

ZOOKEEPER + SIGNALFX INTEGRATION

What is ZooKeeper?

Apache ZooKeeper is an open-source coordination service for distributed applications. It exposes a simple set of operations that applications can build on for higher-level, robust services such as service discovery, dynamic configuration management, synchronization, and distributed locking. ZooKeeper is used to serialize tasks across clusters so that synchronization does not have to be built separately into each service and project. Ultimately, ZooKeeper cuts down on management complexity and increases resiliency of applications as they are deployed.



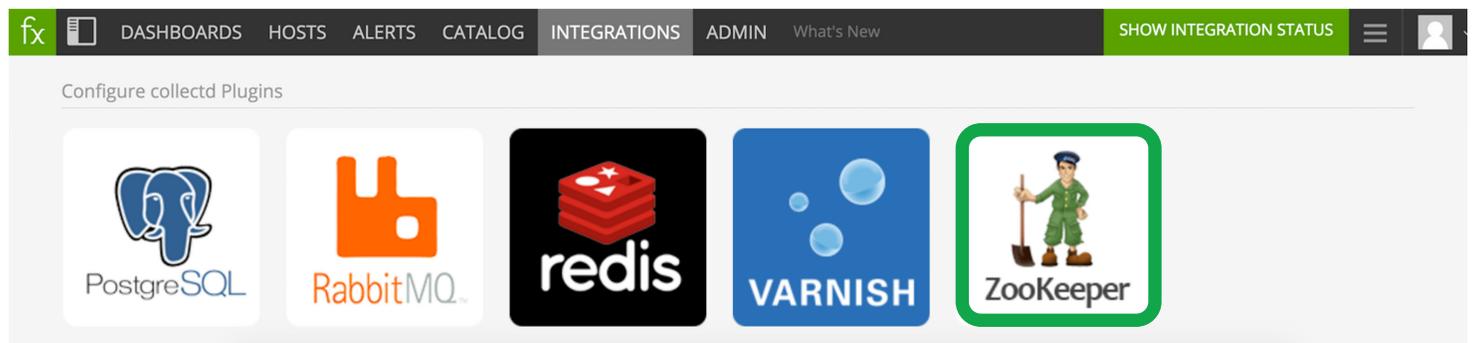
ZooKeeper allows distributed processes to coordinate with each other through a shared hierarchal namespace that is organized similarly to a directory tree structure of standard file systems. The name space consists of data registers, called znodes, that behave like files and directories. Unlike a typical file system, ZooKeeper provides low latency, high availability, and strict ordered access to the znodes. The performance aspects of ZooKeeper allow it to be used in large distributed systems, and its strict ordering allows sophisticated synchronization operations to be implemented.

Sending ZooKeeper Metrics to SignalFx

Use collectd and the collectd-zookeeper plugin to capture ZooKeeper metrics and track data by node. Key ZooKeeper metrics include node count, packet count, latency, watch count, data size, and open file descriptors. SignalFx provides built-in ZooKeeper monitoring dashboards displaying useful production metrics at the node, host, and cluster levels.

Monitoring ZooKeeper

From SignalFx's experience monitoring ZooKeeper in production, there are four primary indicators to manage a healthy ZooKeeper service: disk usage, request metrics, active connections, and total znode count. In most cases, changes in these indicators occur at the node level, as most ZooKeeper clusters tend to be small. Creating alerts on these leading indicators will result in meaningful notifications as patterns emerge at the service level, helping the operations team or the service owner in the development organization address the trend before it leads to a performance issue in production.



DISK USAGE ON ZOOKEEPER INSTANCES: ZooKeeper contains files that are persistent copies of the znodes stored as snapshots and transactional log files. As changes are made to the znodes, these changes are appended to the transactional log, and, eventually, a snapshot of the current state of all znodes is written to the file system. Disk space can be properly managed by occasionally cleaning older znode snapshot files, therefore allowing ZooKeeper to have hundreds of days of uptime. However, ZooKeeper becomes non-operational when disk capacity on a node runs out due to the high volume of snapshot data and transactional log data, and losing ZooKeeper is critical to the overall operations of your environment.

Disk usage should be consistent and grow in parallel across all znodes of a cluster. However, unexpected increases in disk usage over a short amount of time for one ZooKeeper host indicates increased writes to disk memory. Because snapshots are only deleted after a certain time period, the sudden increase in volume of snapshots written to disk impacts the remaining availability of disk space on the host. An alert for one host often indicates that other hosts are nearing a similar issue, and running out of disk capacity for a ZooKeeper cluster is an early indication of service failure.

CAPACITY THROUGH REQUEST METRICS:

ZooKeeper is intended to be used as a control panel, not as a heavy database with high throughput. One leading indicator of capacity is the number of outstanding requests. An increasing number of outstanding requests indicates a lack of capacity to serve these client requests, or that a client service is behaving incorrectly and overwhelming the ZooKeeper cluster with requests.

