

ClickHouse 集群搭建

AUTHOR: 彭玲 TIME: 2023/7/28

ClickHouse 集群搭建

- 一、集群节点
- 二、ZooKeeper 集群搭建
 - 1. 下载与安装
 - 2. 配置
 - 3. 启动
 - 4. 集群安装验证
- 三、ClickHouse 集群搭建
 - 1. 安装
 - 2. 配置
 - 3. 启动
- 四、验证集群能力
 - 1. 创建数据库
 - node37
 - node38
 - node39
 - 2. 创建复制表
 - node37
 - node38
 - node39
 - 3. 创建分布式表
 - 4. 插入数据
 - 5. 查看表数据
 - 分布式表数据
 - 本地复制表数据

一、集群节点

```
1 | 10.8.30.37 -> node37
2 | 10.8.30.38 -> node38
3 | 10.8.30.39 -> node39
```

二、ZooKeeper 集群搭建

参考官网: <https://zookeeper.apache.org/doc/r3.8.2/zookeeperStarted.html>

在 node37、node38 和 node39 三个节点上分别执行以下操作。

1. 下载与安装

```
1 $ wget https://archive.apache.org/dist/zookeeper/zookeeper-3.8.1/apache-zookeeper-3.8.1-bin.tar.gz
2
3 $ tar -zxvf apache-zookeeper-3.8.1-bin.tar.gz
```

2. 配置

```
1 anxin@node37:~/julin/apache-zookeeper-3.8.1-bin$ cp conf/zoo_sample.cfg conf/zoo.cfg
2
3 anxin@node37:~/julin/apache-zookeeper-3.8.1-bin$ vi conf/zoo.cfg
4
5 # do not use /tmp for storage, /tmp here is just
6 # example sakes.
7 dataDir=/var/lib/zookeeper
8 # the port at which the clients will connect
9 clientPort=2181
10 # the maximum number of client connections.
11 # increase this if you need to handle more clients
12 #maxClientCnxns=60
13 #
14 # Be sure to read the maintenance section of the
15 # administrator guide before turning on autopurge.
16 #
17 # https://zookeeper.apache.org/doc/current/zookeeperAdmin.html#sc_maintenance
18 #
19 # The number of snapshots to retain in dataDir
20 #autopurge.snapRetainCount=3
21 # Purge task interval in hours
22 # Set to "0" to disable auto purge feature
23 #autopurge.purgeInterval=1
24
25 ## Metrics Providers
26 #
27 # https://prometheus.io Metrics Exporter
28 #metricsProvider.className=org.apache.zookeeper.metrics.prometheus.PrometheusMetricsProvider
29 #metricsProvider.httpHost=0.0.0.0
30 #metricsProvider.httpPort=7000
31 #metricsProvider.exportJvmInfo=true
32
33 server.1=10.8.30.37:2888:3888
34 server.2=10.8.30.38:2888:3888
35 server.3=10.8.30.39:2888:3888
```

3. 启动

```
1 | anxin@node37:~/julin/apache-zookeeper-3.8.1-bin$ ./bin/zkServer.sh start
```

4. 集群安装验证

用 `zkServer.sh status` 命令分别查看三个节点的状态：3台机器中，有2台成为了 `follower`，1台成为了 `leader` 节点。

```
1 | anxin@node37:~/julin/apache-zookeeper-3.8.1-bin$ ./bin/zkServer.sh status
2 | ZooKeeper JMX enabled by default
3 | Using config: /home/anxin/julin/apache-zookeeper-3.8.1-bin/bin/../conf/zoo.cfg
4 | Client port found: 2181. Client address: localhost. Client SSL: false.
5 | Mode: follower
6 |
7 | anxin@node38:~/julin/apache-zookeeper-3.8.1-bin$ ./bin/zkServer.sh status
8 | ZooKeeper JMX enabled by default
9 | Using config: /home/anxin/julin/apache-zookeeper-3.8.1-bin/bin/../conf/zoo.cfg
10 | Client port found: 2181. Client address: localhost. Client SSL: false.
11 | Mode: leader
12 |
13 | anxin@node39:~/julin/apache-zookeeper-3.8.1-bin$ ./bin/zkServer.sh status
14 | ZooKeeper JMX enabled by default
15 | Using config: /home/anxin/julin/apache-zookeeper-3.8.1-bin/bin/../conf/zoo.cfg
16 | Client port found: 2181. Client address: localhost. Client SSL: false.
17 | Mode: follower
```

