Arenadata™ Streaming

Версия - v1.5-RUS

Руководство администратора по работе с ADS

Оглавление

В документации приведены необходимые сведения для работы с платформой ADS.

Инструкция может быть полезна администраторам, программистам, разработчикам и сотрудникам подразделений информационных технологий, осуществляющих внедрение и сопровождение системы.

Important: Контактная информация службы поддержки – e-mail: info@arenadata.io

Глава 1

Инструменты управления

В **ADS** предусмотрено 2 вида операций:

- *Операции на уровне кластера* – предполагают выполнение операций последовательно над всеми сервисами;
- *Операции на уровне сервиса* – предполагают выполнение операций над отдельным сервисом.

1.1 Операции на уровне кластера

Запуск и остановка **ADS** – существует возможность последовательного запуска и остановки всех сервисов кластера *ADS* через **ADCM**. Для этого необходимо открыть в **ADCM** кластер *ADS* и нажать кнопку *Start* или *Stop* в зависимости от требуемой работы с кластером (Рис.1.1).

Рис.1.1.: Операции на уровне кластера

1.2 Операции на уровне сервиса

Для каждого из сервисов доступна возможность проверки его работоспособности, а также управления им независимо от остальных. Например, проверка работоспособности сервиса *Kafka* представляет собой создание тестовых топиков и проверку их доступности на каждом из хостов *BROKER*. А проверка работоспособности сервиса *Zookeeper* представляет собой подключение к кворуму *Zookeeper*, создание в нем тестовой *znode* и проверку доступности созданной *znode* каждому из хостов кворума.

Проверка состояний сервисов и вывод результатов действий над ними осуществляется по единому алгоритму, разобранному на примере сервиса *Zookeeper*:

1. В ADCM перейти в кластер *ADS*. На вкладке “Services” для сервиса *Zookeeper* в поле “Actions” нажать на пиктограмму и выбрать действие *Check* (Рис.1.2).

Рис.1.2.: Запуск проверки состояния сервиса *Zookeeper*

2. Открыть вкладку “JOBS” (Рис.1.3).
3. Выбрать последнее действие над кластером *ADS* и в открывшемся окне проверить результаты (Рис.1.4).

1.3 Запуск и остановка сервисов

Для каждого из сервисов есть возможность управления им независимо от остальных, выполняя такие операции как *Stop*, *Start*, *Restart*.

Например, чтобы перезапустить все компоненты сервиса *Kafka*, необходимо в **ADCM** перейти в кластер *ADS*, на вкладке “Services” для сервиса *Kafka* в поле “Actions” нажать на пиктограмму и выбрать действие *Restart*.

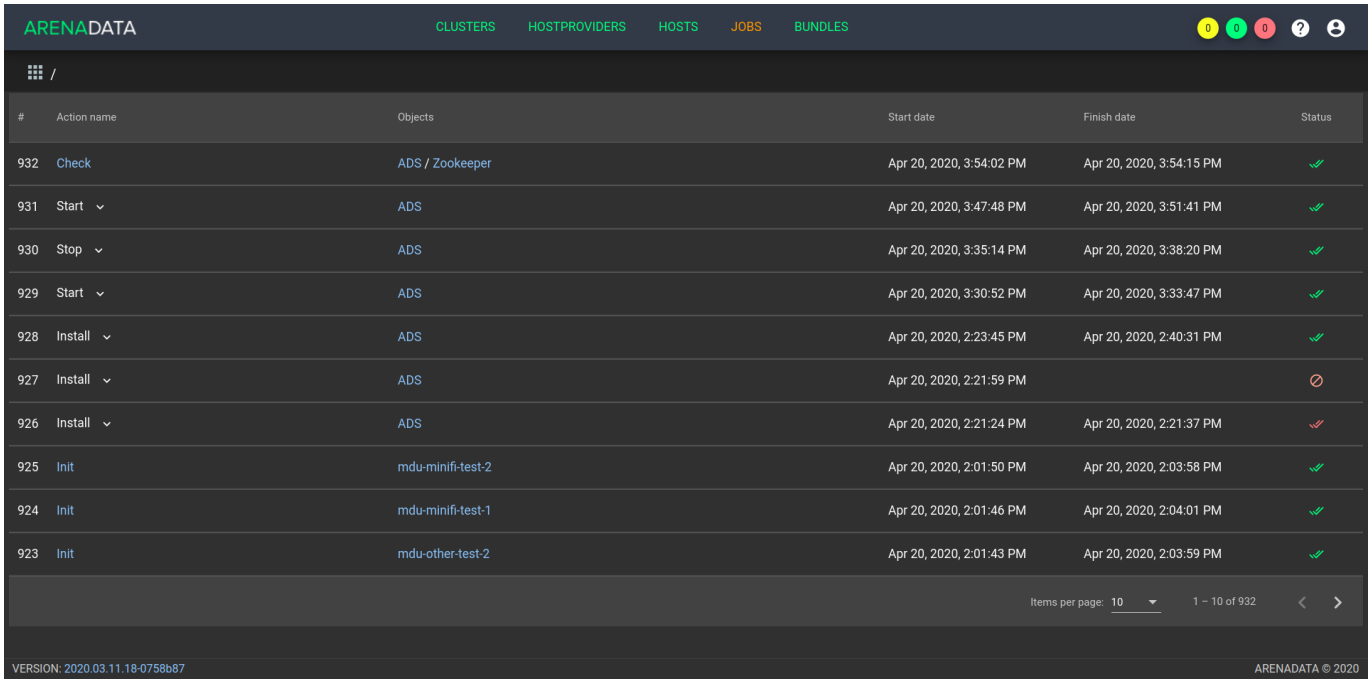

Рис.1.3.: Вкладка “JOBS”

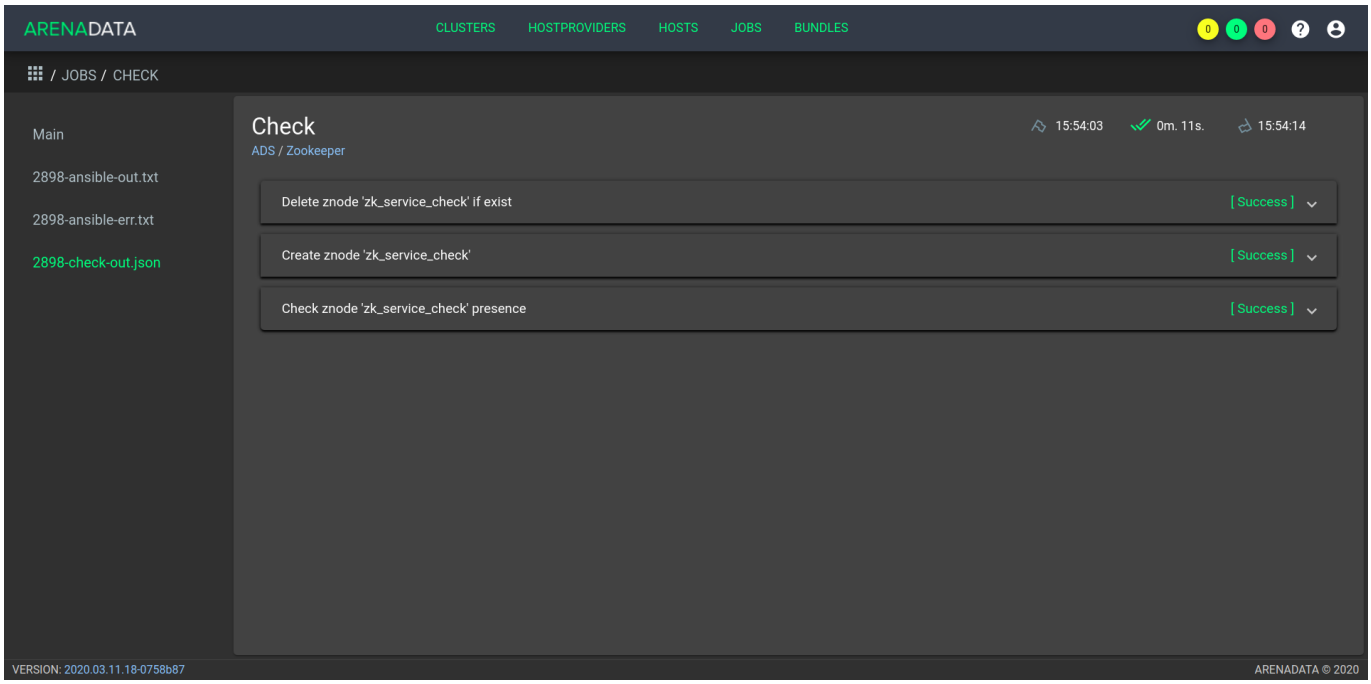

Рис.1.4.: Проверка состояния сервиса *Zookeeper*

Глава 2

Обновление кластера ADS

Доступно с версии 1.4.11

ADCM предоставляет возможность обновления существующего кластера **ADS**.

Процесс обновления состоит из двух последовательных шагов:

- Обновление бандла;
- Обновление кластера.

Important: В текущей версии доступно обновление кластеров как версий *1.3.X*, так и *1.4.X*

2.1 Обновление бандла

Для обновления бандла необходимо:

1. Загрузить бандл *ADS* новой версии. После его загрузки на вкладке “Clusters” в строке кластера с более старой версией бандла в колонке “Upgrade” появляется пиктограмма, указывающая на возможность обновления (Рис.2.1).

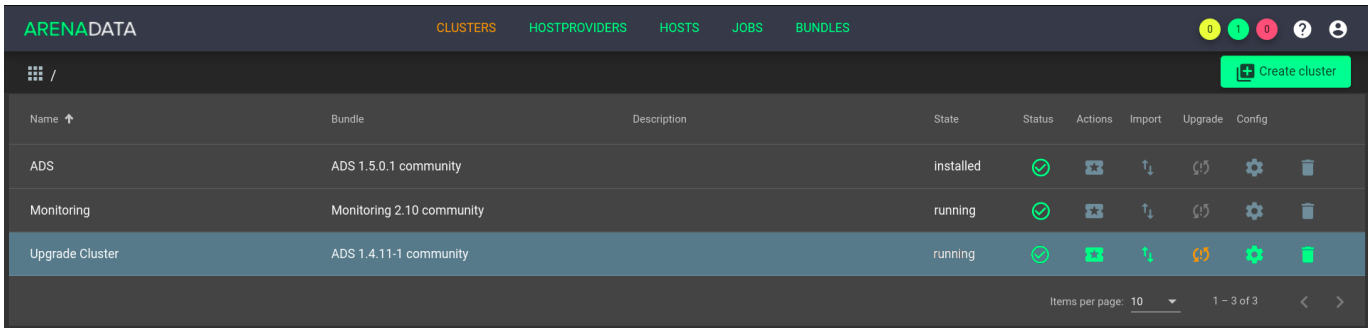

Рис.2.1.: Доступно обновление бандла

2. Нажать на пиктограмму в колонке “Upgrade” и выбрать доступную требуемую версию из списка (Рис.2.2).
3. В открывшемся диалоговом окне подтвердить действие, после чего кластер меняет состояние на *upgrade from 1.3.X* или *upgrade from 1.4.X* в зависимости от установленной версии бандла (Рис.2.3).

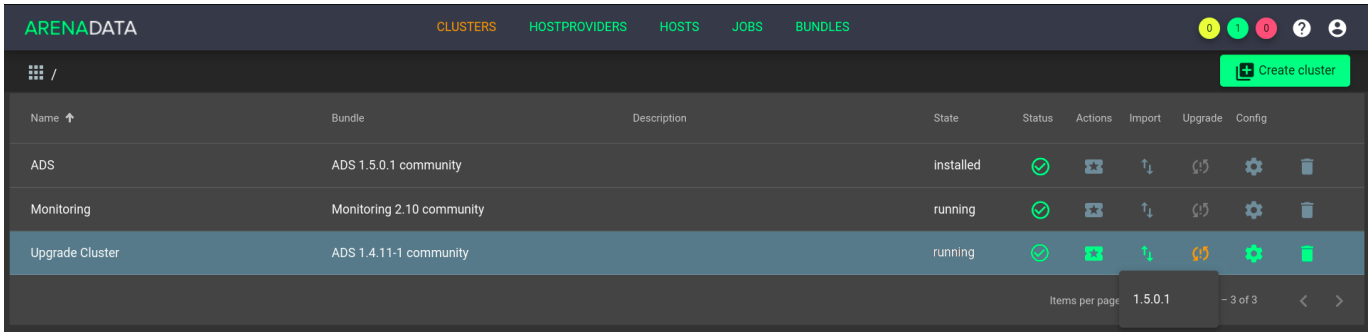

Рис.2.2.: Доступные обновления

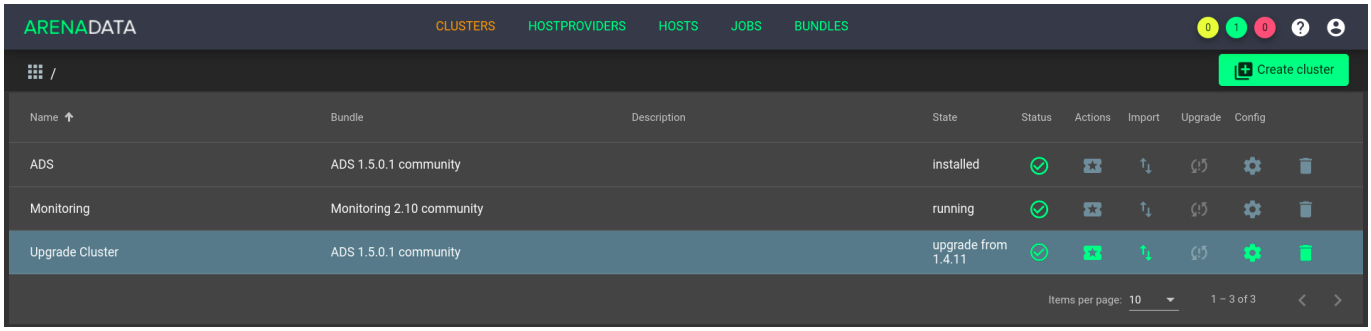

Рис.2.3.: Изменение состояния кластера после обновления

Important: Если заданные по умолчанию настройки сервисов *Zookeeper*, *Kafka* и *Nifi* изменены, то их необходимо скопировать и сохранить прежде, чем приступить к обновлению конфигураций сервисов. В частности это касается файлов *nifi.properties*, *zoo.cfg* и *server.properties* сервисов *Nifi*, *Zookeeper* и *Kafka* соответственно

2.2 Обновление кластера

После завершения операции “Upgrade Configs” в кластере становится доступным действие “Upgrade”. Данная операция применяет новые настройки, полученные на предыдущем шаге, и обновляет пакеты всех сервисов до указанных версий.

1. В поле “Actions” для обновляемого кластера нажать на пиктограмму и выбрать действие “Upgrade” (Рис.2.4).
2. Подтвердить действие в открывшемся диалоговом окне нажатием кнопки “Run”.

   После успешного завершения операции “Upgrade” кластеру присваивается состояние *installed*.

Important: Если заданные по умолчанию настройки сервисов были изменены перед обновлением, то после операции “Upgrade Configs” необходимо выполнить действия для соответствующих сервисов:

- Перейти к настройкам сервиса *Zookeeper*, проверить раздел *zoo.cfg* и при необходимости внести сохраненные ранее изменения;
- Перейти к настройкам сервиса *Kafka*, проверить разделы *Main* и *server.properties* и при необходимости

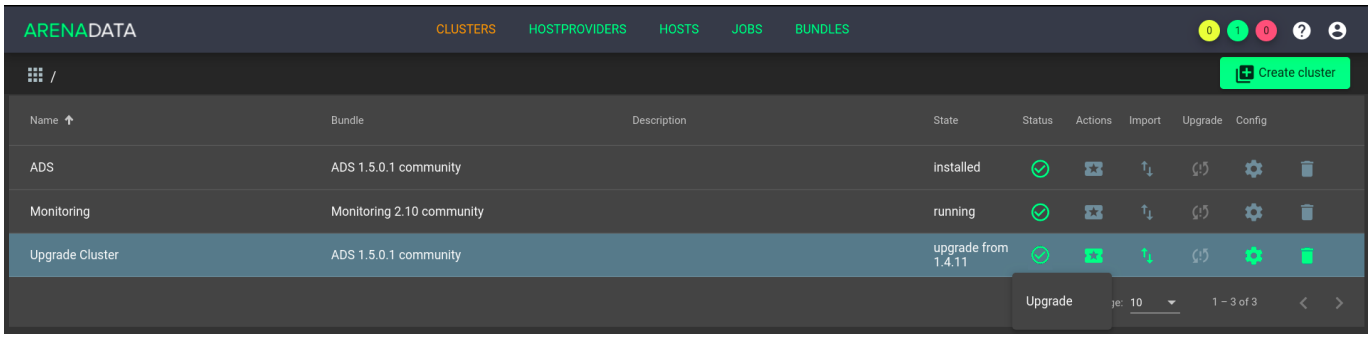

Рис.2.4.: Обновление пакетов сервисов

внести сохраненные ранее изменения;

- Перейти к настройкам сервиса *Nifi*, проверить разделы *Main*, *Directories* и *nifi.properties* и при необходимости внести сохраненные ранее изменения.

Глава 3

Настройки сервисов при помощи ADCM

В главе приведено описание сервисов **ADS** и их настройка при помощи **ADCM**:

- *Zookeeper*;
- *Kafka*;
- *Nifi*;
- *Schema-registry*;
- *Kafka REST Proxy*;
- *KSQL*;
- *Kafka-Manager*;
- *MiNifi*;
- *Monitoring Clients*.

3.1 Zookeeper

Для перехода к настройкам сервиса *Zookeeper* необходимо нажать кнопку с пиктограммой шестеренки в соответствующей строке вкладки “SERVICES” в интерфейсе **ADCM** и перейти в раздел меню “Configuration”. При этом открывается окно настроек сервиса *Zookeeper* (Рис.3.1.).

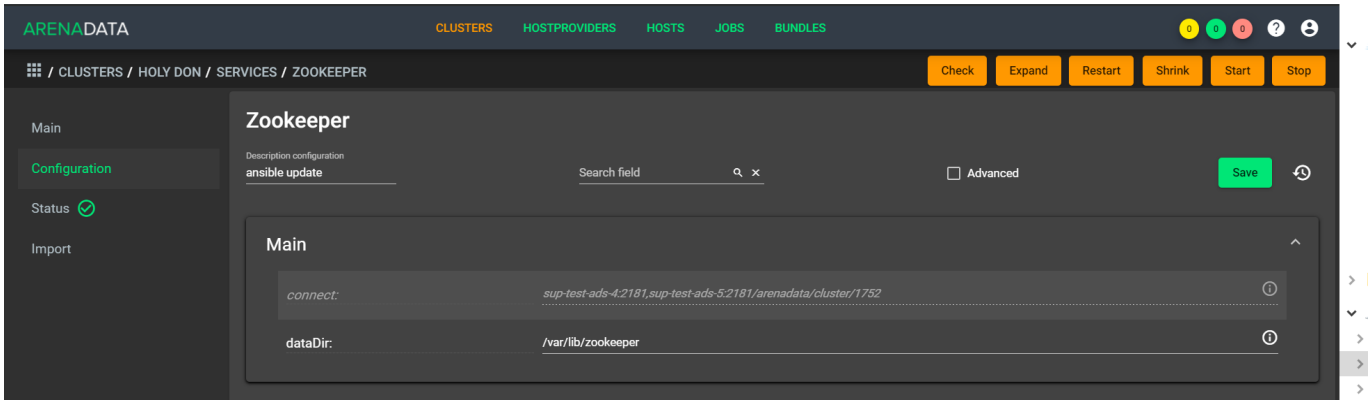

Рис.3.1.: Настройки сервиса Zookeeper

В блоке настроек “Main” задаются основные параметры:

- `connect` – строка подключения к Znode, в которой Zookeeper хранит конфигурацию текущего кластера, используется сервисом Kafka. В текущей реализации данный параметр недоступен для редактирования и автоматически генерируется на стороне ADCM;
- `dataDir` – каталог для хранения снапшотов и транзакционных логов Zookeeper. Одноименный параметр в конфигурационном файле *zoo.cfg*.

При простановке флага в поле “Advanced” открывается блок дополнительных настроек сервиса *Zookeeper* (Рис.3.2.).

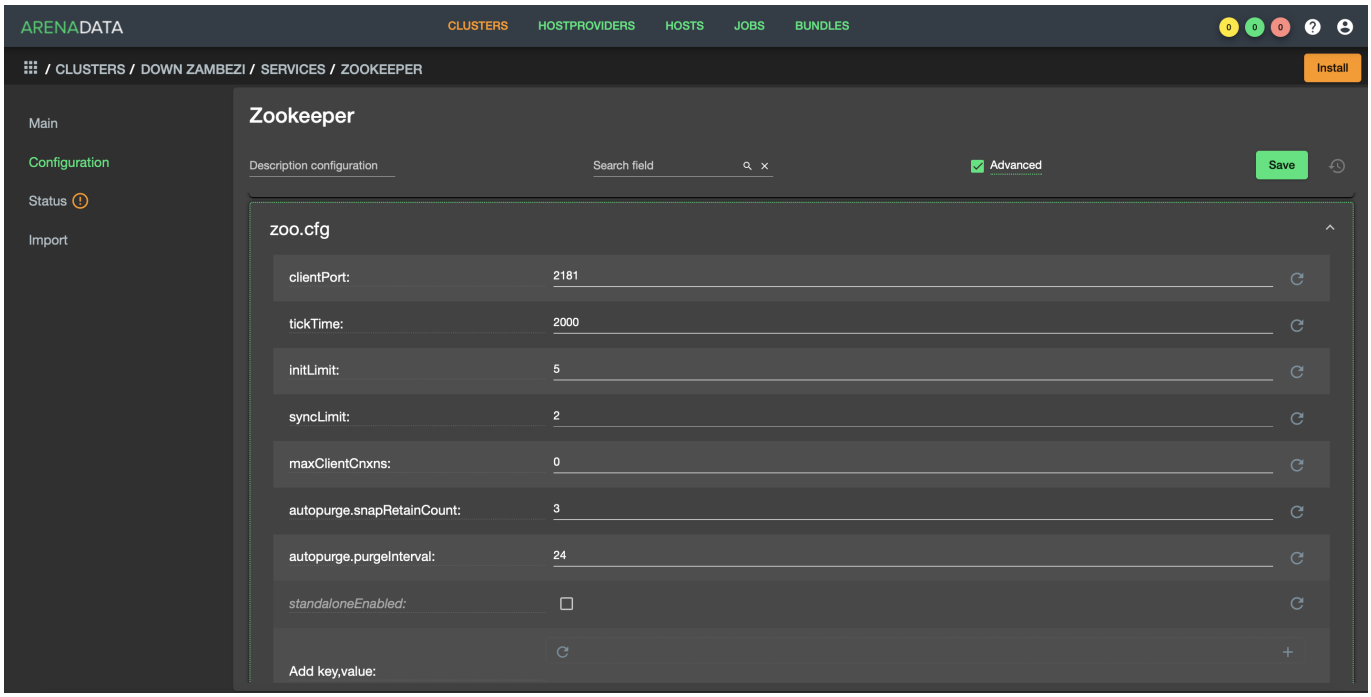

Рис.3.2.: Дополнительные настройки сервиса Zookeeper

В группе настроек *zoo.cfg* находится ряд конфигураций, представляющих собой одноименные настройки в *zoo.cfg*-файле:

- `clientPort` – порт, на котором Zookeeper слушает клиентские подключения;
- `tickTime` – базовая единица времени в миллисекундах, используемая ZooKeeper. Используется при отправке heartbeats-сообщений, при этом минимальное время ожидания сессии в два раза больше установленного значения в параметре;
- `initLimit` – это тайм-ауты, которые ZooKeeper использует для ограничения времени, в течение которого серверы ZooKeeper в кворуме должны соединиться с лидером;
- `syncLimit` – параметр ограничивает, насколько устаревшим может быть сервер от лидера;
- `maxClientCnxns` – ограничивает количество одновременных подключений (на уровне сокетов), которые может выполнить один идентифицируемый по IP-адресу клиент одному члену ансамбля ZooKeeper. Используется для предотвращения определенных классов DoS-атак, включая исчерпание файловых дескрипторов. Установка значения в *0* полностью снимает ограничение для одновременных подключений;
- `autopurge_snapRetainCount` – при включенном параметре функция автоматической очистки ZooKeeper сохраняет самые последние снапшоты и соответствующие журналы транзакций в `dataDir` и `dataLogDir` соответственно и удаляет остальные;

- `autopurge_purgeInterval` – интервал времени в часах, в течение которого должна быть запущена задача очистки. Для включения автоматической очистки значение параметра должно быть установлено на положительное целое число (*1* и выше);
- `standaloneEnabled` – включение работы в автономном режиме.

Если необходимая настройка отсутствует в списке группы *zoo.cfg*, то для добавления таковой следует воспользоваться строкой `Add key,value`, где требуется написать ключ и значение в соответствующих полях.

Далее в группе настроек файла *zookeeper-env.sh* задаются параметры, которые используются для внесения переменных окружения сервиса *Zookeeper* (Рис.3.3.).

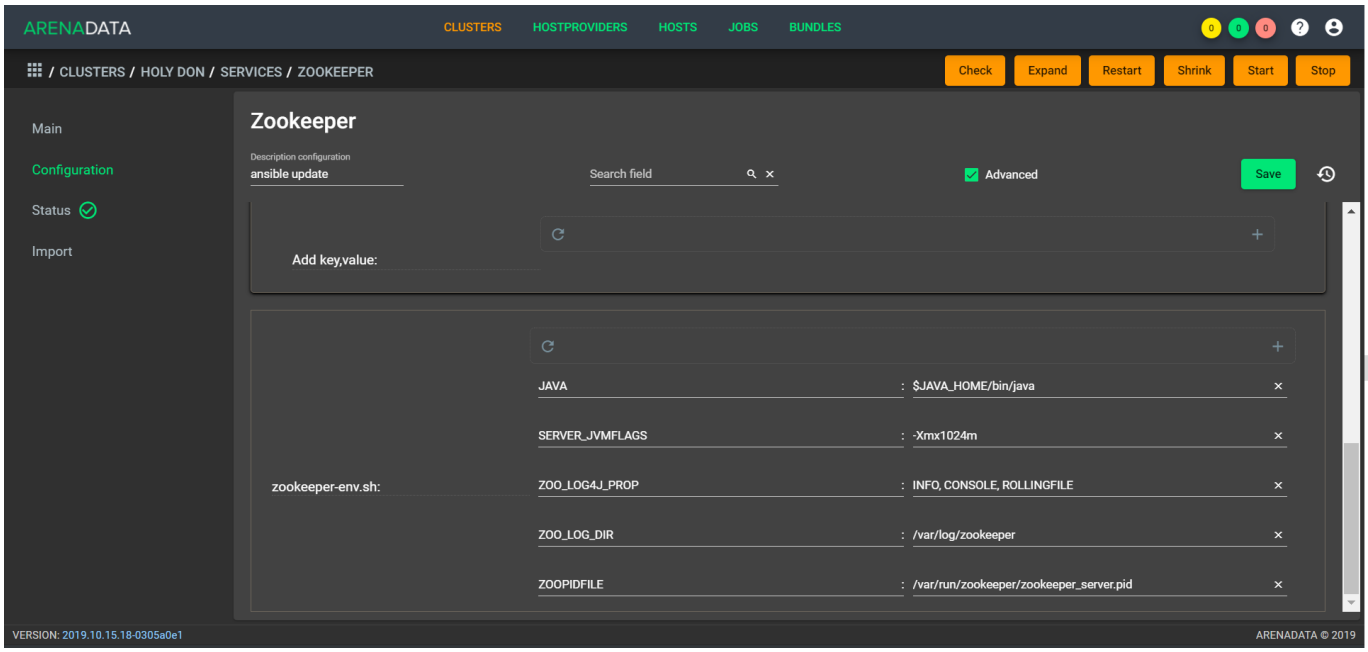

Рис.3.3.: Настройки переменных окружения сервиса Zookeeper

3.2 Kafka

Для перехода к настройкам сервиса *Kafka* необходимо нажать кнопку с пиктограммой шестеренки в соответствующей строке вкладки “SERVICES” в интерфейсе **ADCM** и перейти в раздел меню “Configuration”. При этом открывается окно настроек сервиса *Kafka* (Рис.3.4.).

В блоке настроек “Main” задаются основные параметры сервиса *Kafka*:

- `log.dirs` – каталоги, в которых Kafka хранит данные журнала. Одноименное свойство в файле конфигурации server.properties;
- `listeners` – список URI (протокол, хост и порт, на котором поднят брокер), разделенный запятыми. Если используется не *PLAINTEXT* протокол, то необходимо также указать `listener.security.protocol.map`. Для привязки ко всем интерфейсам указать имя хоста как *0.0.0.0*. Оставить имя хоста пустым для привязки к интерфейсу по умолчанию. Указывается в качестве параметра `listeners` в конфигурационном файле server.properties;
- `default.replication.factor` – коэффициенты репликации по умолчанию для автоматически создаваемых топиков. Одноименное свойство в файле конфигурации server.properties;
- `delete.topic.enable` – данный параметр позволяет удалять топики. Если параметр выключен, то удаление топика через инструменты администрирования не приводит к фактическому удалению. Одноименное

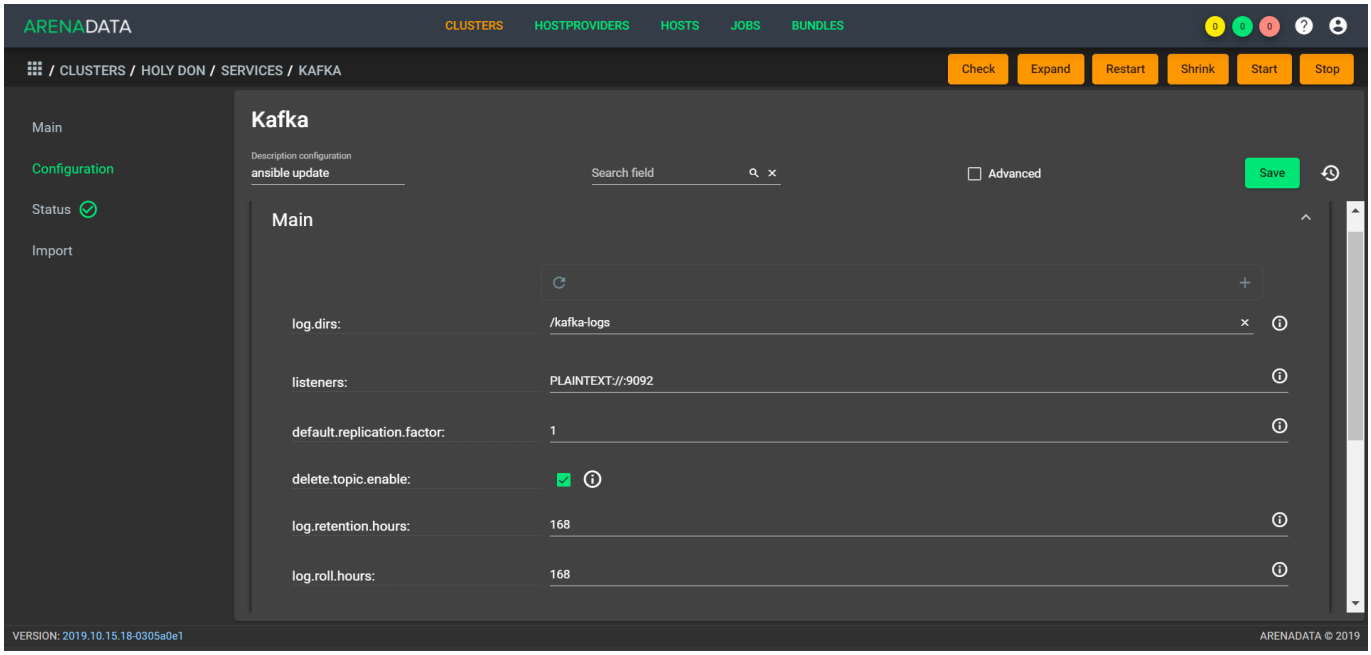

Рис.3.4.: Настройки сервиса Kafka

свойство в файле конфигурации server.properties;

- `log.retention.hours` – количество часов, в течение которых топики хранятся в Kafka. Одноименный параметр в конфигурационном файле server.properties;
- `log.roll.hours` – количество часов, по истечению которых появляется новый журнал сегмента, даже если старый журнал не переполнен. Одноименный параметр в конфигурационном файле server.properties.

При простановке флага в поле “Advanced” открывается блок дополнительных настроек сервиса *Kafka* (Рис.3.5.).

В группе настроек *server.properties* находится ряд конфигураций, представляющих собой одноименные настройки в *server.properties*-файле:

- `auto.leader.rebalance.enable` – включение автоматической балансировки лидера. Балансировка лидера в фоновом режиме через регулярные промежутки времени;
- `queued.max.requests` – количество запросов в очереди до блокировки сетевых потоков;
- `num.network.threads` – количество потоков, используемых сервером для получения запросов от сети и отправки ответов в сеть;
- `num.io.threads` – число потоков, используемых сервером для обработки запросов, которые могут включать дисковые операции ввода-вывода;
- `unclean.leader.election.enable` – указывает, следует ли включить не входящие в набор ISR реплики и установка последнего средства в качестве лидера, даже если это может привести к потере данных;
- `offsets.topic.replication.factor` – коэффициент репликации для топика смещения (устанавливается выше с целью обеспечения доступности). Создание внутреннего топика невозможно, пока размер кластера не соответствует данному требованию коэффициента репликации;
- `transaction.state.log.min.isr` – переопределение конфигурации *min.insync.replicas* для топика транзакции;

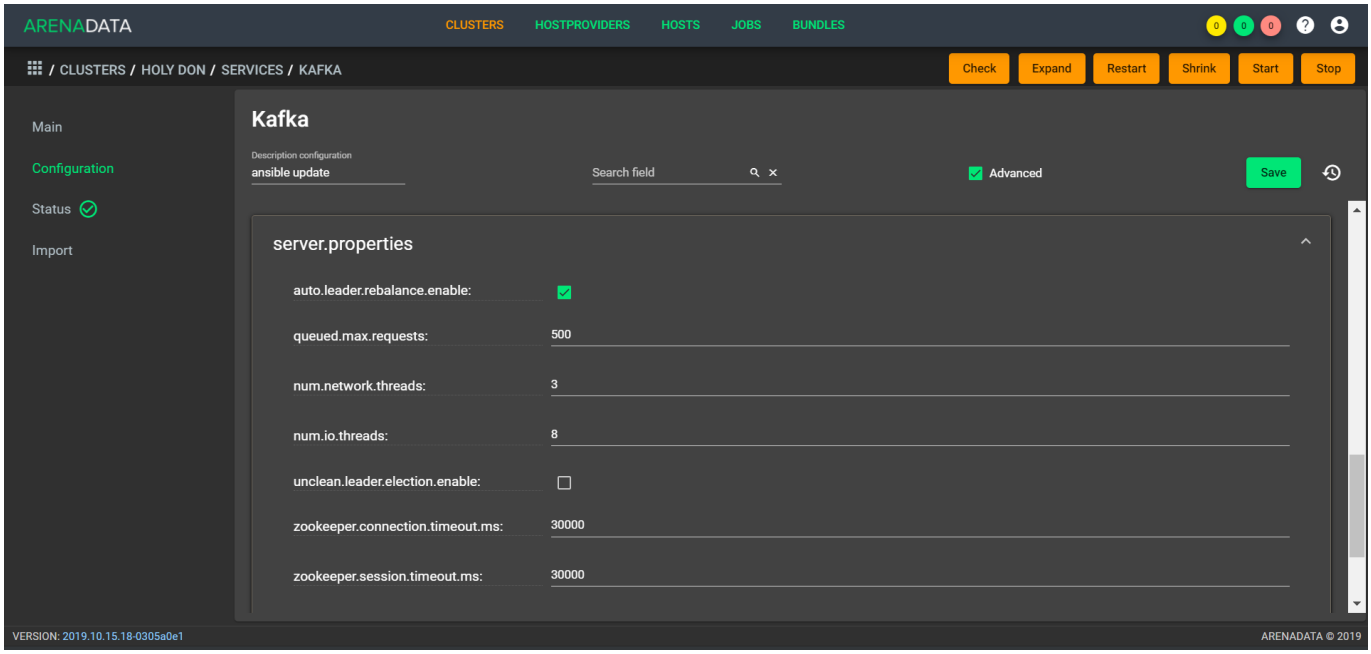

Рис.3.5.: Дополнительные настройки сервиса Kafka

- `transaction.state.log.replication.factor` – коэффициент репликации для топика транзакции (задается выше для обеспечения доступности). Создание внутреннего топика завершается ошибкой, пока размер кластера не соответствует данному требованию к фактору репликации;
- `zookeeper.connection.timeout.ms` – максимальное время ожидания клиентом установки соединения с Zookeeper. Если параметр не задан, используется значение для *zookeeper.session.timeout.ms*. Указывается в миллисекундах;
- `zookeeper.session.timeout.ms` – тайм-аут сессии Zookeeper. Указывается в миллисекундах;
- `zookeeper.sync.time.ms` – удаленность последователя Zookeeper от лидера Zookeeper. Указывается в миллисекундах;
- `num.partitions` – число партиций по умолчанию для каждого топика.

Если необходимая настройка отсутствует в списке группы *server.properties*, то для добавления таковой следует воспользоваться строкой `Add key,value`, где требуется написать ключ и значение в соответствующих полях.

Далее в группе настроек файла *kafka-env.sh* задаются параметры, которые используются для внесения переменных окружения сервиса *Kafka* (Рис.3.6.).

3.3 Nifi

Для перехода к настройкам сервиса *Nifi* необходимо нажать кнопку с пиктограммой шестеренки в соответствующей строке вкладки “SERVICES” в интерфейсе **ADCM** и перейти в раздел меню “Configuration”. При этом открывается окно настроек сервиса *Nifi* (Рис.3.7.).

В блоке настроек “Main” задаются основные параметры:

- `Nifi UI port` – http-порт, на котором поднимается веб-интерфейс сервиса Nifi. Указывается в качестве параметра `nifi.web.http.port` в конфигурационном файле *nifi.properties*;

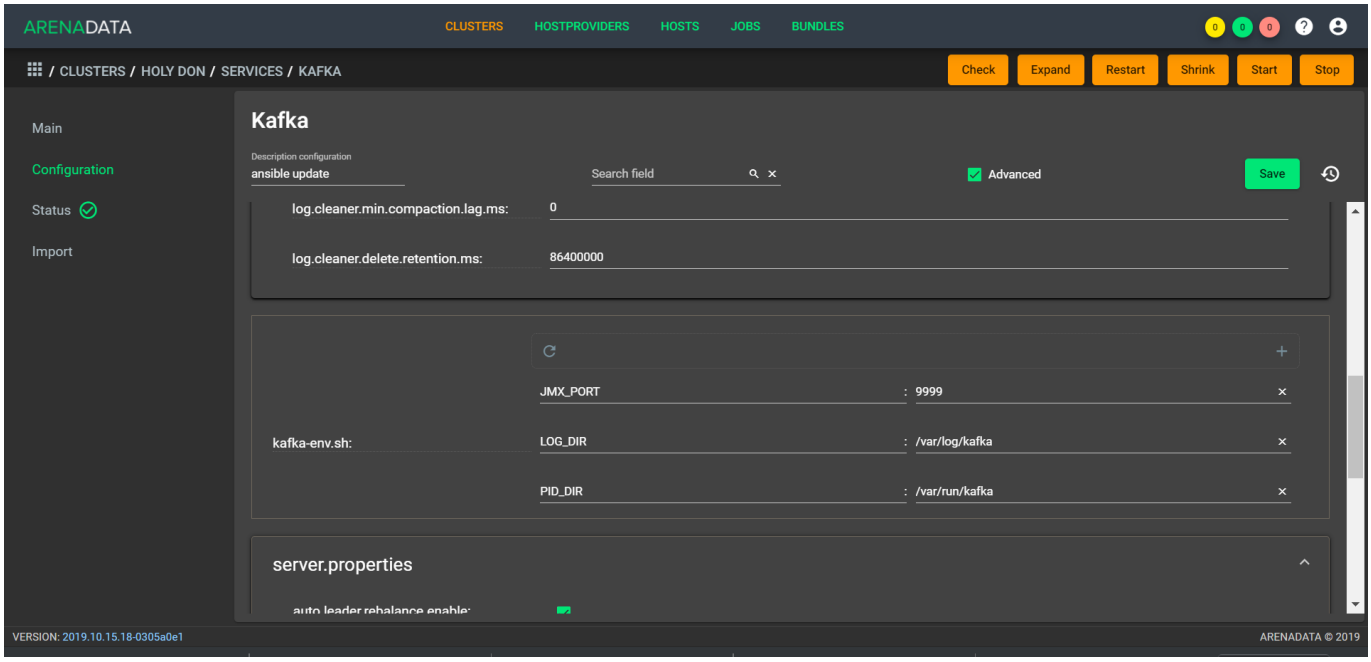

Рис.3.6.: Настройки переменных окружения сервиса Kafka

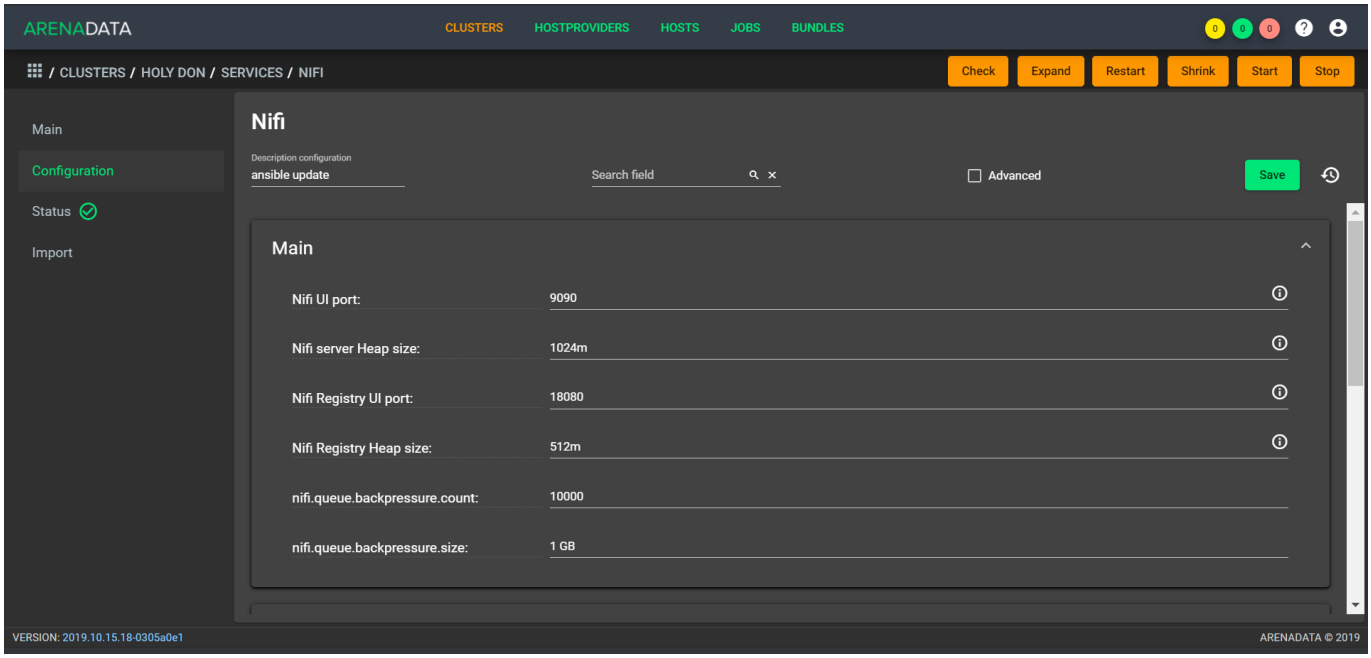

Рис.3.7.: Настройки сервиса Nifi

- `Nifi server Heap size` – размер кучи, выделяемой процессу сервиса Nifi. Указывается в конфигурационном файле *bootstrap.conf*;
- `Nifi Registry UI port` – http-порт реестра Nifi. Указывается в качестве параметра `nifi.registry.web.http.port` в файле конфигурации *nifi.properties*;
- `Nifi Registry Heap size` – размер кучи, выделяемой реестру Nifi. Указывается в конфигурационном файле *bootstrap.conf*.
- `nifi.queue.backpressure.count` – при создании нового соединения между двумя компонентами это значение по умолчанию для порогового значения объекта обратного воздействия этого соединения. Значение должно быть целым числом (integer);
- `nifi.queue.backpressure.size` – при создании нового соединения между двумя компонентами это значение по умолчанию для порогового значения размера данных обратного воздействия этого соединения. Значение должно быть размером данных, включая единицу измерения.

В блоке настроек “Directories” задаются параметры расположения репозиториев сервиса *Nifi* (Рис.3.8.):

- `nifi.flowfile.repository.directory` – расположение репозитория FlowFile. Значением по умолчанию является *./flowfile_repository*;
- `nifi.content.repository.directory` – расположение репозитория Content. Значением по умолчанию является *./content_repository*;
- `nifi.provenance.repository.directory` – расположение репозитория Provenance. Значением по умолчанию является *./provenance_repository*;
- `nifi.database.directory` – расположение директории H2 database. Значением по умолчанию является *./database_repository*;
- `nifi.registry.db.directory` – расположение директории Registry database;
- `nifi.nar.library.directory.lib` – параметр следует использовать в случае добавления custom nars (необязательный параметр).

