# 概述

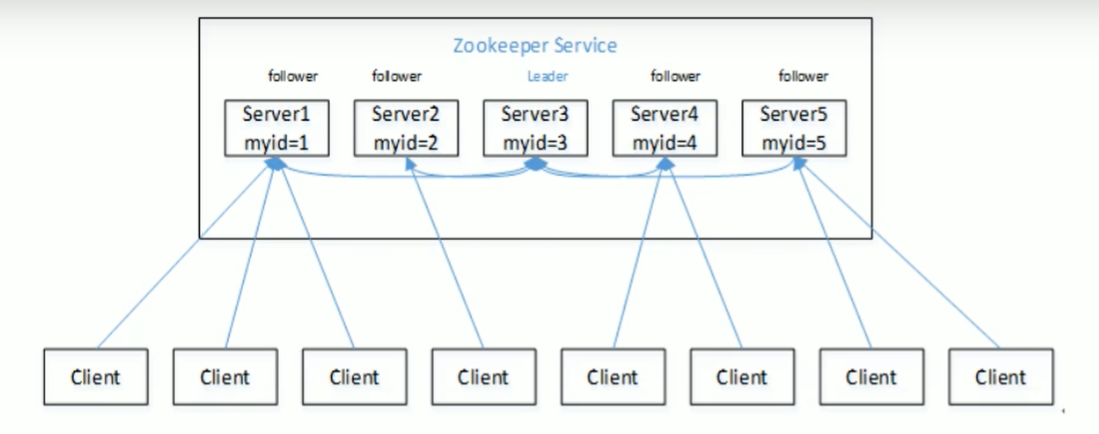
## 概述

Zookeeper=文件系统+监听通知机制。

客户端注册监听它关心的目录节点，当目录节点发生变化（数据改变、被删除、子目录节点增加删除）时，zookeeper会通知客户端。

Zookeeper从设计模式角度来理解：是一个基于观察者模式设计的分布式服务管理框架，它负责存储和管理大家都关心的数据，然后观察者的注册，一旦这些数据的状态发生变化，zookeeper就将负责通知已经在zookeeper上注册的那些观察者做出相应的反应。

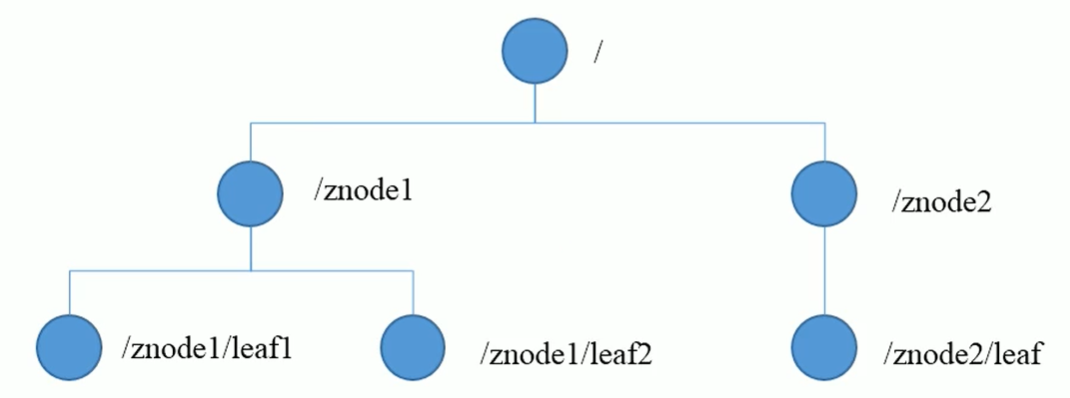
## 特点



1. zookeeper：一个领导者（leader），多个跟随者（follower）组成的集群；
2. 集群只要有半数以上节点存活，zookeeper集群就能正常服务；
3. 全局数据一致性：每个server保存一份相同的数据副本，client无论连接到哪个server，数据都是一致的；
4. 更新请求顺序进行，来自同一个client的更新请求按其发送顺序依次执行；
5. 数据更新原子性，一次数据更新要么成功，要么失败；
6. 实时性，在一定时间范围内，client能读到最新数据。

