**ИНСТРУКЦИЯ АДМИНИСТРАТОРА**

**RT.STREAMING**

2021

ОГЛАВЛЕНИЕ

[1 АДМИНИСТРИРОВАНИЕ KAFKA 6](#_Toc81489999)

[1.1 API 6](#_Toc81490000)

[1.1.1 Producer API 6](#_Toc81490001)

[1.1.2 Consumer API 6](#_Toc81490002)

[1.1.3 Streams API 7](#_Toc81490003)

[1.1.4 Connect API 7](#_Toc81490004)

[1.1.5 Admin API 7](#_Toc81490005)

[1.2 Конфигурации брокера 7](#_Toc81490006)

[1.2.1 Обновление конфигураций брокера 63](#_Toc81490007)

[1.3 Конфигурации топика 68](#_Toc81490008)

[1.4 Конфигурации поставщика 79](#_Toc81490009)

[1.5 Конфигурации потребителя 100](#_Toc81490010)

[1.6 Конфигурации подключения к Kafka 122](#_Toc81490011)

[1.6.1 Конфигурации коннектора источника 146](#_Toc81490012)

[1.6.2 Конфигурации sink-коннектора 150](#_Toc81490013)

[1.7 Конфигурации стримов 155](#_Toc81490014)

[1.8 Конфигурации администратора 166](#_Toc81490015)

[1.9 Основные операции Kafka 179](#_Toc81490016)

[1.9.1 Добавление и удаление топиков 179](#_Toc81490017)

[1.9.2 Изменение топиков 180](#_Toc81490018)

[1.9.3 Плавное выключение 180](#_Toc81490019)

[1.9.4 Балансировка лидерства 181](#_Toc81490020)

[1.9.5 Балансировка реплик в стойках 181](#_Toc81490021)

[1.9.6 Проверка позиции потребителя 182](#_Toc81490022)

[1.9.7 Управление группами потребителей 183](#_Toc81490023)

[1.9.8 Расширение кластера 186](#_Toc81490024)

[1.9.9 Вывод брокеров из эксплуатации 190](#_Toc81490025)

[1.9.10 Увеличение фактора репликации 190](#_Toc81490026)

[1.9.11 Дросселирование полосы пропускания во время миграции данных 191](#_Toc81490027)

[1.9.12 Установка квот 194](#_Toc81490028)

[2 АДМИНИСТРИРОВАНИЕ NIFI 197](#_Toc81490029)

[2.1 Конфигурация порта 197](#_Toc81490030)

[2.1.1 NiFi 197](#_Toc81490031)

[2.1.2 Встроенный ZooKeeper 197](#_Toc81490032)

[2.2 Конфигурация безопасности 198](#_Toc81490033)

[2.2.1 Набор инструментов для генерации TLS 200](#_Toc81490034)

[2.3 Аутентификация пользователя 205](#_Toc81490035)

[2.3.1 Single User 206](#_Toc81490036)

[2.3.2 LDAP 207](#_Toc81490037)

[2.3.3 Kerberos 210](#_Toc81490038)

[2.3.4 OpenId Connect 210](#_Toc81490039)

[2.3.5 SAML 211](#_Toc81490040)

[2.3.6 Apache Knox 213](#_Toc81490041)

[2.4 Системные свойства 214](#_Toc81490042)

[2.4.1 Рекомендации по обновлению 214](#_Toc81490043)

[2.4.2 Основные свойства 214](#_Toc81490044)

[2.4.3 Управление состояниями 218](#_Toc81490045)

[2.4.4 Конфигурация H2 219](#_Toc81490046)

[2.4.5 Репозиторий FlowFile 219](#_Toc81490047)

[2.4.6 Управление swap 227](#_Toc81490048)

[2.4.7 Репозиторий Content 227](#_Toc81490049)

[2.4.8 Репозиторий Provenance 231](#_Toc81490050)

[2.4.9 Репозиторий Status History 240](#_Toc81490051)

[2.4.10 Свойства Site to Site 242](#_Toc81490052)

[2.4.11 Свойства Web 245](#_Toc81490053)

[2.4.12 Свойства Security 248](#_Toc81490054)

[2.4.13 Свойства Identity Mapping 250](#_Toc81490055)

[2.4.14 Общие свойства кластера 251](#_Toc81490056)

[2.4.15 Свойства нод кластера 251](#_Toc81490057)

[2.4.16 Свойства ZooKeeper 254](#_Toc81490058)

[2.4.17 Свойства Kerberos 255](#_Toc81490059)

[2.4.18 Свойства Analytics 256](#_Toc81490060)

[2.4.19 Свойства Runtime Monitoring 257](#_Toc81490061)

[2.4.20 Свойства Custom 257](#_Toc81490062)

[2.5 Расположение процессоров 258](#_Toc81490063)

[2.5.1 Доступные параметры конфигурации 258](#_Toc81490064)

[2.5.2 Установка кастомных процессоров 258](#_Toc81490065)

[2.5.3 Автозагрузка кастомных процессоров 259](#_Toc81490066)

[2.5.4 Провайдеры NAR 259](#_Toc81490067)

[3 АДМИНИСТРИРОВАНИЕ AIRFLOW 261](#_Toc81490068)

[3.1 Справочник по конфигурации 261](#_Toc81490069)

[3.1.1 Раздел [core] 261](#_Toc81490070)

[3.1.2 Раздел [logging] 269](#_Toc81490071)

[3.1.3 Раздел [metrics] 272](#_Toc81490072)

[3.1.4 Раздел [secrets] 273](#_Toc81490073)

[3.1.5 Раздел [cli] 274](#_Toc81490074)

[3.1.6 Раздел [debug] 274](#_Toc81490075)

[3.1.7 Раздел [api] 275](#_Toc81490076)

[3.1.8 Раздел [lineage] 276](#_Toc81490077)

[3.1.9 Раздел [atlas] 276](#_Toc81490078)

[3.1.10 Раздел [operators] 277](#_Toc81490079)

[3.1.11 Раздел [hive] 278](#_Toc81490080)

[3.1.12 Раздел [webserver] 278](#_Toc81490081)

[3.1.13 Раздел [email] 283](#_Toc81490082)

[3.1.14 Раздел [smtp] 284](#_Toc81490083)

[3.1.15 Раздел [sentry] 285](#_Toc81490084)

[3.1.16 Раздел [celery\_kubernetes\_executor] 286](#_Toc81490085)

[3.1.17 Раздел [celery] 286](#_Toc81490086)

[3.1.18 Раздел [celery\_broker\_transport\_options] 291](#_Toc81490087)

[3.1.19 Раздел [dask] 292](#_Toc81490088)

[3.1.20 Раздел [scheduler] 292](#_Toc81490089)

[3.1.21 Раздел [kerberos] 296](#_Toc81490090)

[3.1.22 Раздел [github\_enterprise] 297](#_Toc81490091)

[3.1.23 Раздел [elasticsearch] 297](#_Toc81490092)

[3.1.24 Раздел [elasticsearch\_configs] 298](#_Toc81490093)

[3.1.25 Раздел [kubernetes] 299](#_Toc81490094)

[3.1.26 Раздел [smart\_sensor] 302](#_Toc81490095)

АДМИНИСТРИРОВАНИЕ KAFKA

API

Kafka включает пять основных API:

**Producer API** — позволяет приложениям отправлять потоки данных по топикам в кластер Kafka.

**Consumer API** — позволяет приложениям читать потоки данных из топиков кластера Kafka.

**Streams API** — позволяет преобразовывать потоки данных из входных топиков в выходные топики.

**Connect API** — позволяет реализовать коннекторы, которые непрерывно извлекают из некоторой исходной системы или приложения в Kafka или отправляют из Kafka в какую-либо систему или приложение-приёмник.

**Admin API** — позволяет управлять топиками, брокерами и другими объектами Kafka и просматривать их.

Kafka предоставляет все свои функции через независимый от языка протокол, у которого есть клиенты, доступные на многих языках программирования. Однако только клиенты Java поддерживаются как часть основного проекта Kafka, остальные доступны как независимые проекты с открытым исходным кодом.

Producer API

Producer API позволяет приложениям отправлять потоки данных по топикам в кластер Kafka.

Чтобы использовать producer (поставщик), вы можете использовать следующую зависимость maven:

|  |
| --- |
| <dependency>  <groupId>org.apache.kafka</groupId>  <artifactId>kafka-clients</artifactId>  <version>2.8.0</version>  </dependency> |

Consumer API

Consumer API позволяет приложениям читать потоки данных из топиков кластера Kafka.

Чтобы использовать потребителя (consumer), вы можете использовать следующую зависимость maven:

|  |
| --- |
| <dependency>  <groupId>org.apache.kafka</groupId>  <artifactId>kafka-clients</artifactId>  <version>2.8.0</version>  </dependency> |

Streams API

Streams API позволяет преобразовывать потоки данных из входных топиков в выходные разделы.

Чтобы использовать Kafka Streams, вы можете использовать следующую зависимость maven:

|  |
| --- |
| <dependency>  <groupId>org.apache.kafka</groupId>  <artifactId>kafka-streams</artifactId>  <version>2.8.0</version>  </dependency> |

Connect API

Connect API позволяет реализовать коннекторы, которые постоянно имплементируют из некоторой системы-источника данных в Kafka или отправляют из Kafka в какую-либо систему-приёмник данных.

Для многих пользователей Connect нет необходимости использовать данный API напрямую, однако они могут использовать готовые коннекторы без необходимости писать какой-либо код.

Admin API

Admin API поддерживает управление и проверку топиков, брокеров, списков контроля доступа и других объектов Kafka.

Чтобы использовать Admin API, добавьте следующую зависимость Maven:

|  |
| --- |
| <dependency>  <groupId>org.apache.kafka</groupId>  <artifactId>kafka-clients</artifactId>  <version>2.8.0</version>  </dependency> |

Конфигурации брокера

Основные конфигурации следующие:

1. broker.id.

log.dirs.

zookeeper.connect.

Конфигурации и их значения по умолчанию представлены в таблице ниже в алфавитном порядке по наименованию конфигурации.

Для более подробной информации о настройке брокера ищите в классе scala kafka.server.KafkaConfig.

