Assignment 4 - Introduction to Apache Zookeeper

**In this lab, you will learn how to:**

* Install and Configure Apache Zookeeper
* Use the Zookeeper CLI to interact with the Zookeeper Ensemble

**What is Zookeeper?**

Zookeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. The implementation of distributed applications is heavy on fixing bugs, race conditions, handling updates, change management, etc. The Zookeeper service helps manage all of this and more with the various APIs and services it implements.

[More on Apache Zookeeper](https://levelup.gitconnected.com/what-is-apache-zookeeper-540cdff6d45f)

**Deliverables:**

1. Screenshot of Zookeeper Server having started.
2. Screenshot of Listing of the created ZNodes
3. Screenshot of Listing of the deleted ephemeral node
4. Screenshot of Data and Metadata associated with the created sequential znode.
5. Screenshot of Listing of all the children nodes
6. Screenshot of each of the clients listing all the znodes
7. Screenshot of the Watched Event and the Zookeeper Server logs.

**Prerequisites:**

1. Linux OS
2. Java (JDK 6 or greater)

To verify installation/ install, refer [this](https://docs.google.com/document/d/1KQiwwTps2VuFK96LPYzcKAdy3LsqdlzYTqTAVm04pXc/edit?usp=sharing) document.

**TASK A: Installing Zookeeper**

1. Download the Zookeeper framework onto your system from [here](https://www.apache.org/dyn/closer.lua/zookeeper/zookeeper-3.8.0/apache-zookeeper-3.8.0-bin.tar.gz).
2. Extract the tar file using the following command -

tar -zxf apache-zookeeper-3.8.0-bin.tar.gz

1. Move the extracted file to /opt directory.

sudo mv /path/to/download/apache-zookeeper-3.8.0-bin /opt

cd /opt/apache-zookeeper-3.8.0-bin

1. Create a Data Directory

mkdir data

1. Create a Configuration file (conf/zoo.cfg) for Zookeeper with the following parameters -

tickTime = 2000

dataDir = /path/to/zookeeper/data

clientPort = 2181

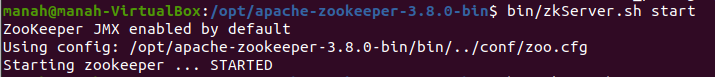
initLimit = 5

syncLimit = 2

1. Start the Zookeeper Server

bin/zkServer.sh start

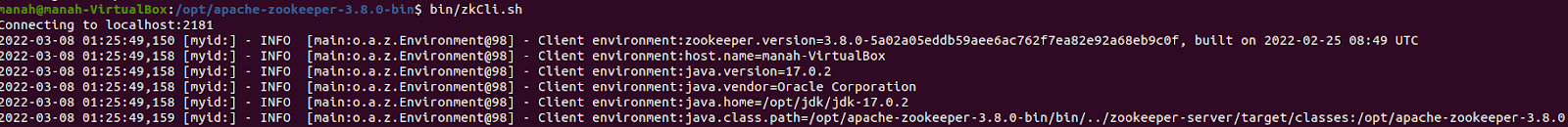
The output should resemble -

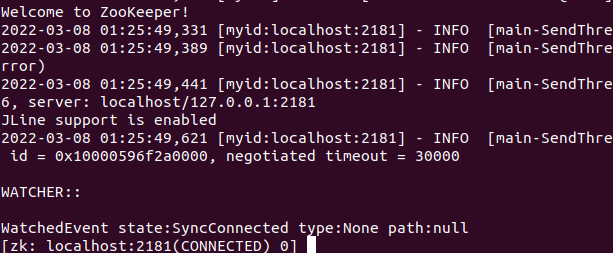


1. Start the Zookeeper CLI

bin/zkCli.sh

The output should resemble -



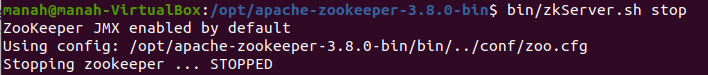


Use quit command to exit from the ZK CLI.

