# Apache ZooKeeper

**Apache ZooKeeper** is an open-source server for highly reliable distributed coordination of cloud applications.[2] It is a project of the Apache Software Foundation.

ZooKeeper is essentially a service for distributed systems offering a hierarchical key-value store, which is used to provide a distributed configuration service, synchronization service, and naming registry for large distributed systems (see *Use cases*).[3] ZooKeeper was a sub-project of Hadoop but is now a top-level Apache project in its own right.

**Apache ZooKeeper**


| | |
|---|---|
| **Developer(s)** | Apache Software Foundation |
| **Stable release** | 3.6.3 / April 13, 2021[1] |
| **Repository** | ZooKeeper Repository (https://gitbox.apache.org/repos/asf?p=zookeeper.git) |
| **Written in** | Java |
| **Operating system** | Cross-platform |
| **Type** | Distributed computing |
| **License** | Apache License 2.0 |
| **Website** | zookeeper.apache.org (https://zookeeper.apache.org) |

## Contents

**Overview**
**Architecture**
**Use cases**
**Client libraries**
**Apache projects using ZooKeeper**
**See also**
**References**
**External links**

## Overview

ZooKeeper's architecture supports high availability through redundant services. The clients can thus ask another ZooKeeper leader if the first fails to answer. ZooKeeper nodes store their data in a hierarchical name space, much like a file system or a tree data structure. Clients can read from and write to the nodes and in this way have a shared configuration service. ZooKeeper can be viewed as an atomic broadcast system, through which updates are totally ordered. The ZooKeeper Atomic Broadcast (ZAB) protocol is the core of the system.[4]

ZooKeeper is used by companies including Yelp, Rackspace, Yahoo!,[5] Odnoklassniki, Reddit,[6] NetApp SolidFire,[7] Facebook,[8] Twitter[9] and eBay as well as open source enterprise search systems like Solr.[10]

ZooKeeper is modeled after Google's Chubby lock service[11][12] and was originally developed at Yahoo! for streamlining the processes running on big-data clusters by storing the status in local log files on the ZooKeeper servers. These servers communicate with the client machines to provide them the information. ZooKeeper was developed in order to fix the bugs that occurred while deploying distributed big-data applications.

Some of the prime features of Apache ZooKeeper are:

- Reliable System: This system is very reliable as it keeps working even if a node fails.
- Simple Architecture: The architecture of ZooKeeper is quite simple as there is a shared hierarchical namespace which helps coordinating the processes.
- Fast Processing: ZooKeeper is especially fast in "read-dominant" workloads (i.e. workloads in which reads are much more common than writes).
- Scalable: The performance of ZooKeeper can be improved by adding nodes.

# Architecture

Some common terminologies regarding the ZooKeeper architecture:

- Node: The systems installed on the cluster
- ZNode: The nodes where the status is updated by other nodes in cluster
- Client applications: The tools that interact with the distributed applications
- Server applications: Allows the client applications to interact using a common interface

The services in the cluster are replicated and stored on a set of servers (called an "ensemble"), each of which maintains an in-memory database containing the entire data tree of state as well as a transaction log and snapshots stored persistently. Multiple client applications can connect to a server, and each client maintains a TCP connection through which it sends requests and heartbeats and receives responses and watch events for monitoring.[13]

# Use cases

Typical use cases for ZooKeeper are:

- Naming service
- Configuration management
- Data Synchronization
- Leader election
- Message queue
- Notification system

# Client libraries

In addition to the client libraries included with the ZooKeeper distribution, a number of third-party libraries such as Apache Curator and Kazoo are available that make using ZooKeeper easier, add additional functionality, additional programming languages, etc.

## Apache projects using ZooKeeper

- Apache Hadoop
- Apache Accumulo
- Apache HBase
- Apache Hive
- Apache Kafka
- Apache Solr
- Apache Spark
- Apache NiFi
- Apache Druid
- Apache Helix

## See also

- Hadoop

## References


1. "Apache ZooKeeper - Releases" (https://zookeeper.apache.org/releases.html). Retrieved 29 July 2021.
2. "Apache Zookeeper4" (https://zookeeper.apache.org/). Retrieved 31 January 2021.
3. "Index - Apache ZooKeeper - Apache Software Foundation" (https://cwiki.apache.org/confluence/display/ZOOKEEPER/). *cwiki.apache.org*. Retrieved 2016-08-26.
4. "Zookeeper Overview" (https://cwiki.apache.org/confluence/display/ZOOKEEPER/ProjectDescription).
5. "ZooKeeper/Powered By" (https://web.archive.org/web/20131209063307/http://wiki.apache.org/hadoop/ZooKeeper/PoweredBy). Archived from the original (http://wiki.apache.org/hadoop/ZooKeeper/PoweredBy) on 2013-12-09. Retrieved 2012-01-25.
6. "Why Reddit was down on Aug 11" (https://www.reddit.com/r/announcements/comments/4y0m56/why_reddit_was_down_on_aug_11/).
7. "5 Big DaaS Challenges and How to Overcome Them | NetApp Newsroom" (https://newsroom.netapp.com/blogs/5-big-daas-challenges-and-how-to-overcome-them/). *NetApp Newsroom*. 2016-06-20. Retrieved 2017-05-24.
8. "Location-Aware Distribution: Configuring servers at scale" (https://code.fb.com/data-infrastructure/location-aware-distribution-configuring-servers-at-scale/). *Facebook Code*. 2018-07-19. Retrieved 2018-07-20.
9. "ZooKeeper at Twitter" (https://blog.twitter.com/engineering/en_us/topics/infrastructure/2018/zookeeper-at-twitter.html). *Twitter Engineering Blog*. 2018-10-11. Retrieved 2018-12-08.
10. "SolrCloud" (https://cwiki.apache.org/confluence/display/solr/SolrCloud).
11. Burrows, Mike (2006). "The Chubby lock service for loosely-coupled distributed systems" (https

11. Burrows, Mike (2006). "The Chubby lock service for loosely-coupled distributed systems" (https://research.google/pubs/pub27897/). *7th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*.
12. Chandra, Tushar Deepak; Griesemer, Robert; Redstone, Joshua (2007). "Paxos Made Live - An Engineering Perspective (2006 Invited Talk)" (https://research.google/pubs/pub33002/). *Google Research*. Retrieved 2020-03-03.
13. "Zookeeper" (https://zookeeper.apache.org/doc/current/zookeeperOver.html).

# External links

- Official website (https://zookeeper.apache.org)