Quick Ref: Zookeeper

|  |  |  |
| --- | --- | --- |
| **S.No** | **Topic** | **Desc** |
|  | **Web Ref** |  |
|  | Admin GUIDE | <https://zookeeper.apache.org/doc/trunk/zookeeperAdmin.html> |
|  | Configuration guide | <https://zookeeper.apache.org/doc/trunk/zookeeperAdmin.html#sc_configuration> |
|  | Important link | <https://www.youtube.com/watch?v=dwTlW_HrsVg> |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **General Info** |  |
| **1** | **Zookeeper** | ZooKeeper is a king of coordination. it is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. All of these kinds of services are used in some form or another by distributed applications |
| 2 | HA Enable requires Zookeeper as pre-request | # HA requires zookeeper because  ~~It only keeps track of which NameNode is Active at any given time~~  It both keeps track of which NameNode is Active at any given time, and manages the Edits file. Which is a log of changes to the HDFS filesystem. |
| 3 | Multi cluster installation | Usually three servers is more than enough for a production (zookeeper) install, but for maximum reliability during maintenance, you may wish to install five servers. With three servers, if you perform maintenance on one of them, you are vulnerable to a failure on one of the other two servers during that maintenance |
| 4 | Configuration file | #Use the following settings as a starting point  tickTime=2000  dataDir=/var/lib/zookeeper/  clientPort=2181  initLimit=5  syncLimit=2  server.1=zoo1:2888:3888  server.2=zoo2:2888:3888  server.3=zoo3:2888:3888 |
| 4.1 | Configuration file  (minimum configuration) | # **clientPort** - the port to listen for client connections; that is, the port that clients attempt to connect to.  # **dataDir** - the location where ZooKeeper will store the in-memory database snapshots and, unless specified otherwise, the transaction log of updates to the database. Note: Be careful where you put the transaction log. A dedicated transaction log device is key to consistent good performance. Putting the log on a busy device will adversely effect performance  **#tickTime** - the length of a single tick, which is the basic time unit used by ZooKeeper, as measured in milliseconds. It is used to regulate heartbeats, and timeouts. For example, the minimum session timeout will be two ticks. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **Zookeeper commands** |  |
|  | **Basic admin commands** | 4 character commands |
| 1 | >zookeeper-client  >zookeeper-server | # To connect to zookeeper |
| 2 | >help | # To get help |
| 3 | >ls –l | # To list details...  <https://zookeeper.apache.org/doc/trunk/zookeeperStarted.html#sc_ConnectingToZooKeeper> |
| 4 | >ruok  *<R u ok?>* | # To check zookeeper is ok (server status).  # nc works as a telnet. Nc connect & port are mandatory  >echo ruok | nc localhost 2181 |
| 5 | >conf | # To check the server configuration (from zoo.cfg) |
|  | >cons | # List full connection/session details for all clients connected to this server. Includes information on numbers of packets received/sent, session id, operation latencies, last operation performed, etc |
|  | >envi | # Server Environment |
|  | >srvr | # Statistics.znodes, mode(standalone, leader, follower) |
|  | >stat | # server statistics and connected clients |
|  | >srst | #Rest server statistics |
|  | >isro | # Is it in read-only(ro) mode? |
|  | >wchs | #  Lists brief information on watches for the server |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **Basic admin commands** | Non - 4 character commands |
| **1** | **Manage znode**  *<persistent node>* | 1. Persistent node - Data will remain stored even zookeeper service stopped 2. Ephemeral node - When the session ends the znode is deleted. It can be child but It cannot have children. It is useful to identify who are all connected 3. Sequential node |
| 1.1 | Create znode (or)  Set znode | # Create or set a znode jan01 with 3 file name. it will be a master directory and it doesn’t have  >create /jan01 “dir1,dir2,dir3” |
| 1.2 | Get znode (or) ls znode | # ‘get’ is recommended compare to ‘ls’  # To get znode  >get /jan01 |
| 2 | **Manage znode**  *<ephemeral node>* | 1. Persistent node 2. Ephemeral node - When the session ends the znode is deleted. It can be child but It cannot have children. It is useful to identify who are all connected 3. Sequential node |
| 2.1 | Create ephemeral node | >create –e /jan01/ephTest “eph” |
| 2.2 | Get ephemeral node | >get /jan01 |
| 2.3 | Deleted automatically | # As soon as your session is inactive, it will be deleted automatically ‘in few mins’ |
| **3** | **Manage znode**  *<sequential node>* | 1. Persistent node 2. Ephemeral node 3. Sequential node |
| 3.1 | Create sequential node | >create –s /jan01/ “1” |
| 3.2 |  |  |
|  |  |  |