

ClickHouse Two-Node Cluster Setup Guide

This guide provides step-by-step instructions for setting up a ClickHouse two-node cluster with ZooKeeper coordination and replication.

Prerequisites

- **Two Ubuntu servers** (24.04 LTS recommended)
- **Passwordless SSH configured** between nodes
- **Java 11** installed on both nodes
- **Root/sudo access** on both nodes

Current System Status

✅ Already Configured:

- Hostnames set (clickhouse-primary, clickhouse-secondary)
- SSH passwordless access working
- ClickHouse installed on both nodes
- Java 11 available on both nodes

❌ Still Needed:

- ZooKeeper installation and configuration
- ClickHouse cluster configuration
- Service startup and testing

Network Configuration

- **Primary Node:** 10.149.252.178 → clickhouse-primary
- **Secondary Node:** 10.149.252.239 → clickhouse-secondary
- **ClickHouse Ports:** 8123 (HTTP), 9001 (TCP), 9009 (interserver)
- **ZooKeeper Ports:** 2181 (client), 2888 (peer), 3888 (leader election)

Quick Setup Commands

Step 1: Install ZooKeeper on Both Nodes

```
# Run on primary node
cd /tmp
wget https://archive.apache.org/dist/zookeeper/zookeeper-3.8.3/apache-
zookeeper-3.8.3-bin.tar.gz
sudo tar -xzf apache-zookeeper-3.8.3-bin.tar.gz -C /opt/
sudo mv /opt/apache-zookeeper-3.8.3-bin /opt/zookeeper

# Create zookeeper user and set permissions
sudo useradd -r -s /bin/false zookeeper 2>/dev/null || true
sudo chown -R zookeeper:zookeeper /opt/zookeeper
```

```
sudo mkdir -p /var/lib/zookeeper
sudo chown zookeeper:zookeeper /var/lib/zookeeper

# Copy to secondary node via SSH
scp -r /opt/zookeeper clickhouse-secondary:/opt/
ssh clickhouse-secondary 'sudo useradd -r -s /bin/false zookeeper
2>/dev/null || true'
ssh clickhouse-secondary 'sudo chown -R zookeeper:zookeeper /opt/zookeeper'
ssh clickhouse-secondary 'sudo mkdir -p /var/lib/zookeeper'
ssh clickhouse-secondary 'sudo chown zookeeper:zookeeper
/var/lib/zookeeper'
```

Step 2: Configure ZooKeeper

```
# Create ZooKeeper configuration on primary node
sudo tee /opt/zookeeper/conf/zoo.cfg > /dev/null <<EOF
tickTime=2000
initLimit=10
syncLimit=5
dataDir=/var/lib/zookeeper
clientPort=2181
maxClientCnxns=60
server.1=clickhouse-primary:2888:3888
server.2=clickhouse-secondary:2888:3888
EOF

# Copy config to secondary node
scp /opt/zookeeper/conf/zoo.cfg clickhouse-secondary:/opt/zookeeper/conf/

# Set server IDs
echo "1" | sudo tee /var/lib/zookeeper/myid
ssh clickhouse-secondary 'echo "2" | sudo tee /var/lib/zookeeper/myid'
```

Step 3: Create ZooKeeper Service

```
# Create service file on primary node
sudo tee /etc/systemd/system/zookeeper.service > /dev/null <<EOF
[Unit]
Description=Apache ZooKeeper server
Documentation=https://zookeeper.apache.org
After=network.target

[Service]
Type=simple
User=zookeeper
Group=zookeeper
ExecStart=/opt/zookeeper/bin/zkServer.sh start-foreground
ExecStop=/opt/zookeeper/bin/zkServer.sh stop
WorkingDirectory=/var/lib/zookeeper
```

```
Restart=on-failure

[Install]
WantedBy=multi-user.target
EOF

# Copy service to secondary node
scp /etc/systemd/system/zookeeper.service clickhouse-
secondary:/etc/systemd/system/

# Start ZooKeeper on both nodes
sudo systemctl daemon-reload
sudo systemctl start zookeeper
sudo systemctl enable zookeeper

ssh clickhouse-secondary 'sudo systemctl daemon-reload'
ssh clickhouse-secondary 'sudo systemctl start zookeeper'
ssh clickhouse-secondary 'sudo systemctl enable zookeeper'
```