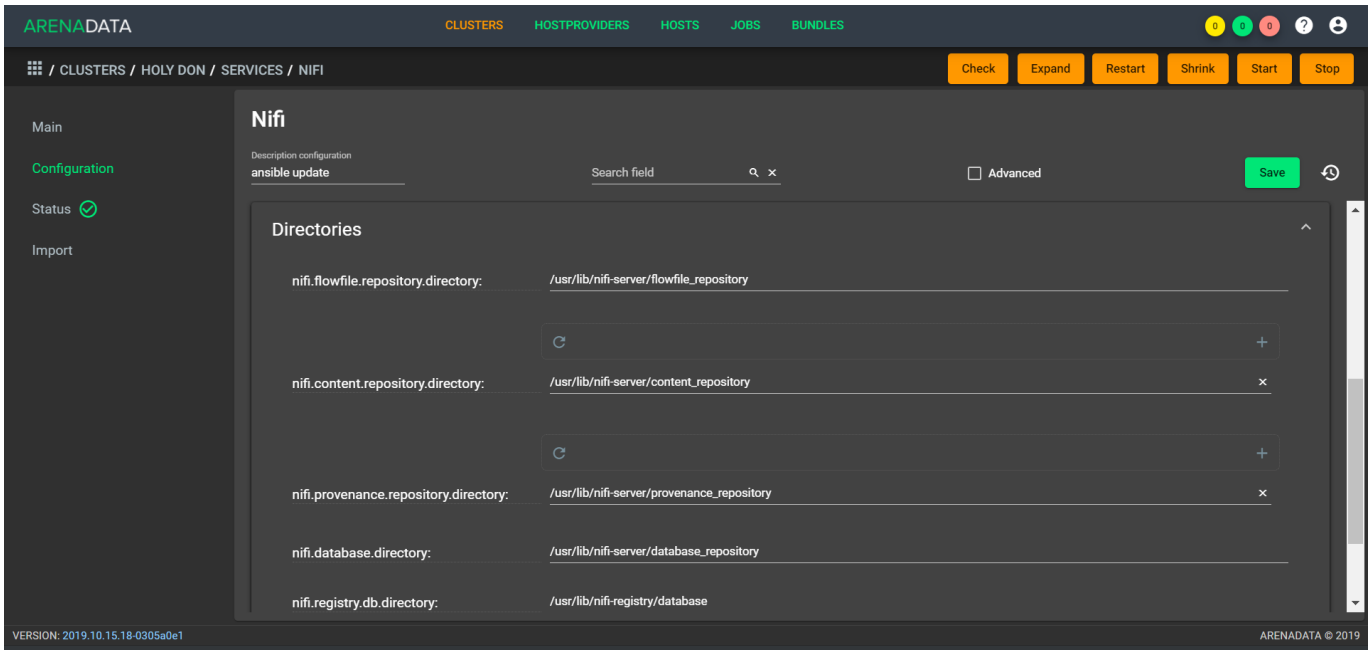

Рис.3.8.: Настройки директорий репозиториев сервиса Nifi

В группе настроек *Analytics Framework* (Рис.3.9.) можно задать следующие параметры:

- `nifi.analytics.predict.interval` – интервал времени, в который должны быть сделаны аналитические прогнозы (например, насыщение очереди). Значение по умолчанию составляет *3* минуты;
- `nifi.analytics.query.interval` – интервал времени для запроса прошлых наблюдений (например, последние 3 минуты снапшотов). Значение по умолчанию составляет *5* минут. Примечание: значение должно быть как минимум в 3 раза больше, чем заданный `nifi.components.status.snapshot.frequency`, для обеспечения получения достаточного количества наблюдений для прогнозов;
- `nifi.analytics.connection.model.implementation` – класс реализации для модели анализа состояния, используемой для прогнозирований соединения. Значением по умолчанию является *org.apache.nifi.controller.status.analytics.models.OrdinaryLeastSquares*;
- `nifi.analytics.connection.model.score.name` – имя типа скоринга, которое следует использовать для оценки модели. Значением по умолчанию является *rSquared*;
- `nifi.analytics.connection.model.score.threshold` – порог для значения скоринга (модель score должна быть выше заданного порога). Значением по умолчанию является *90*.

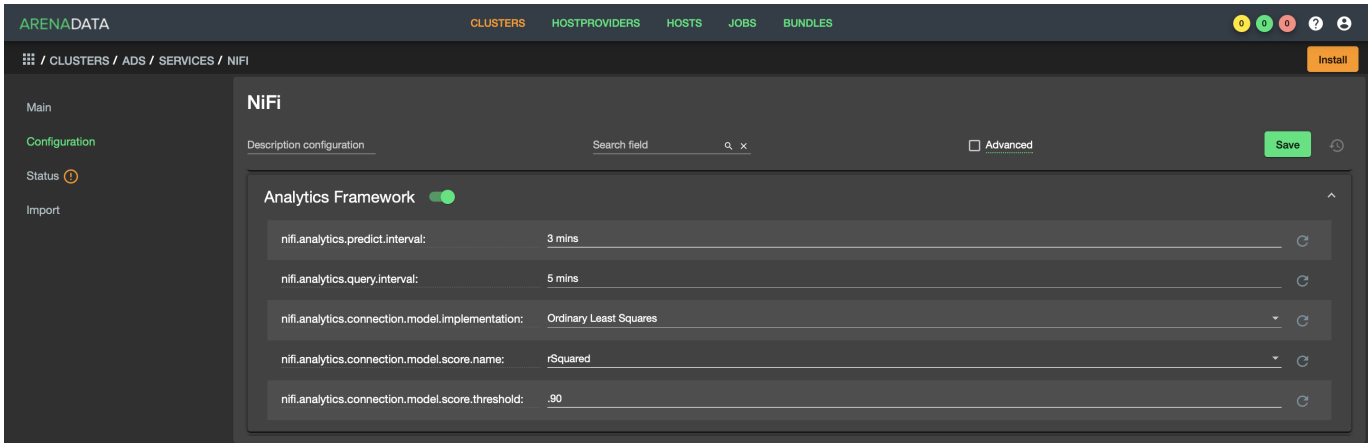

Рис.3.9.: Настройки секции Analytics Framework

В группе настроек *Nifi-Registry Provider* находятся конфигурации потоков сервиса *Nifi* (Рис.3.10.):

- `Flow Persistence Provider Type` – тип провайдера потока, по умолчанию – файловая система;
- `Flow Storage Directory` – директория хранения потока;
- `Bundle Persistence Provider Type` – тип провайдера бандла, по умолчанию – файловая система;
- `Extension Bundle Storage Directory` – директория хранения бандла.

При простановке флага в поле “Advanced” открывается блок дополнительных настроек сервиса *Nifi* (Рис.3.11.).

В группе настроек *nifi.properties* находится ряд конфигураций, представляющих собой одноименные настройки в *nifi.properties*-файле:

- `nifi.flow.configuration.file` – расположение файла конфигурации потока (то есть файла, который содержит то, что в текущий момент отображается на графике NiFi). Значением по умолчанию является *./conf/flow.xml.gz*;
- `nifi.flow.configuration.archive.enabled` – указывает, создает ли NiFi автоматически резервную копию потока при обновлении потока. Значение по умолчанию *true*;
- `nifi.cluster.node.connection.timeout` – при подключении к другому узлу в кластере указывает, как долго этот узел должен ждать, прежде чем считать соединение неудачным;

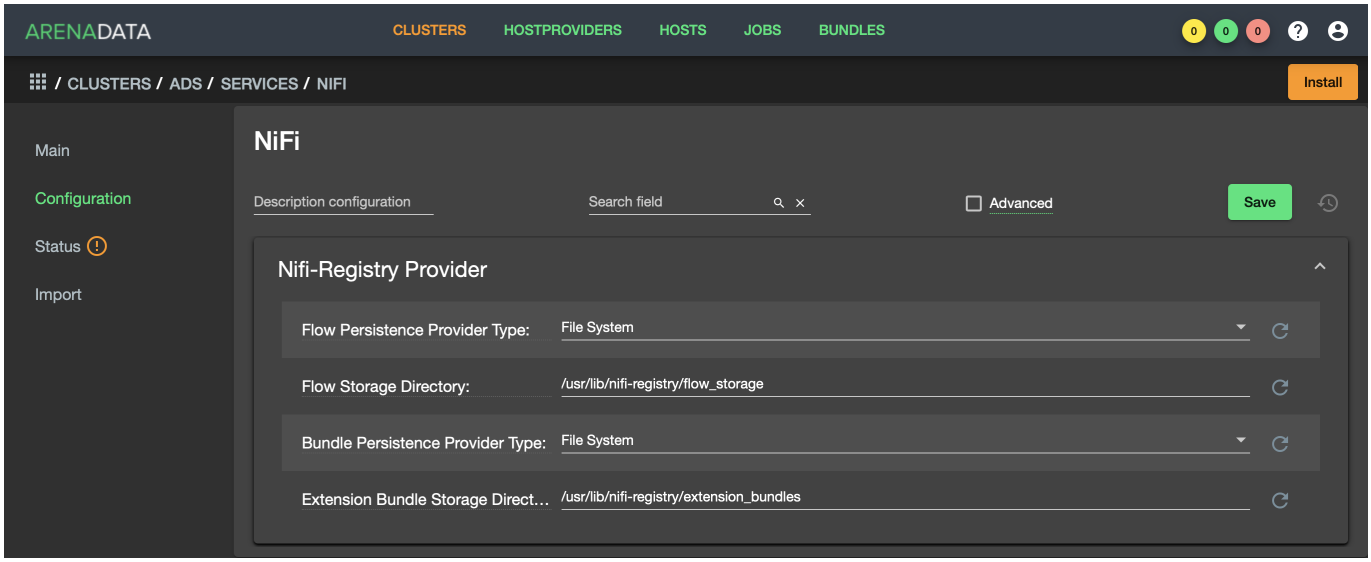

Рис.3.10.: Настройки Nifi-Registry Provider

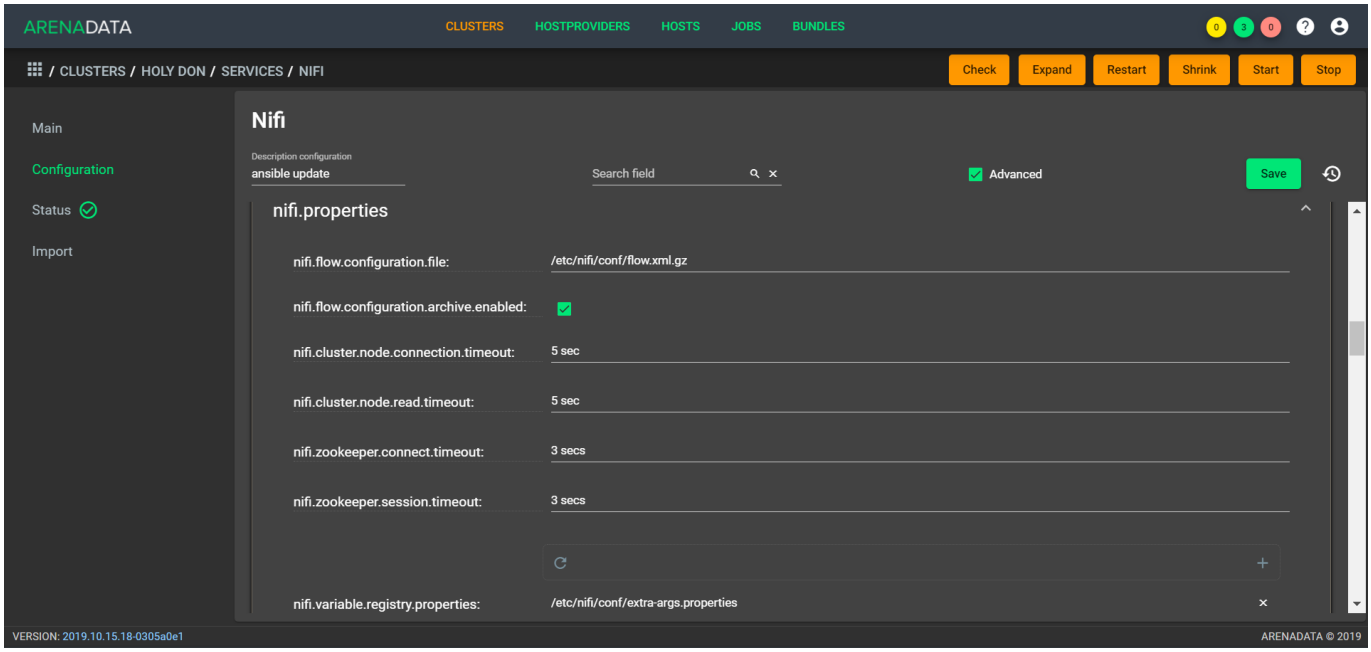

Рис.3.11.: Дополнительные настройки сервиса Nifi

- `nifi.cluster.node.read.timeout` – при связи с другим узлом в кластере указывает, как долго этот узел должен ожидать получения информации от удаленного узла, прежде чем считать связь с узлом неудачной;
- `nifi.zookeeper.connect.timeout` – время ожидания при подключении к ZooKeeper, прежде чем подключение считается неудачным;
- `nifi.zookeeper.session.timeout` – время ожидания после потери соединения с ZooKeeper до истечения сессии;
- `nifi.variable.registry.properties` – разделенный запятыми список путей расположения файлов для одного или нескольких файлов индивидуальных свойств.

Далее в группе настроек файла *nifi-env.sh* задаются параметры, которые используются для внесения переменных окружения сервиса *Nifi* (Рис.3.12.).

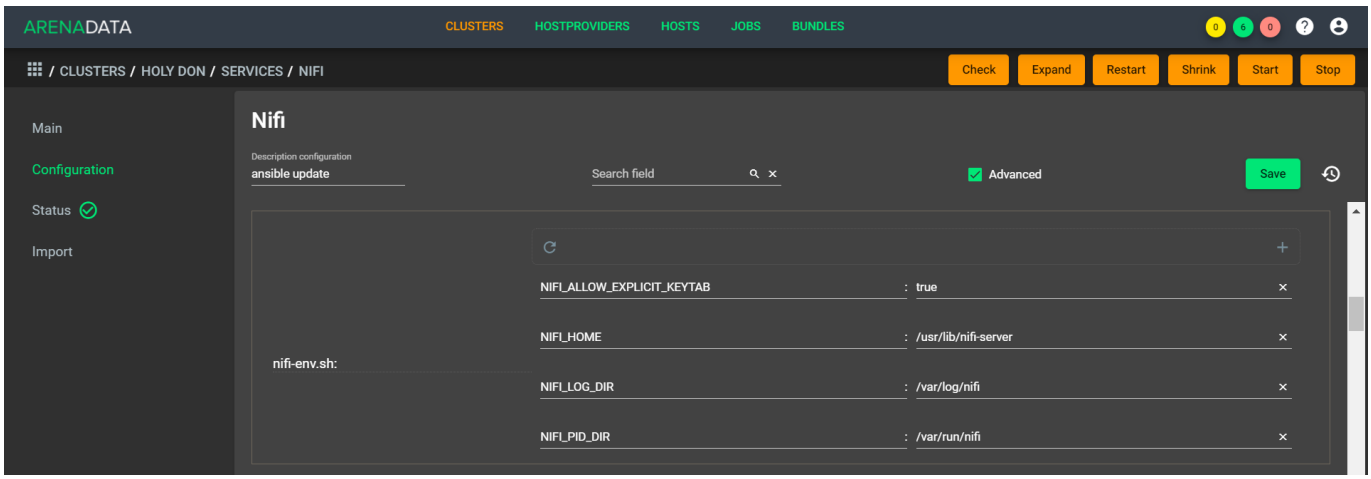

Рис.3.12.: Настройки переменных окружения сервиса Nifi

Далее в группе настроек файла *nifi-registry-env.sh* задаются параметры, которые используются для внесения переменных окружения сервиса *Nifi Regisrtry* (Рис.3.13.).

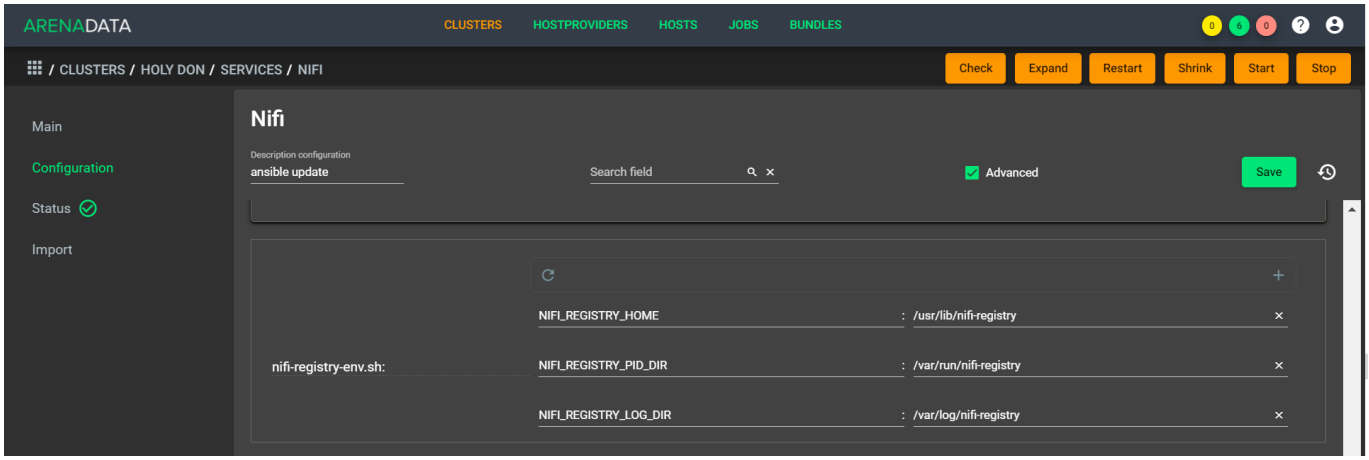

Рис.3.13.: Настройки переменных окружения сервиса Nifi Registry

3.4 Schema-registry

Для перехода к настройкам сервиса *schema-registry* необходимо нажать кнопку с пиктограммой шестеренки в соответствующей строке вкладки “SERVICES” и перейти в раздел меню “Configuration”. При этом открывается окно настроек сервиса *schema-registry* (Рис.3.14.).

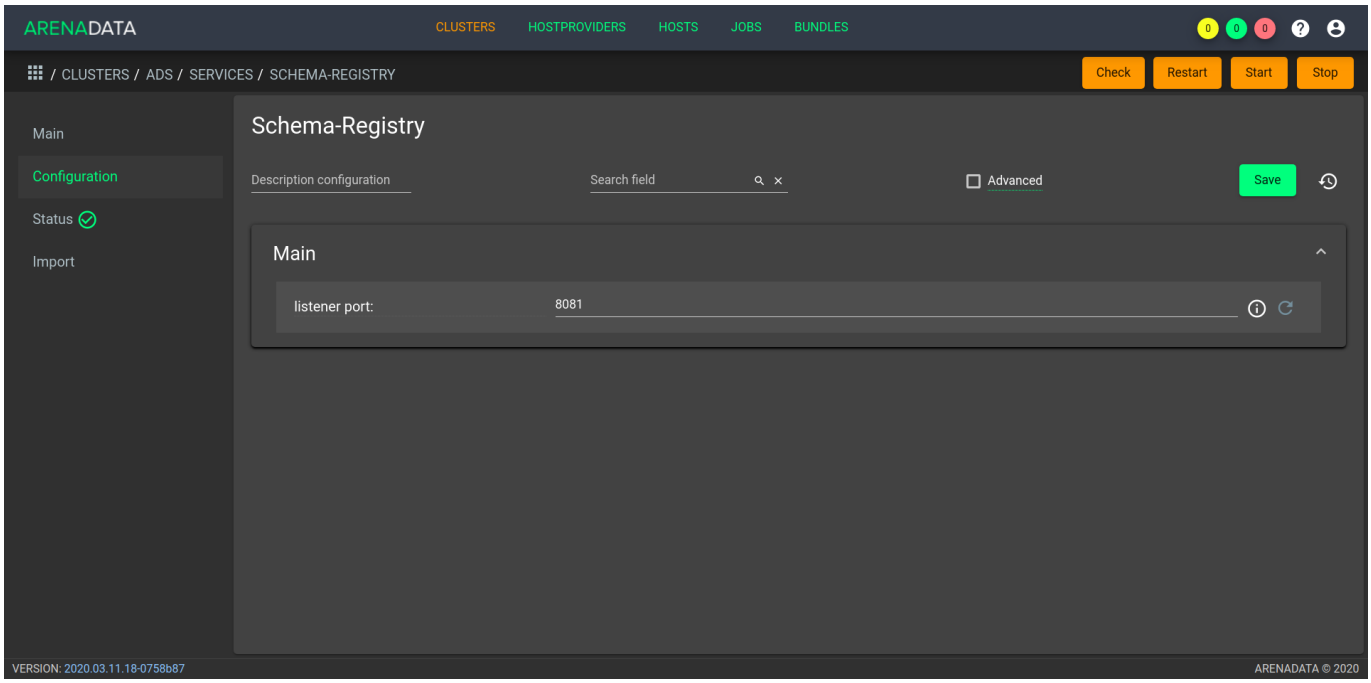

Рис.3.14.: Настройки сервиса Schema-registry

В блоке настроек “Main” задаются следующие параметры:

- `listener_port` – порт, который слушает *schema-registry*. Указывается в качестве параметра `listeners` в конфигурационном файле *schema-registry.properties*;

При простановке флага в поле “Advanced” открывается блок дополнительных настроек сервиса *Schema-Registry*. В группе настроек файла *schema-registry-env.sh* задаются параметры, которые используются для внесения переменных окружения сервиса *Schema-Registry* (Рис.3.15.).

3.5 Kafka REST Proxy

Для перехода к настройкам сервиса *Kafka REST Proxy* необходимо нажать кнопку с пиктограммой шестеренки в соответствующей строке вкладки “SERVICES” и перейти в раздел меню “Configuration”. При этом открывается окно настроек сервиса *Kafka REST Proxy* (Рис.3.16.).

В блоке настроек “Main” задаются следующие параметры:

- `rest_heap_opts` – размер кучи, выделяемой процессу Kafka REST Proxy. Указывается в качестве параметра `KAFKAREST_HEAP_OPTS` в файле *kafka-rest-env.sh*;
- `rest_listener_port` – порт, который слушает REST Proxy. Указывается в качестве параметра `listeners` в конфигурационном файле *kafka-rest.properties*;
- `rest_jmx_port` – порт, по которому Kafka REST Proxy отдает jmx-метрики. Указывается в качестве параметра `JMX_PORT` в файле *kafka-rest-env.sh*.

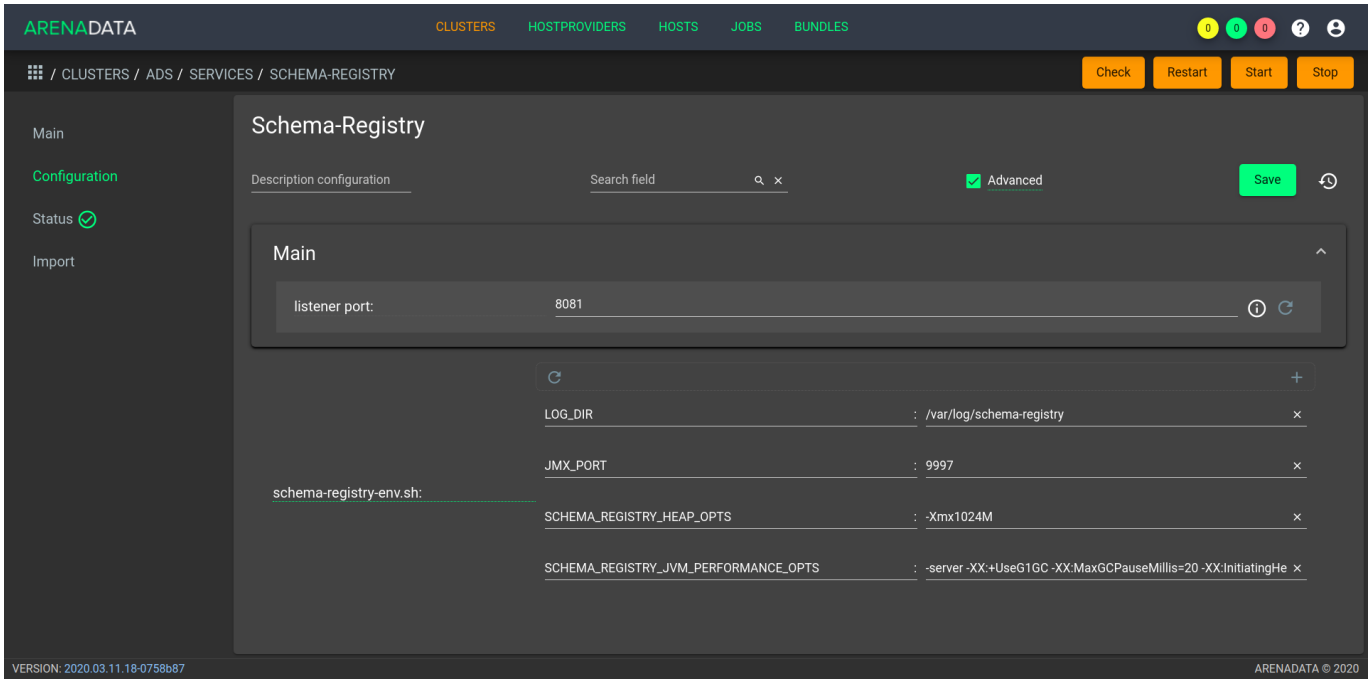

Рис.3.15.: Настройки переменных окружения сервиса Schema-registry

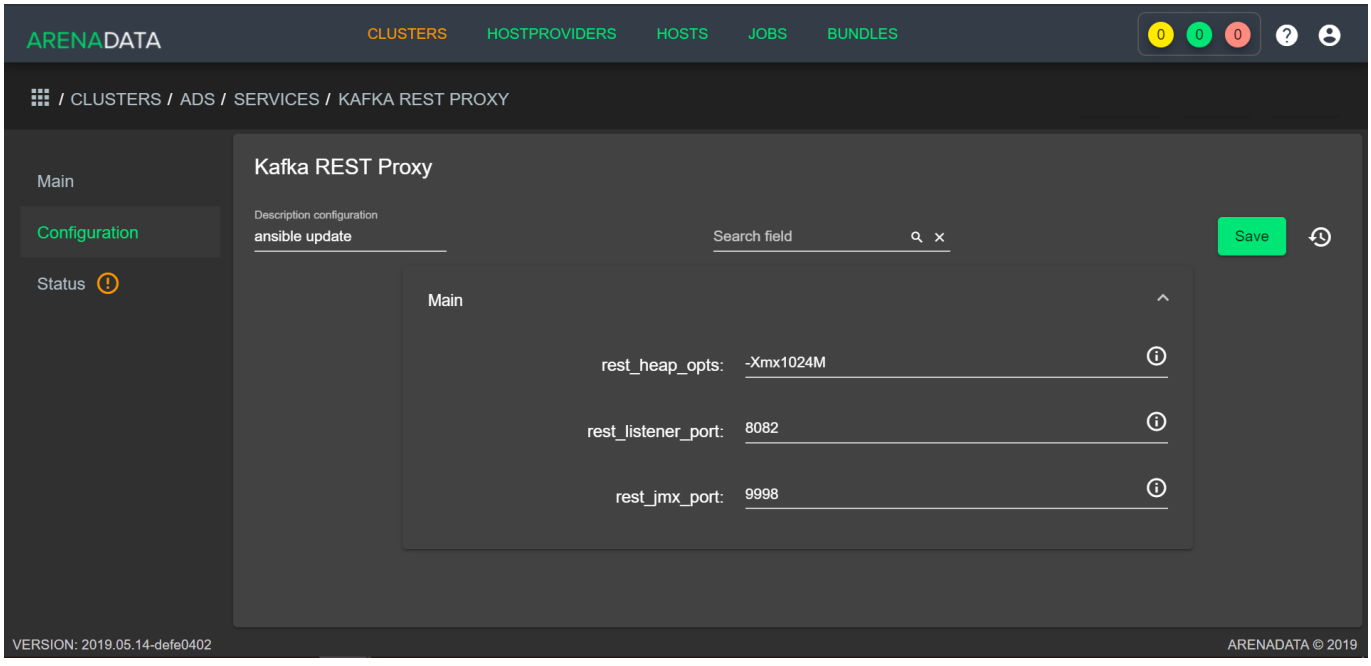

Рис.3.16.: Настройки сервиса Kafka REST Proxy

3.6 KSQL

Для перехода к настройкам сервиса *KSQL* необходимо нажать кнопку с пиктограммой шестеренки в соответствующей строке вкладки “SERVICES” и перейти в раздел меню “Configuration”. При этом открывается окно настроек сервиса *KSQL* (Рис.3.17.).

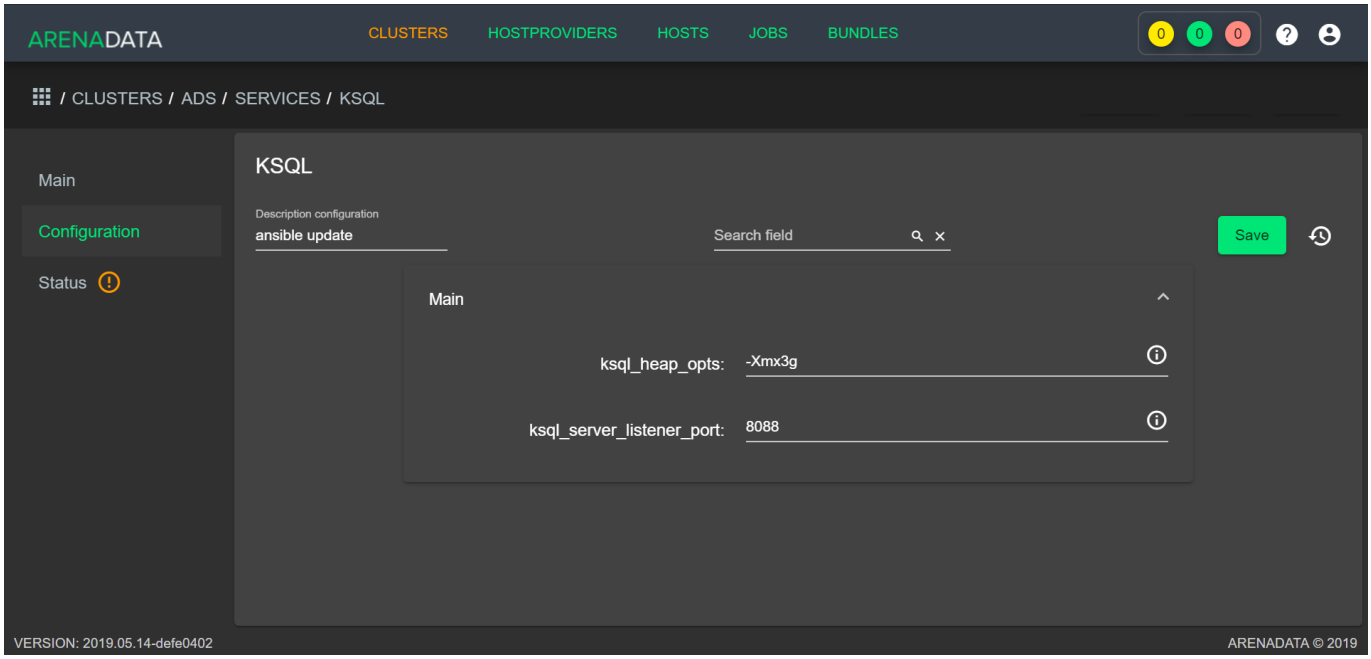

Рис.3.17.: Настройки сервиса KSQL

В блоке настроек “Main” задаются следующие параметры:

- `ksql_heap_opts` – размер кучи, выделяемой процессу KSQL. Указывается в качестве параметра `KSQL_HEAP_OPTS` в файле *ksql-env.sh*;
- `ksql_server_listener_port` – порт, который слушает сервер KSQL. Указывается в качестве параметра `listeners` в конфигурационном файле *ksql-server.properties*.

3.7 Kafka-Manager

Для перехода к настройкам сервиса *Kafka-Manager* необходимо нажать кнопку с пиктограммой шестеренки в соответствующей строке вкладки “SERVICES” и перейти в раздел меню “Configuration”. При этом открывается окно настроек сервиса *Kafka-Manager* (Рис.3.18.).

В блоке настроек “Main” задается следующий параметр:

- `manager_port` – порт, на котором поднимается Kafka-Manager. Указывается в файле *kafka-manager-env*.

При простановке флага в поле “Advanced” открывается блок дополнительных настроек сервиса *Kafka-Manager*. В группе настроек файла *Default POST data for Kafka cluster* задаются параметры, которые используются для добавления *Kafka*-кластера в сервис *Kafka-Manager* (Рис.3.19.).

3.8 MiNifi

Для перехода к настройкам сервиса *MiNifi* необходимо нажать кнопку с пиктограммой шестеренки в соответствующей строке вкладки “SERVICES” и перейти в раздел меню “Configuration”. При этом открывается

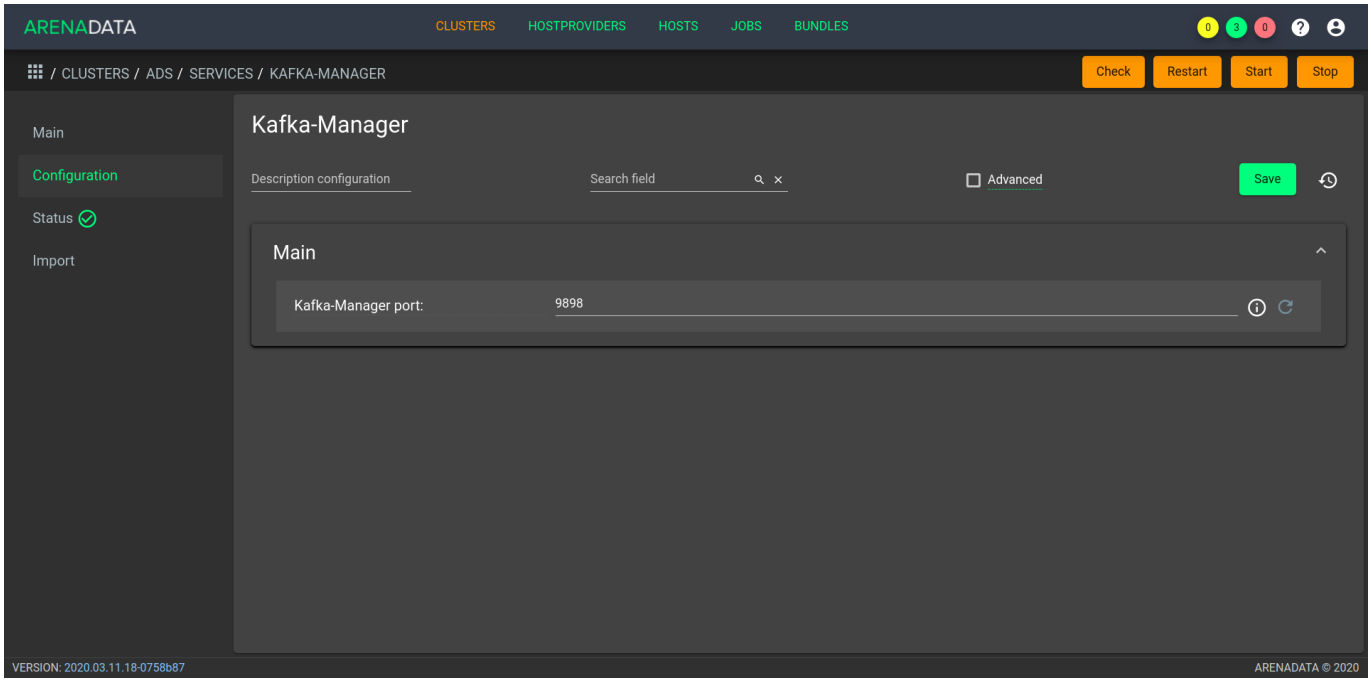

Рис.3.18.: Настройки сервиса Kafka-Manager

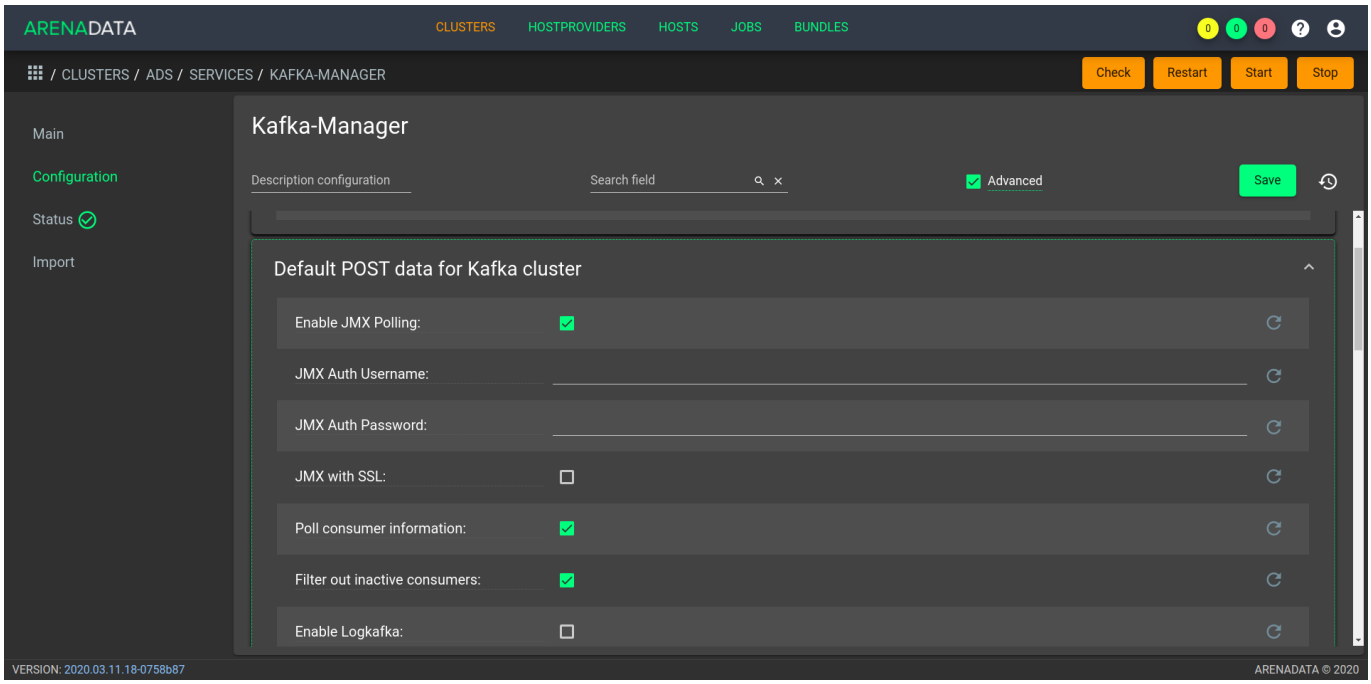

Рис.3.19.: Настройки добаления *Kafka*-кластера в сервис Kafka-Manager

окно конфигурации сервиса *MiNifi* (Рис.3.20.).

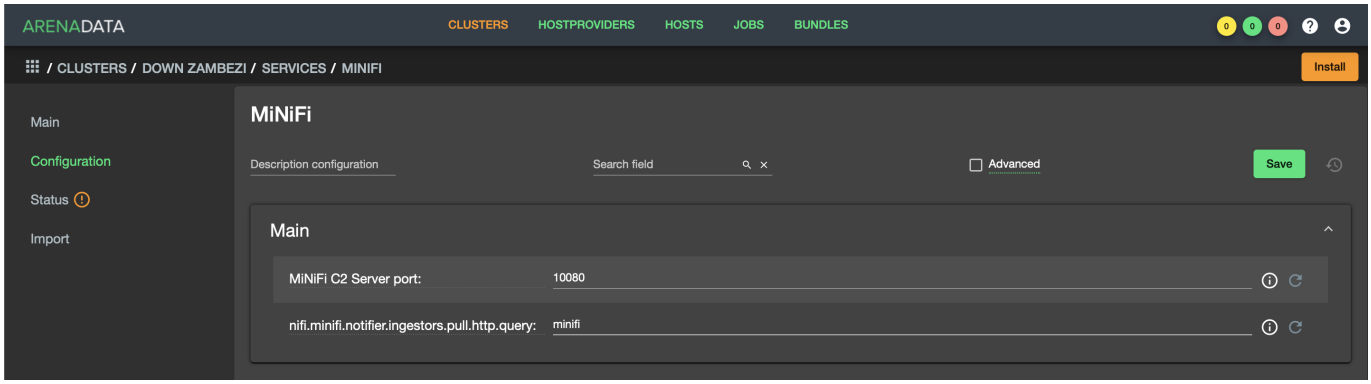

Рис.3.20.: Окно конфигурации сервиса MiNifi

В блоке настроек “Main” задаются следующие параметры:

- `MiNiFi C2 Server port` – порт сервера, по умолчанию *10080*;
- `nifi.minifi.notifier.ingestors.pull.http.query` – строка запроса для извлечения конфигураций.

При простановке флага в поле “Advanced” открывается блок дополнительных настроек сервиса *MiNifi*, где задаются параметры, используемые для внесения переменных окружения сервиса (Рис.3.21.).

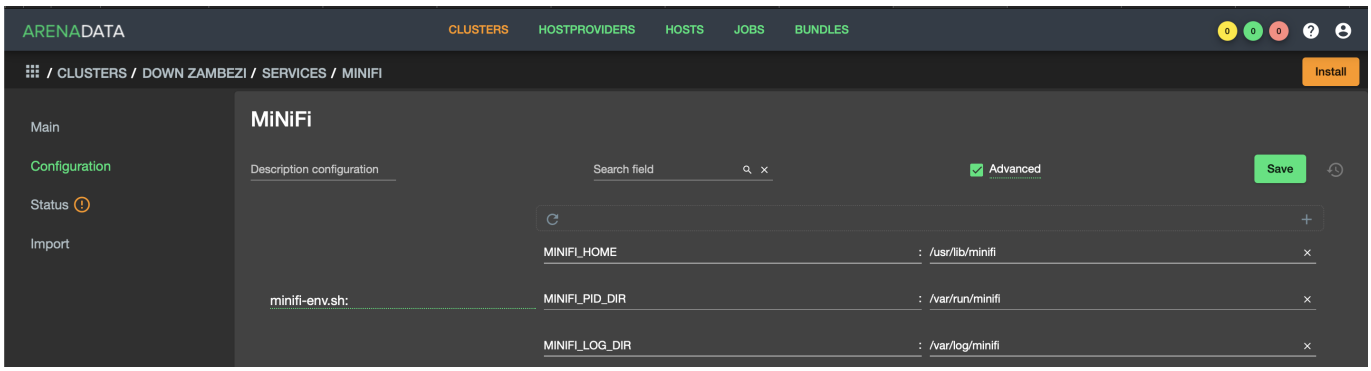

Рис.3.21.: Настройки переменных окружения сервиса MiNifi

Далее в группе дополнительных настроек *MiNiFi Agent bootstrap.conf* задаются параметры, которые используются для агента *Bootstrap* (Рис.3.22.).

- `MiNiFi Agent Heap size` – heap-размер агента, по умолчанию *256m*;
- `nifi.minifi.notifier.ingestors.pull.http.period.ms` – период проверки обновлений, по умолчанию *300000* мс;
- `nifi.minifi.status.reporter.log.query` – запрос состояния экземпляра MiNiFi, по умолчанию *instance:health,bulletins*, где *health* – состояние отчета экземпляра, активные потоки, наличие или отсутствие бюллетеней и каких-либо ошибок проверки; *bulletins* – список всех текущих бюллетеней (если есть). Так же доступно *stats* – текущее состояние экземпляра, включая, но не ограничиваясь, байты чтения/записи и отправленные/переданные FlowFiles;
- `nifi.minifi.status.reporter.log.level` – уровень журнала, на котором регистрируется статус. Доступные значения: *TRACE*, *DEBUG*, *INFO*, *WARN* и *ERROR*. По умолчанию *INFO*;

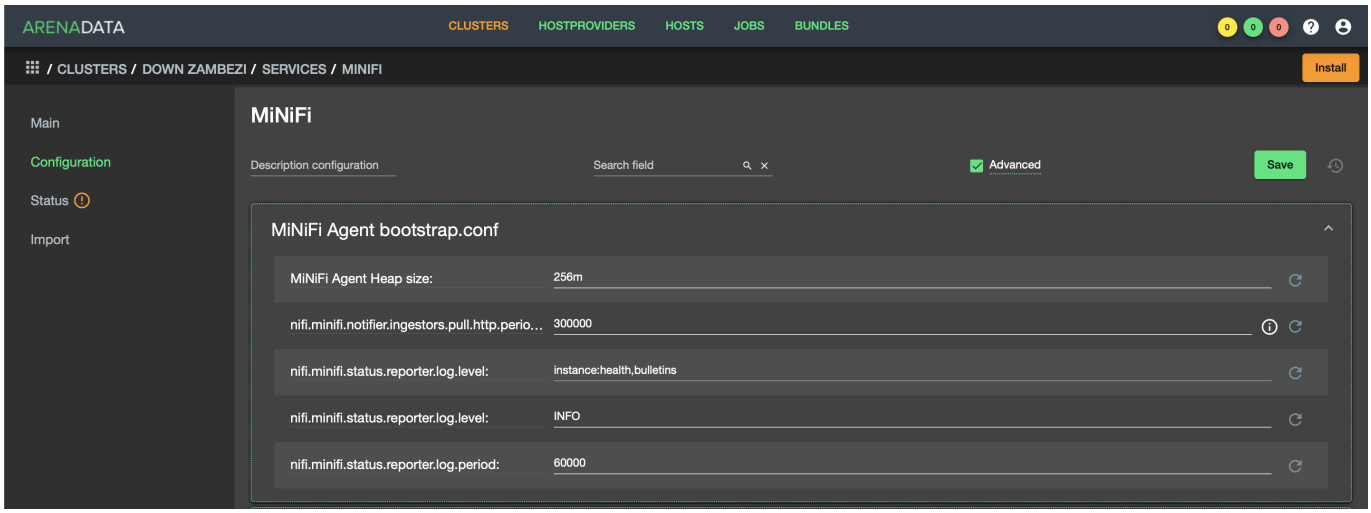

Рис.3.22.: Группа настроек MiNiFi Agent bootstrap.conf

- `nifi.minifi.status.reporter.log.period` – задержка между каждым запросом (в миллисекундах). По умолчанию *60000* мс.

3.9 Monitoring Clients

Для перехода к настройкам сервиса *monitoring clients* необходимо нажать кнопку с пиктограммой шестеренки в соответствующей строке вкладки “SERVICES” и перейти в раздел меню “Configuration”. При этом открывается окно конфигурации сервиса *monitoring clients* (Рис.3.23.).

