

ИНСТРУКЦИЯ ПО УСТАНОВКЕ И ИСПОЛЬЗОВАНИЮ МОДУЛЯ PMCONTROLLING: PMDATABASE

Дата: 01.04.2025
Версия: 2.0

2025 г.

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ.....	2
1. ВВЕДЕНИЕ	3
2. УСТАНОВКА И НАСТРОЙКА СУБД POSTGRESQL.....	3
3. РАЗВЕРТЫВАНИЕ БАЗЫ ДАННЫХ.....	4
4. УСТАНОВКА И НАСТРОЙКА DOCKER	4
5. УСТАНОВКА И НАСТРОЙКА CLICKHOUSE	5
6. УСТАНОВКА И НАСТРОЙКА PMDATABASE	6
7. РАБОТА В ИНТЕРФЕЙСЕ АДМИНИСТРАТОРА	8

1. ВВЕДЕНИЕ

Настоящее руководство описывает процесс установки и настройки модуля PMControlling: PMDataBase на операционную систему GNU Linux.

Подразумевается, что все необходимые компоненты системы размещены в директории /opt.

2. УСТАНОВКА И НАСТРОЙКА СУБД POSTGRESQL

Установить и запустить СУБД PostgreSQL 15:

```
dnf install -y postgresql15-server postgresql15-contrib  
postgresql-15-setup initdb  
systemctl enable postgresql-15
```

Отредактировать файл /var/lib/pgsql/15/data/postgresql.conf

Редактируем listen_addresses = "*"

#JIT

jit = off # allow JIT compilation

locale

datestyle = 'iso, mdy'

timezone = 'Etc/UTC'

lc_messages = 'en_US.UTF-8' # locale for system error message

lc_monetary = 'en_US.UTF-8' # locale for monetary formatting

lc_numeric = 'en_US.UTF-8' # locale for number formatting

lc_time = 'en_US.UTF-8' # locale for time formatting

#connections

max_connections = 1000

Отредактировать файл /var/lib/pgsql/15/data/pg_hba.conf, заменив содержимое на:

"local" is for Unix domain socket connections only

local all all peer

IPv4 local connections:

ssh tunnel for PostgreSQL DBAs

host all all all md5

IPv6 local connections:

host all all ::1/128 ident

Allow replication connections from localhost, by a user with the

replication privilege.

local replication all peer

host replication all all ident

```
host replication all ::1/128 ident
```

Перезапустить PostgreSQL:

```
systemctl restart postgresql-15
```

```
systemctl status postgresql-15
```

3. РАЗВЕРТЫВАНИЕ БАЗЫ ДАННЫХ

Создать пользователя, создать базу данных, загрузить дамп базы данных:

Распаковать архив db_pmc.tar.gz, расположенный в /opt/db_dumps

```
tar xvf db_pmc.tar.gz
```

```
chown postgres /opt/db_dumps/db_pmc.sql
```

```
su - postgres
```

```
psql
```

```
CREATE USER pmduser WITH password 'password';
```

```
ALTER USER pmduser SUPERUSER;
```

```
CREATE EXTENSION IF NOT EXISTS "dblink";
```

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

```
CREATE EXTENSION IF NOT EXISTS "uuid-osspl";
```

```
CREATE DATABASE db_pmd WITH OWNER pmduser;
```

```
CREATE DATABASE db_pmc WITH OWNER pmduser;*
```

```
exit
```

```
psql db_pmc < /opt/db_dumps/db_pmc.sql
```

Exit

*В данной инструкции базы db_pmc и db_pmd, установленные на одном сервере, в нагруженной среде, для корректной работы должны использоваться отдельные сервера postgresql-15.