## 数据结构

Zookeeper数据模型的结构与Unix文件系统很相似，整体上可以看作是一棵树，每个节点称作一个ZNode。每一个ZNode默认能够存储1MB的数据，每个ZNode都可以通过其路径唯一标识。

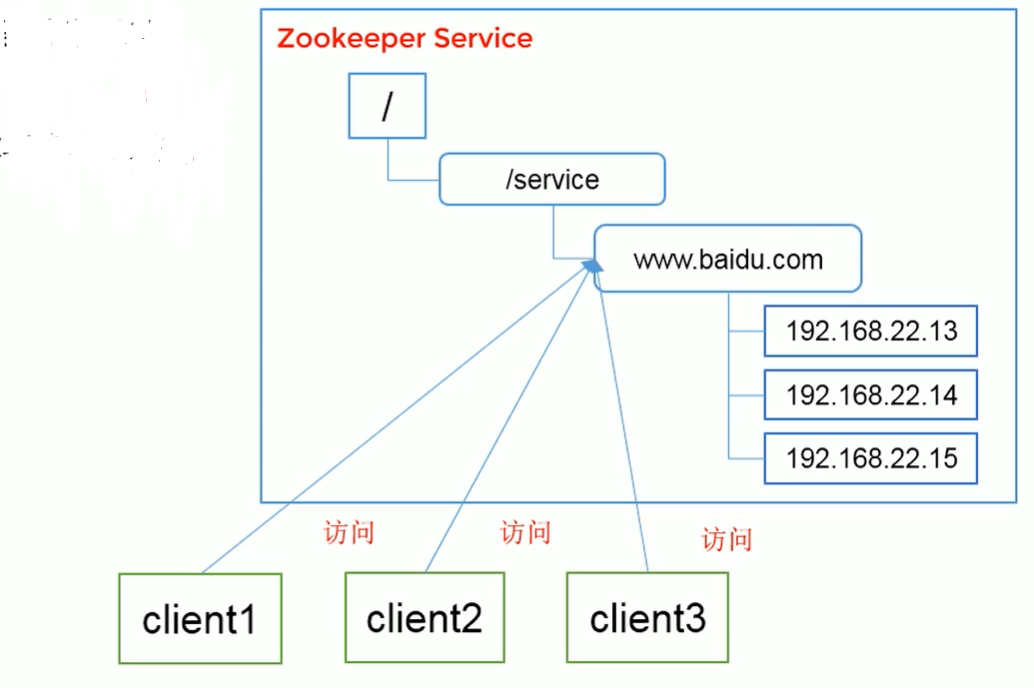


## 应用场景

提供服务包括：统一命名服务、统一配置管理、统一集群管理、服务器节点动态上下线、软负载均衡等。

### 统一命名服务

在分布式环境下，经常需要对应用/服务进行统一命名，便于识别。例如：IP不容易记住，而域名容易记住。



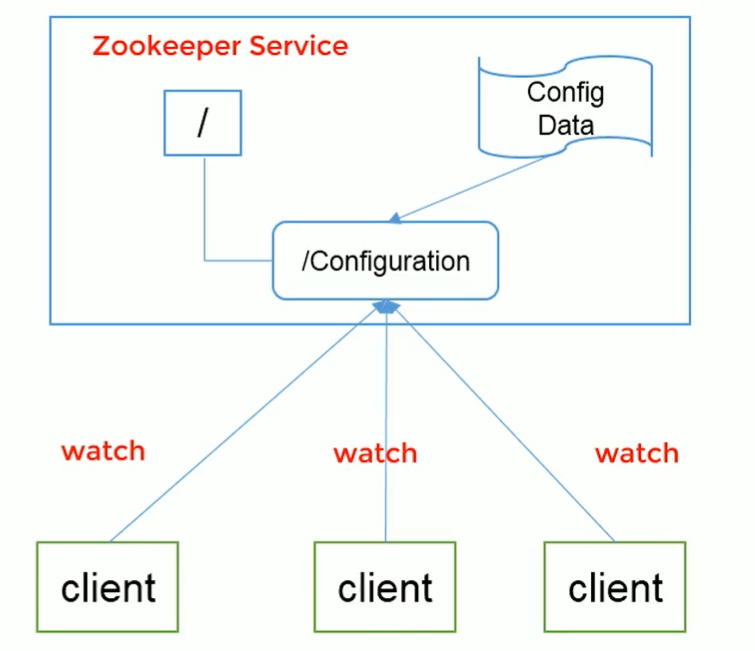
### 统一配置管理

在分布式环境下，配置文件同步非常常见：

1. 一般要求一个集群中，所有节点的配置信息是一致的，比如kafka集群；
2. 对配置文件修改后，希望能够快速同步到各个节点上；

配置管理可交由zookeeper实现：

1. 可将配置信息写入zookeeper上的一个ZNode；
2. 各个客户端服务器监听这个ZNode；
3. 一旦ZNode中的数据被修改，zookeeper将通知各个客户端服务器。