Таблица — Конфигурации брокера

| Конфигурация | Тип | Значение по умолчанию | Действительные значения | Важность | Режим обновления | Описание |
| --- | --- | --- | --- | --- | --- | --- |
| advertised.host.name | string | null | — | high | read-only | Используется только тогда, когда параметры advertised.listeners или listeners не установлены. Рекомендуется использовать advertised.listeners.  Имя хоста для публикации в ZooKeeper для использования клиентами. В средах IaaS он может отличаться от интерфейса, к которому привязан брокер. Если параметр не установлен, будет использоваться значение параметра host.name, если он установлен. В противном случае будет использоваться значение, возвращаемое от java.net.InetAddress.getCanonicalHostName(). |
| advertised.listeners | string | null | — | high | per-broker | Слушатели для публикации в ZooKeeper для использования клиентами, если они отличаются от свойства конфигурации listeners. В средах IaaS он может потребовать отличаться от интерфейса, к которому привязан брокер. Если он не установлен, будет использоваться значение для listeners. В отличие от listeners, анонсировать мета-адрес 0.0.0.0 недопустимо.  Также, в отличие от listeners, в этом свойстве могут быть дублированные порты, так что один слушатель может быть настроен для объявления адреса другого слушателя. Это может быть полезно в некоторых случаях, когда используются внешние балансировщики нагрузки. |
| advertised.port | int | null | — | high | read-only | Используется только тогда, когда advertised.listeners или listeners не установлены. Рекомендуется использовать advertised.listeners.  Порт для публикации в ZooKeeper для использования клиентами. В средах IaaS этот порт может отличаться от порта, к которому привязан брокер. Если он не установлен, он опубликует тот же порт, к которому привязан брокер. |
| alter.config.policy.class.name | class | null | — | low | read-only | Класс политики изменяемых конфигураций, который следует использовать для их валидации. Класс должен имплементировать интерфейс org.apache.kafka.server.policy.AlterConfigPolicy. |
| alter.log.dirs.replication.quota.window.num | int | 11 | [1,…] | low | read-only | Количество выборок, сохраняемых в памяти для квот репликации изменяемых журналов. |
| alter.log.dirs.replication.quota.window.size.seconds | int | 1 | [1,…] | low | read-only | Временной интервал каждой выборки для квот репликации изменяемых журналов. |
| authorizer.class.name | string | "" | — | low | read-only | Полное имя класса, имплементирующего интерфейс sorg.apache.kafka.server.authorizer.Authorizer, который используется брокером для авторизации. Данная конфигурация также поддерживает авторизаторы, которые имплементируют устаревшую черту kafka.security.auth.Authorizer, которая ранее использовалась для авторизации. |
| auto.create.topics.enable | boolean | true | — | high | read-only | Включает автоматическое создание топика на сервере. |
| auto.leader.rebalance.enable | boolean | true | — | high | read-only | Включает автоматическую балансировку лидера. Фоновый поток проверяет распределение лидеров партиций через регулярные промежутки времени, настраиваемые с помощью leader.imbalance.check.interval.seconds. Если дисбаланс лидера превышает leader.imbalance.per.broker.percentage, срабатывает перебалансирование лидера на предпочтительного лидера для партиций, по которым сработал триггер. |
| background.threads | int | 10 | [1,…] | high | cluster-wide | Количество потоков, используемых для различных задач фоновой обработки. |
| broker.id | int | -1 | — | high | read-only | Идентификатор брокера для данного сервера. Если значение не задано, будет сгенерирован уникальный идентификатор брокера. Чтобы избежать конфликтов между идентификаторами брокера, сгенерированными zookeeper, и идентификаторами брокера, настроенными пользователем, сгенерированные идентификаторы брокера начинаются с reserved.broker.max.id + 1. |
| broker.id.generation.enable | boolean | true | — | medium | read-only | Включение автоматического создания идентификатора брокера на сервере. Если этот параметр включён, необходимо проверить значение, настроенное для reserved.broker.max.id. |
| broker.rack | string | null | — | medium | read-only | Стойка брокера. Значение будет использоваться при назначении репликации с учётом стойки для обеспечения отказоустойчивости. Примеры: RACK1, us-east-1d. |
| client.quota.callback.class | class | null | — | low | read-only | Полное имя класса, имплементирующего интерфейс ClientQuotaCallback, который используется для определения лимитов квот, применяемых к клиентским запросам. Для любого заданного запроса применяется наиболее конкретная квота, которая соответствует принципалу пользователя сеанса и идентификатору клиента запроса. |
| compression.type | string | producer | — | high | cluster-wide | Указывает окончательный тип сжатия для указанного топика. Эта конфигурация принимает стандартные кодеки сжатия (gzip, snappy, lz4, zstd). Кроме того, он принимает uncompressed, что эквивалентно отсутствию сжатия, и producer, что означает сохранение исходного кодека сжатия, установленного поставщиком. |
| connection.failed.authentication.delay.ms | int | 100 | [0,…] | low | read-only | Задержка закрытия соединения при неудачной аутентификации. Это время (в миллисекундах), на которое будет отложено закрытие соединения при неудачной аутентификации. Значение должно быть меньше, чем connections.max.idle.ms, чтобы предотвратить тайм-аут соединения. |
| connections.max.idle.ms | long | 600000 (10 минут) | — | medium | read-only | Тайм-аут незанятых соединений: потоки процессора сокета сервера закрывают соединения, которые простаивают больше установленного значения. |
| connections.max.reauth.ms | long | 0 | — | medium | read-only | Если явно задано положительное число (по умолчанию 0, а не положительное число), время жизни сеанса, которое не превысит настроенное значение, будет сообщаться клиентам v2.2.0 или более поздней версии при аутентификации. Брокер отключит любое такое соединение, которое не было повторно аутентифицировано в течение срока действия сеанса и которое впоследствии используется для любых целей, кроме повторной аутентификации. Имена конфигураций могут быть дополнены префиксом слушателя и именем механизма SASL в нижнем регистре. Например, listener.name.sasl\_ssl.oauthbearer.connections.max.reauth.ms = 3600000. |
| controlled.shutdown.enable | boolean | true | — | medium | read-only | Включение контролируемого выключения сервера. |
| controlled.shutdown.max.retries | int | 3 | — | medium | read-only | Контролируемое выключение может завершиться ошибкой по некоторым причинам. Значение определяет количество повторных попыток, когда происходит такой сбой. |
| controlled.shutdown.retry.backoff.ms | long | 5000 (5 секунд) | — | medium | read-only | Перед каждой повторной попыткой системе необходимо время для восстановления из состояния, которое вызвало предыдущий сбой (переключение контроллера, задержка реплики и т.д.). Данная конфигурация определяет время ожидания перед повторной попыткой. |
| controller.socket.timeout.ms | int | 30000 (30 секунд) | — | medium | read-only | Тайм-аут сокета для каналов от контроллера к брокеру. |
| controller.quota.window.num | int | 11 | [1,…] | low | read-only | Количество выборок, сохраняемых в памяти для квот мутации контроллера. |
| controller.quota.window.size.seconds | int | 1 | [1,…] | low | read-only | Временной интервал каждой выборки для квот мутаций контроллера. |
| control.plane.listener.name | string | null | — | high | read-only | Имя слушателя, используемого для взаимодействия между контроллером и брокерами. Брокер будет использовать control.plane.listener.name, чтобы найти конечную точку в списке слушателей и прослушивать соединения от контроллера. Например, если конфигурация брокера:  listeners = INTERNAL://192.1.1.8:9092, EXTERNAL://10.1.1.5:9093, CONTROLLER://192.1.1.8:9094  listener.security.protocol.map = INTERNAL:PLAINTEXT, EXTERNAL:SSL, CONTROLLER:SSL  control.plane.listener.name = CONTROLLER  при запуске брокер начнет прослушивать «192.1.1.8:9094» по протоколу безопасности «SSL».  На стороне контроллера, когда он обнаруживает опубликованные конечные точки брокера через zookeeper, он будет использовать control.plane.listener.name для поиска конечной точки, которую он будет использовать для установления соединения с брокером.  Например, если опубликованные брокером конечные точки на zookeeper:  "endpoints" : ["INTERNAL://broker1.example.com:9092","EXTERNAL://broker1.example.com:9093","CONTROLLER://broker1.example.com:9094"]  а конфигурация контроллера:  listener.security.protocol.map = INTERNAL:PLAINTEXT, EXTERNAL:SSL, CONTROLLER:SSL  control.plane.listener.name = CONTROLLER  тогда контроллер будет использовать «broker1.example.com:9094» с протоколом безопасности «SSL» для подключения к брокеру.  Если явно не настроено, значение по умолчанию будет нулевым, и не будет выделенных конечных точек для подключений контроллера. |
| create.topic.policy.class.name | class | null | — | low | read-only | Класс политики создания топика, который следует использовать для валидации. Класс должен имплементировать интерфейс org.apache.kafka.server.policy.CreateTopicPolicy. |
| default.replication.factor | int | 1 | — | medium | read-only | Коэффициенты репликации по умолчанию для автоматически созданных топиков. |
| delegation.token.expiry.check.interval.ms | long | 3600000 (1 час) | [1,…] | low | read-only | Интервал сканирования для удаления истекших токенов делегирования. |
| delegation.token.expiry.time.ms | long | 86400000 (1 день) | [1,…] | medium | read-only | Срок действия токена в мс до того, как токен потребуется продлить. Значение по умолчанию 1 день. |
| delegation.token.master.key | password | null | — | medium | read-only | Алиас delegation.token.secret.key, который следует использовать вместо данной конфигурации. |
| delegation.token.max.lifetime.ms | long | 604800000 (7 дней) | [1,…] | medium | read-only | Максимальное время жизни токена, после которого его нельзя больше продлить. Значение по умолчанию 7 дней. |
| delegation.token.secret.key | password | null | — | medium | read-only | Секретный ключ для генерации и проверки токенов делегирования. Один и тот же ключ должен быть настроен для всех брокеров. Если ключ не установлен или установлен как пустая строка, брокеры отключат поддержку токена делегирования. |
| delete.records.purgatory.purge.interval.requests | int | 1 | — | medium | read-only | Интервал очистки (в количестве запросов) записей на удаление. |
| delete.topic.enable | boolean | true | — | high | read-only | Включает возможность удаления топика. Удаление топика через инструмент администратора не будет иметь никакого эффекта, если эта конфигурация отключена. |
| fetch.max.bytes | int | 57671680 (55 МиБ) | [1024,…] | medium | read-only | Максимальное количество байт, которое возвращаются для запроса на выборку. Значение должно быть не менее 1024. |
| fetch.purgatory.purge.interval.requests | int | 1000 | — | medium | read-only | Интервал очистки (в количестве запросов) запросов на выборку. |
