

RAFT Implementation - Quick Start Guide

Github Repository: [RAFT](#)

Step-by-Step Setup (Ensure you're in the source code root directory)

1. Install Dependencies (1 minute)

```
pip install -r requirements.txt
```

This installs:

- grpcio (gRPC framework)
- grpcio-tools (Protocol buffer compiler)
- protobuf (Protocol buffer runtime)

2. Generate gRPC Code (10 seconds)

```
python scripts/generate_proto.py
```

This generates:

- `proto/raft_pb2.py` - Protocol buffer messages
- `proto/raft_pb2_grpc.py` - gRPC service stubs

3. Start the Cluster (5 seconds)

```
python scripts/start_cluster.py
```

This starts 5 nodes on ports 5001-5005. Keep this terminal window open.

4. Use the Client (in a new terminal)

```
python scripts/client.py
```

Try these commands:

```
raft> SET name Alice  
raft> SET age 25  
raft> GET name
```

```
raft> DELETE age
raft> exit
```

Testing the Implementation

Quick Test (2 minutes)

In a new terminal while cluster is running:

```
# Test leader election
python tests/test_leader_election.py

# Test log replication
python tests/test_log_replication.py
```

Full Test Suite (10 minutes)

```
# Run all tests
python tests/run_all_tests.py
```

Manual Testing Scenarios

Test 1: Leader Failure

1. Start cluster: `python scripts/start_cluster.py`
2. Note which node is leader (check terminal output)
3. Kill the leader process (Ctrl+C in that node's window)
4. Wait 3-5 seconds
5. Use client to submit command - should work with new leader!

Test 2: Network Partition

1. Start cluster
2. Run: `python tests/test_network_partition.py`
3. Watch as cluster splits into majority/minority
4. Observe only majority can commit
5. See cluster heal automatically

Common Commands

Start Individual Node

```
python scripts/run_node.py --node-id node1 --host localhost --port 5001 --peers
"node2=localhost:5002,node3=localhost:5003,node4=localhost:5004,node5=localhost:5005"
```

```
05"
```

Single Command Execution

```
python scripts/client.py --command "SET test value"
```

Check Stored Data

Look in the `data/` directory:

- `node_1_db.json` - Key-value store for node1
- `node_1_state.json` - RAFT state for node1
- (same for other nodes)

All committed data should be identical across nodes!

Troubleshooting

"No module named 'raft_pb2'"

Run: `python scripts/generate_proto.py`

"Address already in use"

Kill processes on ports 5001-5005:

```
# Windows
netstat -ano | findstr :500
taskkill /PID <pid> /F
```

"No leader found"

Wait 3-5 seconds after starting cluster for election to complete.

Performance Tips

- Election timeout: 150-300ms (adjust for faster/slower networks)
- Heartbeat interval: 50ms (keep < election_timeout/3)
- For more nodes: Increase timeouts proportionally