1. Stop the Zookeeper Server

bin/zkServer.sh stop

The output should resemble -



**TASK B: Getting Familiar with Zookeeper CLI**

Start the Zookeeper Server and the Zookeeper CLI.

1. Create Znodes

Usage: create [FLAGS] /path /data

The flags are used to specify the type of znode.

* Ephemeral znodes (flag e) will be automatically deleted when a session expires, or a client disconnects.
* Sequential znodes (flag s) are those nodes that guarantee that the path will be unique. Each sequential znode that gets created is suffixed a sequential counter with 10 digit padding to do so. These nodes are persistent.

1. Create a sequential node and name it with your\_srn\_seq

create -s /your\_srn\_seq “a sequential node”

1. Create an ephemeral node and name it with your\_srn\_eph

create -e /your\_srn\_eph “an ephemeral node”

1. List the nodes

ls /

Notice the sequence appended to the sequential node. Take a screenshot.

1. Now, quit the Zookeeper Client and start it up again after an approximate of 30s. List all the nodes.

You will observe that the ephemeral node does not exist anymore while the sequential node does.

Take a screenshot.

1. Set and Get Data

Usage: set /path /data

Affix the string “updated” to the data of the sequential node created.

Get the data and metadata associated with the specified znode.

Usage: get -s /path

Get the information related to the sequential node, and take a screenshot.

get -s /your\_srn\_seq<some\_sequence>

One may use get -s -w /path to add watch, which you will do in the next section.

1. Create Children/ Sub-Znode

Creating children is similar to creating new znodes. The only difference is that the path of the child znode will have the parent path as well.

Usage: create /parent/path/subnode/path /data

create /parent

create /parent/child “child node”

Let the parent be your\_srn and child be your\_name

Create two children, one with your first name and one with your last name.

List all children. Take a screenshot.

1. Delete Znode

Usage: delete /path

OR

Usage: rmr /path

**TASK B: Naive Leader Election with Zookeeper CLI - Understanding the Algorithm**

Leader Election Algorithm in Zookeeper:

Every sequential ephemeral znode is associated with a sequence as observed earlier. The znode with the smallest sequence is the leader, and all other znodes are followers. Each follower sets a ‘watch’ on the znode sequentially before it. If the leader goes down, the follower that is sequentially the next gets notified of the event because of the watch. It checks if there is a znode with a sequentially smaller number, and if not, elects itself as leader.

In the following task, you will explore how ‘watches’ work.

a) Set-Up the Server and clients with znodes.

1. Spin up 4 terminals. One of these will run the zookeeper server while the other three will function as zookeeper clients.
   1. Start up the zookeeper server on one of the terminals. Run it in the foreground.

