# ZOOKEEPER

Coordinating your cluster

# What is ZooKeeper?


- It basically keeps track of information that must be synchronized across your cluster
  - *Which node is the master?*
  - *What tasks are assigned to which workers?*
  - *Which workers are currently available?*
- It's a tool that applications can use to recover from partial failures in your cluster.
- An integral part of HBase, High-Availability (HA) MapReduce, Drill, Storm, Solr, and much more

# Failure modes

- Master crashes, needs to fail over to a backup
- Worker crashes – its work needs to be redistributed
- Network trouble – part of your cluster can’t see the rest of it

# “Primitive” operations in a distributed system

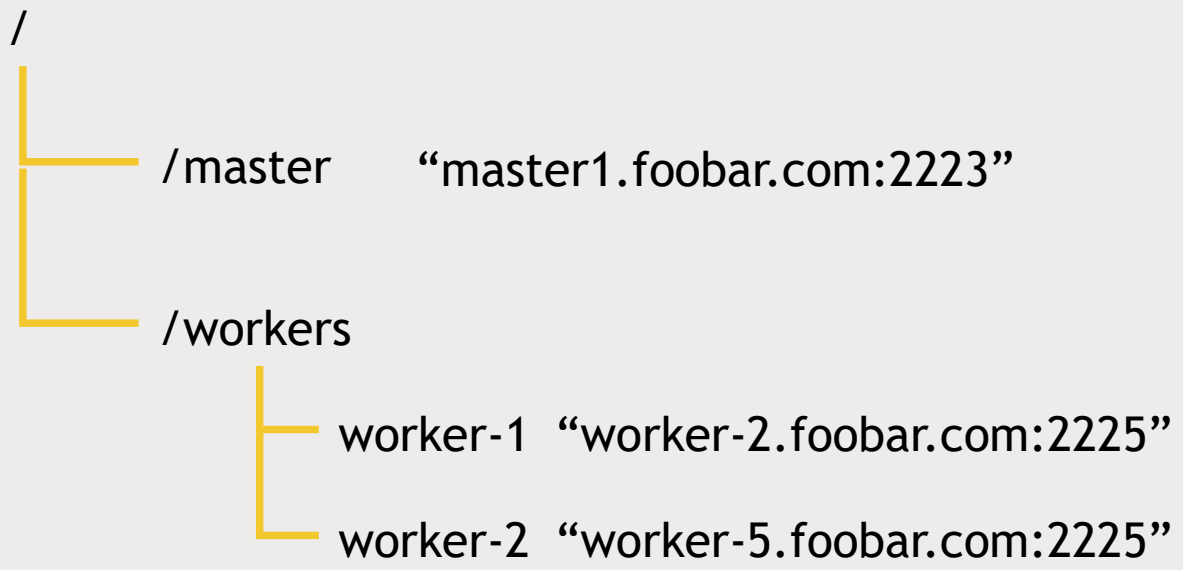
- Master election
  - *One node registers itself as a master, and holds a “lock” on that data*
  - *Other nodes cannot become master until that lock is released*
  - *Only one node allowed to hold the lock at a time*
- Crash detection
  - *“Ephemeral” data on a node’s availability automatically goes away if the node disconnects, or fails to refresh itself after some time-out period.*
- Group management
- Metadata
  - *List of outstanding tasks, task assignments*

# But ZooKeeper's API is not about these primitives.

- Instead they have built a more general purpose system that makes it easy for applications to implement them.

# Zookeeper's API

- Really a little distributed file system
  - *With strong consistency guarantees*
  - *Replace the concept of "file" with "znode" and you've pretty much got it*
- Here's the ZooKeeper API:
  - *Create, delete, exists, setData, getData, getChildren*



