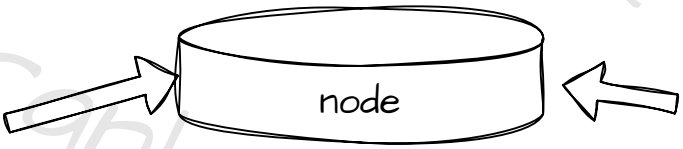
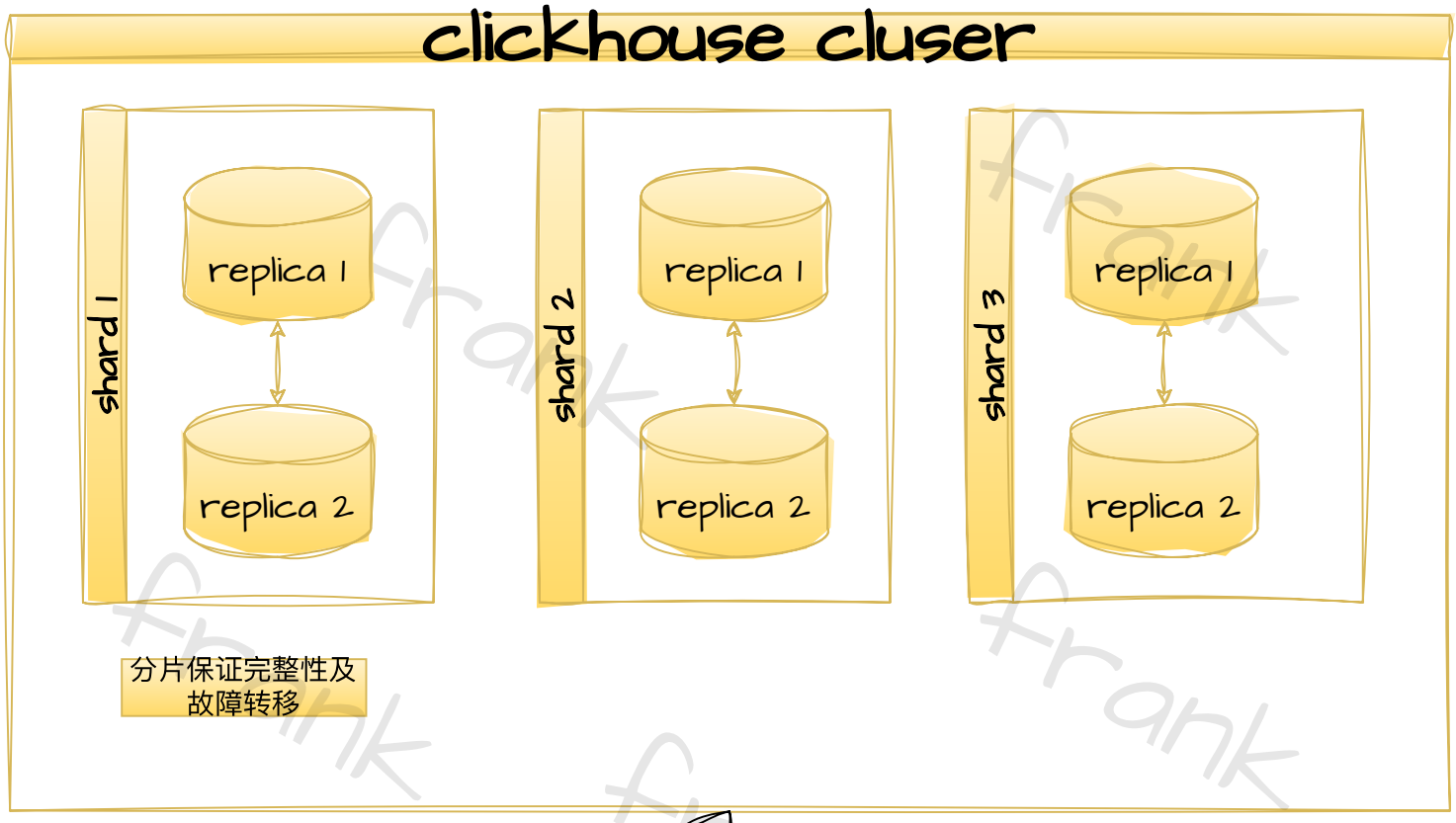
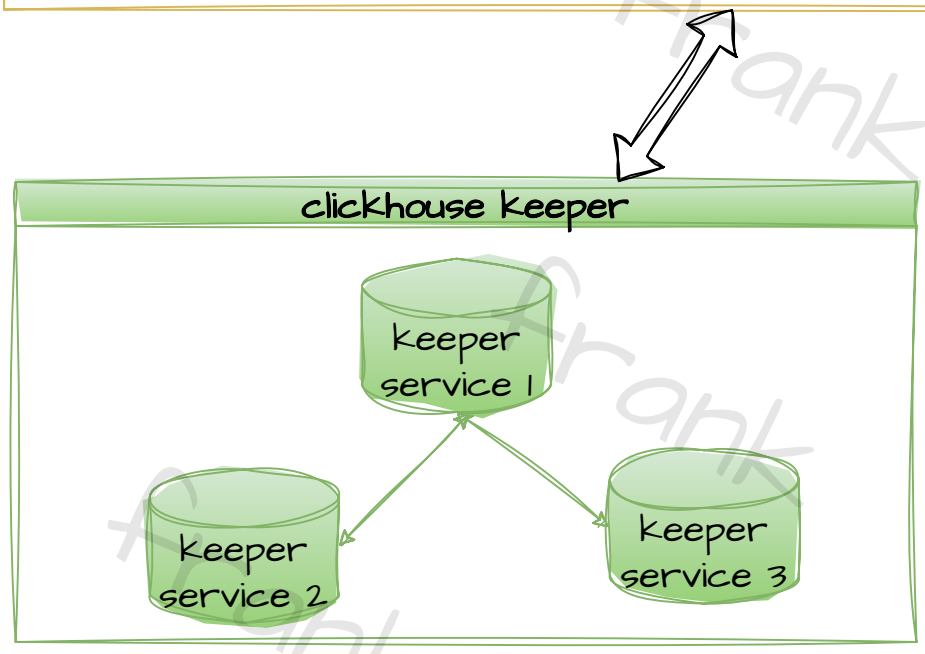
clickhouse

