

ArenadataTM Quick Marts

Версия - master

Прокси-сервер baje

Оглавление

1 Установка	3
2 Настройка	4

Прокси-сервер *baje* позволяет настраивать доступ для аналитиков до **ClickHouse** и балансировку запросов между хостами.

Глава 1

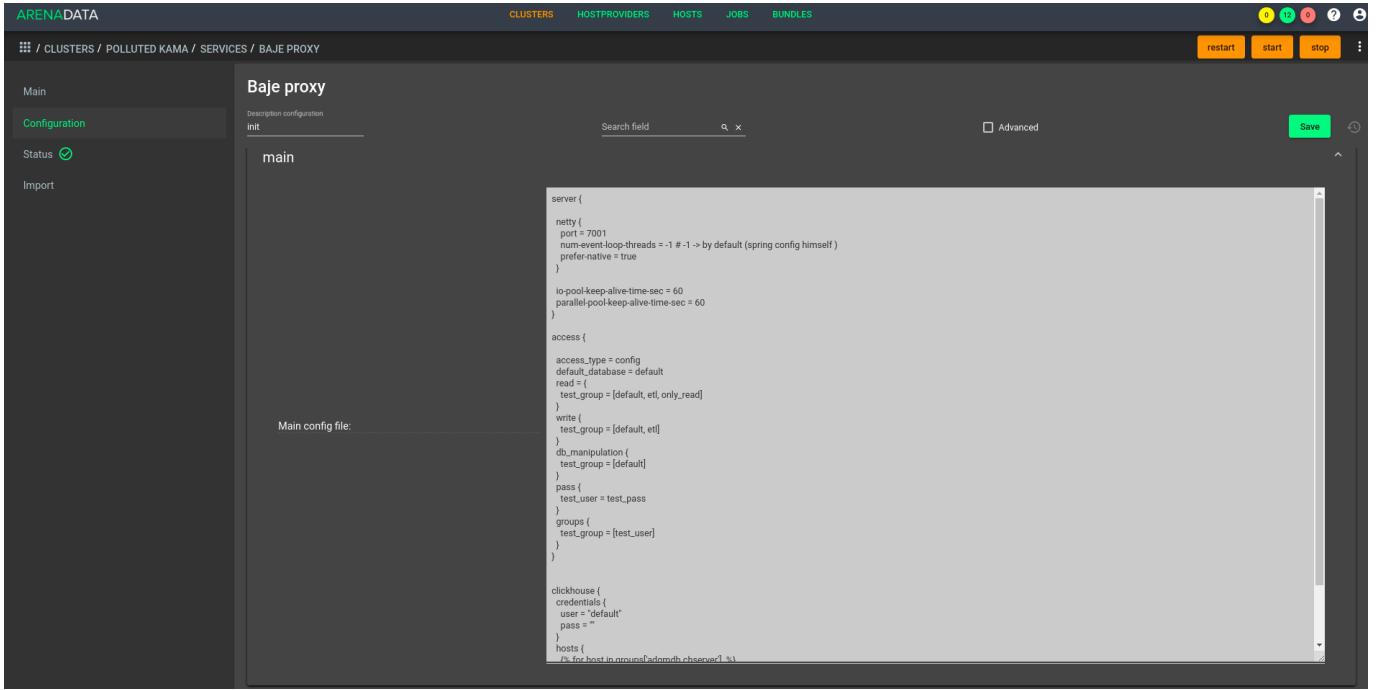
Установка

Прокси-сервер *baje* входит в список сервисов **ADQM** и устанавливается вместе с остальными компонентами.

Глава 2

Настройка

На данный момент конфигурация *baje* доступна только в формате файла (Рис.2.1).



```
server {
    netty {
        port = 7001
        num-event-loop-threads = -1 # -1 -> by default (spring config himself)
        prefer-native = true
    }
    io-pool-keep-alive-time-sec = 60
    parallel-pool-keep-alive-time-sec = 60
}
access {
    access_type = config
    default_database = default
    read = [
        test_group = [default, etl, only_read]
    ]
    write = [
        test_group = [default, etl]
    ]
    db_manipulation = [
        test_group = [default]
    ]
    pass = [
        test_user = test_pass
    ]
    groups = [
        test_group = [test_user]
    ]
}
clickhouse {
    credentials {
        user = "default"
        pass = ""
    }
    hosts = [
        % for host in groups["adminchdb.chserver1"] %
    ]
}
```

Рис.2.1.: Конфигурация *baje*

- **server.netty.port** – http-port, на котором приложение слушает входные запросы;
 - **server.netty.num-event-loop-threads** – количество потоков, обрабатывающих входной поток запросов. По умолчанию -1 – *baje* сам решает сколько необходимо потоков;
 - **server.netty.prefer-native** – опция включения epoll/kqueue, по умолчанию **true**.
-
- **access.access_type** – опция определяет, откуда берет списки пользователей и групп. Существует два варианта: **config** и **ldap**.