4. УСТАНОВКА И НАСТРОЙКА DOCKER

Для RedOS убедиться, что в файле nano /etc/yum.repos.d/RedOS-Updates.repo установлено значение enabled=1.

```
yum install docker-ce docker-compose -y
```

```
sudo systemctl start docker
```

```
sudo systemctl enable docker
```

5. УСТАНОВКА И НАСТРОЙКА CLICKHOUSE

Распаковать архив clickhouse.tar.gz, расположенный в /opt/clickhouse/

Выполнить команды по установке:

```
sh clickhouse-common-static-22.8.12.45/install/doinst.sh
```

```
sh clickhouse-common-static-dbg-22.8.12.45/install/doinst.sh
```

```
sh clickhouse-server-22.8.12.45/install/doinst.sh
```

```
sh clickhouse-client-22.8.12.45/install/doinst.sh
```

Enter password for default user: оставляем пустым, enter

```
systemctl enable clickhouse-server.service
```

```
systemctl start clickhouse-server.service
```

Настроить в /etc/clickhouse-server/config.xml

```
<listen_host>0.0.0.0</listen_host>
```

```
<max_connections>4096</max_connections>
```

```
<keep_alive_timeout>600</keep_alive_timeout>
```

```
<max_concurrent_queries>0</max_concurrent_queries>
```

```
<max_server_memory_usage>0</max_server_memory_usage>
```

```
<default_session_timeout>3600</default_session_timeout>
```

Настроить в /etc/clickhouse-server/users.xml

Показана иерархия, куда положить свойство:

```
<clickhouse>
```

```
  <profiles>
```

```
    <default>
```

```
      ...
```

```
        <max_query_size>0</max_query_size>
```

```
      ...
```

```
    </default>
```

```
  ...
```

```
</profiles>
```

```
</clickhouse>
```

```
systemctl restart clickhouse-server.service
```

6. УСТАНОВКА И НАСТРОЙКА PMDATABASE

Распаковать архив pmc_image.tar.gz, расположенный в /opt/pmc_image
tar xvf pmc_image.tar.gz

Выполнить / load-release-images.sh для загрузки docker-образов в систему.

```
Loaded image: rabbitmq:3.9-management-alpine
[load] file redis.7.0.2.tar as image redis:7.0.2
08249ce7456a: Loading layer [=====>] 83.9MB/83.9MB
5659b3a1146e: Loading layer [=====>] 338.4kB/338.4kB
cf3ae502d7fa: Loading layer [=====>] 4.161MB/4.161MB
4ca33072d026: Loading layer [=====>] 32MB/32MB
58bcc523fc92: Loading layer [=====>] 2.048kB/2.048kB
be56018ff479: Loading layer [=====>] 4.096kB/4.096kB
Loaded image: redis:7.0.2
```

Распаковать архив demo-runtime.tar.gz, расположенный в /opt/demo-runtime

Выполнить команду, подставив вместо 10.0.1.444 IP адрес своего сервера с postgresql и PMDataBase.

```
find ./ -type f -exec sed -i 's/10\0\1\248/10.0.1.444/g' {} +
```

Выполнить команду sh runtime-cli.sh install && sh runtime-cli.sh start && sh runtime-cli.sh inspect

Получите в выводе список запущенных docker-контейнеров

```
inspect] CONTAINERS LIST -----
CONTAINER ID    STATUS    IMAGE
documentsmicroservice 0d424bfce707 Up 2 minutes cs/pmc/documentsmicroservice:5.1
servicepmcintegration 2931cc773ecc Up 2 minutes cs/pmc/integration:5.1
servicesafety 18546a705e8e Up 2 minutes cs/pmc/safety:5.1
servicepmcmonolith 65a0c358f57b Up 2 minutes cs/pmc/monolith:5.1
redis 97355089443d Up 2 minutes redis:7.0.2
rabbitmq 8dcfebc7134d Up 2 minutes rabbitmq:3.9-management-alpine
```

Распаковать архив pmd_image.tar.gz, расположенный в /opt/pmd_image
tar xvf pmd_image.tar.gz

Выполнить./ load-release-images.sh для загрузки docker-образов в систему.

