

ГОСУДАРСТВЕННАЯ КОРПОРАЦИЯ
ПО АТОМНОЙ ЭНЕРГИИ «РОСАТОМ»

**Акционерное общество
«Концерн по производству электрической и тепловой
энергии на атомных станциях»
(АО «Концерн Росэнергоатом»)**

**Технический автономный модуль комплексной системы
централизованного оперативного планирования и
управления производством для передачи данных
предиктивной аналитики**

Совершенствование ПО

на 9 листах

Москва 2024

Оглавление

1. Общая информация.....	3
2. Apache NiFi.....	4
3. БДРВ ClickHouse.....	5
4. СУБД Postgres Pro.....	6
5. Apache Zookeeper.....	7
6. HAProxy.....	8
7. Grafana.....	8
8. Prometheus.....	9

1. Общая информация

Технический автономный модуль комплексной системы централизованного оперативного планирования и управления производством для передачи данных предиктивной аналитики предназначен для решения задач сбора данных временных рядов из технологических систем энергоблоков АЭС, обеспечения долговременного хранения данных и возможности предоставления срезов по ним в иные системы по запросу, состоит из следующих компонентов:

- ClickHouse
- Apache NiFi
- Zookeeper
- Postgres Pro Enterprise
- Grafana
- Prometheus

В данной документации рассматривается совершенствование компонентов Технического автономного модуля комплексной системы централизованного оперативного планирования и управления производством для передачи данных предиктивной аналитики.

Доработка Технического автономного модуля комплексной системы централизованного оперативного планирования и управления производством для передачи данных предиктивной аналитики осуществляется в соответствии с лицензией OpenSource компонентов. Все изменения вносятся в первую очередь на тестовом контуре во избежание потери работоспособности продуктивного контура. Тестовый контур должен полностью дублировать продуктивный контур.

2. Apache NiFi

В качестве совершенствования компонента Apache NiFi рекомендуется обновление версии:

1. Перед обновлением версии необходимо сделать резервную копию всех пользовательских NAR-файлов.

2. Получить новую версию ПО с официального сайта Apache NiFi:
nifi.apache.org/download.html

3. Создать новый каталог в каталоге с текущей версией NiFi и распаковать в него архив с новой версией ПО.

4. Перед началом обновления необходимо остановить все процессоры и дождаться обработки Flow-файлов, присутствующих в потоке. После завершения всех потоков остановить службу NiFi командой: `systemctl stop nifi`.

5. Сконфигурировать файлы `authorizers.xml` и `login-identity-providers.xml` в соответствии с текущей версией Apache NiFi.

6. Сконфигурировать файл `nifi.properties` в соответствии с текущей версией Apache NiFi.

7. Сконфигурируйте файлы `bootstrap-messages-services.xml` и `bootstrap.conf` в соответствии с текущей версией.

8. Добавьте пользовательский журнал `logback.xml` в новую версию NiFi.

9. Скопируйте файл `state-management.xml` из текущей версии в новую версию NiFi.

10. Скопируйте содержимое каталогов текущей версии в новую:
`./conf/flow.xml.gz`

11. Скопировать `flow.xml.gz` из каталога архива текущей версии в новую версию NiFi.

12. Повторить все операции для каждой ноды кластера NiFi.

13. Запустить все узлы NiFi. Убедиться в работоспособности.

3. БДРВ ClickHouse

В качестве совершенствования компонента ClickHouse рекомендуется обновление версии:

1. Перед началом обновления версии необходимо сделать резервную копию БДРВ ClickHouse.

2. Получить новую версию ПО с официального сайта ClickHouse:

<https://packages.clickhouse.com/deb/pool/main/c/clickhouse/>

При установке ClickHouse не из deb-пакетов необходимо получить следующие файлы:

- clickhouse-client_xx.yy.a.b_all.deb
- clickhouse-common-static-dbg_xx.yy.a.b_amd64.deb
- clickhouse-common-static_xx.yy.a.b_amd64.deb
- clickhouse-server_xx.yy.a.b_all.deb,

где xx.yy.a.b – это номер версии ClickHouse.

3. Выполнить установку новой версии ClickHouse, используя команды (для ClickHouse, установленного не из deb-пакетов):

```
$ sudo apt-get update
$ sudo apt-get install clickhouse-server=xx.yy.a.b clickhouse-client=xx.yy.a.b
clickhouse-common-static=xx.yy.a.b
$ sudo service clickhouse-server restart
```

4. Выполнить установку новой версии ClickHouse, используя команды (для ClickHouse, установленного из deb-пакетов):

```
$ sudo apt-get update
$ sudo apt-get install clickhouse-client clickhouse-server
$ sudo service clickhouse-server restart
```

5. Убедиться в работоспособности:

- Проверить состояние службы выполнив команду `service clickhouse server status`. Убедиться, что служба запущена.
- Проверить лог-файл, выполнив команду `nano/db/clickhouse/logs/clickhouse-server.log`. Убедиться в отсутствии ошибок.
- Проверить таблицы `system.query_log`, `system.crash_log` и `system.errors` на наличие ошибок.

