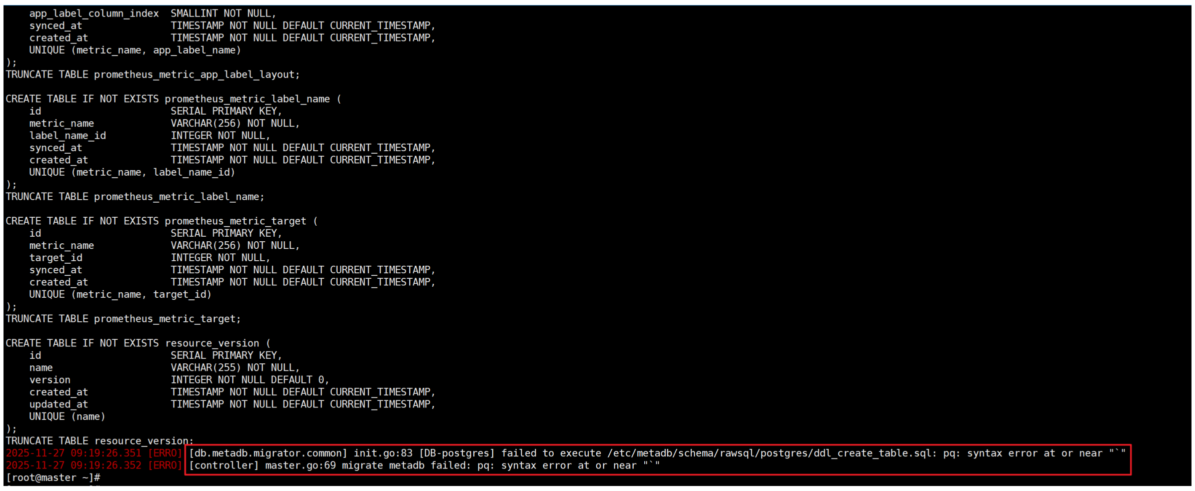
deepflow v7.0 使用 postgres 数据库遇到的问题以及改造

1. ` `上引号不能用

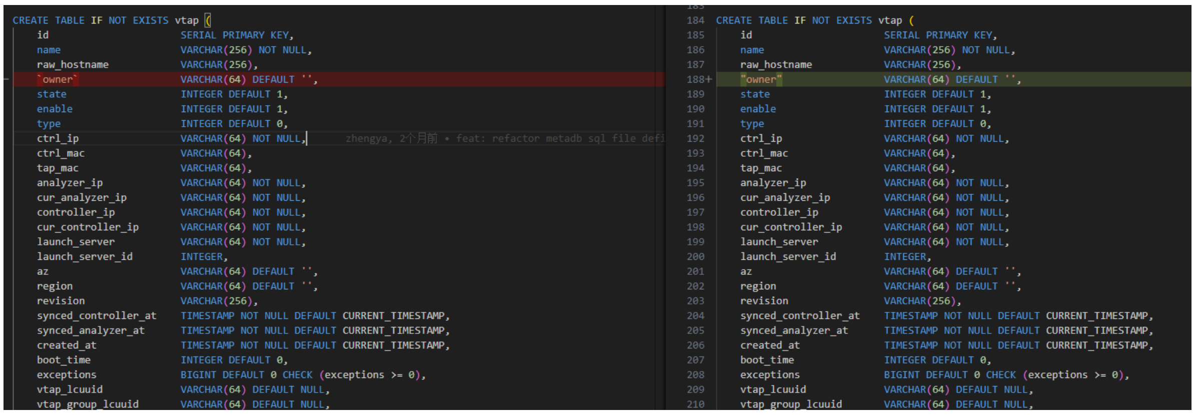
1.1. 报错



```
    app_label_column_index  SMALLINT NOT NULL,
    synced_at               TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    created_at              TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    UNIQUE (metric_name, app_label_name)
);
TRUNCATE TABLE prometheus_metric_app_label_layout;

CREATE TABLE IF NOT EXISTS prometheus_metric_label_name (
    id                      SERIAL PRIMARY KEY,
    metric_name             VARCHAR(256) NOT NULL,
    label_name_id           INTEGER NOT NULL,
    synced_at               TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    created_at              TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    UNIQUE (metric_name, label_name_id)
);
TRUNCATE TABLE prometheus_metric_label_name;

CREATE TABLE IF NOT EXISTS prometheus_metric_target (
    id                      SERIAL PRIMARY KEY,
    metric_name             VARCHAR(256) NOT NULL,
    target_id               INTEGER NOT NULL,
    synced_at               TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    created_at              TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    UNIQUE (metric_name, target_id)
);
TRUNCATE TABLE prometheus_metric_target;

CREATE TABLE IF NOT EXISTS resource_version (
    id                      SERIAL PRIMARY KEY,
    name                    VARCHAR(255) NOT NULL,
    version                 INTEGER NOT NULL DEFAULT 0,
    created_at              TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    updated_at              TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    UNIQUE (name)
);
TRUNCATE TABLE resource_version;
2025-11-27 09:19:26.351 [ERRO] [db.metadb.migrator.common] init.go:83 [DB-postgres] failed to execute /etc/metadb/schema/rawsql/postgres/ddl_create_table.sql: pq: syntax error at or near "`"
2025-11-27 09:19:26.352 [ERRO] [controller] master.go:69 migrate metadb failed: pq: syntax error at or near "`"
[root@master ~]#
```

1.2. 改造

postgres 中使用双引号""转义而不是`上引号，把`上引号改成双引号""

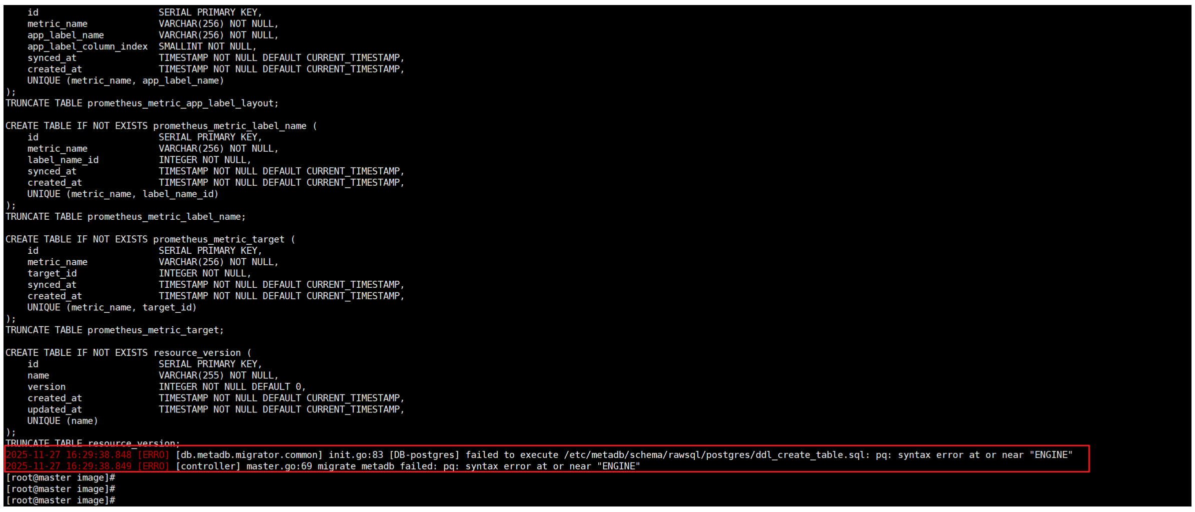


```
CREATE TABLE IF NOT EXISTS vtap (
    id                      SERIAL PRIMARY KEY,
    name                    VARCHAR(256) NOT NULL,
    raw_hostname            VARCHAR(256),
    `owner`                 VARCHAR(64) DEFAULT '',
    state                   INTEGER DEFAULT 1,
    enable                  INTEGER DEFAULT 1,
    type                    INTEGER DEFAULT 0,
    ctrl_ip                 VARCHAR(64) NOT NULL,       zhengya, 2个月前 • feat: refactor metadb sql file defi
    ctrl_mac                VARCHAR(64),
    tap_mac                 VARCHAR(64),
    analyzer_ip             VARCHAR(64) NOT NULL,
    cur_analyzer_ip         VARCHAR(64) NOT NULL,
    controller_ip           VARCHAR(64) NOT NULL,
    cur_controller_ip       VARCHAR(64) NOT NULL,
    launch_server           VARCHAR(64) NOT NULL,
    launch_server_id        INTEGER,
    az                      VARCHAR(64) DEFAULT '',
    region                  VARCHAR(64) DEFAULT '',
    revision                VARCHAR(256),
    synced_controller_at    TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    synced_analyzer_at      TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    created_at              TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    boot_time               INTEGER DEFAULT 0,
    exceptions              BIGINT DEFAULT 0 CHECK (exceptions >= 0),
    vtap_lcuuid             VARCHAR(64) DEFAULT NULL,
    vtap_group_lcuuid       VARCHAR(64) DEFAULT NULL,
```

```
CREATE TABLE IF NOT EXISTS vtap (
    id                      SERIAL PRIMARY KEY,
    name                    VARCHAR(256) NOT NULL,
    raw_hostname            VARCHAR(256),
+   "owner"                 VARCHAR(64) DEFAULT '',
    state                   INTEGER DEFAULT 1,
    enable                  INTEGER DEFAULT 1,
    type                    INTEGER DEFAULT 0,
    ctrl_ip                 VARCHAR(64) NOT NULL,
    ctrl_mac                VARCHAR(64),
    tap_mac                 VARCHAR(64),
    analyzer_ip             VARCHAR(64) NOT NULL,
    cur_analyzer_ip         VARCHAR(64) NOT NULL,
    controller_ip           VARCHAR(64) NOT NULL,
    cur_controller_ip       VARCHAR(64) NOT NULL,
    launch_server           VARCHAR(64) NOT NULL,
    launch_server_id        INTEGER,
    az                      VARCHAR(64) DEFAULT '',
    region                  VARCHAR(64) DEFAULT '',
    revision                VARCHAR(256),
    synced_controller_at    TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    synced_analyzer_at      TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    created_at              TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    boot_time               INTEGER DEFAULT 0,
    exceptions              BIGINT DEFAULT 0 CHECK (exceptions >= 0),
    vtap_lcuuid             VARCHAR(64) DEFAULT NULL,
    vtap_group_lcuuid       VARCHAR(64) DEFAULT NULL,
```

2. mysql 表声明需要去掉

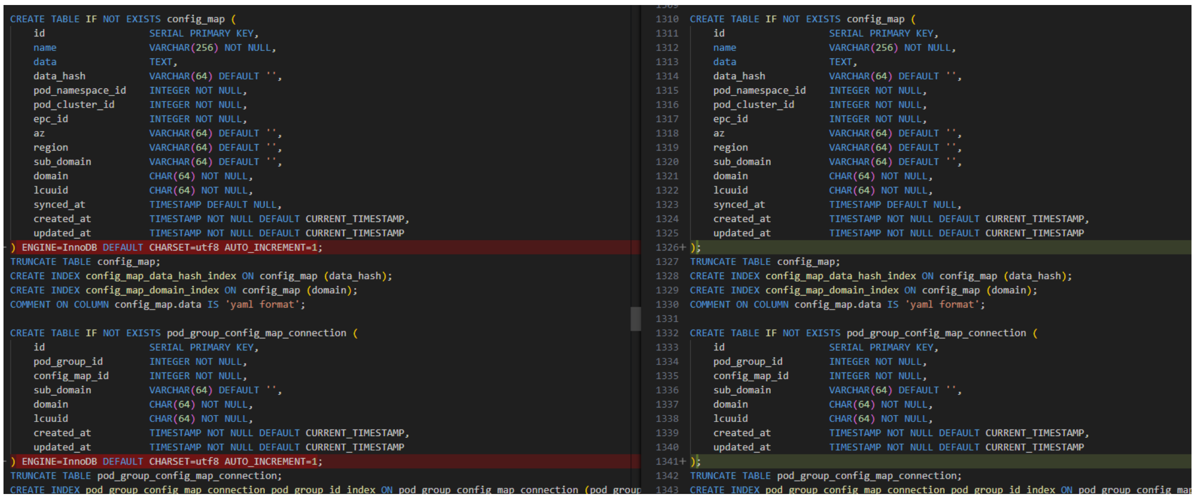
2.1. 报错



```
    id                     SERIAL PRIMARY KEY,
    metric_name            VARCHAR(256) NOT NULL,
    app_label_name         VARCHAR(256) NOT NULL,
    app_label_column_index  SMALLINT NOT NULL,
    synced_at              TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    created_at             TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    UNIQUE (metric_name, app_label_name)
);
TRUNCATE TABLE prometheus_metric_app_label_layout;

CREATE TABLE IF NOT EXISTS prometheus_metric_label_name (
    id                     SERIAL PRIMARY KEY,
    metric_name            VARCHAR(256) NOT NULL,
    label_name_id          INTEGER NOT NULL,
    synced_at              TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    created_at             TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    UNIQUE (metric_name, label_name_id)
);
TRUNCATE TABLE prometheus_metric_label_name;

CREATE TABLE IF NOT EXISTS prometheus_metric_target (
    id                     SERIAL PRIMARY KEY,
    metric_name            VARCHAR(256) NOT NULL,
    target_id              INTEGER NOT NULL,
    synced_at              TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    created_at             TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    UNIQUE (metric_name, target_id)
);
TRUNCATE TABLE prometheus_metric_target;

CREATE TABLE IF NOT EXISTS resource_version (
    id                     SERIAL PRIMARY KEY,
    name                   VARCHAR(255) NOT NULL,
    version                INTEGER NOT NULL DEFAULT 0,
    created_at             TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    updated_at             TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    UNIQUE (name)
);
TRUNCATE TABLE resource_version;
2025-11-27 16:29:38.848 [ERRO] [db.metadb.migrator.common] init.go:83 [DB-postgres] failed to execute /etc/metadb/schema/rawsql/postgres/ddl_create_table.sql: pq: syntax error at or near "ENGINE"
2025-11-27 16:29:38.849 [ERRO] [controller] master.go:69 migrate metadb failed: pq: syntax error at or near "ENGINE"
[root@master image]#
[root@master image]#
[root@master image]#
```