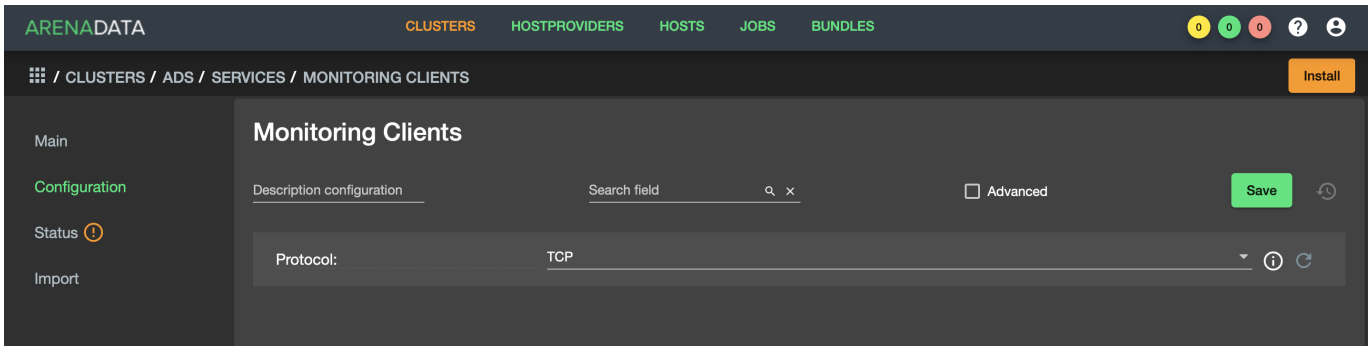

Рис.3.23.: Окно конфигурации сервиса Monitoring Clients

Глава 4

Интеграция сервисов MiNiFi и NiFi

ADS предоставляет, помимо *NiFi* и *MiNiFi*, поддержку централизованного управления *MiNiFi Agent* с помощью *MiNiFi C2 Server*. Данный сервис обеспечивает автоматическое обновление конфигураций *MiNiFi Agent* без сторонних вспомогательных средств. В данном разделе приведены основные шаги для настройки взаимодействи между *MiNiFi* и *NiFi* сервисами:

- *Создание шаблона*;
- *Проверка конфигурации*.

4.1 Создание шаблона

Для выполнения какой-либо задачи *MiNiFi Agent* необходимо создать шаблон в UI *NiFi*. В данном разделе представлен элементарный шаблон для сбора содержимого файла с машин *MiNiFi Agent*.

После установки *NiFi* и *MiNiFi* с помощью **ADCM**, в разделе *Template* появляется шаблон с название *simple-minifi-listener*, который состоит из следующих элементов (Рис.4.1.).

Чтобы создать шаблон для Агентов, необходимо перейти в *MiNiFi Process Group* и создать *Flow*, который будет выполнятся непосредственно *MiNiFi Agent*.

В нашем случае созданный *Flow* (Рис.4.2.) содержит процессор *TailFile*, который считывает содержимое файла и передает экземпляру *NiFi*

Для успешной загрузки *Flow* на *MiNiFi Agent*, необходимо сохранить шаблон с названием указанным в nifi.minifi.notifier.ingestors.pull.http.query с добавлением версии (например, *minifi.v1*). Если вы изменили *Flow*, то для актуализации его на агентах необходимо увеличить версию шаблона (например, *minifi.v2*)

Автоматическое обновление конфигурации *MiNiFi Agent* происходить с периодичностью, заданной nifi.minifi.notifier.ingestors.pull.http.period.ms. Если шаблон был неправильно собран, то Агенты продолжат работу на последней работоспособной конфигурации.

4.2 Проверка конфигурации

Текущую конфигурацию *Flow*, которую запрашивают *MiNiFi Agent* у *MiNiFi C2 Server*, можно проверить, обратившись к API *MiNiFi C2 Server* (Рис.4.3.). Ссылка указана в описании сервиса *MiNiFi* в **ADCM**.

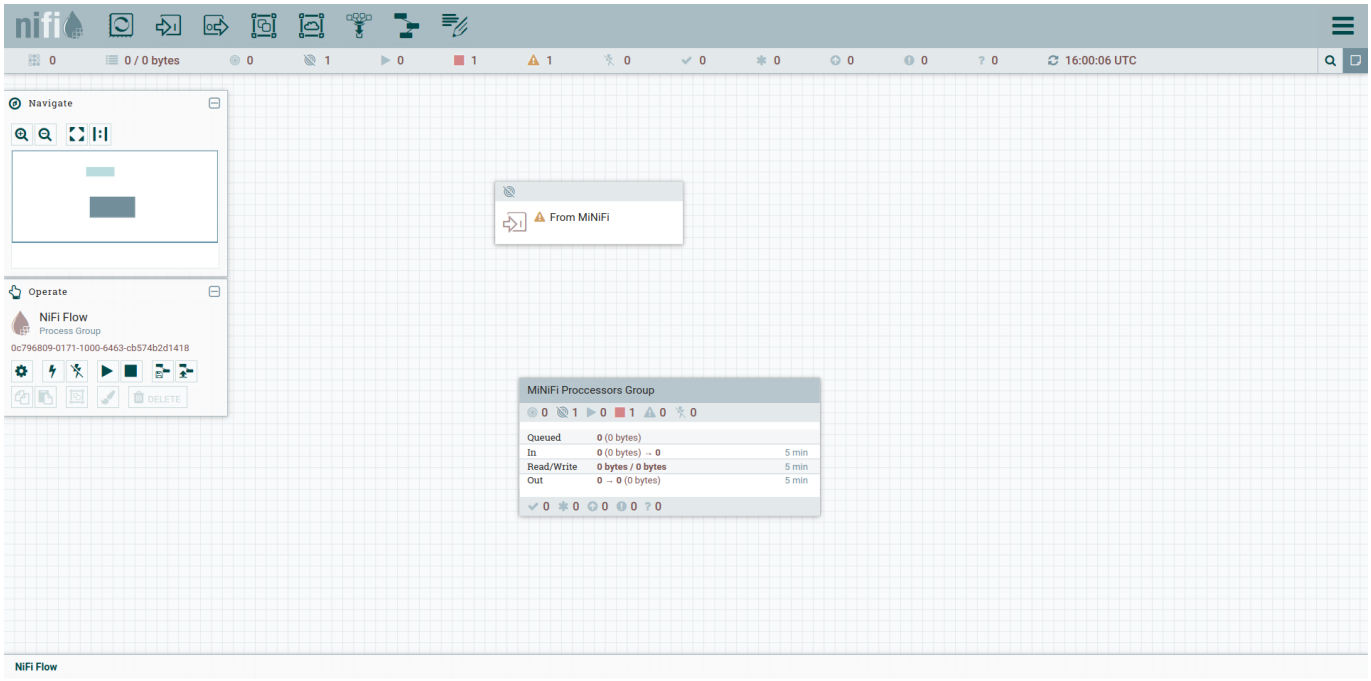

Рис.4.1.: Основные элементы шаблона

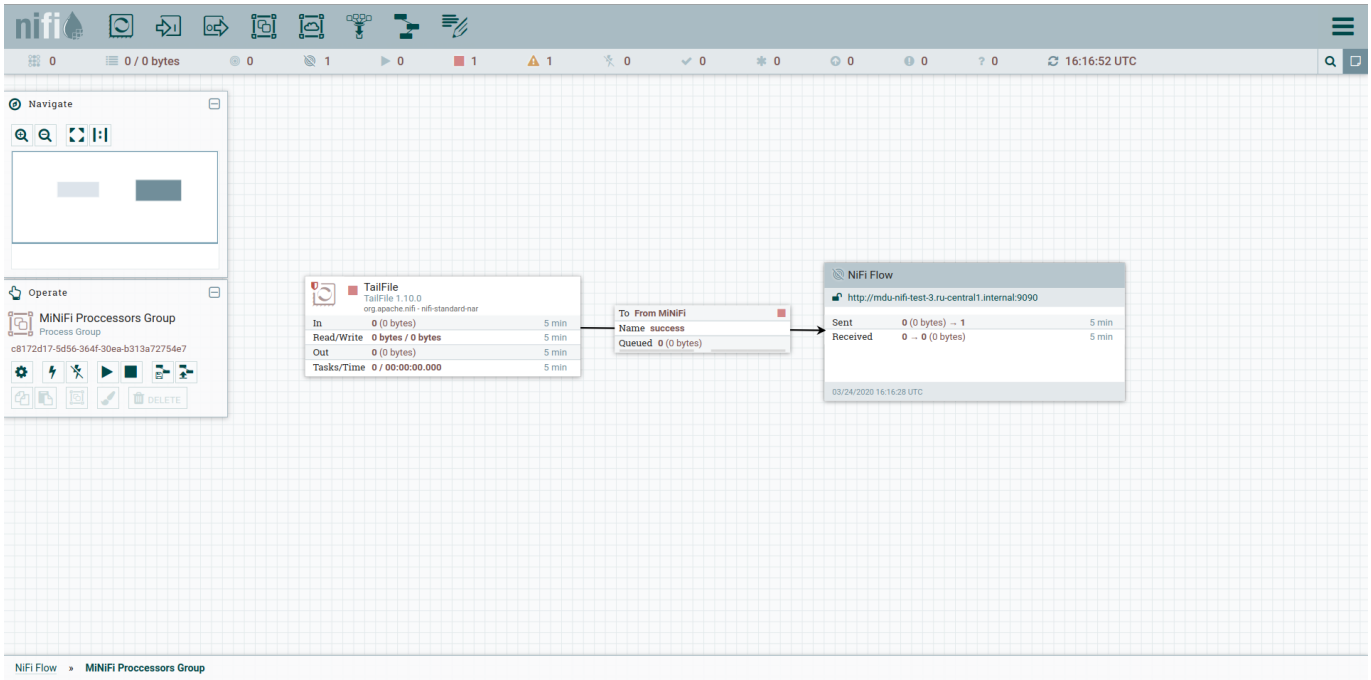

Рис.4.2.: Flow Агентов

```
MiNiFi Config Version: 3
Flow Controller:
  name: minifi.v1
  comment: ''
Core Properties:
  flow controller graceful shutdown period: 10 sec
  flow service write delay interval: 500 ms
  administrative yield duration: 30 sec
  bored yield duration: 10 millis
  max concurrent threads: 1
  variable registry properties: ''
FlowFile Repository:
  partitions: 256
  checkpoint interval: 2 mins
  always sync: false
  Swap:
    threshold: 20000
    in period: 5 sec
    in threads: 1
    out period: 5 sec
    out threads: 4
Content Repository:
  content claim max appendable size: 10 MB
  content claim max flow files: 100
  always sync: false
Provenance Repository:
  provenance rollover time: 1 min
  implementation: org.apache.nifi.provenance.MiNiFiPersistentProvenanceRepository
Component Status Repository:
  buffer size: 1440
  snapshot frequency: 1 min
Security Properties:
  keystore: ''
  keystore type: ''
  keystore password: ''
  key password: ''
  truststore: ''
  truststore type: ''
  truststore password: ''
  ssl protocol: ''
  Sensitive Props:
    key:
    algorithm: PBEWITHMD5AND256BITAES-CBC-OPENSSL
    provider: BC
Processors:
- id: 8805d9a2-4830-3033-0000-000000000000
  name: TailFile
```

Рис.4.3.: Результат обращения к MiNiFi C2 Server

Глава 5

Руководство администратора по работе с Nifi

В руководстве приведены сведения для администраторов системы по работе с платформой ADS в части сервиса Nifi – рекомендации по конфигурации, аутентификация пользователей с настройками их политик и доступа, Kerberos.

Руководство может быть полезно администраторам, программистам, разработчикам и сотрудникам подразделений информационных технологий, осуществляющих сопровождение платформы.

Important: Контактная информация службы поддержки – e-mail: info@arenadata.io

5.1 Рекомендации по конфигурации

При работе в **Linux** необходимо учесть последующие рекомендации, так как типичные значения по умолчанию для **Linux** могут быть не настроены под нужды такого интенсивного приложения с высоким уровнем ввода-вывода, как **NiFi**.

Important: При установке ADS через ADCM рекомендации из данного раздела применяются автоматически

5.1.1 Максимальное число дескрипторов

Сервис **NiFi** в любой момент может открыть очень большое количество файлов, поэтому необходимо увеличить лимиты, отредактировав файл */etc/security/limits.conf*. Например:

```
*  hard  nofile  50000
*  soft  nofile  50000
```

5.1.2 Maximum Forked Processes

NiFi может быть настроен для генерации существенного количества потоков. Для увеличения их допустимого числа необходимо отредактировать файл */etc/security/limits.conf*, например:

```
*  hard  nproc  10000
*  soft  nproc  10000
```

При этом дистрибутив может потребовать так же изменить значение в */etc/security/limits.d/90-nproc.conf*, тогда следует добавить в него:

```
*  soft  nproc  10000
```

5.1.3 Количество доступных портов TCP

Увеличение количества доступных портов TCP особенно важно в случае, если поток устанавливает и срывает большое количество сокетов за короткий промежуток времени:

```
sudo sysctl -w net.ipv4.ip_local_port_range="10000 65000"
```

5.1.4 Статус сокетов TIMED_WAIT

Для того, чтобы сокеты долго не задерживались, и при необходимости быстрой настройки и отключения новых сокетов следует изменить время нахождения сокетов в статусе *TIMED_WAIT* при их закрытии, например:

```
sudo sysctl -w net.ipv4.netfilter.ip_conntrack_tcp_timeout_time_wait="1"
```

5.1.5 Отключение swapping в Linux

Для **NiFi** сервисов, всегда находящихся в непрерывной работе, swapping не подходит. Поэтому, чтобы сообщить **Linux** об отключении подкачки, следует отредактировать файл */etc/sysctl.conf*, добавив строку:

```
vm.swappiness = 0
```

При этом для партиций, обрабатывающих различные NiFi-репозитории, необходимо отключить `atime`, что в результате приводит к увеличению производительности. Поэтому рекомендуется отредактировать файл */etc/fstab*, а для конкретных партиций добавить опцию `noatime`.

5.2 Настройка безопасности

В целях безопасности сервис **NiFi** предоставляет несколько различных параметров конфигурации. Наиболее важными являются свойства под заголовком *"security properties"* в файле *nifi.properties*.

Для безопасной работы должны быть установлены свойства, приведенные в таблице.

Таблица5.1.: Описание свойств безопасности NiFi

Свойство	Описание
nifi.security.keystore	Имя файла Keystore, содержащего закрытый ключ сервера
nifi.security.keystoreType	Тип Keystore. Должен быть либо PKCS12, либо JKS. JKS является предпочтительным типом, файлы PKCS12 загружаются библиотекой BouncyCastle
nifi.security.keystorePasswd	Пароль Keystore
nifi.security.keyPasswd	Пароль для сертификата в Keystore. Если значение не установлено, используется `nifi.security.keystorePasswd`
nifi.security.truststore	Имя файла Truststore для авторизации при подключении к NiFi. Защищенный инстанс без Truststore отклоняет все входящие подключения
nifi.security.truststoreType	Тип Truststore. Должен быть либо PKCS12, либо JKS. JKS является предпочтительным типом, файлы PKCS12 загружаются библиотекой BouncyCastle
nifi.security.truststorePasswd	Пароль Truststore
nifi.security.needClientAuth	Значение `true` требует пройти аутентификацию при подключении клиентов. Свойство используется протоколом кластера NiFi для подтверждения, что узлы в кластере аутентифицированы и имеют сертификаты, которым доверяют Truststores

После настройки перечисленных свойств можно разрешить доступ к пользовательскому интерфейсу через HTTPS вместо HTTP. Это достигается путем установки свойств *nifi.web.https.host* и *nifi.web.https.port*. Свойство *nifi.web.https.host* указывает, на каком хосте должен работать сервер. При необходимости доступности интерфейса HTTPS со всех сетевых интерфейсов следует использовать значение `0.0.0.0`. Для того, чтобы администраторы могли настраивать приложение для работы только на определенных сетевых интерфейсах, следует указать свойства *nifi.web.http.network.interface* и *nifi.web.https.network.interface*.

Important: При включении HTTPS необходимо исключить свойство *nifi.web.http.port*, так как NiFi поддерживает либо HTTP, либо HTTPS

Так же и с *nifi.security.needClientAuth* – веб-сервер может быть настроен на требование аутентификации на основе сертификатов у пользователей, обращающихся к интерфейсу. Для этого веб-сервер не должен поддерживать аутентификацию имени пользователя и пароля с помощью протокола **LDAP** или **Kerberos**, так как любой из этих параметров настраивает проверку подлинности клиента на основе сертификатов, а у кого их нет, могут войти в систему под своими учетными данными или получить анонимный доступ. Но если аутентификация по имени пользователя и паролю и анонимный доступ не настроены, то веб-сервер запрашивает аутентификацию клиента на основе сертификата (см. Аутентификация пользователя).

После защиты пользовательского интерфейса следует обеспечить внутренние кластерные коммуникации и связь между сайтами. Это достигается установкой свойств *nifi.remote.input.secure* и *nifi.cluster.protocol.is.secure* в значение `true`.

5.2.1 Набор средств генерации TLS

Для упрощения установки **NiFi** и автоматического создания необходимых хранилищ ключей, доверительного хранилища и соответствующих файлов конфигурации можно использовать утилиту командной строки **tls-toolkit**, что так же обеспечит безопасность многочисленных узлов **NiFi**.

Wildcard-сертификаты (т. е. два узла *node1.nifi.apache.org* и *node2.nifi.apache.org*, которым назначается тот же сертификат с записью CN или SAN *.nifi.apache.org*) официально не поддерживаются и не рекомендуются. Их использование имеет множество недостатков и приемлемо только, если каждый сертификат поддерживает дополнительную уникальную запись SAN и запись CN.

Потенциальные проблемы использования wildcard-сертификатов:

- Кластерные связи многократно используют идентификаторы сертификатов для определения узла, а если сертификат представляет собой подстановочное DN, то он не даст ответа;
- Администраторам может потребоваться предоставить кастомный идентификатор узла в *authorizers.xml* для *.nifi.apache.org*, поскольку все действия прокси-сервера разрешаются только в сертификате DN (см. Аутентификация пользователя);
- Администраторы не имеют возможности отслеживать, в каком узле выполняется действие, так как все они направляются в один и тот же DN;
- Администраторы, запускающие несколько инстансов на одном компьютере и используя разные порты для их идентификации, могут случайно поместить узел *node1* с портом *node2*, и адрес будет в итоге удален, потому что он использует тот же сертификат, а обработчик узла блокирует его, так как имя узла *node1* не указано в качестве допустимого хоста для инистанса *node2*;
- Если wildcard-сертификат скомпрометирован, все узлы оказываются под угрозой.

Important: Для keystores и truststores в NiFi рекомендуются JKS. Этот инструмент позволяет задавать другие типы хранилищ ключей в командной строке и игнорировать тип PKCS12 для использования в качестве доверительного хранилища, потому что данный формат имеет проблемы совместимости между реализациями BouncyCastle и Oracle

Инструмент командной строки **tls-toolkit** имеет два основных режима работы:

- *Standalone* (автономный) – создает организацию сертификатов, хранилища ключей, доверительные хранилища и файлы *nifi.properties* в одной команде;
- *Client/Server* (Клиент/Сервер) – использует Certificate Authority Server, который принимает запросы на подписание сертификатов от клиентов, подписывает и отправляет обратно. И клиент, и сервер проверяют идентификацию друг друга через общий секрет.

Standalone

Автономный режим вызывается запуском `./bin/tls-toolkit.sh standalone -h` и отображает информацию об использовании с описаниями опций, которые могут быть указаны.

В автономном режиме с **tls-toolkit** можно использовать следующие параметры командной строки:

- `-a`, `--keyAlgorithm <arg>` – алгоритм использования сгенерированных ключей (по умолчанию: *RSA*);
- `-B`, `--clientCertPassword <arg>` – пароль сертификата клиента. Должно быть либо одно значение, либо одно для каждого DN клиента (если не задано, генерируется автоматически);
- `-c`, `--certificateAuthorityHostname <arg>` – имя хоста NiFi Certificate Authority (по умолчанию: *localhost*);
- `-C`, `--clientCertDn <arg>` – создание сертификата клиента, подходящего для использования в браузере, с указанным DN (может быть указан несколько раз);
- `-d`, `--days <arg>` – количество дней, в течение которых выданный сертификат является действительным (по умолчанию: *1095*);
- `-f`, `--nifiPropertiesFile <arg>` – базовый файл *nifi.properties* для обновления (если не указан, используется встроенный файл, идентичный файлу по умолчанию при установке **NiFi**);
- `-g`, `--differentKeyAndKeystorePasswords` – использование другого сгенерированного пароля для ключа и хранилища ключей;

- `-G`, `--globalPortSequence <arg>` – использование последовательных портов, которые вычисляются для всех хостов в соответствии с предоставленными выражениями имен хостов (могут быть указаны несколько раз, но должны быть одинаковыми от запуска до запуска);
- `-h`, `--help` – печать справки и выход;
- `-k`, `--keySize <arg>` – количество бит для генерации ключей (по умолчанию: *2048*);
- `-K`, `--keyPassword <arg>` – пароль ключа. Либо одно значение, либо одинаковое для каждого хоста (если не задано, генерируется автоматически);
- `-n`, `--hostnames <arg>` – список имен хостов через запятую;
- `--nifiDnPrefix <arg>` – строка для добавления имени хоста (в начало) при определении DN (по умолчанию: *CN=*);
- `--nifiDnSuffix <arg>` – строка для добавления имени хоста (в конец) при определении DN (по умолчанию: *OU=NIFI*);
- `-o`, `--outputDirectory <arg>` – каталог для вывода keystore, truststore и config-файлов (по умолчанию: *../bin*);
- `-O`, `--isOverwrite` – перезапись существующего вывода хоста;
- `-P`, `--trustStorePassword <arg>` – пароль truststore. Либо одно значение, либо одинаковое для каждого хоста (если не задано, генерируется автоматически);
- `-s`, `--signingAlgorithm <arg>` – алгоритм подписи сертификатов (по умолчанию: *SHA256WITHRSA*);
- `-S`, `--keyStorePassword <arg>` – пароль keytstore. Либо одно значение, либо одинаковое для каждого хоста (если не задано, генерируется автоматически);
- `--subjectAlternativeNames <arg>` – разделенный запятыми список доменов для использования в качестве альтернативных имен в сертификате;
- `-T`, `--keyStoreType <arg>` – тип создаваемого хранилища ключей (по умолчанию: *jks*).

Шаблоны имен хостов:

- Для указания диапазона имен хостов используются квадратные скобки, например: `[01-20]`;
- Круглые скобки используются для определения, что на хосте (хостах) работает больше, чем один инстанс **NiFi**, например: `(5)`.

Примеры:

- Создать 4 набора хранилищ ключей, truststore, nifi.properties для localhost вместе с сертификатом клиента с предоставленным DN:

```
bin/tls-toolkit.sh standalone -n 'localhost(4)' -C 'CN=username,OU=NIFI'
```

- Создать хранилище ключей, truststore, nifi.properties для 10 имен хостов NiFi в каждом из 4 поддоменов:

```
bin/tls-toolkit.sh standalone -n 'nifi[01-10].subdomain[1-4].domain'
```

- Создать 2 набора хранилищ ключей, truststore, nifi.properties для 10 имен хостов NiFi в каждом из 4 поддоменов вместе с сертификатом клиента с предоставленным DN:

```
bin/tls-toolkit.sh standalone -n 'nifi[01-10].subdomain[1-4].domain(2)' -C 'CN=username,OU=NIFI'
```

Client/Server

Режим Клиент/Сервер опирается на Центр сертификации (Certificate Authority, CA) для выдачи сертификатов. Центр можно остановить, если узлы не подключены к сети.

Server

Сервер CA вызывается запуском `./bin/tls-toolkit.sh -h`, который печатает информацию об использовании с описаниями опций, которые могут быть заданы.

В режиме сервера с **tls-toolkit** можно использовать следующие параметры командной строки:

- `-a`, `--keyAlgorithm <arg>` – алгоритм использования сгенерированных ключей (по умолчанию: *RSA*);
- `--configJsonIn <arg>` – место для чтения информации о конфигурации, подразумевает *useConfigJson*, если установлено (по умолчанию: значение *configJson*);
- `-d`, `--days <arg>` – количество дней, в течение которых выданный сертификат является действительным (по умолчанию: *1095*);
- `-D`, `--dn <arg>` – DN для сертификата CA (по умолчанию: *CN=YOUR_CA_HOSTNAME,OU=NIFI*);
- `-f`, `--configJson <arg>` – место записи информации о конфигурации (по умолчанию: *config.json*);
- `-F`, `--useConfigJson` – флаг, указывающий, что вся конфигурация считывается из *configJson* (для облегчения автоматического использования, иначе в *configJson* производится только запись);
- `-g`, `--differentKeyAndKeystorePasswords` – использование другого сгенерированного пароля для ключа и хранилища ключей;
- `-h`, `--help` – печать справки и выход;
- `-k`, `--keySize <arg>` – количество бит для генерации ключей (по умолчанию: *2048*);
- `-p`, `--PORT <arg>` – порт для прослушивания центром сертификации (по умолчанию: *8443*);
- `-s`, `--signingAlgorithm <arg>` – алгоритм подписи сертификатов (по умолчанию: *SHA256WITHRSA*);
- `-T`, `--keyStoreType <arg>` – тип создаваемого хранилища ключей (по умолчанию: *jks*);
- `-t`, `--token <arg>` – маркер для предотвращения MITM (должен быть таким же, как тот, что используется клиентами).

Client

Клиент может использоваться для запроса новых сертификатов из центра сертификации. Утилита клиента генерирует пару ключей и запрос подписи сертификата (CSR, Certificate Signing Request), после чего отправляет CSR в центр сертификации. Клиент вызывается запуском `./bin/tls-toolkit.sh client -h`, который печатает информацию об использовании с описаниями опций, которые могут быть заданы.

В режиме клиента с **tls-toolkit** можно использовать следующие параметры командной строки:

- `-a`, `--keyAlgorithm <arg>` – алгоритм использования сгенерированных ключей (по умолчанию: *RSA*);
- `-c`, `--certificateAuthorityHostname <arg>` – имя хоста NiFi Certificate Authority (по умолчанию: *localhost*);
- `-C`, `--certificateDirectory <arg>` – каталог записи сертификата CA (по умолчанию: .);
- `--configJsonIn <arg>` – место для чтения информации о конфигурации, подразумевает *useConfigJson*, если установлено (по умолчанию: значение *configJson*);
- `-D`, `--dn <arg>` – DN для сертификата клиента (по умолчанию: *CN=<localhost name>,OU=NIFI*, заполняется автоматически инструментом);
- `-f`, `--configJson <arg>` – место записи информации о конфигурации (по умолчанию: *config.json*);
- `-F`, `--useConfigJson` – флаг, указывающий, что вся конфигурация считывается из *configJson* (для облегчения автоматического использования, иначе в *configJson* производится только запись);

- `-g`, `--differentKeyAndKeystorePasswords` – использование другого сгенерированного пароля для ключа и хранилища ключей;
- `-h`, `--help` – печать справки и выход;
- `-k`, `--keySize <arg>` – количество бит для генерации ключей (по умолчанию: *2048*);
- `-p`, `--PORT <arg>` – порт для прослушивания центром сертификации (по умолчанию: *8443*);
- `--subjectAlternativeNames <arg>` – разделенный запятыми список доменов для использования в качестве альтернативных имен в сертификате;
- `-T`, `--keyStoreType <arg>` – тип создаваемого хранилища ключей (по умолчанию: *jks*);
- `-t`, `--token <arg>` – маркер для предотвращения MITM (должен быть таким же, как тот, что используется клиентами).

В результате запуска клиента предоставляется сертификат CA, keystore, truststore и config.json с информацией о них, а также их пароли.

Сертификат клиента можно легко импортировать в браузер, указав: `-T PKCS12`.

5.3 Аутентификация пользователя

NiFi поддерживает аутентификацию пользователей через сертификаты клиента, через имя пользователя и пароль, через **Apache Knox** или через OpenId Connect.

Проверка подлинности имени пользователя и пароля выполняется с помощью "Идентификатора входа в систему" ("Login Identity Provider") – это подключаемый механизм для аутентификации пользователей через их имя и пароль, настройка которого осуществляется в файле *nifi.properties*. В настоящее время **NiFi** предлагает проверку имени пользователя и пароля с параметрами Provider Identity Provider для **LDAP** и **Kerberos**.

Свойство *nifi.login.identity.provider.configuration.file* указывает файл конфигурации для Идентификатора входа в систему. Свойство *nifi.security.user.login.identity.provider* указывает, какой из настроенных Login Identity Provider должен использоваться. По умолчанию свойство не настроено, что означает, что username/password должно быть явно включено.

При аутентификации через **OpenId Connect** сервер **NiFi** перенаправляет пользователей для проверки подлинности в Провайдер, а затем **NiFi** вызывает Провайдер для получения идентификации пользователя.

При аутентификации через **Apache Knox** сервер **NiFi** перенаправляет пользователей для проверки подлинности в **Apache Knox**, а затем **NiFi** во время аутентификации пользователя проверяет токен **Apache Knox**.

Important: Аутентификация пользователя в NiFi может быть настроена только по username/password, OpenId Connect или Apache Knox. Сервер не поддерживает одновременный запуск каждого из них. При этом NiFi потребуются сертификаты клиентов для аутентификации пользователей через HTTPS, если ничто иное не настроено

К защищенному инстансу **NiFi** нельзя получить доступ анонимно, если в **LDAP** или **Kerberos** не настроен Login Identity Provider, который, в свою очередь, должен быть настроен на явное разрешение анонимного доступа. При этом анонимный доступ в настоящее время невозможен по умолчанию в *FileAuthorizer* (NIFI-2730).

Important: NiFi не выполняет аутентификацию пользователя через HTTP (используя HTTP, всем пользователям предоставляются все роли)

5.3.1 Lightweight Directory Access Protocol (LDAP)

Далее приведен пример с описанием настроек Login Identity Provider, который интегрируется с Directory Server для аутентификации пользователей.

```
<provider>
    <identifier>ldap-provider</identifier>
    <class>org.apache.nifi.ldap.LdapProvider</class>
    <property name="Authentication Strategy">START_TLS</property>

    <property name="Manager DN"></property>
    <property name="Manager Password"></property>

    <property name="TLS - Keystore"></property>
    <property name="TLS - Keystore Password"></property>
    <property name="TLS - Keystore Type"></property>
    <property name="TLS - Truststore"></property>
    <property name="TLS - Truststore Password"></property>
    <property name="TLS - Truststore Type"></property>
    <property name="TLS - Client Auth"></property>
    <property name="TLS - Protocol"></property>
    <property name="TLS - Shutdown Gracefully"></property>

    <property name="Referral Strategy">FOLLOW</property>
    <property name="Connect Timeout">10 secs</property>
    <property name="Read Timeout">10 secs</property>

    <property name="Url"></property>
    <property name="User Search Base"></property>
    <property name="User Search Filter"></property>

    <property name="Identity Strategy">USE_DN</property>
    <property name="Authentication Expiration">12 hours</property>
</provider>
```

С помощью данной конфигурации аутентификация имени пользователя и пароля может быть активирована путем ссылки на провайдер в *nifi.properties*:

```
nifi.security.user.login.identity.provider=ldap-provider
```

Таблица5.2.: Описание настроек Login Identity Provider для LDAP

Свойство	Описание
Authentication Strategy	Аутентификация подключения к LDAP-серверу. Возможные значения: ANONYMOUS, SIMPLE, LDAPS или START_TLS
Manager DN	DN менеджера, который используется для привязки к LDAP-серверу для поиска пользователей
Manager Password	Пароль менеджера, который используется для привязки к LDAP-серверу для поиска пользователей
TLS - Keystore	Путь к Keystore при подключении к LDAP с использованием LDAPS или START_TLS
TLS - Keystore Password	Пароль для Keystore при подключении к LDAP с использованием LDAPS или START_TLS
TLS - Keystore Type	Тип Keystore при подключении к LDAP с использованием LDAPS или START_TLS (то есть JKS или PKCS12)
TLS - Truststore	Путь к Truststore при подключении к LDAP с использованием LDAPS или START_TLS
TLS - Truststore Password	Пароль для Truststore при подключении к LDAP с использованием LDAPS или START_TLS
TLS - Truststore Type	Тип Truststore при подключении к LDAP с использованием LDAPS или START_TLS (то есть JKS или PKCS12)
TLS - Client Auth	Политика аутентификации клиента при подключении к LDAP с использованием LDAPS или START_TLS. Возможные значения: REQUIRED, WANT, NONE
TLS - Protocol	Протокол при подключении к LDAP с использованием LDAPS или START_TLS (TLS, TLSv1.1, TLSv1.2 и т.д.)
TLS - Shutdown Gracefully	Указывает, следует ли корректно завершать работу TLS перед закрытием целевого контекста. По умолчанию: false
Referral Strategy	Стратегия обработки рефералов. Возможные значения: FOLLOW, IGNORE, THROW
Connect Timeout	Время ожидания соединения (10 секунд)
Read Timeout	Время ожидания чтения (10 секунд)
Url	Разделенный пробелами список URL-адресов серверов LDAP (например, ldap://<hostname>:<port>)
User Search Base	Базовый DN для поиска пользователей (например, CN=Users,DC=example,DC=com)
User Search Filter	Фильтр для поиска пользователей в User Search Base (sAMAccountName={0}). Указанное пользователем имя вставляется в *{0}*
Identity Strategy	Стратегия идентификации пользователей. Возможные значения: USE_DN и USE_USERNAME. По умолчанию: - USE_DN (для сохранения обратной совместимости). USE_DN использует полный DN пользовательской записи (рекомендуется). USE_USERNAME использует имя пользователя, под которым он вошел в систему
Authentication Expiration	Продолжительность действия проверки подлинности пользователя. Если пользователь никогда не выходит
	из системы, он должен будет снова войти в систему в течение указанного времени

5.3.2 Kerberos

Далее приведен пример с описанием настроек Login Identity Provider, который интегрируется с Kerberos Key Distribution Center (KDC) для аутентификации пользователей.

```
<provider>
    <identifier>kerberos-provider</identifier>
    <class>org.apache.nifi.kerberos.KerberosProvider</class>
    <property name="Default Realm">NIFI.APACHE.ORG</property>
    <property name="Kerberos Config File">/etc/krb5.conf</property>
    <property name="Authentication Expiration">12 hours</property>
</provider>
```

С помощью данной конфигурации аутентификация имени пользователя и пароля может быть активирована путем ссылки на провайдер в *nifi.properties*:

```
nifi.security.user.login.identity.provider=kerberos-provider
```

Таблица5.3.: Описание настроек Login Identity Provider для Kerberos

Свойство	Описание
Default Realm	Область по умолчанию для предоставления пользователю в случае, если пользователь вводит неполный пользовательский принципал (например, NIFI.APACHE.ORG)
Kerberos Config File	Абсолютный путь к файлу конфигурации клиента Kerberos
Authentication Expiration	Продолжительность действия проверки подлинности пользователя. Если пользователь никогда не выходит из системы, он должен будет снова войти в систему в течение указанного времени

Описание настройки допуска по единому входу через клиентские тикеты Kerberos приведено в Kerberos Service.

5.3.3 OpenId Connect

Для включения аутентификации через OpenId Connect необходимо настроить свойства в *nifi.properties*, представленные далее в таблице.

Таблица5.4.: Описание настроек для аутентификации через OpenId Connect

Свойство	Описание
nifi.security.user.oidc.discovery.url	URL-адрес обнаружения необходимого OpenId Connect Provider (http://openid.net/specs/openid-connect-discovery-1_0.html)
nifi.security.user.oidc.connect.timeout	Время ожидания соединения при обмене данными с OpenId Connect Provider
nifi.security.user.oidc.read.timeout	Время ожидания чтения при обмене данными с OpenId Connect Provider
nifi.security.user.oidc.client.id	Идентификатор клиента для NiFi после регистрации в OpenId Connect Provider
nifi.security.user.oidc.client.secret	Секрет клиента для NiFi после регистрации в OpenId Connect Provider
nifi.security.user.oidc.preferred.jwsalgorithm	Предпочтительный алгоритм проверки токенов идентификации. Если значение свойства пустое, по умолчанию используется *RS 256*, поддерживаемое OpenId Connect Provider в соответствии со спецификацией. Если значение равно *HS256*, *HS384* или *HS512*, NiFi пытается проверить защищенные токены HMAC, используя указанный секретный ключ. Если значение свойства равно *none*, NiFi пытается проверить незащищенные/простые токены. Иные значения для алгоритма анализируются как алгоритм RSA или EC, который используется совместно с JSON Web Key (JWK), предоставленным через jwks_uri в метаданных URL-обнаружения

5.3.4 Apache Knox

Для включения аутентификации через **Apache Knox** необходимо настроить свойства в *nifi.properties*, представленные далее в таблице.

Таблица5.5.: Описание настроек для аутентификации через Apache Knox

Свойство	Описание
nifi.security.user.knox.url	URL-адрес страницы входа в Apache Knox
nifi.security.user.knox.publicKey	Путь к открытому ключу Apache Knox для проверки подписей токенов аутентификации в HTTP Cookie
nifi.security.user.knox.cookieName	Имя файла HTTP Cookie, которое Apache Knox создает после успешного входа в систему
nifi.security.user.knox.audiences	(Опционально) Разделенный запятыми список разрешенных подключений. Если значение задано, подключение должно присутствовать в списке. Разрешенные подключения, заполненные токеном, могут быть настроены в Knox

5.4 Настройка пользователей и политик доступа

В зависимости от характеристик настроенных параметров *UserGroupProvider* и *AccessPolicyProvider* пользователи, группы и политики могут конфигурироваться в пользовательском интерфейсе. Если расширения

не настроены, то в пользовательском интерфейсе пользователи, группы и политики доступны только для чтения. Если сконфигурированный авторизатор не использует *UserGroupProvider* и *AccessPolicyProvider*, пользователи и политики могут быть или не быть видимыми и настраиваемыми в пользовательском интерфейсе на основе базовой реализации.

Далее в главе предполагается, что пользователи, группы и политики настраиваются в пользовательском интерфейсе, и описывается:

- *Создание пользователей и групп*
- *Политики доступа*
- *Настройка политик доступа на основе конкретных примеров*

5.4.1 Создание пользователей и групп

В пользовательском интерфейсе необходимо выбрать “Users” в глобальном меню, при этом открывыается диалоговое окно для создания пользователей и групп и управления ими (Рис.5.1.). Для создания пользователей и групп используется кнопка “Add User”.

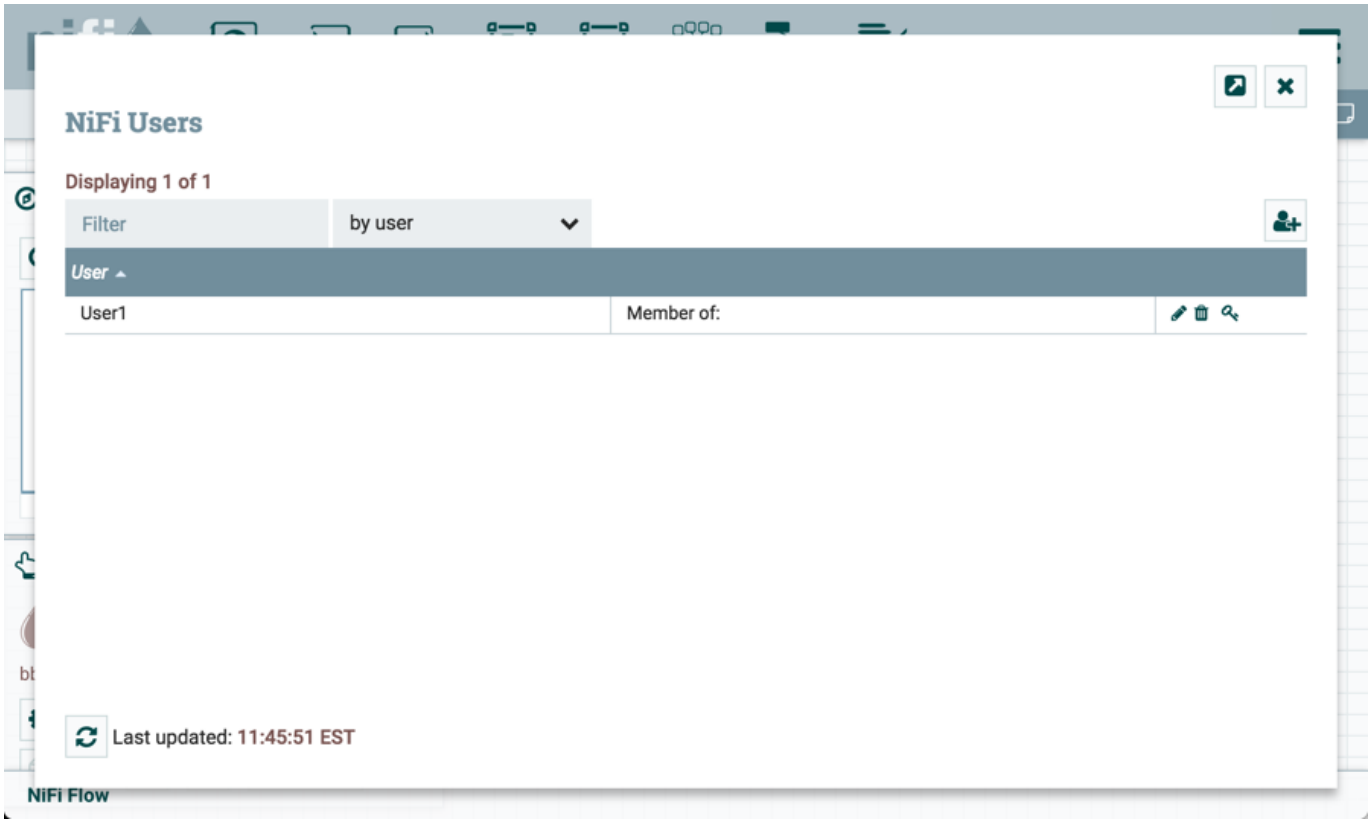

Рис.5.1.: Nifi Users

Для создания пользователя в новом открывшемся диалоговом окне необходимо выбрать “Individual” и ввести информацию “Identity”, соответствующую методу аутентификации защиты инстанса **NiFi**. После чего нажать “OK” (Рис.5.2.).

Для создания группы в диалоговом окне следует выбрать “Group”, ввести имя группы в поле “Identity” и отметить пользователей в “Members”, которые необходимо включить в группу. После чего нажать “OK” (Рис.5.3.).

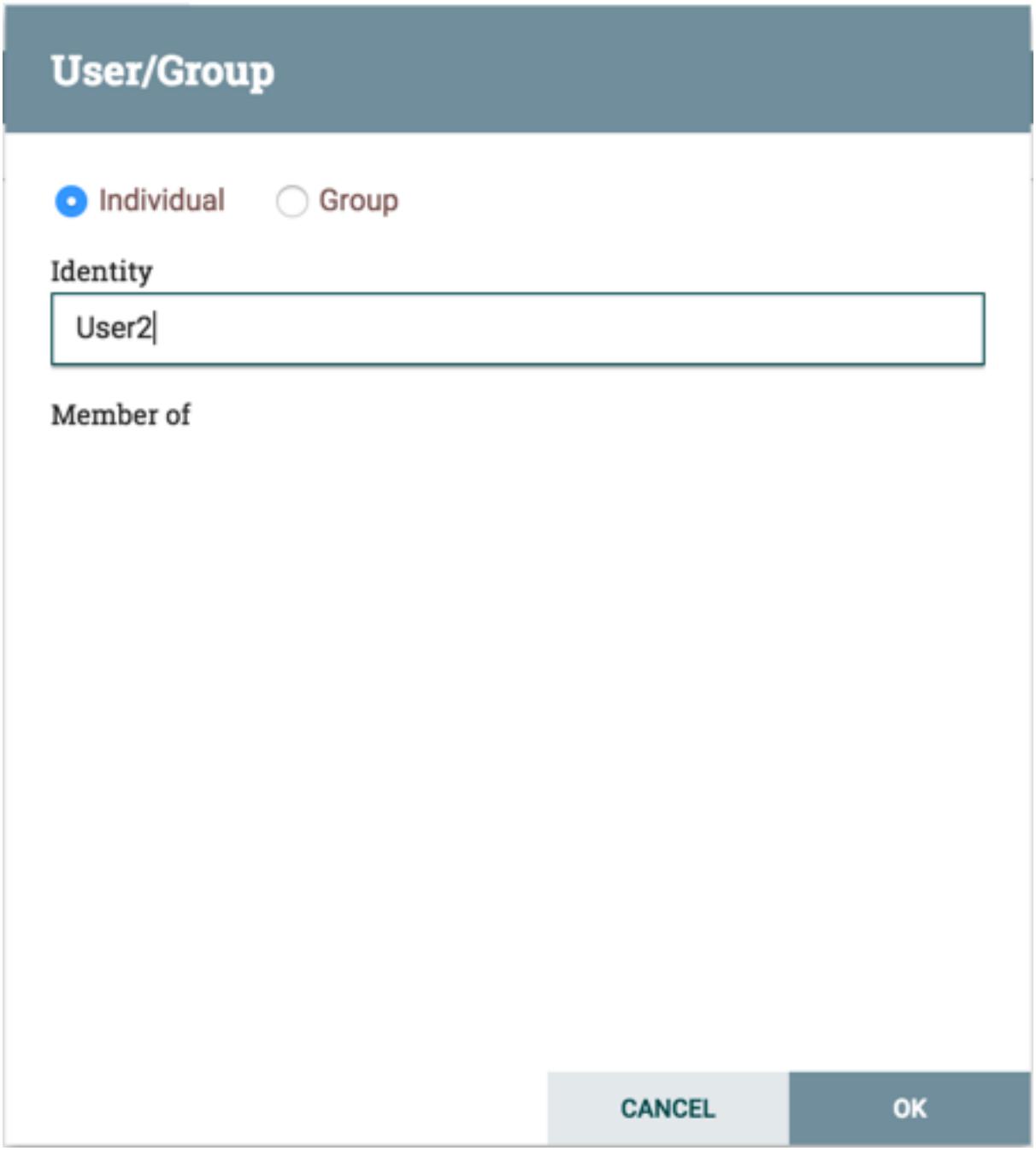

Рис.5.2.: Создание пользователя

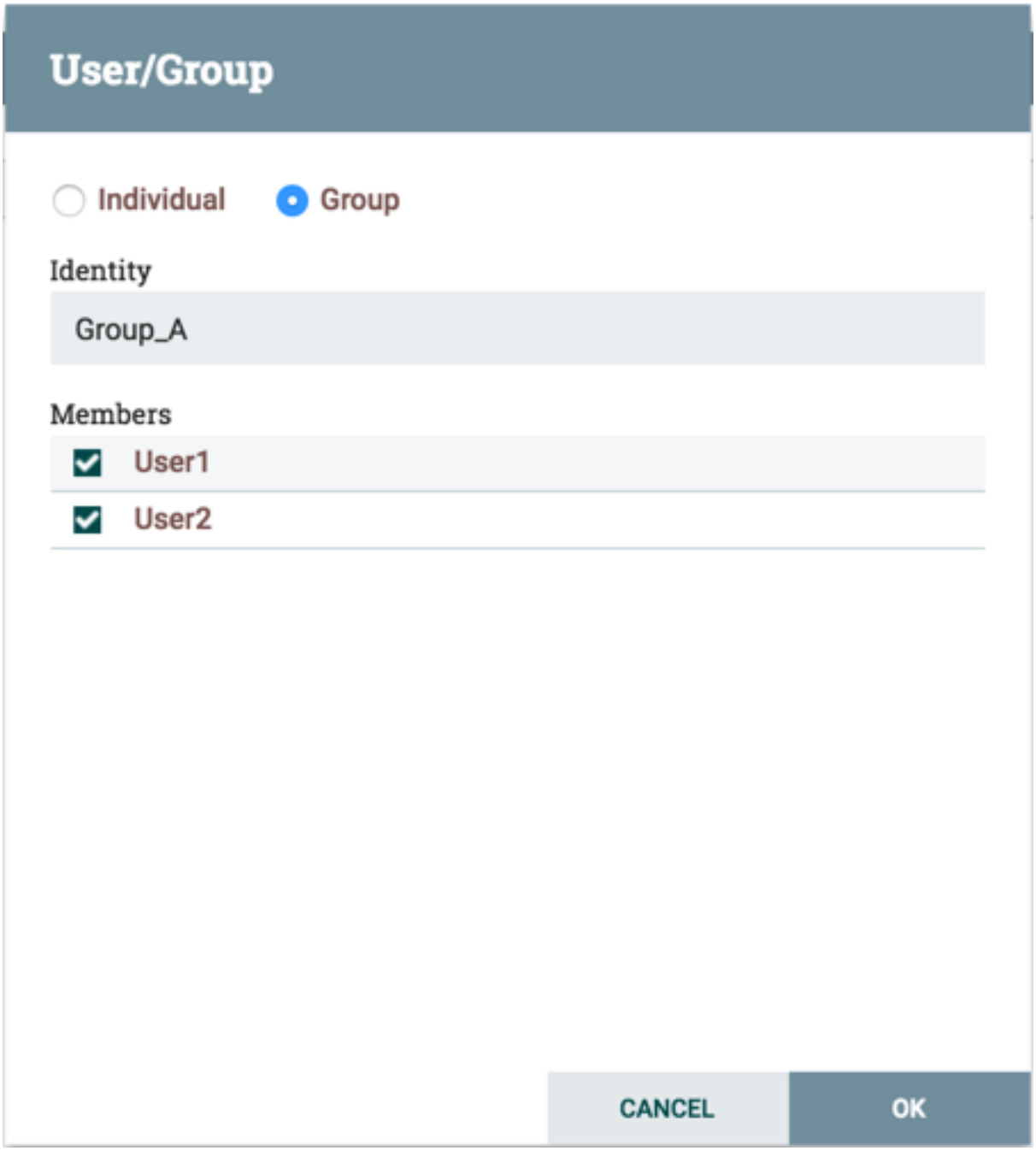

Рис.5.3.: Создание группы

5.4.2 Политики доступа

Управление возможностями пользователей и групп **NiFi** осуществляется с помощью политик доступа. Существует два типа политик доступа, которые могут быть применены к ресурсу:

- View (просмотр) – если ресурсу назначается политика просмотра, то добавленные в эту политику пользователи и группы могут только видеть детали данного ресурса;
- Modify (изменение) – если ресурсу назначается политика изменения, то добавленные в эту политику пользователи и группы могут изменить конфигурацию данного ресурса.

