

clickhouse-cluster

- [Some Essential checks](#)
- [Create db and replicated tables](#)
- [New Page](#)

Some Essential checks

Check keeper :

```
echo ruok | nc 127.0.0.1 9181
```

```
sudo systemctl restart clickhouse-keeper
```

Check cluster

```
SELECT * FROM system.clusters WHERE cluster='default';
```

Additional Keeper tests

Cluster monitoring

```
echo mntr | nc 127.0.0.1 9181
```

Should show:

```
zk_server_state    leader    (on one node)
zk_server_state    follower (on other two nodes)
```

Server stats

```
echo stat | nc 127.0.0.1 9181
```

List znodes

```
echo ls / | nc 127.0.0.1 9181
```

Create db and replicated tables

Build Replicated ClickHouse Tables

Keeper is fully working.

Now you can create **replicated database + replicated tables** that sync across all 3 servers.

? Step-by-step instructions (copy/paste)

1. Create database on all nodes:

```
CREATE DATABASE replicated_db;
```

2. Create table with replication

Node1:

```
CREATE TABLE replicated_db.events  
(  
    id UInt64,  
    msg String  
)
```

```
ENGINE = ReplicatedMergeTree('/clickhouse/tables/events', 'replica1')  
ORDER BY id;
```

Node2:

```
CREATE TABLE replicated_db.events
(
  id UInt64,
  msg String
)
ENGINE = ReplicatedMergeTree('/clickhouse/tables/events', 'replica2')
ORDER BY id;
```

Node3:

```
CREATE TABLE replicated_db.events
(
  id UInt64,
  msg String
)
ENGINE = ReplicatedMergeTree('/clickhouse/tables/events', 'replica3')
ORDER BY id;
```

? Test replication:

Insert into Node1:

```
INSERT INTO replicated_db.events VALUES (1, 'hello');
```

Check on Node2:

```
SELECT * FROM replicated_db.events;
```

Check on Node3:

```
SELECT * FROM replicated_db.events;
```

You will see:

```
1 | hello
```

New Page

```
CREATE TABLE my_table ON CLUSTER default
(
  `id` UInt64,
  `name` String,
  `created_at` String
)
ENGINE = MergeTree
ORDER BY id
```

```
INSERT INTO demo.my_table
(id, name, created_at)
VALUES(0, '228', '');
```

```
select * from my_table
```

```
INSERT INTO demo.my_table_all
(id, name, created_at)
VALUES(0, 'X', '');
```

```
CREATE TABLE my_table_all ON CLUSTER default AS my_table
ENGINE = Distributed(default, demo, my_table, rand())
```

```
select * from my_table_all
```