鼓励单机扩展，只有在单机瓶颈出现，无法在线性变化时(容量，io，cpu)，才考虑水平扩展







clickhouse keeper 被用作共识系统，保证集群分片的所有副本以相同顺序执行相同操作。内部仅存储元数据，与replica的复制是独立的

cluset config

```
<remote_servers>
    <eub>
        <shard>
            <internal_replication>true</internal_replication>
            <replica>
                <host>host1</host>
                <port>9000</port>
            </replica>
            <replica>
                <host>host2</host>
                <port>9000</port>
            </replica>
        </shard>
        <shard>
            <internal_replication>true</internal_replication>
            <replica>
                <host>host3</host>
                <port>9000</port>
            </replica>
            <replica>
                <host>host4</host>
                <port>9000</port>
            </replica>
        </shard>
        <shard>
            <internal_replication>true</internal_replication>
            <replica>
                <host>host5</host>
                <port>9000</port>
            </replica>
            <replica>
                <host>host6</host>
                <port>9000</port>
            </replica>
        </shard>
    </eub>
</remote_servers>
```

node macro config

macro 定义的额shard，replica均可在需要创建复制表时手动给到，但是如果出错和不好管理

```
<macros>
   <replica>a</replica>
   <shard>1</shard>
</macros>
```

```
<macros>
   <replica>b</replica>
   <shard>1</shard>
</macros>
```

```
<macros>
   <replica>a</replica>
   <shard>2</shard>
</macros>
```

```
<macros>
   <replica>b</replica>
   <shard>2</shard>
</macros>
```

```
<macros>
   <replica>a</replica>
   <shard>3</shard>
</macros>
```

```
<macros>
   <replica>b</replica>
   <shard>3</shard>
</macros>
```

keeper config

```
<zookeeper>
        <node index="1">
            <host>host7</host>
            <port>9181</port>
        </node>
        <node index="2">
            <host>host8</host>
            <port>9181</port>
        </node>
        <node index="3">
            <host>host9</host>
            <port>9181</port>
        </node>
    </zookeeper>
```