| group.initial.rebalance.delay.ms | int | 3000 (3 секунды) | — | medium | read-only | Время, в течение которого координатор группы будет ждать, пока больше потребителей присоединятся к новой группе, перед выполнением первой перебалансировки. Более длительная задержка означает потенциально меньшее количество перебалансировок, но увеличивает время до начала обработки. |
| group.max.session.timeout.ms | int | 1800000 (30 минут) | — | medium | read-only | Максимально допустимое время ожидания сеанса для зарегистрированных потребителей. Более длительные тайм-ауты дают потребителям больше времени для обработки сообщений между контрольными сигналами за счёт более длительного времени на обнаружение сбоев. |
| group.max.size | int | 2147483647 | [1,…] | medium | read-only | Максимальное количество потребителей, которое может вместить одна группа потребителей. |
| group.min.session.timeout.ms | int | 6000 (6 секунд) | — | medium | read-only | Минимально допустимое время ожидания сеанса для зарегистрированных потребителей. Более короткие тайм-ауты приводят к более быстрому обнаружению сбоев за счёт более частых heartbeat-сообщений потребителей, которые могут перегрузить ресурсы брокера. |
| host.name | string | "" | — | high | read-only | Используется только тогда, когда listeners не установлены. Вместо этого параметра рекомендуется использовать listeners.  Имя хоста брокера. Если он установлен, он будет привязан только к этому адресу. Если этот параметр не установлен, он будет привязан ко всем интерфейсам. |
| inter.broker.listener.name | string | null | — | medium | read-only | Имя слушателя, используемого для связи между брокерами. Если значение не установлено, имя слушателя определяется свойством security.inter.broker.protocol. Нельзя устанавливать одновременно данное свойство и свойство security.inter.broker.protocol. |
| inter.broker.protocol.version | string | 2.8-IV1 | [0.8.0, 0.8.1, 0.8.2, 0.9.0, 0.10.0-IV0, 0.10.0-IV1, 0.10.1-IV0, 0.10.1-IV1, 0.10.1-IV2, 0.10.2-IV0, 0.11.0-IV0, 0.11.0-IV1, 0.11.0-IV2, 1.0-IV0, 1.1-IV0, 2.0-IV0, 2.0-IV1, 2.1-IV0, 2.1-IV1, 2.1-IV2, 2.2-IV0, 2.2-IV1, 2.3-IV0, 2.3-IV1, 2.4-IV0, 2.4-IV1, 2.5-IV0, 2.6-IV0, 2.7-IV0, 2.7-IV1, 2.7-IV2, 2.8-IV0, 2.8-IV1] | medium | read-only | Указывает, какая версия межброкерского протокола будет использоваться.  Обычно значение устанавливают после того, как все брокеры были обновлены до новой версии.  Примеры 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. Полный список представлен в поле Действительные значения. |
| kafka.metrics.polling.interval.secs | int | 10 | [1,…] | low | read-only | Интервал опроса метрик (в секундах), который можно использовать в имплементациях kafka.metrics.reporters. |
| kafka.metrics.reporters | list | "" | — | low | read-only | Перечень классов для использования в качестве кастомных репортеров метрик Yammer. Репортеры должны имплементировать трейт kafka.metrics.KafkaMetricsReporter. Если клиент хочет раскрыть операции JMX в кастомном репортере, кастомный репортер должен дополнительно имплементировать трейт MBean, который расширяет трейт kafka.metrics.KafkaMetricsReporterMBean, чтобы зарегистрированный MBean соответствовал стандартному соглашению MBean. |
| leader.imbalance.check.interval.seconds | long | 300 | — | high | read-only | Частота, с которой контроллер запускает проверку, чтобы перебалансировать партиции. |
| leader.imbalance.per.broker.percentage | int | 10 | — | high | read-only | Коэффициент дисбаланса лидеров, допустимый на каждого брокера. Контроллер запустит баланс лидера, если он превысит это значение для каждого брокера. Значение указывается в процентах. |
| listener.security.protocol.map | string | PLAINTEXT:PLAINTEXT,SSL:SSL,SASL\_PLAINTEXT:SASL\_PLAINTEXT,SASL\_SSL:SASL\_SSL | — | low | per-broker | Маппинг имён слушателей и протоколов безопасности. Свойство должно быть определено для того, чтобы один и тот же протокол безопасности мог использоваться более чем на одном порте или IP-адресе. Например, внутренний и внешний трафик можно разделить, даже если для обоих требуется SSL. Конкретно, пользователь может определять слушателей с именами INTERNAL и EXTERNAL и данным свойством как INTERNAL:SSL,EXTERNAL:SSL. Как показано, ключ и значение разделяются двоеточием, а записи маппинга разделяются запятыми. Имя каждого слушателя должно появляться в маппинге только один раз. Для каждого слушателя можно настроить различные параметры безопасности (SSL и SASL), добавив нормализованный префикс (имя слушателя в нижнем регистре) к имени конфигурации. Например, чтобы установить другое хранилище ключей для слушателя INTERNAL, должна быть установлена конфигурация с именем listener.name.internal.ssl.keystore.location. Если конфигурация для имени слушателя не установлена, конфигурация вернётся к общей конфигурации (например, ssl.keystore.location). |
| listeners | string | null | — | high | read-only | Список слушателей — разделённый запятыми список URI, которые мы будем слушать, и имена слушателей. Если имя слушателя не является протоколом безопасности, также необходимо установить listener.security.protocol.map.  Имена слушателей и номера портов должны быть уникальными.  Укажите имя хоста как 0.0.0.0 для привязки ко всем интерфейсам.  Оставьте имя хоста пустым, чтобы привязать его к интерфейсу по умолчанию.  Примеры легальных списков слушателей:  PLAINTEXT://myhost:9092,SSL://:9091  CLIENT://0.0.0.0:9092,REPLICATION://localhost:9093 |
| log.cleaner.backoff.ms | long | 15000 (15 секунд) | [0,…] | medium | cluster-wide | Продолжительность спящего режима, когда нет логов для очистки. |
| log.cleaner.dedupe.buffer.size | long | 134217728 | — | medium | cluster-wide | Общий объём памяти, используемый для дедупликации лога во всех потоках очистки. |
| log.cleaner.delete.retention.ms | long | 86400000 (1 день) | — | medium | cluster-wide | Продолжительность хранения записей об удалении. |
| log.cleaner.enable | boolean | true | — | medium | read-only | Включение процесса очистки журнала для запуска на сервере. Очистка должна быть включена, если используются какие-либо топики с cleanup.policy=compact, включая топиков внутренних смещений. Если данный параметр отключён, топики не будут сжиматься и будут постоянно увеличиваться в размерах. |
| log.cleaner.io.buffer.load.factor | double | 0.9 | — | medium | cluster-wide | Коэффициент загрузки буфера дедупликации для очистки логов. Процент заполнения буфера дедупликации. Более высокое значение позволит одновременно очистить больше логов, но приведёт к большему количеству хеш-коллизий. |
| log.cleaner.io.buffer.size | int | 524288 | [0,…] | medium | cluster-wide | Общий объём памяти, используемый для буферов ввода-вывода для очистки логов во всех потоках очистки. |
| log.cleaner.io.max.bytes.per.second | double | 1.7976931348623157E308 | — | medium | cluster-wide | Очистка лога будет регулироваться, так что сумма её операций чтения и записи будет в среднем меньше установленного значения. |
| log.cleaner.max.compaction.lag.ms | long | 9223372036854775807 | — | medium | cluster-wide | Максимальное время, в течение которого сообщение не подлежит сжатию в логе. Применимо только для логов, для которых установлено сжатие. |
| log.cleaner.min.cleanable.ratio | double | 0.5 | — | medium | cluster-wide | Минимальное отношение сжатого лога к общему количеству лога для лога, подлежащего очистке. Если также указаны конфигурации log.cleaner.max.compaction.lag.ms или log.cleaner.min.compaction.lag.ms, то механизм сжатия логов считает, что лог подходит для сжатия, как только:  был достигнут установленный порог, и в логе были несжатые записи в течение, по крайней мере, времени, установленного в свойстве log.cleaner.min.compaction.lag.ms;  если в логе были несжатые записи в течение времени, установленного в свойстве log.cleaner.max.compaction.lag.ms. |
| log.cleaner.min.compaction.lag.ms | long | 0 | — | medium | cluster-wide | Минимальное время, в течение которого сообщение остаётся в логе без сжатия. Применимо только для логов, для которых установлено сжатие. |
| log.cleaner.threads | int | 1 | [0,…] | medium | cluster-wide | Количество фоновых потоков, используемых для очистки лога. |
| log.cleanup.policy | list | delete | [compact, delete] | medium | cluster-wide | Политика очистки по умолчанию для сегментов за пределами окна хранения. Список допустимых политик, разделённых запятыми. Допустимые политики: delete и compact. |
| log.dir | string | /tmp/kafka-logs | — | high | read-only | Каталог, в котором хранятся данные логов (дополнительно к свойству log.dirs). |
| log.dirs | string | null | — | high | read-only | Каталоги, в которых хранятся данные логов. Если не установлен, используется значение из log.dir. |
| log.flush.interval.messages | long | 9223372036854775807 | [1,…] | high | cluster-wide | Количество сообщений, накопленных в партиции лога до того, как сообщения будут сброшены на диск. |
| log.flush.interval.ms | long | null | — | high | cluster-wide | Максимальное время в мс, в течение которого сообщение в любом топике хранится в памяти перед сбросом на диск. Если не установлен, используется значение из log.flush.scheduler.interval.ms. |
| log.flush.offset.checkpoint.interval.ms | int | 60000 (1 минута) | [0,…] | high | read-only | Частота, с которой мы обновляем постоянную запись последнего сброса, которая действует как точка восстановления лога. |
| log.flush.scheduler.interval.ms | long | 9223372036854775807 | — | high | read-only | Частота в мс, с которой средство очистки лога проверяет, нужно ли сбрасывать лог на диск. |
| log.flush.start.offset.checkpoint.interval.ms | int | 60000 (1 минута) | [0,…] | high | read-only | Частота, с которой мы обновляем постоянную запись смещения начала лога. |
| log.index.interval.bytes | int | 4096 (4 КиБ) | [0,…] | medium | cluster-wide | Интервал, с которым добавляется запись в индекс смещения. |
| log.index.size.max.bytes | int | 10485760 (10 МиБ) | [4,…] | medium | cluster-wide | Максимальный размер в байтах индекса смещения. |
| log.message.downconversion.enable | boolean | true | — | low | cluster-wide | Данная конфигурация определяет, разрешено ли понижающее преобразование форматов сообщений для удовлетворения запросов на потребление. Если установлено значение false, брокер не будет выполнять понижающее преобразование для потребителей, ожидающих более старого формата сообщения. Брокер отвечает ошибкой UNSUPPORTED\_VERSION для запросов на потребление от таких старых клиентов. Данная конфигурация не применяется к любому преобразованию формата сообщения, которое может потребоваться для репликации подписчикам. |