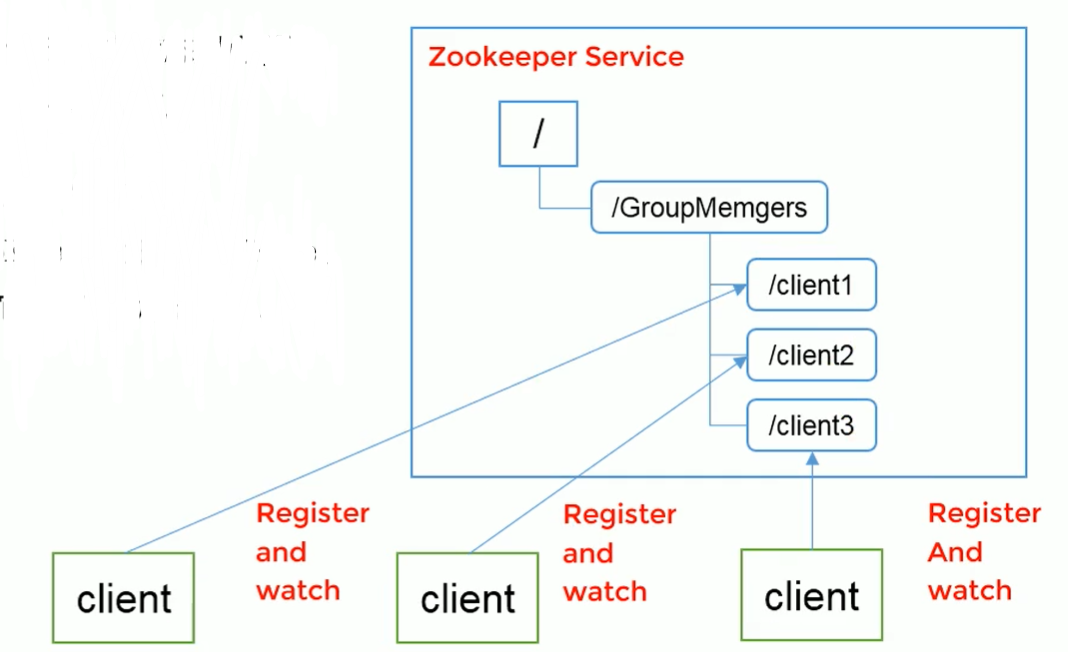


### 统一集群管理