ZooKeeper Metrics

Packets Sent	Packets Received
Size of the Data Tree	Average Request Latency
Number of Ephemeral Nodes in the Data Tree	Maximum Number of File Descriptors
Maximum Request Latency	Minimum Request Latency
Number of Active Clients	Number of File Descriptors
Outstanding Requests	Number of Watches
Number of Znodes	

Request latency is another leading indicator of capacity. The longer it takes to process the request, the more likely there is limited capacity available. Monitor the percentile distribution of request latencies to understand outliers. If the average latency increases for one outlier, investigate whether there is one machine causing slower performance for the entire service. If all machines show increased latency, investigate whether this degraded performance is tied to a specific infrastructure metric—such as high CPU—or whether scaling the ZooKeeper cluster vertically or horizontally may help better handle increased traffic.

ACTIVE CLIENT CONNECTIONS: ZooKeeper maintains an active connection with all its clients and keeps a session for each active client that is connected to it. When a client is disconnected from ZooKeeper for more than a specified timeout, the session expires.

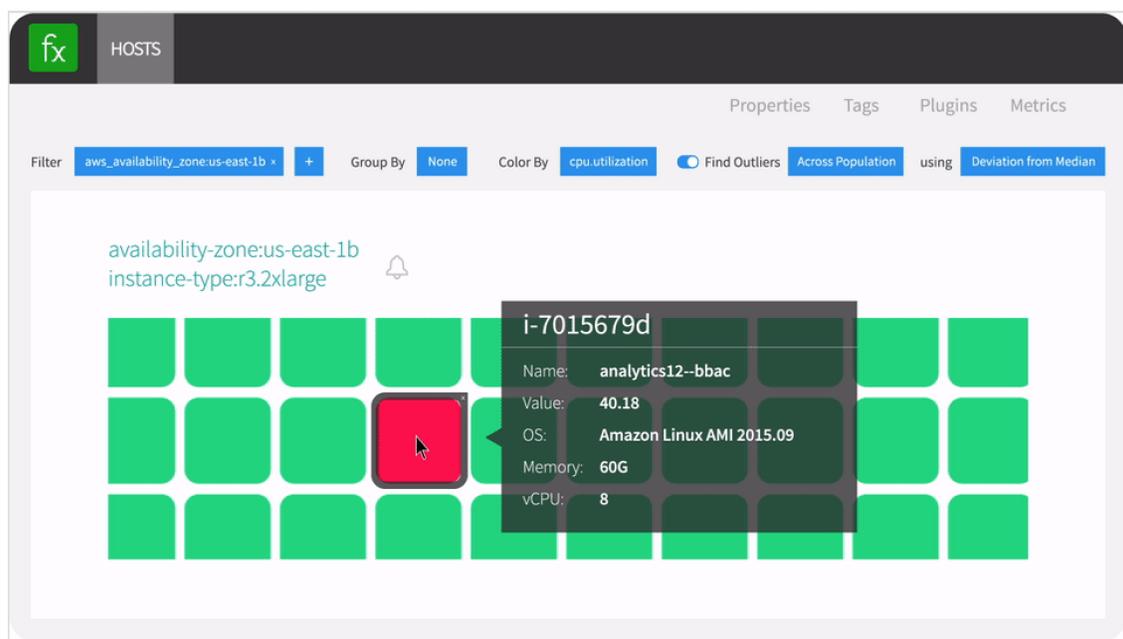
For alerting, use the number of active, connected sessions, and measure the growth rate of these sessions over a specified time period—such as one minute—as a leading indicator that there are too many open client connections to the ZooKeeper service. Too many client connections on a single znode can cause bursts of traffic and limited scalability, impacting available capacity and service performance. Sudden decreases in client connections is likely an indication of issues in the underlying infrastructure such as network or server components.

CLUSTER HEALTH ACROSS ALL NODES: For reliable service, ZooKeeper hosts are deployed in a cluster and, as long as a majority of hosts are up, the service will be available. The structure of ZooKeeper mandates that there is one master host and an expected number of n-1 follower hosts for the entire cluster. The total node count inside the ZooKeeper tree should remain consistent, unless a node died or a network partition occurred.