2.2. 改造

ENGINE=innodb DEFAULT CHARSET=utf8 表声明需去掉，postgres 不支持

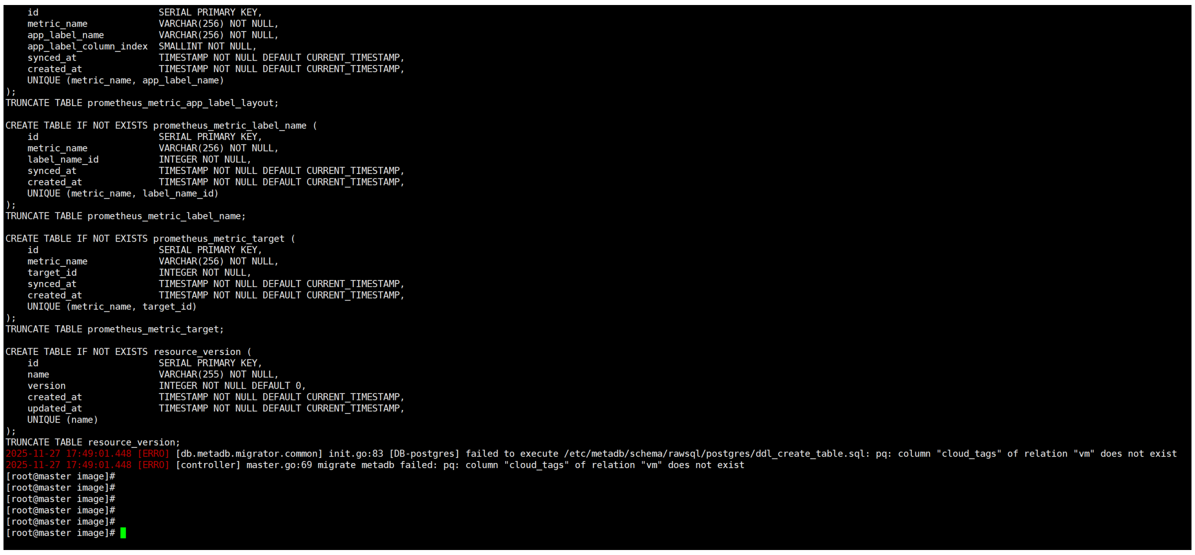


```
CREATE TABLE IF NOT EXISTS config_map (
    id                  SERIAL PRIMARY KEY,
    name                VARCHAR(256) NOT NULL,
    data                TEXT,
    data_hash           VARCHAR(64) DEFAULT '',
    pod_namespace_id    INTEGER NOT NULL,
    pod_cluster_id      INTEGER NOT NULL,
    epc_id              INTEGER NOT NULL,
    az                  VARCHAR(64) DEFAULT '',
    region              VARCHAR(64) DEFAULT '',
    sub_domain          VARCHAR(64) DEFAULT '',
    domain              CHAR(64) NOT NULL,
    lcuuid              CHAR(64) NOT NULL,
    synced_at           TIMESTAMP DEFAULT NULL,
    created_at          TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    updated_at          TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1;
TRUNCATE TABLE config_map;
CREATE INDEX config_map_data_hash_index ON config_map (data_hash);
CREATE INDEX config_map_domain_index ON config_map (domain);
COMMENT ON COLUMN config_map.data IS 'yaml format';

CREATE TABLE IF NOT EXISTS pod_group_config_map_connection (
    id                  SERIAL PRIMARY KEY,
    pod_group_id        INTEGER NOT NULL,
    config_map_id       INTEGER NOT NULL,
    sub_domain          VARCHAR(64) DEFAULT '',
    domain              CHAR(64) NOT NULL,
    lcuuid              CHAR(64) NOT NULL,
    created_at          TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    updated_at          TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1;
TRUNCATE TABLE pod_group_config_map_connection;
CREATE INDEX pod_group_config_map_connection_pod_group_id_index ON pod_group_config_map_connection (pod_group
```

```
CREATE TABLE IF NOT EXISTS config_map (
    id                  SERIAL PRIMARY KEY,
    name                VARCHAR(256) NOT NULL,
    data                TEXT,
    data_hash           VARCHAR(64) DEFAULT '',
    pod_namespace_id    INTEGER NOT NULL,
    pod_cluster_id      INTEGER NOT NULL,
    epc_id              INTEGER NOT NULL,
    az                  VARCHAR(64) DEFAULT '',
    region              VARCHAR(64) DEFAULT '',
    sub_domain          VARCHAR(64) DEFAULT '',
    domain              CHAR(64) NOT NULL,
    lcuuid              CHAR(64) NOT NULL,
    synced_at           TIMESTAMP DEFAULT NULL,
    created_at          TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    updated_at          TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
);
TRUNCATE TABLE config_map;
CREATE INDEX config_map_data_hash_index ON config_map (data_hash);
CREATE INDEX config_map_domain_index ON config_map (domain);
COMMENT ON COLUMN config_map.data IS 'yaml format';

CREATE TABLE IF NOT EXISTS pod_group_config_map_connection (
    id                  SERIAL PRIMARY KEY,
    pod_group_id        INTEGER NOT NULL,
    config_map_id       INTEGER NOT NULL,
    sub_domain          VARCHAR(64) DEFAULT '',
    domain              CHAR(64) NOT NULL,
    lcuuid              CHAR(64) NOT NULL,
    created_at          TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    updated_at          TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
);
TRUNCATE TABLE pod_group_config_map_connection;
CREATE INDEX pod_group_config_map_connection_pod_group_id_index ON pod_group_config_ma
```

3. vm 表中 cloud_tags字段不存在，注释需删除

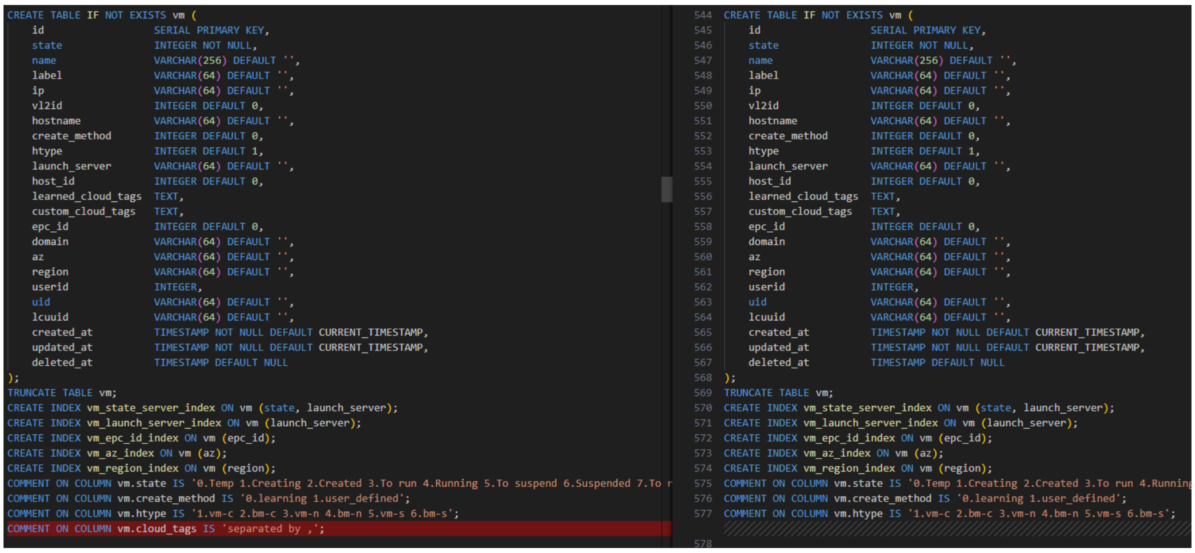
3.1. 报错



```
    id                     SERIAL PRIMARY KEY,
    metric_name            VARCHAR(256) NOT NULL,
    app_label_name         VARCHAR(256) NOT NULL,
    app_label_column_index  SMALLINT NOT NULL,
    synced_at              TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    created_at             TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    UNIQUE (metric_name, app_label_name)
);
TRUNCATE TABLE prometheus_metric_app_label_layout;

CREATE TABLE IF NOT EXISTS prometheus_metric_label_name (
    id                     SERIAL PRIMARY KEY,
    metric_name            VARCHAR(256) NOT NULL,
    label_name_id          INTEGER NOT NULL,
    synced_at              TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    created_at             TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    UNIQUE (metric_name, label_name_id)
);
TRUNCATE TABLE prometheus_metric_label_name;

CREATE TABLE IF NOT EXISTS prometheus_metric_target (
    id                     SERIAL PRIMARY KEY,
    metric_name            VARCHAR(256) NOT NULL,
    target_id              INTEGER NOT NULL,
    synced_at              TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    created_at             TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    UNIQUE (metric_name, target_id)
);
TRUNCATE TABLE prometheus_metric_target;

CREATE TABLE IF NOT EXISTS resource_version (
    id                     SERIAL PRIMARY KEY,
    name                   VARCHAR(255) NOT NULL,
    version                INTEGER NOT NULL DEFAULT 0,
    created_at             TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    updated_at             TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    UNIQUE (name)
);
TRUNCATE TABLE resource_version;
2025-11-27 17:49:01.448 [ERRO] [db.metadb.migrator.common] init.go:83 [DB-postgres] failed to execute /etc/metadb/schema/rawsql/postgres/ddl_create_table.sql: pq: column "cloud_tags" of relation "vm" does not exist
2025-11-27 17:49:01.448 [ERRO] [controller] master.go:69 migrate metadb failed: pq: column "cloud_tags" of relation "vm" does not exist
[root@master image]#
[root@master image]#
[root@master image]#
[root@master image]#
[root@master image]#
[root@master image]#
```

3.2. 改造

cloud_tags字段不存在， COMMENT ON COLUMN vm.cloud_tags IS 'separated by ,'; 该注释需删除

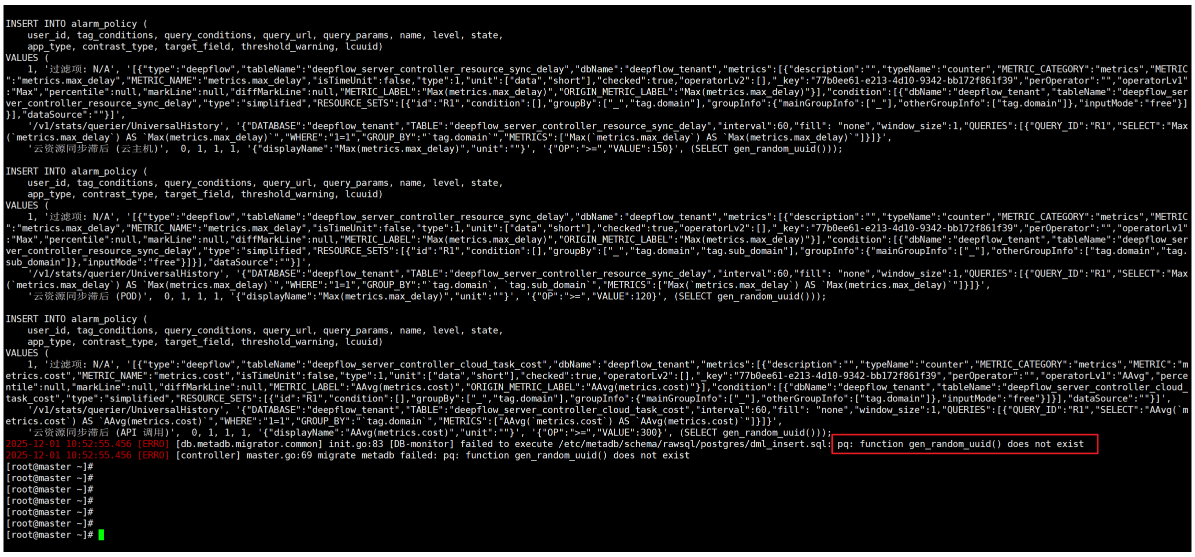


```
CREATE TABLE IF NOT EXISTS vm (
    id                  SERIAL PRIMARY KEY,
    state               INTEGER NOT NULL,
    name                VARCHAR(256) DEFAULT '',
    label               VARCHAR(64) DEFAULT '',
    ip                  VARCHAR(64) DEFAULT '',
    vl2id               INTEGER DEFAULT 0,
    hostname            VARCHAR(64) DEFAULT '',
    create_method       INTEGER DEFAULT 0,
    htype               INTEGER DEFAULT 1,
    launch_server       VARCHAR(64) DEFAULT '',
    host_id             INTEGER DEFAULT 0,
    learned_cloud_tags  TEXT,
    custom_cloud_tags   TEXT,
    epc_id              INTEGER DEFAULT 0,
    domain              VARCHAR(64) DEFAULT '',
    az                  VARCHAR(64) DEFAULT '',
    region              VARCHAR(64) DEFAULT '',
    userid              INTEGER,
    uid                 VARCHAR(64) DEFAULT '',
    lcuuid              VARCHAR(64) DEFAULT '',
    created_at          TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    updated_at          TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    deleted_at          TIMESTAMP DEFAULT NULL
);
TRUNCATE TABLE vm;
CREATE INDEX vm_state_server_index ON vm (state, launch_server);
CREATE INDEX vm_launch_server_index ON vm (launch_server);
CREATE INDEX vm_epc_id_index ON vm (epc_id);
CREATE INDEX vm_az_index ON vm (az);
CREATE INDEX vm_region_index ON vm (region);
COMMENT ON COLUMN vm.state IS '0.Temp 1.Creating 2.Created 3.To run 4.Running 5.To suspend 6.Suspended 7.To r
COMMENT ON COLUMN vm.create_method IS '0.learning 1.user_defined';
COMMENT ON COLUMN vm.htype IS '1.vm-c 2.bm-c 3.vm-n 4.bm-n 5.vm-s 6.bm-s';
COMMENT ON COLUMN vm.cloud_tags IS 'separated by ,';
```

```
CREATE TABLE IF NOT EXISTS vm (
    id                  SERIAL PRIMARY KEY,
    state               INTEGER NOT NULL,
    name                VARCHAR(256) DEFAULT '',
    label               VARCHAR(64) DEFAULT '',
    ip                  VARCHAR(64) DEFAULT '',
    vl2id               INTEGER DEFAULT 0,
    hostname            VARCHAR(64) DEFAULT '',
    create_method       INTEGER DEFAULT 0,
    htype               INTEGER DEFAULT 1,
    launch_server       VARCHAR(64) DEFAULT '',
    host_id             INTEGER DEFAULT 0,
    learned_cloud_tags  TEXT,
    custom_cloud_tags   TEXT,
    epc_id              INTEGER DEFAULT 0,
    domain              VARCHAR(64) DEFAULT '',
    az                  VARCHAR(64) DEFAULT '',
    region              VARCHAR(64) DEFAULT '',
    userid              INTEGER,
    uid                 VARCHAR(64) DEFAULT '',
    lcuuid              VARCHAR(64) DEFAULT '',
    created_at          TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    updated_at          TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    deleted_at          TIMESTAMP DEFAULT NULL
);
TRUNCATE TABLE vm;
CREATE INDEX vm_state_server_index ON vm (state, launch_server);
CREATE INDEX vm_launch_server_index ON vm (launch_server);
CREATE INDEX vm_epc_id_index ON vm (epc_id);
CREATE INDEX vm_az_index ON vm (az);
CREATE INDEX vm_region_index ON vm (region);
COMMENT ON COLUMN vm.state IS '0.Temp 1.Creating 2.Created 3.To run 4.Running
COMMENT ON COLUMN vm.create_method IS '0.learning 1.user_defined';
COMMENT ON COLUMN vm.htype IS '1.vm-c 2.bm-c 3.vm-n 4.bm-n 5.vm-s 6.bm-s';
```