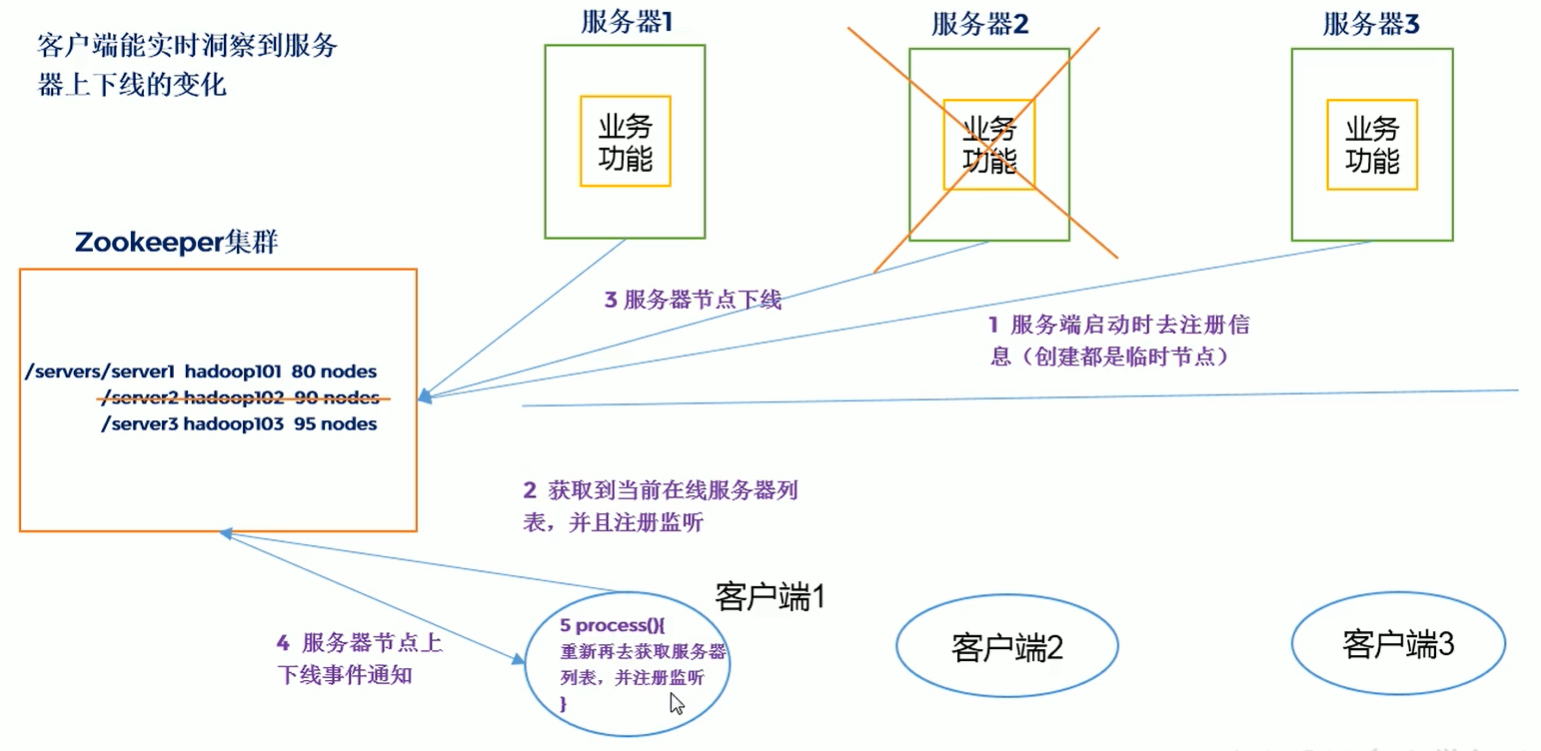
在分布式环境中，实时掌握每个节点的状态是必要的，可以根据节点实时状态做出一些调整。