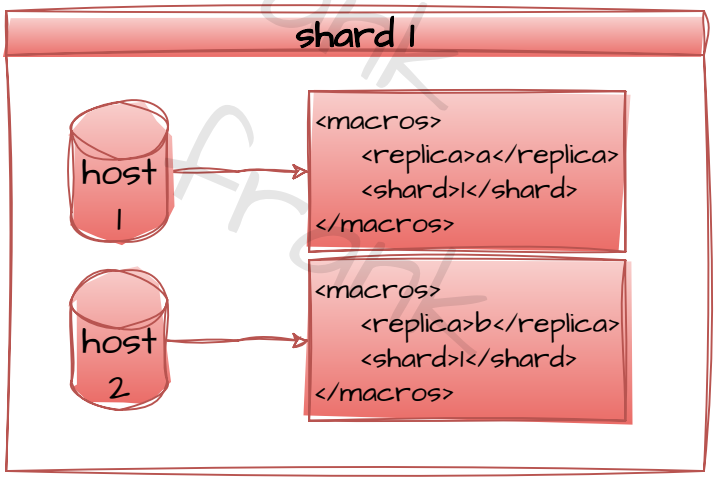
keeper node config

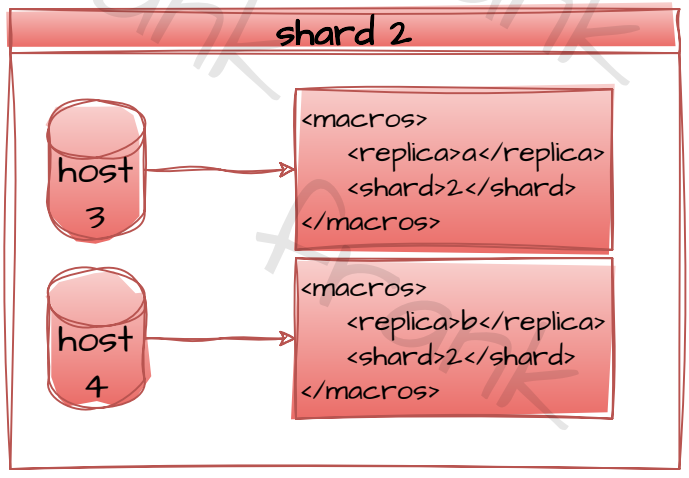
```
<keeper_server>
        <tcp_port>9181</tcp_port>
        <server_id>1</server_id>

        <raft_configuration>
            <server>
                <id>1</id>
                <hostname>host7</hostname>
                <port>9234</port>
            </server>
            <server>
                <id>2</id>
                <hostname>host8</hostname>
                <port>9234</port>
            </server>
            <server>
                <id>3</id>
                <hostname>host9</hostname>
                <port>9234</port>
            </server>
        </raft_configuration>
</keeper_server>
```