4. PostgreSQL 默认没有启用 `gen_random_uuid()` 函数，并且该函数只能在 public 下使用

4.1. 报错



```
INSERT INTO alarm_policy (
    user_id, tag_conditions, query_conditions, query_url, query_params, name, level, state,
    app_type, contrast_type, target_field, threshold_warning, lcuuid)
VALUES (
    1, '过滤项: N/A', '[{"type":"deepflow","tableName":"deepflow_server_controller_resource_sync_delay","dbName":"deepflow_tenant","metrics":[{"description":"","typeName":"counter","METRIC_CATEGORY":"metrics","METRIC
":"metrics.max_delay","METRIC_NAME":"metrics.max_delay","isTimeUnit":false,"type":1,"unit":["data","short"],"checked":true,"operatorLv2":[],"_key":"77b0ee61-e213-4d10-9342-bb172f861f39","perOperator":"","operatorLv1"
:"Max","percentile":null,"markLine":null,"diffMarkLine":null,"METRIC_LABEL":"Max(metrics.max_delay)","ORIGIN_METRIC_LABEL":"Max(metrics.max_delay)"}],"condition":[{"dbName":"deepflow_tenant","tableName":"deepflow_ser
ver_controller_resource_sync_delay","type":"simplified","RESOURCE_SETS":[{"id":"R1","condition":[],"groupBy":["_","tag.domain"],"groupInfo":{"mainGroupInfo":["_"],"otherGroupInfo":["tag.domain"]},"inputMode":"free"}]
}],"dataSource":""}]',
    '/v1/stats/querier/UniversalHistory', '{"DATABASE":"deepflow_tenant","TABLE":"deepflow_server_controller_resource_sync_delay","interval":60,"fill": "none","window_size":1,"QUERIES":[{"QUERY_ID":"R1","SELECT":"Max
(`metrics.max_delay`) AS `Max(metrics.max_delay)`","WHERE":"1=1","GROUP_BY":"`tag.domain`","METRICS":["Max(`metrics.max_delay`) AS `Max(metrics.max_delay)`"]}]}',
    '云资源同步滞后 (云主机)',  0, 1, 1, 1, '{"displayName":"Max(metrics.max_delay)","unit":""}', '{"OP":">=","VALUE":150}', (SELECT gen_random_uuid()));

INSERT INTO alarm_policy (
    user_id, tag_conditions, query_conditions, query_url, query_params, name, level, state,
    app_type, contrast_type, target_field, threshold_warning, lcuuid)
VALUES (
    1, '过滤项: N/A', '[{"type":"deepflow","tableName":"deepflow_server_controller_resource_sync_delay","dbName":"deepflow_tenant","metrics":[{"description":"","typeName":"counter","METRIC_CATEGORY":"metrics","METRIC
":"metrics.max_delay","METRIC_NAME":"metrics.max_delay","isTimeUnit":false,"type":1,"unit":["data","short"],"checked":true,"operatorLv2":[],"_key":"77b0ee61-e213-4d10-9342-bb172f861f39","perOperator":"","operatorLv1"
:"Max","percentile":null,"markLine":null,"diffMarkLine":null,"METRIC_LABEL":"Max(metrics.max_delay)","ORIGIN_METRIC_LABEL":"Max(metrics.max_delay)"}],"condition":[{"dbName":"deepflow_tenant","tableName":"deepflow_ser
ver_controller_resource_sync_delay","type":"simplified","RESOURCE_SETS":[{"id":"R1","condition":[],"groupBy":["_","tag.domain","tag.sub_domain"],"groupInfo":{"mainGroupInfo":["_"],"otherGroupInfo":["tag.domain","tag.
sub_domain"]},"inputMode":"free"}]}],"dataSource":""}]',
    '/v1/stats/querier/UniversalHistory', '{"DATABASE":"deepflow_tenant","TABLE":"deepflow_server_controller_resource_sync_delay","interval":60,"fill": "none","window_size":1,"QUERIES":[{"QUERY_ID":"R1","SELECT":"Max
(`metrics.max_delay`) AS `Max(metrics.max_delay)`","WHERE":"1=1","GROUP_BY":"`tag.domain`, `tag.sub_domain`","METRICS":["Max(`metrics.max_delay`) AS `Max(metrics.max_delay)`"]}]}',
    '云资源同步滞后 (POD)',  0, 1, 1, 1, '{"displayName":"Max(metrics.max_delay)","unit":""}', '{"OP":">=","VALUE":120}', (SELECT gen_random_uuid()));

INSERT INTO alarm_policy (
    user_id, tag_conditions, query_conditions, query_url, query_params, name, level, state,
    app_type, contrast_type, target_field, threshold_warning, lcuuid)
VALUES (
    1, '过滤项: N/A', '[{"type":"deepflow","tableName":"deepflow_server_controller_cloud_task_cost","dbName":"deepflow_tenant","metrics":[{"description":"","typeName":"counter","METRIC_CATEGORY":"metrics","METRIC":"m
etrics.cost","METRIC_NAME":"metrics.cost","isTimeUnit":false,"type":1,"unit":["data","short"],"checked":true,"operatorLv2":[],"_key":"77b0ee61-e213-4d10-9342-bb172f861f39","perOperator":"","operatorLv1":"AAvg","perce
ntile":null,"markLine":null,"diffMarkLine":null,"METRIC_LABEL":"AAvg(metrics.cost)","ORIGIN_METRIC_LABEL":"AAvg(metrics.cost)"}],"condition":[{"dbName":"deepflow_tenant","tableName":"deepflow_server_controller_cloud_
task_cost","type":"simplified","RESOURCE_SETS":[{"id":"R1","condition":[],"groupBy":["_","tag.domain"],"groupInfo":{"mainGroupInfo":["_"],"otherGroupInfo":["tag.domain"]},"inputMode":"free"}]}],"dataSource":""}]',
    '/v1/stats/querier/UniversalHistory', '{"DATABASE":"deepflow_tenant","TABLE":"deepflow_server_controller_cloud_task_cost","interval":60,"fill": "none","window_size":1,"QUERIES":[{"QUERY_ID":"R1","SELECT":"AAvg(`m
etrics.cost`) AS `AAvg(metrics.cost)`","WHERE":"1=1","GROUP_BY":"`tag.domain`","METRICS":["AAvg(`metrics.cost`) AS `AAvg(metrics.cost)`"]}]}',
    '云资源同步滞后 (API 调用)',  0, 1, 1, 1, '{"displayName":"AAvg(metrics.cost)","unit":""}', '{"OP":">=","VALUE":300}', (SELECT gen_random_uuid()));
2025-12-01 10:52:55.456 [ERRO] [db.metadb.migrator.common] init.go:83 [DB-monitor] failed to execute /etc/metadb/schema/rawsql/postgres/dml_insert.sql: pq: function gen_random_uuid() does not exist
2025-12-01 10:52:55.456 [ERRO] [controller] master.go:69 migrate metadb failed: pq: function gen_random_uuid() does not exist
[root@master ~]#
[root@master ~]#
[root@master ~]#
[root@master ~]#
[root@master ~]#
[root@master ~]#
[root@master ~]#
```

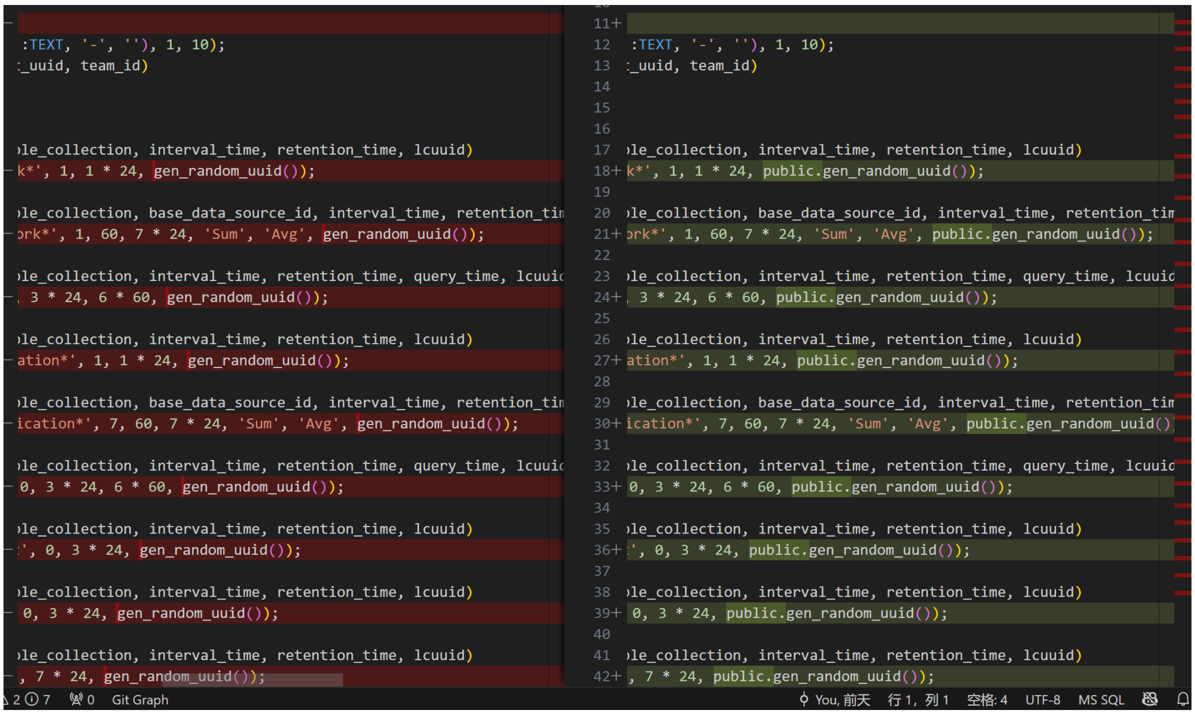
4.2. 改造

PostgreSQL 默认没有启用 `**gen_random_uuid()**` 函数。该函数属于 `pgcrypto` 扩展，必须安装后才能使用，而且必须需要超级管理员才能安装

因为一般不会给应用使用超级管理员用户，因此`pgcrypto` 需要手动使用如下 sql 创建（无法把该 sql 语句放在 deepflow 初始化 sql 里）

```
CREATE EXTENSION IF NOT EXISTS pgcrypto;
```

安装 pgcrypto 后， `**gen_random_uuid()**` 由于只能在 public 这个 schema 下使用，但是配置的 schema 不一定是 public， 因此需要在 sql 中用到 `**gen_random_uuid()**` 函数的地方加上 public.