Создавать и применять политики доступа можно как на глобальном уровне (*Global Access Policies*), так и на уровне компонентов (*Component Level Access Policies*).

Global Access Policies

Политики глобального доступа управляют следующими полномочиями на уровне системы:

Таблица5.6.: Global Access Policies

Policy	Privilege	Global Menu Selection	Resource Descriptor
view the UI	Разрешение пользователям просматривать UI	N/A	/flow
access the controller	Позволяет пользователям просматривать/изменять контроллер, включая задачи отчетности, службы контроллеров и узлы в кластере	Controller Settings	/controller
query provenance	Позволяет пользователям отправлять Provenance Search и запрашивать Event Lineage	Data Provenance	/provenance
access restricted components	Позволяет пользователям создавать/изменять ограниченные компоненты при условии наличия других разрешений. Ограниченные компоненты могут указывать, какие конкретные разрешения требуются. Разрешения могут предоставляться для определенных ограничений или независимо от них. Если разрешение предоставляется независимо от ограничений, пользователь может создавать/изменять все ограниченные компоненты	N/A	/restricted-components
access all policies	Позволяет пользователям просматривать/изменять политики для всех компонентов	Policies	/policies
access users/user groups	Позволяет пользователям просматривать/изменять пользователей и группы пользователей	Users	/tenants
retrieve site-to-site details	Позволяет другим инстансам NiFi извлекать информацию site-to-site	N/A	/site-to-site
view system diagnostics	Позволяет пользователям просматривать системную диагностику	Summary	/system

Component Level Access Policies

Политики доступа на уровне компонентов управляют следующими полномочиями на уровне компонентов:

Таблица5.7.: Component Level Access Policies

Policy	Privilege	Resource Descriptor & Action
view the component	Позволяет пользователям просматривать детали конфигурации компонентов	resource=”/<component-type>/<component-UUID>” action=”R”
modify the component	Позволяет пользователям изменять детали конфигурации компонентов	resource=”/<component-type>/<component-UUID>” action=”W”
view provenance	Позволяет пользователям просматривать события происхождения, созданные компонентом	resource=”/provenance-data/<component-type>/<comp action=”R”
view the data	Позволяет пользователям просматривать метаданные и содержимое компонента в очередях потока в исходящих соединениях и через события происхождения	resource=”/data/<component-type>/<component-UUID action=”R”
modify the data	Позволяет пользователям очищать очереди потоков в исходящих соединениях и повторно отправлять через события происхождения	resource=”/data/<component-type>/<component-UUID action=”W”
view the policies	Позволяет пользователям просматривать список пользователей, которые могут просматривать/изменять компонент	resource=”/policies/<component-type>/<component-U action=”R”
modify the policies	Позволяет пользователям изменять список пользователей, которые могут просматривать/изменять компонент	resource=”/policies/<component-type>/<component-U action=”W”
receive data via site-to-site	Позволяет порту получать данные из инстансов NiFi	resource=”/data-transfer/input-ports/<port-UUID>” action=”W”
send data via site-to-site	Позволяет порту отправлять данные из инстансов NiFi	resource=”/data-transfer/output-ports/<port-UUID>” action=”W”

Important: Политики доступа можно применять ко всем типам компонентов, кроме соединений. Разрешения на соединения определяются по индивидуальным политикам доступа к исходному и целевому компонентам соединения, а так же по политике доступа группы процессов, содержащей компоненты. Более подробно это описано далее в примерах

Important: Для доступа к List Queue и Delete Queue для соединения пользователю требуются политики “view the data” и “modify the data” на компоненте. Так же все узлы в кластерной среде должны быть добавлены к этим политикам, так как запрос пользователя может быть реплицирован через любой узел в кластере

5.4.3 Настройка политик доступа на основе конкретных примеров

Самый эффективный способ понять, как создавать и применять политики доступа, – это пройтись по некоторым распространенным примерам. В приведенных далее сценариях *User1* является администратором, а *User2* – недавно добавленным пользователем, которому предоставлен доступ только к пользовательскому интерфейсу. На рисунке в качестве отправных точек показаны два процессора в рабочей области: GenerateFlowFile и LogAttribute (Рис.5.4.).

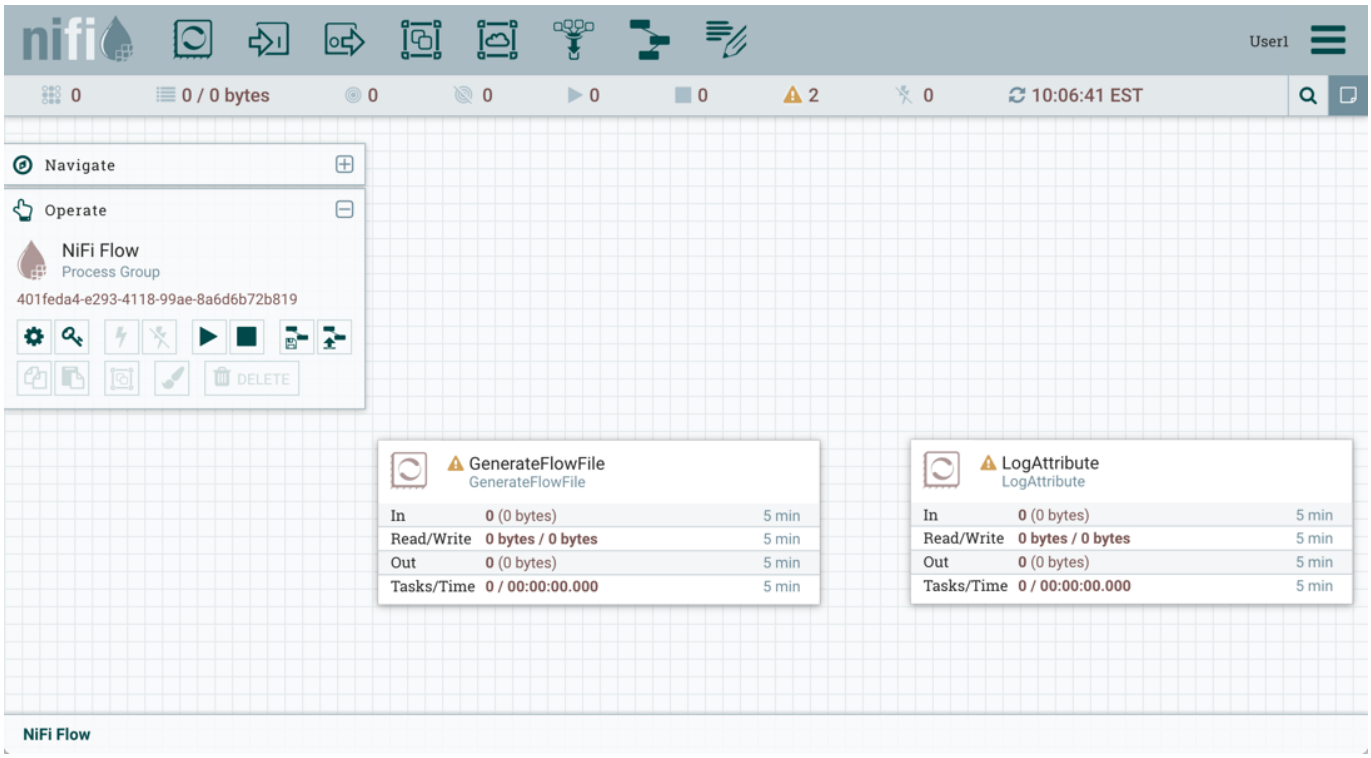

Рис.5.4.: GenerateFlowFile и LogAttribute

User1 может добавлять компоненты в поток данных, а так же перемещать, редактировать и подключать все процессоры. Детали и свойства процессоров и групп процессов root видны для *User1* (Рис.5.5.).

User2 не может добавлять компоненты в поток данных, а так же перемещать, редактировать и подключать компоненты. Детали и свойства процессоров и групп процессов root скрыты от *User2* (Рис.5.6.).

Перемещение процессора

User1 необходимо выполнить следующие шаги для выдачи разрешения пользователю *User2* на перемещение процессора GenerateFlowFile в потоке данных с сохранением привилегий у *User1*:

1. Выбрать процессор GenerateFlowFile.
2. Нажать значок “Access Policies” на панели управления “Operate”. При этом открывается диалоговое окно “Access Policies”.
3. Выбрать “modify the component” в раскрывающемся списке политики. Политика “modify the component”, которая в настоящее время существует на процессоре (дочернем), является унаследованной от группы процессов root (родительской), на которой *User1* имеет привилегии (Рис.5.7.).
4. Нажать ссылку “Override”. При замещении политики необходимо выбрать ее переопределение либо на копию унаследованной политики, либо на пустую политику. Для создания копии следует в диалоговом окне “Override Policy” выбрать “Copy” и нажать кнопку “Override” (Рис.5.8.).

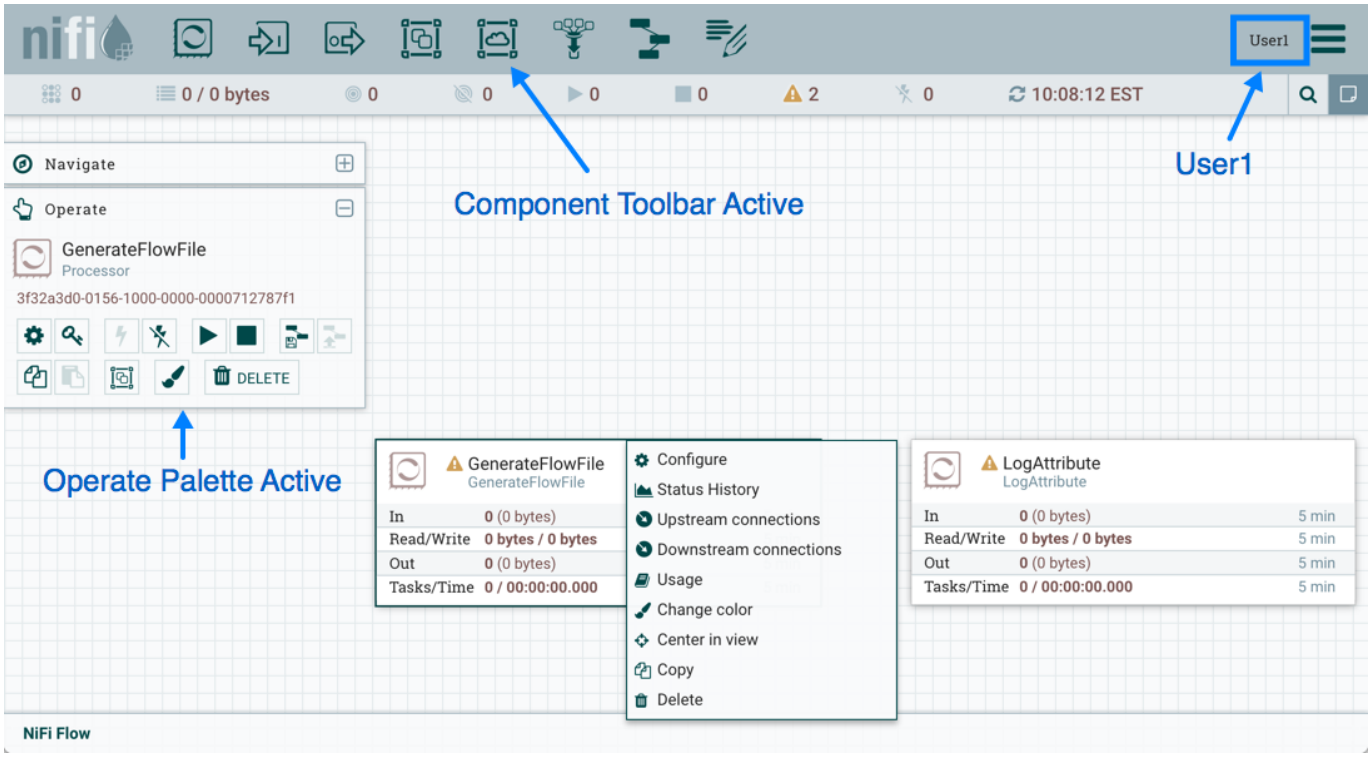

Рис.5.5.: User1 (администратор)

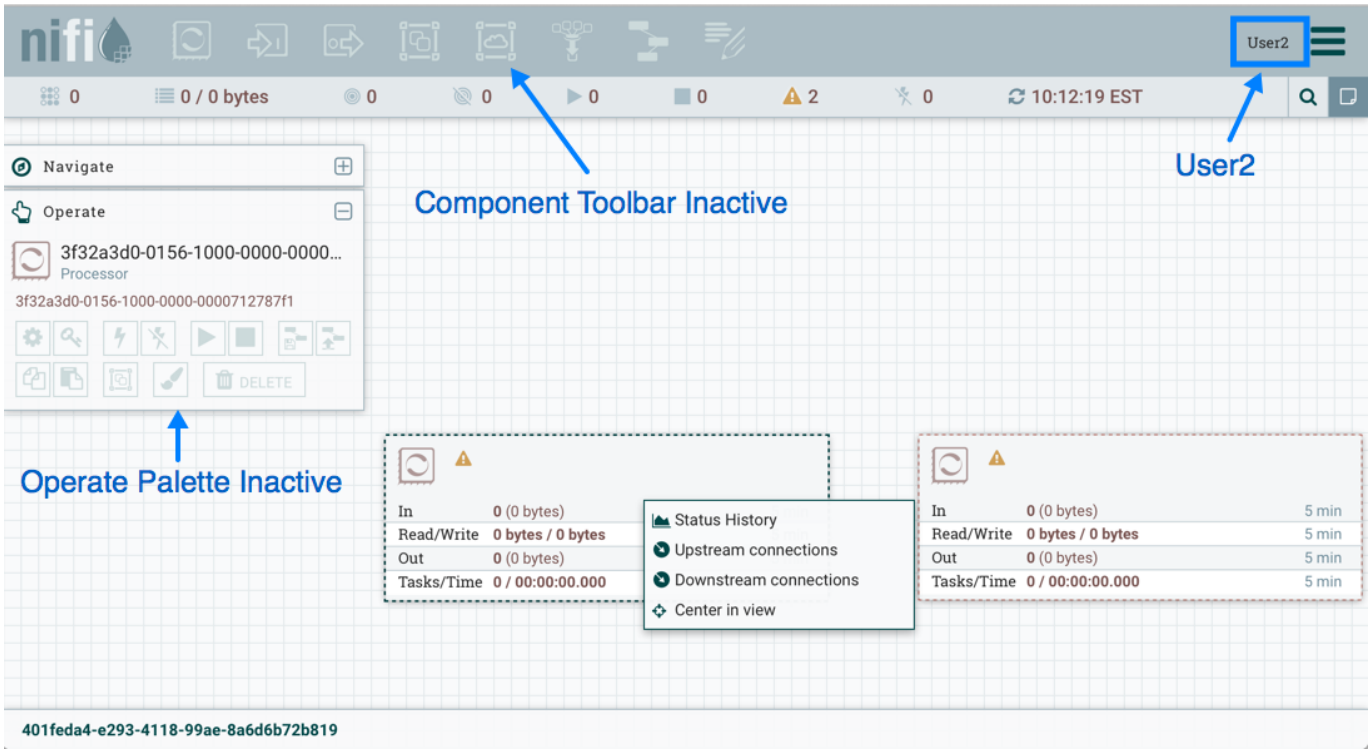

Рис.5.6.: User2 (недавно добавленный пользователь)

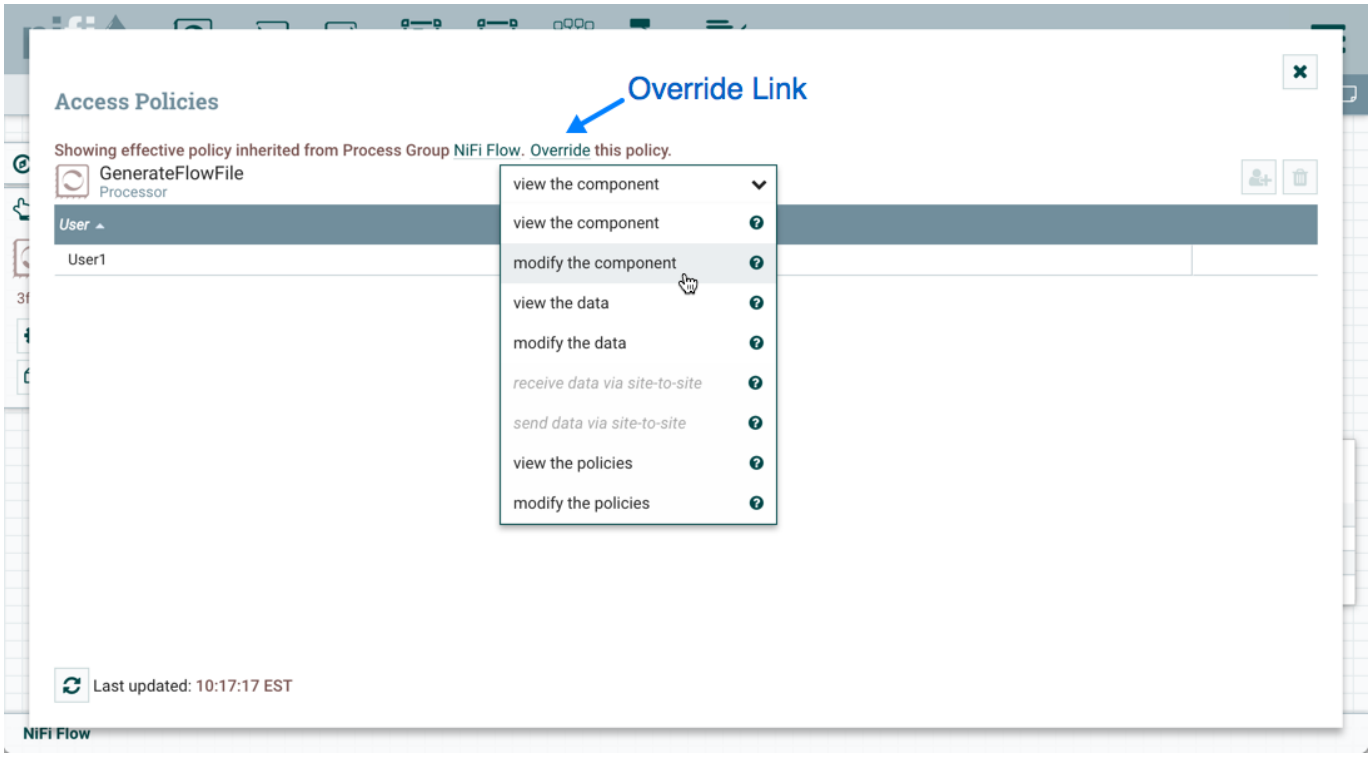

Рис.5.7.: Modify the component

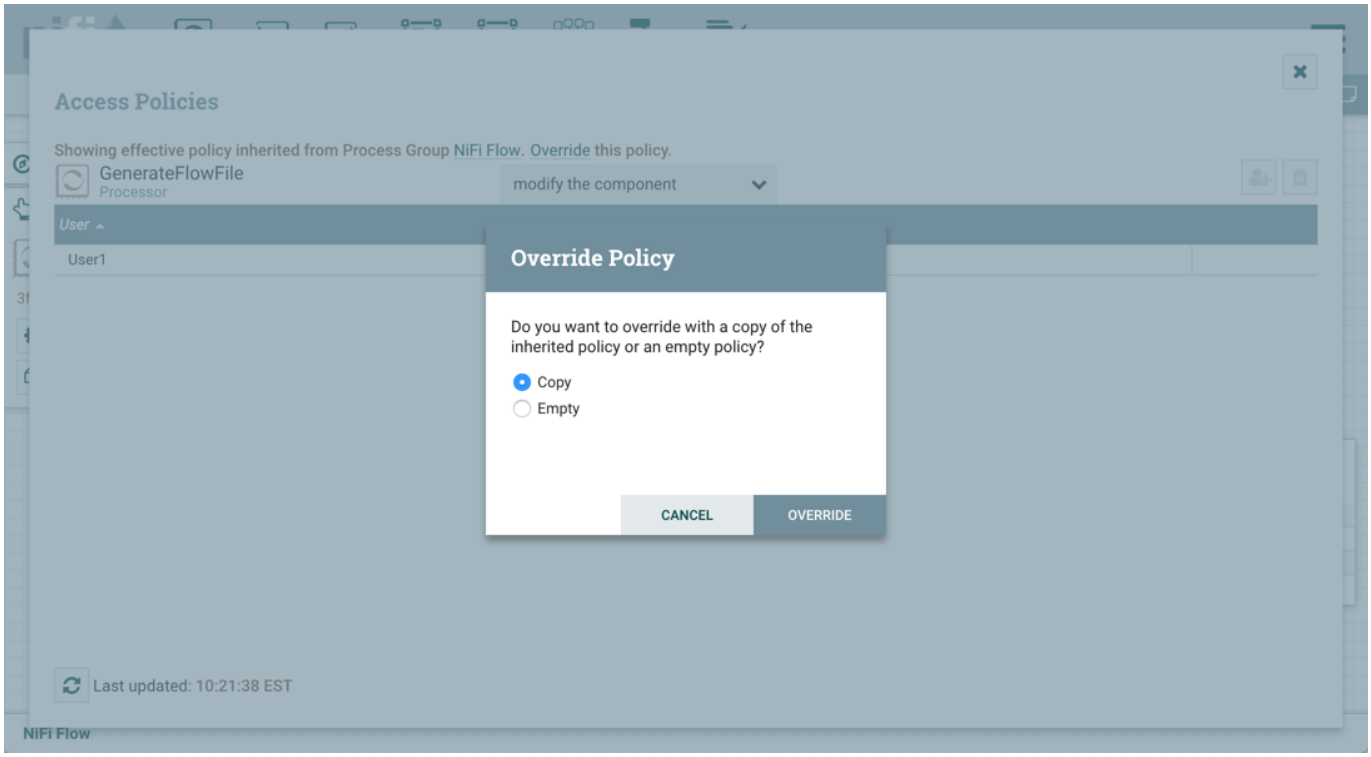

Рис.5.8.: Override Policy

5. В созданной политике выбрать значок “Add User”. В поле “User Identity” ввести вручную или найти в списке *User2* и нажать “OK” (Рис.5.9.).

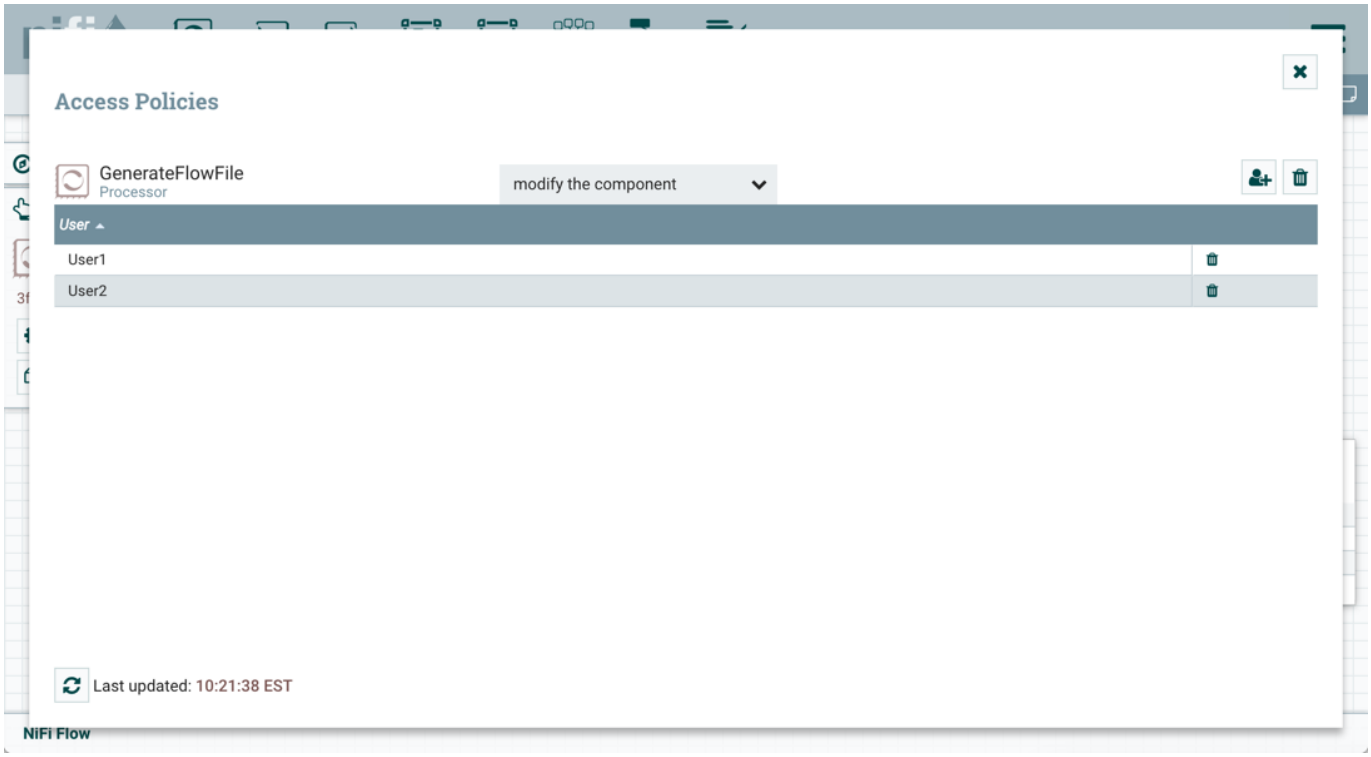

Рис.5.9.: Добавление User2 в политику

С такими изменениями *User1* сохраняет возможность перемещения обоих процессоров в рабочей области. А *User2* теперь может перемещать процессор GenerateFlowFile (Рис.5.10.).

Изменение процессора

В приведенном примере “Перемещение процессора” *User2* добавлен в политику “modify the component” для процессора GenerateFlowFile. Но без возможности просмотра свойств процессора *User2* не может изменять его конфигурацию – чтобы отредактировать компонент, пользователь должен быть также включен в политику “view the component”.

User1 необходимо выполнить следующие шаги для реализации возможности изменения конфигурации процессора пользователю *User2*:

1. Выбрать процессор GenerateFlowFile.
2. Нажать значок “Access Policies” на панели управления “Operate”. При этом открывается диалоговое окно “Access Policies”.
3. Выбрать “view the component” в раскрывающемся списке политики. Политика “view the component”, которая в настоящее время существует на процессоре (дочернем), является унаследованной от группы процессов root (родительской), на которой *User1* имеет привилегии (Рис.5.11.).
4. Нажать ссылку “Override” и в открывшемся диалоговом окне, сохранив политику копирования по умолчанию, нажать кнопку “Override”.
5. В созданной политике выбрать значок “Add User”. В поле “User Identity” ввести вручную или найти в списке *User2* и нажать “OK” (Рис.5.12.).

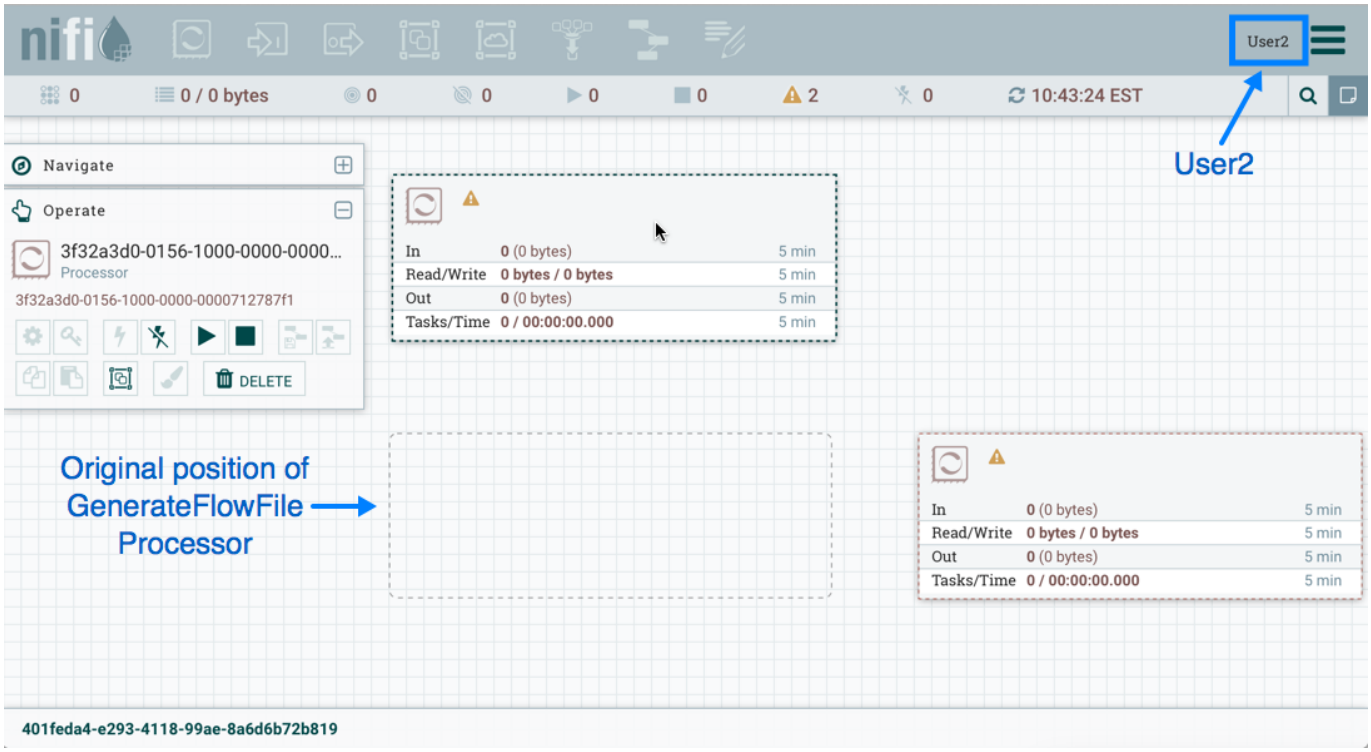

Рис.5.10.: Результат действий

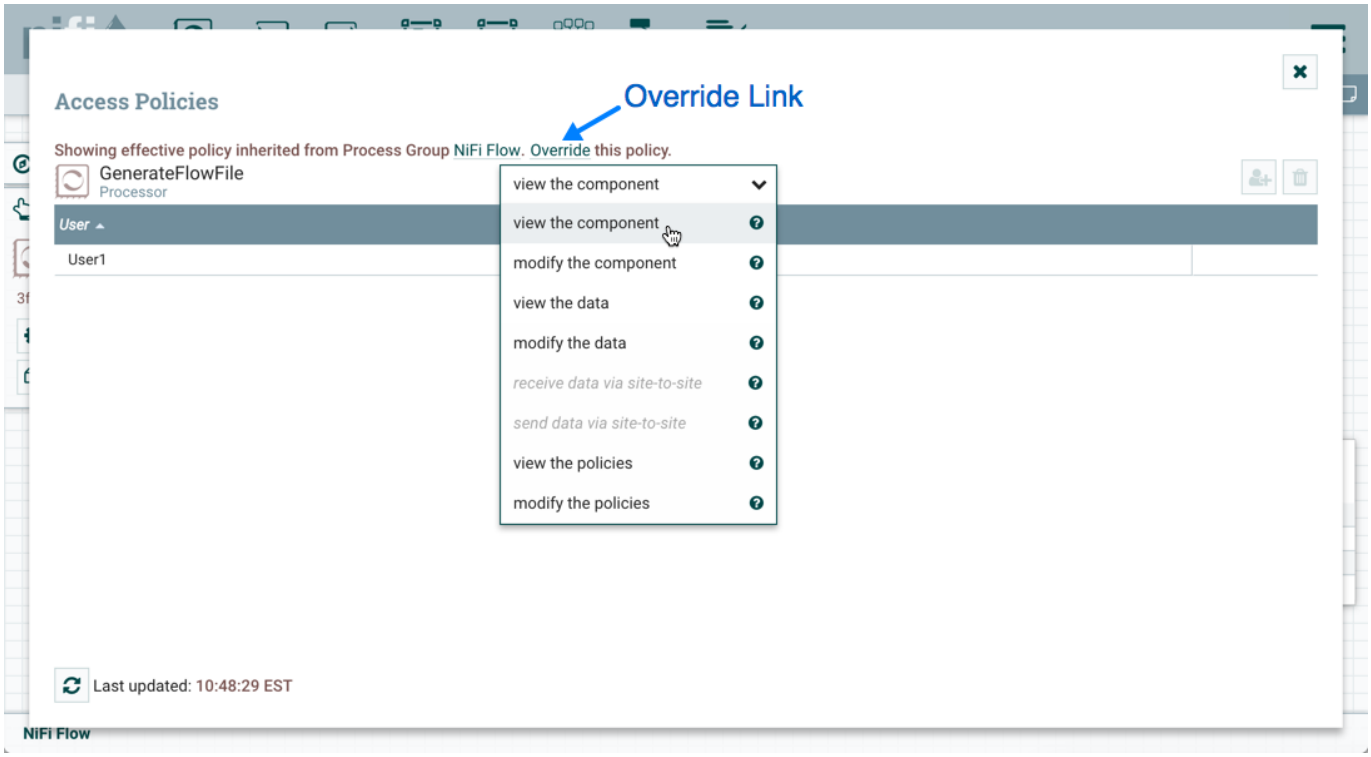

Рис.5.11.: View the component

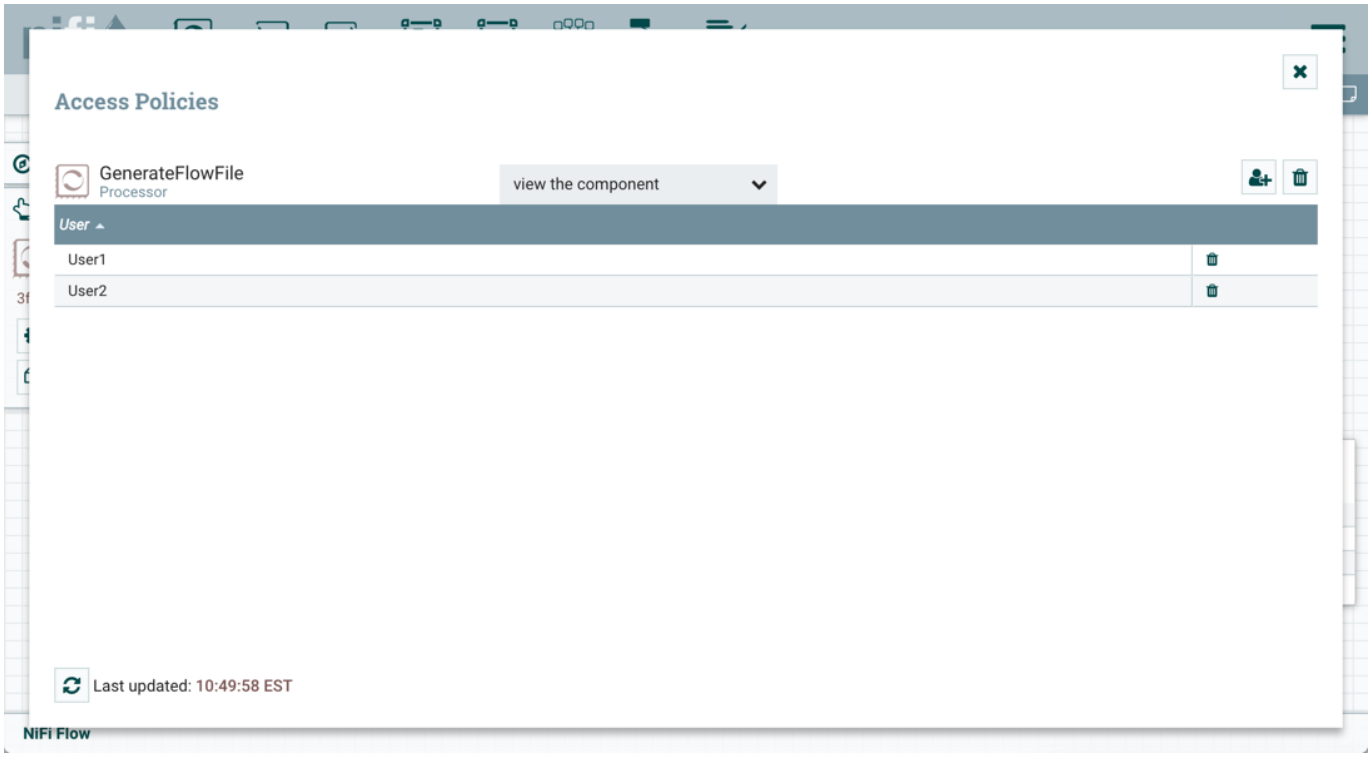

Рис.5.12.: Добавление User2 в политику

С такими изменениями *User1* сохраняет возможность просмотра и редактирования процессоров в рабочей области. А *User2* теперь может просматривать и редактировать процессор GenerateFlowFile (Рис.5.13.).

Создание подключения

При настройке политик так, как описано в предыдущих двух примерах, *User1* может подключить GenerateFlowFile к LogAttribute (Рис.5.14.).

При этом *User2* не имеет права доступа на установку соединения процессоров (Рис.5.15.).

Это объясняется тем, что:

- *User2* не имеет доступа к изменениям в группе процессов;
- Несмотря на то, что *User2* имеет право на просмотр и изменение исходного компонента (GenerateFlowFile), *User2* не имеет политики доступа к целевому компоненту (LogAttribute).

User1 необходимо выполнить следующие шаги для реализации возможности подключения GenerateFlowFile к LogAttribute пользователю *User2*:

1. Выбрать группу процессов root, при этом панель управления “Operate” обновляется с подробными сведениями.
2. Выбрать значок “Access Policies” на панели управления “Operate”. При этом открывается диалоговое окно “Access Policies”.
3. В диалоговом окне в раскрывающемся списке политики выбрать “modify the component” (Рис.5.16.).
4. Выбрать значок “Add User”. В поле “User Identity” ввести вручную или найти в списке *User2* и нажать “OK” (Рис.5.17.).

