ZooKeeper Ensemble Setup Guide

Please be careful when you copy paste commands from this document. Often characters such as - gets translated to some garbage from a PDF.

Setup

The following steps are for a ZooKeeper ensemble of 3, but you can set up more if you would like to. It is recommended that you chose an odd number of Zookeeper servers.

It is important to follow the installation directory names as-is. Not doing so can result in the template code given to you to not compile/execute as those scripts assume the installation directory mentioned below. You may have to edit the corresponding scripts to make appropriate changes if that is the case.

The examples below use tr-open-01, tr-open-02 and tr-open-03 to run the ZooKeeper servers. Replace these names with the actual servers that you will be using. If you and your group partner use the same lab servers for the ZooKeeper ensemble setup, they will compete with each other for the same network ports and you will run into issues. So if you and your group partner plan to work concurrently on the tasks, pick each different sets of lab servers for ZK ensemble setup.

1. From the command line of one the servers, download ZooKeeper binary to your home directory.
   ```
   $ wget https://dlcdn.apache.org/zookeeper/zookeeper-3.8.4/apache-zookeeper-3.8.4-bin.tar.gz
   ```

2. Extract the contents of the tar file.
   ```
   $ tar -zxvf ~/apache-zookeeper-3.8.4-bin.tar.gz
   ```

3. Create a separate data directory for each ZooKeeper servers (remember to use server names that you are planning to work with).
   ```
   $ cd ~/apache-zookeeper-3.8.4-bin
   $ for h in tr-open-01 tr-open-02 tr-open-03
     do
       mkdir -p $h/data
     done
   ```

4. Make a copy of the sample config file.
   ```
   $ cd conf
   $ cp -p zoo_sample.cfg zoo-base.cfg
   ```

5. Edit the `zoo-base.cfg` config file to modify/add the following contents. This is where you are letting each member know who are the other members in the ZooKeeper ensemble, the port for communications, including that for the client, etc. Do not remove any other existing variables in the config file which you are not modifying/adding.

   Replace tr-open-01, etc., with the servers you have decided to use for your ensemble. replace XX with your group number (e.g. group 4 -> 21804 for client port and so forth.)

   ENSURE THAT THERE ARE NO EXTRA SPACES IN THESE LINES (ESPECIALLY AT THE END OF THE LINES). In vi, you can use `:se list` to double check these invisible problems.

```
tickTime=2000
initLimit=10
syncLimit=5
dataDir=./data
clientPort=218XX
server.1=tr-open-01.cs.mcgill.ca:222XX:223XX
server.2=tr-open-01.cs.mcgill.ca:222XX:223XX
server.3=tr-open-01.cs.mcgill.ca:222XX:223XX
```

6. We need to give unique identifiers to each member of the ensemble. We will use the `n` from `server.n` in the above config file for this. (Again, remember to replace the server names with the ones that you are actually using).

   The following commands are to be executed from the `apache-zookeeper-3.8.4-bin` directory.

```
$ cd ~/apache-zookeeper-3.8.4-bin
$ echo 1 > tr-open-01/data/myid
$ echo 2 > tr-open-02/data/myid
$ echo 3 > tr-open-03/data/myid
```

Startup / Shutdown of ZooKeeper ensemble.

Startup

7. Log in to each of the three lab servers that you had configured for the ZooKeeper ensemble and execute the following steps in ALL of them.

   ```
   $ cd ~/apache-zookeeper-3.8.4-bin/$(hostname)
   $ ~/apache-zookeeper-3.8.4-bin/bin/zkServer.sh start zoo-base.cfg
   ```

8. You can verify the ZooKeeper process is up and running by executing the following command in corresponding servers.

   ```
   $ jps -l
   ```

If you want to test the setup, you can go over to the next section on command line CLI examples.