| log.message.format.version | string | 2.8-IV1 | [0.8.0, 0.8.1, 0.8.2, 0.9.0, 0.10.0-IV0, 0.10.0-IV1, 0.10.1-IV0, 0.10.1-IV1, 0.10.1-IV2, 0.10.2-IV0, 0.11.0-IV0, 0.11.0-IV1, 0.11.0-IV2, 1.0-IV0, 1.1-IV0, 2.0-IV0, 2.0-IV1, 2.1-IV0, 2.1-IV1, 2.1-IV2, 2.2-IV0, 2.2-IV1, 2.3-IV0, 2.3-IV1, 2.4-IV0, 2.4-IV1, 2.5-IV0, 2.6-IV0, 2.7-IV0, 2.7-IV1, 2.7-IV2, 2.8-IV0, 2.8-IV1] | medium | read-only | Установка версии формата сообщения, которую брокер будет использовать для добавления сообщений в логи. Значение должно быть валидным ApiVersion. Вот несколько примеров: 0.8.2, 0.9.0.0, 0.10.0. Устанавливая конкретную версию формата сообщения, пользователь удостоверяет, что все существующие сообщения на диске меньше или равны указанной версии. Неправильная установка данного значения приведёт к тому, что потребители более старых версий перестанут работать, поскольку они будут получать сообщения в формате, который им не понятен. |
| log.message.timestamp.difference.max.ms | long | 9223372036854775807 | — | medium | cluster-wide | Максимально допустимая разница между отметкой времени, когда брокер получает сообщение, и отметкой времени, указанной в сообщении. Если log.message.timestamp.type=CreateTime, сообщение будет отклонено, если разница во времени превышает этот порог. Эта конфигурация игнорируется, если log.message.timestamp.type=LogAppendTime. Максимально допустимая разница в отметках времени не должна превышать log.retention.ms, чтобы избежать излишне частой прокрутки лога. |
| log.message.timestamp.type | string | CreateTime | [CreateTime, LogAppendTime] | medium | cluster-wide | Определяет, является ли метка времени в сообщении временем создания сообщения или временем добавления в лог. Значение должно быть либо CreateTime, либо LogAppendTime. |
| log.preallocate | boolean | false | — | medium | cluster-wide | Устанавливает, следует ли предварительно выделить файл при создании нового сегмента. Если вы используете Kafka в Windows, вам, вероятно, потребуется установить для него значение true. |
| log.retention.bytes | long | -1 | — | high | cluster-wide | Максимальный размер лога перед его удалением. |
| log.retention.check.interval.ms | long | 300000 (5 минут) | [1,…] | medium | read-only | Частота в мс, с которой механизм очистки логов проверяет, подходит ли какой-либо лог для удаления. |
| log.retention.hours | int | 168 | — | high | read-only | Количество часов для хранения файла лога перед его удалением (в часах), третичное значение для свойства log.retention.ms. |
| log.retention.minutes | int | null | — | high | read-only | Количество минут для хранения файла лога перед его удалением (в минутах), вторичное по отношению к свойству log.retention.ms. Если свойство не задано, используется значение из свойства log.retention.hours. |
| log.retention.ms | long | null | — | high | cluster-wide | Количество миллисекунд для хранения файла лога перед его удалением (в миллисекундах). Если свойство не установлено, используется значение из свойства log.retention.minutes. Если установлено значение -1, ограничение по времени не применяется. |
| log.roll.hours | int | 168 | [1,…] | high | read-only | Максимальное время до развёртывания нового сегмента лога (в часах), вторично по отношению к свойству log.roll.ms. |
| log.roll.jitter.hours | int | 0 | [0,…] | high | read-only | Максимальный джиттер, вычитаемый из logRollTimeMillis (в часах), вторичный по отношению к свойству log.roll.jitter.ms. |
| log.roll.jitter.ms | long | null | — | high | cluster-wide | Максимальный джиттер, вычитаемый из logRollTimeMillis (в миллисекундах). Если свойство не установлено, используется значение из log.roll.jitter.hours. |
| log.roll.ms | long | null | — | high | cluster-wide | Максимальное время до развёртывания нового сегмента лога (в миллисекундах). Если свойство не установлено, используется значение из log.roll.hours. |
| log.segment.bytes | int | 1073741824 (1 ГиБ) | [14,…] | high | cluster-wide | Максимальный размер одного файла лога. |
| log.segment.delete.delay.ms | long | 60000 (1 минута) | [0,…] | high | cluster-wide | Время ожидания перед удалением файла из файловой системы. |
| max.connection.creation.rate | int | 2147483647 | [0,…] | medium | cluster-wide | Максимальная скорость создания соединений, которая разрешена в брокере в любое время. Ограничения на уровне слушателя также могут быть настроены путём добавления к имени конфигурации префикса слушателя, например listener.name.internal.max.connection.creation.rate. Ограничение скорости соединения для всего брокера должно быть настроено на основе пропускной способности брокера в то время, как ограничения слушателя должны быть настроены в зависимости от требований приложения. Новые соединения будут регулироваться, если будет достигнут предел пслушателя или брокера, за исключением слушателя между брокерами. Соединения на слушателе между брокерами будут регулироваться только при достижении предела скорости на уровне слушателя. |
| max.connections | int | 2147483647 | [0,…] | medium | cluster-wide | Максимальное количество подключений, которые разрешены в брокере в любое время. Это ограничение применяется в дополнение к любым ограничениям для каждого IP-адреса, настроенным с помощью max.connections.per.ip. Ограничения на уровне слушателя также можно настроить, добавив к имени конфигурации префикс слушателя, например listener.name.internal.max.connections. Ограничение на уровне брокера должно быть настроено на основе возможностей брокера, а ограничения на слушателя должны быть настроены на основе требований приложения. Новые подключения блокируются, если достигнут предел слушателя или брокера. Соединения на слушателе между брокерами разрешены, даже если достигнут лимит брокера. В этом случае будет закрыто последнее использованное соединение на другом слушателе. |
| max.connections.per.ip | int | 2147483647 | [0,…] | medium | cluster-wide | Максимальное количество подключений, которые разрешены с каждого IP-адреса. Значение может быть установлено 0, если имеются переопределения, настроенные с помощью свойства max.connections.per.ip.overrides. При достижении лимита новые соединения с IP-адреса сбрасываются. |
| max.connections.per.ip.overrides | string | "" | — | medium | cluster-wide | Разделённый запятыми список для каждого IP-адреса или имени хоста переопределяет максимальное количество подключений по умолчанию. Пример значения: hostName:100,127.0.0.1:200. |
| max.incremental.fetch.session.cache.slots | int | 1000 | [0,…] | medium | read-only | Максимальное количество сеансов инкрементной выборки, которое будет поддерживаться. |
| message.max.bytes | int | 1048588 | [0,…] | high | cluster-wide | Максимальный размер пакета записей, разрешённый Kafka (после сжатия, если сжатие включено). Если размер был увеличен и есть потребители менее 0.10.2, размер выборки потребителей также должен быть увеличен, чтобы они могли получать пакеты записей установленного размера. В последней версии формата сообщений, записи всегда сгруппированы в пакеты для повышения эффективности. В предыдущих версиях формата сообщений, несжатые записи не группировались в пакеты, и в этом случае это ограничение применяется только к одной записи. Размер можно установить для каждого топика в конфигурации max.message.bytes на уровне топика. |
| metric.reporters | list | "" | — | low | cluster-wide | Перечень классов для использования в качестве репортеров метрик. Имплементация интерфейса org.apache.kafka.common.metrics.MetricsReporter позволяет подключать классы, которые будут получать уведомления о создании новой метрики. JmxReporter всегда включён для регистрации статистики JMX. |
| metrics.num.samples | int | 2 | [1,…] | low | read-only | Количество образцов, поддерживаемых для вычисления метрик. |
| metrics.recording.level | string | INFO | — | low | read-only | Самый высокий уровень записи метрик. |
| metrics.sample.window.ms | long | 30000 (30 секунд) | [1,…] | low | read-only | Временной промежуток, в течение которого вычисляется выборка метрики. |
| min.insync.replicas | int | 1 | [1,…] | high | cluster-wide | Когда поставщик устанавливает для свойства acks значение all (или -1), min.insync.replicas указывает минимальное количество реплик, которые должны подтвердить запись, чтобы запись считалась успешной. Если этот минимум не может быть соблюден, поставщик вызовет исключение (NotEnoughReplicas или NotEnoughReplicasAfterAppend).  Совместное использование min.insync.replicas и acks обеспечивает бОльшую надёжность. Типичный сценарий — создать топик с коэффициентом репликации равным 3, установить min.insync.replicas равным 2 и acks равным all. Установленные настройки гарантирует, что поставщик вызовет исключение, если большинство реплик не получит запись. |
| num.io.threads | int | 8 | [1,…] | high | cluster-wide | Количество потоков, которые сервер использует для обработки запросов, которые могут включать дисковый ввод-вывод. |
| num.network.threads | int | 3 | [1,…] | high | cluster-wide | Количество потоков, которые сервер использует для получения запросов из сети и отправки ответов в сеть. |
| num.partitions | int | 1 | [1,…] | medium | read-only | Количество партиций лога по умолчанию для каждого топика. |
| num.recovery.threads.per.data.dir | int | 1 | [1,…] | high | cluster-wide | Количество потоков в каталоге данных, которые будут использоваться для восстановления лога при запуске и очистки при завершении работы. |
| num.replica.alter.log.dirs.threads | int | null | — | high | read-only | Количество потоков, которые могут перемещать реплики между каталогами логов, включая дисковый ввод-вывод. |
| num.replica.fetchers | int | 1 | — | high | cluster-wide | Количество потоков сборщика, используемых для репликации сообщений от исходного брокера. Увеличение этого значения может увеличить степень параллелизма ввода-вывода в брокере-подписчике. |
| offset.metadata.max.bytes | int | 4096 (4 КиБ) | — | high | read-only | Максимальный размер записи метаданных, связанной с фиксацией смещения. |
| offsets.commit.required.acks | short | -1 | — | high | read-only | Необходимые подтверждения перед принятием фиксации. Как правило, значение по умолчанию (-1) не следует изменять. |
| offsets.commit.timeout.ms | int | 5000 (5 секунд) | [1,…] | high | read-only | Фиксация смещения будет отложена до тех пор, пока все реплики для темы смещения не получат фиксацию или пока не будет достигнут этот тайм-аут. Это похоже на тайм-аут запроса поставщика. |
| offsets.load.buffer.size | int | 5242880 | [1,…] | high | read-only | Размер пакета для чтения из сегментов смещений при загрузке смещений в кэш (мягкий лимит, отменяется, если записи слишком велики). |
| offsets.retention.check.interval.ms | long | 600000 (10 минут) | [1,…] | high | read-only | Частота проверки устаревших смещений. |
| offsets.retention.minutes | int | 10080 | [1,…] | high | read-only | После того, как группа потребителей потеряет всех своих потребителей (т. е. станет пустой), её смещения будут храниться в течение установленного периода хранения, прежде чем они будут отброшены. Для автономных потребителей (использующих назначение вручную) смещения будут истекать по истечении времени последней фиксации плюс установленный данным свойством период хранения. |
| offsets.topic.compression.codec | int | 0 | — | high | read-only | Кодек сжатия для топика смещений — сжатие может использоваться для достижения "атомарных" смещений. |