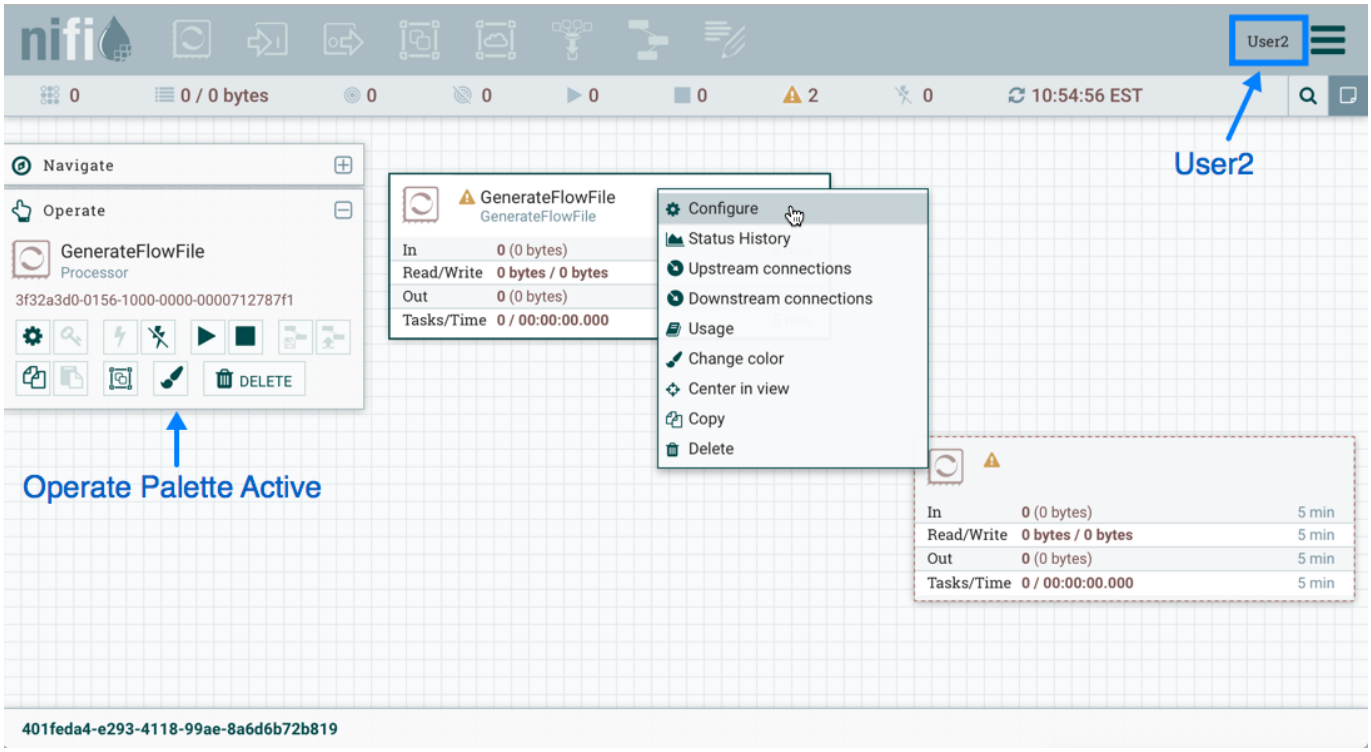

Рис.5.13.: Результат действий

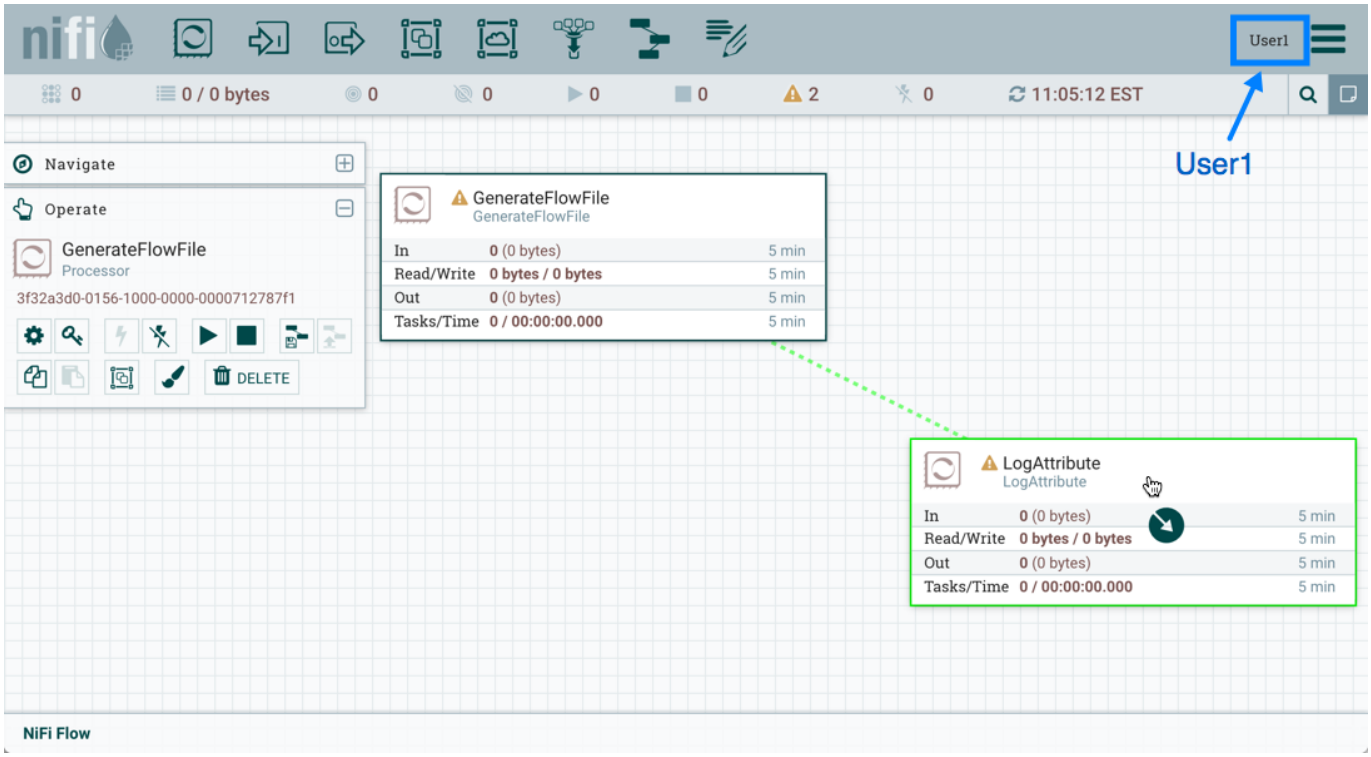

Рис.5.14.: User1 – подключение процессоров

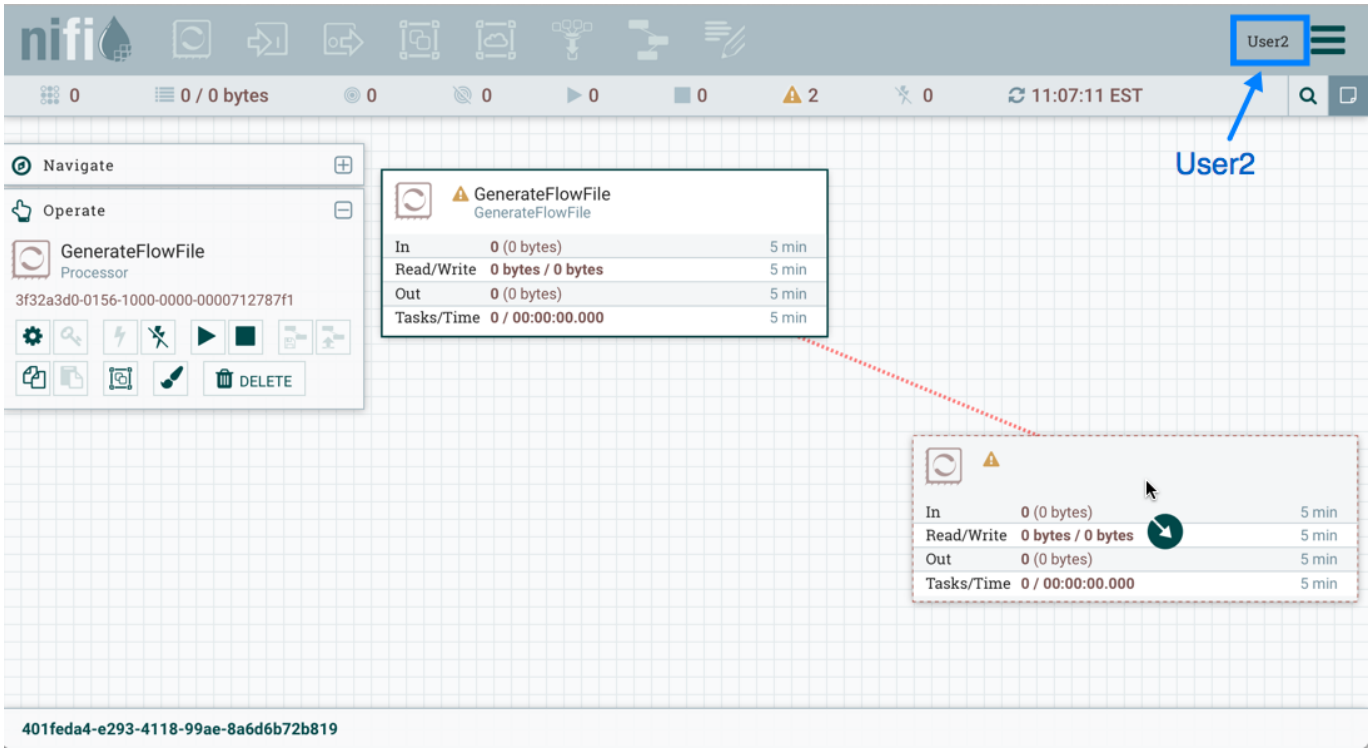

Рис.5.15.: User2 – невозможность подключения процессоров

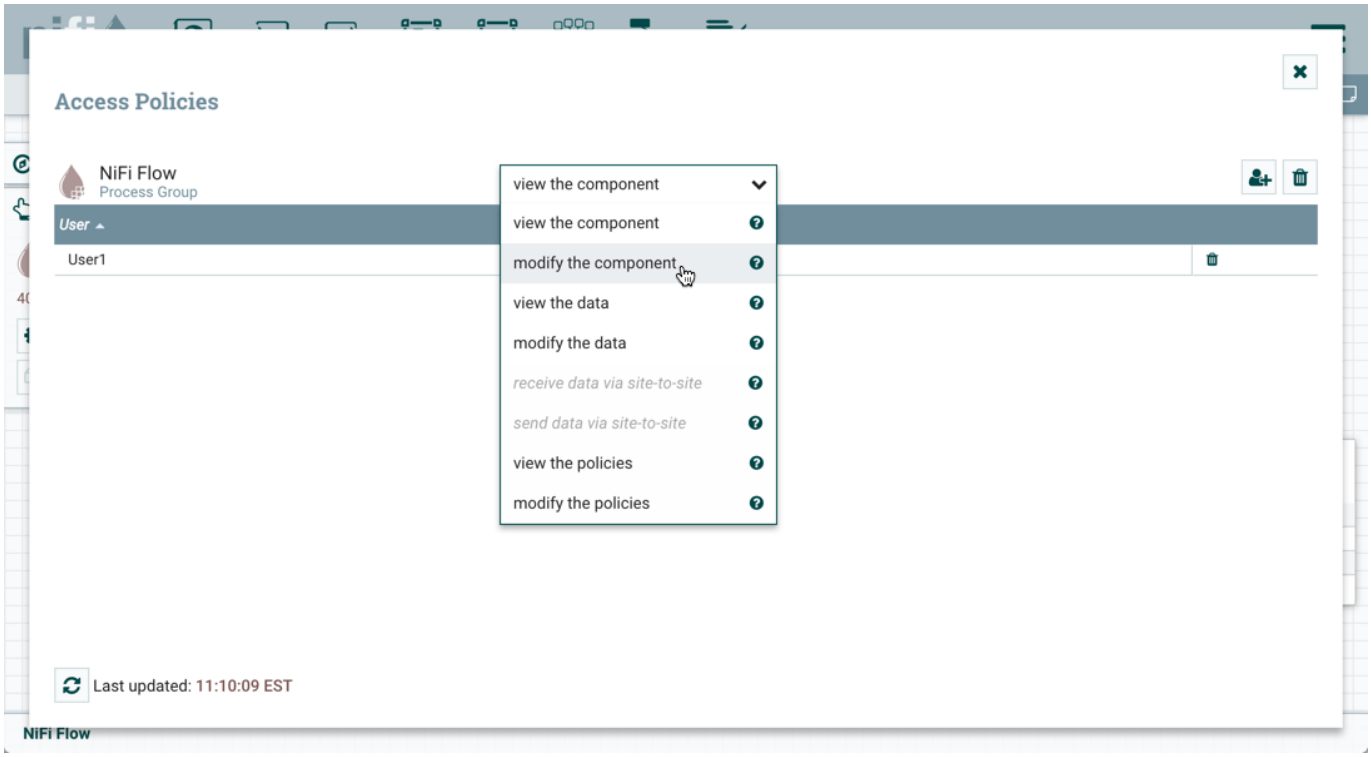

Рис.5.16.: Modify the component

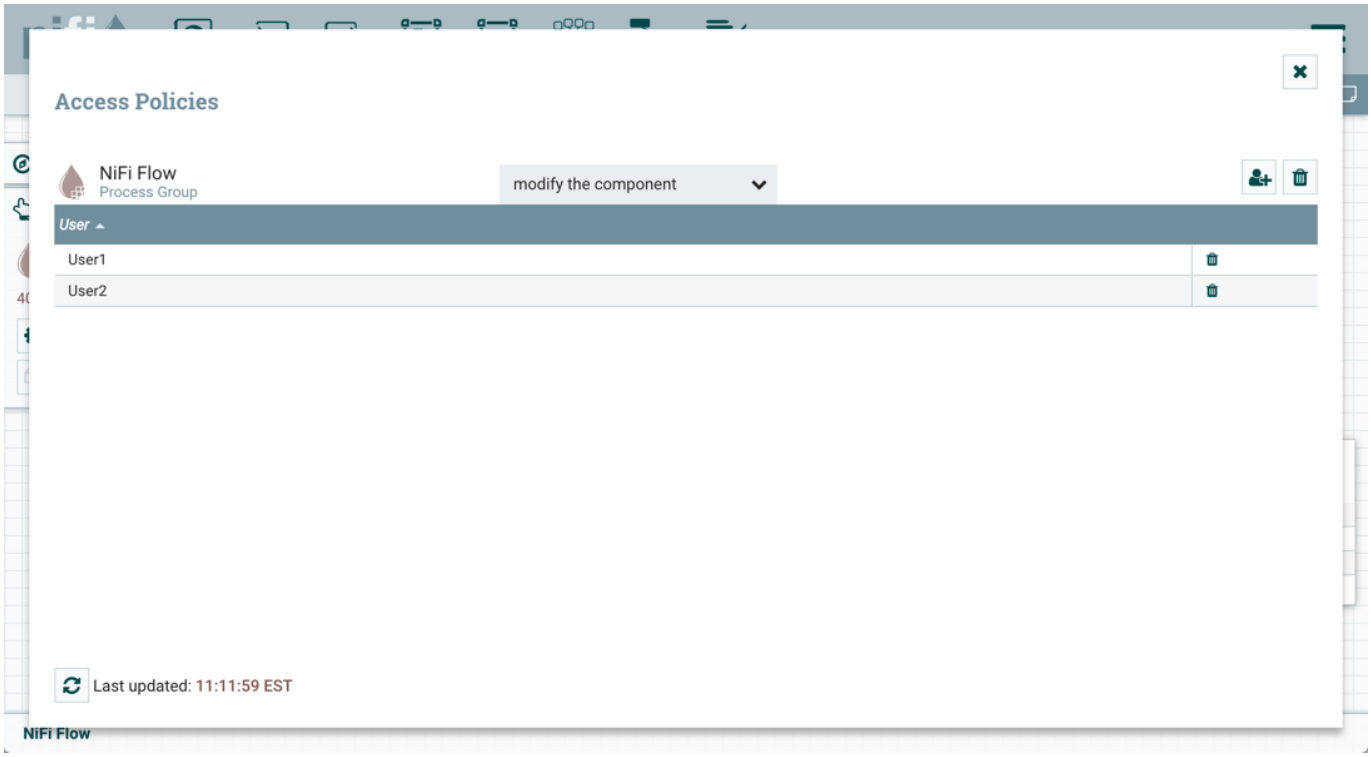

Рис.5.17.: Добавление User2 в политику группы

Добавляя *User2* в политику “modify the component” группы процессов, *User2* так же добавляется к политике “modify the component” в процессоре LogAttribute путем наследования. Чтобы проверить это, необходимо в рабочей области выделить процессор LogAttribute и выбрать значок “Access Policies” на панели управления “Operate”. При этом открывается диалоговое окно политик доступа процессора LogAttribute с наличием пользователя *User2* в политике “modify the component” (Рис.5.18.).

С такими изменениями *User2* теперь может подключать процессор GenerateFlowFile к процессору LogAttribute (Рис.5.19., Рис.5.20.).

Изменение соединения

В следующем сценарии *User1* и *User2* добавляют процессор ReplaceText в группу процессов root (Рис.5.21.).

User1 может выбрать и изменить существующее соединение между GenerateFlowFile и LogAttribute, чтобы подключить GenerateFlowFile к ReplaceText (Рис.5.22.).

При этом *User2* не имеет возможности выполнить такое действие (Рис.5.23.).

User1 необходимо выполнить следующие шаги для реализации возможности подключения GenerateFlowFile к ReplaceText пользователю *User2*:

1. Выбрать группу процессов root, при этом панель управления “Operate” обновляется с подробными сведениями.
2. Выбрать значок “Access Policies” на панели управления “Operate”. При этом открывается диалоговое окно “Access Policies”.
3. В диалоговом окне в раскрывающемся списке политики выбрать “view the component” (Рис.5.24.).

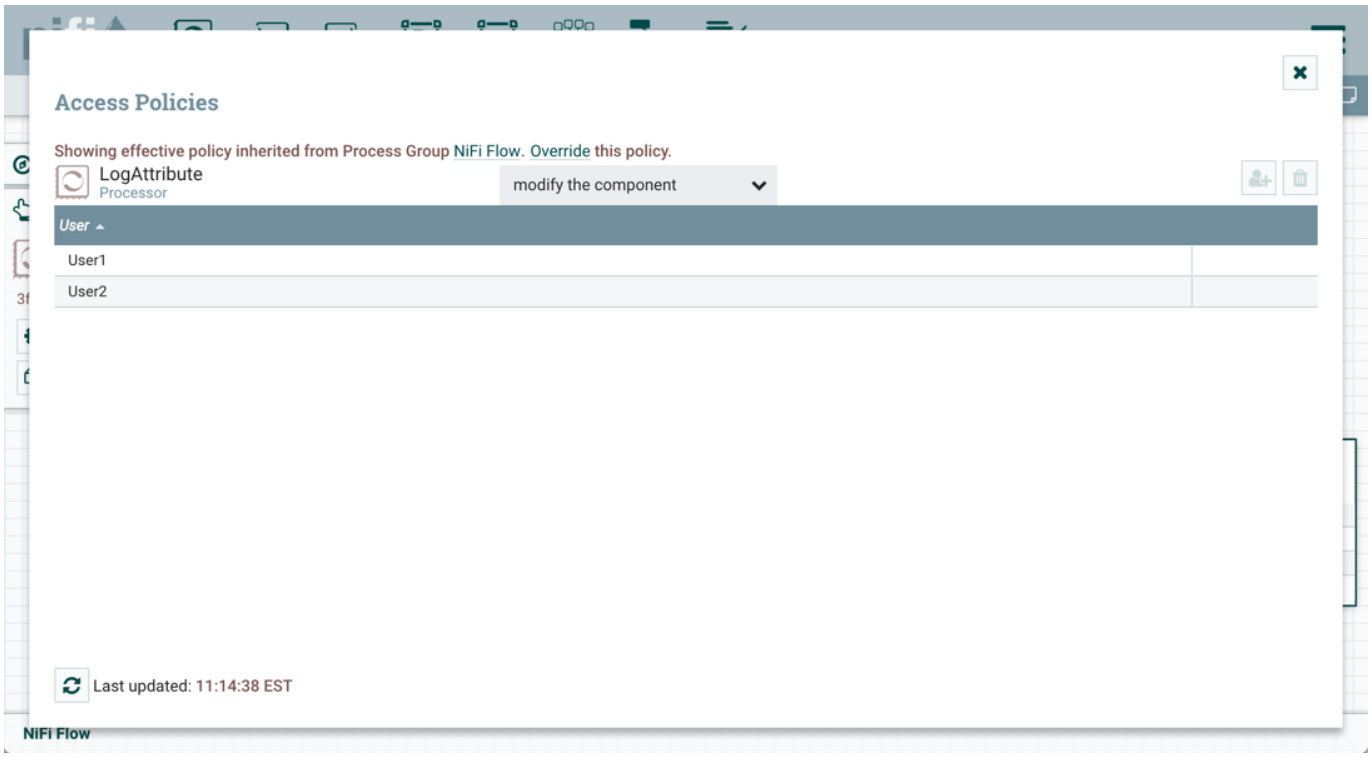

Рис.5.18.: Проверка наличия политики User2

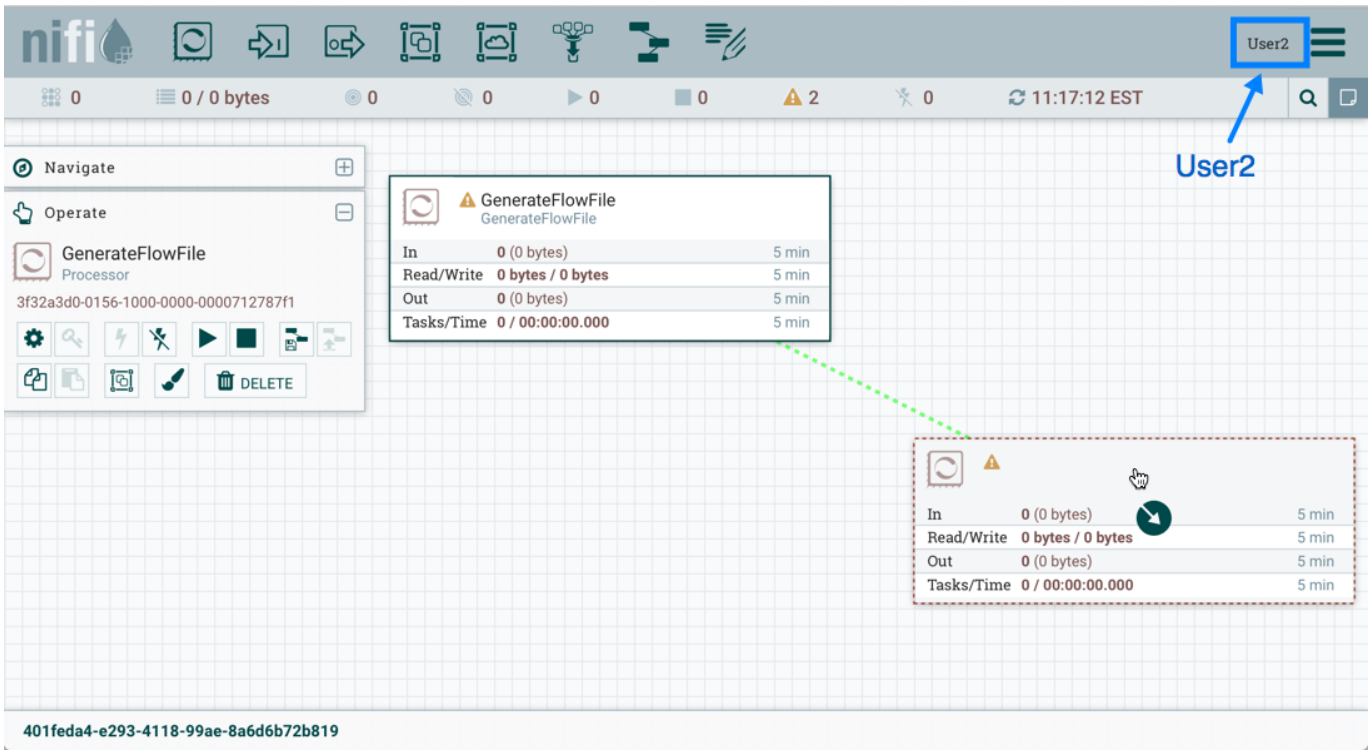

Рис.5.19.: User2 – подключение процессоров

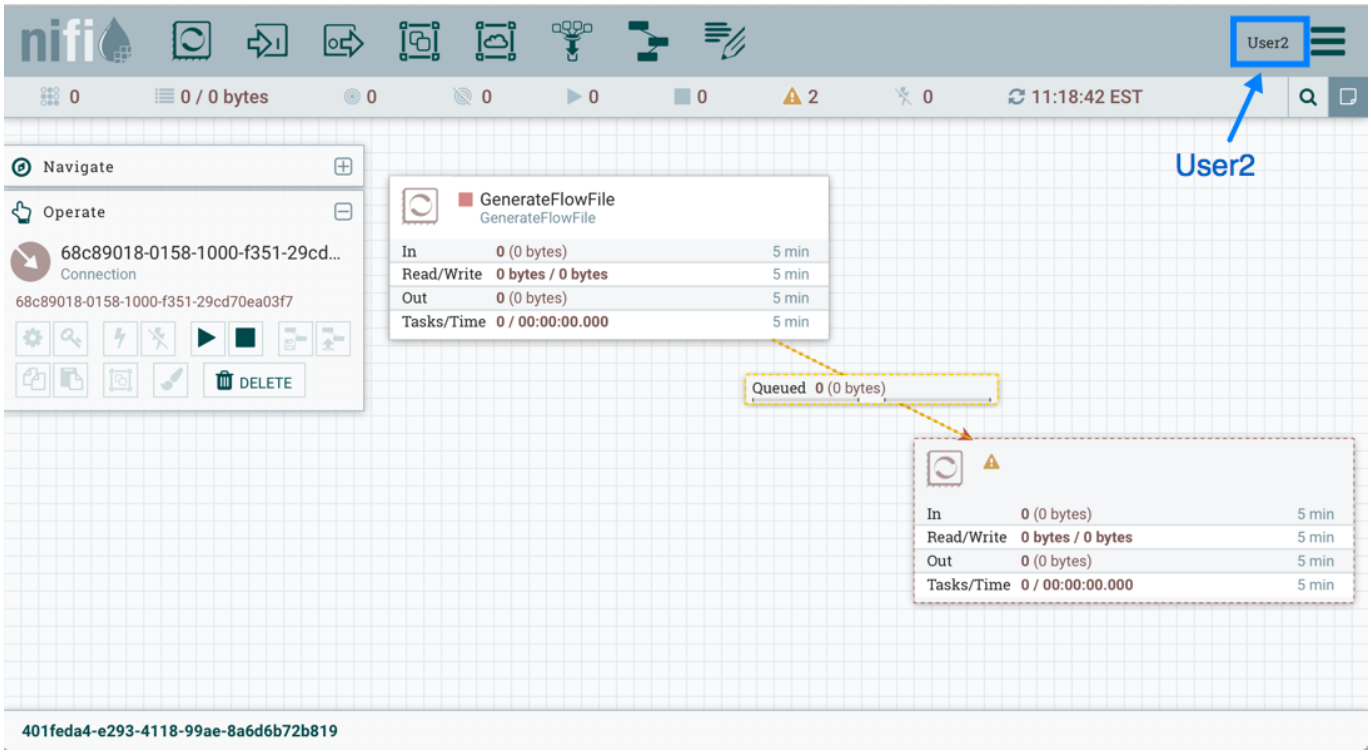

Рис.5.20.: User2 – подключение процессоров

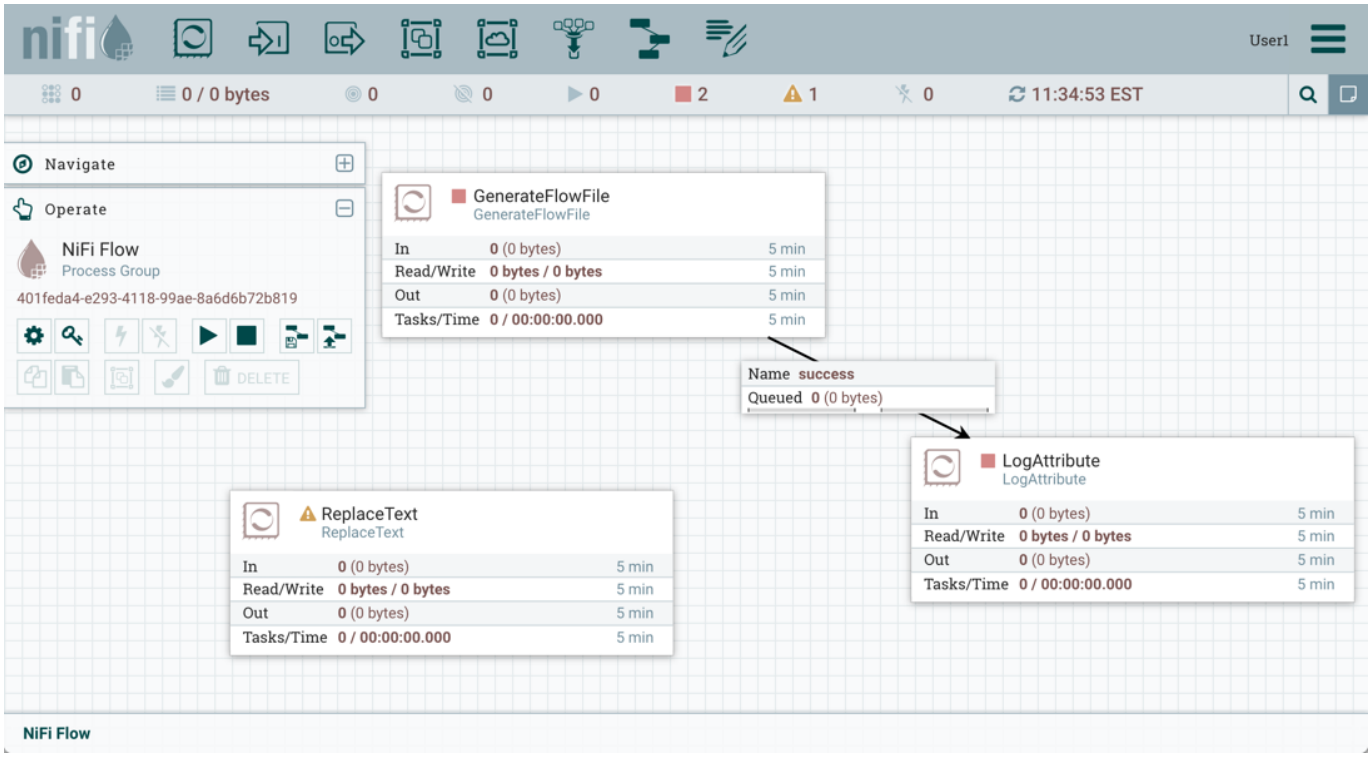

Рис.5.21.: Добавление процессора ReplaceText

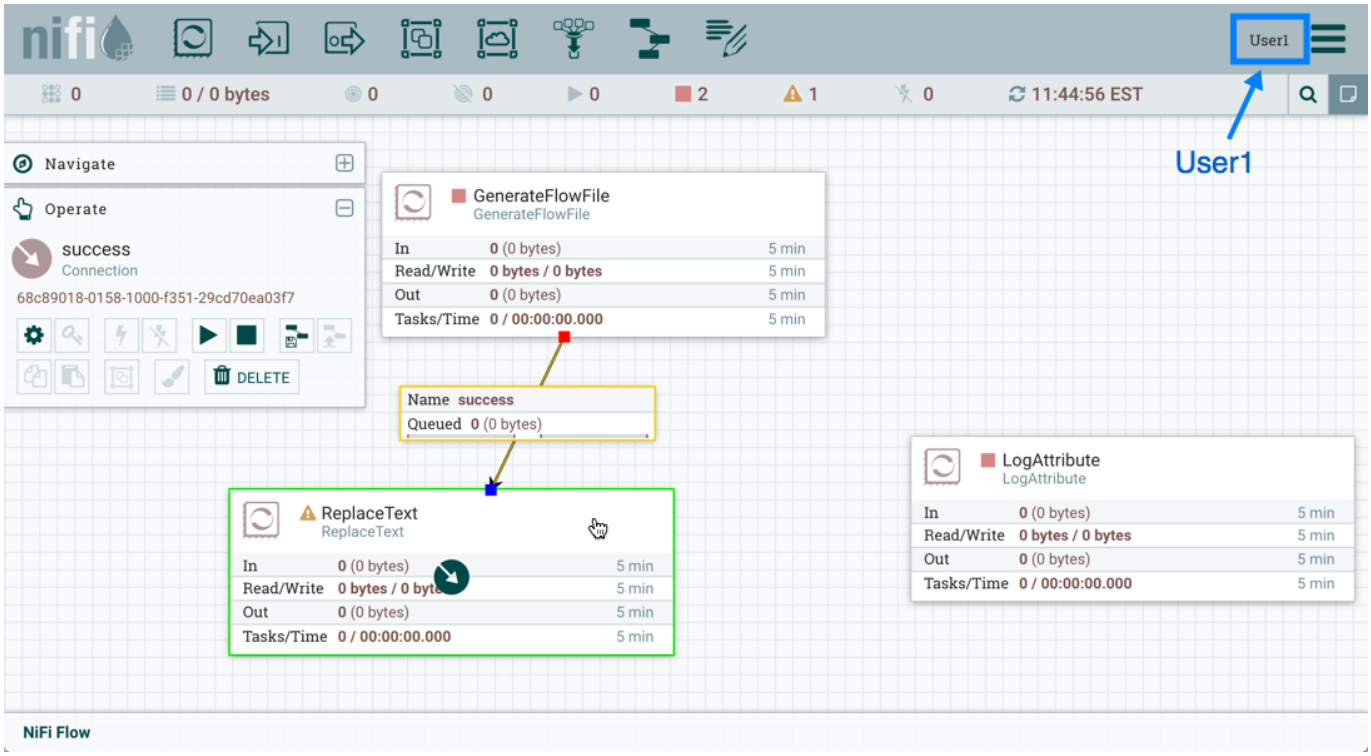

Рис.5.22.: User1 – изменение соединения

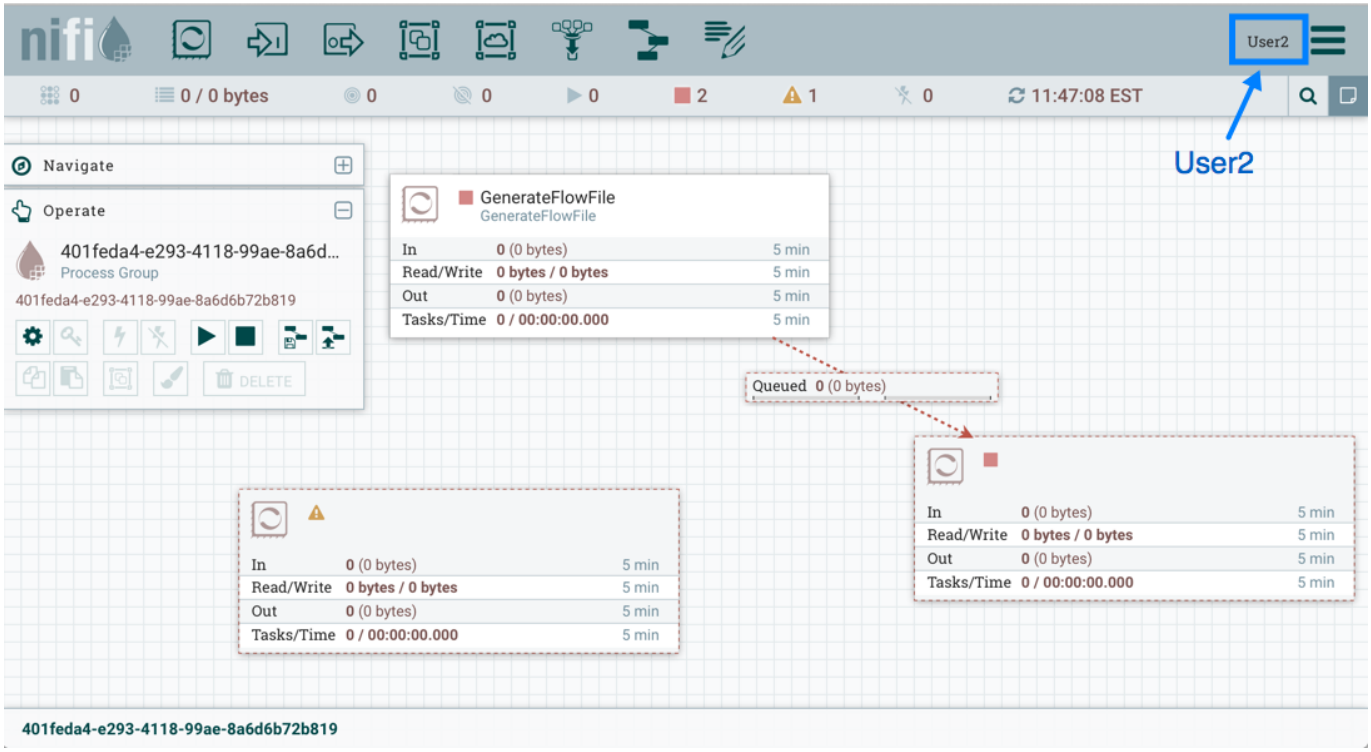

Рис.5.23.: User2 недоступно изменение соединения

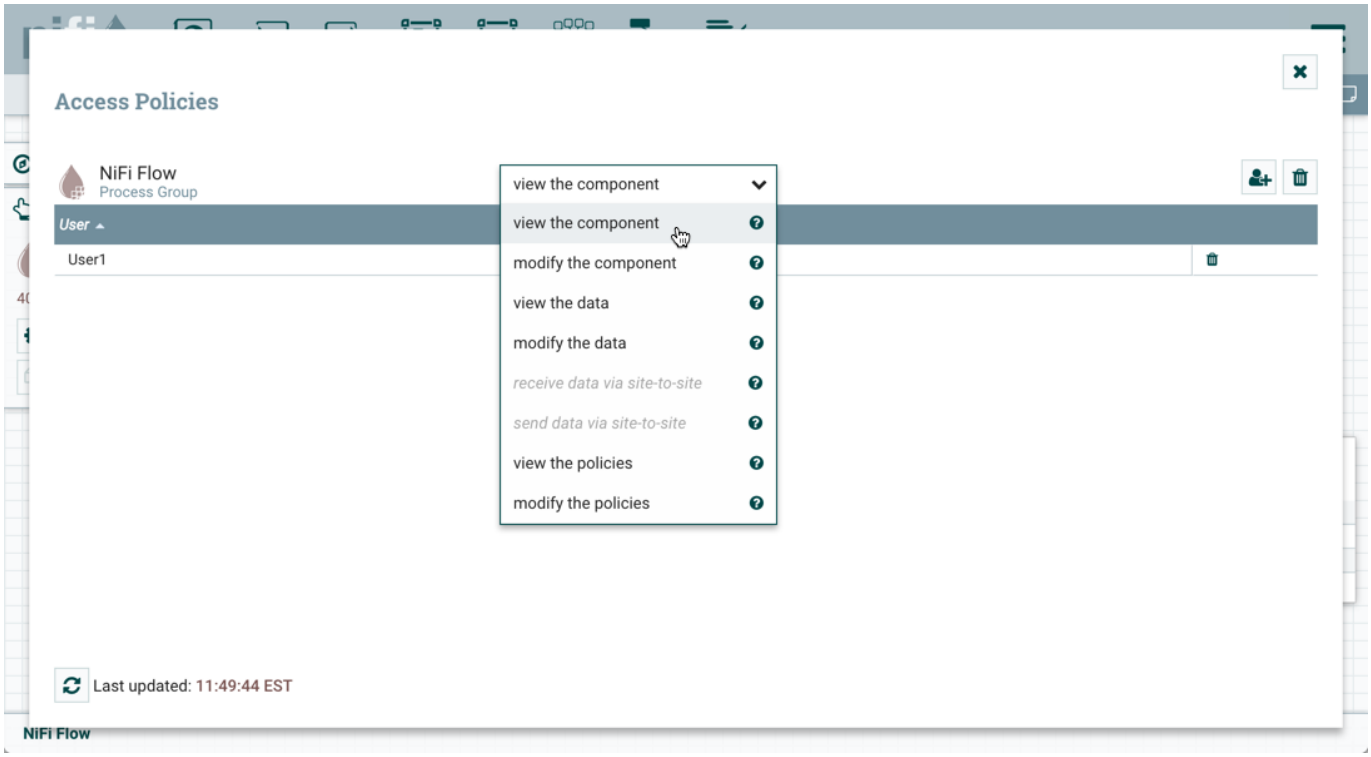

Рис.5.24.: View the component

4. Выбрать значок “Add User”. В поле “User Identity” ввести вручную или найти в списке *User2* и нажать “OK” (Рис.5.25.).

Будучи добавленным к политикам просмотра и изменения для группы процессов *User2* теперь может подключать процессор GenerateFlowFile к процессору ReplaceText (Рис.5.26.).

5.5 Kerberos Service

NiFi может быть настроен для использования Kerberos SPNEGO (или “Kerberos Service”) для аутентификации. В таком случае пользователи попадают в конечную точку REST */access/kerberos*, и сервер отвечает кодом состояния *401* с заголовком задачи *WWW-Authenticate: Negotiate*. Далее сервер связывается с браузером для использования GSS-API и загрузки тикета пользователя Kerberos с указанием его в качестве заголовка *Base64* в последующем запросе. Он принимает форму *Authorization: Negotiate YII...*, и **NiFi** пробует подтвердить этот билет с помощью KDC. В случае успеха принципал пользователя возвращается как подлинный, и поток следует аутентификации login/credential, в результате которой в ответ выдается JWT для предотвращения ненужных издержек аутентификации Kerberos при каждом последующем запросе. В случае если билет пользователя не подтверждается, то он возвращается с соответствующим кодом ошибки. После чего пользователь может предоставить свои учетные данные для формы регистрации Kerberos при условии настроенного *KerberosLoginIdentityProvider*. Дополнительную информация приведена в главе Аутентификация пользователя – Kerberos.

NiFi отвечает на запросы Kerberos SPNEGO только по соединению HTTPS, поскольку незащищенные запросы никогда не проходят проверку подлинности.

Для включения аутентификации службы Kerberos в *nifi.properties* должны быть настроены следующие свойства:

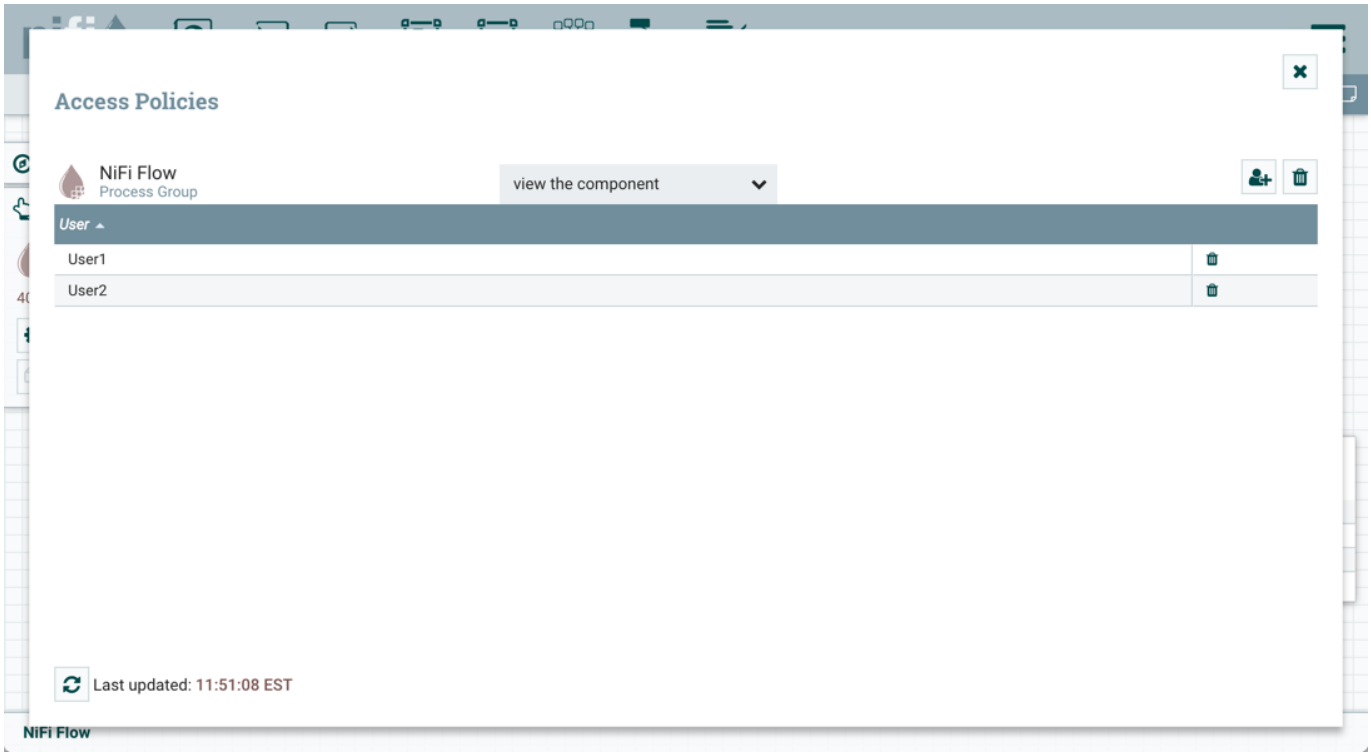

Рис.5.25.: Добавление User2 в политику группы

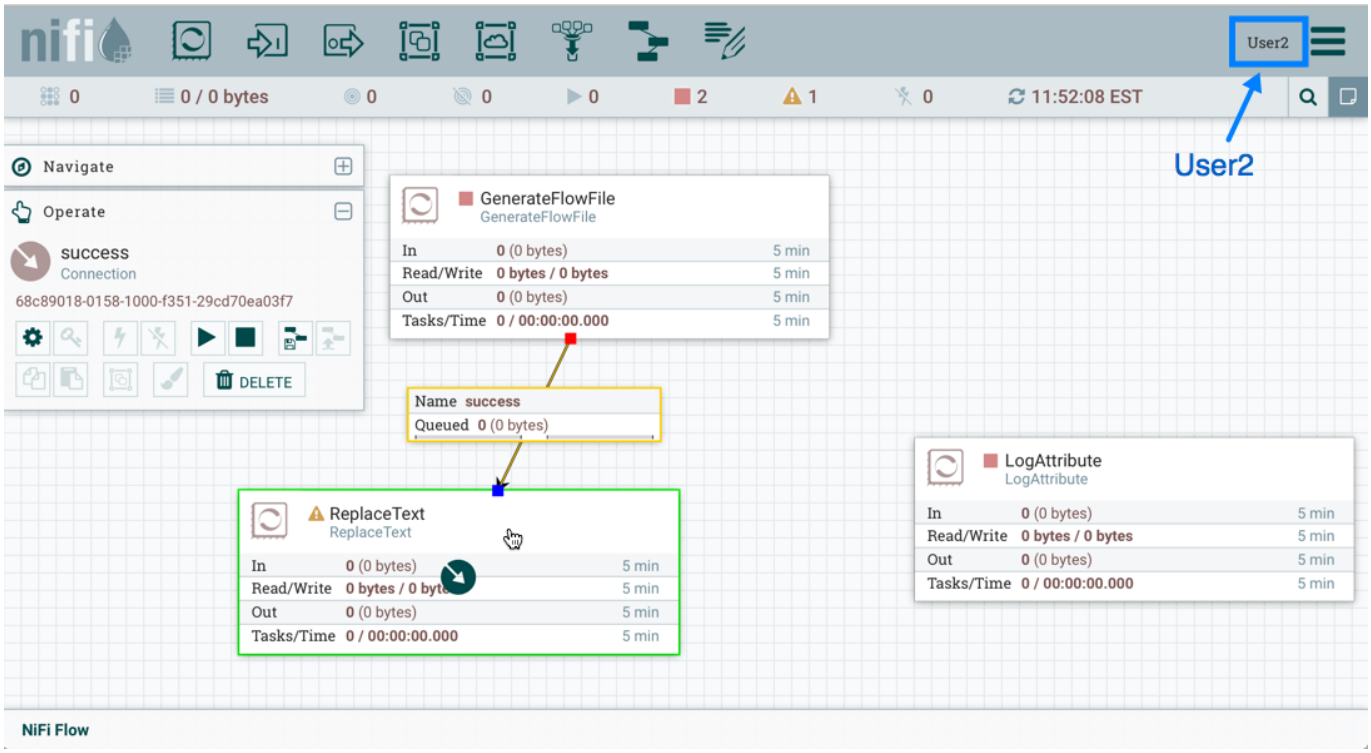

Рис.5.26.: User2 – изменение соединения

Таблица5.8.: Свойства для включения аутентификации Kerberos Service

Свойство	Значаение	Описание
Service Principal	true	Принципал сервиса, используемый NiFi для связи с KDC
Keytab Location	true	Путь к файлу keytab, содержащему принципал сервиса

Примечания:

- Kerberos чувствителен к регистру во многих местах, и сообщения об ошибках (или их отсутствие) могут быть недостаточно понятны. Рекомендуется проверить у службы чувствительность к регистру в конфигурационных файлах. Конвенция – *HTTP/fully.qualified.domain@REALM*;
- Браузеры имеют разные уровни ограничений при работе с SPNEGO. Некоторые из них предоставляют локальный тикет Kerberos в любой запрашиваемый домен, в то время как другие выдают пустой список доверенных доменов. Справочная информация для общих браузеров приведена по ссылке;
- Некоторые браузеры (например, устаревший IE) не поддерживают последние алгоритмы шифрования, такие как AES, и ограничены устаревшими алгоритмами (например, DES). Это следует учитывать при создании keytabs;
- Должен быть настроен KDC, определен принципал сервиса для NiFi и экспортирован keytab. Подробные инструкции по настройке и администрированию Kerberos Service выходят за рамки данного документа (MIT Kerberos Admin Guide), но далее приведен пример.

Пример добавления принципала для сервера на *nifi.nifi.apache.org* и экспорта ключа из KDC:

```
root@kdc:/etc/krb5kdc# kadmin.local
Authenticating as principal admin/admin@NIFI.APACHE.ORG with password.
kadmin.local:  listprincs
K/M@NIFI.APACHE.ORG
admin/admin@NIFI.APACHE.ORG
...
kadmin.local:  addprinc -randkey HTTP/nifi.nifi.apache.org
WARNING: no policy specified for HTTP/nifi.nifi.apache.org@NIFI.APACHE.ORG; defaulting to no␣
↪policy
Principal "HTTP/nifi.nifi.apache.org@NIFI.APACHE.ORG" created.
Principal "HTTP/nifi.nifi.apache.org@NIFI.APACHE.ORG" created.
kadmin.local:  ktadd -k /http-nifi.keytab HTTP/nifi.nifi.apache.org
Entry for principal HTTP/nifi.nifi.apache.org with kvno 2, encryption type des3-cbc-sha1 added to␣
↪keytab WRFILE:/http-nifi.keytab.
Entry for principal HTTP/nifi.nifi.apache.org with kvno 2, encryption type des-cbc-crc added to␣
↪keytab WRFILE:/http-nifi.keytab.
kadmin.local:  listprincs
HTTP/nifi.nifi.apache.org@NIFI.APACHE.ORG
K/M@NIFI.APACHE.ORG
admin/admin@NIFI.APACHE.ORG
...
kadmin.local: q
root@kdc:~# ll /http*
-rw------- 1 root root 162 Mar 14 21:43 /http-nifi.keytab
root@kdc:~#
```

5.6 Удаление/Добавление компонентов сервиса Nifi

Доступно с версии 1.4.11

Если кластер **ADS** разворачивается с помощью **ADCM**, то операции по добавлению/удалению хоста в сервис *NiFi* могут быть выполнены автоматически. После выполнения планирования нового аппаратного обеспечения необходимо добавить новые хосты в выбранный кластер в интерфейсе **ADCM**, используя кнопку “Add hosts” на вкладке “Hosts”. Кроме того, необходимо выполнить инициализацию каждого хоста, если того требует провайдер хостов.

