

ClickHouse

<https://clickhouse.com/docs/en/quick-start/>

```
curl https://clickhouse.com/ | sh
sudo ./clickhouse install
sudo clickhouse start
```

访问 <http://10.8.30.38:8123/play> 默认用户default

物化视图同步postgresql

<https://clickhouse.com/docs/en/engines/database-engines/materialized-postgresql/>

需要提前将postgres数据中wal日志级别调制logical以上

```
logical decoding requires wal_level >= logical
```

重启postgresql数据库: service postgres restart || sudo /etc/init.d/postgresql restart

确保pg_hba.conf

```
local replication all peer
host replication all 127.0.0.1/32 scram-sha-256
host replication all ::1/128 scram-sha-256
```

在node36上 (PG9.6做如下设置) :

```
max_wal_senders = 20      # max number of walsender processes
# (change requires restart)
#wal_keep_segments = 0    # in logfile segments, 16MB each; 0 disables
#wal_sender_timeout = 60s # in milliseconds; 0 disables

max_replication_slots = 20 # max number of replication slots
```

也无法创建slot, 报错

```
relation "pg_publication" does not exist
```

确保使用的用户有REPLICATION权限

```
ALTER USER "FashionAdmin" REPLICATION;
```

```
-- sql要一起选中执行
SET allow_experimental_database_materialized_postgresql=1;

-- DROP database anxinyun;
CREATE DATABASE IF NOT EXISTS anxinyun
```

```
ENGINE = MaterializedPostgreSQL('10.8.30.75', 'Anxinyun0810', 'postgres',
'postgres') ;
```

-- 上面的写法，无法实现新增数据同步。加上表名列表后可以

```
CREATE DATABASE IF NOT EXISTS anxinyun
ENGINE = MaterializedPostgreSQL('10.8.30.75', 'Anxinyun0810', 'postgres',
'postgres');
SETTINGS materialized_postgresql_max_block_size = 65536,
          materialized_postgresql_tables_list = 't_user_token,api_log';
```

-- 上面的写法必须列举表名比较麻烦，可以使用如下方式

```
SET allow_experimental_database_materialized_postgresql=1;
CREATE DATABASE IF NOT EXISTS pepca8
ENGINE = MaterializedPostgreSQL('10.8.30.75', 'emis0630', 'postgres',
'postgres')
SETTINGS materialized_postgresql_max_block_size = 65536,
          materialized_postgresql_schema = 'public';
```

p.s.

查询库中所有表名

```
select string_agg(b.relname,',') from (
    select cls.relname
from pg_class cls
join pg_roles rol on rol.oid = cls.relowner
join pg_namespace nsp on nsp.oid = cls.relnamespace
where nsp.nspname not in ('information_schema', 'pg_catalog')
and cls.relkind='r'
and rol.rolname = current_user --- remove this if you want to see all
objects
order by nsp.nspname, cls.relname
) as b;
```

注意事项:

```
-- 要确保同步的表必须包含主键或唯一索引
-- 查询没有主键的表
select tab.table_schema,
       tab.table_name
from information_schema.tables tab
left join information_schema.table_constraints tco
      on tab.table_schema = tco.table_schema
      and tab.table_name = tco.table_name
      and tco.constraint_type = 'PRIMARY KEY'
where tab.table_type = 'BASE TABLE'
      and tab.table_schema not in ('pg_catalog', 'information_schema')
      and tco.constraint_name is null
order by table_schema,
       table_name;

-- 没有主键的创建主键，无法单列进行主键的，使用唯一索引做REPLICA IDENTITY
alter table t_project_type
```

```

        add constraint t_project_type_pk
            primary key (id);
alter table t_temp_modify
    add constraint t_temp_modify_pk
        primary key (id);
alter table t_weather_history
    add constraint t_weather_history_pk
        primary key (id);

create unique index t_units_name_dimension_uindex
    on t_units (name, dimension);
ALTER TABLE t_units REPLICA IDENTITY USING INDEX t_units_name_dimension_uindex;

```

clickhouse play中执行：

https://clickhouse01.anxinyun.cn?allow_experimental_database_materialized_postgresql=1

官网翻译：

必备条件

- 在postgresql配置文件中将`wal_level`设置为`logical`，将`max_replication_slots`设置为`2`。
- 每个复制表必须具有以下一个`replica identity`:(副本唯一键)

1. **default** (主键)
2. **index**

```

postgres# CREATE TABLE postgres_table (a Integer NOT NULL, b Integer, c Integer
NOT NULL, d Integer, e Integer NOT NULL);
postgres# CREATE unique INDEX postgres_table_index on postgres_table(a, c, e);
postgres# ALTER TABLE postgres_table REPLICA IDENTITY USING INDEX
postgres_table_index;

SELECT CASE relreplident
    WHEN 'd' THEN 'default'
    WHEN 'n' THEN 'nothing'
    WHEN 'f' THEN 'full'
    WHEN 'i' THEN 'index'
END AS replica_identity
FROM pg_class
WHERE oid = 't_sensor'::regclass;

```

动态添加表到副本 (replication) :

```
ATTACH TABLE postgres_database.new_table;
```

注意：小于22.1的版本需要手动删除pg中的复制槽

动态删除表：

```
DETACH TABLE postgres_database.table_to_remove;
```

配置项：

- materialized_postgresql_tables_list
选取的表名列表
- materialized_postgresql_schema
选取的schema
- materialized_postgresql_schema_list
选取的schema列表
- materialized_postgresql_allow_automatic_update
>=22.1 自动加载表的变更。默认是'0'
- materialized_postgresql_max_block_size
Flush之前内存中最大加载的行。默认'65535'
- materialized_postgresql_replication_slot
用户创建的slot
- materialized_postgresql_snapshot
必须和materialized_postgresql_replication_slot一并使用，指定从指定快照位置复制。

博文推荐: <https://www.cnblogs.com/traditional/tag/ClickHouse%EF%BC%9A%E4%B8%80%E6%AC%BE%E9%80%9F%E5%BA%A6%E5%BF%AB%E5%88%B0%E8%AE%A9%E4%BA%BA%E5%8F%91%E6%8C%87%E7%9A%84%E5%88%97%E5%BC%8F%E5%AD%98%E5%82%A8%E6%95%B0%E6%8D%AE%E5%BA%93/>