

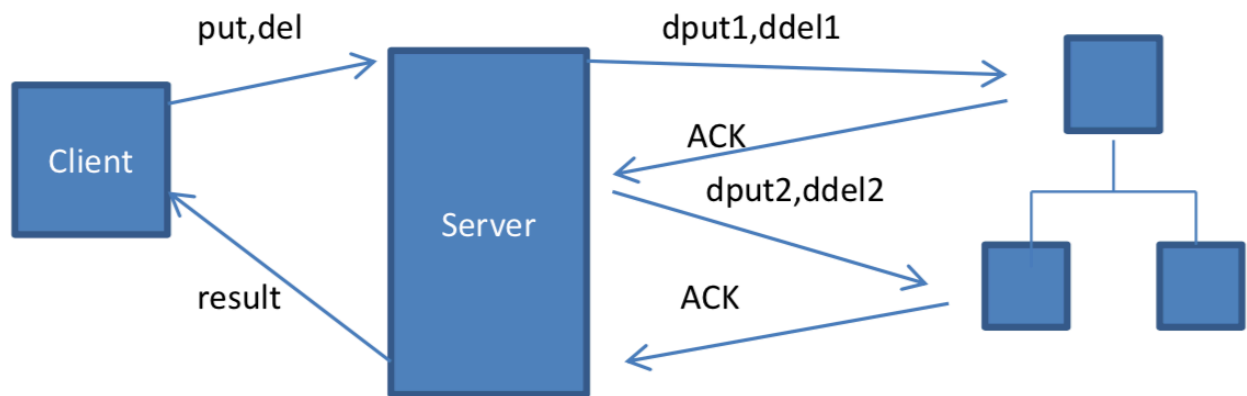
# Team: Ibrahim Diabate, Lingwei Meng, Yuxiao Guo

Tracking method implemented: TCP centralized membership KV Store

To run the system - AWS:

## Structure:

### Structure



## How to use:

### 1) First start the membership server:

- Open the **docker\_server** in one terminal window.
- Open **Dockerfile** and make sure **COPY MembershipServer.jar /** is uncommented.
- Make sure **COPY GenericNode.jar /** is commented.
- Open **runserver.sh** and make sure the line **java -jar MembershipServer.jar ts 4410** is the only line uncommented.
- Save, then build & run the membership server with the following lines:
  - **\$sudo docker build -t nodelist .**
  - **\$sudo docker run -d --rm -p 4410:4410 nodelist**
  - # Port must be 4410.

- Use the **exec -it** command to get the IP of this container (membership server). Save that ip address.

### 2) Launch the remaining servers:

- Open a new terminal window.
- Access the **docker\_server** directory

# Team: Ibrahim Diabate, Lingwei Meng, Yuxiao Guo

- Open **Dockerfile** and make sure **COPY GenericNode.jar /** is uncommented.
  - Make sure the line **COPY MembershipServer.jar /** is commented.
- Open **runserver.sh** and make sure the line **java -jar MembershipServer.jar ts 4410** is commented
  - Make sure the line **java -jar MembershipServer.jar ts 4410 <ip-address>** is uncommented.
  - Add the ip address that you saved in the previous step where **<ip-address>** is.
  - Save changes.
- Build & launch servers with the following lines. In this case we are launching 3 servers, mapping to different ports.
  - **\$sudo docker build -t nodes .**
  - **\$sudo docker run -d --rm -p 1234:1234 nodes**
  - **\$sudo docker run -d --rm -p 1235:1234 nodes**
  - **\$sudo docker run -d --rm -p 1236:1234 nodes**
- Use the **exec -it** command on each of these nodes to display IP addresses of each of these 3 nodes. Save those ip addresses, in the next step, we refer to them with the **<ip\_of\_node#>** tag.

## 3) Test with client requests

- Open a third terminal window and access the **docker\_client** folder.
- Build and launch a client container:
  - **\$sudo docker build -t client .**
  - **\$sudo docker run -d --rm client**
- Use the **exec -it** command to log onto the client container.
- From there, send tcp put, get or del requests to each of the servers:  
E.g:

```
java -jar GenericNode.jar tc <ip_of_node1> 1234 put a 555
java -jar GenericNode.jar tc <ip_of_node2> 1234 put b 321
java -jar GenericNode.jar tc <ip_of_node3> 1234 put c 111
java -jar GenericNode.jar tc <ip_of_node2> 1234 put h 222
java -jar GenericNode.jar tc <ip_of_node3> 1234 put z 444
java -jar GenericNode.jar tc <ip_of_node1> 1234 del a
```

P.S: Send different requests to either of the **<ip\_of\_node#>** addresses.

## 4) Were requests replicated?

- Send client store requests to the different ip addresses. The result should be the same for each server ip.  
E.g:

## Team: Ibrahim Diabate, Lingwei Meng, Yuxiao Guo

```
java -jar GenericNode.jar tc <ip_of_node3> 1234 store  
java -jar GenericNode.jar tc <ip_of_node2> 1234 store  
java -jar GenericNode.jar tc <ip_of_node1> 1234 store
```

The end.