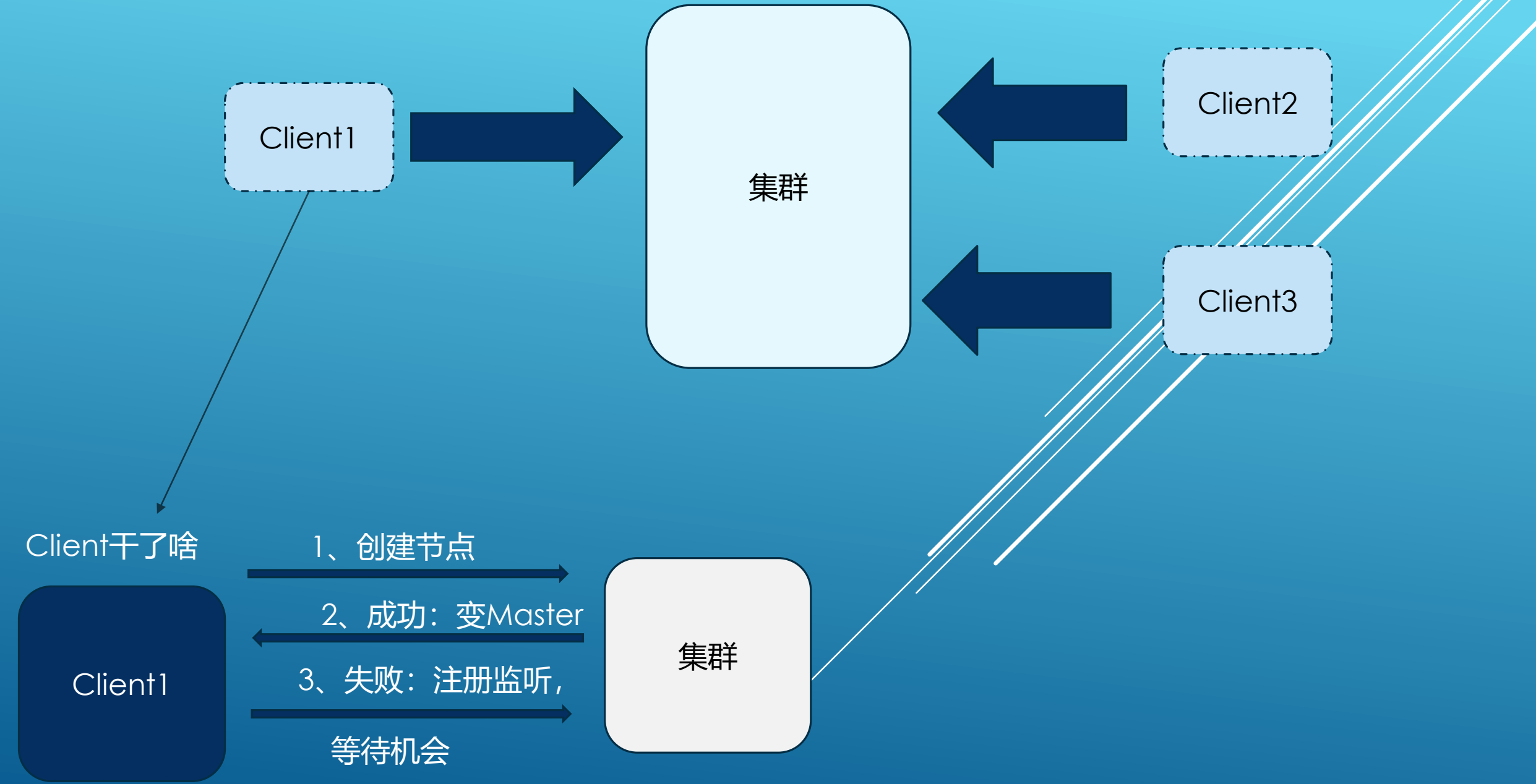
Watch机制

Client可以监控节点发生的事件，例如节点结构改变，节点数据改变

## Watch机制的流程



经典场景-集群管理 (Master的产生)



# Zookeeper命令行操作

## 创建节点

```
[zk: localhost:2181(CONNECTED) 34] create /config cm  
Created /config
```

```
[zk: localhost:2181(CONNECTED) 35] ls /  
[zookeeper, config]  
[zk: localhost:2181(CONNECTED) 36] get /config  
cm  
cZxid = 0x29  
ctime = Fri Nov 22 12:03:31 CST 2019  
mZxid = 0x29  
mtime = Fri Nov 22 12:03:31 CST 2019  
pZxid = 0x29  
cversion = 0  
dataVersion = 0  
aclVersion = 0  
ephemeralOwner = 0x0  
dataLength = 2  
numChildren = 0
```