Step 4: Configure ClickHouse Cluster

```
# Create cluster config on primary node
sudo tee /etc/clickhouse-server/config.d/cluster.xml > /dev/null <<'EOF'
<?xml version="1.0"?>
<clickhouse>
  <!-- Remote servers definition -->
  <remote_servers>
    <cluster_2shards_1replicas>
      <shard>
        <replica>
          <host>clickhouse-primary</host>
          <port>9001</port>
        </replica>
      </shard>
      <shard>
        <replica>
          <host>clickhouse-secondary</host>
          <port>9001</port>
        </replica>
      </shard>
    </cluster_2shards_1replicas>

    <!-- Replicated cluster for high availability -->
    <cluster_1shard_2replicas>
      <shard>
        <internal_replication>true</internal_replication>
        <replica>
          <host>clickhouse-primary</host>
          <port>9001</port>
          <user>default</user>
        </replica>
      </shard>
    </cluster_1shard_2replicas>
  </remote_servers>
</clickhouse>
```

```

        <replica>
            <host>clickhouse-secondary</host>
            <port>9001</port>
            <user>default</user>
        </replica>
    </shard>
</cluster_1shard_2replicas>
</remote_servers>

<!-- ZooKeeper configuration -->
<zookeeper>
    <node index="1">
        <host>clickhouse-primary</host>
        <port>2181</port>
    </node>
    <node index="2">
        <host>clickhouse-secondary</host>
        <port>2181</port>
    </node>
</zookeeper>

<!-- Macros for cluster configuration -->
<macros>
    <shard>1</shard>
    <replica>clickhouse-primary</replica>
</macros>

<!-- Enable distributed_ddl -->
<distributed_ddl>
    <path>/clickhouse/task_queue/ddl</path>
</distributed_ddl>
</clickhouse>
EOF

# Create cluster config on secondary node (different replica macro)
ssh clickhouse-secondary 'sudo tee /etc/clickhouse-
server/config.d/cluster.xml > /dev/null <<'\"'\"'EOF'\"'\"'
<?xml version="1.0"?>
<clickhouse>
    <!-- Remote servers definition -->
    <remote_servers>
        <cluster_2shards_1replicas>
            <shard>
                <replica>
                    <host>clickhouse-primary</host>
                    <port>9001</port>
                </replica>
            </shard>
            <shard>
                <replica>
                    <host>clickhouse-secondary</host>
                    <port>9001</port>
                </replica>
            </shard>
        </cluster_2shards_1replicas>
    </remote_servers>
</clickhouse>

```

```

        </cluster_2shards_1replicas>

        <!-- Replicated cluster for high availability -->
        <cluster_1shard_2replicas>
            <shard>
                <internal_replication>true</internal_replication>
                <replica>
                    <host>clickhouse-primary</host>
                    <port>9001</port>
                    <user>default</user>
                </replica>
                <replica>
                    <host>clickhouse-secondary</host>
                    <port>9001</port>
                    <user>default</user>
                </replica>
            </shard>
        </cluster_1shard_2replicas>
    </remote_servers>

    <!-- ZooKeeper configuration -->
    <zookeeper>
        <node index="1">
            <host>clickhouse-primary</host>
            <port>2181</port>
        </node>
        <node index="2">
            <host>clickhouse-secondary</host>
            <port>2181</port>
        </node>
    </zookeeper>

    <!-- Macros for cluster configuration -->
    <macros>
        <shard>1</shard>
        <replica>clickhouse-secondary</replica>
    </macros>

    <!-- Enable distributed_ddl -->
    <distributed_ddl>
        <path>/clickhouse/task_queue/ddl</path>
    </distributed_ddl>
</clickhouse>
EOF'