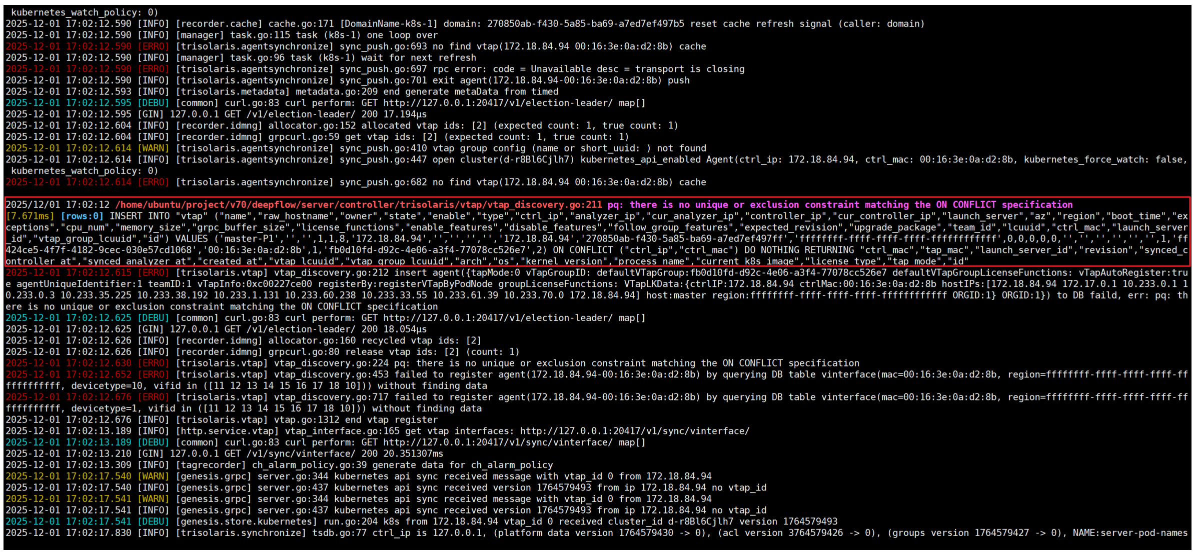


```sql
:TEXT, '-', ''), 1, 10);
:_uuid, team_id)



le_collection, interval_time, retention_time, lcuuid)
k*', 1, 1 * 24, gen_random_uuid());

le_collection, base_data_source_id, interval_time, retention_tim
ork*', 1, 60, 7 * 24, 'Sum', 'Avg', gen_random_uuid());

le_collection, interval_time, retention_time, query_time, lcuuic
, 3 * 24, 6 * 60, gen_random_uuid());

le_collection, interval_time, retention_time, lcuuid)
ation*', 1, 1 * 24, gen_random_uuid());

le_collection, base_data_source_id, interval_time, retention_tim
ication*', 7, 60, 7 * 24, 'Sum', 'Avg', gen_random_uuid());

le_collection, interval_time, retention_time, query_time, lcuuic
0, 3 * 24, 6 * 60, gen_random_uuid());

le_collection, interval_time, retention_time, lcuuid)
', 0, 3 * 24, gen_random_uuid());

le_collection, interval_time, retention_time, lcuuid)
0, 3 * 24, gen_random_uuid());

le_collection, interval_time, retention_time, lcuuid)
, 7 * 24, gen_random_uuid());
```

```sql

:TEXT, '-', ''), 1, 10);
:_uuid, team_id)



le_collection, interval_time, retention_time, lcuuid)
k*', 1, 1 * 24, public.gen_random_uuid());

le_collection, base_data_source_id, interval_time, retention_tim
ork*', 1, 60, 7 * 24, 'Sum', 'Avg', public.gen_random_uuid());

le_collection, interval_time, retention_time, query_time, lcuuic
, 3 * 24, 6 * 60, public.gen_random_uuid());

le_collection, interval_time, retention_time, lcuuid)
ation*', 1, 1 * 24, public.gen_random_uuid());

le_collection, base_data_source_id, interval_time, retention_tim
ication*', 7, 60, 7 * 24, 'Sum', 'Avg', public.gen_random_uuid()

le_collection, interval_time, retention_time, query_time, lcuuic
0, 3 * 24, 6 * 60, public.gen_random_uuid());

le_collection, interval_time, retention_time, lcuuid)
', 0, 3 * 24, public.gen_random_uuid());

le_collection, interval_time, retention_time, lcuuid)
0, 3 * 24, public.gen_random_uuid());

le_collection, interval_time, retention_time, lcuuid)
, 7 * 24, public.gen_random_uuid());
```

5. vtap 表缺少唯一约束

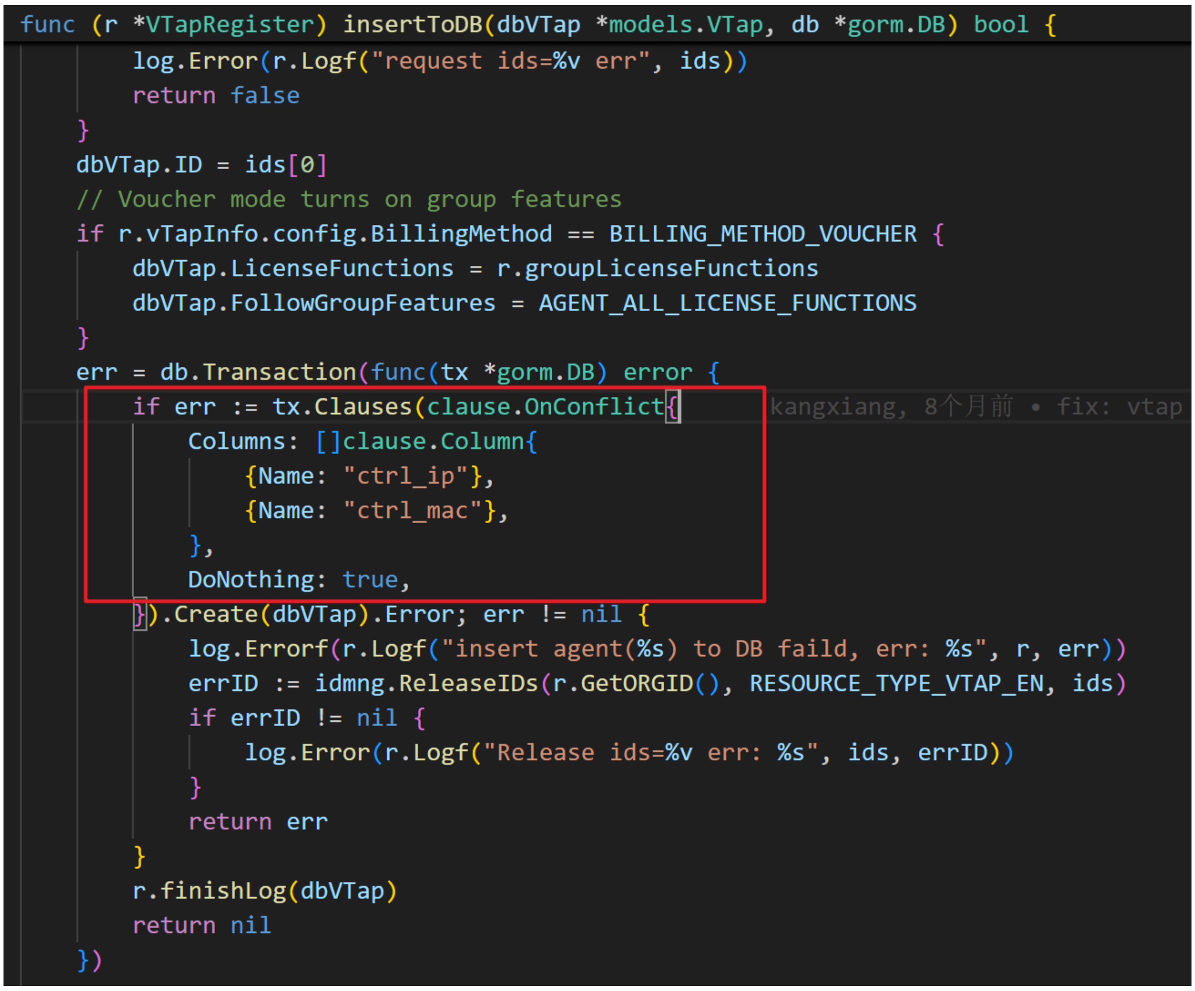
5.1. 报错



```
 kubernetes_watch_policy: 0)
2025-12-01 17:02:12.590 [INFO] [recorder.cache] cache.go:171 [DomainName-k8s-1] domain: 270850ab-f430-5a85-ba69-a7ed7ef497b5 reset cache refresh signal (caller: domain)
2025-12-01 17:02:12.590 [INFO] [manager] task.go:115 task (k8s-1) one loop over
2025-12-01 17:02:12.590 [ERRO] [trisolaris.agentsynchronize] sync_push.go:693 no find vtap(172.18.84.94 00:16:3e:0a:d2:8b) cache
2025-12-01 17:02:12.590 [INFO] [manager] task.go:96 task (k8s-1) wait for next refresh
2025-12-01 17:02:12.590 [ERRO] [trisolaris.agentsynchronize] sync_push.go:697 rpc error: code = Unavailable desc = transport is closing
2025-12-01 17:02:12.590 [INFO] [trisolaris.agentsynchronize] sync_push.go:701 exit agent(172.18.84.94-00:16:3e:0a:d2:8b) push
2025-12-01 17:02:12.593 [INFO] [trisolaris.metadata] metadata.go:209 end generate metaData from timed
2025-12-01 17:02:12.595 [DEBU] [common] curl.go:83 curl perform: GET http://127.0.0.1:20417/v1/election-leader/ map[]
2025-12-01 17:02:12.595 [GIN] 127.0.0.1 GET /v1/election-leader/ 200 17.194µs
2025-12-01 17:02:12.604 [INFO] [recorder.idmng] allocator.go:152 allocated vtap ids: [2] (expected count: 1, true count: 1)
2025-12-01 17:02:12.604 [INFO] [recorder.idmng] grpcurl.go:59 get vtap ids: [2] (expected count: 1, true count: 1)
2025-12-01 17:02:12.614 [WARN] [trisolaris.agentsynchronize] sync_push.go:410 vtap group config (name or short_uuid: ) not found
2025-12-01 17:02:12.614 [INFO] [trisolaris.agentsynchronize] sync_push.go:447 open cluster(d-r8Bl6Cjlh7) kubernetes_api_enabled Agent(ctrl_ip: 172.18.84.94, ctrl_mac: 00:16:3e:0a:d2:8b, kubernetes_force_watch: false,
 kubernetes_watch_policy: 0)
2025-12-01 17:02:12.614 [ERRO] [trisolaris.agentsynchronize] sync_push.go:682 no find vtap(172.18.84.94 00:16:3e:0a:d2:8b) cache

2025/12/01 17:02:12 /home/ubuntu/project/v70/deepflow/server/controller/trisolaris/vtap/vtap_discovery.go:211 pq: there is no unique or exclusion constraint matching the ON CONFLICT specification
[7.671ms] [rows:0] INSERT INTO "vtap" ("name","raw_hostname","owner","state","enable","type","ctrl_ip","analyzer_ip","cur_analyzer_ip","controller_ip","cur_controller_ip","launch_server","az","region","boot_time","ex
ceptions","cpu_num","memory_size","grpc_buffer_size","license_functions","enable_features","disable_features","follow_group_features","expected_revision","upgrade_package","team_id","lcuuid","ctrl_mac","launch_server
_id","vtap_group_lcuuid","id") VALUES ('master-P1','','',1,1,8,'172.18.84.94','','','','','172.18.84.94','270850ab-f430-5a85-ba69-a7ed7ef497ff','ffffffff-ffff-ffff-ffff-ffffffffffff',0,0,0,0,0,'','','','','','',1,'ff
424ce5-4f7f-4182-9cec-030e57cd1068','00:16:3e:0a:d2:8b',1,'fb0d10fd-d92c-4e06-a3f4-77078cc526e7',2) ON CONFLICT ("ctrl_ip","ctrl_mac") DO NOTHING RETURNING "ctrl_mac","tap_mac","launch_server_id","revision","synced_c
ontroller_at","synced_analyzer_at","created_at","vtap_lcuuid","vtap_group_lcuuid","arch","os","kernel_version","process_name","current_k8s_image","license_type","tap_mode","id"
2025-12-01 17:02:12.615 [ERRO] [trisolaris.vtap] vtap_discovery.go:212 insert agent({tapMode:0 vTapGroupID: defaultVTapGroup:fb0d10fd-d92c-4e06-a3f4-77078cc526e7 defaultVTapGroupLicenseFunctions: vTapAutoRegister:tru
e agentUniqueIdentifier:1 teamID:1 vTapInfo:0xc00227ce00 registerBy:registerVTapByPodNode groupLicenseFunctions: VTapLKData:{ctrlIP:172.18.84.94 ctrlMac:00:16:3e:0a:d2:8b hostIPs:[172.18.84.94 172.17.0.1 10.233.0.1 1
0.233.0.3 10.233.35.225 10.233.38.192 10.233.1.131 10.233.60.238 10.233.33.55 10.233.61.39 10.233.70.0 172.18.84.94] host:master region:ffffffff-ffff-ffff-ffff-ffffffffffff ORGID:1} ORGID:1}) to DB faild, err: pq: th
ere is no unique or exclusion constraint matching the ON CONFLICT specification
2025-12-01 17:02:12.625 [DEBU] [common] curl.go:83 curl perform: GET http://127.0.0.1:20417/v1/election-leader/ map[]
2025-12-01 17:02:12.625 [GIN] 127.0.0.1 GET /v1/election-leader/ 200 18.054µs
2025-12-01 17:02:12.626 [INFO] [recorder.idmng] allocator.go:160 recycled vtap ids: [2]
2025-12-01 17:02:12.626 [INFO] [recorder.idmng] grpcurl.go:80 release vtap ids: [2] (count: 1)
2025-12-01 17:02:12.630 [ERRO] [trisolaris.vtap] vtap_discovery.go:224 pq: there is no unique or exclusion constraint matching the ON CONFLICT specification
2025-12-01 17:02:12.652 [ERRO] [trisolaris.vtap] vtap_discovery.go:453 failed to register agent(172.18.84.94-00:16:3e:0a:d2:8b) by querying DB table vinterface(mac=00:16:3e:0a:d2:8b, region=ffffffff-ffff-ffff-ffff-ff
ffffffffff, devicetype=10, vifid in ([11 12 13 14 15 16 17 18 10])) without finding data
2025-12-01 17:02:12.676 [ERRO] [trisolaris.vtap] vtap_discovery.go:717 failed to register agent(172.18.84.94-00:16:3e:0a:d2:8b) by querying DB table vinterface(mac=00:16:3e:0a:d2:8b, region=ffffffff-ffff-ffff-ffff-ff
ffffffffff, devicetype=1, vifid in ([11 12 13 14 15 16 17 18 10])) without finding data
2025-12-01 17:02:12.676 [INFO] [trisolaris.vtap] vtap.go:1312 end vtap register
2025-12-01 17:02:13.189 [INFO] [http.service.vtap] vtap_interface.go:165 get vtap interfaces: http://127.0.0.1:20417/v1/sync/vinterface/
2025-12-01 17:02:13.189 [DEBU] [common] curl.go:83 curl perform: GET http://127.0.0.1:20417/v1/sync/vinterface/ map[]
2025-12-01 17:02:13.210 [GIN] 127.0.0.1 GET /v1/sync/vinterface/ 200 20.351307ms
2025-12-01 17:02:13.309 [INFO] [tagrecorder] ch_alarm_policy.go:39 generate data for ch_alarm_policy
2025-12-01 17:02:17.540 [WARN] [genesis.grpc] server.go:344 kubernetes api sync received message with vtap_id 0 from 172.18.84.94
2025-12-01 17:02:17.540 [INFO] [genesis.grpc] server.go:437 kubernetes api sync received version 1764579493 from ip 172.18.84.94 no vtap_id
2025-12-01 17:02:17.541 [WARN] [genesis.grpc] server.go:344 kubernetes api sync received message with vtap_id 0 from 172.18.84.94
2025-12-01 17:02:17.541 [INFO] [genesis.grpc] server.go:437 kubernetes api sync received version 1764579493 from ip 172.18.84.94 no vtap_id
2025-12-01 17:02:17.541 [DEBU] [genesis.store.kubernetes] run.go:204 k8s from 172.18.84.94 vtap_id 0 received cluster_id d-r8Bl6Cjlh7 version 1764579493
2025-12-01 17:02:17.830 [INFO] [trisolaris.synchronize] tsdb.go:77 ctrl_ip is 127.0.0.1, (platform data version 1764579430 -> 0), (acl version 3764579426 -> 0), (groups version 1764579427 -> 0), NAME:server-pod-names
```