Распаковать архив demo_pmd.tar.gz, расположенный в /opt/demo_pmd

Выполнить команду, подставив вместо 10.0.1.444 IP адрес своего сервера с postgresql и PMDataBase.

```
find ./ -type f -exec sed -i 's/10\0\1\248/10.0.1.444/g' {} +
```

Выполнить из /opt/demo_pmd команды:

```
docker-compose up -d
```

```
docker-compose ps
```

Убедиться, что контейнеры запущены:

NAME	COMMAND	SERVICE	STATUS
demo_pmd-backend-etl-service-1	"docker-entrypoint.s..."	backend-etl-service	running
demo_pmd-backend-master-service-1	"docker-entrypoint.s..."	backend-master-service	running
demo_pmd-backend-read-model-service-1	"docker-entrypoint.s..."	backend-read-model-service	running
demo_pmd-front-storybook-1	"/docker-entrypoint..."	front-storybook	running
demo_pmd-pmc-bi-dash-1	"/docker-entrypoint..."	pmc-bi-dash	running
demo_pmd-pmc-bi-remote-1	"/docker-entrypoint..."	pmc-bi-remote	running

Распаковать архив superset.tar.gz

Перейти в распакованную директорию и выполнить импорт образов:

```
docker load -i ./images/hosts_ssl
```

```
docker load -i ./images/postgres
```

```
docker load -i ./images/redis
```

```
docker load -i ./images/superset
```

Для запуска выполнить скрипт: sh start_CS_compose.sh

```
docker-compose -f docker-compose-image-tag.yml ps
```

Убедиться, что контейнеры запущены:

NAME	IMAGE	COMMAND	SERVICE	CREATED	STATUS
host_ssl	h.contralsystems.ru/cs/core/frontend-hosts-generic-next:1.63.5	"/docker-entrypoint..."	host_ssl	12 days ago	Up 12 days
superset_app	apachesuperset.docker.scarf.sh/apache/superset:4.1.1	"/app/docker/docker..."	superset	13 minutes ago	Up 12 minutes
superset_cache	redis:7	"docker-entrypoint.s..."	redis	5 days ago	Up 5 days
superset_db	postgres:13	"docker-entrypoint.s..."	db	5 days ago	Up 5 days
superset_worker	apachesuperset.docker.scarf.sh/apache/superset:4.1.1	"/app/docker/docker..."	superset-worker	13 minutes ago	Up 12 minutes
superset_worker_beat	apachesuperset.docker.scarf.sh/apache/superset:4.1.1	"/app/docker/docker..."	superset-worker-beat	13 minutes ago	Up 12 minutes

Распаковать архив airflow.tar.gz

Перейти в распакованную директорию и выполнить импорт образов:

```
docker load -i images/postgres
```

```
docker load -i images/redis
```

```
docker load -i images/airflow
```

Выполнить запуск:

```
docker compose up -d
```

Убедиться, что контейнеры запущены:

NAME	IMAGE	COMMAND	SERVICE	CREATED	STATUS
airflow-airflow-init-1	apache/airflow:2.10.5	"/bin/bash -c 'if [..."	airflow-init	4 minutes ago	Exited (0) 56 seconds ago
airflow-airflow-scheduler-1	apache/airflow:2.10.5	"/usr/bin/dumb-init -"	airflow-scheduler	About a minute ago	Up 55 seconds (healthy)
airflow-airflow-triggerer-1	apache/airflow:2.10.5	"/usr/bin/dumb-init -"	airflow-triggerer	About a minute ago	Up 55 seconds (health: starting)
airflow-airflow-webserver-1	apache/airflow:2.10.5	"/usr/bin/dumb-init -"	airflow-webserver	About a minute ago	Up 55 seconds (health: starting)
airflow-airflow-worker-1	apache/airflow:2.10.5	"/usr/bin/dumb-init -"	airflow-worker	About a minute ago	Up 55 seconds (healthy)
airflow-postgres-1	postgres:13	"docker-entrypoint.s..."	postgres	4 minutes ago	Up 4 minutes (healthy)
airflow-redis-1	redis:7.2-bookworm	"docker-entrypoint.s..."	redis	4 minutes ago	Up 4 minutes (healthy)