Zookeeper可以实现实时监控节点状态变化：

1. 可将节点信息写入zookeeper上的一个ZNode；
2. 监听这个ZNode可获取它的实时状态变化。

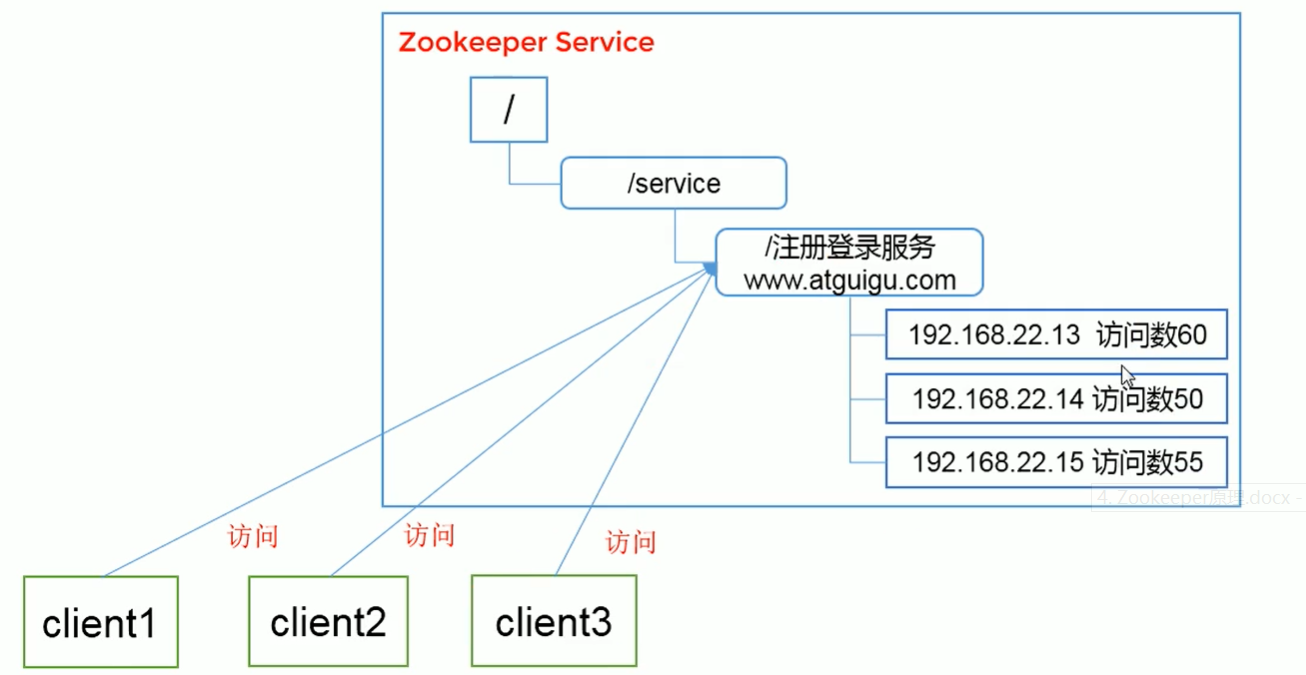


### 服务器节点动态上下线



### 软负载均衡

在zookeeper中记录每台服务器的访问数，让访问数最少的服务器去处理最新的客户端请求。