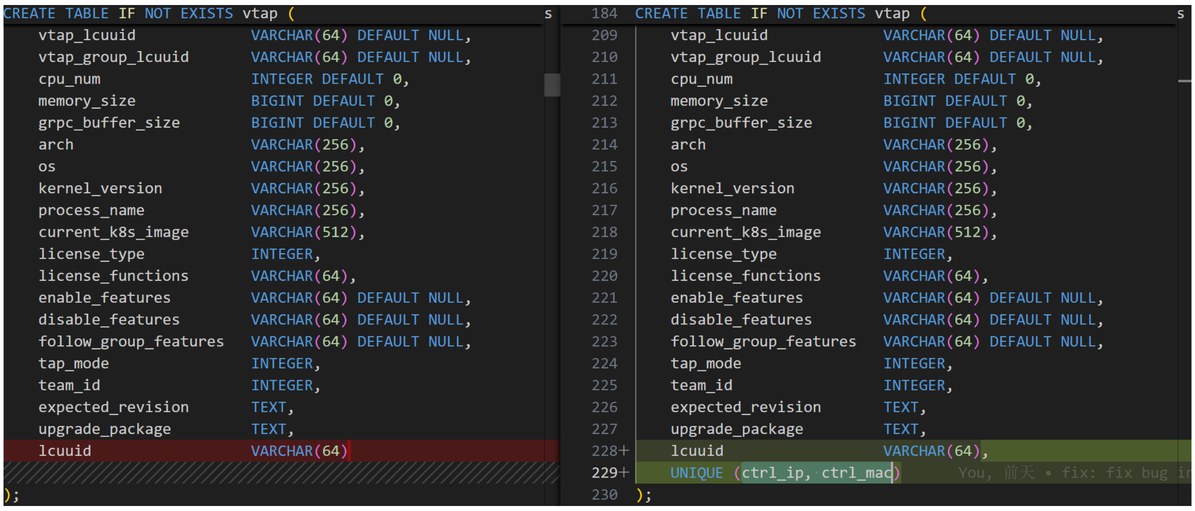
5.2. 改造

使用了ctrl_ip, ctrl_mac 作为 onconflict 字段，但是在创建表时没有进行唯一约束



```go
func (r *VTapRegister) insertToDB(dbVTap *models.VTap, db *gorm.DB) bool {
        log.Error(r.Logf("request ids=%v err", ids))
        return false
    }
    dbVTap.ID = ids[0]
    // Voucher mode turns on group features
    if r.vTapInfo.config.BillingMethod == BILLING_METHOD_VOUCHER {
        dbVTap.LicenseFunctions = r.groupLicenseFunctions
        dbVTap.FollowGroupFeatures = AGENT_ALL_LICENSE_FUNCTIONS
    }
    err = db.Transaction(func(tx *gorm.DB) error {
        if err := tx.Clauses(clause.OnConflict{        kangxiang, 8个月前 • fix: vtap
            Columns: []clause.Column{
                {Name: "ctrl_ip"},
                {Name: "ctrl_mac"},
            },
            DoNothing: true,
        }).Create(dbVTap).Error; err != nil {
            log.Errorf(r.Logf("insert agent(%s) to DB faild, err: %s", r, err))
            errID := idmng.ReleaseIDs(r.GetORGID(), RESOURCE_TYPE_VTAP_EN, ids)
            if errID != nil {
                log.Error(r.Logf("Release ids=%v err: %s", ids, errID))
            }
            return err
        }
        r.finishLog(dbVTap)
        return nil
    })
```

在 vtap 表中添加UNIQUE (ctrl_ip, ctrl_mac)



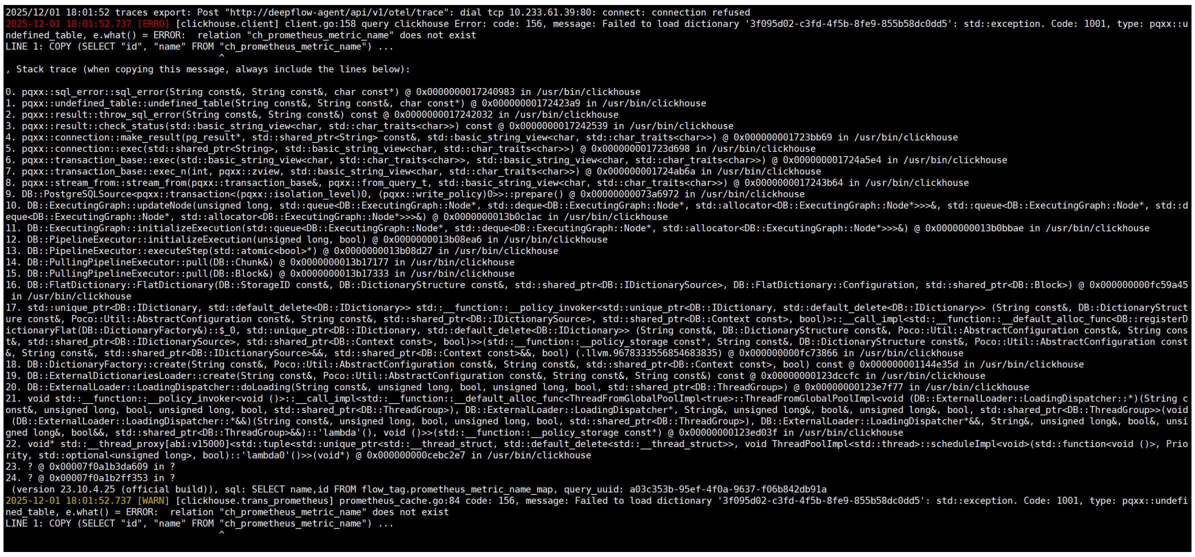
```sql
CREATE TABLE IF NOT EXISTS vtap (
    vtap_lcuuid             VARCHAR(64) DEFAULT NULL,
    vtap_group_lcuuid       VARCHAR(64) DEFAULT NULL,
    cpu_num                 INTEGER DEFAULT 0,
    memory_size             BIGINT DEFAULT 0,
    grpc_buffer_size        BIGINT DEFAULT 0,
    arch                    VARCHAR(256),
    os                      VARCHAR(256),
    kernel_version          VARCHAR(256),
    process_name            VARCHAR(256),
    current_k8s_image       VARCHAR(512),
    license_type            INTEGER,
    license_functions       VARCHAR(64),
    enable_features         VARCHAR(64) DEFAULT NULL,
    disable_features        VARCHAR(64) DEFAULT NULL,
    follow_group_features   VARCHAR(64) DEFAULT NULL,
    tap_mode                INTEGER,
    team_id                 INTEGER,
    expected_revision       TEXT,
    upgrade_package         TEXT,
    lcuuid                  VARCHAR(64)
);
```

```sql
CREATE TABLE IF NOT EXISTS vtap (
    vtap_lcuuid             VARCHAR(64) DEFAULT NULL,
    vtap_group_lcuuid       VARCHAR(64) DEFAULT NULL,
    cpu_num                 INTEGER DEFAULT 0,
    memory_size             BIGINT DEFAULT 0,
    grpc_buffer_size        BIGINT DEFAULT 0,
    arch                    VARCHAR(256),
    os                      VARCHAR(256),
    kernel_version          VARCHAR(256),
    process_name            VARCHAR(256),
    current_k8s_image       VARCHAR(512),
    license_type            INTEGER,
    license_functions       VARCHAR(64),
    enable_features         VARCHAR(64) DEFAULT NULL,
    disable_features        VARCHAR(64) DEFAULT NULL,
    follow_group_features   VARCHAR(64) DEFAULT NULL,
    tap_mode                INTEGER,
    team_id                 INTEGER,
    expected_revision       TEXT,
    upgrade_package         TEXT,
    lcuuid                  VARCHAR(64),
+   UNIQUE (ctrl_ip, ctrl_mac)        You, 前天 • fix: fix bug i
);
```

6. 不支持配置除 public 之外的 schema

6.1. 报错

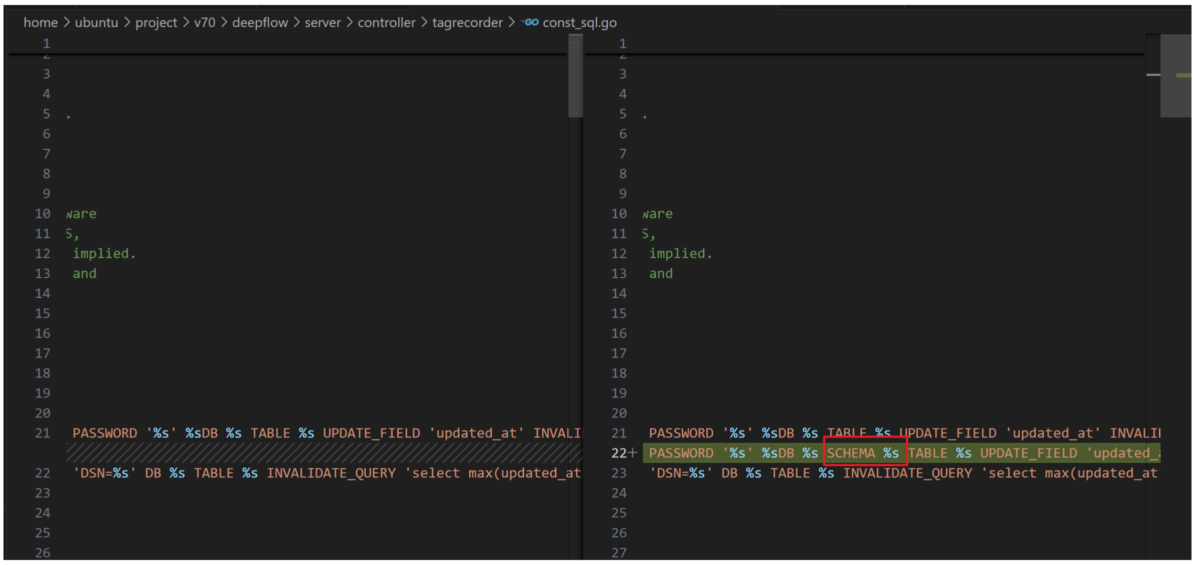
报错找不到字典表，但是该字典表在 postgres 中是有的，原因是因为创建字典时的 sql 没有指定配置的 schema，默认使用的时 public



```
2025/12/01 18:01:52 traces export: Post "http://deepflow-agent/api/v1/otel/trace": dial tcp 10.233.61.39:80: connect: connection refused
2025-12-01 18:01:52.737 [ERRO] [clickhouse.client] client.go:158 query clickhouse Error: code: 156, message: Failed to load dictionary '3f095d02-c3fd-4f5b-8fe9-855b58dc0dd5': std::exception. Code: 1001, type: pqxx::u
ndefined_table, e.what() = ERROR:  relation "ch_prometheus_metric_name" does not exist
LINE 1: COPY (SELECT "id", "name" FROM "ch_prometheus_metric_name") ...
                                       ^

, Stack trace (when copying this message, always include the lines below):

0. pqxx::sql_error::sql_error(String const&, String const&, char const*) @ 0x0000000017240983 in /usr/bin/clickhouse
1. pqxx::undefined_table::undefined_table(String const&, String const&, char const*) @ 0x00000000172423a9 in /usr/bin/clickhouse
2. pqxx::result::throw_sql_error(String const&, String const&) const @ 0x0000000017242032 in /usr/bin/clickhouse
3. pqxx::result::check_status(std::basic_string_view<char, std::char_traits<char>>) const @ 0x0000000017242539 in /usr/bin/clickhouse
4. pqxx::connection::make_result(pg_result*, std::shared_ptr<String> const&, std::basic_string_view<char, std::char_traits<char>>) @ 0x000000001723bb69 in /usr/bin/clickhouse
5. pqxx::connection::exec(std::shared_ptr<String>, std::basic_string_view<char, std::char_traits<char>>) @ 0x000000001723d698 in /usr/bin/clickhouse
6. pqxx::transaction_base::exec(std::basic_string_view<char, std::char_traits<char>>, std::basic_string_view<char, std::char_traits<char>>) @ 0x000000001724a5e4 in /usr/bin/clickhouse
7. pqxx::transaction_base::exec_n(int, pqxx::zview, std::basic_string_view<char, std::char_traits<char>>) @ 0x000000001724ab6a in /usr/bin/clickhouse
8. pqxx::stream_from::stream_from(pqxx::transaction_base&, pqxx::from_query_t, std::basic_string_view<char, std::char_traits<char>>) @ 0x0000000017243b64 in /usr/bin/clickhouse
9. DB::PostgreSQLSource<pqxx::transaction<(pqxx::isolation_level)0, (pqxx::write_policy)0>>::prepare() @ 0x00000000073a6972 in /usr/bin/clickhouse
10. DB::ExecutingGraph::updateNode(unsigned long, std::queue<DB::ExecutingGraph::Node*, std::deque<DB::ExecutingGraph::Node*, std::allocator<DB::ExecutingGraph::Node*>>>&, std::queue<DB::ExecutingGraph::Node*, std::d
eque<DB::ExecutingGraph::Node*, std::allocator<DB::ExecutingGraph::Node*>>>&) @ 0x0000000013b0c1ac in /usr/bin/clickhouse
11. DB::ExecutingGraph::initializeExecution(std::queue<DB::ExecutingGraph::Node*, std::deque<DB::ExecutingGraph::Node*, std::allocator<DB::ExecutingGraph::Node*>>>&) @ 0x0000000013b0bbae in /usr/bin/clickhouse
12. DB::PipelineExecutor::initializeExecution(unsigned long, bool) @ 0x0000000013b08ea6 in /usr/bin/clickhouse
13. DB::PipelineExecutor::executeStep(std::atomic<bool>*) @ 0x0000000013b08d27 in /usr/bin/clickhouse
14. DB::PullingPipelineExecutor::pull(DB::Chunk&) @ 0x0000000013b17177 in /usr/bin/clickhouse
15. DB::PullingPipelineExecutor::pull(DB::Block&) @ 0x0000000013b17333 in /usr/bin/clickhouse
16. DB::FlatDictionary::FlatDictionary(DB::StorageID const&, DB::DictionaryStructure const&, std::shared_ptr<DB::IDictionarySource>, DB::FlatDictionary::Configuration, std::shared_ptr<DB::Block>) @ 0x000000000fc59a45
 in /usr/bin/clickhouse
17. std::unique_ptr<DB::IDictionary, std::default_delete<DB::IDictionary>> std::__function::__policy_invoker<std::unique_ptr<DB::IDictionary, std::default_delete<DB::IDictionary>> (String const&, DB::DictionaryStruct
ure const&, Poco::Util::AbstractConfiguration const&, String const&, std::shared_ptr<DB::IDictionarySource>, std::shared_ptr<DB::Context const>, bool)>::__call_impl<std::__function::__default_alloc_func<DB::registerD
ictionaryFlat(DB::DictionaryFactory&)::$_0, std::unique_ptr<DB::IDictionary, std::default_delete<DB::IDictionary>> (String const&, DB::DictionaryStructure const&, Poco::Util::AbstractConfiguration const&, String cons
t&, std::shared_ptr<DB::IDictionarySource>, std::shared_ptr<DB::Context const>, bool)>>(std::__function::__policy_storage const*, String const&, DB::DictionaryStructure const&, Poco::Util::AbstractConfiguration const
&, String const&, std::shared_ptr<DB::IDictionarySource>&&, std::shared_ptr<DB::Context const>&&, bool) (.llvm.9678333556854683835) @ 0x000000000fc73866 in /usr/bin/clickhouse
18. DB::DictionaryFactory::create(String const&, Poco::Util::AbstractConfiguration const&, String const&, std::shared_ptr<DB::Context const>, bool) const @ 0x000000001144e35d in /usr/bin/clickhouse
19. DB::ExternalDictionariesLoader::create(String const&, Poco::Util::AbstractConfiguration const&, String const&, String const&) const @ 0x00000000123dccfc in /usr/bin/clickhouse
20. DB::ExternalLoader::LoadingDispatcher::doLoading(String const&, unsigned long, bool, unsigned long, bool, std::shared_ptr<DB::ThreadGroup>) @ 0x00000000123e7f77 in /usr/bin/clickhouse
21. void std::__function::__policy_invoker<void ()>::__call_impl<std::__function::__default_alloc_func<ThreadFromGlobalPoolImpl<true>::ThreadFromGlobalPoolImpl<void (DB::ExternalLoader::LoadingDispatcher::*)(String c
onst&, unsigned long, bool, unsigned long, bool, std::shared_ptr<DB::ThreadGroup>), DB::ExternalLoader::LoadingDispatcher*, String&, unsigned long&, bool&, unsigned long&, bool, std::shared_ptr<DB::ThreadGroup>>(void
 (DB::ExternalLoader::LoadingDispatcher::*&&)(String const&, unsigned long, bool, unsigned long, bool, std::shared_ptr<DB::ThreadGroup>), DB::ExternalLoader::LoadingDispatcher*&&, String&, unsigned long&, bool&, unsi
gned long&, bool&&, std::shared_ptr<DB::ThreadGroup>&&)::'lambda'(), void ()>>(std::__function::__policy_storage const*) @ 0x00000000123ed03f in /usr/bin/clickhouse
22. void* std::__thread_proxy[abi:v15000]<std::tuple<std::unique_ptr<std::__thread_struct, std::default_delete<std::__thread_struct>>, void ThreadPoolImpl<std::thread>::scheduleImpl<void>(std::function<void ()>, Prio
rity, std::optional<unsigned long>, bool)::'lambda0'()>>(void*) @ 0x000000000cebc2e7 in /usr/bin/clickhouse
23. ? @ 0x00007f0a1b3da609 in ?
24. ? @ 0x00007f0a1b2ff353 in ?
 (version 23.10.4.25 (official build)), sql: SELECT name,id FROM flow_tag.prometheus_metric_name_map, query_uuid: a03c353b-95ef-4f0a-9637-f06b842db91a
2025-12-01 18:01:52.737 [WARN] [clickhouse.trans_prometheus] prometheus_cache.go:84 code: 156, message: Failed to load dictionary '3f095d02-c3fd-4f5b-8fe9-855b58dc0dd5': std::exception. Code: 1001, type: pqxx::undefi
ned_table, e.what() = ERROR:  relation "ch_prometheus_metric_name" does not exist
LINE 1: COPY (SELECT "id", "name" FROM "ch_prometheus_metric_name") ...
                                       ^
```