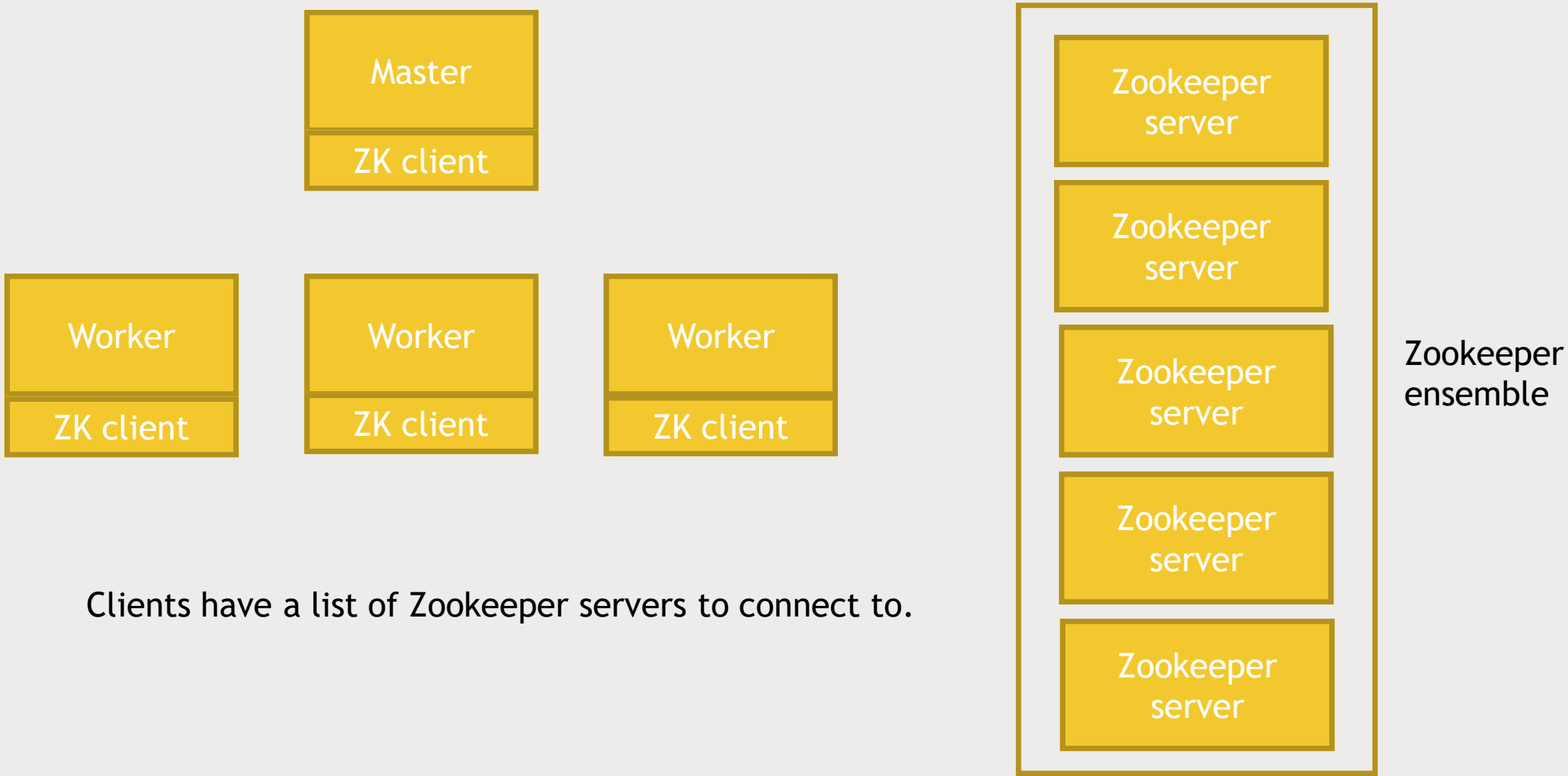
# Notifications

- A client can register for notifications on a znode
  - *Avoids continuous polling*
  - *Example: register for notification on /master – if it goes away, try to take over as the new master.*