Распаковать архив monitoring.tar.gz

Перейти в распакованную директорию и выполнить импорт образов:

```
docker load -i images/prometheus
```

```
docker load -i images/alertmanager
```

```
docker load -i images/grafana
```

```
docker load -i images/node-exporter
```

Выполнить запуск:

```
docker compose up -d
```

Убедиться, что контейнеры запущены:

NAME	IMAGE	COMMAND	SERVICE	CREATED	STATUS
monitoring-alertmanager-1	prom/alertmanager	"/bin/alertmanager -"	alertmanager	18 seconds ago	Up Less than a second
monitoring-cadvisor-1	gcr.io/cadvisor/cadvisor	"/usr/bin/cadvisor -"	cadvisor	18 seconds ago	Up 8 seconds (health: starting)
monitoring-grafana-1	grafana/grafana	"/run.sh"	grafana	18 seconds ago	Up 6 seconds
monitoring-node-exporter-1	quay.io/prometheus/node-exporter:latest	"/bin/node_exporter -"	node-exporter	18 seconds ago	Up 8 seconds
monitoring-prometheus-1	prom/prometheus:v2.36.2	"/bin/prometheus --c_"	prometheus	18 seconds ago	Up 7 seconds

Распаковать архив keycloak.tar.gz

Перейти в распакованную директорию и выполнить импорт образов:

```
docker load -i images/keycloak
```

Создать БД в postgres:

```
create user keycloak with password 'keycloakpwd';
```

```
createdb -O keycloak db_keycloak
```

```
create schema keycloak;
```

Выполнить запуск:

```
docker-compose up -d
```

Убедиться, что контейнеры запущены:

NAME	IMAGE	COMMAND	SERVICE	CREATED	STATUS
keycloak	quay.io/keycloak/keycloak:26.0.5	"/opt/keycloak/bin/k_"	keycloak	2 minutes ago	Up 2 minutes (unhealthy)

7. РАБОТА В ИНТЕРФЕЙСЕ АДМИНИСТРАТОРА

Перейти в браузере на IP адрес Системы PMControlling

http://**.IP.**:4202/ и ввести логин / пароль, логин и пароль необходимо запросить у поставщика Модуля



Перейти в список OLAP-кубов.



Для загрузки нового OLAP-куба необходимо нажать кнопку «импортировать» и выбрать zip-архив для загрузки.