Important: Описанные ниже операции не удаляют/добавляют хост из кластера – они лишь управляют компонентом *NiFi Server* и *NiFi-Registry* на хостах. Удаление хоста из кластера возможно в разделе “Hosts” кластера при условии, что к хосту не привязан ни один компонент

Для добавления или удаления *Nifi* с хостов необходимо воспользоваться соответствующими кнопками выпадающего меню, доступного по нажатию на иконку в поле “Actions” сервиса *Nifi* (Рис.5.27).

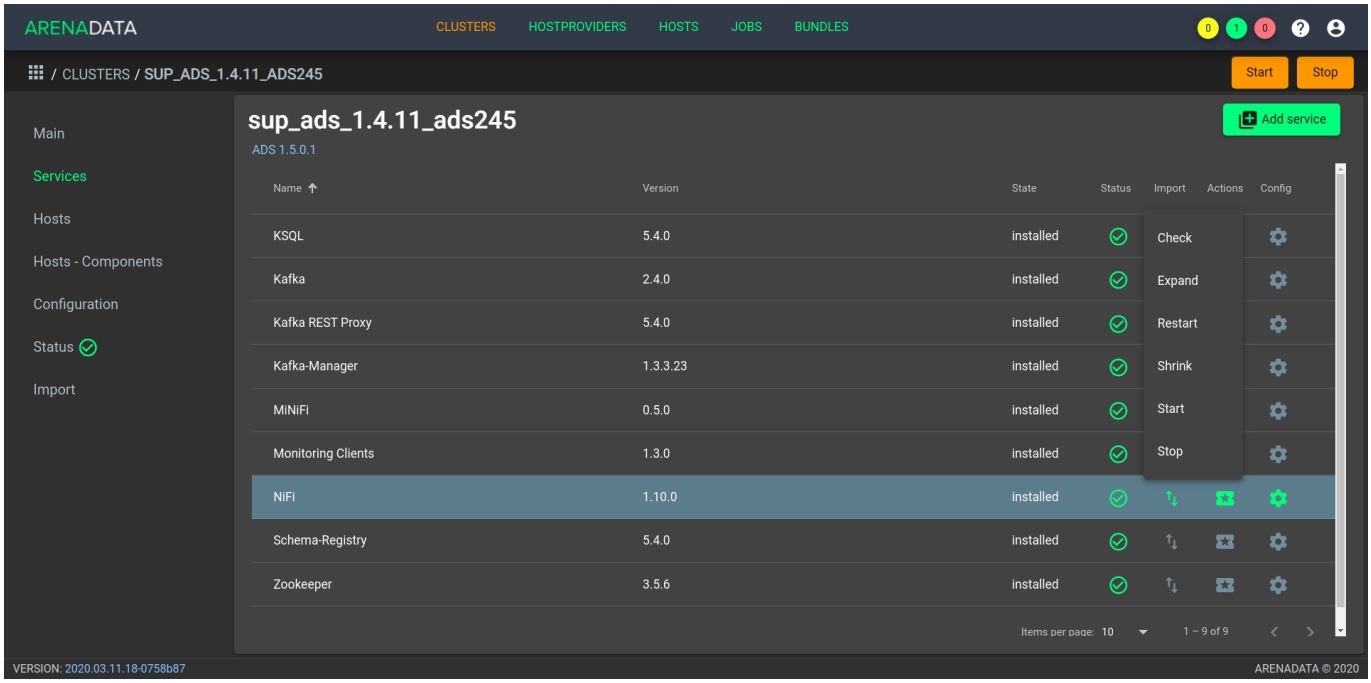

Рис.5.27.: Список допустимых операций над сервисом Nifi

5.6.1 Добавление компонентов Nifi Server и Niifi-Registry

Когда хосты становятся доступными для подключения по ssh для менеджера кластеров, необходимо выбрать действие *Expand* сервиса *Nifi* из списка возможных операций. В появившемся диалоговом окне предоставляется выбор опций (Рис.5.28):

- *Disable SELinux before cluster installation* – отключение SELinux на добавляемых хостах. Для того, чтобы данная настройка применилась, после завершения операции *Expand* необходимо перезагрузить хосты вручную;
- *Disable Firewalld before cluster installation* – выключение firewalld на добавляемых хостах;
- *Install OpenJDK before cluster installation* – установка пакета *java-1.8.0-openjdk* на добавляемых хостах;

- *Set vm.swappiness to 0 for all hosts* – отключение *swapping* на добавлемых хостах;
- *Append hosts into /etc/hosts file before cluster installation* – запись добавляемых нод в */etc/hosts* на всех хостах кластера. Данную опцию рекомендуется отключить, если настроен DNS.

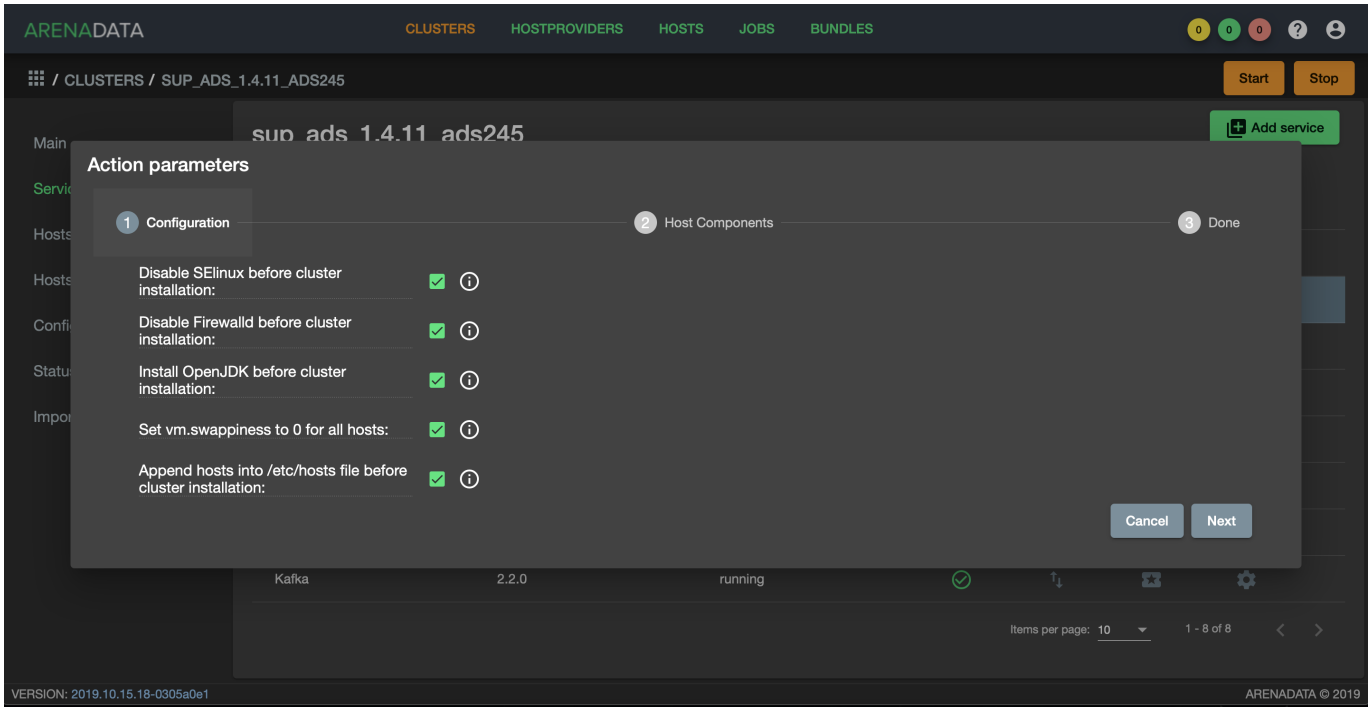

Рис.5.28.: Доступные при расширении настройки

После выбора опций для перехода к следующей странице конфигурации следует нажать кнопку “Next”, и в открывшейся форме необходимо распределить компонент *Nifi Server* по добавляемым хостам (Рис.5.29). Также есть возможность установить *Nifi-Registry*, если ранее он не был установлен. В случае если используется сервис *Monitoring Clinets*, его компоненты также необходимо разместить на добавляемых хостах.

Расширение сервиса запускается кнопкой “Run”. На добавленные хосты устанавливаются необходимые пакеты и производится их настройка. Текущая конфигурация *Flow*, представленная в *flow.xml.gz*, копируется на новый хост.

5.6.2 Удаление Nifi Server

Для удаления одного или нескольких Nifi Server с хостов кластера необходимо:

1. Выбрать действие *Shrink* сервиса *Nifi* из списка возможных операций (см. Рис.5.27), что приводит к появлению окна распределения компонента по хостам (см. Рис.5.29);
2. Любым из двух способов удалить привязку компонента к хосту (компонент *Nifi Server* выделяется белым цветом, как возможный к удалению с хостов):

- Выбрать компонент в колонке “Components” и убрать выделение с хостов в колонке “Hosts”, рамки которых выделены зеленым цветом;
- Выбрать хост в колонке “Hosts” и убрать выделение с компонента *Nifi Server* в колонке “Components”, если рамка компонета *Nifi Server* выделяется зеленым цветом.

3. Нажать кнопку “Run” в нижней части окна.

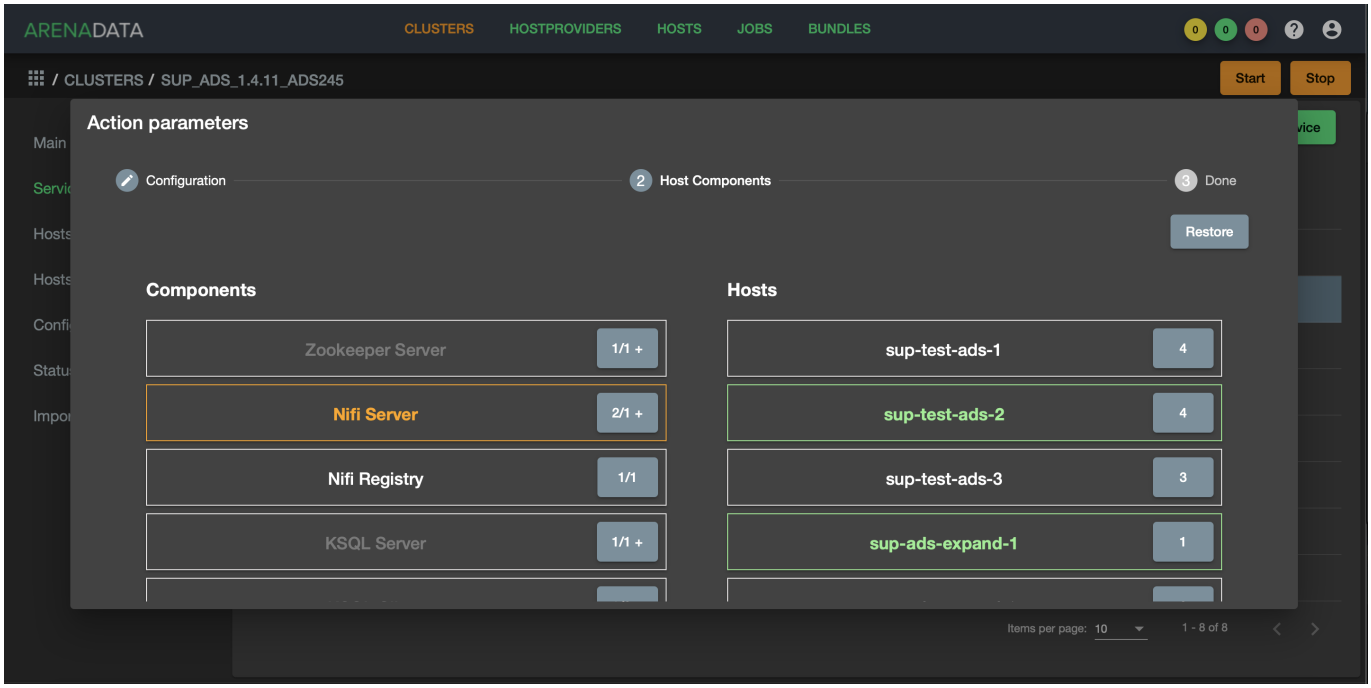

Рис.5.29.: Распределение компонента по хостам

Important: Описанная процедура не удаляет данные и пакет *NiFi* с хоста – она лишь выводит ноду из кластера *NiFi*

Глава 6

Руководство администратора по работе с Kafka

В документации приведены настройки платформы Arenadata Streaming на уровне сервиса Kafka, а именно на уровне брокера и топика, а также конфигурирование Producer, Consumer, Connect и Streams.

Инструкция может быть полезна администраторам, программистам, разработчикам и сотрудникам подразделений информационных технологий, осуществляющих внедрение и сопровождение системы.

Important: Контактная информация службы поддержки – e-mail: info@arenadata.io

Платформа ADS использует пары ключ-значение в формате файла свойств для конфигурации. Данные значения могут быть поставлены либо из файла, либо программно.

6.1 Настройка брокера

Все настройки брокера хранятся в конфигурационном файле */etc/kafka/conf/server.properties*.

Основные конфигурации брокера:

- *broker.id*
- *log.dirs*
- *zookeeper.connect*

Далее приведен список настроек с описанием и указанием их типа, значений по умолчанию и действительных, их важностью и режимом обновления.

zookeeper.connect – Строка хоста Zookeeper

- TYPE – string
- DEFAULT – high
- DYNAMIC UPDATE MODE – read-only

advertised.host.name – Применяется при неустановленном параметре *advertised.listeners* или *listeners*. Рекомендуется использовать *advertized.listeners*. Обозначает имя хоста для публикации в ZooKeeper для использования клиентами. В средах IaaS может отличаться от интерфейса, к которому привязывается брокер. Если параметр не задан, используется настроенное значение для *host.name*. В противном случае – значение из *java.net.InetAddress.getCanonicalHostName()*

- TYPE – string
- DEFAULT – null
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

advertised.listeners – Слушатели для публикации в ZooKeeper для использования клиентами (если есть отличие от свойства *listeners*). В средах IaaS может отличаться от интерфейса, к которому привязывается брокер. Если параметр не задан, используется значение для *listeners*. В отличие от *listeners* значение "0.0.0.0" недопустимо

- TYPE – string
- DEFAULT – null
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – per-broker

advertised.port – Применяется при неустановленном параметре *advertised.listeners* или *listeners*. Рекомендуется использовать *advertized.listeners*. Обозначает имя хоста для публикации в ZooKeeper для использования клиентами. В средах IaaS может отличаться от интерфейса, к которому привязывается брокер. Если параметр не задан, публикуется тот же порт, к которому привязан брокер

- TYPE – int
- DEFAULT – null
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

auto.create.topics.enable – Включение автоматического создания топика на сервере

- TYPE – boolean
- DEFAULT – true
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

auto.leader.rebalance.enable – Включение автоматической балансировки лидера. Балансировка лидера в фоновом режиме через регулярные промежутки времени

- TYPE – boolean
- DEFAULT – true
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

background.threads – Количество потоков для различных задач фоновой обработки

- TYPE – int
- DEFAULT – 10
- VALID VALUES – [1,...]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – cluster-wide

broker.id – Идентификатор брокера для сервера. Если значение не установлено, создается уникальный идентификатор брокера. Чтобы избежать конфликтов между id брокера, созданными с помощью zookeeper, и id брокера, настроенными пользователем, генерация идентификаторов брокера начинается с *reserved.broker.max.id + 1*

- TYPE – int
- DEFAULT – - 1
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

compression.type – Конечный тип сжатия для топика. Конфигурация принимает стандартные кодеки сжатия (“gzip”, “snappy”, “lz4”). Так же возможно “uncompressed”, что эквивалентно отсутствию сжатия; и “producer”, что означает сохранение исходного кодека сжатия, установленного поставщиком

- TYPE – string
- DEFAULT – producer
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – cluster-wide

delete.topic.enable – Включение возможности удаления топика. При отключенном параметре удаление топика через администратора не имеет результата

- TYPE – boolean
- DEFAULT – true
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

host.name – Применяется только когда параметр *listeners* не установлен. Рекомендуется использовать *listeners*. Обозначает имя хоста брокера. Если параметр задан, то привязка выполняется только к данному адресу. Если параметр не задан, привязка выполняется ко всем интерфейсам

- TYPE – string
- DEFAULT – “”
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

leader.imbalance.check.interval.seconds – Частота, с которой контроллер запускает проверку балансировки партиции

- TYPE – long
- DEFAULT – 300
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

leader.imbalance.per.broker.percentage – Коэффициент дисбаланса лидера, допустимый для каждого брокера. Контроллер запускает балансировку лидера, если он превышает данное значение для брокера. Указывается в процентах

- TYPE – int
- DEFAULT – 10
- IMPORTANCE – high

- DYNAMIC UPDATE MODE – read-only

listeners – Listener List – Разделенный запятыми список URI, которые прослушиваются, и имена слушателей сети. Если имя слушателя не является протоколом безопасности, необходимо установить *listener.security.protocol.map*. Для привязки ко всем интерфейсам указать имя хоста “0.0.0.0”. Если имя хоста не указано, привязка осуществляется к интерфейсу по умолчанию. Примеры списков слушателей сети: PLAINTEXT://myhost:9092,SSL://:9091,CLIENT://0.0.0.0:9092,REPLICATION://localhost:9093

- TYPE – string
- DEFAULT – null
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – per-broker

log.dir – Каталог хранения данных журнала (дополнительный для свойства log.dirs)

- TYPE – string
- DEFAULT – /tmp/kafka-logs
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

log.dirs – Каталоги хранения данных журнала. Если параметр не установлен, используется значение свойства *log.dir*

- TYPE – string
- DEFAULT – null
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – high

log.flush.interval.messages – Количество накопленных в партиции журнала данных перед их сбросом на диск

- TYPE – long
- DEFAULT – 9223372036854775807
- VALID VALUES – [1,. . .]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – cluster-wide

log.flush.interval.ms – Максимальное время хранения данных в любом топике в памяти до их сброса на диск. Указывается в миллисекундах. Если параметр не установлен, используется значение *log.flush.scheduler.interval.ms*

- TYPE – long
- DEFAULT – null
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – cluster-wide

log.flush.offset.checkpoint.interval.ms – Частота обновления постоянной записи последнего сброса, который действует как точка восстановления журнала

- TYPE – int
- DEFAULT – 60000

- VALID VALUES – [0,...]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

log.flush.scheduler.interval.ms – Частота log flusher проверки на необходимость сброса какого-либо журнала на диск. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 9223372036854775807
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

log.flush.start.offset.checkpoint.interval.ms – Частота обновления постоянной записи смещения начала журнала

- TYPE – int
- DEFAULT – 60000
- VALID VALUES – [0,...]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

log.retention.bytes – Максимальный размер журнала перед его удалением

- TYPE – long
- DEFAULT – - 1
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – cluster-wide

log.retention.hours – Количество часов для хранения файла журнала перед его удалением, третично по отношению к свойству *log.retention.ms*. Указывается в часах

- TYPE – int
- DEFAULT – 168
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

log.retention.minutes – Количество минут для хранения файла журнала перед его удалением, вторично по отношению к свойству *log.retention.hours*. Указывается в минутах

- TYPE – int
- DEFAULT – null
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

log.retention.ms – Количество миллисекунд для хранения файла журнала перед его удалением. Указывается в миллисекундах. Если параметр не установлен, используется значение *log.retention.minutes*

- TYPE – long
- DEFAULT – null
- IMPORTANCE – high

- DYNAMIC UPDATE MODE – cluster-wide

log.roll.hours – Максимальное время до развертывания нового сегмента журнала, вторично по отношению к свойству *log.roll.ms*. Указывается в часах

- TYPE – int
- DEFAULT – 168
- VALID VALUES – [1,...]
- IMPORTANCE – [1,...]
- DYNAMIC UPDATE MODE – read-only

log.roll.jitter.hours – Максимально допустимое значение джиттера для вычитания из *logRollTimeMillis*, вторично по отношению к свойству *log.roll.jitter.ms*. Указывается в часах

- TYPE – int
- DEFAULT – int
- VALID VALUES – [0,...]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

log.roll.jitter.ms – Максимально допустимое значение джиттера для вычитания из *logRollTimeMillis*. Указывается в миллисекундах. Если параметр не установлен, используется значение *log.roll.jitter.hours*

- TYPE – long
- DEFAULT – long
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – cluster-wide

log.roll.ms – Максимальное время до развертывания нового сегмента журнала. Указывается в миллисекундах. Если параметр не установлен, используется значение *log.roll.hours*

- TYPE – long
- DEFAULT – null
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – cluster-wide

log.segment.bytes – Максимальный размер одного файла журнала

- TYPE – int
- DEFAULT – 1073741824
- VALID VALUES – [14,...]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – cluster-wide

log.segment.delete.delay.ms – Время ожидания перед удалением файла из файловой системы

- TYPE – long
- DEFAULT – 60000
- VALID VALUES – [0,...]

- IMPORTANCE – high
- DYNAMIC UPDATE MODE – cluster-wide

message.max.bytes – Наибольший размер пакета данных, разрешенный ADS. При увеличении параметра следует также увеличить размер выборки для потребителей с целью обеспечения возможности получения пакета данных установленного размера. Параметр можно настроить для каждого топика с помощью поуровневой конфирурации топика *max.message.bytes*

- TYPE – int
- DEFAULT – 1000012
- VALID VALUES – [0,...]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – cluster-wide

min.insync.replicas – При установленном поставщиком подтверждении acks на “all” или “-1”, *min.insync.replicas* задается на минимальное количество реплик для подтверждения записи. Если этот минимум не может быть удовлетворен, то поставщик задает исключение (либо *NotEnoughReplicas*, либо *NotEnoughReplicasAfterAppend*). Совместное использование *min.insync.replicas* и acks обеспечивает более высокую гарантию к устойчивости. Типичным сценарием является создание топика с коэффициентом репликации *3*, параметром *min.insync.replicas* равным *2* и acks установленным на “all”. Это гарантирует, что поставщик задает исключение, если большинство реплик не принимает запись

- TYPE – int
- DEFAULT – 1
- VALID VALUES – [1,...]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – cluster-wide

num.io.threads – Число потоков, используемых сервером для обработки запросов, которые могут включать дисковые операции ввода-вывода

- TYPE – int
- DEFAULT – 8
- VALID VALUES – [1,...]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – cluster-wide

num.network.threads – Количество потоков, используемых сервером для получения запросов от сети и отправки ответов в сеть

- TYPE – int
- DEFAULT – 3
- VALID VALUES – [1,...]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – cluster-wide

num.recovery.threads.per.data.dir – Число потоков в каталоге данных, используемых для восстановления журнала при запуске или при сбросе по прекращению работы

- TYPE – int

- DEFAULT – 1
- VALID VALUES – [1,...]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – cluster-wide

num.replica.alter.log.dirs.threads – Число потоков, которые могут перемещать реплики между каталогами журналов, включая дисковые операции ввода-вывода

- TYPE – int
- DEFAULT – null
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

num.replica.fetchers – Количество потоков выборки, используемых для репликации данных от исходного брокера. Увеличение этого значения может увеличить степень параллелизма ввода-вывода в брокере-подписчике

- TYPE – int
- DEFAULT – 1
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – cluster-wide

offset.metadata.max.bytes – Максимальный размер для записи метаданных с учетом фиксации смещения

- TYPE – int
- DEFAULT – 4096
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

offsets.commit.required.acks – Принятие необходимых подтверждений acks перед фиксацией данных. Значение по умолчанию "-1" не следует переопределять

- TYPE – short
- DEFAULT – - 1
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

offsets.commit.timeout.ms – Фиксация смещения откладывается до тех пор, пока все реплики для топика смещения не получат коммит или данный установленный таймаут не будет достигнут. Аналогично времени ожидания запроса поставщика

- TYPE – int
- DEFAULT – 5000
- VALID VALUES – [1,...]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

offsets.load.buffer.size – Размер пакета для чтения из сегментов смещений при загрузке смещений в кэш

- TYPE – int

- DEFAULT – 5242880
- VALID VALUES – [1,...]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

offsets.retention.check.interval.ms – Частота проверки устаревших смещений

- TYPE – long
- DEFAULT – 600000
- VALID VALUES – [1,...]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

offsets.retention.minutes – Сброс смещений старше установленного срока хранения

- TYPE – int
- DEFAULT – 1440
- VALID VALUES – [1,...]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

offsets.topic.compression.codec – Кодек сжатия для топика смещения. Сжатие может использоваться для достижения “атомных” коммитов

- TYPE – int
- DEFAULT – 0
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

offsets.topic.num.partitions – Количество партиций для коммита топика смещения (не следует изменять после развертывания)

- TYPE – int
- DEFAULT – 50
- VALID VALUES – [1,...]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

offsets.topic.replication.factor – Коэффициент репликации для топика смещения (устанавливается выше с целью обеспечения доступности). Создание внутреннего топика невозможно, пока размер кластера не соответствует данному требованию коэффициента репликации

- TYPE – short
- DEFAULT – 3
- VALID VALUES – [1,...]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

offsets.topic.segment.bytes – Размер сегмента топика смещений в байтах. Значение должно быть относительно небольшим с целью ускорения сжатия журнала и загрузку кэша

- TYPE – int
- DEFAULT – 104857600
- VALID VALUES – [1,. . .]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

port – Применяется при неустановленном параметре *listeners*. Рекомендуется использовать *listeners*. Обозначает порт для прослушивания и приема подключений

- TYPE – int
- DEFAULT – 9092
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

queued.max.requests – Количество запросов в очереди до блокировки сетевых потоков

- TYPE – int
- DEFAULT – 500
- VALID VALUES – [1,. . .]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

quota.consumer.default – Применяется при неустановленном параметре динамических квот по умолчанию в Zookeeper. Любой потребитель группы customerId/consumer дросселируется при получении большего количества байтов, чем данное установленное значение в секунду

- TYPE – long
- DEFAULT – 9223372036854775807
- VALID VALUES – [1,. . .]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

quota.producer.default – Применяется при неустановленном параметре динамических квот по умолчанию в Zookeeper. Любой поставщик с известным clientId дросселируется при получении большего количества байтов, чем данное установленное значение в секунду

- TYPE – long
- DEFAULT – 9223372036854775807
- VALID VALUES – [1,. . .]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

replica.fetch.min.bytes – Минимальное количество байт, ожидаемое для каждого ответа на выборку. При недостаточном объеме срабатывает параметр *replicaMaxWaitTimeMs*

- TYPE – int

- DEFAULT – 1
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

replica.fetch.wait.max.ms – Максимальное время ожидания для каждого запроса на выборку с последующей публикацией реплик. Значение всегда должно быть меньше параметра *replica.lag.time.max.ms* для предотвращения частого сжатия ISR низкопроизводительных топиков

- TYPE – int
- DEFAULT – 500
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

replica.high.watermark.checkpoint.interval.ms – Верхний предел частоты сохранения на диск (Частота сохранения высокого водяного знака на диск)

- TYPE – long
- DEFAULT – 5000
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

replica.lag.time.max.ms – Удаление подписчика лидером из isr в случае, если подписчик не отправил ни одного запроса на выборку или не считал конечное смещение журнала лидеров

- TYPE – long
- DEFAULT – 10000
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

replica.socket.receive.buffer.bytes – Буфер приема сокетов для сетевых запросов

- TYPE – int
- DEFAULT – 65536
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

replica.socket.timeout.ms – Время ожидания сокета для сетевых запросов. Значение должно быть не менее установленного параметра *replica.fetch.wait.max.ms*

- TYPE – int
- DEFAULT – 30000
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

request.timeout.ms – Максимальное время ожидания клиентом ответа на запрос. Если ответ не получен до истечения установленного значения, клиент повторно отправляет запрос при необходимости

- TYPE – int
- DEFAULT – 30000
- IMPORTANCE – high

- DYNAMIC UPDATE MODE – read-only

socket.receive.buffer.bytes – Буфер SO_RCVBUF сокета сервера сокетов. При значении параметра "-1" используется ОС по умолчанию

- TYPE – int
- DEFAULT – 102400
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

socket.request.max.bytes – Максимальное количество байт в запросе сокета

- TYPE – int
- DEFAULT – 104857600
- VALID VALUES – [1,...]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

socket.send.buffer.bytes – Буфер SO_SNDBUF сокета сервера сокетов. При значении параметра "-1" используется ОС по умолчанию

- TYPE – int
- DEFAULT – 102400
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

transaction.max.timeout.ms – Максимально допустимое время ожидания для транзакций. Если запрошенное клиентом время транзакции превышает установленное значение, тогда брокер выдает ошибку в *InitProducerIdRequest*. Это предотвращает чрезмерное превышение времени ожидания для клиента, которое может тормозить чтение данных потребителями из топиков, включенных в транзакцию

- TYPE – int
- DEFAULT – 900000
- VALID VALUES – [1,...]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

transaction.state.log.load.buffer.size – Размер пакета для чтения из сегментов журнала транзакций при загрузке в кэш идентификаторов поставщиков и транзакций

- TYPE – int
- DEFAULT – 5242880
- VALID VALUES – [1,...]
- IMPORTANCE – [1,...]
- DYNAMIC UPDATE MODE – read-only

transaction.state.log.min.isr – Переопределение конфигурации *min.insync.replicas* для топика транзакции

- TYPE – int

- DEFAULT – 2
- VALID VALUES – [1,...]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

transaction.state.log.num.partitions – Количество партиций для топика транзакции (после развертывания параметр должен остаться неизменным)

- TYPE – int
- DEFAULT – 50
- VALID VALUES – [1,...]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

transaction.state.log.replication.factor – Коэффициент репликации для топика транзакции (задается выше для обеспечения доступности). Создание внутреннего топика завершается ошибкой, пока размер кластера не соответствует данному требованию к фактору репликации

- TYPE – short
- DEFAULT – 3
- VALID VALUES – [1,...]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

transaction.state.log.segment.bytes – Байты сегмента топика транзакции должны быть относительно небольшими для ускорения сжатия журнала и загрузки кэша

- TYPE – int
- DEFAULT – 104857600
- VALID VALUES – [1,...]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

transactional.id.expiration.ms – Максимальное время ожидания для координатора транзакций прежде, чем предварительно истечет срок действия идентификатора транзакции поставщика без получения обновлений состояния транзакции. Указывается в миллисекундах

- TYPE – int
- DEFAULT – 604800000
- VALID VALUES – [1,...]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

unclean.leader.election.enable – Указывает, следует ли включить не входящие в набор ISR реплики и установка последнего средства в качестве лидера, даже если это может привести к потере данных

- TYPE – boolean
- DEFAULT – false

- IMPORTANCE – high
- DYNAMIC UPDATE MODE – cluster-wide

zookeeper.connection.timeout.ms – Максимальное время ожидания клиентом установки соединения с Zookeeper. Если параметр не задан, используется значение для *zookeeper.session.timeout.ms*. Указывается в миллисекундах

- TYPE – int
- DEFAULT – null
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

zookeeper.max.in.flight.requests – Максимальное количество неподтвержденных запросов, отправленных клиентом в Zookeeper, перед блокировкой

- TYPE – int
- DEFAULT – 10
- VALID VALUES – [1,. . .]
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

zookeeper.session.timeout.ms – Тайм-аут сессии Zookeeper. Указывается в миллисекундах

- TYPE – int
- DEFAULT – int
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

zookeeper.set.acl – Настройка клиента для использования безопасных списков управления доступом ACL

- TYPE – boolean
- DEFAULT – boolean
- IMPORTANCE – high
- DYNAMIC UPDATE MODE – read-only

broker.id.generation.enable – Автоматическое создание идентификатора брокера на сервере. При включенном параметре значение, настроенное для *reserved.broker.max.id*, должно быть пересмотрено

- TYPE – boolean
- DEFAULT – true
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

broker.rack – Стойка брокера. Используется при назначении репликации в стойке для отказоустойчивости. Примеры: “RACK1”, “us-east-1d”

- TYPE – string
- DEFAULT – string
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

connections.max.idle.ms – Время ожидания бездействующих соединений: потоки процессора сокета сервера закрывают соединения, которые простаивают больше установленного значения. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 600000
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

controlled.shutdown.enable – Включение контролируемого завершения работы сервера

- TYPE – boolean
- DEFAULT – true
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

controlled.shutdown.max.retries – Контролируемое выключение может завершиться ошибкой по нескольким причинам: параметр определяет количество повторных попыток подключения при возникновении таких сбоев

- TYPE – int
- DEFAULT – 3
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

controlled.shutdown.retry.backoff.ms – Перед каждой повторной попыткой подключения системе требуется время для восстановления состояния, вызвавшего предыдущий сбой (сбой контроллера, задержка реплики и т.д.). Параметр определяет время ожидания перед повторной попыткой. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 5000
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

controller.socket.timeout.ms – Время ожидания сокета для каналов контроллер-брокер. Указывается в миллисекундах

- TYPE – int
- DEFAULT – 30000
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

default.replication.factor – Коэффициенты репликации по умолчанию для автоматически создаваемых топиков

- TYPE – int
- DEFAULT – 1
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

delegation.token.expiry.time.ms – Время действия токена перед его обновлением. Значение по умолчанию 1 день. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 86400000
- VALID VALUES – [1,. . .]
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

delegation.token.master.key – Мастер/секретный ключ для создания и проверки делегированных токенов. Один и тот же ключ должен быть настроен для всех брокеров. Если ключ не установлен или задана пустая строка, брокеры отключают поддержку делегированных токенов

- TYPE – password
- DEFAULT – null
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

delegation.token.max.lifetime.ms – Максимальный срок действия токена, по истечении которого он больше не может быть обновлен. Значение по умолчанию 7 дней. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 604800000
- VALID VALUES – [1,. . .]
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

delete.records.purgatory.purge.interval.requests – Интервал очистки записей на удаление. Значение указывается в количестве запросов

- TYPE – int
- DEFAULT – 1
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

fetch.purgatory.purge.interval.requests – Интервал очистки запросов выборки. Значение указывается в количестве запросов

- TYPE – int
- DEFAULT – 1000
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

group.initial.rebalance.delay.ms – Время, в течение которого координатор группы ожидает присоединения большего числа потребителей к новой группе перед выполнением первой перебалансировки. Более длительная задержка означает потенциально меньшее количество перебалансировок, но увеличивает время до начала обработки. Указывается в миллисекундах

- TYPE – int
- DEFAULT – 3000

- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

group.max.session.timeout.ms – Максимально допустимое время ожидания сессии для зарегистрированных потребителей. Более длительные тайм-ауты дают потребителям больше времени для обработки данных между heartbeat-сообщениями за счет большего времени для выявления сбоев. Указывается в миллисекундах

- TYPE – int
- DEFAULT – 300000
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

group.min.session.timeout.ms – Минимально допустимое время ожидания сессии для зарегистрированных потребителей. Более короткие тайм-ауты приводят к более быстрому обнаружению сбоев за счет более частых heartbeat-сообщений, которые могут перегружать ресурсы брокера. Указывается в миллисекундах

- TYPE – int
- DEFAULT – 6000
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

inter.broker.listener.name – Имя слушателя для связи между брокерами. Если параметр не задан, имя слушателя определяется свойством *security.inter.broker.protocol*. Одновременная установка параметров *inter.broker.listener.name* и *security.inter.broker.protocol* вызывает ошибку

- TYPE – string
- DEFAULT – null
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

inter.broker.protocol.version – Версия межброкерского протокола. Обычно параметр задается после обновления всех брокеров до новой версии. Пример некоторых допустимых значений: “0.8.0”, “0.8.1”, “0.8.1.1”, “0.8.2”, “0.8.2.0”, “0.8.2.1”, “0.9.0.0”, “0.9.0.1”. Необходимо проверить ApiVersion для полного списка

- TYPE – string
- DEFAULT – 1.1-IV0
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

log.cleaner.backoff.ms – Время спящего режима при отсутствии журналов для очистки. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 15000
- VALID VALUES – [0,...]
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – cluster-wide

log.cleaner.dedupe.buffer.size – Общая память, используемая для дедупликации журнала во всех чистых потоках

- TYPE – long
- DEFAULT – 134217728
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – cluster-wide

log.cleaner.delete.retention.ms – Длительность хранения удаленных записей. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 86400000
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – cluster-wide

log.cleaner.enable – Включение процесса очистки журналов для запуска на сервере. Параметр должен быть включен, если используются какие-либо топики с помощью *cleanup.policy=compact*, включая топик внутренних смещений. Если параметр отключен, данные топики не сжимаются и постоянно растут в объеме

- TYPE – boolean
- DEFAULT – true
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

log.cleaner.io.buffer.load.factor – Коэффициент загрузки буфера дедуплирования журнала очистки – процент заполнения буфера дедуплирования. Более высокое значение позволит очистить больше журнала, но приведет к большему количеству хэш-конфликтов

- TYPE – double
- DEFAULT – 0.9
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – cluster-wide

log.cleaner.io.buffer.size – Общая память, используемая для ввода-вывода буферов журнала очистки через все чистые потоки

- TYPE – int
- DEFAULT – 524288
- VALID VALUES – [0,...]
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – cluster-wide

log.cleaner.io.max.bytes.per.second – Очистка журнала дросселируется таким образом, чтобы сумма операций чтения и записи была меньше установленного значения

- TYPE – double
- DEFAULT – 1.7976931348623157E308
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – cluster-wide

log.cleaner.min.cleanable.ratio – Минимальное отношение грязного журнала к общему журналу для журнала, пригодного для очистки

- TYPE – double
- DEFAULT – 0.5
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – cluster-wide

log.cleaner.min.compaction.lag.ms – Минимальное время, в течение которого сообщение остается несжатым в журнале. Применяется только для журналов с функцией сжатия. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 0
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – cluster-wide

log.cleaner.threads – Количество фоновых потоков для очистки журнала

- TYPE – int
- DEFAULT – 1
- VALID VALUES – [0,...]
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – cluster-wide

log.cleanup.policy – Политика очистки по умолчанию для сегментов, превышающих период хранения. Допустимые политики: “delete” и “compact”

- TYPE – list
- DEFAULT – delete
- VALID VALUES – [compact, delete]
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – cluster-wide

log.index.interval.bytes – Интервал добавления записи в индекс смещения

- TYPE – int
- DEFAULT – 4096
- VALID VALUES – [0,...]
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – cluster-wide

log.index.size.max.bytes – Максимальный размер индекса смещения. Указывается в байтах

- TYPE – int
- DEFAULT – 10485760
- VALID VALUES – [4,...]
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – cluster-wide

log.message.format.version – Версия формата сообщений, которую брокер использует для добавления данных в журналы. Значение должно быть действительным ApiVersion. Некоторые примеры: “0.8.2”, “0.9.0.0”, “0.10.0”. Необходимо проверить ApiVersion для получения более подробной информации. Установив версию формата сообщений, пользователь подтверждает, что все существующие данные на диске меньше или равны указанной версии. Неправильное задание параметра приводит к тому, что потребители с более старыми версиями получают данные в нечитаемом формате

- TYPE – string
- DEFAULT – 1.1-IV0
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

log.message.timestamp.difference.max.ms – Максимальное допустимое различие между отметкой времени, когда брокер получает сообщение, и отметкой времени, указанной в сообщении. При *log.message.timestamp.type=CreateTime* сообщение отклоняется, если разница в отметке времени превышает указанный порог. Конфигурация игнорируется, если *log.message.timestamp.type=LogAppendTime.* Максимально допустимое различие временных отметок должно быть не больше, чем *log.retention.ms.* Указывается в миллисекундах

- TYPE – long
- DEFAULT – 9223372036854775807
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – cluster-wide

log.message.timestamp.type – Определить, является ли отметка времени в сообщении временем создания сообщения или временем добавления журнала. Параметр может принимать значение “CreateTime” либо “LogAppendTime”

- TYPE – string
- DEFAULT – CreateTime
- VALID VALUES – [CreateTime, LogAppendTime]
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – cluster-wide

log.preallocate – Предварительное выделение файла при создании нового сегмента. При испольовании платформы ADS в Windows рекомендуется установить значение “true”

- TYPE – boolean
- DEFAULT – false
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – cluster-wide

log.retention.check.interval.ms – Частота проверки журналом очистки на наличие какого-либо журнала на удаление. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 300000
- VALID VALUES – [1,. . .]
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

max.connections.per.ip – Максимальное количество подключений с каждого IP-адреса

- TYPE – int
- DEFAULT – 2147483647
- VALID VALUES – [1,...]
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

max.connections.per.ip.overrides – Ip или hostname переопределяет максимальное количество подключений по умолчанию

- TYPE – string
- DEFAULT – “”
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

max.incremental.fetch.session.cache.slots – Максимальное количество сессий инкрементной выборки

- TYPE – int
- DEFAULT – 1000
- VALID VALUES – [0,...]
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

num.partitions – Число партиций по умолчанию для каждого топика

- TYPE – int
- DEFAULT – 1
- VALID VALUES – [1,...]
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

password.encoder.old.secret – Старый секрет для кодирования динамически настроенных паролей. Установка параметра требуется только при обновлении секрета. Если параметр задан, все динамически закодированные пароли декодируются и перекодируются с помощью *password.encoder.secret* при запуске брокера

- TYPE – password
- DEFAULT – null
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

password.encoder.secret – Секрет для кодирования динамически настроенных паролей для данного брокера

- TYPE – password
- DEFAULT – null
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

principal.builder.class – Полное имя класса, реализующего интерфейс ADSPrincipalBuilder, который используется для создания объекта ADSPrincipal во время авторизации. Конфигурация также поддерживает устаревший интерфейс PrincipalBuilder, который ранее использовался для аутентификации клиентов по протоколу SSL. Если параметр не задан, действие по умолчанию зависит от используемого протокола безопасности. Для аутентификации SSL имя принципала отличается от имени из сертификата клиента, если он предоставлен; в противном случае, если аутентификация клиента не требуется, имя принципала задается "ANONYMOUS". Для аутентификации SASL принципал задается на основании правил, определенных в *sasl.kerberos.principal.to.local.rules* с использованием GSSAPI и идентификатора аутентификации SASL для других механизмов. Для PLAINTEXT имя принципала – "ANONYMOUS"

- TYPE – class
- DEFAULT – null
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – per-broker

producer.purgatory.purge.interval.requests – Интервал очистки запросов поставщика. Значение указывается в количестве запросов

- TYPE – int
- DEFAULT – 1000
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

queued.max.request.bytes – Разрешенное число байтов в очереди до того, как запросы не будут прочитаны

- TYPE – long
- DEFAULT – - 1
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

replica.fetch.backoff.ms – Длительность спящего режима при возникновении ошибки партиции. Указывается в миллисекундах

- TYPE – int
- DEFAULT – 1000
- VALID VALUES – [0,. . .]
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

replica.fetch.max.bytes – Количество байтов сообщений, получаемых каждой партицией. Параметр не является абсолютным максимумом. Если первый пакет записей в первой непустой партиции выборки больше установленного значения, пакет данных все равно будет возвращен для обеспечения гарантии возможности выполнения. Максимальный размер пакета записей, принятый брокером, определяется через *message.max.bytes* (конфигурация брокера) или *max.message.bytes* (конфигурация топика)

- TYPE – int
- DEFAULT – 1048576
- VALID VALUES – [0,. . .]
- IMPORTANCE – medium

- DYNAMIC UPDATE MODE – read-only

replica.fetch.response.max.bytes – Максимальное количество байтов, ожидаемое для полного ответа на выборку. Параметр не является абсолютным максимумом. Записи извлекаются пакетами, и если первый пакет записей в первой непустой партиции выборки больше установленного значения, пакет данных все равно будет возвращен для обеспечения гарантии возможности выполнения. Максимальный размер пакета записей, принятый брокером, определяется через *message.max.bytes* (конфигурация брокера) или *max.message.bytes* (конфигурация топика)

- TYPE – int
- DEFAULT – 10485760
- VALID VALUES – [0,...]
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

reserved.broker.max.id – Максимальное число, которое можно использовать для broker.id

- TYPE – int
- DEFAULT – 1000
- VALID VALUES – [0,...]
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

sasl.enabled.mechanisms – Список механизмов SASL, включенных на сервере ADS. Список может содержать любой механизм, для которого обеспечивается безопасность. По умолчанию включен только GSSAPI

- TYPE – list
- DEFAULT – GSSAPI
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – per-broker

sasl.jaas.config – Параметры контекста входа JAAS для соединений SSL в формате, используемом файлами конфигурации JAAS. Формат файла конфигурации JAAS описан по ссылке. Формат значения: "(=)*;"

- TYPE – password
- DEFAULT – null
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – per-broker

sasl.kerberos.kinit.cmd – Путь команд Kerberos kinit

- TYPE – string
- DEFAULT – /usr/bin/kinit
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – per-broker

sasl.kerberos.min.time.before.relogin – Время ожидания авторизации потока между попытками обновления

- TYPE – long
- DEFAULT – 60000

- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – per-broker

sasl.kerberos.principal.to.local.rules – Список правил для сопоставления имен принципалов с короткими именами (обычно с именами пользователей операционной системы). Правила оцениваются по порядку, и первое правило, совпадающее с именем принципала, используется для сопоставления его с коротким именем. Все последующие правила в списке игнорируются. По умолчанию имена принципалов формы {username}/{hostname}@{REALM} сопоставляются с именем {username}. Важно обратить внимание, что данная конфигурация игнорируется, если расширение ADSPrincipalBuilder обеспечивается настройкой *main.builder.class*

- TYPE – list
- DEFAULT – DEFAULT
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – per-broker

sasl.kerberos.service.name – Имя принципала Kerberos, которое запускает ADS. Значение можно определить в конфигурации ADS JAAS либо в конфигурации ADS

- TYPE – string
- DEFAULT – null
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – per-broker

sasl.kerberos.ticket.renew.jitter – Процент случайного джиттера по отношению к времени возобновления

- TYPE – double
- DEFAULT – 0.05
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – per-broker

sasl.kerberos.ticket.renew.window.factor – Время ожидания авторизации потока до тех пор, пока не будет достигнут указанный коэффициент времени от последнего обновления до истечения срока действия тикета, и попытка возобновления тикета за этот период времени

- TYPE – double
- DEFAULT – 0.8
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – per-broker

sasl.mechanism.inter.broker.protocol – Механизм SASL для взаимодействия между брокерами. По умолчанию используется GSSAPI

- TYPE – string
- DEFAULT – GSSAPI
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – per-broker

security.inter.broker.protocol – Протокол безопасности для связи между брокерами. Допустимые значения: “PLAINTEXT”, “SSL”, “SASL_PLAINTEXT”, “SASL_SSL”. Одновременная установка параметров *security.inter.broker.protocol* и *inter.broker.listener.name* вызывает ошибку