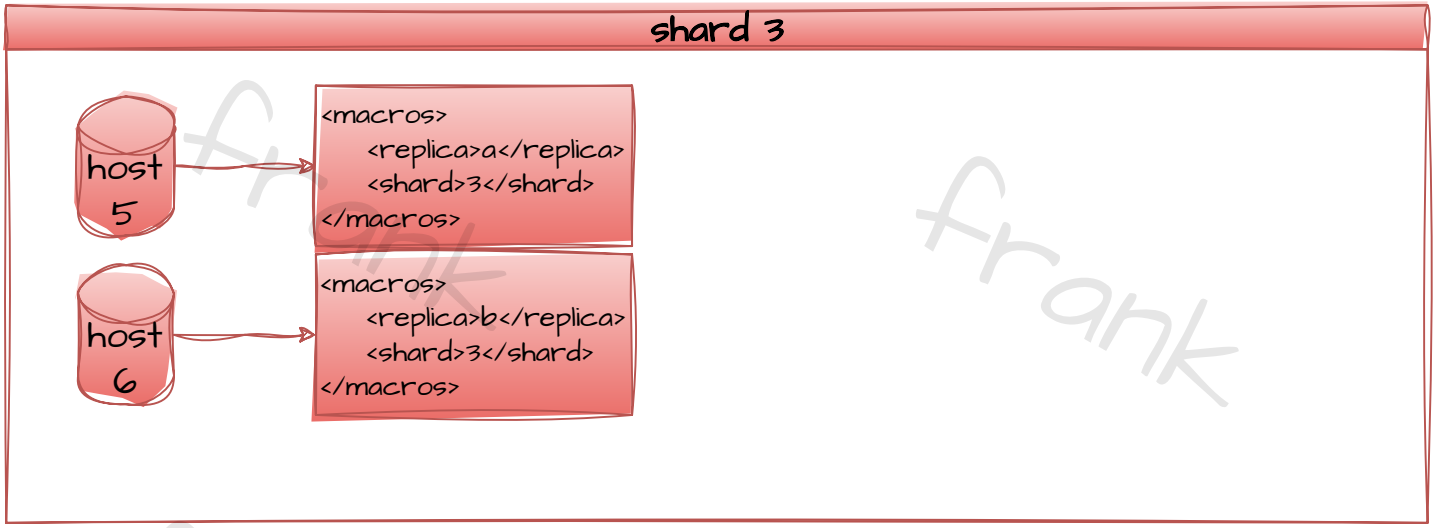
ddl on cluster,为每个节点分别执行ddl

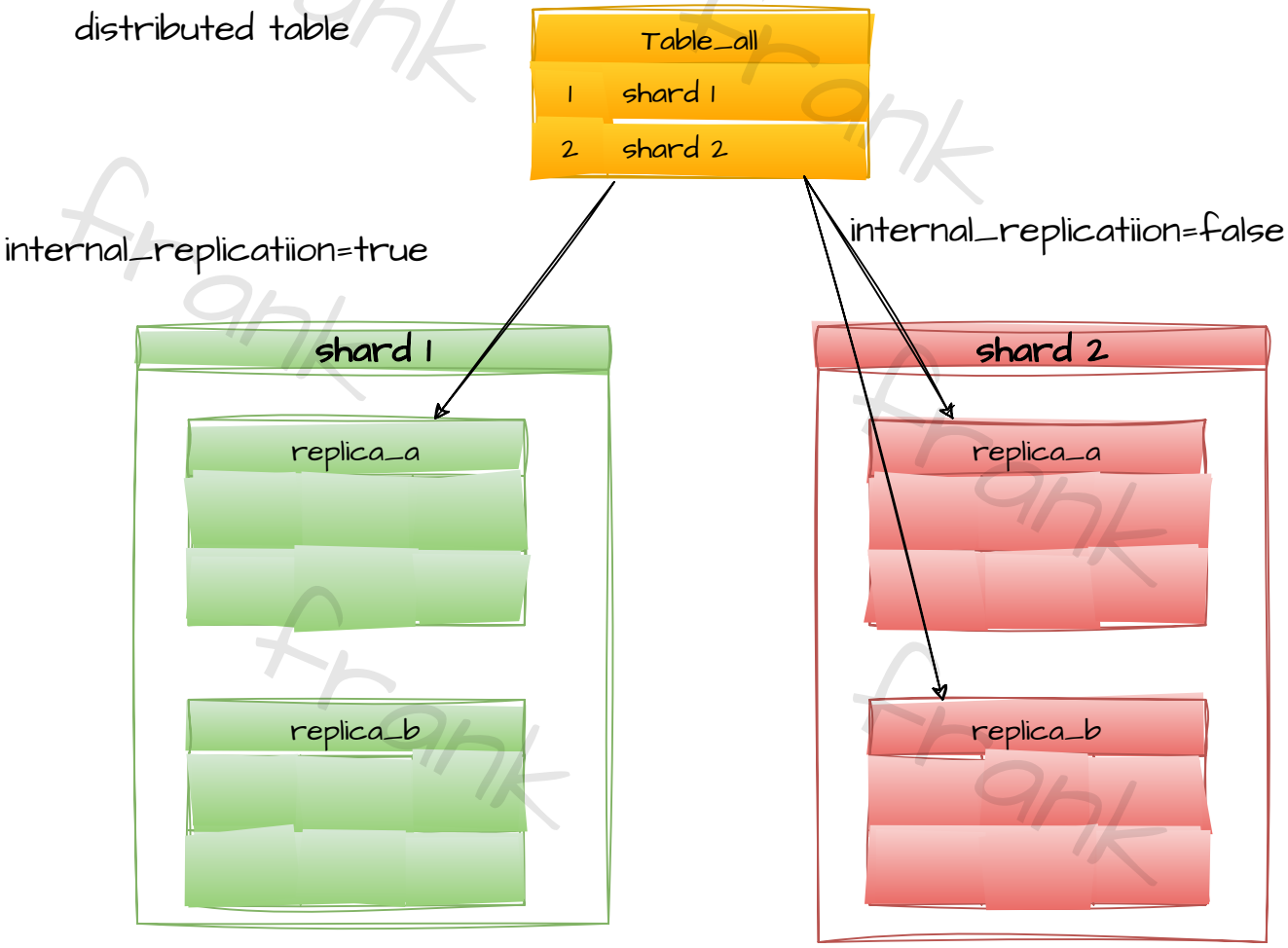
```
create database ods on cluster eub ;

careate table ods.user on cluster eub
(
        openid String ,
        is_fans UInt8,
        source String,
        ctime datetime ,
        utime datetime
)
engine = replicatedMergeTree('clickhouse/tables/user/{shard}', '{replica}')
order by openid,source,utime
primary key openid ;

create table ods.user_all on cluster eub
as ods.test
engine = Distributed(eub, ods, test, cityHash64(openid))
```

replicatedMergeTree('clickhouse/tables/user/{shard}', '{replica}')
shard ,replica 是个服务器上的宏，通过读取conf.xml或者conf.d目录下macro.xml 文件中定义获取