```

Step 5: Configure Network Settings

```

# Create network config on primary node
sudo tee /etc/clickhouse-server/config.d/network.xml > /dev/null <<'EOF'
<?xml version="1.0"?>
<clickhouse>

```

```

<!-- Listen on all interfaces -->
<listen_host>::</listen_host>

<!-- HTTP interface -->
<http_port>8123</http_port>

<!-- TCP interface (use 9001 to avoid Hadoop conflicts) -->
<tcp_port>9001</tcp_port>

<!-- Interserver HTTP for replication -->
<interserver_http_host>clickhouse-primary</interserver_http_host>
<interserver_http_port>9009</interserver_http_port>

<!-- Maximum connections -->
<max_connections>4096</max_connections>

<!-- Keep alive timeout -->
<keep_alive_timeout>3</keep_alive_timeout>
</clickhouse>
EOF

# Create network config on secondary node
ssh clickhouse-secondary 'sudo tee /etc/clickhouse-
server/config.d/network.xml > /dev/null <<'\"'\"'EOF'\"'\"'
<?xml version="1.0"?>
<clickhouse>
  <!-- Listen on all interfaces -->
  <listen_host>::</listen_host>

  <!-- HTTP interface -->
  <http_port>8123</http_port>

  <!-- TCP interface (use 9001 to avoid Hadoop conflicts) -->
  <tcp_port>9001</tcp_port>

  <!-- Interserver HTTP for replication -->
  <interserver_http_host>clickhouse-secondary</interserver_http_host>
  <interserver_http_port>9009</interserver_http_port>

  <!-- Maximum connections -->
  <max_connections>4096</max_connections>

  <!-- Keep alive timeout -->
  <keep_alive_timeout>3</keep_alive_timeout>
</clickhouse>
EOF'

```

Step 6: Start ClickHouse Services

```

# Start ClickHouse on both nodes
sudo systemctl start clickhouse-server

```

```
sudo systemctl enable clickhouse-server

ssh clickhouse-secondary 'sudo systemctl start clickhouse-server'
ssh clickhouse-secondary 'sudo systemctl enable clickhouse-server'

# Check service status
sleep 10
sudo systemctl status clickhouse-server --no-pager -l
ssh clickhouse-secondary 'sudo systemctl status clickhouse-server --no-pager -l'

# Verify ZooKeeper status
sudo systemctl status zookeeper --no-pager -l
ssh clickhouse-secondary 'sudo systemctl status zookeeper --no-pager -l'
```

Step 7: Test Cluster Configuration

```
# Test ClickHouse connectivity
clickhouse-client --port 9001 --query "SELECT version() as clickhouse_version"
clickhouse-client --host clickhouse-secondary --port 9001 --query "SELECT version() as clickhouse_version"

# Check cluster status
clickhouse-client --port 9001 --query "SELECT * FROM system.clusters WHERE cluster LIKE '%cluster%'"

# Test ZooKeeper status
/opt/zookeeper/bin/zkServer.sh status
ssh clickhouse-secondary '/opt/zookeeper/bin/zkServer.sh status'
```

Step 8: Create Test Database and Tables

```
# Create database and test replication
clickhouse-client --port 9001 --query "
CREATE DATABASE test_db ON CLUSTER cluster_1shard_2replicas;
CREATE TABLE test_db.replicated_table ON CLUSTER cluster_1shard_2replicas
(
    id UInt64,
    timestamp DateTime,
    message String
)
ENGINE =
ReplicatedMergeTree('/clickhouse/tables/{shard}/test_db/replicated_table',
'{replica}')
ORDER BY id;
"

# Create distributed table for queries
```

```
clickhouse-client --port 9001 --query "  
CREATE TABLE test_db.distributed_table AS test_db.replicated_table  
ENGINE = Distributed(cluster_2shards_1replicas, test_db, replicated_table,  
rand());  
"  
  
# Insert test data  
clickhouse-client --port 9001 --query "  
INSERT INTO test_db.replicated_table VALUES  
(1, now(), 'Hello from primary'),  
(2, now(), 'Cluster setup successful');  
"  
  
# Verify data replication  
clickhouse-client --port 9001 --query "SELECT * FROM  
test_db.replicated_table ORDER BY id"  
clickhouse-client --host clickhouse-secondary --port 9001 --query "SELECT *  
FROM test_db.replicated_table ORDER BY id"
```