| offsets.topic.num.partitions | int | 50 | [1,…] | high | read-only | Количество партиций для топика смещения фиксации (значение свойства не должно меняться после развёртывания). |
| offsets.topic.replication.factor | short | 3 | [1,…] | high | read-only | Коэффициент репликации для топика смещений (установите более высокое значение, чтобы обеспечить доступность). Создание внутреннего топика не будет выполнено, пока размер кластера не будет соответствовать установленному требованию к коэффициенту репликации. |
| offsets.topic.segment.bytes | int | 104857600 (100 МиБ) | [1,…] | high | read-only | Байты сегмента топика смещения должны быть относительно небольшими, чтобы облегчить более быстрое сжатие лога и загрузку кэша. |
| password.encoder.cipher.algorithm | string | AES/CBC/PKCS5Padding | — | low | read-only | Алгоритм шифрования, используемый для кодирования динамически настраиваемых паролей. |
| password.encoder.iterations | int | 4096 | [1024,…] | low | read-only | Счётчик итераций, используемый для кодирования динамически настраиваемых паролей. |
| password.encoder.key.length | int | 128 | [8,…] | low | read-only | Длина ключа, используемая для кодирования динамически настраиваемых паролей. |
| password.encoder.keyfactory.algorithm | string | null | — | low | read-only | Алгоритм SecretKeyFactory, используемый для кодирования динамически настраиваемых паролей. По умолчанию используется PBKDF2WithHmacSHA512, если доступно, и PBKDF2WithHmacSHA1 в противном случае. |
| password.encoder.old.secret | password | null | — | medium | read-only | Старый секрет, который был использован для кодирования динамически настраиваемых паролей. Свойство требуется только для обновления секрета. Если значение указано, все динамически закодированные пароли декодируются с использованием указанного старого секрета и перекодируются с помощью password.encoder.secret при запуске брокера. |
| password.encoder.secret | password | null | — | medium | read-only | Секрет, используемый для кодирования динамически настраиваемых паролей для этого брокера. |
| port | int | 9092 | — | high | read-only | Используется только тогда, когда listeners не установлено. Рекомендуется использовать listeners.  Порт для прослушивания и приёма соединений. |
| principal.builder.class | class | null | — | medium | per-broker | Полное имя класса, реализующего интерфейс KafkaPrincipalBuilder, который используется для построения объекта KafkaPrincipal, используемого во время авторизации. Данная конфигурация также поддерживает устаревший интерфейс PrincipalBuilder, который ранее использовался для аутентификации клиентов через SSL. Если значение не определено, поведение по умолчанию зависит от используемого протокола безопасности. Для SSL-аутентификации принципал будет получен с использованием правил, определённых в свойстве ssl.principal.mapping.rules, применяемых к отличительному имени из сертификата клиента, если он предоставлен; в противном случае, если аутентификация клиента не требуется, основное имя будет ANONYMOUS. Для аутентификации SASL принципал будет получен с использованием правил, определённых в свойстве sasl.kerberos.principal.to.local.rules, если используется GSSAPI, и идентификатора аутентификации SASL для других механизмов. Для PLAINTEXT имя принципала будет ANONYMOUS. |
| producer.purgatory.purge.interval.requests | int | 1000 | — | medium | read-only | Интервал очистки (в количестве запросов) запросов поставщика. |
| queued.max.requests | int | 500 | [1,…] | high | read-only | Количество запросов в очереди, разрешённых для плоскости данных, до блокировки сетевых потоков. |
| queued.max.request.bytes | long | -1 | — | medium | read-only | Допустимое количество байт в очереди до того, как запросы больше не будут читаться. |
| quota.consumer.default | long | 9223372036854775807 | [1,…] | high | read-only | Используется только тогда, когда динамические квоты по умолчанию не настроены для или в Zookeeper. Любой потребитель, выделенный по clientId/consumer, будет ограничен, если он получит больше байт в секунду, чем установленное значение. |
| quota.producer.default | long | 9223372036854775807 | [1,…] | high | read-only | Используется только в том случае, если динамические квоты по умолчанию не настроены для или в Zookeeper. Любой поставщик с идентификатором clientId будет ограничен, если он производит больше байт в секунду, чем это значение. |
| quota.window.num | int | 11 | [1,…] | low | read-only | Количество выборок для хранения в памяти для квот клиентов. |
| quota.window.size.seconds | int | 1 | [1,…] | low | read-only | Временной интервал каждой выборки для квот клиентов. |
| replica.fetch.backoff.ms | int | 1000 (1 секунда) | [0,…] | medium | read-only | Время ожидания при возникновении ошибки партиции выборки. |
| replica.fetch.max.bytes | int | 1048576 (1 mebibyte) | [0,…] | medium | read-only | Количество байт сообщений, ожидаемые для получения для каждой партиции. Это не абсолютный максимум — если первый пакет записей в первой непустой партиции выборки больше, чем установленное значение, пакет записей всё равно будет возвращён, чтобы гарантировать выполнение прогресса. Максимальный размер пакета записей, принимаемый брокером, определяется через свойства message.max.bytes (конфигурация брокера) или max.message.bytes (конфигурация топика). |
| replica.fetch.min.bytes | int | 1 | — | high | read-only | Минимальное количество байт, ожидаемых для каждого ответа на запрос на выборку. Если байт недостаточно, дождитесь replica.fetch.wait.max.ms (конфигурация брокера). |
| replica.fetch.response.max.bytes | int | 10485760 (10 МиБ) | [0,…] | medium | read-only | Максимальное количество байт, ожидаемых для всего запроса выборки. Выборка записей организуется в пакеты, и если первый пакет записей в первой непустой партиции выборки больше, чем установленное значение, пакет записей всё равно будет возвращён, чтобы гарантировать выполнение прогресса. Таким образом, это не абсолютный максимум. Максимальный размер пакета записей, принимаемый брокером, определяется через свойства message.max.bytes (конфигурация брокера) или max.message.bytes (конфигурация топика). |
| replica.fetch.wait.max.ms | int | 500 | — | high | read-only | Максимальное время ожидания для каждого запроса на выборку, отправленного репликами-подписчиками. Это значение всегда должно быть меньше replica.lag.time.max.ms, чтобы предотвратить частое сокращение ISR для топиков с низкой пропускной способностью. |
| replica.high.watermark.checkpoint.interval.ms | long | 5000 (5 секунд) | — | high | read-only | Частота, с которой высокий водяной знак сохраняется на диск. |
| replica.lag.time.max.ms | long | 5000 | — | high | read-only | Если подписчик не отправлял никаких запросов на выборку или не конечное смещение журнала лидеров в течение хотя бы уставноленного времени, лидер удалит подписчика из isr. |
| replica.selector.class | string | null | — | medium | read-only | Полное имя класса, имплементирующего ReplicaSelector. Свойство используется брокером для поиска предпочтительной реплики для чтения. По умолчанию используется имплементация, которая возвращает лидера. |
| replica.socket.receive.buffer.bytes | int | 65536 (64 КиБ) | — | high | read-only | Буфер приёма сокета для сетевых запросов. |
| replica.socket.timeout.ms | int | 30000 (30 секунд) | — | high | read-only | Тайм-аут сокета для сетевых запросов. Установленное значение должно быть не меньше replica.fetch.wait.max.ms. |
| replication.quota.window.num | int | 11 | [1,…] | low | read-only | Количество выборок, сохраняемых в памяти для квот репликации. |
| replication.quota.window.size.seconds | int | 1 | [1,…] | low | read-only | Временной интервал каждой выборки для квот репликации. |
| request.timeout.ms | int | 30000 | — | high | read-only | Максимальное количество времени, в течение которого клиент будет ждать ответа на запрос. Если ответ не будет получен до истечения тайм-аута, клиент повторно отправит запрос, если это необходимо, или не выполнит запрос, если количество попыток исчерпано. |
| reserved.broker.max.id | int | 1000 | [0,…] | medium | read-only | Максимальный номер, который можно использовать для broker.id. |
| sasl.client.callback.handler.class | class | null | — | medium | read-only | Полное имя класса обработчика обратного вызова SASL-клиента, имплементирующего интерфейс AuthenticateCallbackHandler. |
| sasl.enabled.mechanisms | list | GSSAPI | — | medium | per-broker | Перечень SASL-механизмов, включённых на сервере Kafka. Перечень может содержать любой механизм, для которого активна безопасность. По умолчанию включён GSSAPI. |
| sasl.jaas.config | password | null | — | medium | per-broker | Параметры контекста входа JAAS для SASL-соединений в формате, используемом файлами конфигурации JAAS. [Здесь](http://docs.oracle.com/javase/8/docs/technotes/guides/security/jgss/tutorials/LoginConfigFile.html) описан формат файла конфигурации JAAS. Формат для значения: loginModuleClass controlFlag (optionName=optionValue)\*;. Для брокеров конфигурация должна начинаться с префикса слушателя и имени SASL-механизма в нижнем регистре. Например, listener.name.sasl\_ssl.scram-sha-256.sasl.jaas.config=com.example.ScramLoginModule required; |
| sasl.kerberos.kinit.cmd | string | /usr/bin/kinit | — | medium | per-broker | Путь к команде Kerberos kinit. |
| sasl.kerberos.min.time.before.relogin | long | 60000 | — | medium | per-broker | Время ожидания авторизации потока между попытками обновления. |
| sasl.kerberos.principal.to.local.rules | list | DEFAULT | — | medium | per-broker | Перечень правил для маппинга основных имён с короткими именами (обычно именами пользователей операционной системы). Правила оцениваются по порядку, и первое правило, соответствующее имени принципала, используется для сопоставления его с коротким именем. Все последующие правила в перечне игнорируются. По умолчанию имена принципалов указываются в форме {username}/{hostname}@{REALM}. Дополнительные сведения о формате см. в разделе [авторизация безопасности и ACL](https://kafka.apache.org/documentation/#security_authz). Обратите внимание, что данная конфигурация игнорируется, если расширение KafkaPrincipalBuilder предоставляется конфигурацией principal.builder.class. |
| sasl.kerberos.service.name | string | null | — | medium | per-broker | Имя принципала Kerberos, под которым работает Kafka. Имя можно определить либо в конфигурации Kafka JAAS, либо в конфигурации Kafka. |
| sasl.kerberos.ticket.renew.jitter | double | 0.05 | — | medium | per-broker | Процент случайного джиттера, добавленный ко времени возобновления. |
| sasl.kerberos.ticket.renew.window.factor | double | 0.8 | — | medium | per-broker | Поток входа в систему будет находиться в спящем режиме до тех пор, пока не будет достигнут установленный коэффициент окна времени с момента последнего обновления до истечения срока действия тикета, после чего он попытается обновить тикет. |