6.2. 改造

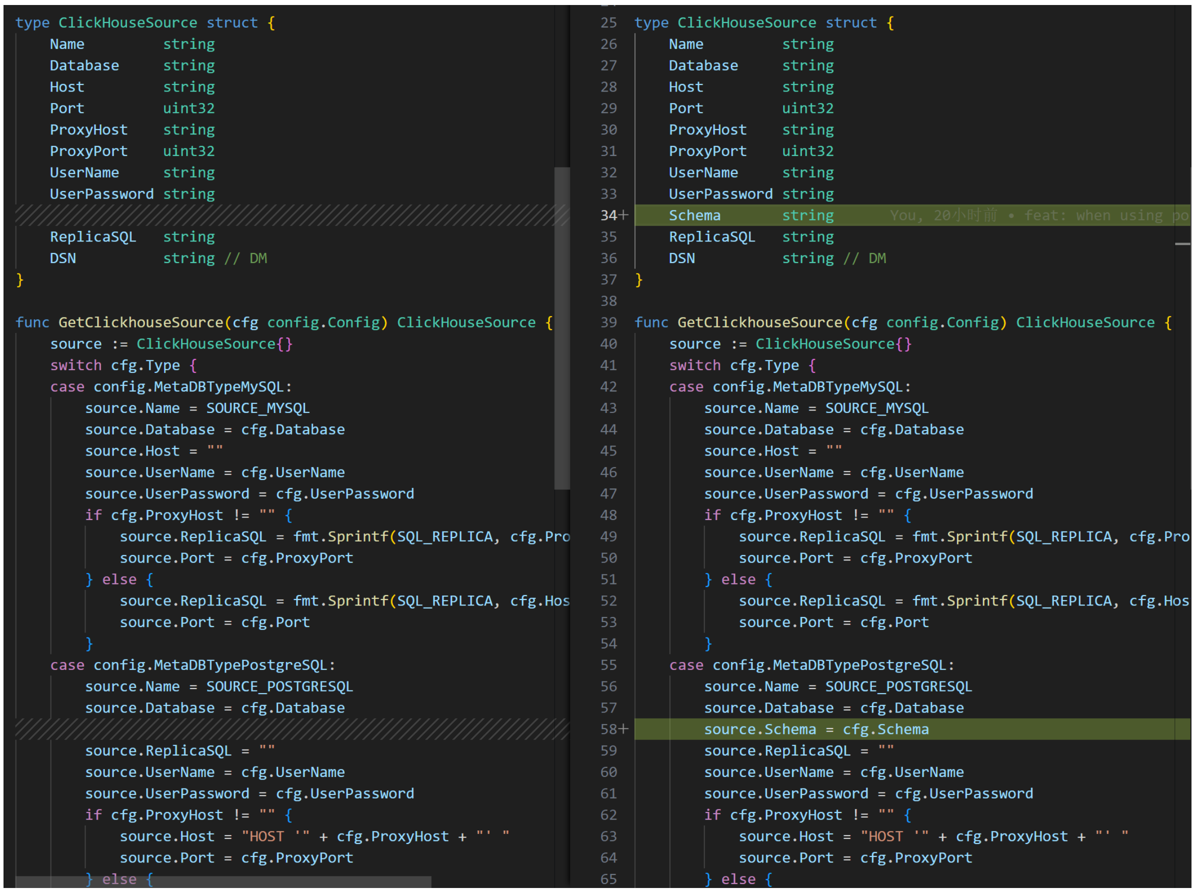
添加 schema 配置，允许自定义 schema



home > ubuntu > project > v70 > deepflow > server > controller > tagrecorder > const_sql.go

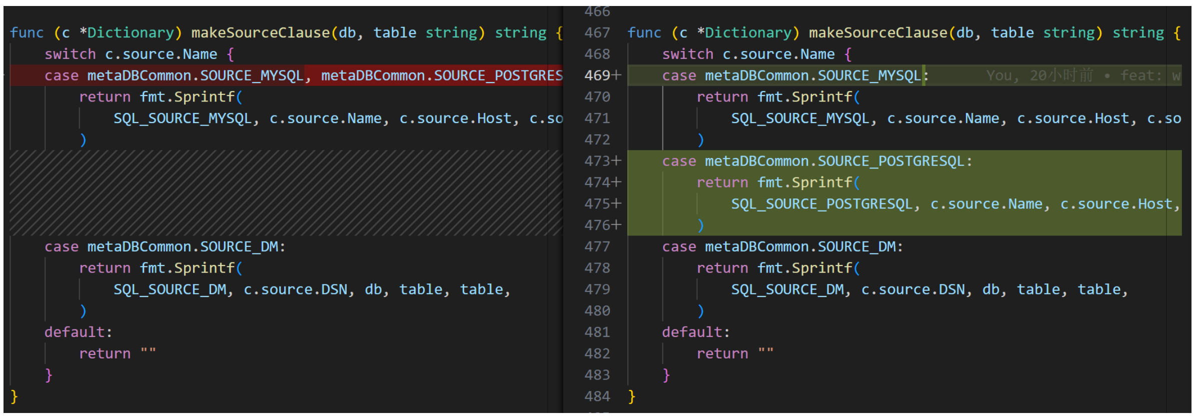
```
.




ware
S,
 implied.
 and







 PASSWORD '%s' %sDB %s TABLE %s UPDATE_FIELD 'updated_at' INVALI

 'DSN=%s' DB %s TABLE %s INVALIDATE_QUERY 'select max(updated_at
```

```
.




ware
S,
 implied.
 and







 PASSWORD '%s' %sDB %s TABLE %s UPDATE_FIELD 'updated_at' INVALI
+ PASSWORD '%s' %sDB %s SCHEMA %s TABLE %s UPDATE_FIELD 'updated_
 'DSN=%s' DB %s TABLE %s INVALIDATE_QUERY 'select max(updated_at
```



```go
type ClickHouseSource struct {
    Name         string
    Database     string
    Host         string
    Port         uint32
    ProxyHost    string
    ProxyPort    uint32
    UserName     string
    UserPassword string

    ReplicaSQL   string
    DSN          string // DM
}

func GetClickhouseSource(cfg config.Config) ClickHouseSource {
    source := ClickHouseSource{}
    switch cfg.Type {
    case config.MetaDBTypeMySQL:
        source.Name = SOURCE_MYSQL
        source.Database = cfg.Database
        source.Host = ""
        source.UserName = cfg.UserName
        source.UserPassword = cfg.UserPassword
        if cfg.ProxyHost != "" {
            source.ReplicaSQL = fmt.Sprintf(SQL_REPLICA, cfg.Pro
            source.Port = cfg.ProxyPort
        } else {
            source.ReplicaSQL = fmt.Sprintf(SQL_REPLICA, cfg.Hos
            source.Port = cfg.Port
        }
    case config.MetaDBTypePostgreSQL:
        source.Name = SOURCE_POSTGRESQL
        source.Database = cfg.Database

        source.ReplicaSQL = ""
        source.UserName = cfg.UserName
        source.UserPassword = cfg.UserPassword
        if cfg.ProxyHost != "" {
            source.Host = "HOST '" + cfg.ProxyHost + "' "
            source.Port = cfg.ProxyPort
        } else {
```

```go
type ClickHouseSource struct {
    Name         string
    Database     string
    Host         string
    Port         uint32
    ProxyHost    string
    ProxyPort    uint32
    UserName     string
    UserPassword string
    Schema       string        You, 20小时前 • feat: when using po
    ReplicaSQL   string
    DSN          string // DM
}

func GetClickhouseSource(cfg config.Config) ClickHouseSource {
    source := ClickHouseSource{}
    switch cfg.Type {
    case config.MetaDBTypeMySQL:
        source.Name = SOURCE_MYSQL
        source.Database = cfg.Database
        source.Host = ""
        source.UserName = cfg.UserName
        source.UserPassword = cfg.UserPassword
        if cfg.ProxyHost != "" {
            source.ReplicaSQL = fmt.Sprintf(SQL_REPLICA, cfg.Pro
            source.Port = cfg.ProxyPort
        } else {
            source.ReplicaSQL = fmt.Sprintf(SQL_REPLICA, cfg.Hos
            source.Port = cfg.Port
        }
    case config.MetaDBTypePostgreSQL:
        source.Name = SOURCE_POSTGRESQL
        source.Database = cfg.Database
        source.Schema = cfg.Schema
        source.ReplicaSQL = ""
        source.UserName = cfg.UserName
        source.UserPassword = cfg.UserPassword
        if cfg.ProxyHost != "" {
            source.Host = "HOST '" + cfg.ProxyHost + "' "
            source.Port = cfg.ProxyPort
        } else {
```