При `access_type=config` необходимо описать следующие параметры:

- `access.default_database` – база данных в ClickHouse, к которой обращается `baje` по умолчанию;
- `access.read` – список групп, которым открыт доступ на чтение до определенных баз. Для каждой группы описывается список баз данных.

Пример:

```
read = {
  group1 = [default, etl, only_read]
  data_engineer_1 = [etl, only_read]
  analytics = [only_read]
}
```

- “`access.write`” – список групп, которым открыт доступ на запись. Под записью понимаются операции `insert`, `alter`, `optimization` и прочие операции с таблицами.

Пример:

```
write {
  group1 = [default, etl]
  data_engineer_1 = [default, etl]
}
```

- `access.db_manipulation` – список групп, которым открыт доступ на создание и удаление баз данных.

Пример:

```
db_manipulation {
  admins = [default, etl, only_read]
}
```

- `access.pass` – словарь с пользователями и паролями.

Пример:

```
pass {
  default_user = test_pass
  master_admin_1 = admin_pass
}
```

- `access.groups` – описывает группы и входящих в них пользователей.

Пример:

```
groups {
  group1 = [default_user, master_admin_1]
  data_engineer_1 = [master_admin_1]
  admins = [master_admin_1]
}
```

При `access_type=ldap` необходимо описать следующие параметры:

- `access.ldap.host` – ip-адрес или имя хоста для подключения к ldap;
- `access.ldap.port` – порт для подключения к ldap;

-
- `access.ldap.user_search_dn` – dn в ldap, в котором производится поиск пользователей;
 - `access.ldap.user_object_type` – тип объекта для пользователя;
 - `access.ldap.password_attr` – атрибут объекта пользователя, в котором хранится пароль. Пароль внутри base хранится в виде хеша;
 - `access.ldap.update_delay_sec` – задержка между обновлениями списка пользователей;
 - `access.ldap.ttl_for_ldap_cache_sec` – время, через которое хеш пароля пользователя удаляется из кеша base при его исчезновении из ldap;
 - `access.ldap.admin_dn` – dn в ldap с пользователем для подключения ldap-клиента base;
 - `access.ldap.admin_pass` – пароль для подключения к `admin_dn`;
 - `access.ldap.group_object_class` – тип объекта для групп пользователей в ldap;
 - `access.ldap.base_dn` – `base_dn` в ldap.
 - `access.default_database` – база данных в ClickHouse, к которой обращается base по умолчанию;
 - `access.read` – список групп, которым открыт доступ на чтение до определенных баз. Для каждой группы описывается список баз данных.

Пример:

```
read = {
    group1 = [default, etl, only_read]
    data_engineer_1 = [etl, only_read]
    analytics = [only_read]
}
```

- `access.write` – список групп, которым открыт доступ на запись. Под записью понимаются операции `insert`, `alter`, `optimization` и прочие операции с таблицами.

Пример:

```
write {
    group1 = [default, etl]
    data_engineer_1 = [default, etl]
}
```

- `access.db_manipulation` – список групп, которым открыт доступ на создание и удаление баз данных.

Пример:

```
db_manipulation {
    admins = [default, etl, only_read]
}
```

-
- `clickhouse.credentials.user` – пользователь, под которым base шлет запросы в ClickHouse;
 - `clickhouse.credentials.pass` – пароль для `clickhouse.credentials.user`;
 - `clickhouse.credentials.hosts` – список хостов ClickHouse с распределением нагрузки.

Пример:

```
hosts {
    "localhost:8123" = 0.5
    "host_1:8123" = 2.23
}
```

Цифры 0.5 и 2.23 следует понимать следующим образом – это веса, на основе которых выбирается хост для каждого нового запроса: 0.5 – 18% трафика, 2.23 – 82% трафика.