- TYPE – string
- DEFAULT – PLAINTEXT
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – read-only

ssl.cipher.suites – Список наборов шифров. Именованная комбинация аутентификации, шифрования, MAC и ключей обмена алгоритма для согласования параметров безопасности для сетевого подключения с использованием протокола TLS или SSL. По умолчанию поддерживаются все доступные варианты шифрования

- TYPE – list
- DEFAULT – “”
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – per-broker

ssl.client.auth – Конфигурация брокера ADS для запроса аутентификации клиента. Следующие настройки являются общими:

- *ssl.client.auth=required* – требование проверки подлинности клиента;
- *ssl.client.auth=request* – аутентификация клиента является необязательной;
- *ssl.client.auth=none* – аутентификация клиента не требуется
- TYPE – string
- DEFAULT – none
- VALID VALUES – [required, requested, none]
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – per-broker

ssl.enabled.protocols – Список протоколов, включенных для соединений SSL

- TYPE – list
- DEFAULT – TLSv1.2,TLSv1.1,TLSv1
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – per-broker

ssl.key.password – Пароль закрытого ключа в файле хранилища ключей. Необязательный параметр для клиента

- TYPE – password
- DEFAULT – null
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – per-broker

ssl.keymanager.algorithm – Алгоритм службы управления ключами для SSL-соединений. Значением по умолчанию является алгоритм, настроенный для Java Virtual Machine

- TYPE – string
- DEFAULT – SunX509
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – per-broker

ssl.keystore.location – Расположение файла хранилища ключей. Необязательный параметр для клиента, может использоваться для двусторонней аутентификации клиента

- TYPE – string
- DEFAULT – null
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – per-broker

ssl.keystore.password – Пароль хранилища для файла хранения ключей. Необязательный параметр для клиента, требуется только при настройке *ssl.keystore.location*

- TYPE – password
- DEFAULT – null
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – per-broker

ssl.keystore.type – Формат файла хранилища ключей. Необязательный параметр для клиента

- TYPE – string
- DEFAULT – JKS
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – per-broker

ssl.protocol – Протокол SSL для генерации SSLContext. Значение по умолчанию – “TLS”, что подходит для большинства случаев. Допустимыми значениями в последних JVM являются “TLS”, “TLSv1.1” и “TLSv1.2”. Протоколы “SSL”, “SSLv2” и “SSLv3” могут поддерживаться в более старых JVM, но их использование не рекомендуется из-за известных уязвимостей безопасности

- TYPE – string
- DEFAULT – TLS
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – per-broker

ssl.provider – Имя поставщика безопасности для соединений SSL. Значением по умолчанию является поставщик безопасности по умолчанию для JVM

- TYPE – string
- DEFAULT – null
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – per-broker

ssl.trustmanager.algorithm – Алгоритм доверенной службы управления ключами для SSL-соединений. Значением по умолчанию является алгоритм, настроенный для Java Virtual Machine

- TYPE – string
- DEFAULT – PKIX
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – per-broker

ssl.truststore.location – Расположение файла хранилища trust store

- TYPE – string
- DEFAULT – null
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – per-broker

ssl.truststore.password – Пароль для файла хранилища trust store. При неустановленном пароле доступ к хранилищу есть, но осуществляется с отключенной проверкой надежности

- TYPE – password
- DEFAULT – null
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – per-broker

ssl.truststore.type – Формат файла хранилища trust store

- TYPE – string
- DEFAULT – JKS
- IMPORTANCE – medium
- DYNAMIC UPDATE MODE – per-broker

alter.config.policy.class.name – Класс политики изменяемых конфигураций для их валидации. Класс осуществляет интерфейс *org.apache.kafka.server.policy.AlterConfigPolicy*

- TYPE – class
- DEFAULT – null
- IMPORTANCE – low
- DYNAMIC UPDATE MODE – read-only

alter.log.dirs.replication.quota.window.num – Количество выборок для сохранения в памяти для квот репликации изменяемых журналов

- TYPE – int
- DEFAULT – 11
- VALID VALUES – [1,. . .]
- IMPORTANCE – low
- DYNAMIC UPDATE MODE – read-only

alter.log.dirs.replication.quota.window.size.seconds – Временной интервал каждой выборки для квот репликации изменяемых журналов. Указывается в секундах

- TYPE – int
- DEFAULT – 1
- VALID VALUES – [1,. . .]
- IMPORTANCE – low
- DYNAMIC UPDATE MODE – read-only

authorizer.class.name – Класс используемой авторизации

- TYPE – string

- DEFAULT – “”
- IMPORTANCE – low
- DYNAMIC UPDATE MODE – read-only

create.topic.policy.class.name – Создание класса политики топика для его валидации. Класс осуществляет интерфейс *org.apache.kafka.server.policy.CreateTopicPolicy*

- TYPE – class
- DEFAULT – null
- IMPORTANCE – low
- DYNAMIC UPDATE MODE – read-only

delegation.token.expiry.check.interval.ms – Интервал сканирования для удаления делегированных токенов с истекшим сроком действия. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 3600000
- VALID VALUES – [1,. . .]
- IMPORTANCE – low
- DYNAMIC UPDATE MODE – read-only

listener.security.protocol.map – Сопоставление имен слушателей и протоколов безопасности. Параметр должен быть определен для того, чтобы один и тот же протокол безопасности мог использоваться в нескольких портах или IP-адресах. Например, внутренний и внешний трафик могут быть разделены, даже если для обоих требуется SSL. То есть, пользователь может определить слушателей с именами “INTERNAL” и “EXTERNAL” свойством: “INTERNAL:SSL, EXTERNAL:SSL”, где ключ и значение разделяются двоеточием, а записи карты разделяются запятыми (без пробелов). Каждое имя слушателя должно отображаться на карте только один раз. Различные настройки безопасности (SSL и SASL) могут быть настроены для каждого слушателя путем добавления стандартизированного префикса (имя слушателя в нижнем регистре) к имени конфигурации. Например, чтобы установить другое хранилище ключей для внутреннего слушателя, будет установлена конфигурация с именем *listener.name.internal.ssl.keystore.location*. Если конфигурация для имени слушателя не задана, используется общая конфигурация (то есть *ssl.keystore.location*)

- TYPE – string
- DEFAULT – PLAINTEXT:PLAINTEXT,SSL:SSL,SASL_PLAINTEXT:SASL_PLAINTEXT,SASL_SSL:SASL_SSL
- IMPORTANCE – low
- DYNAMIC UPDATE MODE – per-broker

metric.reporters – Список классов для использования в качестве репортеров метрик. Реализация интерфейса *org.apache.kafka.common.metrics.MetricsReporter* позволяет подключать классы, которые будут уведомлены о создании новой метрики. JmxReporter всегда включен в реестр статистических данных JMX

- TYPE – list
- DEFAULT – “”
- IMPORTANCE – low
- DYNAMIC UPDATE MODE – cluster-wide

metrics.num.samples – Количество выборок, поддерживаемых для вычисления метрик

- TYPE – int
- DEFAULT – 2

- VALID VALUES – [1,...]
- IMPORTANCE – low
- DYNAMIC UPDATE MODE – read-only

metrics.recording.level – Самый высокий уровень записи для метрик

- TYPE – string
- DEFAULT – INFO
- IMPORTANCE – low
- DYNAMIC UPDATE MODE – read-only

metrics.sample.window.ms – Время ожидания вычисления метрик выборки. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 30000
- VALID VALUES – [1,...]
- IMPORTANCE – low
- DYNAMIC UPDATE MODE – read-only

password.encoder.cipher.algorithm – Алгоритм шифрования, используемый для кодирования динамически настроенных паролей

- TYPE – string
- DEFAULT – AES/CBC/PKCS5Padding
- IMPORTANCE – low
- DYNAMIC UPDATE MODE – read-only

password.encoder.iterations – Число итераций для кодирования динамически настроенных паролей

- TYPE – int
- DEFAULT – 4096
- VALID VALUES – [1024,...]
- IMPORTANCE – low
- DYNAMIC UPDATE MODE – read-only

password.encoder.key.length – Длина ключа, используемая для кодирования динамически настроенных паролей

- TYPE – int
- DEFAULT – 128
- VALID VALUES – [8,...]
- IMPORTANCE – low
- DYNAMIC UPDATE MODE – read-only

password.encoder.keyfactory.algorithm – Алгоритм SecretKeyFactory, используемый для кодирования динамически настроенных паролей. По умолчанию используется “PBKDF2WithHmacSHA512”, если имеется, и “PBKDF2WithHmacSHA1” в противном случае

- TYPE – string

- DEFAULT – null
- IMPORTANCE – low
- DYNAMIC UPDATE MODE – read-only

quota.window.num – Количество выборок, сохраняемых в памяти для квот клиента

- TYPE – int
- DEFAULT – 11
- VALID VALUES – [1,...]
- IMPORTANCE – low
- DYNAMIC UPDATE MODE – read-only

quota.window.size.seconds – Временной интервал каждой выборки для квот клиента. Указывается в секундах

- TYPE – int
- DEFAULT – 1
- VALID VALUES – [1,...]
- IMPORTANCE – low
- DYNAMIC UPDATE MODE – read-only

replication.quota.window.num – Количество выборок, сохраняемых в памяти для квот репликации

- TYPE – int
- DEFAULT – 11
- VALID VALUES – [1,...]
- IMPORTANCE – low
- DYNAMIC UPDATE MODE – read-only

replication.quota.window.size.seconds – Временной интервал каждой выборки для квот репликации. Указывается в секундах

- TYPE – int
- DEFAULT – 1
- VALID VALUES – [1,...]
- IMPORTANCE – low
- DYNAMIC UPDATE MODE – read-only

ssl.endpoint.identification.algorithm – Алгоритм идентификации конечных точек для валидации имени хоста сервера с использованием сертификата сервера

- TYPE – string
- DEFAULT – null
- IMPORTANCE – low
- DYNAMIC UPDATE MODE – per-broker

ssl.secure.random.implementation – Реализация SecureRandom PRNG, используемая для операций шифрования SSL

- TYPE – string
- DEFAULT – null
- IMPORTANCE – low
- DYNAMIC UPDATE MODE – per-broker

transaction.abort.timed.out.transaction.cleanup.interval.ms – Интервал, в течение которого выполняются отложенные транзакции. Указывается в миллисекундах

- TYPE – int
- DEFAULT – 60000
- VALID VALUES – [1,...]
- IMPORTANCE – low
- DYNAMIC UPDATE MODE – read-only

transaction.remove.expired.transaction.cleanup.interval.ms – Интервал удаления транзакций, срок действия которых истекает по установленному параметру *transactional.id.expiration.ms passing*. Указывается в миллисекундах

- TYPE – int
- DEFAULT – 3600000
- VALID VALUES – [1,...]
- IMPORTANCE – low
- DYNAMIC UPDATE MODE – read-only

zookeeper.sync.time.ms – Удаленность последователя Zookeeper от лидера Zookeeper. Указывается в миллисекундах

- TYPE – int
- DEFAULT – 2000
- IMPORTANCE – low
- DYNAMIC UPDATE MODE – read-only

Более подробную информацию о конфигурации брокера можно найти в классе scala *kafka.server.KafkaConfig*.

6.2 Настройка на уровне топика

Настройки топиков имеют как сервер по умолчанию, так и опциональное переопределение каждого топика. Если топику не задана конфигурация, то используется сервер по умолчанию. Переопределение можно установить во время создания топика, указав один или несколько параметров *–config*.

Далее приведены конфигурации на уровне топика. Настройка сервера по умолчанию указана в Server Default Property. Заданное значение по умолчанию для сервера относится только к топику, если у него нет явного переопределения конфигурации.

cleanup.policy – Определение политики хранения старых сегментов журнала. Политика по умолчанию “delete” отбрасывает старые сегменты при достижении их срока хранения или предельного размера. Значение “compact” включает сжатие журнала по топику

- TYPE – list
- DEFAULT – delete

- VALID VALUES – [compact, delete]
- SERVER DEFAULT PROPERTY –log.cleanup.policy
- IMPORTANCE – medium

compression.type – Окончательный тип сжатия топика. Конфигурация принимает стандартные кодеки сжатия (“gzip”, “snappy”, “lz4”). Также принимает “uncompressed”, что эквивалентно отсутствию сжатия; и “producer”, что означает сохранение исходного кодека сжатия, установленного поставщиком

- TYPE – string
- DEFAULT – producer
- VALID VALUES – [uncompressed, snappy, lz4, gzip, producer]
- SERVER DEFAULT PROPERTY – compression.type
- IMPORTANCE – medium

delete.retention.ms – Время хранения маркированных на удаление данных с целью сжатия топиков журнала. Параметр также дает ограничение на время, в течение которого потребитель должен выполнить чтение, если данные начинаются со смещения *0*, для гарантии валидности снапшота заключительного этапа (в противном случае удаление маркированных данных может произойти до завершения их сканирования). Указывается в миллисекундах

- TYPE – long
- DEFAULT – 86400000
- VALID VALUES – [0,. . .]
- SERVER DEFAULT PROPERTY – log.cleaner.delete.retention.ms
- IMPORTANCE – medium

file.delete.delay.ms – Время ожидания перед удалением файла из файловой системы. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 60000
- VALID VALUES – [0,. . .]
- SERVER DEFAULT PROPERTY – log.segment.delete.delay.ms
- IMPORTANCE – medium

flush.messages – Интервал принудительной синхронизации данных, записанных в журнал. Например, если параметр установлен на *1*, синхронизация выполняется после каждого сообщения; если на значение *5* – после каждых пяти сообщений. Установка данного параметра не рекомендуется, эффективней использовать репликацию для обеспечения устойчивости и возможности фоновой очистки операционной системы. Параметр можно переопределить в базовых настройках каждого топика

- TYPE – long
- DEFAULT – 9223372036854775807
- VALID VALUES – [0,. . .]
- SERVER DEFAULT PROPERTY – log.flush.interval.messages
- IMPORTANCE – medium

flush.ms – Временной интервал принудительной синхронизации данных, записанных в журнал. Например, если параметр установлен на *1000*, синхронизация выполняется по истечении 1000 мс. Установка данного параметра не рекомендуется, эффективней использовать репликацию для обеспечения устойчивости и возможности фоновой очистки операционной системы. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 9223372036854775807
- VALID VALUES – [0,...]
- SERVER DEFAULT PROPERTY – log.flush.interval.ms
- IMPORTANCE – medium

follower.replication.throttled.replicas – Список реплик, для которых репликация журнала должна дросселироваться на стороне подписчика. Список должен описывать набор реплик в формате "[PartitionId]:[BrokerId],[PartitionId]:[BrokerId]:..." или можно использовать специальный символ "*" для дросселирования всех реплик в данном топике

- TYPE – list
- DEFAULT – ""
- VALID VALUES – [partitionId],[brokerId]:[partitionId],[brokerId]:...
- SERVER DEFAULT PROPERTY – follower.replication.throttled.replicas
- IMPORTANCE – medium

index.interval.bytes – Частота добавления индексной записи в индекс смещения. Значение по умолчанию гарантирует индексацию сообщения примерно каждые 4096 байт. Большее индексирование позволяет потребителям приближаться к более точному положению в журнале, но увеличивает сам индекс. Рекомендуется значение не менять

- TYPE – int
- DEFAULT – 4096
- VALID VALUES – [0,...]
- SERVER DEFAULT PROPERTY – log.index.interval.bytes
- IMPORTANCE – medium

leader.replication.throttled.replicas – Список реплик, для которых репликация журнала должна дросселироваться на стороне лидера. Список должен описывать набор реплик в формате "[PartitionId]:[BrokerId],[PartitionId]:[BrokerId]:..." или можно использовать специальный символ "*" для дросселирования всех реплик в данном топике

- TYPE – list
- DEFAULT – ""
- VALID VALUES – [partitionId],[brokerId]:[partitionId],[brokerId]:...
- SERVER DEFAULT PROPERTY – leader.replication.throttled.replicas
- IMPORTANCE – medium

max.message.bytes – Наибольший размер пакета данных, разрешенный ADS. При увеличении параметра следует также увеличить размер выборки для потребителей с целью обеспечения возможности получения пакета данных установленного размера

- TYPE – int
- DEFAULT – 1000012

- VALID VALUES – [0,. . .]
- SERVER DEFAULT PROPERTY – message.max.bytes
- IMPORTANCE – medium

message.format.version – Версия формата сообщений, которую брокер использует для добавления данных в журналы. Значение должно быть действительным ApiVersion. Некоторые примеры: “0.8.2”, “0.9.0.0”, “0.10.0”. Необходимо проверить ApiVersion для получения более подробной информации. Установив версию формата сообщений, пользователь подтверждает, что все существующие данные на диске меньше или равны указанной версии. Неправильное задание параметра приводит к тому, что потребители с более старыми версиями получают данные в нечитаемом формате

- TYPE – string
- DEFAULT – 1.1-IV0
- SERVER DEFAULT PROPERTY – log.message.format.version
- IMPORTANCE – medium

message.timestamp.difference.max.ms – Максимальное допустимое различие между отметкой времени, когда брокер получает сообщение, и отметкой времени, указанной в сообщении. При *message.timestamp.type=CreateTime* сообщение отклоняется, если разница в отметке времени превышает указанный порог. Конфигурация игнорируется, если *message.timestamp.type=LogAppendTime*. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 9223372036854775807
- VALID VALUES – [0,. . .]
- SERVER DEFAULT PROPERTY – log.message.timestamp.difference.max.ms
- IMPORTANCE – medium

message.timestamp.type – Определить, является ли отметка времени в сообщении временем создания сообщения или временем добавления журнала. Параметр может принимать значение “CreateTime” либо “LogAppendTime”

- TYPE – string
- DEFAULT – CreateTime
- SERVER DEFAULT PROPERTY – log.message.timestamp.type
- IMPORTANCE – medium

min.cleanable.dirty.ratio – Частота очистки журнала (при условии включенного сжатия). По умолчанию избегается очистка, где сжато более 50% журнала. Это ограничивает максимальное пространство, выделенное в журнале на дубликаты (не более 50% журнала могут занимать дубликаты). Более высокое отношение означает меньшее количество дубликатов и более эффективную очистку, но при этом большее количество потерянного пространства в журнале

- TYPE – double
- DEFAULT – 0.5
- VALID VALUES – [0,. . . ,1]
- SERVER DEFAULT PROPERTY – log.cleaner.min.cleanable.ratio
- IMPORTANCE – medium

min.compaction.lag.ms – Минимальное время, в течение которого сообщение остается несжатым в журнале. Применяется только для журналов с функцией сжатия. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 0
- VALID VALUES – [0,. . .]
- SERVER DEFAULT PROPERTY – log.cleaner.min.compaction.lag.ms
- IMPORTANCE – medium

min.insync.replicas – При установленном поставщиком подтверждении acks на “all” или “-1”, *min.insync.replicas* задается на минимальное количество реплик для подтверждения записи. Если этот минимум не может быть удовлетворен, то поставщик задает исключение (*NotEnoughReplicas* или *NotEnoughReplicasAfterAppend*). Совместное использование *min.insync.replicas* и acks обеспечивает более высокую гарантию к устойчивости. Типичным сценарием является создание топика с коэффициентом репликации *3*, параметром *min.insync.replicas* равным *2* и acks установленным на “all”. Это гарантирует, что поставщик задает исключение, если большинство реплик не принимает запись

- TYPE – int
- DEFAULT – 1
- VALID VALUES – [1,. . .]
- SERVER DEFAULT PROPERTY – min.insync.replicas
- IMPORTANCE – medium

preallocate – Предварительное выделение файла на диске при создании нового сегмента журнала

- TYPE – boolean
- DEFAULT – false
- SERVER DEFAULT PROPERTY – log.preallocate
- IMPORTANCE – medium

retention.bytes – Контроль максимального размера партиции (состоящей из сегментов журнала), который может увеличиваться до момента отказа от старых сегментов журнала с целью освобождения места при использовании политики хранения “delete”. По умолчанию ограничения по размеру нет, есть только ограничение по времени. Поскольку данный предел применяется на уровне партиции, необходимо умножить значение лимита по времени на количество партиций, чтобы вычислить объем хранения топика в байтах

- TYPE – long
- DEFAULT – - 1
- SERVER DEFAULT PROPERTY – log.retention.bytes
- IMPORTANCE – medium

retention.ms – Контроль максимального времени, в течение которого хранится журнал, прежде чем отбрасываются старые сегменты журнала с целью освобождения места при использовании политики хранения “delete”. Параметр представляет собой SLA о том, как скоро потребители должны читать свои данные. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 604800000
- SERVER DEFAULT PROPERTY – log.retention.ms
- IMPORTANCE – medium

segment.bytes – Контроль размера файла сегмента для журнала. Сохранение и очистка файла всегда выполняются единовременно, поэтому больший размер сегмента означает меньшее количество файлов, но при этом менее гранулированный контроль над хранением

- TYPE – int
- DEFAULT – 1073741824
- VALID VALUES – [14,...]
- SERVER DEFAULT PROPERTY – log.segment.bytes
- IMPORTANCE – medium

segment.index.bytes – Контроль размера индекса, который отображает смещения в позициях файла. Предварительно индексный файл выделяется и сокращается только после сжатия журнала. Обычно параметр не требует изменений

- TYPE – int
- DEFAULT – 10485760
- VALID VALUES – [0,...]
- SERVER DEFAULT PROPERTY – log.index.size.max.bytes
- IMPORTANCE – medium

segment.jitter.ms – Максимальный рандомный джиттер. Вычитается из запланированного времени сжатия сегмента во избежание проблемы сегментации thundering herd (огромное количество процессов, ждущих события, в то время как требуется только один процесс). Указывается в миллисекундах

- TYPE – long
- DEFAULT – 0
- VALID VALUES – [0,...]
- SERVER DEFAULT PROPERTY – log.roll.jitter.ms
- IMPORTANCE – medium

segment.ms – Период времени, после которого ADS выполняет сжатие журнала, даже если файл сегмента не заполнен, с целью обеспечения сохранения или сжатия устаревших данных. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 604800000
- VALID VALUES – [0,...]
- SERVER DEFAULT PROPERTY – log.roll.ms
- IMPORTANCE – medium

unclean.leader.election.enable – Указывает, следует ли включить не входящие в набор ISR реплики и установка последнего средства в качестве лидера, даже если это может привести к потере данных

- TYPE – boolean
- DEFAULT – false
- SERVER DEFAULT PROPERTY – unclean.leader.election.enable
- IMPORTANCE – medium

6.3 Конфигурирование Producer

Далее приведены конфигурации Java-поставщика.

key.serializer – Класс Serializer для ключа, реализующего интерфейс *org.apache.kafka.common.serialization.Serializer*

- TYPE – class
- IMPORTANCE – high

value.serializer – Класс Serializer для значения, реализующего интерфейс *org.apache.kafka.common.serialization.Serializer*

- TYPE – class
- IMPORTANCE – high

acks – Количество подтверждений, которые поставщик требует от лидера перед рассмотрением запроса. Параметр контролирует устойчивость отправляемых записей. Возможны следующие настройки:

- *acks=0* – поставщик не ждет подтверждения с сервера. Запись немедленно добавляется в буфер сокета и считается отправленной. Данная настройка не гарантирует получение сервером записи, и конфигурация повторных попыток не вступает в силу (так как клиент обычно не знает о каких-либо сбоях). Смещение, возвращаемое для каждой записи, всегда равно "-1".
- *acks=1* – лидер фиксирует запись в свой локальный журнал, но отвечает, не дожидаясь полного подтверждения от всех подписчиков. В этом случае лидер может выйти из строя сразу после подтверждения записи и до того, как подписчики реплицируют ее, тогда запись теряется
- *acks=all* – лидер ожидает полного набора синхронизированных реплик для подтверждения записи. Данная настройка гарантирует, что запись не будет потеряна, пока хотя бы одна синхронизированная реплика остается в строе. Это наивысшая гарантия. Эквивалентно настройке *acks=-1*
- TYPE – string
- DEFAULT – 1
- VALID VALUES – [all, - 1, 0, 1]
- IMPORTANCE – high

bootstrap.servers – Список пар хост/порт, используемых для установления первоначального подключения к платформе ADS. В дальнейшем клиент будет использовать все сервера, независимо от того, какие указаны в данном параметре – этот список влияет только на начальные хосты, используемые для обнаружения полного набора серверов. Параметр должен быть задан в формате "host1:port1, host2:port2,..." (через запятую и без пробелов). Поскольку данные сервера используются только для первоначального подключения с целью обнаружения полного набора в кластере (который может динамически меняться), списку необязательно содержать полный набор серверов (можно указать более одного, на случай отказа первого)

- TYPE – list
- DEFAULT – ""
- VALID VALUES – org.apache.kafka.common.config.ConfigDef$NonNullValidator@685cb137
- IMPORTANCE – high

buffer.memory – Общий объем памяти в байтах, которую поставщик может использовать для буферизации записей, ожидающих отправки на сервер. Если записи отправляются быстрее, чем они могут быть доставлены на сервер, поставщик блокирует параметр *max.block.ms*, после чего будет сделано исключение. Параметр должен соответствовать примерно общему объему памяти, которая используется поставщиком, но не полному объему, так как не вся память используется для буферизации. Некоторый дополнительный объем используется для сжатия (если оно включено), а также для поддержания запросов на лету

- TYPE – long
- DEFAULT – 33554432
- VALID VALUES – [0,...]
- IMPORTANCE – high

compression.type – Тип сжатия для всех данных, созданных поставщиком. По умолчанию используется значение “none” (без сжатия). Допустимые значения: “none”, “gzip”, “snappy” и “lz4”. Сжатие выполняется над полной партией данных, поэтому эффективность дозирования влияет на коэффициент сжатия (более многочисленное порционирование означает лучшее сжатие)

- TYPE – string
- DEFAULT – none
- IMPORTANCE – high

retries – Установка значения больше нуля приводит к тому, что клиент переотправляет любую запись, передача которой завершается с временной ошибкой. Повторная попытка ничем не отличается от повторной отправки записи клиентом при получении ошибки. Повторная отправка данных без установки параметра *max.in.flight.requests.per.connection* в значение “1” потенциально может изменить порядок записей, так как если две партии данных отправляются в одну партицию, при этом первая партия не выполняется и повторно отправляется, а вторая выполняется успешно, то данные второго пакета появляются в партиции первыми

- TYPE – int
- DEFAULT – 0
- VALID VALUES – [0,...,2147483647]
- IMPORTANCE – high

ssl.key.password – Пароль закрытого ключа в файле хранилища ключей. Необязательный параметр для клиента

- TYPE – password
- DEFAULT – null
- IMPORTANCE – high

ssl.keystore.location – Расположение файла хранилища ключей. Необязательный параметр для клиента, может использоваться для двусторонней аутентификации клиента

- TYPE – string
- DEFAULT – null
- IMPORTANCE – high

ssl.keystore.password – Пароль хранилища для файла хранения ключей. Необязательный параметр для клиента, требуется только при настройке *ssl.keystore.location*

- TYPE – password
- DEFAULT – null
- IMPORTANCE – high

ssl.truststore.location – Расположение файла хранилища trust store

- TYPE – string
- DEFAULT – null
- IMPORTANCE – high

ssl.truststore.password – Пароль для файла хранилища trust store. При неустановленном пароле доступ к хранилищу есть, но осуществляется с отключенной проверкой надежности

- TYPE – password
- DEFAULT – null
- IMPORTANCE – high

batch.size – При отправке нескольких записей в одну и ту же партицию поставщик пытается объединить их. Это помогает производительности как на клиенте, так и на сервере. Конфигурация управляет размером пакета в байтах. Пакетирование большего размера, чем задан в параметре, не осуществляется. В таком случае отправленные брокерам запросы содержат несколько пакетов (по одному для каждой партиции) с доступными для отправки данными. Небольшой размер пакета делает его менее востребованным и может снизить пропускную способность (нулевой размер пакета полностью отключает пакетирование). Очень большой размер пакета может использовать память расточительно, так как всегда выделяется буфер указанного размера пакета в ожидании дополнительных записей

- TYPE – int
- DEFAULT – 16384
- VALID VALUES – [0,...]
- IMPORTANCE – medium

client.id – Строка id для передачи на сервер при выполнении запросов. Целью является возможность отслеживания источника запросов за пределами ip/port, позволяя включать логическое имя приложения в журнал запросов на стороне сервера

- TYPE – string
- DEFAULT – “”
- IMPORTANCE – medium

connections.max.idle.ms – Закрытие бездействующих соединений по истечению заданного периода. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 540000
- IMPORTANCE – medium

linger.ms – Поставщик объединяет в один пакет все записи, поступающие между трансмиссиями запросов. Обычно это происходит, когда данные поступают быстрее, чем могут быть отправлены. Однако клиент может уменьшить количество запросов даже при умеренной загрузке. Это реализуется путем добавления небольшого промежутка времени искусственной задержки, то есть вместо немедленной отправки данных поставщик ждет до указанной отметки с целью пакетирования данных. Это можно рассматривать как аналог алгоритма Nagle в TCP. Параметр дает верхнюю границу задержки по времени для пакетной обработки. Но как только достигается установленный размер пакета данных *batch.size* для партиции, пакет немедленно отправляется (независимо от заданного параметра *linger.ms*). Однако, имея меньший объем байт пакета, чем в указаном параметре *batch.size*, осуществляется задержка в течение времени, заданного *linger.ms*, с целью ожидания появления новых данных. По умолчанию параметр *linger.ms* равен “0” (то есть без задержки). Например, установка “linger.ms=5” приведет к уменьшению количества отправленных запросов, но добавит до *5 мс* задержки для данных, отправленных при отсутствии нагрузки. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 0
- VALID VALUES – [0,...]

- IMPORTANCE – medium

max.block.ms – Время блокировки *ADSProducer.send()* и *ADSProducer.partitionsFor()*. Данные методы могут быть заблокированы либо по причине заполненного буфера, либо из-за недоступности метаданных. Блокировка в предоставленных пользователем сериализаторах или разделителе не учитывается по таймауту данного параметра. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 60000
- VALID VALUES – [0,...]
- IMPORTANCE – medium

max.request.size – Максимальный размер запроса в байтах. Параметр ограничивает количество пакетов данных, которые поставщик отправляет в одном запросе во избежание отправки огромных запросов. Параметр также эффективно ограничивает максимальный размер пакета данных. При этом сервер имеет свой собственный предел размера пакета данных, который может отличаться от указанного

- TYPE – int
- DEFAULT – 1048576
- VALID VALUES – [0,...]
- IMPORTANCE – medium

partitioner.class – Класс Partitioner, реализующий интерфейс *org.apache.kafka.clients.producer.Partitioner*

- TYPE – class
- DEFAULT – org.apache.kafka.clients.producer.internals.DefaultPartitioner
- IMPORTANCE – medium

receive.buffer.bytes – Размер буфера приема TCP (SO_RCVBUF) при чтении данных. Если значение равно "-1", используется ОС по умолчанию

- TYPE – int
- DEFAULT – 32768
- VALID VALUES – [-1,...]
- IMPORTANCE – medium

request.timeout.ms – Максимальное время ожидания клиентом ответа на запрос. Если ответ не получен до истечения установленного значения, клиент повторно отправляет запрос при необходимости. Значение параметра должно быть больше, чем *replica.lag.time.max.ms* (конфигурация брокера), с целью сокращения возможного дублирования данных по причине излишних попыток поставщика. Указывается в миллисекундах

- TYPE – int
- DEFAULT – 30000
- VALID VALUES – [0,...]
- IMPORTANCE – medium

sasl.jaas.config – Параметры контекста входа JAAS для соединений SSL в формате, используемом файлами конфигурации JAAS. Формат файла конфигурации JAAS описан по ссылке. Формат значения: "(=)*;"

- TYPE – password
- DEFAULT – null
- IMPORTANCE – medium

sasl.kerberos.service.name – Имя принципала Kerberos, которое запускает ADS. Значение можно определить в конфигурации ADS JAAS либо в конфигурации ADS

- TYPE – string
- DEFAULT – null
- IMPORTANCE – medium

sasl.mechanism – Механизм SASL для клиентских подключений. Может быть любой механизм, для которого обеспечивается безопасность. По умолчанию используется GSSAPI

- TYPE – string
- DEFAULT – GSSAPI
- IMPORTANCE – medium

security.protocol – Протокол безопасности для связи между брокерами. Допустимые значения: “PLAINTEXT”, “SSL”, “SASL_PLAINTEXT”, “SASL_SSL”

- TYPE – string
- DEFAULT – PLAINTEXT
- IMPORTANCE – medium

send.buffer.bytes – Размер буфера отправки TCP (SO_SNDBUF) при отправке данных. Если значение равно “-1”, используется ОС по умолчанию

- TYPE – int
- DEFAULT – 131072
- VALID VALUES – [-1,. . .]
- IMPORTANCE – medium

ssl.enabled.protocols – Список протоколов, включенных для соединений SSL

- TYPE – list
- DEFAULT – TLSv1.2,TLSv1.1,TLSv1
- IMPORTANCE – medium

ssl.keystore.type – Формат файла хранилища ключей. Необязательный параметр для клиента

- TYPE – string
- DEFAULT – JKS
- IMPORTANCE – medium

ssl.protocol – Протокол SSL для генерации SSLContext. Значение по умолчанию – “TLS”, что подходит для большинства случаев. Допустимыми значениями в последних JVM являются “TLS”, “TLSv1.1” и “TLSv1.2”. Протоколы “SSL”, “SSLv2” и “SSLv3” могут поддерживаться в более старых JVM, но их использование не рекомендуется из-за известных уязвимостей безопасности

- TYPE – string
- DEFAULT – TLS
- IMPORTANCE – medium

ssl.provider – Имя поставщика безопасности для соединений SSL. Значением по умолчанию является поставщик безопасности по умолчанию для JVM

- TYPE – string

- DEFAULT – null
- IMPORTANCE – medium

ssl.truststore.type – Формат файла хранилища trust store

- TYPE – string
- DEFAULT – JKS
- IMPORTANCE – medium

enable.idempotence – При установленном значении “true” поставщик гарантирует, что ровно одна копия каждого сообщения записывается в поток. При значении “false” в поток могут быть записаны дубликаты сообщений при повторных попытках отправки данных поставщиком из-за сбоев брокера или по другим причинам. Данный параметр требует, чтобы свойство *max.in.flight.requests.per.connection* было меньше или равно “5”, повторные попытки более “0”, и acks установлены на “all”. Если перечисленные настройки явно не заданы пользователем, выбираются подходящие значения. При установке несовместимых значений, выдается ConfigException

- TYPE – boolean
- DEFAULT – false
- IMPORTANCE – low

interceptor.classes – Список классов для использования в качестве интерсепторов. Реализация интерфейса *org.apache.kafka.clients.producer.ProducerInterceptor* позволяет перехватывать (и, возможно, видоизменять) записи, полученные поставщиком до их публикации в кластере ADS. По умолчанию интерсепторы не установлены

- TYPE – list
- DEFAULT – “”
- VALID VALUES – org.apache.kafka.common.config.ConfigDef$NonNullValidator@6a41eaa2
- IMPORTANCE – low

max.in.flight.requests.per.connection – Максимальное количество неподтвержденных запросов, отправляемых клиентом по одному соединению перед блокировкой. Если параметр имеет значение больше *1*, то в случае сбоев существует риск переупорядочения данных из-за повторных попыток (если они включены)

- TYPE – int
- DEFAULT – 5
- VALID VALUES – [1,...]
- IMPORTANCE – low

metadata.max.age.ms – Период времени, после которого принудительно обновляются метаданные даже при отсутствии видимых изменений в лидере партиции с целью предварительного обнаружения новых брокеров или партиций. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 300000
- VALID VALUES – [0,...]
- IMPORTANCE – low

metric.reporters – Список классов для использования в качестве репортеров метрик. Реализация интерфейса *org.apache.kafka.common.metrics.MetricsReporter* позволяет подключать классы, которые будут уведомлены о создании новой метрики. JmxReporter всегда включен в реестр статистических данных JMX

- TYPE – list
- DEFAULT – “”
- VALID VALUES – org.apache.kafka.common.config.ConfigDef$NonNullValidator@7cd62f43
- IMPORTANCE – low

metrics.num.samples – Количество выборок, поддерживаемых для вычисления метрик

- TYPE – int
- DEFAULT – 2
- VALID VALUES – [1,...]
- IMPORTANCE – low

metrics.recording.level – Самый высокий уровень записи для метрик

- TYPE – string
- DEFAULT – INFO
- VALID VALUES – [INFO, DEBUG]
- IMPORTANCE – low

metrics.sample.window.ms – Время ожидания вычисления метрик выборки. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 30000
- VALID VALUES – [0,...]
- IMPORTANCE – low

reconnect.backoff.max.ms – Максимальный период времени ожидания повторного подключения к брокеру при неоднократных сбоях соединения. Отсрочка на хост увеличивается экспоненциально для каждого последующего сбоя соединения, вплоть до установленного максимума. После расчета увеличения отсрочки к значению добавляется *20%* случайного джиттера во избежание помех связи. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 1000
- VALID VALUES – [0,...]
- IMPORTANCE – low

reconnect.backoff.ms – Базовый период времени ожидания повторного подключения к хосту. Позволяет избегать многократного подключения к узлу в узком цикле. Данная отсрочка применяется ко всем попыткам подключения клиента к брокеру. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 50
- VALID VALUES – [0,...]
- IMPORTANCE – low

retry.backoff.ms – Время ожидания перед повторной попыткой отправки неудавшегося запроса в партицию топика. Указывается в миллисекундах

- TYPE – long

- DEFAULT – 100
- VALID VALUES – [0,...]
- IMPORTANCE – low

sasl.kerberos.kinit.cmd – Путь команд Kerberos kinit

- TYPE – string
- DEFAULT – /usr/bin/kinit
- IMPORTANCE – low

sasl.kerberos.min.time.before.relogin – Время ожидания авторизации потока между попытками обновления

- TYPE – long
- DEFAULT – 60000
- IMPORTANCE – low

sasl.kerberos.ticket.renew.jitter – Процент случайного джиттера по отношению к времени возобновления

- TYPE – double
- DEFAULT – 0.05
- IMPORTANCE – low

sasl.kerberos.ticket.renew.window.factor – Время ожидания авторизации потока до тех пор, пока не будет достигнут указанный коэффициент времени от последнего обновления до истечения срока действия тикета, и попытка возобновления тикета за этот период времени

- TYPE – double
- DEFAULT – 0.8
- IMPORTANCE – low

ssl.cipher.suites – Список наборов шифров. Именованная комбинация аутентификации, шифрования, MAC и ключей обмена алгоритма для согласования параметров безопасности для сетевого подключения с использованием протокола TLS или SSL. По умолчанию поддерживаются все доступные варианты шифрования

- TYPE – list
- DEFAULT – null
- IMPORTANCE – low

ssl.endpoint.identification.algorithm – Алгоритм идентификации конечных точек для валидации имени хоста сервера с использованием сертификата сервера

- TYPE – string
- DEFAULT – null
- IMPORTANCE – low

ssl.keymanager.algorithm – Алгоритм службы управления ключами для SSL-соединений. Значением по умолчанию является алгоритм, настроенный для Java Virtual Machine

- TYPE – string
- DEFAULT – SunX509
- IMPORTANCE – low

ssl.secure.random.implementation – Реализация SecureRandom PRNG, используемая для операций шифрования SSL

- TYPE – string
- DEFAULT – null
- IMPORTANCE – low

ssl.trustmanager.algorithm – Алгоритм доверенной службы управления ключами для SSL-соединений. Значением по умолчанию является алгоритм, настроенный для Java Virtual Machine

- TYPE – string
- DEFAULT – PKIX
- IMPORTANCE – low

transaction.timeout.ms – Максимальный интервал времени, который координатор транзакции ожидает для обновления статуса транзакции от поставщика перед тем, как будет прервана текущая транзакция. Если установленное значение больше, чем значение *transaction.max.timeout.ms* в настройках брокера, запрос завершается ошибкой *InvalidTransactionTimeout*

- TYPE – int
- DEFAULT – 60000
- IMPORTANCE – low

transactional.id – Идентификатор транзакции. Параметр позволяет использовать семантику достоверности, которая охватывает несколько сессий поставщика, и позволяет гарантировать клиенту, что транзакции, использующие тот же TransactionalId, завершены до начала любых новых транзакций. Если TransactionalId не указан, то поставщик ограничивается идемпотентной доставкой. Важно, что параметр *enable.idempotence* должен быть включен при сконфигурированном *TransactionalId*. Значение по умолчанию “null”, что означает невозможность использования транзакций. Для транзакций требуется по меньшей мере три брокера по умолчанию, что является рекомендуемым параметром для продуктивной системы; для разработки можно изменить настройки в параметре брокера *transaction.state.log.replication.factor*

- TYPE – string
- DEFAULT – null
- VALID VALUES – non-empty string
- IMPORTANCE – low

6.4 Конфигурирование Consumer

Далее приведены конфигурации для нового потребителя.

key.deserializer – Класс десериализатора для ключа, реализующего интерфейс *org.apache.kafka.common.serialization.Deserializer*

- TYPE – class
- IMPORTANCE – high

value.deserializer – Класс десериализатора для значения, реализующего интерфейс *org.apache.kafka.common.serialization.Deserializer*

- TYPE – class
- IMPORTANCE – high

bootstrap.servers – Список пар хост/порт, используемых для установления первоначального подключения к платформе ADS. В дальнейшем клиент будет использовать все сервера, независимо от того, какие указаны в данном параметре – этот список влияет только на начальные хосты, используемые для обнаружения полного набора серверов. Параметр должен быть задан в формате "host1:port1, host2:port2,..." (через запятую и без пробелов). Поскольку данные сервера используются только для первоначального подключения с целью обнаружения полного набора в кластере (который может динамически меняться), списку необязательно содержать полный набор серверов (можно указать более одного, на случай отказа первого)

- TYPE – list
- DEFAULT – ""
- VALID VALUES – org.apache.kafka.common.config.ConfigDef$NonNullValidator@7cd62f43
- IMPORTANCE – high

fetch.min.bytes – Минимальный объем данных, которые сервер должен вернуть по запросу на выборку. При недостаточном объеме данных запрос ожидает их накопления до установленного значения. Значение по умолчанию *1 байт* означает, что запросы на выборку отвечают, как только доступен *1 байт* данных, или по истечению времени ожидания запроса. Установка большего значения заставляет сервер ожидать больших объемов данных для накопления, что может немного повысить пропускную способность сервера за счет некоторой дополнительной задержки

- TYPE – int
- DEFAULT – 1
- VALID VALUES – [0,...]
- IMPORTANCE – high

group.id – Уникальная строка, идентифицирующая группу потребителей. Свойство требуется, если потребитель использует функциональность группового управления с помощью подписки (топика) или стратегии управления смещением на основе ADS

- TYPE – string
- DEFAULT – ""
- IMPORTANCE – high

heartbeat.interval.ms – Время ожидания для координатора потребителя между heartbeat-сообщениями при использовании средств группового управления ADS. Heartbeat-сообщения используются для обеспечения активности сессии потребителя и перебалансировки, когда пользователи присоединяются или покидают группу. Значение должно быть установлено ниже, чем параметр *session.timeout.ms* (обычно оно не более *1/3* от этого значения). Его можно настроить еще ниже с целью контроля ожидаемого времени для нормальных перебалансировок. Указывается в миллисекундах

- TYPE – int
- DEFAULT – 3000
- IMPORTANCE – high

max.partition.fetch.bytes – Максимальный объем данных для каждой партиции, который будет возвращен серверу. Из пакетов записи извлекаются потребителями. Если первый пакет записей в первой непустой партиции выборки больше установленного значения, пакет данных все равно будет возвращен для обеспечения гарантии возможности выполнения. Максимальный размер пакета записей, принятый брокером, определяется через *message.max.bytes* (конфигурация брокера) или *max.message.bytes* (конфигурация топика). Для ограничения размера запроса потребителя используется параметр *fetch.max.bytes*

- TYPE – int
- DEFAULT – 1048576

- VALID VALUES – [0,...]
- IMPORTANCE – high

session.timeout.ms – Время ожидания для выявления сбоев потребителей при использовании группового управления ADS. Потребитель посылает периодические heartbeat-сообщения брокеру с целью подтверждения своей активности. Если брокер не получает ни одного бита до истечения установленного времени сеанса, то удаляет данного потребителя из группы и начинает балансировку. Значение параметра должно быть в допустимом диапазоне конфигурации брокера *group.min.session.timeout.ms* и *group.max.session.timeout.ms*. Указывается в миллисекундах

- TYPE – int
- DEFAULT – 10000
- IMPORTANCE – high

ssl.key.password – Пароль закрытого ключа в файле хранилища ключей. Необязательный параметр для клиента

- TYPE – password
- DEFAULT – null
- IMPORTANCE – high

ssl.keystore.location – Расположение файла хранилища ключей. Необязательный параметр для клиента, может использоваться для двусторонней аутентификации клиента

- TYPE – string
- DEFAULT – null
- IMPORTANCE – high

ssl.keystore.password – Пароль хранилища для файла хранения ключей. Необязательный параметр для клиента, требуется только при настройке *ssl.keystore.location*

- TYPE – password
- DEFAULT – null
- IMPORTANCE – high

ssl.truststore.location – Расположение файла хранилища trust store

- TYPE – string
- DEFAULT – null
- IMPORTANCE – high

ssl.truststore.password – Пароль для файла хранилища trust store. При неустановленном пароле доступ к хранилищу есть, но осуществляется с отключенной проверкой надежности

- TYPE – password
- DEFAULT – null
- IMPORTANCE – null

auto.offset.reset – В случае если в ADS нет начального смещения или текущее смещение больше не существует на сервере (например, так как данные удалены):

- *earliest*: автоматически сбросить смещение до самого раннего смещения;
- *latest*: автоматически сбросить смещение до последнего смещения;

- *none*: исключение для потребителя, если предыдущее смещение для группы потребителей не найдено;
- *anything else*: исключение для потребителя.
- TYPE – string
- DEFAULT – latest
- VALID VALUES – [latest, earliest, none]
- IMPORTANCE – medium

connections.max.idle.ms – Закрытие бездействующих соединений по истечению заданного периода. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 540000
- IMPORTANCE – medium

enable.auto.commit – При значении “true” смещение потребителя периодически фиксируется в фоновом режиме

- TYPE – boolean
- DEFAULT – true
- IMPORTANCE – medium

exclude.internal.topics – Предоставление потребителю записей из внутренних топиков (например, смещения). При значении “true” единственным способом получения записей из внутреннего топика является подписка на него

- TYPE – boolean
- DEFAULT – true
- IMPORTANCE – medium

fetch.max.bytes – Максимальный объем данных, который сервер должен вернуть для запроса на выборку. Параметр не является абсолютным максимумом. Из пакетов записи извлекаются потребителями. Если первый пакет записей в первой непустой партиции выборки больше установленного значения, пакет данных все равно будет возвращен для обеспечения гарантии возможности выполнения. Максимальный размер пакета записей, принятый брокером, определяется через *message.max.bytes* (конфигурация брокера) или *max.message.bytes* (конфигурация топика). При этом потребитель параллельно выполняет несколько выборок

