# Lab 3: Fault-Tolerant Applications with Apache ZooKeeper

## Introduction to Apache ZooKeeper

Apache ZooKeeper is a framework for building fault-tolerant applications. It provides distributed coordination, configuration management, leader election, and service discovery, making it a crucial component for highly available systems. ZooKeeper ensures reliability even in the presence of failures and is widely used in distributed systems to maintain fault tolerance.

For more information, refer to the official documentation: <https://zookeeper.apache.org/>

## Objective

In this lab, you will set up a distributed payment processing system using Apache ZooKeeper. You will:

- Import an existing Eclipse project `ftdemo`.  
- Configure external JAR libraries for ZooKeeper.

* Run ZooKeeper Server in Terminal.
* Open ZooKeeper CLI Client.
* Set up PaymentServer project in Eclipse.
* Run the PaymentServer.java to simulate a dummy server (you can start multiple servers).
* Configure a payment client to send requests using random selection or round-robin selection.
* Kill the server and analyze its impact.
* Observe how requests are distributed among servers.
* Modify the load-balancing strategy and evaluate its effectiveness.
* Answer discussion questions on distributed systems concepts.

## Pre-requisites

- Eclipse IDE installed.  
- Apache ZooKeeper binaries included in the Eclipse project directory.  
- Basic knowledge of Java and distributed systems.  
- The Eclipse project (`ftdemo`) is already provided.

## Step 1: Import the Eclipse Project

1. Open Eclipse and select your workspace.  
2. Click File > Import...  
3. Select Existing Projects into Workspace under General, then click Next.  
4. Click Browse and locate the `ftdemo` project folder.  
5. Ensure Copy projects into workspace is unchecked.  
6. Click Finish to import the project.

## Step 2: Configuring Apache ZooKeeper Libraries

Since the absolute path of the external JAR files may vary, you may have to manually configure them (This step has to be done only if dependency errors occur related to ZooKeeper Library):  
  
1. Right-click the project in Eclipse and select Properties.  
2. Go to Java Build Path > Libraries tab.  
3. Remove any existing incorrect or broken ZooKeeper JAR references (make sure not to remove the JRE system library).  
4. Click Add External JARs...  
5. Navigate to the `lib` directory inside ‘apache-zookeeper/lib’ in your `ftdemo` project, which contains ZooKeeper libraries.  
6. Select all JAR files in the `lib` directory and click Open.  
7. Click Apply and Close.

## Step 3: Running ZooKeeper

1. Start ZooKeeper:  
 - Open Command Prompt and navigate to the `bin` directory inside ‘apache-zookeeper’ directory the `ftdemo` project:  
 cd apache-zookeeper\bin

- Start ZooKeeper with:  
zkServer.cmd **(**for Mac and Linux use ./zkServer.sh start)  
 **- Keep this terminal open.**  
  
2. Verify ZooKeeper is running:  
 - Open another Command Prompt.  
 - Run the ZooKeeper CLI:  
 zkCli.cmd (for Mac and Linux use ./zkCli.sh)  
 - In the ZooKeeper CLI, type:  
 ls /  
 - You should see output similar to: [zookeeper].  
 - Type quit to exit.

## Step 4: Running the Payment System

1. Start multiple servers:  
 - In Eclipse, right-click on `PaymentServer.java`.  
 - Select Run As > Java Application.  
 - Repeat this step in multiple Eclipse windows to start multiple servers.  
 - Note: The `PaymentServer` is a dummy server that does not perform actual payment processing. It simply prints a message whenever it receives a request.  
  
2. Start the payment client:  
 - Right-click on `PaymentClient.java`.  
 - Select Run As > Java Application.  
 - The client will send payment requests at 3-second intervals.  
 - By default, it will randomly distribute requests across servers.

## Step 5: Observing Failures

1. Kill a server:  
 - In Eclipse, while the payment requests are being sent, manually stop one of the running `PaymentServer` instances by clicking the Terminate button.  
 - Observe the client logs.  
 - Discuss:  
 - What happens to the requests that were supposed to go to the killed server?  
 - How does ZooKeeper handle this failure?

## Step 6: Changing the Load Balancing Strategy

1. Modify `PaymentClient.java` to use round-robin selection:  
 - Comment out the random selection code.  
 - Uncomment the round-robin selection code.  
 - Recompile and run the client.  
 - Observe whether the requests are evenly distributed.  
  
2. Questions:  
 - Why does round-robin ensure even distribution of requests?  
 - What are the advantages and disadvantages of each approach?

## Submission

Upload the updated project and the answers in a text file in a single zip file to the courseweb link. The zip file name should your registration number.

**Note:** If the submission file is larger than the maximum file limit of courseweb, remove the zookeeper directory inside the project directory and upload.

## Optional Exercise

Implement a load balancer as a separate class which will manage the servers and the clients may send the requests to the load balancer and not directly to the servers. The load balancer should direct the requests to the servers.

# Steps to Fix Zookeeper Startup Issues in Windows

## Check for Running Zookeeper Processes

Open Command Prompt (cmd) as Administrator and run:  
tasklist | findstr /i "zookeeper"  
If a Zookeeper process is running, note its Process ID (PID).

## Kill the Zookeeper Process

Terminate the existing process using:  
taskkill /F /PID <PID>  
Replace <PID> with the actual process ID.

## Check for Port Conflicts

If Zookeeper still doesn’t start, check if port 2181 (default Zookeeper port) is already in use:  
netstat -ano | findstr :2181  
If a process is using this port, note its PID.

## Kill the Process Using the Port

Stop the process that is blocking the Zookeeper port:  
taskkill /F /PID <PID>  
Replace <PID> with the actual process ID from Step 3.

## Disable Firewall (If Needed)

If the server still fails to start, temporarily disable the Windows Firewall:  
netsh advfirewall set allprofiles state off  
⚠ Warning: Turning off the firewall may expose your system to security risks. Enable it again after troubleshooting:  
netsh advfirewall set allprofiles state on

## Restart Zookeeper

Once you have completed the above steps, try starting Zookeeper again and verify if it is running:  
tasklist | findstr /i "zookeeper"