| sasl.login.callback.handler.class | class | null | — | medium | read-only | Полное имя класса обработчика обратного вызова входа SASL, имплементирующего интерфейс AuthenticateCallbackHandler. Для брокеров конфигурация обработчика обратного вызова при входе в систему должна иметь префикс с префиксом слушателя и именем SASL-механизма в нижнем регистре. Например, listener.name.sasl\_ssl.scram-sha-256.sasl.login.callback.handler.class = com.example.CustomScramLoginCallbackHandler. |
| sasl.login.class | class | null | — | medium | read-only | Полное имя класса, имплементирующего интерфейс входа в систему. Для брокеров конфигурация входа в систему должна начинаться с префикса слушателя и имени SASL-механизма в нижнем регистре. Например, listener.name.sasl\_ssl.scram-sha-256.sasl.login.class = com.example.CustomScramLogin. |
| sasl.login.refresh.buffer.seconds | short | 300 | — | medium | per-broker | Время буферизации до истечения срока действия учётных данных, которое необходимо поддерживать при обновлении учётных данных, в секундах. Если в противном случае обновление произошло бы ближе к истечению срока действия, чем количество секунд буфера, то обновление будет перемещено вверх, чтобы сохранить как можно большую часть времени буфера. Допустимые значения от 0 до 3600 (1 час). Если значение не указано, используется значение по умолчанию 300 (5 минут). Установленное значение и значение sasl.login.refresh.min.period.seconds игнорируются, если их сумма превышает оставшееся время жизни учётных данных. В настоящее время относится только к OAUTHBEARER. |
| sasl.login.refresh.min.period.seconds | short | 60 | — | medium | per-broker | Желаемое минимальное время ожидания потока обновления входа в систему перед обновлением учётных данных, в секундах. Допустимые значения от 0 до 900 (15 минут). Если значение не указано, используется значение по умолчанию 60 (1 минута). Установленное значение и значение sasl.login.refresh.buffer.seconds игнорируются, если их сумма превышает оставшееся время жизни учётных данных. В настоящее время относится только к OAUTHBEARER. |
| sasl.login.refresh.window.factor | double | 0.8 | — | medium | per-broker | Поток обновления входа в систему будет находиться в спящем режиме до тех пор, пока не будет достигнут установленный коэффициент окна относительно времени жизни учётных данных, после чего он попытается обновить учётные данные. Допустимые значения от 0.5 (50%) до 1.0 (100%) включительно. Если значение не указано, используется значение по умолчанию 0.8 (80%). В настоящее время относится только к OAUTHBEARER. |
| sasl.login.refresh.window.jitter | double | 0.05 | — | medium | per-broker | Максимальное количество случайного джиттера относительно времени жизни учётных данных, которое добавляется ко времени ожидания потока обновления входа в систему. Допустимые значения от 0 до 0.25 (25%) включительно. Если значение не указано, используется значение по умолчанию 0,05 (5%). В настоящее время относится только к OAUTHBEARER. |
| sasl.mechanism.inter.broker.protocol | string | GSSAPI | — | medium | per-broker | SASL-механизм, который используется для взаимодействия между брокерами. По умолчанию GSSAPI. |
| sasl.server.callback.handler.class | class | null | — | medium | per-broker | Полное имя класса обработчика обратного вызова SASL-сервера, имплементирующего интерфейс AuthenticateCallbackHandler. Обработчики обратных вызовов сервера должны начинаться с префикса слушателя и имени SASL-механизма в нижнем регистре. Например, listener.name.sasl\_ssl.plain.sasl.server.callback.handler.class = com.example.CustomPlainCallbackHan. |
| security.inter.broker.protocol | string | PLAINTEXT | [PLAINTEXT, SSL, SASL\_PLAINTEXT, SASL\_SSL] | medium | read-only | Протокол безопасности, который используется для взаимодействия между брокерами. Допустимые значения: PLAINTEXT, SSL, SASL\_PLAINTEXT, SASL\_SSL. Нельзя задавать ондовременно данное свойство и свойство inter.broker.listener.name. |
| security.providers | string | null | — | low | read-only | Перечень настраиваемых классов создателей, каждый из которых возвращает поставщика, имплементирующего алгоритмы безопасности. Установленный классы должны имплементировать интерфейс org.apache.kafka.common.security.auth.SecurityProviderCreator. |
| socket.connection.setup.timeout.max.ms | long | 30000 (30 секунд) | — | medium | read-only | Максимальное время, в течение которого клиент будет ожидать установления соединения с сокетом. Тайм-аут установки соединения будет экспоненциально увеличиваться при каждом последующем сбое соединения до установленного максимума. Чтобы избежать штормов соединений, к тайм-ауту будет применён коэффициент рандомизации 0.2, в результате чего получится случайный диапазон от 20% ниже до 20% выше вычисленного значения. |
| socket.connection.setup.timeout.ms | long | 10000 (10 секунд) | — | medium | read-only | Время, в течение которого клиент будет ожидать установления соединения с сокетом. Если соединение не будет установлено до истечения тайм-аута, клиенты закроют канал сокета. |
| socket.receive.buffer.bytes | int | 102400 (100 КиБ) | — | high | read-only | Буфер SO\_RCVBUF сокетов сервера сокетов. Если установлено значение -1, будет использоваться значение по умолчанию ОС. |
| socket.request.max.bytes | int | 104857600 (100 МиБ) | [1,…] | high | read-only | Максимальное количество байт в запросе сокета. |
| socket.send.buffer.bytes | int | 102400 (100 КиБ) | — | high | read-only | Буфер SO\_RCVBUF сокетов сервера сокетов. Если установлено значение -1, будет использоваться значение по умолчанию ОС. |
| ssl.cipher.suites | list | "" | — | medium | per-broker | Перечень комплектов шифров. Это именованная комбинация алгоритма аутентификации, шифрования, MAC и обмена ключами, используемая для согласования настроек безопасности для сетевого подключения с использованием сетевого протокола TLS или SSL. По умолчанию поддерживаются все доступные наборы шифров. |
| ssl.client.auth | string | none | [required, requested, none] | medium | per-broker | Настройка брокера Kafka для запроса аутентификации клиента. Следующие настройки являются общими:   * ssl.client.auth=required — требуется аутентификация клиента; * ssl.client.auth=requested — аутентификация клиента не является обязательной. В отличие от значения required, если этот параметр установлен, клиент может отказаться от предоставления аутентификационной информации о себе; * ssl.client.auth=none — аутентификация клиента не требуется. |
| ssl.enabled.protocols | list | TLSv1.2 | — | medium | per-broker | Перечень протоколов, разрешённых для SSL-соединений. По умолчанию используется TLSv1.2, TLSv1.3 при работе с Java 11 или новее, в противном случае — TLSv1.2. Со значением по умолчанию для Java 11 клиенты и серверы предпочтут TLSv1.3, если оба поддерживают его, и откатятся к TLSv1.2 в противном случае (при условии, что оба поддерживают как минимум TLSv1.2). Это значение по умолчанию подходит для большинства случаев. |
| ssl.endpoint.identification.algorithm | string | https | — | low | per-broker | Алгоритм идентификации конечной точки для валидации имени хоста сервера с помощью сертификата сервера. |
| ssl.engine.factory.class | class | null | — | low | per-broker | Класс типа org.apache.kafka.common.security.auth.SslEngineFactory для предоставления объектов SSLEngine. Значение по умолчанию org.apache.kafka.common.security.ssl.DefaultSslEngineFactory. |
| ssl.key.password | password | null | — | medium | per-broker | Пароль закрытого ключа в файле хранилища ключей или ключ PEM, указанный в ssl.keystore.key. Свойство требуется для клиентов, только если настроена двусторонняя аутентификация. |
| ssl.keymanager.algorithm | string | SunX509 | — | medium | per-broker | Алгоритм, используемый службой управления ключами для SSL-соединений. Значение по умолчанию — это алгоритм службы, настроенный Java Virtual Machine. |
| ssl.keystore.certificate.chain | password | null | — | medium | per-broker | Цепочка сертификатов в формате, указанном свойством ssl.keystore.type. По умолчанию служба SSL-механизма поддерживает только формат PEM с перечнем сертификатов X.509. |
| ssl.keystore.key | password | null | — | medium | per-broker | Закрытый ключ в формате, указанном свойством ssl.keystore.type. По умолчанию служба SSL-механизма поддерживает только формат PEM с ключами PKCS#8. Если ключ зашифрован, пароль ключа должен быть указан с помощью ssl.key.password. |
| ssl.keystore.location | string | null | — | medium | per-broker | Расположение файла хранилища ключей. Свойство опционально для клиента, и может использоваться для двусторонней аутентификации клиента. |
| ssl.keystore.password | password | null | — | medium | per-broker | Пароль хранилища для файла хранилища ключей. Свойство опционально для клиента, и требуется только в том случае, если настроено свойство ssl.keystore.location. Пароль хранилища ключей не поддерживается для формата PEM. |
| ssl.keystore.type | string | JKS | — | medium | per-broker | Формат файла хранилища ключей. Свойство опционально для клиента. |
| ssl.principal.mapping.rules | string | DEFAULT | — | low | read-only | Перечень правил для маппинга отличительного имени сертификата клиента с коротким именем. Правила оцениваются по порядку, и первое правило, соответствующее имени принципала, используется для сопоставления его с коротким именем. Все последующие правила в перечне игнорируются. По умолчанию отличительное имя сертификата X.500 будет принципалом. Обратите внимание, что данная конфигурация игнорируется, если расширение KafkaPrincipalBuilder предоставляется конфигурацией Principal.builder.class. |
| ssl.protocol | string | TLSv1.2 | — | medium | per-broker | SSL-протокол, используемый для генерации SSLContext. По умолчанию используется значение TLSv1.3 при работе с Java 11 или новее, в противном случае — TLSv1.2. Данное значение подходит для большинства случаев использования. Допустимые значения в последних JVM: TLSv1.2 и TLSv1.3. TLS, TLSv1.1, SSL, SSLv2 и SSLv3 могут поддерживаться в старых JVM, но их использование не рекомендуется из-за известных уязвимостей безопасности. Со значением по умолчанию для данной конфигурации и свойства ssl.enabled.protocols клиенты будут переходить на TLSv1.2, если сервер не поддерживает TLSv1.3. Если для данной конфигурации установлено значение TLSv1.2, клиенты не будут использовать TLSv1.3, даже если оно одно из значений в свойстве ssl.enabled.protocols, а сервер поддерживает только TLSv1.3. |
| ssl.provider | string | null | — | medium | per-broker | Имя поставщика безопасности, используемого для SSL-соединений. Значение по умолчанию — это поставщик безопасности для JVM по умолчанию. |
| ssl.secure.random.implementation | string | null | — | low | per-broker | Имплементация SecureRandom PRNG, используемая для операций шифрования SSL. |
| ssl.trustmanager.algorithm | string | PKIX | — | medium | per-broker | Алгоритм доверенной службы управления ключами для SSL-соединений. Значение по умолчанию — это алгоритм службы, настроенный для Java Virtual Machine. |
| ssl.truststore.certificates | password | null | — | medium | per-broker | Доверенные сертификаты в формате, указанном свойством ssl.truststore.type. Служба SSL-механизма по умолчанию поддерживает только формат PEM с сертификатами X.509. |