6. Повторить все операции для каждой ноды кластера ClickHouse

4. СУБД Postgres Pro

В качестве совершенствования компонента Postgres Pro рекомендуется обновление версии:

1. Перед началом обновления версии необходимо сделать резервную копию PostgreSQL:

```
su – postgres
cd /tmp
pg_dumpall > my_pg_backup.sql
```

2. Получить новую версию PostgreSQL с официального сайта: <https://www.postgresql.org/download>

3. Остановить сервер PostgreSQL:
systemctl stop postgresql

4. Выполнить обновление PostgreSQL:

```
/postgresql/[new_version]/bin/pg_upgrade \
--old-datadir=/postgresql/[old_version]/main \
--new-datadir=/ postgresql/[new_version]/main \
--old-bindir=/postgresql/[old_version]/bin \
--new-bindir=/postgresql/[new_version]/bin \
--old-options '-c config_file=/postgresql/[old_version]/main/postgresql.conf' \
--new-options '-c config_file=/postgresql/[new_version]/main/postgresql.conf' \
```

5. При возникновении ошибки postmaster, необходимо принудительно остановить его:

```
/postgresql/[old_version]/bin/pg_ctl stop -pgdata=/postgresql/[old_version]/main -
mode=fast
/postgresql/[new_version]/bin/pg_ctl stop -pgdata=/postgresql/[new_version]/main -
mode=fast
```

6. После остановки postmaster повторить п.4

7. После завершения процесса обновления возможна ошибка шифрования в файле pg_hba.conf. Необходимо изменить тип шифрования в соответствии с версией PostgreSQL

8. Запустить кластер в новой версии PostgreSQL:

```
/postgresql/[new_version]/bin/pg_ctl start -pgdata=/postgresql/[new_version]/main
```

9. Запустить сервер PostgreSQL:

```
systemctl start postgresql
```

10. Проверить состояние активности версий:

```
/postgresql/[new_version]/bin/pg_ctl status --pgdata=/postgresql/[new_version]/main
/postgresql/[old_version]/bin/pg_ctl status --pgdata=/postgresql/[old_version]/main
```

11. Проверить версию PostgreSQL из самой СУБД:

```
show server_version;
```

12. После успешного обновления, удалить старую версию кластера (при необходимости):

```
/postgresql/[old_version]/bin/vacuumdb --all --analyze-in-stages
```

5. Apache Zookeeper

В качестве совершенствования компонента Apache Zookeeper рекомендуется обновление версии:

1. Получить новую версию Apache Zookeeper с официального сайта:

<https://nifi.apache.org/download.html>

2. Определить лидера кворума:

```
su -zookeeper
echo stat | nc node1 2181 | grep Mode
echo stat | nc node2 2181 | grep Mode
exit
```

3. Остановить сервер Zookeeper, который находится в режиме follower:

```
systemctl stop zookeeper
```

4. Изменить версию Zookeeper:

```
su -zookeeper -c "source /hdp/current/zookeeper-server/conf/zookeeper-
env.sh ; env ZOOCFGDIR=/hdp/current/zookeeper-server/conf
ZOOCFG=zoo.cfg /hdp/current/zookeeper-server/bin/zkServer.sh stop"
hdp-select set zookeeper-server [version]-$BUILD
su -zookeeper -c "source /hdp/current/zookeeper-server/conf/zookeeper-
env.sh ; env ZOOCFGDIR=/hdp/current/zookeeper-server/conf
ZOOCFG=zoo.cfg /hdp/current/zookeeper-server/bin/zkServer.sh start"
```

5. Проверьте обновление версии на сервере follower. Убедитесь, что кворум восстановлен

6. Создайте новую znode на обновленной ноде Zookeeper.

```
/hdp/current/zookeeper-server/bin/zkCli.sh /znode-1 my_data
ls /znode-1
```

7. При успешном создании ноды, удалите тестовую ноду:

```
delete /znode-1
quit
```

8. Повторите п.2-6 для следующих нод Zookeeper

6. HAProxy

В качестве совершенствования компонента HAProxy рекомендуется обновление версии:

1. Получить новую версию HAProxy с официального сайта:
<http://www.haproxy.org/download/2.0/src/>

2. Выполнить распаковку архива новой версии HAProxy:

```
tar xzvf [NEWVersion_HAPROXY].tar.gz
```

3. Выполнить бекапирование конфигурационного файла:

```
cp /haproxy/haproxy.cfg /haproxy/haproxy.cfg.bk
```

4. Выполнить обновление версии HAProxy:

```
apt-get install --only-upgrade haproxy
```

5. Убедиться, что новая версия haproxy установлена успешно:

```
haproxy --version
```

6. Повторить п.3-5 для следующих нод

7. Grafana

В качестве совершенствования компонента Grafana рекомендуется обновление версии:

1. Получить новую версию Grafana с официального сайта:
<https://grafana.com/grafana/download>

2. Выполнить бекапирование базы данных Grafana в PostgreSQL:

```
pg_dump Grafana > grafana_bk
или
pg_restore Grafana > grafana_bk
```

3. Выполнить бекапирование установленных плагинов

4. Выполнить команды:


```
dpkg -i grafana_[version]_amd64.deb  
apt-get update  
apt-get upgrade
```

5. Повторить п.2-4 для следующих нод

8. Prometheus

В качестве совершенствования компонента Prometheus рекомендуется обновление версии:

1. Получить новую версию Prometheus с официального сайта:

<https://prometheus.io/download>

2. Остановить службу Prometheus:

```
systemctl stop prometheus
```

3. Выполнить распаковку архива новой версии Prometheus:

```
tar xvf Prometheus-[version].linux-amd64.tar.gz
```

4. Выполнить бэкапирование старой версии Prometheus:

```
mv prometheus_[old_version] prometheus_[old_version]_bk
```

5. Выполнить копирование каталог promtool и prometheus из каталога новой версии Prometheus в каталог старой версии Prometheus:

```
cp prometheus_[new_version]/prometheus prometheus_[old_version]/prometheus  
cp prometheus_[new_version]/promtool prometheus_[old_version]/promtool
```

6. Запустить Prometheus

```
systemctl start prometheus
```

7. Убедиться в работоспособности

8. Повторить п.4-7 для следующих нод