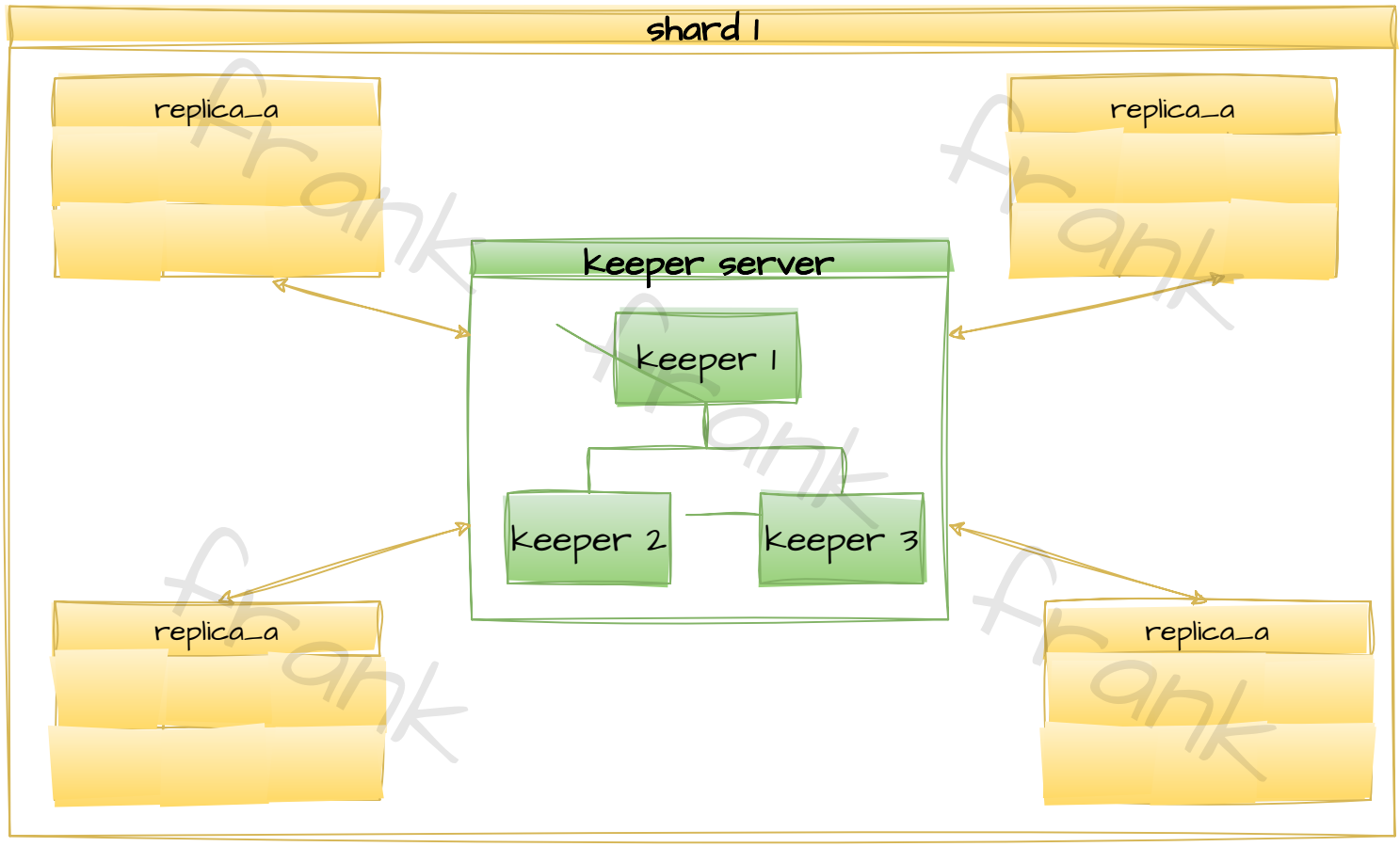








replicatedMergeTree table



multiple cluster config

可以同样4个节点的配置多个集群，以满足不同情况下需求，如example配置，可以创建2分片2副本的分布式表a ，同时创建一份4副本的复制表b，当a join b时可就地匹配，提升速度

```
<remote_servers>
    <eub>
        <shard>
            <internal_replication>true</internal_replication>
            <replica>
                <host>host1</host>
                <port>9000</port>
            </replica>
            <replica>
                <host>host2</host>
                <port>9000</port>
            </replica>
        </shard>
        <shard>
            <internal_replication>true</internal_replication>
            <replica>
                <host>host3</host>
                <port>9000</port>
            </replica>
            <replica>
                <host>host4</host>
                <port>9000</port>
            </replica>
        </shard>
    </eub>
    <eub_one>
        <shard>
            <internal_replication>true</internal_replication>
            <replica>
                <host>host1</host>
                <port>9000</port>
            </replica>
            <replica>
                <host>host2</host>
                <port>9000</port>
            </replica>
            <replica>
                <host>host3</host>
                <port>9000</port>
            </replica>
            <replica>
                <host>host4</host>
                <port>9000</port>
            </replica>
        </shard>
    </eub_one>
</remote_servers>
```

multiple cluster macro config

```
<macros>
    <replica>a</replica>
    <shard_eub>1</shard_eub>
    <shard_eub_one>1</shard_eub_one>
</macros>
```

```
<macros>
    <replica>b</replica>
    <shard_eub>1</shard_eub>
    <shard_eub_one>2</shard_eub_one>
</macros>
```

```
<macros>
    <replica>c</replica>
    <shard_eub>2</shard_eub>
    <shard_eub_one>3</shard_eub_one>
</macros>
```

```
<macros>
    <replica>d</replica>
    <shard_eub>2</shard_eub>
    <shard_eub_one>4</shard_eub_one>
</macros>
```

创建分布式表a，复制表b，分别使用不同的集群名称，分片名称

```
create database ods on cluster eub ;

careate table ods.a on cluster eub