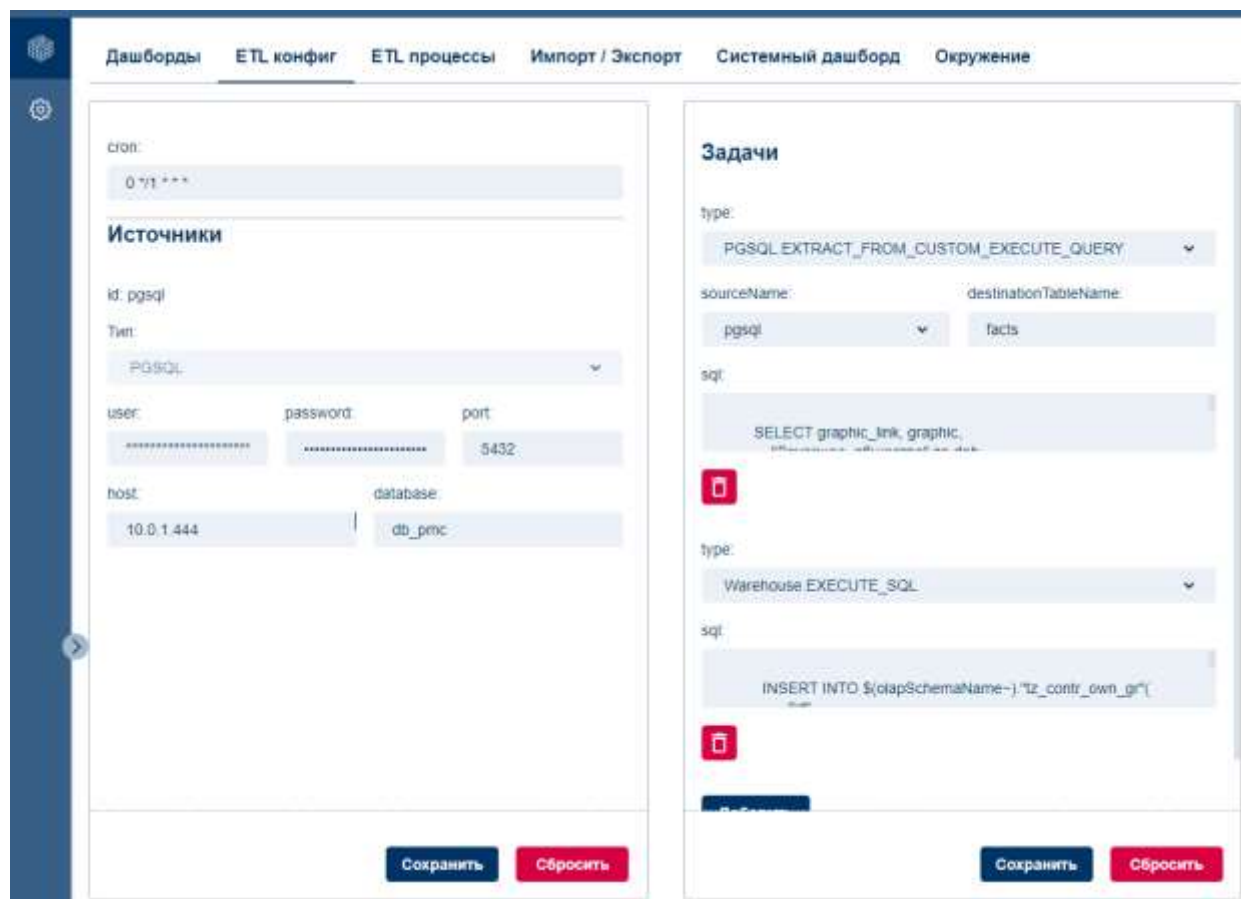
Для настройки существующего нажать ЛКМ на название куба. Будут доступны следующие разделы:

Дашборд – графическое отображение текущего OLAP-куба.

ETL конфиг – настройка источника данных, из БД db_pmc.

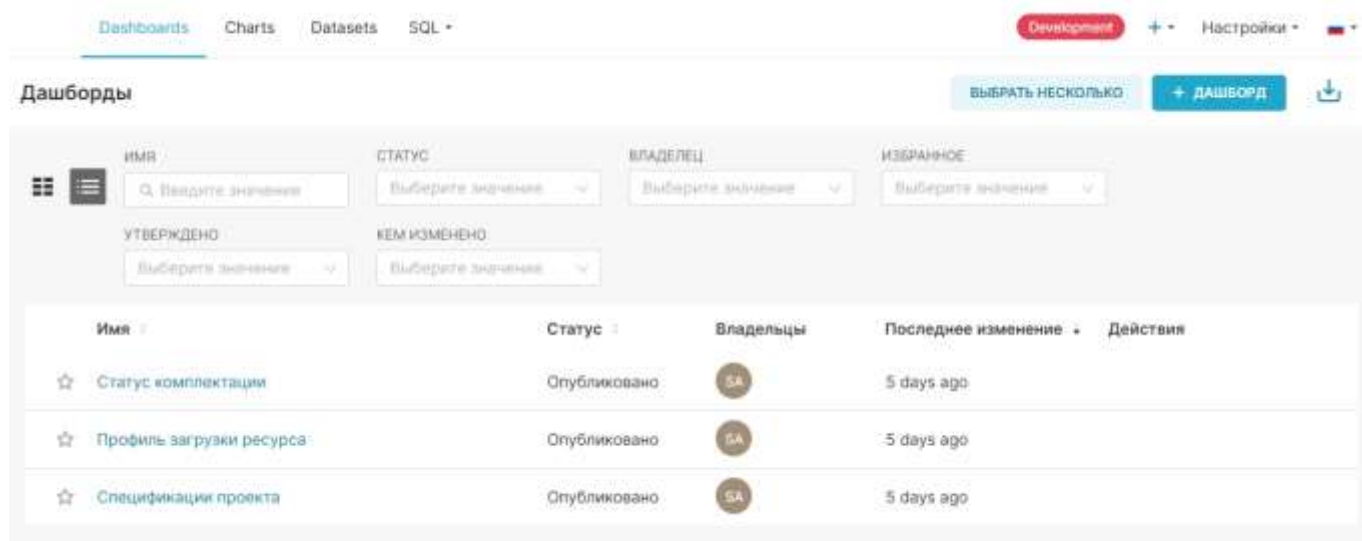
ETL процессы – запуск нового расчёта OLAP-куба из данных БД db_pmc.