三、ClickHouse 集群搭建

参考官网：<https://clickhouse.com/docs/zh/getting-started/tutorial>

在 `node37`、`node38` 和 `node39` 三个节点上分别执行以下操作。

1. 安装

```
1 | $ sudo apt-get install -y apt-transport-https ca-certificates dirmngr
2 | $ sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 8919F6BD2B48D754
3 |
4 | $ echo "deb https://packages.clickhouse.com/deb stable main" | sudo tee \
5 |   /etc/apt/sources.list.d/clickhouse.list
6 | $ sudo apt-get update
7 |
8 | $ sudo apt-get install -y clickhouse-server clickhouse-client
```

2. 配置

参考：<https://altinity.com/blog/2018/5/10/circular-replication-cluster-topology-in-clickhouse>

ClickHouse 集群配置（`node37`、`node38` 和 `node39` 三个节点配置相同）：

```

1 root@node37:/etc/clickhouse-server# vi config.xml
2
3 <remote_servers>
4   <!-- Test only shard config for testing distributed storage -->
5   <perftest_3shards_2replicas>
6     <shard>
7       <internal_replication>true</internal_replication>
8       <replica>
9         <default_database>testcluster_shard_1</default_database>
10        <host>10.8.30.37</host>
11        <port>9000</port>
12      </replica>
13      <replica>
14        <default_database>testcluster_shard_1</default_database>
15        <host>10.8.30.38</host>
16        <port>9000</port>
17      </replica>
18    </shard>
19    <shard>
20      <internal_replication>true</internal_replication>
21      <replica>
22        <default_database>testcluster_shard_2</default_database>
23        <host>10.8.30.38</host>
24        <port>9000</port>
25      </replica>
26      <replica>
27        <default_database>testcluster_shard_2</default_database>
28        <host>10.8.30.39</host>
29        <port>9000</port>
30      </replica>
31    </shard>
32    <shard>
33      <internal_replication>true</internal_replication>
34      <replica>
35        <default_database>testcluster_shard_3</default_database>
36        <host>10.8.30.39</host>
37        <port>9000</port>
38      </replica>
39      <replica>
40        <default_database>testcluster_shard_3</default_database>
41        <host>10.8.30.37</host>
42        <port>9000</port>
43      </replica>
44    </shard>
45  </perftest_3shards_2replicas>
46 </remote_servers>

```

ZooKeeper 集群配置（node37、node38 和 node39 三个节点配置相同）：

```

1 root@node37:/etc/clickhouse-server# vi config.xml
2
3 <zookeeper>
4   <node>

```

```

5         <host>10.8.30.37</host>
6         <port>2181</port>
7     </node>
8     <node>
9         <host>10.8.30.38</host>
10        <port>2181</port>
11    </node>
12    <node>
13        <host>10.8.30.39</host>
14        <port>2181</port>
15    </node>
16 </zookeeper>

```

3. 启动

```

1 $ sudo service clickhouse-server start    # 启动clickhouse服务
2 $ sudo service clickhouse-server restart  # 重启clickhouse
3 $ sudo service clickhouse-server stop     # 停止clickhouse
4 $ sudo service clickhouse-server status   # 查看clickhouse的状态

