

BookKeeper Getting Started Guide

by

Table of contents

| | |
|---|---|
| 1 Getting Started: Setting up BookKeeper to write logs..... | 2 |
| 1.1 Pre-requisites..... | 2 |
| 1.2 Download..... | 2 |
| 1.3 LocalBookKeeper..... | 2 |
| 1.4 Setting up bookies..... | 2 |
| 1.5 Setting up ZooKeeper..... | 3 |
| 1.6 Example..... | 3 |

1 Getting Started: Setting up BookKeeper to write logs.

This document contains information to get you started quickly with BookKeeper. It is aimed primarily at developers willing to try it out, and contains simple installation instructions for a simple BookKeeper installation and a simple programming example. For further programming detail, please refer to [BookKeeper Programmer's Guide](#).

1.1 Pre-requisites

See [System Requirements](#) in the Admin guide.

1.2 Download

BookKeeper is distributed along with ZooKeeper. To get a ZooKeeper distribution, download a recent [stable](#) release from one of the Apache Download Mirrors.

1.3 LocalBookKeeper

Under `org.apache.bookkeeper.util`, you'll find a java program called `LocalBookKeeper.java` that sets you up to run BookKeeper on a single machine. This is far from ideal from a performance perspective, but the program is useful for both test and educational purposes.

1.4 Setting up bookies

If you're bold and you want more than just running things locally, then you'll need to run bookies in different servers. You'll need at least three bookies to start with.

For each bookie, we need to execute a command like the following:

```
java -cp ../zookeeper-<version>-bookkeeper.jar:./
zookeeper-<version>.jar\ :lib/slf4j-api-1.6.1.jar:lib/
slf4j-log4j12-1.6.1.jar:lib/log4j-1.2.15.jar
-Dlog4j.configuration=log4j.properties\
org.apache.bookkeeper.proto.BookieServer 3181 127.0.0.1:2181 /
path_to_log_device/\ /path_to_ledger_device/
```

`"/path_to_log_device/"` and `"/path_to_ledger_device/"` are different paths. Also, port 3181 is the port that a bookie listens on for connection requests from clients. `127.0.0.1:2181` is the hostname:port for the ZooKeeper server. In this example, the standalone ZooKeeper server is running locally on port 2181. If we had multiple ZooKeeper servers, this parameter would be a comma separated list of all the hostname:port values corresponding to them.

1.5 Setting up ZooKeeper

ZooKeeper stores metadata on behalf of BookKeeper clients and bookies. To get a minimal ZooKeeper installation to work with BookKeeper, we can set up one server running in standalone mode. Once we have the server running, we need to create a few znodes:

1. /ledgers
2. /ledgers/available
3. For each bookie, we add one znode such that the name of the znode is the concatenation of the machine name and the port number that the bookie is listening on. For example, if a bookie is running on bookie.foo.com and is listening on port 3181, we add a znode /ledgers/available/bookie.foo.com:3181.

1.6 Example

In the following excerpt of code, we:

1. Create a ledger;
2. Write to the ledger;
3. Close the ledger;
4. Open the same ledger for reading;
5. Read from the ledger;
6. Close the ledger again;

```
LedgerHandle lh = bkc.createLedger(ledgerPassword);
ledgerId = lh.getId();
ByteBuffer entry = ByteBuffer.allocate(4);

for(int i = 0; i < 10; i++){
    entry.putInt(i);
    entry.position(0);
    entries.add(entry.array());
    lh.addEntry(entry.array());
}
lh.close();
lh = bkc.openLedger(ledgerId, ledgerPassword);

Enumeration<LedgerEntry> ls = lh.readEntries(0, 9);
int i = 0;
while(ls.hasMoreElements()){
    ByteBuffer origbb = ByteBuffer.wrap(
        entries.get(i++));
    Integer origEntry = origbb.getInt();
    ByteBuffer result = ByteBuffer.wrap(
        ls.nextElement().getEntry());

    Integer retrEntry = result.getInt();
}
lh.close();
```

