

1. Привести 2 - 3 практических примера использования БД ClickHouse в компаниях
2. Уточнить специфику проекта
3. Подготовить отчет со ссылками на ресурсы

## Привести 2 - 3 практических примера использования БД ClickHouse в компаниях

### 1. Сбор логов приложения в clickhouse

Сервис пишет логи в rsyslog > файлы > vector читает файлы с логами > структурирует файлы service.slog и requests.slog, остальные логи оставляет без обработки > отправляет в кафку, структурированные в один топик, остальные в другой > СН читает топики и пишет структурированные в таблицу logs.service\_logs остальные в logs.misc\_logs

Зачем структурировать данные

У структурирования данных - две причины:

Проще найти данные, потому-что не надо искать поле по всему телу лога

Данные гораздо лучше упаковываются, потому-что похожие данные лежат рядом. Это экономит как место на диске, так и ускоряет обработку

### 2. Интеграция с postgres

запросы в postgres из кх - движок бд/таблицы PostgreSQL

```
create table postgres_connect (user, pwd) engine = PostgreSQL(...)
```

```
select from postgres_connect.vacancy join vacancyResponseDistr
```

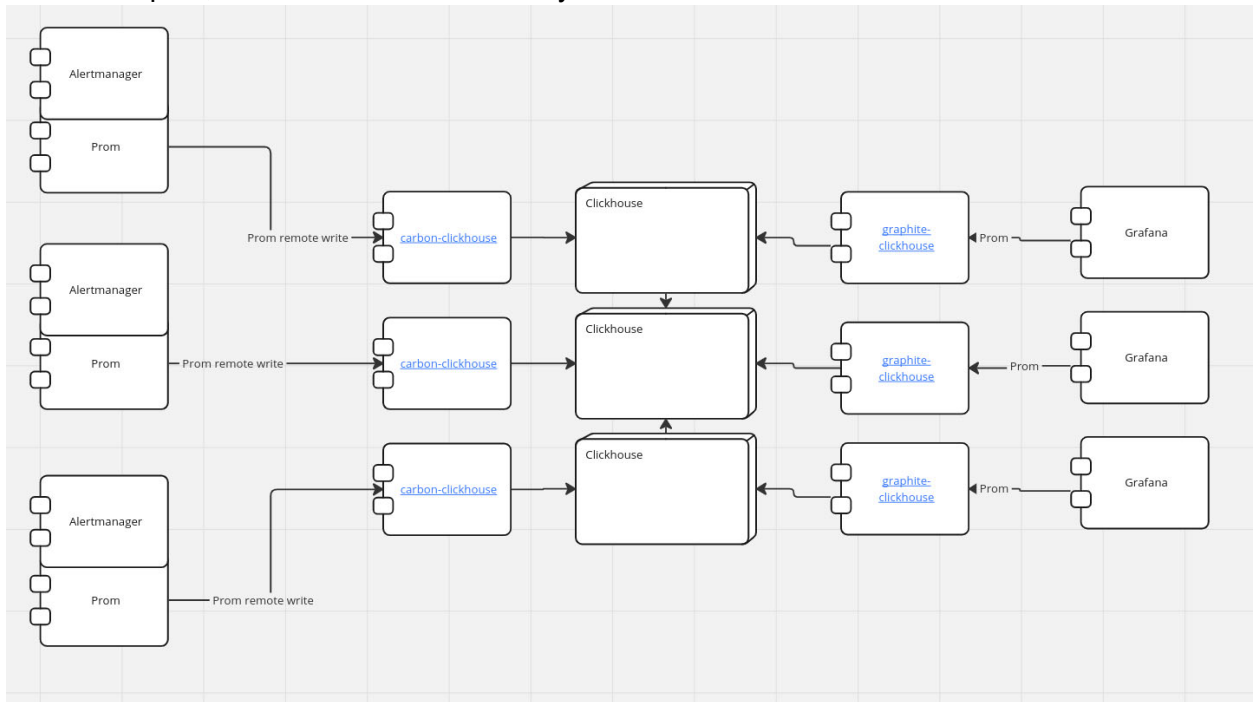
репликация таблиц из postgres - движок бд/таблиц MaterializedPostgreSQL (experimental)

гарантируется транзакционная целостность, т.е. в каком порядке данные попадали в постгрес, в таком же они попадут в кх

есть механизм для репликации DDL

создается еще одна таблица, а потом происходит подмена

### 3. Мониторинг, в качестве базы используется clickhouse



кластер мониторинга состоит из следующих компонентов:

1. Clickhouse. Колоночная БД. Использует ts-optimized движок [GraphiteMergeTree](#) и [MergeTree](#). Не используем дистрибуцию, только репликацию. На каждом из 3-х инстансов ClickHouse хранится полная копия БД с метриками. Структура таблиц взята [отсюда](#)
2. [Carbone-clickhouse](#) - один из компонентов [graphite-go](#)  
Позволяет получать метрики в различных форматах/протоколах (graphite, prometheus, etc) и сохранять их для долгосрочного хранения в Clickhouse
3. [Graphite-clickhouse](#) - один из компонентов [graphite-go](#)  
Позволяет читать метрики из Clickhouse и отдавать их по различным протоколам (Graphite, Prometheus, etc)
4. Prometheus - собирает метрики и пишет их через remote-write в Clickhouse через carbone-clickhouse
5. Prometheus AlertManager - используется для алертинга
6. Grafana - читает метрики из ClickHouse через graphite-clickhouse по протоколу Prometheus

Дедупликация метрик при записи осуществляется за счет движка MergeTree в Clickhouse. При отправке алертов AlertManager использует внутренний механизм дедупликации алертов.

## Уточнить специфику проекта

Сервис трудоустройства.

## Подготовить отчет со ссылками на ресурсы

GraphiteMergeTree - <https://clickhouse.com/docs/ru/engines/table-engines/mergetree-family/graphitemergetree>

MergeTree - <https://clickhouse.com/docs/ru/engines/table-engines/mergetree-family/mergetree>

**carbon-clickhouse** - <https://github.com/go-graphite/carbon-clickhouse>

graphite-go - <https://github.com/go-graphite/>

Graphite-clickhouse - <https://github.com/go-graphite/graphite-clickhouse/>