9. If you need to take a look at individual ZooKeeper server logs, they can be found here

   ```
   $ cd ~/apache-zookeeper-3.8.4-bin/logs
   ```

Shutdown

10. Login to each of the servers and execute the following steps.

    ```
    $ cd ~/apache-zookeeper-3.8.4-bin/$(hostname)
    $ ~/apache-zookeeper-3.8.4-bin/bin/zkServer.sh stop zoo-base.cfg
    ```

Command line client CLI examples

Zookeeper comes with a command line client where you can submit the ZooKeeper API commands in an interactive mode .

11. To connect to the ZooKeeper ensemble using client command line interface CLI, execute the following command from any lab server (need not be part of the ZooKeeper ensemble). You must give the addresses of all the zookeeper machines. The underlying library will then connect to one of them.

    REMEMBER to replace tr-open-01, etc., with your server names running the ZooKeeper ensemble and XX with your project group number.

    ```
    $ cd ~
    $  ~/apache-zookeeper-3.8.4-bin/bin/zkCli.sh -server tr-open-01.cs.mcgill.ca:218XX,tr-open-02.cs.mcgill.ca:218XX,tr-open-02.cs.mcgill.ca:218XX
    ```

    There is no space between the comma (,) and the next servername in the above arguments.

    This will give you a prompt something like below (edited to keep it short). You can type your client CLI commands here.

    ```
    [zk: ...(CONNECTED) 0]
    ```

12. To disconnect the client CLI, simply issue the command, quit.

    ```
    [zk: ...(CONNECTED) 0] quit
    ```

13. To list the contents use ls command

    ```
    [zk: ...(CONNECTED) 0] ls /
    [zookeeper]
    ```

14. Create a new znode using the create command

    ```
    [zk: ...(CONNECTED) 0] create /myapp ""
    ```

15. You can also create znodes with some data in it.

    ```
    [zk: ...(CONNECTED) 0] create /myapp/workers "All the worker znodes will be under this"
    ```

    Remember that you NEED to use the full path in ZooKeeper. There is no relative path concept.

    ```
    [zk: ...(CONNECTED) 0] ls /myapp
    [workers]
    ```

16. You can create ephemeral znodes using the -e option of create. These are automatically removed when the client disconnects.

    ```
    [zk: ...(CONNECTED) 0]create -e /myapp/workers/worker-tr-open-01
    "Hello from tr-open-01"
    ```

17. You can create sequential znodes (persistent or ephemeral) using the -s option of create. These automatically append a sequence id (based on the number of nodes already under that znode).

    ```
    [zk: ...(CONNECTED) 0]create  -s /myapp/workers/worker- "Hello, I
    am another worker"
    Created /myapp/workers/worker-0000000004
    [zk: ...(CONNECTED) 1]ls /myapp/workers
    [worker-0000000004, worker-tr-open-01]
    ```

18. To retrieve the “data” element of a znode, use the get command.

    ```
    [zk: ...(CONNECTED) 0]get /myapp/workers
    All the worker znodes will be under this
    ```

19. You can delete a znode using the delete command if it is not empty (has no child znodes).

    ```
    [zk: ...(CONNECTED) 0] delete /myapp/workers/worker-tr-open-01
    ```

20. To recursively delete a znode and everything under it, use the delete all command.

    ```
    [zk: ...(CONNECTED) 0] deleteall /myapp
    [zk: ...(CONNECTED) 1] ls /
    [zookeeper]
    ```

21. More ZooKeeper CLI commands (including some examples of setting simple watches from the command line) can be found here.
    https://zookeeper.apache.org/doc/r3.8.4/zookeeperCLI.html

ZooKeeper Programming and Execution Environment Setup

22. For successful compilation and execution of ZooKeeper based java programs, make sure you setup the ZooKeeper environment by executing

    ```
    $ export ZOOBINDIR=~/apache-zookeeper-3.8.4-bin/bin
    $ . $ZOOBINDIR/zkEnv.sh
    ```

    In the shell used for compilation as well as for executing the corresponding java programs.

23. Good Luck !