Verification Commands

Check Service Status

```
# ClickHouse services  
sudo systemctl status clickhouse-server  
ssh clickhouse-secondary 'sudo systemctl status clickhouse-server'  
  
# ZooKeeper services  
sudo systemctl status zookeeper  
ssh clickhouse-secondary 'sudo systemctl status zookeeper'  
  
# Process status  
ps aux | grep -E '(clickhouse|zookeeper)'  
ssh clickhouse-secondary 'ps aux | grep -E '(clickhouse|zookeeper)''
```

Check Network Ports

```
# ClickHouse ports  
sudo netstat -tulpn | grep -E '(8123|9001|9009)'  
ssh clickhouse-secondary 'sudo netstat -tulpn | grep -E "(8123|9001|9009)"'  
  
# ZooKeeper ports  
sudo netstat -tulpn | grep -E '(2181|2888|3888)'  
ssh clickhouse-secondary 'sudo netstat -tulpn | grep -E "(2181|2888|3888)"'
```

Test Cluster Operations


```
# Test distributed queries
clickhouse-client --port 9001 --query "
SELECT
    cluster() as cluster_name,
    shardNum() as shard,
    replicaNum() as replica,
    hostName() as hostname
FROM system.one
"

# Test data distribution
clickhouse-client --port 9001 --query "SELECT * FROM
test_db.distributed_table"

# Check replication status
clickhouse-client --port 9001 --query "
SELECT database, table, is_leader, is_readonly, absolute_delay
FROM system.replicas
WHERE database = 'test_db'
"
```

Web Interfaces

- **Primary Node ClickHouse:** <http://clickhouse-primary:8123>
- **Secondary Node ClickHouse:** <http://clickhouse-secondary:8123>

Cluster Management

Start Services

```
# Start ZooKeeper first
sudo systemctl start zookeeper
ssh clickhouse-secondary 'sudo systemctl start zookeeper'

# Then start ClickHouse
sudo systemctl start clickhouse-server
ssh clickhouse-secondary 'sudo systemctl start clickhouse-server'
```

Stop Services

```
# Stop ClickHouse first
sudo systemctl stop clickhouse-server
ssh clickhouse-secondary 'sudo systemctl stop clickhouse-server'

# Then stop ZooKeeper
sudo systemctl stop zookeeper
ssh clickhouse-secondary 'sudo systemctl stop zookeeper'
```

Troubleshooting

Common Issues

1. **ClickHouse fails to start:** Check ZooKeeper is running and accessible
2. **Replication not working:** Verify network connectivity between nodes
3. **Port conflicts:** Ensure ports 8123, 9001, 9009, 2181, 2888, 3888 are available

Log Locations

- **ClickHouse logs:** `/var/log/clickhouse-server/clickhouse-server.log`
- **ZooKeeper logs:** `/opt/zookeeper/logs/zookeeper.log`

Useful Commands

```
# Check ClickHouse logs
sudo tail -f /var/log/clickhouse-server/clickhouse-server.log

# Check ZooKeeper logs
sudo tail -f /opt/zookeeper/logs/zookeeper.log

# Test ZooKeeper connectivity
/opt/zookeeper/bin/zkCli.sh -server localhost:2181

# Restart services if needed
sudo systemctl restart clickhouse-server
sudo systemctl restart zookeeper
```

Success Criteria

✅ Cluster is working when:

- Both ClickHouse services are running
- Both ZooKeeper services are running (one leader, one follower)
- Data replicates between nodes automatically
- Distributed queries return results from both nodes
- Web interfaces are accessible on both nodes