- TYPE – int
- DEFAULT – 52428800
- VALID VALUES – [0,. . .]
- IMPORTANCE – medium

isolation.level – Контроль транзакционно записанных данных. Если установлено значение “read_committed”, *consumer.poll()* возвращает только совершенные транзакционные сообщения. При значении “read_uncommitted” (по умолчанию), *consumer.poll()* возвращает все сообщения, даже прерванные. Нетранзакционные сообщения возвращаются в любом режиме. Данные всегда возвращаются в порядке смещения. Следовательно, в режиме “read_committed” *consumer.poll()* возвращает сообщения только до последнего стабильного смещения (LSO), которое меньше, чем смещение первой открытой транзакции. То есть любые данные, появляющиеся после текущей транзакциии, удерживаются до завершения соответствующей транзакции. Потребители “read_committed” не могут считывать высокий водяной знак в процессе транзакции. Так же в режиме “read_committed” метод *seekToEnd* возвращает LSO

- TYPE – string

- DEFAULT – read_uncommitted
- VALID VALUES – [read_committed, read_uncommitted]
- IMPORTANCE – medium

max.poll.interval.ms – Максимальная задержка времени между вызовами *poll()* при управлении группами потребителей. Параметр устанавливает верхнюю границу времени, в течение которого потребитель может бездействовать. Если *poll()* не вызывается до истечения установленного таймаута, потребитель считается неисправным, и группа перебалансируется с целью переназначения партиций. Указывается в миллисекундах

- TYPE – int
- DEFAULT – 300000
- VALID VALUES – [1,...]
- IMPORTANCE – medium

max.poll.records – Максимальное число записей, возвращаемых за один вызов *poll()*

- TYPE – int
- DEFAULT – 500
- VALID VALUES – [1,...]
- IMPORTANCE – medium

partition.assignment.strategy – Класс стратегии назначения партиций, которую клиент использует для распределения принадлежности партиции инстансам потребителя при групповом управлении

- TYPE – list
- DEFAULT – class org.apache.kafka.clients.consumer.RangeAssignor
- VALID VALUES – org.apache.kafka.common.config.ConfigDef$NonNullValidator@6d4b1c02
- IMPORTANCE – medium

receive.buffer.bytes – Размер буфера приема TCP (SO_RCVBUF) при чтении данных. Если значение равно "-1", используется ОС по умолчанию

- TYPE – int
- DEFAULT – 65536
- VALID VALUES – [-1,...]
- IMPORTANCE – medium

request.timeout.ms – Максимальное время ожидания клиентом ответа на запрос. Если ответ не получен до истечения установленного значения, клиент повторно отправляет запрос при необходимости. Указывается в миллисекундах

- TYPE – int
- DEFAULT – 305000
- VALID VALUES – [0,...]
- IMPORTANCE – medium

sasl.jaas.config – Параметры контекста входа JAAS для соединений SSL в формате, используемом файлами конфигурации JAAS. Формат файла конфигурации JAAS описан по ссылке. Формат значения: "(=)*;"

- TYPE – password
- DEFAULT – null

- IMPORTANCE – medium

sasl.kerberos.service.name – Имя принципала Kerberos, которое запускает ADS. Значение можно определить в конфигурации ADS JAAS либо в конфигурации ADS

- TYPE – string
- DEFAULT – null
- IMPORTANCE – medium

sasl.mechanism – Механизм SASL для клиентских подключений. Может быть любой механизм, для которого обеспечивается безопасность. По умолчанию используется GSSAPI

- TYPE – string
- DEFAULT – GSSAPI
- IMPORTANCE – medium

security.protocol – Протокол безопасности для связи между брокерами. Допустимые значения: “PLAINTEXT”, “SSL”, “SASL_PLAINTEXT”, “SASL_SSL”

- TYPE – string
- DEFAULT – PLAINTEXT
- IMPORTANCE – medium

send.buffer.bytes – Буфер SO_SNDBUF сокета сервера сокетов. При значении параметра “-1” используется ОС по умолчанию

- TYPE – int
- DEFAULT – 131072
- VALID VALUES – [-1,...]
- IMPORTANCE – medium

ssl.enabled.protocols – Список протоколов, включенных для соединений SSL

- TYPE – list
- DEFAULT – TLSv1.2,TLSv1.1,TLSv1
- IMPORTANCE – medium

ssl.keystore.type – Формат файла хранилища ключей. Необязательный параметр для клиента

- TYPE – string
- DEFAULT – JKS
- IMPORTANCE – medium

ssl.protocol – Протокол SSL для генерации SSLContext. Значение по умолчанию – “TLS”, что подходит для большинства случаев. Допустимыми значениями в последних JVM являются “TLS”, “TLSv1.1” и “TLSv1.2”. Протоколы “SSL”, “SSLv2” и “SSLv3” могут поддерживаться в более старых JVM, но их использование не рекомендуется из-за известных уязвимостей безопасности

- TYPE – string
- DEFAULT – TLS
- IMPORTANCE – medium

ssl.provider – Имя поставщика безопасности для соединений SSL. Значением по умолчанию является поставщик безопасности по умолчанию для JVM

- TYPE – string
- DEFAULT – null
- IMPORTANCE – medium

ssl.truststore.type – Формат файла хранилища trust store

- TYPE – string
- DEFAULT – JKS
- IMPORTANCE – medium

auto.commit.interval.ms – Частота автофиксации потребительских смещений при включенном параметре *enable.auto.commit* (значение "true"). Указывается в миллисекундах

- TYPE – int
- DEFAULT – 5000
- VALID VALUES – [0,...]
- IMPORTANCE – low

check.crcs – Автоматическая проверка CRC32 считываемых записей. Проверка добавляет некоторые накладные расходы, поэтому она может быть отключена в случаях, требующих высокой производительности

- TYPE – boolean
- DEFAULT – true
- IMPORTANCE – low

client.id – Строка id для передачи на сервер при выполнении запросов. Целью является возможность отслеживания источника запросов за пределами ip/port, позволяя включать логическое имя приложения в журнал запросов на стороне сервера

- TYPE – string
- DEFAULT – ""
- IMPORTANCE – low

fetch.max.wait.ms – Максимальный период времени, в течение которого сервер блокируется, прежде чем ответить на запрос выборки (в случае недостаточного объема данных для незамедлительного ответа, заданного функцией *fetch.min.bytes*). Указывается в миллисекундах

- TYPE – int
- DEFAULT – 500
- VALID VALUES – [0,...]
- IMPORTANCE – low

interceptor.classes – Список классов для использования в качестве интерсепторов. Реализация интерфейса *org.apache.kafka.clients.consumer.ConsumerInterceptor* позволяет перехватывать (и, возможно, видоизменять) полученные потребителем записи. По умолчанию интерсепторы не установлены

- TYPE – list
- DEFAULT – ""
- VALID VALUES – org.apache.kafka.common.config.ConfigDef$NonNullValidator@6093dd95
- IMPORTANCE – low

metadata.max.age.ms – Период времени, после которого принудительно обновляются метаданные даже при отсутствии видимых изменений в лидере партиции с целью предварительного обнаружения новых брокеров или партиций. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 300000
- VALID VALUES – [0,...]
- IMPORTANCE – low

metric.reporters – Список классов для использования в качестве репортеров метрик. Реализация интерфейса *org.apache.kafka.common.metrics.MetricsReporter* позволяет подключать классы, которые будут уведомлены о создании новой метрики. JmxReporter всегда включен в реестр статистических данных JMX

- TYPE – list
- DEFAULT – “”
- VALID VALUES – org.apache.kafka.common.config.ConfigDef$NonNullValidator@5622fdf
- IMPORTANCE – low

metrics.num.samples – Количество выборок, поддерживаемых для вычисления метрик

- TYPE – int
- DEFAULT – 2
- VALID VALUES – [1,...]
- IMPORTANCE – low

metrics.recording.level – Самый высокий уровень записи для метрик

- TYPE – string
- DEFAULT – INFO
- VALID VALUES – [INFO, DEBUG]
- IMPORTANCE – low

metrics.sample.window.ms – Время ожидания вычисления метрик выборки. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 30000
- VALID VALUES – [0,...]
- IMPORTANCE – low

reconnect.backoff.max.ms – Максимальный период времени ожидания повторного подключения к брокеру при неоднократных сбоях соединения. Отсрочка на хост увеличивается экспоненциально для каждого последующего сбоя соединения, вплоть до установленного максимума. После расчета увеличения отсрочки к значению добавляется *20%* случайного джиттера во избежание помех связи. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 1000
- VALID VALUES – [0,...]
- IMPORTANCE – low

reconnect.backoff.ms – Базовый период времени ожидания повторного подключения к хосту. Позволяет избегать многократного подключения к узлу в узком цикле. Данная отсрочка применяется ко всем попыткам подключения клиента к брокеру. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 50
- VALID VALUES – [0,...]
- IMPORTANCE – low

retry.backoff.ms – Время ожидания перед повторной попыткой отправки неудавшегося запроса в партицию топика. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 100
- VALID VALUES – [0,...]
- IMPORTANCE – low

sasl.kerberos.kinit.cmd – Путь команд Kerberos kinit

- TYPE – string
- DEFAULT – /usr/bin/kinit
- IMPORTANCE – low

sasl.kerberos.min.time.before.relogin – Время ожидания авторизации потока между попытками обновления

- TYPE – long
- DEFAULT – 60000
- IMPORTANCE – low

sasl.kerberos.ticket.renew.jitter – Процент случайного джиттера по отношению к времени возобновления

- TYPE – double
- DEFAULT – 0.05
- IMPORTANCE – low

sasl.kerberos.ticket.renew.window.factor – Время ожидания авторизации потока до тех пор, пока не будет достигнут указанный коэффициент времени от последнего обновления до истечения срока действия тикета, и попытка возобновления тикета за этот период времени

- TYPE – double
- DEFAULT – 0.8
- IMPORTANCE – low

ssl.cipher.suites – Список наборов шифров. Именованная комбинация аутентификации, шифрования, MAC и ключей обмена алгоритма для согласования параметров безопасности для сетевого подключения с использованием протокола TLS или SSL. По умолчанию поддерживаются все доступные варианты шифрования

- TYPE – list
- DEFAULT – null
- IMPORTANCE – low

ssl.endpoint.identification.algorithm – Алгоритм идентификации конечных точек для валидации имени хоста сервера с использованием сертификата сервера

- TYPE – string
- DEFAULT – null
- IMPORTANCE – low

ssl.keymanager.algorithm – Алгоритм службы управления ключами для SSL-соединений. Значением по умолчанию является алгоритм, настроенный для Java Virtual Machine

- TYPE – string
- DEFAULT – SunX509
- IMPORTANCE – low

ssl.secure.random.implementation – Реализация SecureRandom PRNG, используемая для операций шифрования SSL

- TYPE – string
- DEFAULT – null
- IMPORTANCE – low

ssl.trustmanager.algorithm – Алгоритм доверенной службы управления ключами для SSL-соединений. Значением по умолчанию является алгоритм, настроенный для Java Virtual Machine

- TYPE – string
- DEFAULT – PKIX
- IMPORTANCE – low

6.5 Конфигурирование Streams

Далее приведены конфигурации для клиентской библиотеки ADS Streams.

application.id – Идентификатор приложения потоковой обработки. Должен быть уникальным в пределах платформы ADS. Используется как: 1) префикс идентификатора клиента по умолчанию, 2) идентификатор группы для управления членством, 3) префикс топика изменений в журнале

- TYPE – string
- IMPORTANCE – high

bootstrap.servers – Список пар хост/порт, используемых для установления первоначального подключения к платформе ADS. В дальнейшем клиент будет использовать все сервера, независимо от того, какие указаны в данном параметре – этот список влияет только на начальные хосты, используемые для обнаружения полного набора серверов. Параметр должен быть задан в формате “host1:port1, host2:port2,...” (через запятую и без пробелов). Поскольку данные сервера используются только для первоначального подключения с целью обнаружения полного набора в кластере (который может динамически меняться), списку необязательно содержать полный набор серверов (можно указать более одного, на случай отказа первого)

- TYPE – list
- IMPORTANCE – high

replication.factor – Коэффициент репликации для топиков журнала изменений и репартицирования топиков, созданных приложением потоковой обработки

- TYPE – int

- DEFAULT – 1
- IMPORTANCE – high

state.dir – Местоположение каталога хранилища state store

- TYPE – string
- DEFAULT – /tmp/kafka-streams
- IMPORTANCE – high

cache.max.bytes.buffering – Максимальный объем памяти в байтах, используемый для буферизации всех потоков

- TYPE – long
- DEFAULT – 10485760
- VALID VALUES – [0,...]
- IMPORTANCE – medium

client.id – Строка префикса ID, используемая для идентификаторов клиента внутреннего потребителя и поставщика (восстановления потребителя); шаблон "-StreamThread–"

- TYPE – string
- DEFAULT – ""
- IMPORTANCE – medium

default.deserialization.exception.handler – Класс обработки исключений, реализующий интерфейс *org.apache.kafka.streams.errors.DeserializationExceptionHandler*

- TYPE – class
- DEFAULT – org.apache.kafka.streams.errors.LogAndFailExceptionHandler
- IMPORTANCE – medium

default.key.serde – Класс сериализатора/десериализатора по умолчанию для ключа, реализующего интерфейс *org.apache.kafka.common.serialization.Serde*

- TYPE – class
- DEFAULT – org.apache.kafka.common.serialization.Serdes$ByteArraySerde
- IMPORTANCE – medium

default.production.exception.handler – Класс обработки исключений, реализующий интерфейс *org.apache.kafka.streams.errors.ProductionExceptionHandler*

- TYPE – class
- DEFAULT – org.apache.kafka.streams.errors.DefaultProductionExceptionHandler
- IMPORTANCE – medium

default.timestamp.extractor – Класс выделения временных меток по умолчанию, реализующий интерфейс *org.apache.kafka.streams.processor.TimestampExtractor*

- TYPE – class
- DEFAULT – org.apache.kafka.streams.processor.FailOnInvalidTimestamp
- IMPORTANCE – medium

default.value.serde – Класс сериализатора/десериализатора по умолчанию для значения, реализующего интерфейс *org.apache.kafka.common.serialization.Serde*

- TYPE – class
- DEFAULT – org.apache.kafka.common.serialization.Serdes$ByteArraySerde
- IMPORTANCE – medium

num.standby.replicas – Число резервных реплик для каждой задачи

- TYPE – int
- DEFAULT – 0
- IMPORTANCE – medium

num.stream.threads – Количество потоков для выполнения потоковой обработки

- TYPE – int
- DEFAULT – 1
- IMPORTANCE – medium

processing.guarantee – Гарантия на обработку. Возможные значения: “at_least_once” (по умолчанию) и “exact_once”. Для обработки “exact_once” требуется по меньшей мере три брокера по умолчанию, что является рекомендуемой настройкой для продуктивной среды; для разработки можно изменить параметр, при этом переустановив настройку брокера *transaction.state.log.replication.factor*

- TYPE – string
- DEFAULT – at_least_once
- VALID VALUES – [at_least_once, exactly_once]
- IMPORTANCE – medium

security.protocol – Протокол безопасности для связи между брокерами. Допустимые значения: “PLAINTEXT”, “SSL”, “SASL_PLAINTEXT”, “SASL_SSL”

- TYPE – string
- DEFAULT – PLAINTEXT
- IMPORTANCE – medium

application.server – Пара “host:port”, указывающая на встроенную конечную точку пользователя, которая может использоваться для обнаружения местоположений хранилищ state stores в рамках одного приложения

- TYPE – string
- DEFAULT – “”
- IMPORTANCE – low

buffered.records.per.partition – Максимальное количество записей в буфере для каждой партиции

- TYPE – int
- DEFAULT – 1000
- IMPORTANCE – low

commit.interval.ms – Частота сохранения положения процессора. Если параметр *processing.guarantee* установлен на “exactly_once”, значение по умолчанию равно “100”, иначе (при “at_least_once”) значение по умолчанию равно “30000”. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 30000
- IMPORTANCE – low

connections.max.idle.ms – Закрытие бездействующих соединений по истечению заданного периода. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 540000
- IMPORTANCE – low

key.serde – Сериализатор/десериализатор для ключа, реализующего интерфейс *org.apache.kafka.common.serialization.Serde*. Данная конфигурация устарела, вместо нее используется *default.key.serde*

- TYPE – class
- DEFAULT – null
- IMPORTANCE – low

metadata.max.age.ms – Период времени, после которого принудительно обновляются метаданные даже при отсутствии видимых изменений в лидере партиции с целью предварительного обнаружения новых брокеров или партиций. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 300000
- VALID VALUES – [0,. . .]
- IMPORTANCE – low

metric.reporters – Список классов для использования в качестве репортеров метрик. Реализация интерфейса *org.apache.kafka.common.metrics.MetricsReporter* позволяет подключать классы, которые будут уведомлены о создании новой метрики. JmxReporter всегда включен в реестр статистических данных JMX

- TYPE – list
- DEFAULT – “”
- IMPORTANCE – low

metrics.num.samples – Количество выборок, поддерживаемых для вычисления метрик

- TYPE – int
- DEFAULT – 2
- VALID VALUES – [1,. . .]
- IMPORTANCE – low

metrics.recording.level – Самый высокий уровень записи для метрик

- TYPE – string
- DEFAULT – INFO
- VALID VALUES – [INFO, DEBUG]
- IMPORTANCE – low

metrics.sample.window.ms – Время ожидания вычисления метрик выборки. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 30000
- VALID VALUES – [0,...]
- IMPORTANCE – low

partition.grouper – Класс Partition grouper, реализующий интерфейс *org.apache.kafka.streams.processor.PartitionGrouper*

- TYPE – class
- DEFAULT – org.apache.kafka.streams.processor.DefaultPartitionGrouper
- IMPORTANCE – low

poll.ms – Время блокировки ожидания ввода. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 100
- IMPORTANCE – low

receive.buffer.bytes – Размер буфера приема TCP (SO_RCVBUF) при чтении данных. Если значение равно "-1", используется ОС по умолчанию

- TYPE – int
- DEFAULT – 32768
- VALID VALUES – [0,...]
- IMPORTANCE – low

reconnect.backoff.max.ms – Максимальный период времени ожидания повторного подключения к брокеру при неоднократных сбоях соединения. Отсрочка на хост увеличивается экспоненциально для каждого последующего сбоя соединения, вплоть до установленного максимума. После расчета увеличения отсрочки к значению добавляется *20%* случайного джиттера во избежание помех связи. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 1000
- VALID VALUES – [0,...]
- IMPORTANCE – low

reconnect.backoff.ms – Базовый период времени ожидания повторного подключения к хосту. Позволяет избегать многократного подключения к узлу в узком цикле. Данная отсрочка применяется ко всем попыткам подключения клиента к брокеру. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 50
- VALID VALUES – [0,...]
- IMPORTANCE – low

request.timeout.ms – Максимальное время ожидания клиентом ответа на запрос. Если ответ не получен до истечения установленного значения, клиент повторно отправляет запрос при необходимости. Указывается в миллисекундах

- TYPE – int
- DEFAULT – 40000

- VALID VALUES – [0,...]
- IMPORTANCE – low

retries – Установка значения больше нуля приводит к тому, что клиент переотправляет любую запись, передача которой завершается с временной ошибкой

- TYPE – int
- DEFAULT – 0
- VALID VALUES – [0,...,2147483647]
- IMPORTANCE – low

retry.backoff.ms – Время ожидания перед повторной попыткой отправки неудавшегося запроса в партицию топика. Позволяет избежать неоднократной отправки запросов в сжатом цикле. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 100
- VALID VALUES – [0,...]
- IMPORTANCE – low

rocksdb.config.setter – Класс или имя класса установщика конфигурации базы данных Rocks, реализующий интерфейс *org.apache.kafka.streams.state.RocksDBConfigSetter*

- TYPE – class
- DEFAULT – null
- IMPORTANCE – low

send.buffer.bytes – Размер буфера отправки TCP (SO_SNDBUF) при отправке данных. Если значение равно "-1", используется ОС по умолчанию

- TYPE – int
- DEFAULT – 131072
- VALID VALUES – [0,...]
- IMPORTANCE – low

state.cleanup.delay.ms – Время ожидания перед удалением state каталогов после перемещения партиции. Удаляются только state каталоги, которые не были изменены. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 600000
- IMPORTANCE – low

timestamp.extractor – Класс выделения временных меток, реализующий интерфейс *org.apache.kafka.streams.processor.TimestampExtractor*. Данная конфигурация устарела, вместо нее используется *default.timestamp.extractor*

- TYPE – class
- DEFAULT – null
- IMPORTANCE – low

value.serde – Класс сериализатора/десериализатора для значения, реализующего интерфейс *org.apache.kafka.common.serialization.Serde*. Данная конфигурация устарела, вместо нее используется *ddefault.value.serde*

- TYPE – class
- DEFAULT – null
- IMPORTANCE – low

windowstore.changelog.additional.retention.ms – Добавление *maintainMs* с целью исключения риска преждевременного удаления данных из журнала. Позволяет осуществлять отставание часов. По умолчанию устанавливается 1 день. Указывается в миллисекундах

- TYPE – long
- DEFAULT – 86400000
- IMPORTANCE – low

zookeeper.connect – Соединение строки для управления топиками ADS через Zookeeper. Конфигурация устарела и игнорируется, поскольку Streams API больше не использует Zookeeper

- TYPE – string
- DEFAULT – “”
- IMPORTANCE – low

6.6 Удаление/Добавление компонентов сервиса Kafka

Доступно с версии 1.4.11

Если кластер **ADS** разворачивается с помощью **ADCM**, то операции по добавлению/удалению хоста в сервис *Kafka* могут быть выполнены автоматически. После выполнения планирования нового аппаратного обеспечения необходимо добавить новые хосты в выбранный кластер в интерфейсе **ADCM**, используя кнопку “Add hosts” на вкладке “Hosts”. Кроме того, необходимо выполнить инициализацию каждого хоста, если того требует провайдер хостов.

Important: Описанные ниже операции не удаляют/добавляют хост из кластера – они лишь управляют компонентом *Kafka Broker* на хостах. Удаление хоста из кластера возможно в разделе “Hosts” кластера при условии, что к хосту не привязан ни один компонент

Для добавления или удаления *Kafka Broker* с хостов необходимо воспользоваться соответствующими кнопками выпадающего меню, доступного по нажатию на иконку в поле “Actions” сервиса *Kafka* (Рис.6.1).

6.6.1 Добавление компонентов Kafka Broker

Когда хосты становятся доступными для подключения по ssh для менеджера кластеров, необходимо выбрать действие *Expand* сервиса *Kafka* из списка возможных операций. В появившемся диалоговом окне предоставляется выбор опций (Рис.6.2):

- *Disable SELinux before cluster installation* – отключение SELinux на добавляемых хостах. Для того, чтобы данная настройка применилась, после завершения операции *Expand* необходимо перезагрузить хосты вручную;
- *Disable Firewalld before cluster installation* – выключение firewalld на добавляемых хостах;
- *Install OpenJDK before cluster installation* – установка пакета *java-1.8.0-openjdk* на добавляемых хостах;

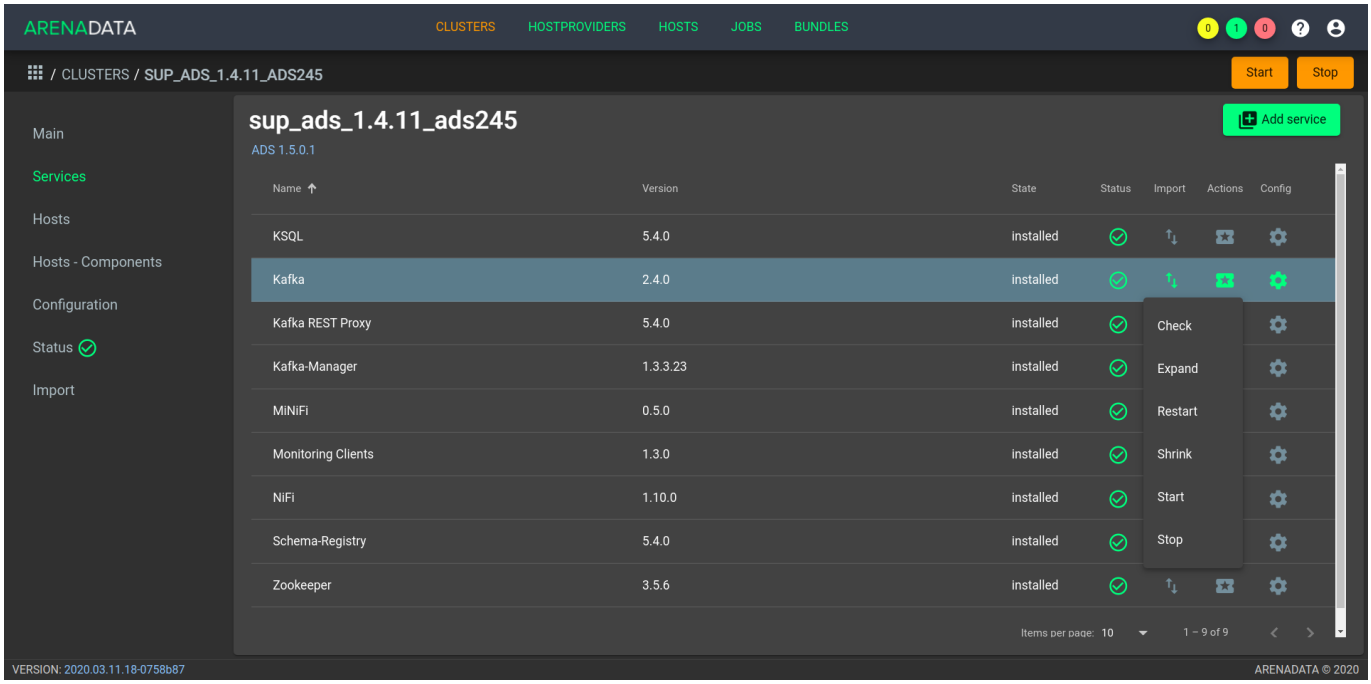

Рис.6.1.: Список допустимых операций над сервисом Kafka

- *Set vm.swappiness to 0 for all hosts* – отключение *swapping* на добавлемых хостах;
- *Append hosts into /etc/hosts file before cluster installation* – запись добавляемых нод в */etc/hosts* на всех хостах кластера. Данную опцию рекомендуется отключить, если настроен DNS.

После выбора опций для перехода к следующей странице конфигурации следует нажать кнопку “Next”, и в открывшейся форме необходимо распределить компонент *Kafka Broker* по добавляемым хостам (Рис.6.3). В случае если используется сервис *Monitoring Clinets*, его компоненты также необходимо разместить на добавляемых хостах.

Расширение сервиса запускается кнопкой “Run”. На добавленные хосты устанавливаются необходимые пакеты и производится их настройка.

Important: Расширение сервиса *Kafka* не приводит к перемещению существующих партиций топиков на новый хост. Если требуется перемещение партиций на только что добавленные в *Kafka* хосты, рекомендуется использовать *Kafka-Manager*

6.6.2 Удаление Kafka Broker

Для удаления одного или нескольких *Kafka Broker* с хостов кластера необходимо:

1. Выбрать действие *Shrink* сервиса *Kafka* из списка возможных операций (см. Рис.6.1), что приводит к появлению окна распределения компонента по хостам (см. Рис.6.3);
2. Любым из двух способов удалить привязку компонента к хосту (компонент *Kafka Broker* выделяется белым цветом, как возможный к удалению с хостов):

- Выбрать компонент в колонке “Components” и убрать выделение с хостов в колонке “Hosts”, рамки которых выделены зеленым цветом;

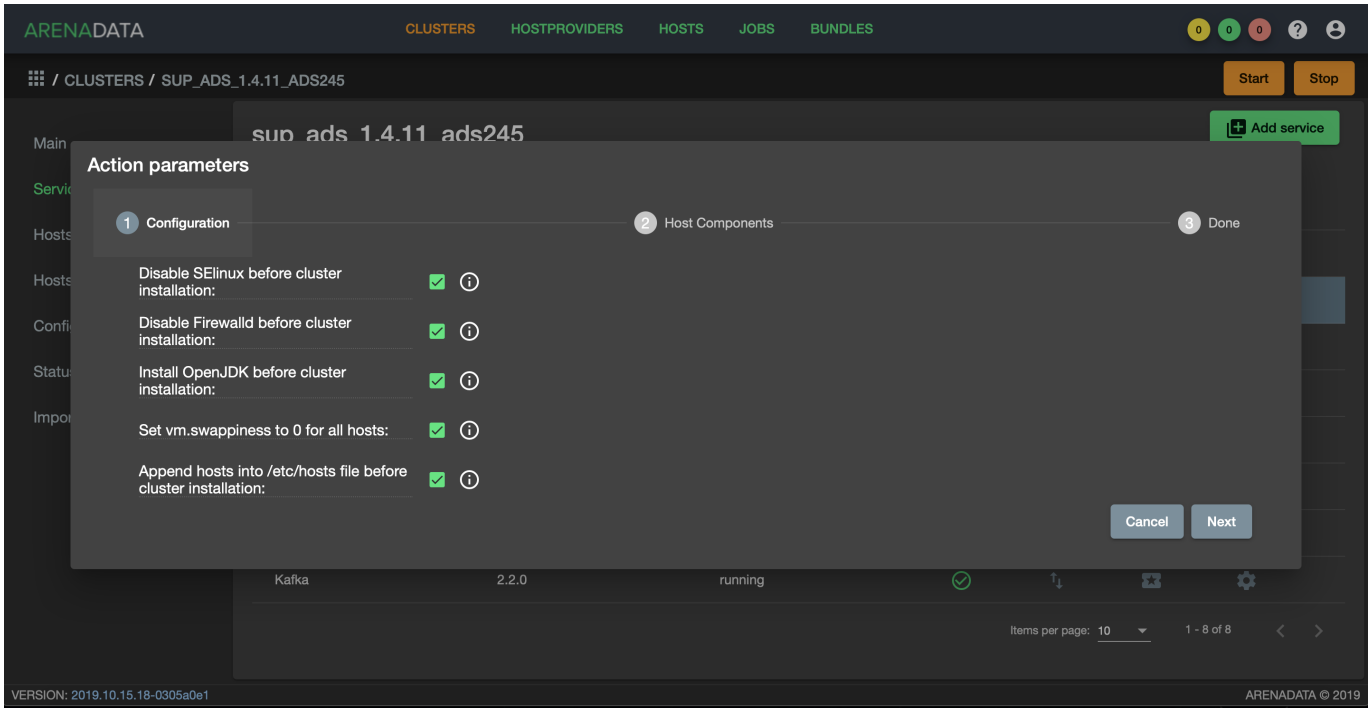

Рис.6.2.: Доступные при расширении настройки

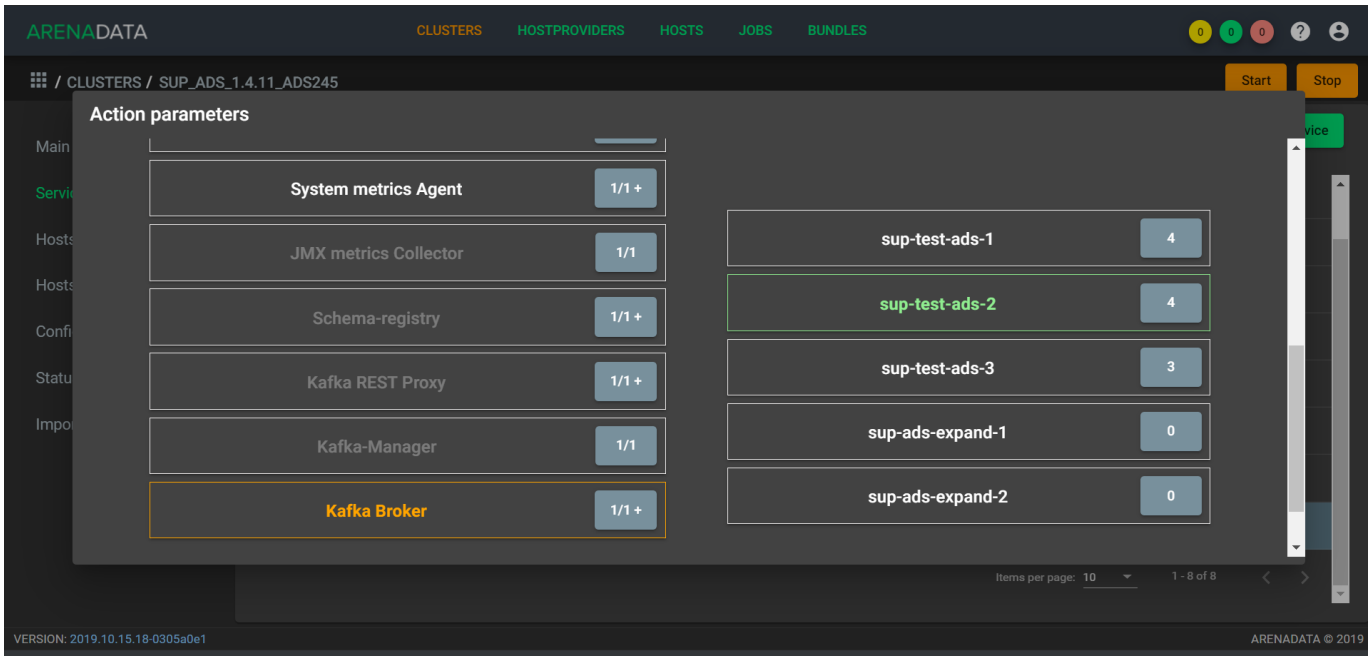

Рис.6.3.: Распределение компонента по хостам

- Выбрать хост в колонке “Hosts” и убрать выделение с компонента *Kafka Broker* в колонке “Components”, если рамка компонета *Kafka Broker* выделяется зеленым цветом.

3. Нажать кнопку “Run” в нижней части окна.

Important: Описанная процедура не удаляет данные и пакет *Kafka* с хоста – она лишь выводит ноду из кластера *Kafka*

6.7 Удаление/Добавление компонентов сервиса Zookeeper

Доступно с версии 1.4.11

Если кластер **ADS** разворачивается с помощью **ADCM**, то операции по добавлению/удалению хоста в сервис *Zookeeper* могут быть выполнены автоматически. После выполнения планирования нового аппаратного обеспечения необходимо добавить новые хосты в выбранный кластер в интерфейсе **ADCM**, используя кнопку “Add hosts” на вкладке “Hosts”. Кроме того, необходимо выполнить инициализацию каждого хоста, если того требует провайдер хостов.

Important: Описанные ниже операции не удаляют/добавляют хост из кластера – они лишь управляют компонентом *Zookeeper Server* на хостах. Удаление хоста из кластера возможно в разделе “Hosts” кластера при условии, что к хосту не привязан ни один компонент

Для добавления или удаления *Zookeeper Server* с хостов необходимо воспользоваться соответствующими кнопками выпадающего меню, доступного по нажатию на иконку в поле “Actions” сервиса *Zookeeper* (Рис.6.4).

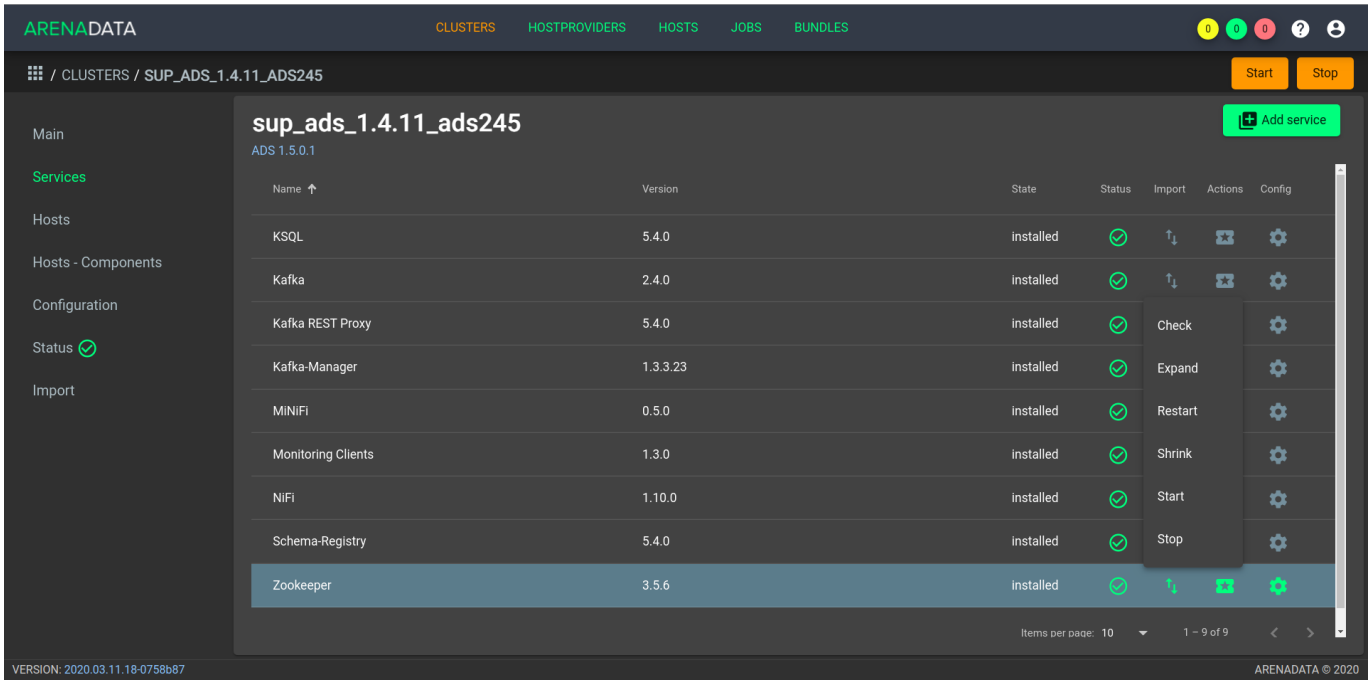

Рис.6.4.: Список допустимых операций над сервисом Zookeeper

Important: Рекомендуется использовать нечетное количество компонента *Zookeeper Server*

6.7.1 Добавление компонентов Zookeeper Server

Когда хосты становятся доступными для подключения по ssh для менеджера кластеров, необходимо выбрать действие *Expand* сервиса *Zookeeper* из списка возможных операций. В появившемся диалоговом окне предоставляется выбор опций (Рис.6.5):

- *Disable SELinux before cluster installation* – отключение SELinux на добавляемых хостах. Для того, чтобы данная настройка применилась, после завершения операции *Expand* необходимо перезагрузить хосты вручную;
- *Disable Firewalld before cluster installation* – выключение firewalld на добавляемых хостах;
- *Install OpenJDK before cluster installation* – установка пакета *java-1.8.0-openjdk* на добавляемых хостах;
- *Set vm.swappiness to 0 for all hosts* – отключение *swapping* на добавлемых хостах;
- *Append hosts into /etc/hosts file before cluster installation* – запись добавляемых нод в */etc/hosts* на всех хостах кластера. Данную опцию рекомендуется отключить, если настроен DNS.

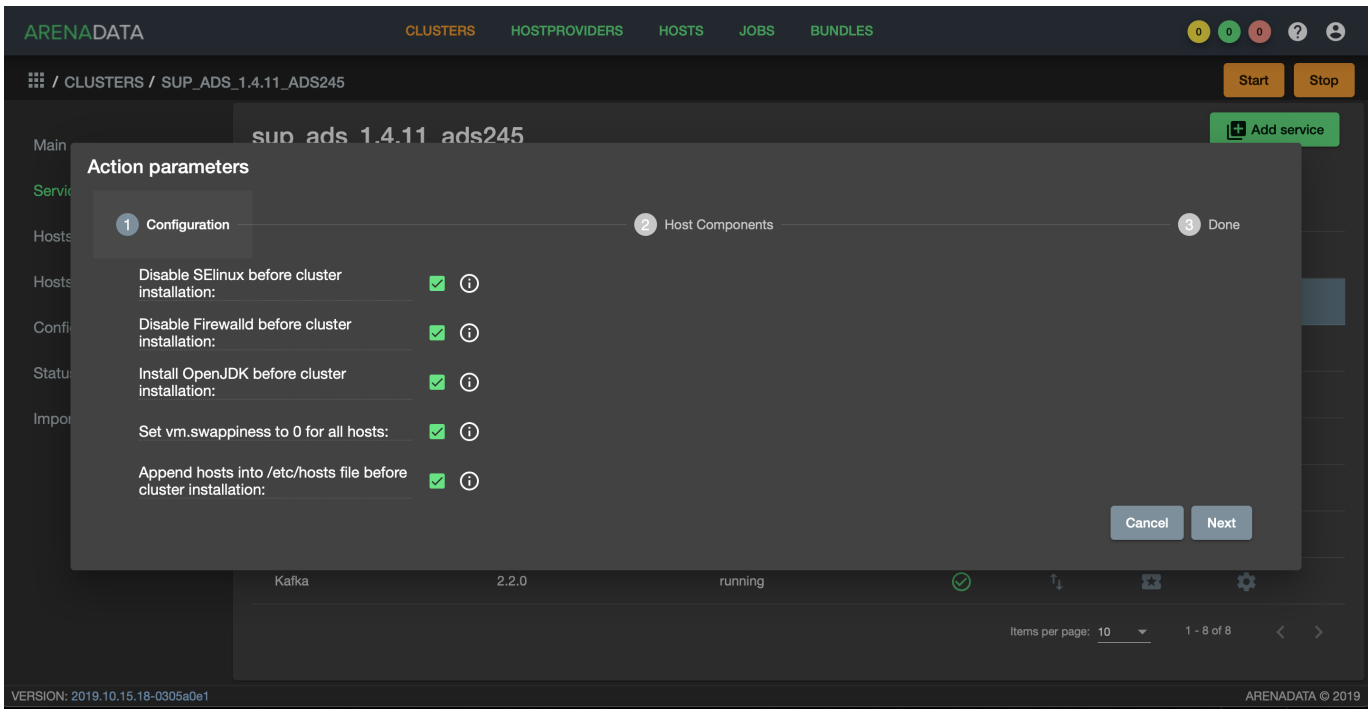

Рис.6.5.: Доступные при расширении настройки

После выбора опций для перехода к следующей странице конфигурации следует нажать кнопку “Next”, и в открывшейся форме необходимо распределить компонент *Zookeeper Server* по добавляемым хостам (Рис.6.6).

Расширение сервиса запускается кнопкой “Run”. На добавленные хосты устанавливаются необходимые пакеты и производится их настройка.

6.7.2 Удаление Zookeeper Server

Для удаления одного или нескольких *Zookeeper Server* с хостов кластера необходимо:

1. Выбрать действие *Shrink* сервиса *Zookeeper* из списка возможных операций (см. Рис.6.4), что приводит к появлению окна распределения компонента по хостам (см. Рис.6.6);
2. Любым из двух способов удалить привязку компонента к хосту (компонент *Zookeeper Server* выделяется белым цветом, как возможный к удалению с хостов):

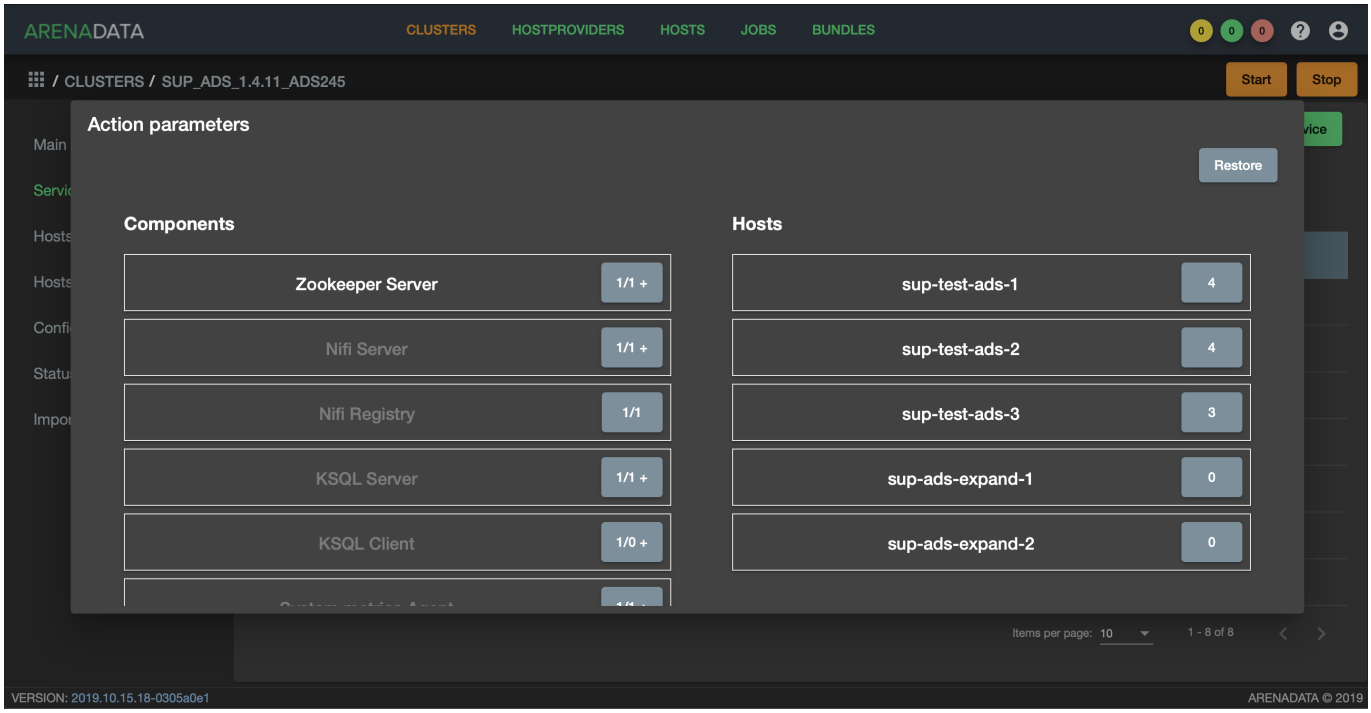

Рис.6.6.: Распределение компонента по хостам

- Выбрать компонент в колонке “Components” и убрать выделение с хостов в колонке “Hosts”, рамки которых выделены зеленым цветом;
- Выбрать хост в колонке “Hosts” и убрать выделение с компонента *Zookeeper Server* в колонке “Components”, если рамка компонета *Zookeeper Server* выделяется зеленым цветом.

3. Нажать кнопку “Run” в нижней части окна.

Important: Описанная процедура не удаляет данные и пакет *Zookeeper* с хоста – она лишь выводит ноду из кластера *Zookeeper*