```

注意: 状态也要是 active (running) 才可以。

```

1 # node37 Active: active (running)
2 root@node37:/etc/clickhouse-server# service clickhouse-server status
3 • clickhouse-server.service - ClickHouse Server (analytic DBMS for big data)
4   Loaded: loaded (/lib/systemd/system/clickhouse-server.service; enabled; vendor
5   preset: enabled)
6   Active: active (running) since Fri 2023-07-28 16:03:58 CST; 18s ago
7   Main PID: 5184 (clickhouse-serv)
8   Tasks: 299 (limit: 4915)
9   CGroup: /system.slice/clickhouse-server.service
10          └─5165 clickhouse-watchdog --config=/etc/clickhouse-server/config.xml --
11          pid-file=/run/clickhouse-server/clickhouse-server.pid
12          └─5184 /usr/bin/clickhouse-server --config=/etc/clickhouse-
13          server/config.xml --pid-file=/run/clickhouse-server/clickhouse-server.pid
14
15 Jul 28 16:03:57 node37 systemd[1]: Starting ClickHouse Server (analytic DBMS for big
16 data)...
17 Jul 28 16:03:57 node37 clickhouse-server[5165]: Processing configuration file
18 '/etc/clickhouse-server/config.xml'.
19 Jul 28 16:03:57 node37 clickhouse-server[5165]: Logging trace to /var/log/clickhouse-
20 server/clickhouse-server.log
21 Jul 28 16:03:57 node37 clickhouse-server[5165]: Logging errors to
22 /var/log/clickhouse-server/clickhouse-server.err.log
23 Jul 28 16:03:57 node37 systemd[1]: clickhouse-server.service: Supervising process
24 5184 which is not our child. we'll most likely not notice when it exits.
25 Jul 28 16:03:57 node37 clickhouse-server[5165]: Processing configuration file
26 '/etc/clickhouse-server/config.xml'.
27 Jul 28 16:03:57 node37 clickhouse-server[5165]: Saved preprocessed configuration to
28 '/var/lib/clickhouse/preprocessed_configs/config.xml'.
29 Jul 28 16:03:57 node37 clickhouse-server[5165]: Processing configuration file
30 '/etc/clickhouse-server/users.xml'.

```

```
20 Jul 28 16:03:57 node37 clickhouse-server[5165]: Saved preprocessed configuration to
'/var/lib/clickhouse/preprocessed_configs/users.xml'.
21 Jul 28 16:03:58 node37 systemd[1]: Started ClickHouse Server (analytic DBMS for big
data).
22
23 # node38 Active: active (running)
24 root@node38:/etc/clickhouse-server# service clickhouse-server status
25 • clickhouse-server.service - ClickHouse Server (analytic DBMS for big data)
26   Loaded: loaded (/lib/systemd/system/clickhouse-server.service; enabled; vendor
preset: enabled)
27   Active: active (running) since Fri 2023-07-28 16:04:06 CST; 55s ago
28   Main PID: 15711 (clickhouse-serv)
29   Tasks: 300 (limit: 4915)
30   CGroup: /system.slice/clickhouse-server.service
31           └─15694 clickhouse-watchdog --config=/etc/clickhouse-server/config.xml --
pid-file=/run/clickhouse-server/clickhouse-server.pid
32           └─15711 /usr/bin/clickhouse-server --config=/etc/clickhouse-
server/config.xml --pid-file=/run/clickhouse-server/clickhouse-server.pid
33
34 Jul 28 16:04:05 node38 systemd[1]: Starting ClickHouse Server (analytic DBMS for big
data)...
35 Jul 28 16:04:05 node38 clickhouse-server[15694]: Processing configuration file
'/etc/clickhouse-server/config.xml'.
36 Jul 28 16:04:05 node38 clickhouse-server[15694]: Logging trace to
/var/log/clickhouse-server/clickhouse-server.log
37 Jul 28 16:04:05 node38 clickhouse-server[15694]: Logging errors to
/var/log/clickhouse-server/clickhouse-server.err.log
38 Jul 28 16:04:05 node38 systemd[1]: clickhouse-server.service: Supervising process
15711 which is not our child. We'll most likely not notice when it exits.
39 Jul 28 16:04:06 node38 clickhouse-server[15694]: Processing configuration file
'/etc/clickhouse-server/config.xml'.
40 Jul 28 16:04:06 node38 clickhouse-server[15694]: Saved preprocessed configuration to
'/var/lib/clickhouse/preprocessed_configs/config.xml'.
41 Jul 28 16:04:06 node38 clickhouse-server[15694]: Processing configuration file
'/etc/clickhouse-server/users.xml'.
42 Jul 28 16:04:06 node38 clickhouse-server[15694]: Saved preprocessed configuration to
'/var/lib/clickhouse/preprocessed_configs/users.xml'.
43 Jul 28 16:04:06 node38 systemd[1]: Started ClickHouse Server (analytic DBMS for big
data).
44
45 # node39 Active: active (running)
46 root@node39:/etc/clickhouse-server# service clickhouse-server status
47 • clickhouse-server.service - ClickHouse Server (analytic DBMS for big data)
48   Loaded: loaded (/lib/systemd/system/clickhouse-server.service; enabled; vendor
preset: enabled)
49   Active: active (running) since 五 2023-07-28 16:04:09 CST; 1min 10s ago
50   Main PID: 25541 (clickhouse-serv)
51   Tasks: 319
52   Memory: 243.5M
53   CPU: 5.448s
54   CGroup: /system.slice/clickhouse-server.service
55           └─25526 clickhouse-watchdog --config=/etc/clickhouse-
server/config.xml --pid-file=/run/clickhouse-server/clickhouse-server.pid
```

```
56     └─25541 /usr/bin/clickhouse-server --config=/etc/clickhouse-
server/config.xml --pid-file=/run/clickhouse-server/clickhouse-server.pid
57
58 7月 28 16:04:09 node39 systemd[1]: Starting ClickHouse Server (analytic DBMS for big
data)...
59 7月 28 16:04:09 node39 clickhouse-server[25526]: Processing configuration file
'/etc/clickhouse-server/config.xml'.
60 7月 28 16:04:09 node39 clickhouse-server[25526]: Logging trace to
/var/log/clickhouse-server/clickhouse-server.log
61 7月 28 16:04:09 node39 clickhouse-server[25526]: Logging errors to
/var/log/clickhouse-server/clickhouse-server.err.log
62 7月 28 16:04:09 node39 systemd[1]: clickhouse-server.service: Supervising process
25541 which is not our child. We'll most likely not notice when it exits.
63 7月 28 16:04:09 node39 clickhouse-server[25526]: Processing configuration file
'/etc/clickhouse-server/config.xml'.
64 7月 28 16:04:09 node39 clickhouse-server[25526]: Saved preprocessed configuration to
'/var/lib/clickhouse/preprocessed_configs/config.xml'.
65 7月 28 16:04:09 node39 clickhouse-server[25526]: Processing configuration file
'/etc/clickhouse-server/users.xml'.
66 7月 28 16:04:09 node39 clickhouse-server[25526]: Saved preprocessed configuration to
'/var/lib/clickhouse/preprocessed_configs/users.xml'.
67 7月 28 16:04:09 node39 systemd[1]: Started ClickHouse Server (analytic DBMS for big
data).
```