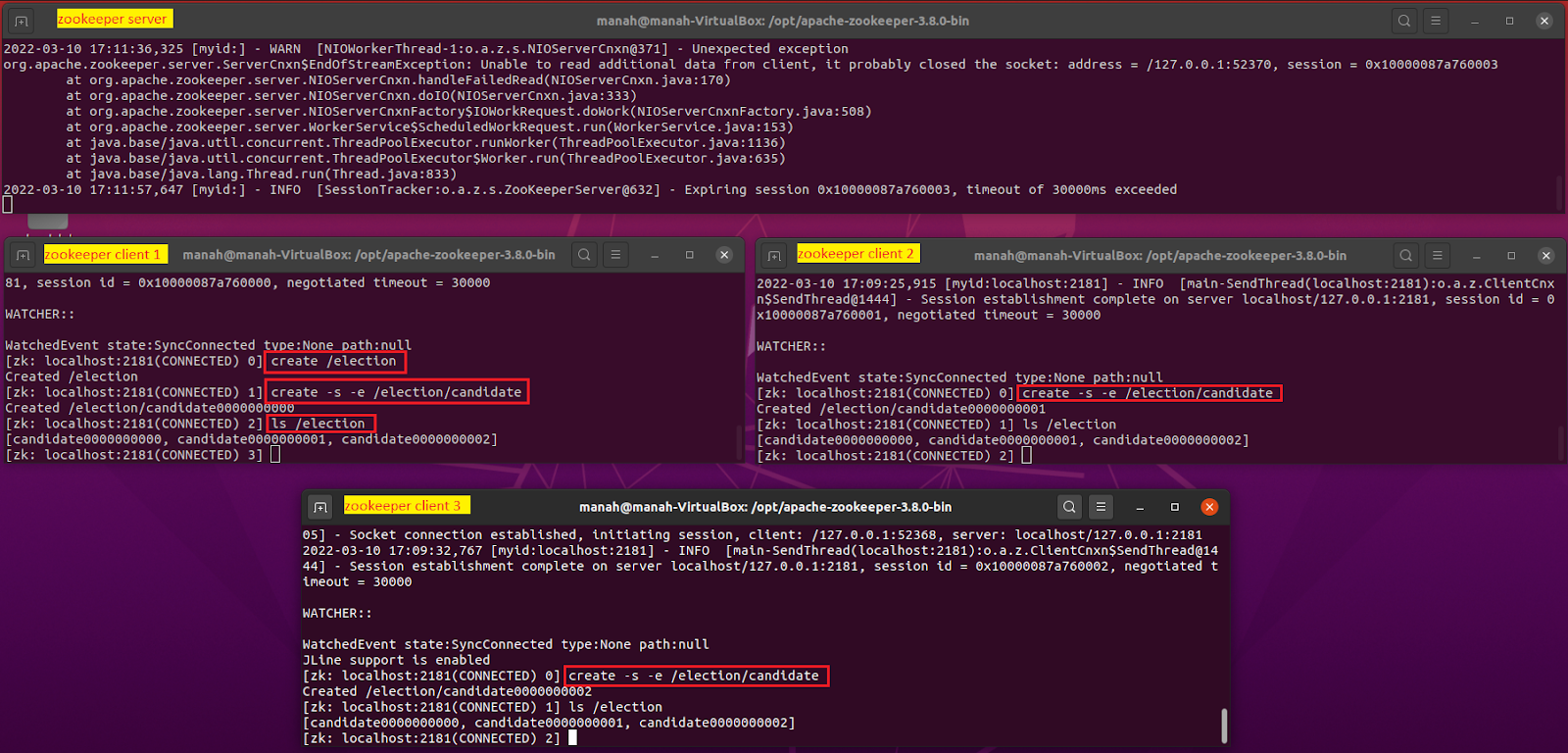
bin/zkServer.sh start-foreground

* 1. Start up the zookeeper cli on the other terminals. These will function as clients.

bin/zkCli.sh

1. On any one of the clients, under the root, create a znode and name it with your\_srn. This will function as the root znode for this task. Do not specify any type.
2. On each of the clients, create a sequential, ephemeral znode under the root znode created earlier. I.e., under /your\_srn. You may name these znodes ‘candidate’.
3. List all the nodes under the root znode (/your\_srn) on each of the clients. Take a screenshot.

Every client is now running a znode. For an example, refer the screenshot below:

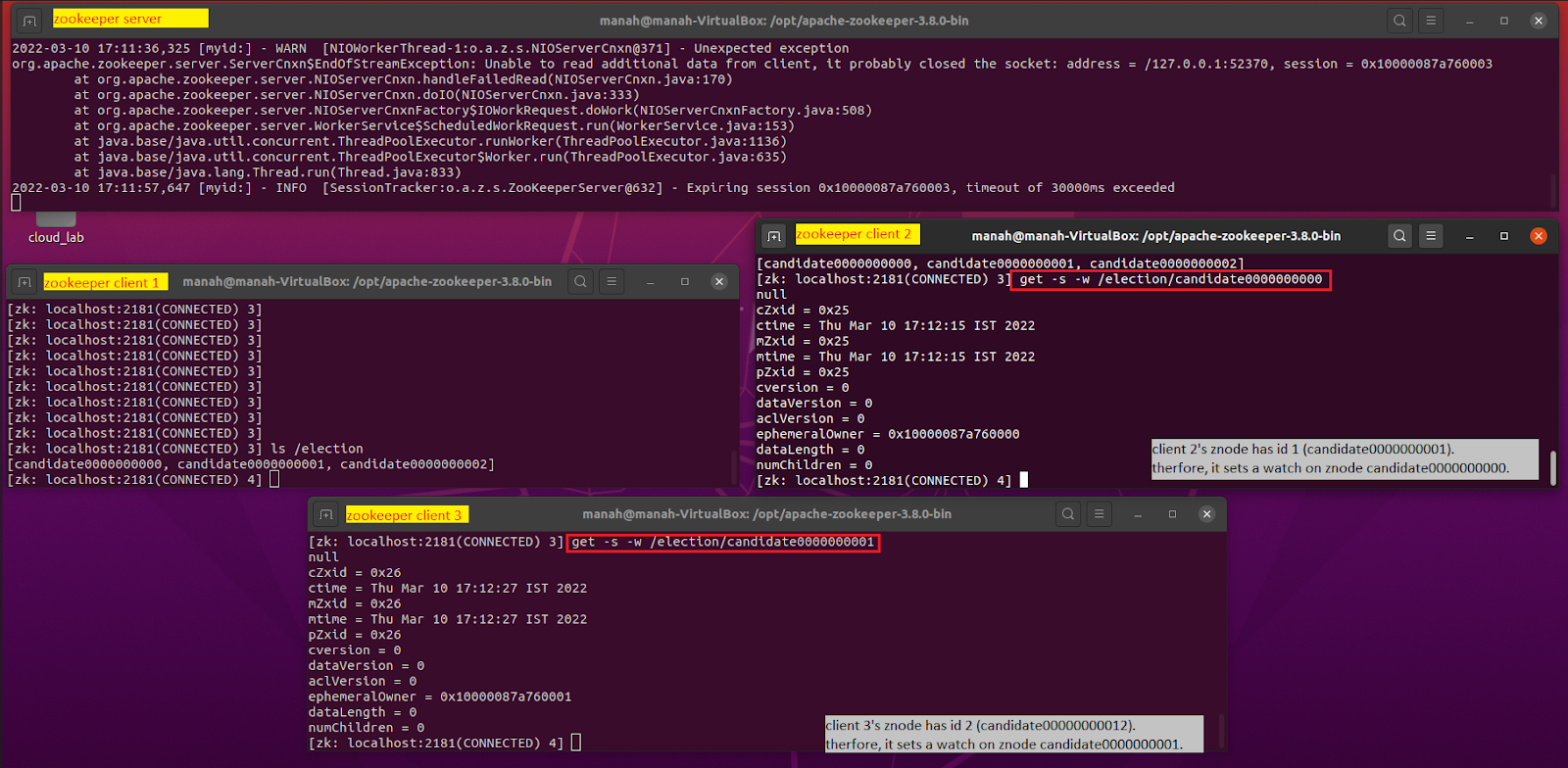


b) Implement Leader Election

Leader Election is implemented in a manner such that every znode ‘watches’ the znode that is sequentially before it. If the znode that is the leader fails, the znode watching the leader receives a watch event and elects itself as leader.

(-w is for watch – would this be persistent or non-persistent)