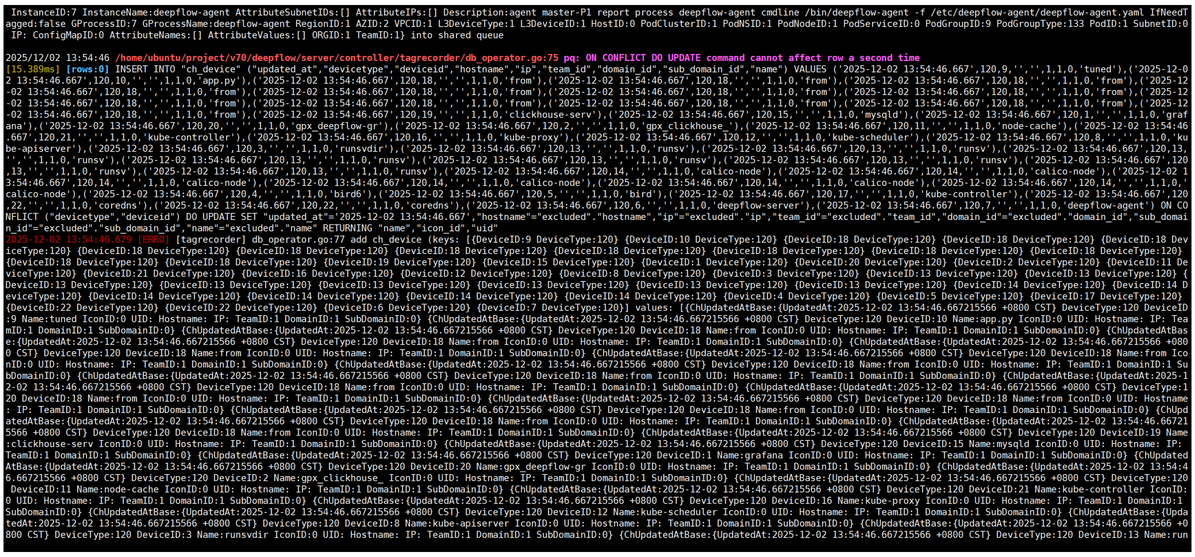
```go
func (c *Dictionary) makeSourceClause(db, table string) string {
    switch c.source.Name {
    case metaDBCommon.SOURCE_MYSQL, metaDBCommon.SOURCE_POSTGRES
        return fmt.Sprintf(
            SQL_SOURCE_MYSQL, c.source.Name, c.source.Host, c.so
        )

    case metaDBCommon.SOURCE_DM:
        return fmt.Sprintf(
            SQL_SOURCE_DM, c.source.DSN, db, table, table,
        )
    default:
        return ""
    }
}
```

```go
func (c *Dictionary) makeSourceClause(db, table string) string {
    switch c.source.Name {
    case metaDBCommon.SOURCE_MYSQL:        You, 20小时前 • feat: w
        return fmt.Sprintf(
            SQL_SOURCE_MYSQL, c.source.Name, c.source.Host, c.so
        )
    case metaDBCommon.SOURCE_POSTGRESQL:
        return fmt.Sprintf(
            SQL_SOURCE_POSTGRESQL, c.source.Name, c.source.Host,
        )
    case metaDBCommon.SOURCE_DM:
        return fmt.Sprintf(
            SQL_SOURCE_DM, c.source.DSN, db, table, table,
        )
    default:
        return ""
    }
}
```

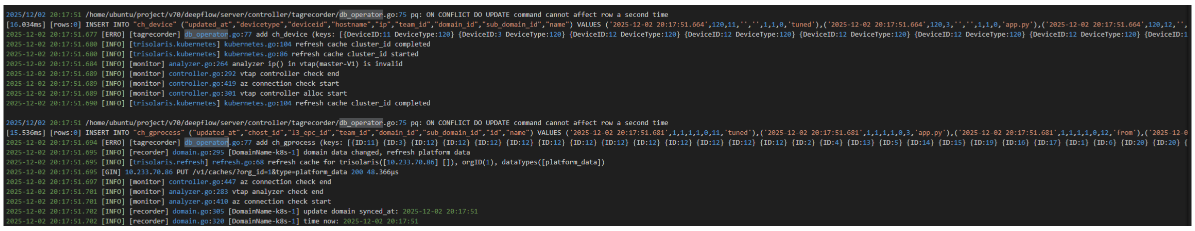
7. 在 postgres 中，同一个 insert 语句不允许多行更新同一个冲突目标