四、验证集群能力

ClickHouse 中创建表与数据，验证集群能力。

执行 `clickhouse-client` 命令进入 CK 数据库。

1. 创建数据库

- node37 节点的数据库
 - testcluster_shard_1
 - testcluster_shard_3
- node38 节点的数据库
 - testcluster_shard_2
 - testcluster_shard_1
- node39 节点的数据库
 - testcluster_shard_3
 - testcluster_shard_2

node37

```
1 root@node37:/etc/clickhouse-server# clickhouse-client
2 ClickHouse client version 23.6.2.18 (official build).
3 Connecting to localhost:9000 as user default.
4 Connected to ClickHouse server version 23.6.2 revision 54464.
5
6 node37 :)
7 node37 :) CREATE DATABASE testcluster_shard_1;
8 node37 :) CREATE DATABASE testcluster_shard_3;
```

node38

```
1 root@node38:/etc/clickhouse-server# clickhouse-client
2 ClickHouse client version 23.6.2.18 (official build).
3 Connecting to localhost:9000 as user default.
4 Connected to ClickHouse server version 23.6.2 revision 54464.
5
6 node38 :)
7 node38 :) CREATE DATABASE testcluster_shard_2;
8 node38 :) CREATE DATABASE testcluster_shard_1;
```

node39

```
1 root@node39:/etc/clickhouse-server# clickhouse-client
2 ClickHouse client version 23.6.2.18 (official build).
3 Connecting to localhost:9000 as user default.
4 Connected to ClickHouse server version 23.6.2 revision 54464.
5
6 node39 :)
7 node39 :) CREATE DATABASE testcluster_shard_3;
8 node39 :) CREATE DATABASE testcluster_shard_2;
```

2. 创建复制表

node37

```
1 node37 :) CREATE TABLE testcluster_shard_1.tc_shard
2 (
3     id Int32,
4     name String,
5     createDate Date
6 )
7 Engine=ReplicatedMergeTree('/clickhouse/tables/tc_shard_1/events', 'replica_1')
8 ORDER BY id
9
10 node37 :) CREATE TABLE testcluster_shard_3.tc_shard
11 (
12     id Int32,
13     name String,
14     createDate Date
15 )
```



```
16 Engine=ReplicatedMergeTree('/clickhouse/tables/tc_shard_3/events', 'replica_2')
17 ORDER BY id
```