# 安装配置

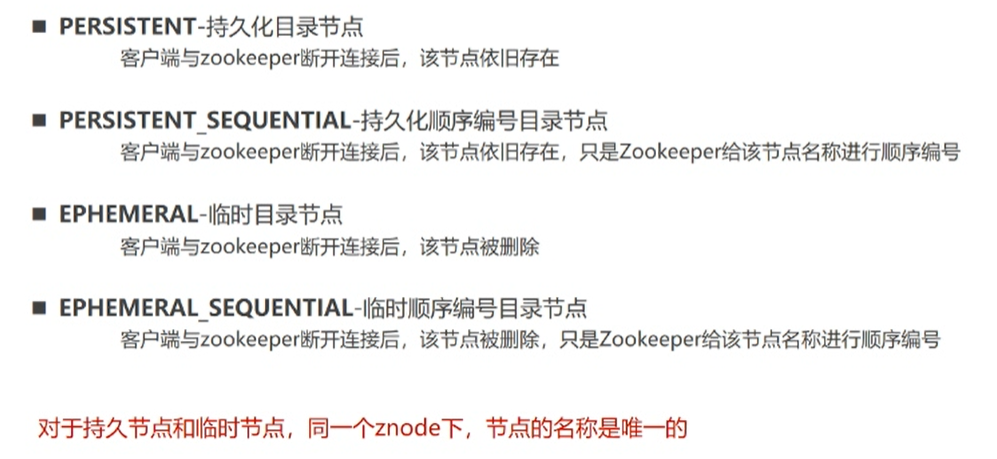
## 本地模式安装部署

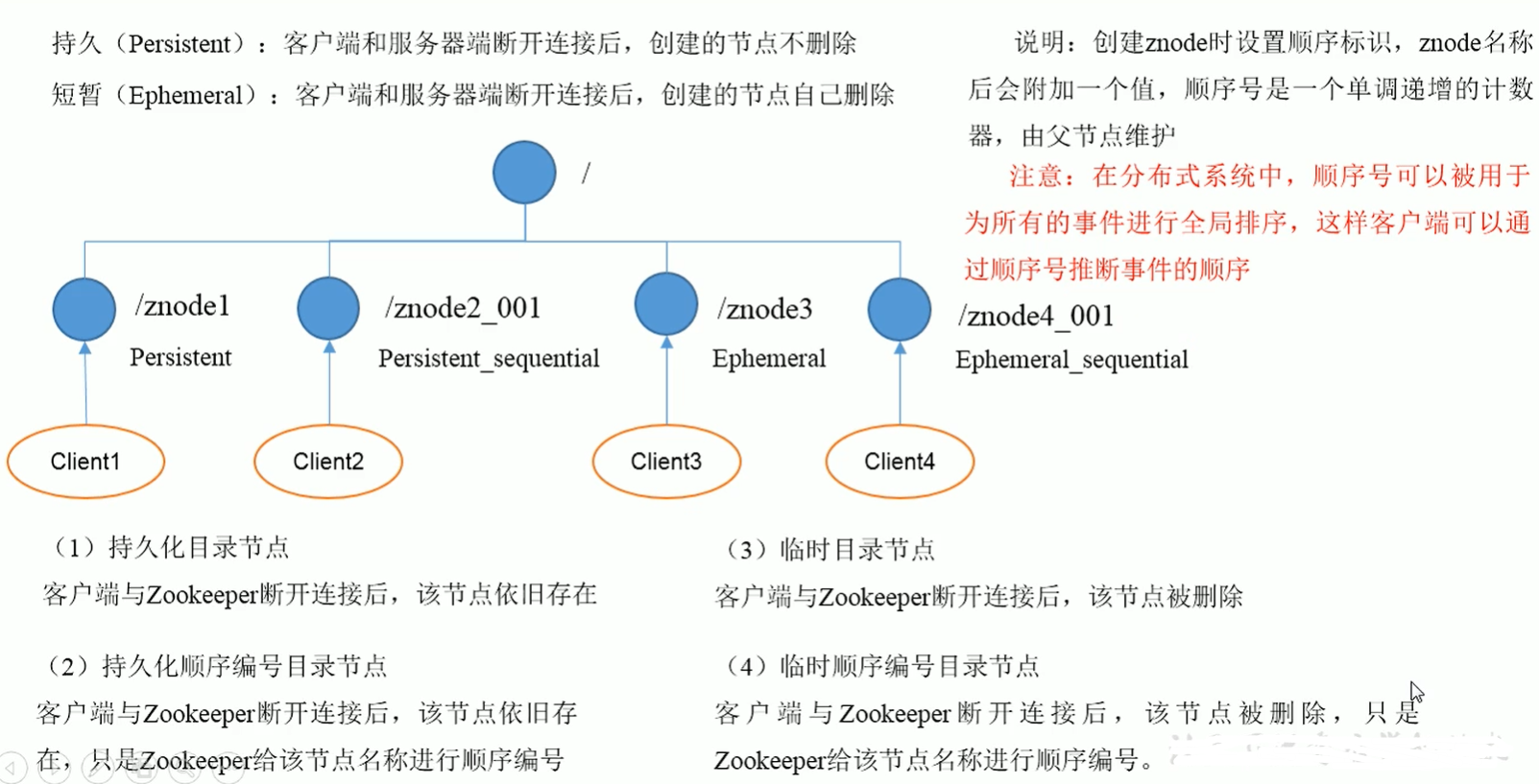
## 配置参数

# 原理

## 选举机制

## 节点类型





## Stat结构体

## 监听器原理

## 写数据流程

# 部署方式

# ZAB协议

# 常用指令

## 创建一个节点

## 获取子节点并监听数据变化

## 判断ZNode是否存在