# Persistent and ephemeral znodes

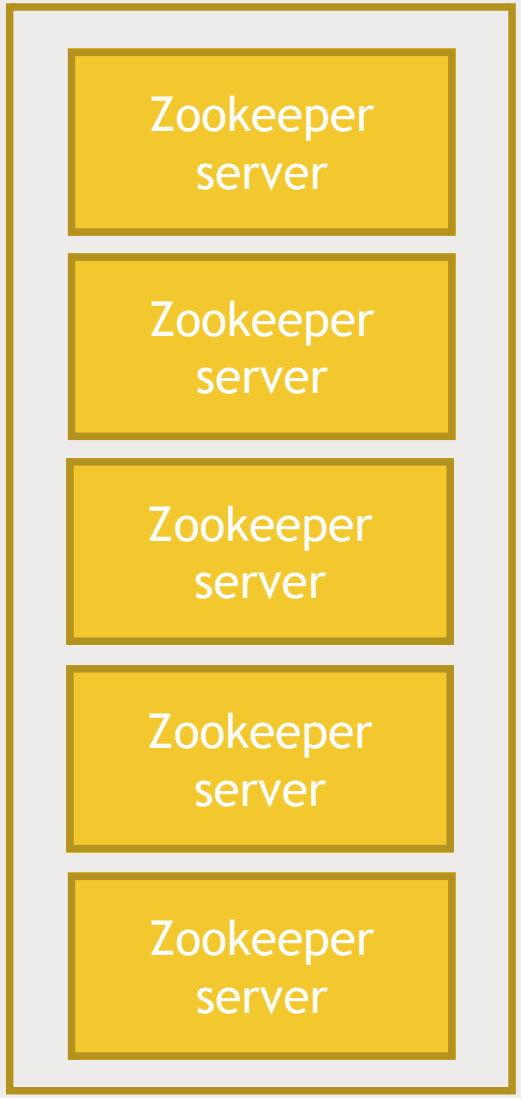
- Persistent znodes remain stored until explicitly deleted
  - *i.e., assignment of tasks to workers must persist even if master crashes*
- Ephemeral znodes go away if the client that created it crashes or loses its connection to ZooKeeper
  - *i.e., if the master crashes, it should release its lock on the znode that indicates which node is the master!*

# ZooKeeper Architecture



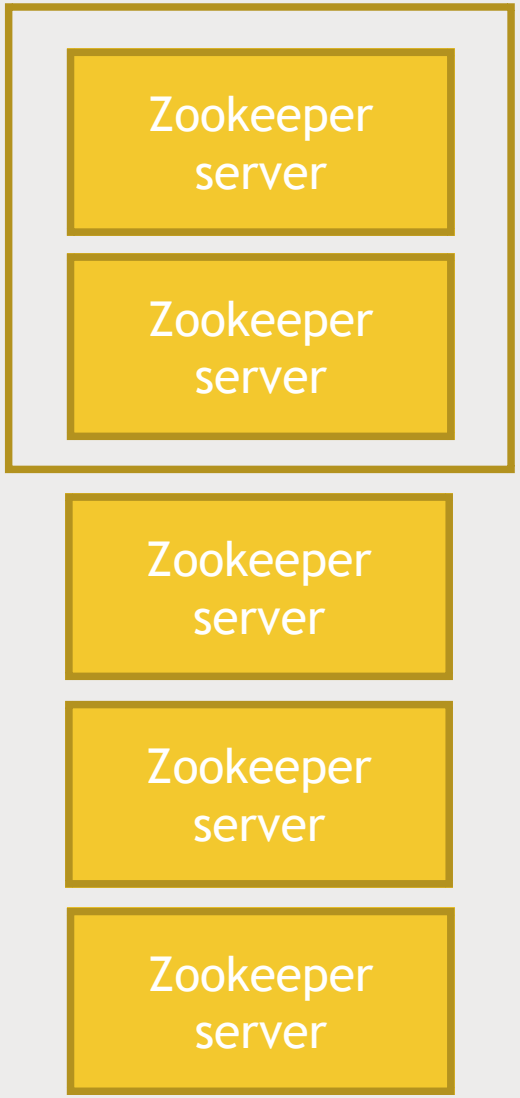

Clients have a list of Zookeeper servers to connect to.

# ZooKeeper quorums




Zookeeper ensemble

# ZooKeeper quorums




Zookeeper ensemble

# Sounds a lot like how MongoDB works

# Let’s play with the ZooKeeper.