7.1. 报错

在同一个 `**INSERT**` 语句中，有多个行尝试更新（或插入）到同一个冲突目标（即相同的 `**ON CONFLICT (...)**` 列组合）上



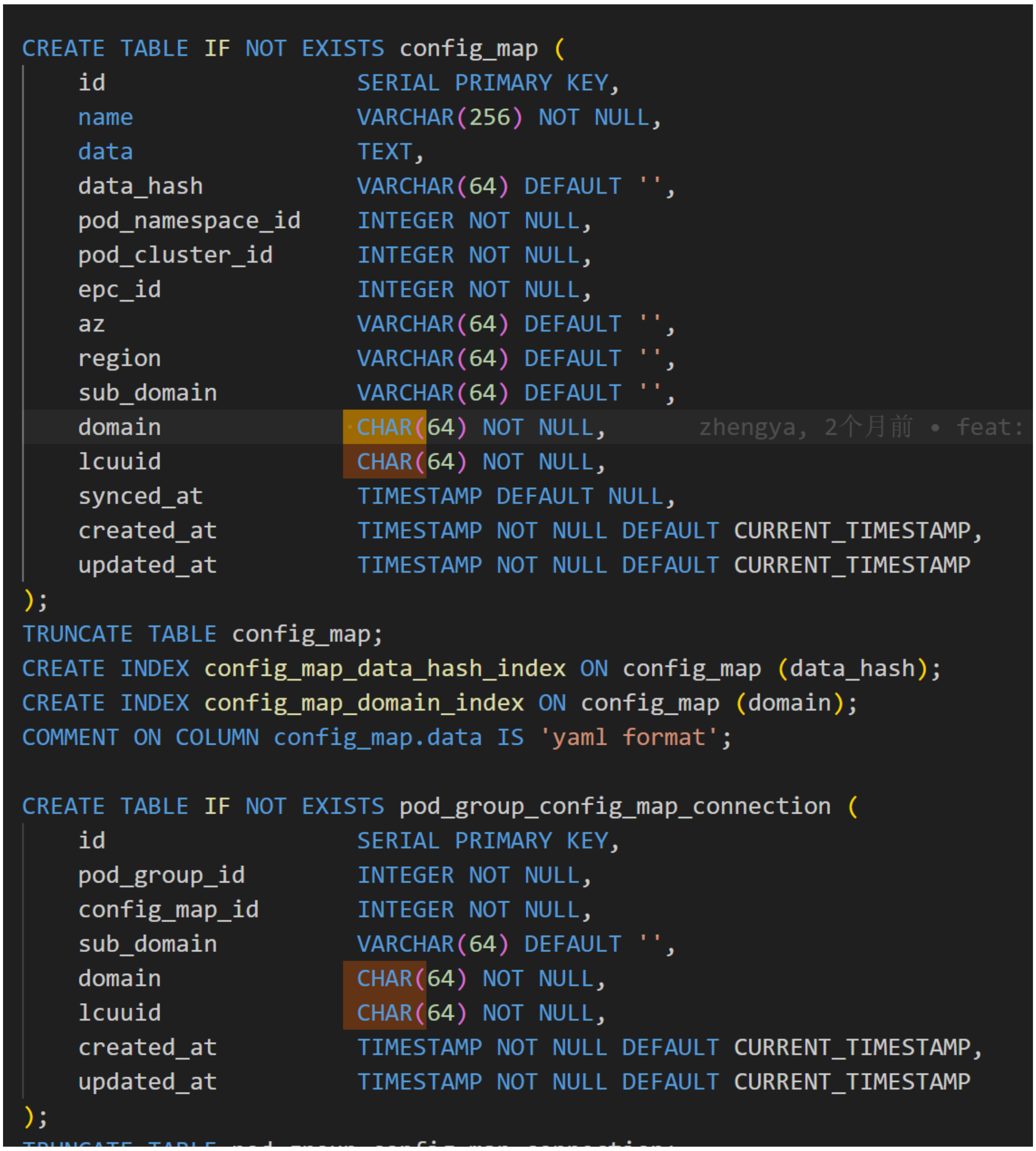
```
 InstanceID:7 InstanceName:deepflow-agent AttributeSubnetIDs:[] AttributeIPs:[] Description:agent master-P1 report process deepflow-agent cmdline /bin/deepflow-agent -f /etc/deepflow-agent/deepflow-agent.yaml IfNeedT
agged:false GProcessID:7 GProcessName:deepflow-agent RegionID:1 AZID:2 VPCID:1 L3DeviceType:1 L3DeviceID:1 HostID:0 PodClusterID:1 PodNSID:1 PodNodeID:1 PodServiceID:0 PodGroupID:9 PodGroupType:133 PodID:1 SubnetID:0
 IP: ConfigMapID:0 AttributeNames:[] AttributeValues:[] ORGID:1 TeamID:1} into shared queue

2025/12/02 13:54:46 /home/ubuntu/project/v70/deepflow/server/controller/tagrecorder/db_operator.go:75 pq: ON CONFLICT DO UPDATE command cannot affect row a second time
[15.389ms] [rows:0] INSERT INTO "ch_device" ("updated_at","devicetype","deviceid","hostname","ip","team_id","domain_id","sub_domain_id","name") VALUES ('2025-12-02 13:54:46.667',120,9,'','',1,1,0,'tuned'),('2025-12-0
2 13:54:46.667',120,10,'','',1,1,0,'app.py'),('2025-12-02 13:54:46.667',120,18,'','',1,1,0,'from'),('2025-12-02 13:54:46.667',120,18,'','',1,1,0,'from'),('2025-12-02 13:54:46.667',120,18,'','',1,1,0,'from'),('2025-12
-02 13:54:46.667',120,18,'','',1,1,0,'from'),('2025-12-02 13:54:46.667',120,18,'','',1,1,0,'from'),('2025-12-02 13:54:46.667',120,18,'','',1,1,0,'from'),('2025-12-02 13:54:46.667',120,18,'','',1,1,0,'from'),('2025-12
-02 13:54:46.667',120,18,'','',1,1,0,'from'),('2025-12-02 13:54:46.667',120,18,'','',1,1,0,'from'),('2025-12-02 13:54:46.667',120,18,'','',1,1,0,'from'),('2025-12-02 13:54:46.667',120,18,'','',1,1,0,'from'),('2025-12
-02 13:54:46.667',120,18,'','',1,1,0,'from'),('2025-12-02 13:54:46.667',120,19,'','',1,1,0,'clickhouse-serv'),('2025-12-02 13:54:46.667',120,15,'','',1,1,0,'mysqld'),('2025-12-02 13:54:46.667',120,1,'','',1,1,0,'graf
ana'),('2025-12-02 13:54:46.667',120,20,'','',1,1,0,'gpx_deepflow-gr'),('2025-12-02 13:54:46.667',120,2,'','',1,1,0,'gpx_clickhouse_'),('2025-12-02 13:54:46.667',120,11,'','',1,1,0,'node-cache'),('2025-12-02 13:54:46
.667',120,21,'','',1,1,0,'kube-controller'),('2025-12-02 13:54:46.667',120,16,'','',1,1,0,'kube-proxy'),('2025-12-02 13:54:46.667',120,12,'','',1,1,0,'kube-scheduler'),('2025-12-02 13:54:46.667',120,8,'','',1,1,0,'ku
be-apiserver'),('2025-12-02 13:54:46.667',120,3,'','',1,1,0,'runsvdir'),('2025-12-02 13:54:46.667',120,13,'','',1,1,0,'runsv'),('2025-12-02 13:54:46.667',120,13,'','',1,1,0,'runsv'),('2025-12-02 13:54:46.667',120,13,
'','',1,1,0,'runsv'),('2025-12-02 13:54:46.667',120,13,'','',1,1,0,'runsv'),('2025-12-02 13:54:46.667',120,13,'','',1,1,0,'runsv'),('2025-12-02 13:54:46.667',120,13,'','',1,1,0,'runsv'),('2025-12-02 13:54:46.667',120
,13,'','',1,1,0,'runsv'),('2025-12-02 13:54:46.667',120,13,'','',1,1,0,'runsv'),('2025-12-02 13:54:46.667',120,14,'','',1,1,0,'calico-node'),('2025-12-02 13:54:46.667',120,14,'','',1,1,0,'calico-node'),('2025-12-02 1
3:54:46.667',120,14,'','',1,1,0,'calico-node'),('2025-12-02 13:54:46.667',120,14,'','',1,1,0,'calico-node'),('2025-12-02 13:54:46.667',120,14,'','',1,1,0,'calico-node'),('2025-12-02 13:54:46.667',120,14,'','',1,1,0,'
calico-node'),('2025-12-02 13:54:46.667',120,4,'','',1,1,0,'bird6'),('2025-12-02 13:54:46.667',120,5,'','',1,1,0,'bird'),('2025-12-02 13:54:46.667',120,17,'','',1,1,0,'kube-controller'),('2025-12-02 13:54:46.667',120
,22,'','',1,1,0,'coredns'),('2025-12-02 13:54:46.667',120,22,'','',1,1,0,'coredns'),('2025-12-02 13:54:46.667',120,6,'','',1,1,0,'deepflow-server'),('2025-12-02 13:54:46.667',120,7,'','',1,1,0,'deepflow-agent') ON CO
NFLICT ("devicetype","deviceid") DO UPDATE SET "updated_at"='2025-12-02 13:54:46.667',"hostname"="excluded"."hostname","ip"="excluded"."ip","team_id"="excluded"."team_id","domain_id"="excluded"."domain_id","sub_domai
n_id"="excluded"."sub_domain_id","name"="excluded"."name" RETURNING "name","icon_id","uid"
2025-12-02 13:54:46.679 [ERRO] [tagrecorder] db_operator.go:77 add ch_device (keys: [{DeviceID:9 DeviceType:120} {DeviceID:10 DeviceType:120} {DeviceID:18 DeviceType:120} {DeviceID:18 DeviceType:120} {DeviceID:18 Dev
iceType:120} {DeviceID:18 DeviceType:120} {DeviceID:18 DeviceType:120} {DeviceID:18 DeviceType:120} {DeviceID:18 DeviceType:120} {DeviceID:18 DeviceType:120} {DeviceID:18 DeviceType:120} {DeviceID:18 DeviceType:120}
{DeviceID:18 DeviceType:120} {DeviceID:18 DeviceType:120} {DeviceID:19 DeviceType:120} {DeviceID:15 DeviceType:120} {DeviceID:1 DeviceType:120} {DeviceID:20 DeviceType:120} {DeviceID:2 DeviceType:120} {DeviceID:11 De
viceType:120} {DeviceID:21 DeviceType:120} {DeviceID:16 DeviceType:120} {DeviceID:12 DeviceType:120} {DeviceID:8 DeviceType:120} {DeviceID:3 DeviceType:120} {DeviceID:13 DeviceType:120} {DeviceID:13 DeviceType:120} {
DeviceID:13 DeviceType:120} {DeviceID:13 DeviceType:120} {DeviceID:13 DeviceType:120} {DeviceID:13 DeviceType:120} {DeviceID:13 DeviceType:120} {DeviceID:13 DeviceType:120} {DeviceID:14 DeviceType:120} {DeviceID:14 D
eviceType:120} {DeviceID:14 DeviceType:120} {DeviceID:14 DeviceType:120} {DeviceID:14 DeviceType:120} {DeviceID:14 DeviceType:120} {DeviceID:4 DeviceType:120} {DeviceID:5 DeviceType:120} {DeviceID:17 DeviceType:120}
{DeviceID:22 DeviceType:120} {DeviceID:22 DeviceType:120} {DeviceID:6 DeviceType:120} {DeviceID:7 DeviceType:120}] values: [{ChUpdatedAtBase:{UpdatedAt:2025-12-02 13:54:46.667215566 +0800 CST} DeviceType:120 DeviceID
:9 Name:tuned IconID:0 UID: Hostname: IP: TeamID:1 DomainID:1 SubDomainID:0} {ChUpdatedAtBase:{UpdatedAt:2025-12-02 13:54:46.667215566 +0800 CST} DeviceType:120 DeviceID:10 Name:app.py IconID:0 UID: Hostname: IP: Tea
mID:1 DomainID:1 SubDomainID:0} {ChUpdatedAtBase:{UpdatedAt:2025-12-02 13:54:46.667215566 +0800 CST} DeviceType:120 DeviceID:18 Name:from IconID:0 UID: Hostname: IP: TeamID:1 DomainID:1 SubDomainID:0} {ChUpdatedAtBas
e:{UpdatedAt:2025-12-02 13:54:46.667215566 +0800 CST} DeviceType:120 DeviceID:18 Name:from IconID:0 UID: Hostname: IP: TeamID:1 DomainID:1 SubDomainID:0} {ChUpdatedAtBase:{UpdatedAt:2025-12-02 13:54:46.667215566 +080
0 CST} DeviceType:120 DeviceID:18 Name:from IconID:0 UID: Hostname: IP: TeamID:1 DomainID:1 SubDomainID:0} {ChUpdatedAtBase:{UpdatedAt:2025-12-02 13:54:46.667215566 +0800 CST} DeviceType:120 DeviceID:18 Name:from Ico
nID:0 UID: Hostname: IP: TeamID:1 DomainID:1 SubDomainID:0} {ChUpdatedAtBase:{UpdatedAt:2025-12-02 13:54:46.667215566 +0800 CST} DeviceType:120 DeviceID:18 Name:from IconID:0 UID: Hostname: IP: TeamID:1 DomainID:1 Su
bDomainID:0} {ChUpdatedAtBase:{UpdatedAt:2025-12-02 13:54:46.667215566 +0800 CST} DeviceType:120 DeviceID:18 Name:from IconID:0 UID: Hostname: IP: TeamID:1 DomainID:1 SubDomainID:0} {ChUpdatedAtBase:{UpdatedAt:2025-1
2-02 13:54:46.667215566 +0800 CST} DeviceType:120 DeviceID:18 Name:from IconID:0 UID: Hostname: IP: TeamID:1 DomainID:1 SubDomainID:0} {ChUpdatedAtBase:{UpdatedAt:2025-12-02 13:54:46.667215566 +0800 CST} DeviceType:1
20 DeviceID:18 Name:from IconID:0 UID: Hostname: IP: TeamID:1 DomainID:1 SubDomainID:0} {ChUpdatedAtBase:{UpdatedAt:2025-12-02 13:54:46.667215566 +0800 CST} DeviceType:120 DeviceID:18 Name:from IconID:0 UID: Hostname
: IP: TeamID:1 DomainID:1 SubDomainID:0} {ChUpdatedAtBase:{UpdatedAt:2025-12-02 13:54:46.667215566 +0800 CST} DeviceType:120 DeviceID:18 Name:from IconID:0 UID: Hostname: IP: TeamID:1 DomainID:1 SubDomainID:0} {ChUpd
atedAtBase:{UpdatedAt:2025-12-02 13:54:46.667215566 +0800 CST} DeviceType:120 DeviceID:18 Name:from IconID:0 UID: Hostname: IP: TeamID:1 DomainID:1 SubDomainID:0} {ChUpdatedAtBase:{UpdatedAt:2025-12-02 13:54:46.66721
5566 +0800 CST} DeviceType:120 DeviceID:18 Name:from IconID:0 UID: Hostname: IP: TeamID:1 DomainID:1 SubDomainID:0} {ChUpdatedAtBase:{UpdatedAt:2025-12-02 13:54:46.667215566 +0800 CST} DeviceType:120 DeviceID:19 Name
:clickhouse-serv IconID:0 UID: Hostname: IP: TeamID:1 DomainID:1 SubDomainID:0} {ChUpdatedAtBase:{UpdatedAt:2025-12-02 13:54:46.667215566 +0800 CST} DeviceType:120 DeviceID:15 Name:mysqld IconID:0 UID: Hostname: IP:
TeamID:1 DomainID:1 SubDomainID:0} {ChUpdatedAtBase:{UpdatedAt:2025-12-02 13:54:46.667215566 +0800 CST} DeviceType:120 DeviceID:1 Name:grafana IconID:0 UID: Hostname: IP: TeamID:1 DomainID:1 SubDomainID:0} {ChUpdated
AtBase:{UpdatedAt:2025-12-02 13:54:46.667215566 +0800 CST} DeviceType:120 DeviceID:20 Name:gpx_deepflow-gr IconID:0 UID: Hostname: IP: TeamID:1 DomainID:1 SubDomainID:0} {ChUpdatedAtBase:{UpdatedAt:2025-12-02 13:54:4
6.667215566 +0800 CST} DeviceType:120 DeviceID:2 Name:gpx_clickhouse_ IconID:0 UID: Hostname: IP: TeamID:1 DomainID:1 SubDomainID:0} {ChUpdatedAtBase:{UpdatedAt:2025-12-02 13:54:46.667215566 +0800 CST} DeviceType:120
 DeviceID:11 Name:node-cache IconID:0 UID: Hostname: IP: TeamID:1 DomainID:1 SubDomainID:0} {ChUpdatedAtBase:{UpdatedAt:2025-12-02 13:54:46.667215566 +0800 CST} DeviceType:120 DeviceID:21 Name:kube-controller IconID:
0 UID: Hostname: IP: TeamID:1 DomainID:1 SubDomainID:0} {ChUpdatedAtBase:{UpdatedAt:2025-12-02 13:54:46.667215566 +0800 CST} DeviceType:120 DeviceID:16 Name:kube-proxy IconID:0 UID: Hostname: IP: TeamID:1 DomainID:1
SubDomainID:0} {ChUpdatedAtBase:{UpdatedAt:2025-12-02 13:54:46.667215566 +0800 CST} DeviceType:120 DeviceID:12 Name:kube-scheduler IconID:0 UID: Hostname: IP: TeamID:1 DomainID:1 SubDomainID:0} {ChUpdatedAtBase:{Upda
tedAt:2025-12-02 13:54:46.667215566 +0800 CST} DeviceType:120 DeviceID:8 Name:kube-apiserver IconID:0 UID: Hostname: IP: TeamID:1 DomainID:1 SubDomainID:0} {ChUpdatedAtBase:{UpdatedAt:2025-12-02 13:54:46.667215566 +0
800 CST} DeviceType:120 DeviceID:3 Name:runsvdir IconID:0 UID: Hostname: IP: TeamID:1 DomainID:1 SubDomainID:0} {ChUpdatedAtBase:{UpdatedAt:2025-12-02 13:54:46.667215566 +0800 CST} DeviceType:120 DeviceID:13 Name:run
```



```
2025/12/02 20:17:51 /home/ubuntu/project/v70/deepflow/server/controller/tagrecorder/db_operator.go:75 pq: ON CONFLICT DO UPDATE command cannot affect row a second time
[16.034ms] [rows:0] INSERT INTO "ch_device" ("updated_at","devicetype","deviceid","hostname","ip","team_id","domain_id","sub_domain_id","name") VALUES ('2025-12-02 20:17:51.664',120,11,'','',1,1,0,'tuned'),('2025-12-02 20:17:51.664',120,3,'','',1,1,0,'app.py'),('2025-12-02 20:17:51.664',120,12,'',
2025-12-02 20:17:51.677 [ERRO] [tagrecorder] db_operator.go:77 add ch_device (keys: [{DeviceID:11 DeviceType:120} {DeviceID:3 DeviceType:120} {DeviceID:12 DeviceType:120} {DeviceID:12 DeviceType:120} {DeviceID:12 DeviceType:120} {DeviceID:12 DeviceType:120} {DeviceID:12 DeviceType:120} {DeviceID:12 DeviceType:120} {DeviceID:1
2025-12-02 20:17:51.680 [INFO] [trisolaris.kubernetes] kubernetes.go:104 refresh cache cluster_id completed
2025-12-02 20:17:51.680 [INFO] [trisolaris.kubernetes] kubernetes.go:86 refresh cache cluster_id started
2025-12-02 20:17:51.684 [INFO] [monitor] analyzer.go:264 analyzer ip() in vtap(master-V1) is invalid
2025-12-02 20:17:51.689 [INFO] [monitor] controller.go:292 vtap controller check end
2025-12-02 20:17:51.689 [INFO] [monitor] controller.go:419 az connection check start
2025-12-02 20:17:51.689 [INFO] [monitor] controller.go:301 vtap controller alloc start
2025-12-02 20:17:51.690 [INFO] [trisolaris.kubernetes] kubernetes.go:104 refresh cache cluster_id completed

2025/12/02 20:17:51 /home/ubuntu/project/v70/deepflow/server/controller/tagrecorder/db_operator.go:75 pq: ON CONFLICT DO UPDATE command cannot affect row a second time
[15.536ms] [rows:0] INSERT INTO "ch_gprocess" ("updated_at","chost_id","l3_epc_id","team_id","domain_id","sub_domain_id","id","name") VALUES ('2025-12-02 20:17:51.681',1,1,1,1,0,11,'tuned'),('2025-12-02 20:17:51.681',1,1,1,1,0,3,'app.py'),('2025-12-02 20:17:51.681',1,1,1,1,0,12,'from'),('2025-12-0
2025-12-02 20:17:51.694 [ERRO] [tagrecorder] db_operator.go:77 add ch_gprocess (keys: [{ID:11} {ID:3} {ID:12} {ID:12} {ID:12} {ID:12} {ID:12} {ID:12} {ID:12} {ID:12} {ID:12} {ID:12} {ID:2} {ID:4} {ID:13} {ID:5} {ID:14} {ID:15} {ID:19} {ID:16} {ID:17} {ID:1} {ID:6} {ID:20} {ID:20} {
2025-12-02 20:17:51.695 [INFO] [recorder] domain.go:295 [DomainName-k8s-1] domain data changed, refresh platform data
2025-12-02 20:17:51.695 [INFO] [trisolaris.refresh] refresh.go:68 refresh cache for trisolaris([10.233.70.86] []), orgID(1), dataTypes([platform_data])
2025-12-02 20:17:51.695 [GIN] 10.233.70.86 PUT /v1/caches/?org_id=1&type=platform_data 200 48.366µs
2025-12-02 20:17:51.697 [INFO] [monitor] controller.go:447 az connection check end
2025-12-02 20:17:51.701 [INFO] [monitor] analyzer.go:283 vtap analyzer check end
2025-12-02 20:17:51.701 [INFO] [monitor] analyzer.go:410 az connection check start
2025-12-02 20:17:51.702 [INFO] [recorder] domain.go:305 [DomainName-k8s-1] update domain synced_at: 2025-12-02 20:17:51
2025-12-02 20:17:51.702 [INFO] [recorder] domain.go:320 [DomainName-k8s-1] time now: 2025-12-02 20:17:51
```

8. PostgreSQL数据库的CHAR类型填充数据会保留空格，改成VARCHAR则不会保留

是否需要改造?



```sql
CREATE TABLE IF NOT EXISTS config_map (
    id                      SERIAL PRIMARY KEY,
    name                    VARCHAR(256) NOT NULL,
    data                    TEXT,
    data_hash               VARCHAR(64) DEFAULT '',
    pod_namespace_id        INTEGER NOT NULL,
    pod_cluster_id          INTEGER NOT NULL,
    epc_id                  INTEGER NOT NULL,
    az                      VARCHAR(64) DEFAULT '',
    region                  VARCHAR(64) DEFAULT '',
    sub_domain              VARCHAR(64) DEFAULT '',
    domain                  CHAR(64) NOT NULL,        zhengya, 2个月前 • feat:
    lcuuid                  CHAR(64) NOT NULL,
    synced_at               TIMESTAMP DEFAULT NULL,
    created_at              TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    updated_at              TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
);
TRUNCATE TABLE config_map;
CREATE INDEX config_map_data_hash_index ON config_map (data_hash);
CREATE INDEX config_map_domain_index ON config_map (domain);
COMMENT ON COLUMN config_map.data IS 'yaml format';

CREATE TABLE IF NOT EXISTS pod_group_config_map_connection (
    id                      SERIAL PRIMARY KEY,
    pod_group_id            INTEGER NOT NULL,
    config_map_id           INTEGER NOT NULL,
    sub_domain              VARCHAR(64) DEFAULT '',
    domain                  CHAR(64) NOT NULL,
    lcuuid                  CHAR(64) NOT NULL,
    created_at              TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    updated_at              TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
);
TRUNCATE TABLE pod_group_config_map_connection;
```