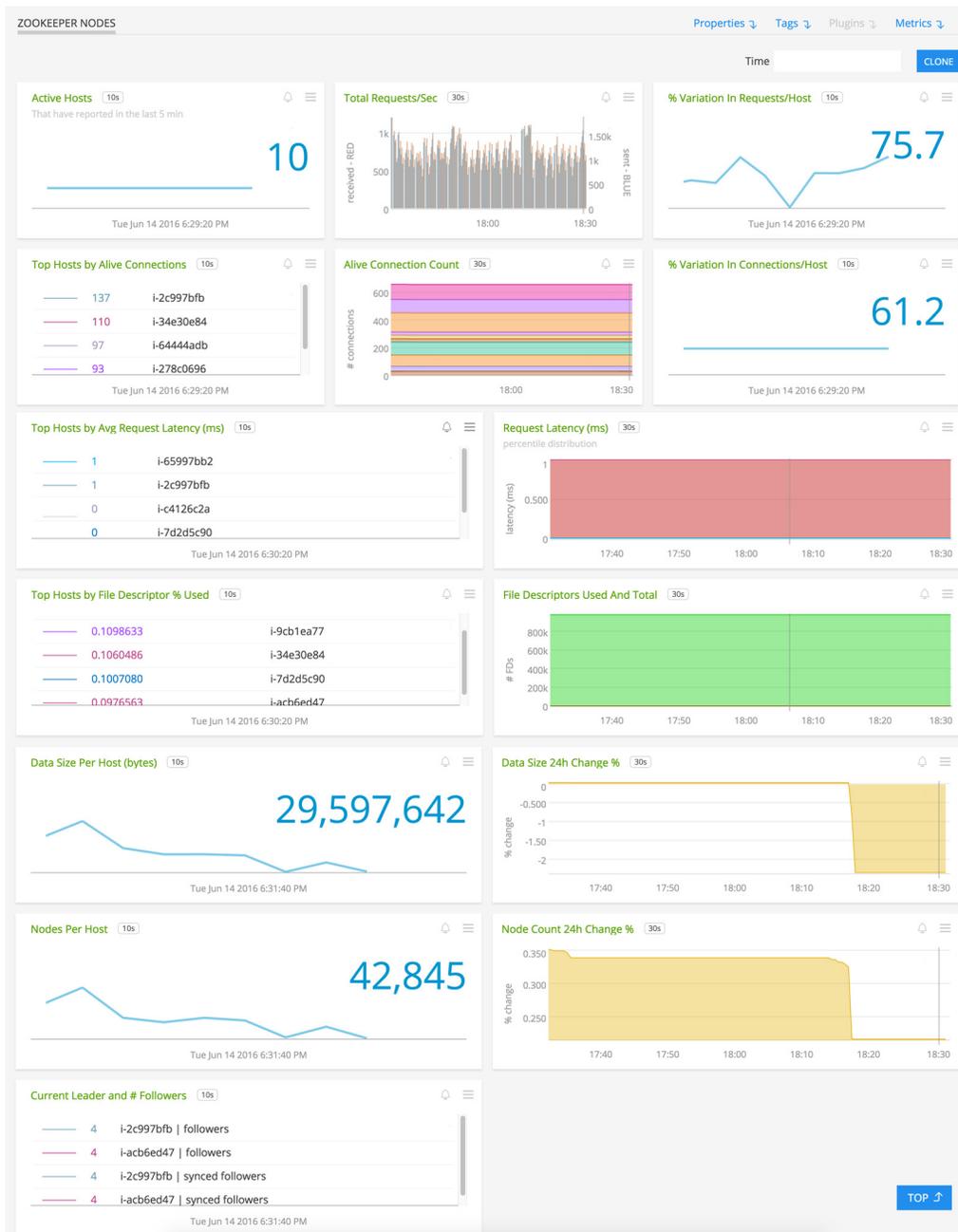
The SignalFx Difference

MEANINGFUL ALERTS: Raw metrics sometimes do not provide enough insight on their own to serve as the basis of meaningful, reliable alerts. Applying duration conditions to alert rules can help to determine whether an issue requires attention. ZooKeeper clusters tend to be small with low throughput, and, therefore, increasing latency is an indication of an emerging issue. SignalFx helps set duration and percent of duration conditions so that you know if a problem persists longer than the window required to self-adjust. You aren't alerted of and forced to troubleshoot a false-positive notification.

INSTANT OUTLIER DETECTION: When it comes to operating ZooKeeper, it is expected that the znodes across the cluster are all relatively balanced and consistent over time. Therefore, when a znode exhibits symptoms of outlier behavior, users need to quickly identify the issue and investigate the root cause. With SignalFx's Host Navigator view, anomalous servers appear red and all others appear green when automatic Outlier Detection is activated. The statistical algorithms that underlie Outlier Detection enable SignalFx users to quickly isolate the hosts within services or CPUs that deviate from the mean or median so you can immediately start to drill down into the cause of an issue without the typical, annoying trial-and-error.



JUMPSTART SUCCESS: There are many metrics specific to ZooKeeper, and knowing where to start and what to monitor can be tricky. SignalFx curates the ZooKeeper metrics that matter right out of the box alongside data from the other applications and cloud services in your infrastructure. SignalFx provides built-in dashboards and alert detector templates that give you a jumpstart on monitoring ZooKeeper in your environment.



About SignalFx

SignalFx is the most advanced monitoring and alerting solution for modern infrastructure. Our mission is to help cloud-ready organizations drive high levels of availability in today's elastic, agile, distributed environments. With SignalFx, development and operations teams gain a real-time view of, interact with, and take action on the infrastructure and application metrics that matter. We have enterprise customers including Yelp, Cisco, Zuora, and Hubspot and thousands of users analyzing billions of metrics every day. SignalFx was founded in 2013 by former Facebook and VMware executives, launched in 2015, and is backed by Andreessen Horowitz and Charles River Ventures.