| ssl.truststore.location | string | null | — | medium | per-broker | Расположение файла хранилища доверенных сертификатов. |
| ssl.truststore.password | password | null | — | medium | per-broker | Пароль для файла доверенного хранилища. Если пароль не установлен, настроенный файл хранилища доверенных сертификатов по-прежнему будет использоваться, но проверка целостности будет отключена. Пароль доверенного хранилища не поддерживается для формата PEM. |
| ssl.truststore.type | string | JKS | — | medium | per-broker | Формат файла хранилища доверенных сертификатов. |
| transaction.abort.timed.out.transaction.cleanup.interval.ms | int | 10000 (10 секунд) | [1,…] | low | read-only | Интервал, в который откатываются транзакции, срок действия которых истёк. |
| transaction.max.timeout.ms | int | 900000 (15 минут) | [1,…] | high | read-only | Максимально допустимый тайм-аут для транзакций. Если время транзакции запроса клиента превышает установленное значение, брокер вернёт ошибку в InitProducerIdRequest. Это предотвращает слишком большой тайм-аут для клиента, который может помешать потребителям читать из топиков, включённых в транзакцию. |
| transaction.remove.expired.transaction.cleanup.interval.ms | int | 3600000 (1 час) | [1,…] | low | read-only | Интервал, в который удаляются транзакции, срок действия которых истёк в соответствии со значением свойства transactional.id.expiration.ms. |
| transaction.state.log.load.buffer.size | int | 5242880 | [1,…] | high | read-only | Размер пакета для чтения из сегментов лога транзакций при загрузке идентификаторов поставщиков и транзакций в кэш (мягкое ограничение, отменяется, если записи слишком велики). |
| transaction.state.log.min.isr | int | 2 | [1,…] | high | read-only | Переопределение конфигурации min.insync.replicas для топика транзакции. |
| transaction.state.log.num.partitions | int | 50 | [1,…] | high | read-only | Количество партиций для топика транзакции (не должно меняться после развёртывания). |
| transaction.state.log.replication.factor | short | 3 | [1,…] | high | read-only | Коэффициент репликации для топика транзакции (установка более высокого значения обеспечивает доступность). Создание внутреннего топика будет завершаться ошибкой, пока размер кластера не будет соответствовать данному требованию фактора репликации. |
| transaction.state.log.segment.bytes | int | 104857600 (100 МиБ) | [1,…] | high | read-only | Байты сегмента топика транзакции должны быть относительно небольшими, чтобы облегчить более быстрое сжатие лога и загрузку кэша. |
| transactional.id.expiration.ms | int | 604800000 (7 дней) | [1,…] | high | read-only | Время в мс, в течение которого координатор транзакции будет ждать без получения обновлений статуса транзакции для текущей транзакции до истечения срока действия её идентификатора транзакции. Данный параметр также влияет на истечение срока действия идентификатора поставщика — идентификаторы поставщика истекают по истечении установленного времени после последней записи с данным идентификатором поставщика. Обратите внимание, что идентификаторы поставщиков могут истечь раньше, если последняя запись от идентификатора поставщика будет удалена из-за настроек хранения топика. |
| unclean.leader.election.enable | boolean | false | — | high | cluster-wide | Указывает, следует ли включать реплики, не входящие в набор ISR, для выбора лидером в крайнем случае, даже если это может привести к потере данных. |
| zookeeper.clientCnxnSocket | string | null | — | medium | read-only | Обычно устанавливается на org.apache.zookeeper.ClientCnxnSocketNetty при использовании подключения TLS к ZooKeeper. Переопределяет любое явное значение, установленное через одноименное системное свойство zookeeper.clientCnxnSocket. |
| zookeeper.connect | string | null | — | high | read-only | Задаёт строку подключения ZooKeeper в форме hostname:port, где hostname и port — это хост и порт сервера ZooKeeper. Чтобы разрешить подключение через другие ноды ZooKeeper, если основная машина ZooKeeper не работает, вы также можете указать несколько хостов в форме hostname1:port1, hostname2:port2, hostname3:port3.  Сервер также может иметь chroot-путь ZooKeeper как часть своей строки подключения ZooKeeper, которая помещает свои данные по некоторому пути в глобальном пространстве имён ZooKeeper. Например, чтобы указать chroot-путь для /chroot/path, вы должны указать строку подключения как hostname1:port1,hostname2:port2,hostname3:port3/chroot/path. |
| zookeeper.connection.timeout.ms | int | null | — | high | read-only | Максимальное время, в течение которого клиент ожидает установления соединения с zookeeper. Если свойство не установлено, используется значение из свойства zookeeper.session.timeout.ms. |
| zookeeper.max.in.flight.requests | int | 10 | [1,…] | high | read-only | Максимальное количество неподтверждённых запросов, которые клиент отправит Zookeeper перед блокировкой. |
| zookeeper.session.timeout.ms | int | 18000 (18 секунд) | — | high | read-only | Тайм-аут сеанса Zookeeper. |
| zookeeper.set.acl | boolean | false | — | high | read-only | Настройка клиента для использования безопасных списков контроля доступа (ACL, Access Control List). |
| zookeeper.ssl.cipher.suites | list | null | — | low | read-only | Задаёт включённые комплекты шифров, которые будут использоваться при согласовании TLS ZooKeeper (csv). Переопределяет любое явное значение, установленное с помощью системного свойства zookeeper.ssl.ciphersuites (обратите внимание на слово «ciphersuites»). Значение по умолчанию null означает, что список включённых комплектов шифров определяется используемой средой выполнения Java. |
| zookeeper.ssl.client.enable | boolean | false | — | medium | read-only | Настройка клиента для использования TLS при подключении к ZooKeeper. Явное значение переопределяет любое значение, установленное с помощью системного свойства zookeeper.client.secure (обратите внимание на другое имя). По умолчанию false. Если установлено значение true, должно быть установлено значение для zookeeper.clientCnxnSocket (обычно org.apache.zookeeper.ClientCnxnSocketNetty). Могут быть установлены значения для свойств zookeeper.ssl.cipher.suites, zookeeper.ssl.crl.enable, zookeeper.ssl.enabled.protocols, zookeeper.ssl.endpoint.identification.algorithm, zookeeper.ssl.keystore.location, zookeeper.ssl.keystore.password, zookeeper.ssl.keystore.type, zookeeper.ssl.ocsp.enable, zookeeper.ssl.protocol, zookeeper.ssl.truststore.location, zookeeper.ssl.truststore.password, zookeeper.ssl.truststore.type. |
| zookeeper.ssl.crl.enable | boolean | false | — | low | read-only | Указывает, следует ли включить перечень отзыва сертификатов (Certificate Revocation List) в протоколах ZooKeeper TLS. Переопределяет любое явное значение, установленное через системное свойство zookeeper.ssl.crl (обратите внимание на более короткое имя). |
| zookeeper.ssl.enabled.protocols | list | null | — | low | read-only | Указывает включённые протоколы при согласовании ZooKeeper TLS (csv). Переопределяет любое явное значение, установленное с помощью системного свойства zookeeper.ssl.enabledProtocols (обратите внимание на camelCase). Значение по умолчанию null означает, что включённый протокол будет значением свойства конфигурации zookeeper.ssl.protocol. |
| zookeeper.ssl.endpoint.identification.algorithm | string | HTTPS | — | low | read-only | Указывает, следует ли включать проверку имени хоста в процессе согласования ZooKeeper TLS, при этом (без учёта регистра) https означает, что проверка имени хоста ZooKeeper включена, а явное пустое значение означает, что оно отключено (отключение рекомендуется только в целях тестирования). Явное значение переопределяет любое значение true или false, установленное с помощью системного свойства zookeeper.ssl.hostnameVerification (обратите внимание на другое имя и значения: true подразумевает https, а false подразумевает пустое значение). |
| zookeeper.ssl.keystore.location | string | false | — | medium | read-only | Расположение хранилища ключей при использовании сертификата на стороне клиента с подключением TLS к ZooKeeper. Переопределяет любое явное значение, установленное через системное свойство zookeeper.ssl.keyStore.location (обратите внимание на camelCase). |
| zookeeper.ssl.keystore.password | password | null | — | medium | read-only | Пароль хранилища ключей при использовании сертификата на стороне клиента с подключением TLS к ZooKeeper. Переопределяет любое явное значение, установленное с помощью системного свойства zookeeper.ssl.keyStore.password (обратите внимание на camelCase). Обратите внимание, что ZooKeeper не поддерживает пароль ключа, отличный от пароля хранилища ключей, поэтому убедитесь, что пароль ключа в хранилище ключей идентичен паролю хранилища; в противном случае попытка подключения к Zookeeper не удастся. |
| zookeeper.ssl.keystore.type | string | null | — | medium | read-only | Тип хранилища ключей при использовании сертификата на стороне клиента с подключением TLS к ZooKeeper. Переопределяет любое явное значение, установленное через системное свойство zookeeper.ssl.keyStore.type (обратите внимание на camelCase). Значение по умолчанию null означает, что тип будет обнаружен автоматически на основе расширения имени файла хранилища ключей. |
| zookeeper.ssl.ocsp.enable | boolean | false | — | low | read-only | Указывает, следует ли включить Online Certificate Status Protocol в протоколах ZooKeeper TLS. Переопределяет любое явное значение, установленное с помощью системного свойства zookeeper.ssl.ocsp (обратите внимание на более короткое имя). |
| zookeeper.ssl.protocol | string | TLSv1.2 | — | low | read-only | Задаёт протокол, который будет использоваться при согласовании ZooKeeper TLS. Явное значение переопределяет любое значение, установленное через одноимённое системное свойство zookeeper.ssl.protocol. |
| zookeeper.ssl.truststore.location | string | null | — | medium | read-only | Расположение доверенного хранилища при использовании TLS-подключения к ZooKeeper. Переопределяет любое явное значение, установленное через системное свойство zookeeper.ssl.trustStore.location (обратите внимание на camelCase). |
| zookeeper.ssl.truststore.password | password | null | — | medium | read-only | Пароль хранилища доверенных сертификатов при использовании TLS-подключения к ZooKeeper. Переопределяет любое явное значение, установленное с помощью системного свойства zookeeper.ssl.trustStore.password (обратите внимание на camelCase). |
| zookeeper.ssl.truststore.type | string | null | — | medium | read-only | Тип доверенного хранилища при использовании TLS-подключения к ZooKeeper. Переопределяет любое явное значение, установленное с помощью системного свойства zookeeper.ssl.trustStore.type (обратите внимание на camelCase). Значение по умолчанию null означает, что тип будет обнаружен автоматически на основе расширения имени файла хранилища доверенных сертификатов. |
| zookeeper.sync.time.ms | int | 2000 (2 секунды) | — | low | read-only | Отставание подписчика ZooKeeper от лидера ZooKeeper. |