查看节点列表

获取节点数据

## Zookeeper命令行操作

### 节点删除、创建临时节点

```
[zk: localhost:2181(CONNECTED) 37] delete /config  
[zk: localhost:2181(CONNECTED) 38] ls /  
[zookeeper]
```

```
[zk: localhost:2181(CONNECTED) 4] create -e /tmpNode data  
Created /tmpNode  
[zk: localhost:2181(CONNECTED) 5] ls /  
[tmpNode, zookeeper]
```

# Zookeeper命令行操作

## 创建顺序节点

```
[zk: localhost:2181(CONNECTED) 3] create -s /seqNode data
Created /seqNode0000000003
[zk: localhost:2181(CONNECTED) 4] ls /
[zookeeper, seqNode0000000003]
[zk: localhost:2181(CONNECTED) 5] create -s /seqNode data
Created /seqNode0000000004
[zk: localhost:2181(CONNECTED) 6] ls /
[zookeeper, seqNode0000000003, seqNode0000000004]
```

与创建临时节点的区别在哪？

## Watch机制演示

### 创建节点并设置节点数据

```
[zk: localhost:2181(CONNECTED) 15] create /config cluster
Created /config
[zk: localhost:2181(CONNECTED) 16] create /config/service-name 163.com
Created /config/service-name
[zk: localhost:2181(CONNECTED) 17] get /config/service-name
163.com
cZxid = 0x38
ctime = Fri Nov 22 13:47:15 CST 2019
mZxid = 0x38
mtime = Fri Nov 22 13:47:15 CST 2019
pZxid = 0x38
cversion = 0
dataVersion = 0
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 7
numChildren = 0
```



## Watch机制演示

### 获取节点数据的同时注册待监听的事件

```
[zk: localhost:2181(CONNECTED) 18] get /config/service-name changeEvent
163.com
cZxid = 0x38
ctime = Fri Nov 22 13:47:15 CST 2019
mZxid = 0x38
mtime = Fri Nov 22 13:47:15 CST 2019
pZxid = 0x38
cversion = 0
dataVersion = 0
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 7
numChildren = 0
[zk: localhost:2181(CONNECTED) 19]
WATCHER::

WatchedEvent state:SyncConnected type:NodeDataChanged path:/config/service-name
```

这里还差哪个步骤？

集群节点宕机场景演示 --真实情况与这个有一定差异，这里重点是为演示ZK的分布式协调功能

刚开始，namenodes节点的子节点为空

```
[zk: localhost:2181(CONNECTED) 40] ls /cluster/namenodes  
[]  
[zk: localhost:2181(CONNECTED) 41]
```

另起一个客户端C1，创建临时节点 n1

```
[zk: localhost:2181(CONNECTED) 1] create -e /cluster/namenodes/n1 n1  
Created /cluster/namenodes/n1
```

集群节点宕机场景演示 --真实情况与这个有一定差异，这里重点是为演示ZK的分布式协调功能

另起一个客户端C2，创建临时节点 n2

```
WatchedEvent state:SyncConnected type:None path:null
```

```
[zk: localhost:2181(CONNECTED) 0]
```

```
[zk: localhost:2181(CONNECTED) 0] create -e /cluster/namenodes/n2 n2
```

```
Created /cluster/namenodes/n2
```

集群节点宕机场景演示 --真实情况与这个有一定差异，这里重点是为演示ZK的分布式协调功能

另起一个客户端C3，注册节点列表监听事件

```
[zk: localhost:2181(CONNECTED) 45] ls /cluster/namenodes nm-event  
[n1, n2]
```

集群节点宕机场景演示 --真实情况与这个有一定差异，这里重点是为演示ZK的分布式协调功能

将客户端C2强行中断，观察C3的输出

```
[zk: localhost:2181(CONNECTED) 47] ls /cluster/namenodes nm-event
[n1, n2]
[zk: localhost:2181(CONNECTED) 48]
WATCHER::

WatchedEvent state:SyncConnected type:NodeChildrenChanged path:/cluster/namenodes

[zk: localhost:2181(CONNECTED) 48] ls /cluster/namenodes nm-event
[n1]
```