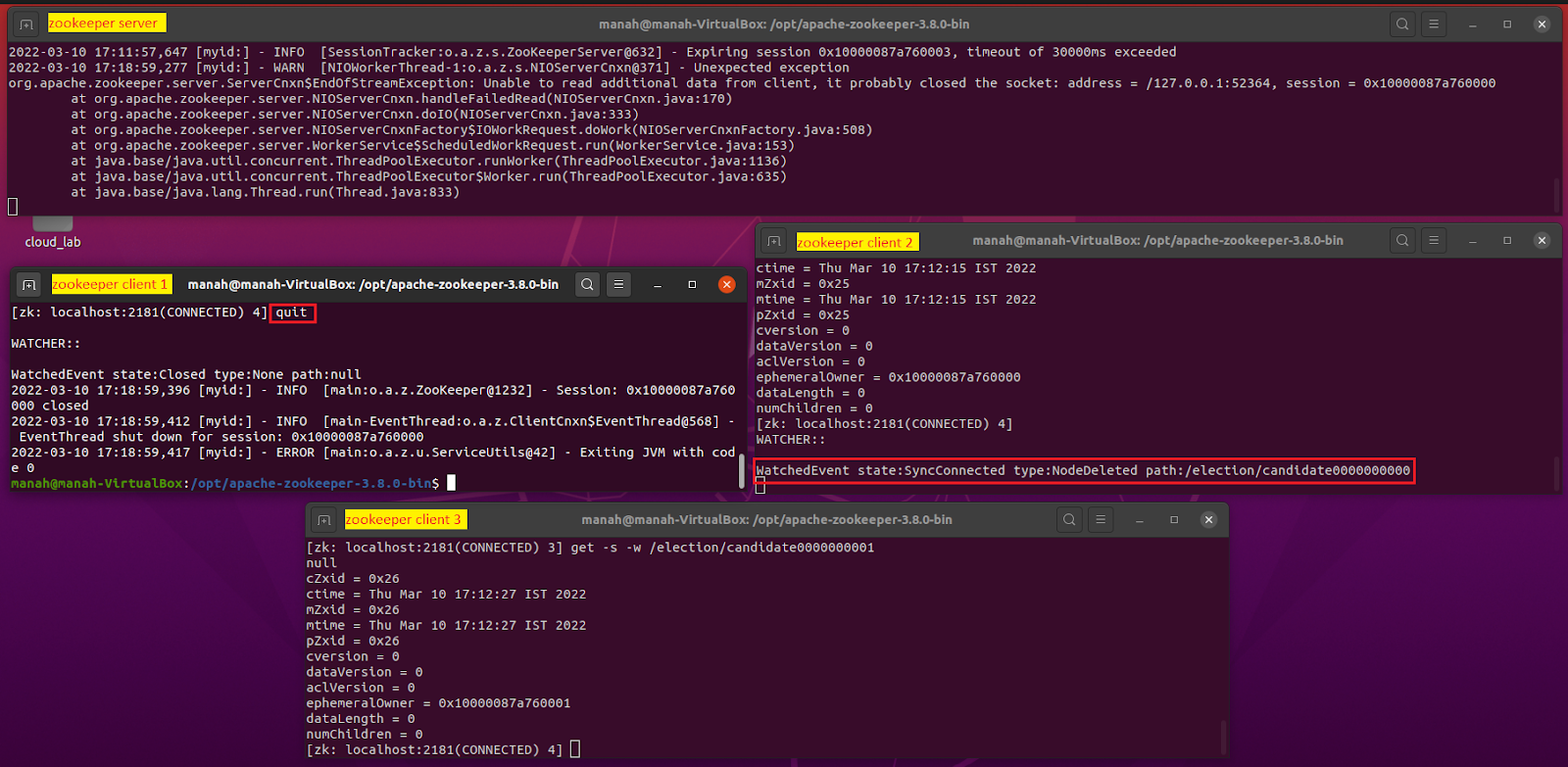
1. Setup watches such that every znode ‘watches’ the znode that’s sequentially before it.



1. Now, quit any of the clients (except the client which had the sequentially greatest znode/ the one with no watches set on it). Since the znodes created were ephemeral, the znodes will die as well.
2. Watch the logs on the zookeeper server terminal as well as the zookeeper client terminal that set a watch on the client whose znode was killed.

Observe the WatchedEvent logs and take a screenshot that shows both the zookeeper logs and the WatchedEvent logs.

Observe the details presented in the log, see how it specifies a NodeDeleted event.



**[OPTIONAL]**

Apache also provides a Zookeeper API, using which an application can connect, interact, manipulate data, coordinate, and finally disconnect from a Zookeeper ensemble. This API may be used to implement the leader election logic and more. You may explore the same [here](https://www.allprogrammingtutorials.com/tutorials/leader-election-using-apache-zookeeper.php).