node38

```
1 node38 :) CREATE TABLE testcluster_shard_2.tc_shard
2 (
3     id Int32,
4     name String,
5     createDate Date
6 )
7 Engine=ReplicatedMergeTree('/clickhouse/tables/tc_shard_2/events', 'replica_1')
8 ORDER BY id
9
10 node38 :) CREATE TABLE testcluster_shard_1.tc_shard
11 (
12     id Int32,
13     name String,
14     createDate Date
15 )
16 Engine=ReplicatedMergeTree('/clickhouse/tables/tc_shard_1/events', 'replica_2')
17 ORDER BY id
```

node39

```
1 node39 :) CREATE TABLE testcluster_shard_3.tc_shard
2 (
3     id Int32,
4     name String,
5     createDate Date
6 )
7 Engine=ReplicatedMergeTree('/clickhouse/tables/tc_shard_3/events', 'replica_1')
8 ORDER BY id
9
10 node39 :) CREATE TABLE testcluster_shard_2.tc_shard
11 (
12     id Int32,
13     name String,
14     createDate Date
15 )
16 Engine=ReplicatedMergeTree('/clickhouse/tables/tc_shard_2/events', 'replica_2')
17 ORDER BY id
```

3. 创建分布式表

在 node37、node38 和 node39 三个节点上分别执行以下操作。

```

1 CREATE TABLE tc_distributed
2 (
3     id Int32,
4     name String,
5     createDate Date
6 )
7 ENGINE = Distributed('perfctest_3shards_2replicas', '', tc_shard, rand())

```

4. 插入数据

在 `node37`、`node38` 和 `node39` 三个节点上，任意选择节点插入数据到 `tc_distributed` 分布式表中：

```

1 insert into tc_distributed (id, name, createDate) values (1, '张三', '2023-07-28');
2 insert into tc_distributed values (2, '李四', '2023-07-28');
3 insert into tc_distributed values (3, '王五', '2023-07-28');

```

再试试将数据插入到 `tc_shard` 本地复制表中：

```

1 -- 将数据插入到 node37 复制表
2 node37 :) insert into testcluster_shard_1.tc_shard (id, name, createDate) values (4,
3     '赵六', '2023-07-28');
4 -- 将数据插入到 node39 复制表
5 node39 :) insert into testcluster_shard_3.tc_shard values (5, '龙七', '2023-07-28');

```

5. 查看表数据

分布式表数据

三个节点的查询结果相同，且停止其中任意一个节点，均不影响查询结果。

- `node37` 分布式表数据

```

1 node37 :) select * from tc_distributed order by id
2
3 SELECT *
4 FROM tc_distributed
5 ORDER BY id ASC
6
7 Query id: c72d73b1-0b77-42f0-a71c-dcfff51bb8b3
8
9 ┌-id-┴-┬-name-┴-┬-createDate-┴-┐
10 | 1 | 张三 | 2023-07-28 |
11 └──────────────────────────┘
12 ┌-id-┴-┬-name-┴-┬-createDate-┴-┐
13 | 2 | 李四 | 2023-07-28 |
14 └──────────────────────────┘
15 ┌-id-┴-┬-name-┴-┬-createDate-┴-┐
16 | 3 | 王五 | 2023-07-28 |
17 └──────────────────────────┘
18 ┌-id-┴-┬-name-┴-┬-createDate-┴-┐

```

```

19 | 4 | 赵六 | 2023-07-28 |
20 |-----|
21 | id | name | createDate |
22 | 5 | 龙七 | 2023-07-28 |
23 |-----|
24
25 5 rows in set. Elapsed: 0.008 sec.

```

- node38 分布式表数据

```

1 node38 :) select * from tc_distributed order by id
2
3 SELECT *
4 FROM tc_distributed
5 ORDER BY id ASC
6
7 Query id: b9f95af6-5e3b-46e0-8135-bd5870561e67
8
9 | id | name | createDate |
10 | 1 | 张三 | 2023-07-28 |
11 |-----|
12 | id | name | createDate |
13 | 2 | 李四 | 2023-07-28 |
14 |-----|
15 | id | name | createDate |
16 | 3 | 王五 | 2023-07-28 |
17 |-----|
18 | id | name | createDate |
19 | 4 | 赵六 | 2023-07-28 |
20 |-----|
21 | id | name | createDate |
22 | 5 | 龙七 | 2023-07-28 |
23 |-----|
24
25 5 rows in set. Elapsed: 0.037 sec.