(
      uid String ,
      utime datetime
)
engine = replicatedMergeTree('clickhouse/tables/a/{shard_eub}', '{replica}')
order by uid ;

create table ods.a_all on cluster eub
as ods.test
engine = Distributed(eub, ods, a, cityHash64(uid))

create table ods.b on cluster eub_one
as
ods.a
engine = replicatedMergeTree('clickhouse/tables/b/{shard_eub_one}', '{replica}')
```

cross cluster config

当集群机器不够又需要使用分布式+复制表时，可以有2种方式做到，1 单节点2个clickhouse实例。缺点时搭建复杂，同时2个实例一台机器会有资源竞争的情况。2 单节点单实例，通过2个数据库完成，但建立复制表时候需要手动指定分片及副本，配置麻烦，同时需要default_databases指定，因为创建分布式表需要指定数据库，由于使用了2个数据库的相同表来互为备份，所以创建时不指定数据库，会通过集群配置查找默认数据库来执行

config.xml

```
<remote_servers>
    <cluster_cross>
        <shard>
            <internal_replication>true</internal_replication>
            <replica>
                <default_databases>default</default_databases>
                <host>host1</host>
                <port>9000</port>
            </replica>
            <replica>
                <default_databases>defcross</default_databases>
                <host>host2</host>
                <port>9000</port>
            </replica>
        </shard>
        <shard>
            <internal_replication>true</internal_replication>
            <replica>
                <default_databases>default</default_databases>
                <host>host2</host>
                <port>9000</port>
            </replica>
            <replica>
                <default_databases>defcross</default_databases>
                <host>host1</host>
                <port>9000</port>
            </replica>
        </shard>
    </cluster_cross>
</remote_servers>
```

ddl sql

```
create table default.test{...}engine = replicatedMergeTree('clickhouse/tables/1/', 'a') ;

create table defcross.test{...}engine = replicatedMergeTree('clickhouse/tables/1/', 'b')

create table default.test{...}engine = replicatedMergeTree('clickhouse/tables/2/', 'a')

create table defcross.test{...}engine = replicatedMergeTree('clickhouse/tables/2/', 'b')


2个节点分别创建一次，如果只创建一个，则只有这个创建的节点可以访问分布式表test_all
create table test_all
{...}
engine = Distributed('cluster_cross', '', test)
```