Импорт / Экспорт – позволяет выгрузить zip-архив с настроенным OLAP-кубом или загрузить новую версию.



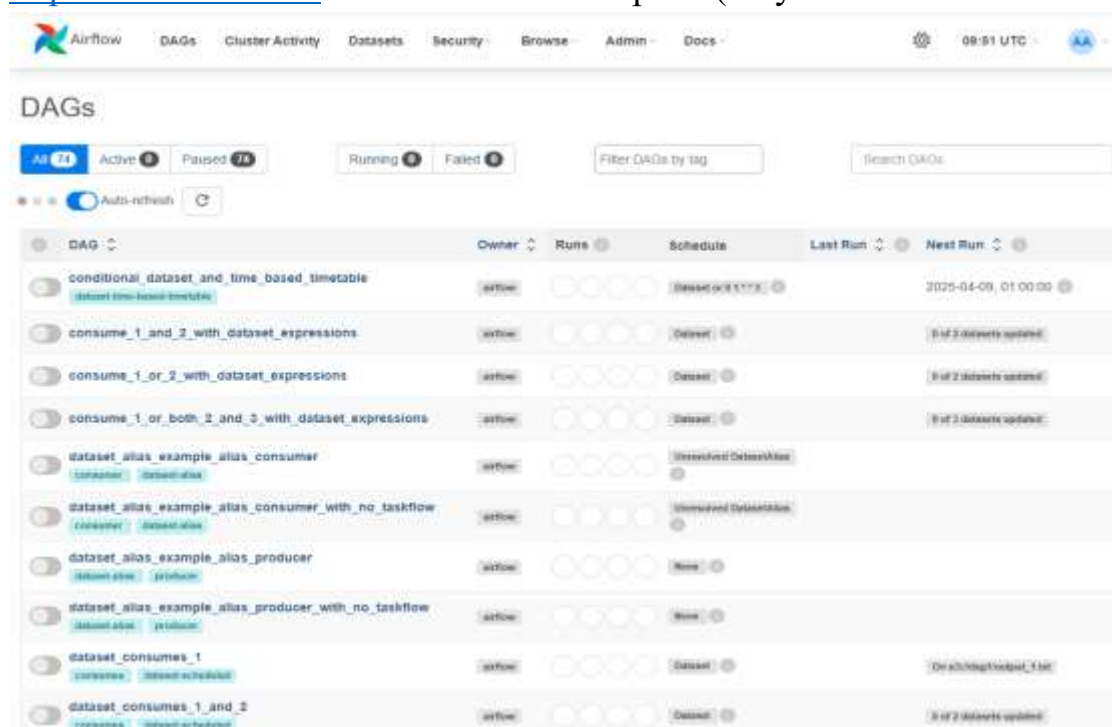
Перейти в браузере на IP адрес Системы PMControlling

http://**.IP.**:8088/ и ввести логин / пароль (по умолчанию admin/admin)



Перейти в браузере на IP адрес Системы PMControlling

http://**.IP.**:8080/ и ввести логин / пароль (по умолчанию airflow/airflow)



Перейти в браузере на IP адрес Системы PMControlling

http://**.IP.**:8080/ и ввести логин / пароль (по умолчанию admin/admin)