```

- node39 分布式表数据

```

1 node39 :) select * from tc_distributed order by id
2
3 SELECT *
4 FROM tc_distributed
5 ORDER BY id ASC
6
7 Query id: 189ef2d1-15c4-410f-aed6-30a13102aa39
8
9 | id | name | createDate |
10 | 1 | 张三 | 2023-07-28 |
11 |-----|
12 | id | name | createDate |
13 | 2 | 李四 | 2023-07-28 |
14 |-----|

```

```

15  ┌-id-┴-┬-name-┴-┬-createDate-┴-┐
16  │ 3 │ 王五 │ 2023-07-28 │
17  └──────────────────────────┘
18  ┌-id-┴-┬-name-┴-┬-createDate-┴-┐
19  │ 4 │ 赵六 │ 2023-07-28 │
20  └──────────────────────────┘
21  ┌-id-┴-┬-name-┴-┬-createDate-┴-┐
22  │ 5 │ 龙七 │ 2023-07-28 │
23  └──────────────────────────┘
24
25  5 rows in set. Elapsed: 0.085 sec.

```

本地复制表数据

三个节点上：相同分片上的数据查询结果是相同的（达到副本的效果）；3个分片上的数据各不相同，组成完整的数据集。

- node37 复制表数据

```

1  node37 :) select * from testcluster_shard_1.tc_shard
2
3  SELECT *
4  FROM testcluster_shard_1.tc_shard
5
6  Query id: 05cff44d-4182-410a-b540-7e1410d146e7
7
8  ┌-id-┴-┬-name-┴-┬-createDate-┴-┐
9  │ 4 │ 赵六 │ 2023-07-28 │
10 └──────────────────────────┘
11
12  1 row in set. Elapsed: 0.003 sec.
13
14  node37 :)
15  node37 :) select * from testcluster_shard_3.tc_shard
16
17  SELECT *
18  FROM testcluster_shard_3.tc_shard
19
20  Query id: c26823de-c4f4-44ce-a1eb-ad170538899d
21
22  ┌-id-┴-┬-name-┴-┬-createDate-┴-┐
23  │ 1 │ 张三 │ 2023-07-28 │
24  │ 3 │ 王五 │ 2023-07-28 │
25  │ 5 │ 龙七 │ 2023-07-28 │
26  └──────────────────────────┘
27
28  3 rows in set. Elapsed: 0.004 sec.

```

- node38 复制表数据

```

1  node38 :) select * from testcluster_shard_1.tc_shard
2

```

```

3 SELECT *
4 FROM testcluster_shard_1.tc_shard
5
6 Query id: a8c7a001-2e2b-416c-b1e3-1ba5ce0eef4f
7
8 ┌-id-┴-name-┴-createDate-┐
9 | 4 | 赵六 | 2023-07-28 |
10 └────────────────────────┘
11
12 1 row in set. Elapsed: 0.003 sec.
13
14 node38 :)
15 node38 :) select * from testcluster_shard_2.tc_shard
16
17 SELECT *
18 FROM testcluster_shard_2.tc_shard
19
20 Query id: 1d8e156c-4ec4-4159-b4a3-26e750eb9f18
21
22 ┌-id-┴-name-┴-createDate-┐
23 | 2 | 李四 | 2023-07-28 |
24 └────────────────────────┘
25
26 1 row in set. Elapsed: 0.003 sec.

```

- node39 复制表数据

```

1 node39 :) select * from testcluster_shard_2.tc_shard
2
3 SELECT *
4 FROM testcluster_shard_2.tc_shard
5
6 Query id: d37e6389-d4f9-4699-9558-22672e2d0485
7
8 ┌-id-┴-name-┴-createDate-┐
9 | 2 | 李四 | 2023-07-28 |
10 └────────────────────────┘
11
12 1 row in set. Elapsed: 0.021 sec.
13
14 node39 :)
15 node39 :) select * from testcluster_shard_3.tc_shard
16
17 SELECT *
18 FROM testcluster_shard_3.tc_shard
19
20 Query id: fb0fb18a-7098-4d63-8e8e-a81bcf806314
21
22 ┌-id-┴-name-┴-createDate-┐
23 | 1 | 张三 | 2023-07-28 |
24 | 3 | 王五 | 2023-07-28 |
25 | 5 | 龙七 | 2023-07-28 |
26 └────────────────────────┘

```

27

28 3 rows in set. Elapsed: 0.004 sec.