Обновление конфигураций брокера

Начиная с версии Kafka 1.1 и новее, некоторые конфигурации брокера можно обновлять без перезапуска брокера. См. в таблице конфигурации брокера (Таблица 1) столбец **Режим обновления**, чтобы узнать о режиме обновления каждой конфигурации брокера.

read-only — требуется перезапуск брокера для обновления;

per-broker — динамическоеобновление для каждого брокера;

cluster-wide — динамическое обновление по умолчанию для всего кластера. Также может быть обновлено как значение per-broker для тестирования.

Чтобы изменить текущие конфигурации брокера для идентификатора брокера 0 (например, количество потоков очистки журнала):

|  |
| --- |
| > bin/kafka-configs.sh --bootstrap-server localhost:9092 --entity-type brokers --entity-name 0 --alter --add-config log.cleaner.threads=2 |

Чтобы описать текущие динамические конфигурации брокера для брокера с идентификатором 0:

|  |
| --- |
| > bin/kafka-configs.sh --bootstrap-server localhost:9092 --entity-type brokers --entity-name 0 --describe |

Чтобы удалить переопределение конфигурации и вернуться к статически настроенному значению или значению по умолчанию для идентификатора брокера 0 (например, количество потоков очистки журнала):

|  |
| --- |
| > bin/kafka-configs.sh --bootstrap-server localhost:9092 --entity-type brokers --entity-name 0 --alter --delete-config log.cleaner.threads |

Некоторые конфигурации могут быть настроены по умолчанию для всего кластера (cluster-wide), чтобы поддерживать консистентные значения во всём кластере. Все брокеры в кластере будут обрабатывать обновление кластера по умолчанию. Например, чтобы обновить потоки очистки журналов на всех брокерах:

|  |
| --- |
| > bin/kafka-configs.sh --bootstrap-server localhost:9092 --entity-type brokers --entity-default --alter --add-config log.cleaner.threads=2 |

Чтобы описать текущие настроенные динамические конфигурации по умолчанию для всего кластера (cluster-wide):

|  |
| --- |
| > bin/kafka-configs.sh --bootstrap-server localhost:9092 --entity-type brokers --entity-default --describe |

Все конфигурации, которые настраиваются на уровне кластера, также могут быть настроены на уровне каждого брокера (например, для тестирования). Если значение конфигурации определено на разных уровнях, используется следующий порядок приоритета:

1. Динамическая конфигурация для каждого брокера (per-broker), хранящаяся в ZooKeeper.

Динамическая конфигурация по умолчанию для всего кластера (cluster-wide), хранящаяся в ZooKeeper.

Статическая конфигурация брокера из свойства server.properties.

Конфигурация Kafka по умолчнаию (см. Конфигурации брокера).

Динамическое обновление конфигураций паролей

Значения конфигурации пароля, которые динамически обновляются, зашифровываются перед сохранением в ZooKeeper. Конфигурация брокера password.encoder.secret должна быть настроена в server.properties, чтобы разрешить динамическое обновление конфигураций паролей. Секрет может быть разным у разных брокеров.

Секрет, используемый для кодирования пароля, может быть изменён при непрерывном перезапуске брокеров. Старый секрет, используемый для кодирования паролей в настоящее время в ZooKeeper, должен быть указан в статической конфигурации брокера password.encoder.old.secret, а новый секрет должен быть указан в password.encoder.secret. Все динамические конфигурации паролей, хранящиеся в ZooKeeper, будут перекодированы с новым секретом при запуске брокера.

В Kafka 1.1.x все динамически обновляемые конфигурации паролей должны предоставляться в каждом запросе на изменение при обновлении конфигураций с помощью kafka-configs.sh, даже если конфигурация пароля не изменяется. Это ограничение будет удалено в следующем выпуске.

Обновление конфигураций паролей в ZooKeeper перед запуском брокеров

Начиная с Kafka 2.0.0, kafka-configs.sh позволяет обновлять динамические конфигурации брокеров с помощью ZooKeeper перед запуском брокеров для начальной загрузки. Такое обновление позволяет хранить все конфигурации паролей в зашифрованном виде, избегая необходимости в открытых паролях в server.properties. Конфигурация брокера password.encoder.secret также должна быть указана, если какие-либо конфигурации пароля включены в команду alter. Также могут быть указаны дополнительные параметры шифрования. Конфигурации кодировщика паролей не сохраняются в ZooKeeper. Например, чтобы сохранить пароль SSL-ключа для слушателя INTERNAL на брокере 0:

|  |
| --- |
| > bin/kafka-configs.sh --zookeeper localhost:2182 --zk-tls-config-file zk\_tls\_config.properties --entity-type brokers --entity-name 0 --alter --add-config  'listener.name.internal.ssl.key.password=key-password,password.encoder.secret=secret,password.encoder.iterations=8192' |

Конфигурация listener.name.internal.ssl.key.password будет сохранена в ZooKeeper в зашифрованном виде с использованием предоставленных конфигураций кодировщика. Секрет кодировщика и итерации не сохраняются в ZooKeeper.

Обновление хранилища SSL-ключей существующего слушателя

Брокеры могут быть сконфигурированы с использованием хранилищ SSL-ключей с короткими сроками действия, чтобы снизить риск взлома сертификатов. Хранилища ключей могут обновляться динамически без перезапуска брокера. Имя конфигурации должно начинаться с префикса слушателя listener.name.{listenerName}., из-за этого обновляется только конфигурация хранилища ключей определённого слушателя. Следующие конфигурации могут быть обновлены за один запрос на изменение на уровне брокера:

ssl.keystore.type;

ssl.keystore.location;

ssl.keystore.password;

ssl.key.password.

Если слушатель является слушателем между брокерами, обновление разрешено только в том случае, если новому хранилищу ключей доверяет хранилище доверенных сертификатов, настроенное для данного слушателя. Для других слушателей брокер не выполняет проверку доверия в хранилище ключей. Сертификаты должны быть подписаны тем же центром сертификации, который подписал старый сертификат, чтобы избежать сбоев проверки подлинности клиента.

Обновление хранилища доверенных сертификатов SSL существующего слушателя

Хранилища доверенных сертификатов брокера могут обновляться динамически без перезапуска брокера для добавления или удаления сертификатов. Обновленное хранилище доверенных сертификатов будет использоваться для аутентификации новых клиентских подключений. Имя конфигурации должно начинаться с префикса слушателя listener.name.{ListenerName}., из-за этого обновляется только конфигурация хранилища доверенных сертификатов определённого слушателя. Следующие конфигурации могут быть обновлены в одном запросе на изменение на уровне брокера:

ssl.truststore.type;

ssl.truststore.location;

ssl.truststore.password.

Если слушатель является слушателем между брокерами, обновление разрешено только в том случае, если существующее хранилище ключей для данного слушателя является доверенным для нового хранилища доверенных сертификатов. Для других слушателей брокер не выполняет проверку доверия перед обновлением. Удаление сертификатов CA, используемых для подписи клиентских сертификатов из нового хранилища доверенных сертификатов, может привести к сбоям проверки подлинности клиента.

Обновление конфигурации топика по умолчанию

Параметры по умолчанию конфигурации топика, используемые брокерами, могут быть обновлены без перезапуска брокера. Конфигурации применяются к топикамм без переопределения конфигурации топика для эквивалентной конфигурации для каждого топика. Одна или несколько из этих конфигураций могут быть переопределены на уровне кластера по умолчанию, используемом всеми брокерами.

log.segment.bytes;

log.roll.ms;

log.roll.hours;

log.roll.jitter.ms;

log.roll.jitter.hours;

log.index.size.max.bytes;

log.flush.interval.messages;

log.flush.interval.ms;

log.retention.bytes;

log.retention.ms;

log.retention.minutes;

log.retention.hours;

log.index.interval.bytes;

log.cleaner.delete.retention.ms;

log.cleaner.min.compaction.lag.ms;

log.cleaner.max.compaction.lag.ms;

log.cleaner.min.cleanable.ratio;

log.cleanup.policy;

log.segment.delete.delay.ms;

unclean.leader.election.enable;

min.insync.replicas;

max.message.bytes;

compression.type;

log.preallocate;

log.message.timestamp.type;

log.message.timestamp.difference.max.ms.

Начиная с версии Kafka 2.0.0 и далее, выбор грязного лидера автоматически включается контроллером при динамическом обновлении конфигурации unclean.leader.election.enable. В Kafka версии 1.1.x изменения unclean.leader.election.enable вступают в силу только при выборе нового контроллера. Переизбрание контролера может быть принудительно выполнено путём выполнения:

|  |
| --- |
| > bin/zookeeper-shell.sh localhost  rmr /controller |

Обновление конфигураций очистки журналов

Конфигурации очистки журналов могут обновляться динамически на уровне кластера по умолчанию, используемом всеми брокерами. Изменения вступят в силу при следующей итерации очистки журнала. Можно обновить одну или несколько из этих конфигураций:

log.cleaner.threads;

log.cleaner.io.max.bytes.per.second;

log.cleaner.dedupe.buffer.size;

log.cleaner.io.buffer.size;

log.cleaner.io.buffer.load.factor;

log.cleaner.backoff.ms.

Обновление конфигураций потоков

Размер различных пулов потоков, используемых брокером, может динамически обновляться на уровне кластера по умолчанию, используемом всеми брокерами. Обновления ограничены диапазоном от currentSize / 2 до currentSize \* 2, чтобы гарантировать корректную обработку обновлений конфигурации.

num.network.threads;

num.io.threads;

num.replica.fetchers;

num.recovery.threads.per.data.dir;

log.cleaner.threads;

background.threads.

Обновление конфигураций ConnectionQuota

Максимальное количество подключений, разрешённое брокером для данного IP-адреса/хоста, может динамически обновляться на уровне кластера по умолчанию, используемом всеми брокерами. Изменения будут применяться для создания новых подключений, а количество существующих подключений будет учтено новыми ограничениями.

max.connections.per.ip;

max.connections.per.ip.overrides.

Добавление и удаление слушателей

Слушатели могут быть добавлены или удалены динамически. Когда добавляется новый слушатель, конфигурации безопасности слушателя должны быть предоставлены как конфигурации слушателя с префиксом слушателя listener.name.{ListenerName}.. Если новый слушатель использует SASL, конфигурация JAAS слушателя должна быть предоставлена ​​с помощью JAAS конфигурационное свойство sasl.jaas.config с префиксом слушателя и механизма.

В Kafka версии 1.1.x слушатель, используемый промежуточным брокером, не может обновляться динамически. Чтобы обновить слушателя между брокерами до нового слушателя, новый слушатель может быть добавлен ко всем брокерам без перезапуска брокера. Затем требуется непрерывный перезапуск для обновления inter.broker.listener.name.

В дополнение ко всем конфигурациям безопасности новых слушателей, следующие конфигурации могут динамически обновляться на уровне каждого брокера:

listeners;

advertised.listeners;

listener.security.protocol.map.

Межброкерский слушатель должен быть настроен с использованием статической конфигурации брокера inter.broker.listener.name или inter.broker.security.protocol.

Конфигурации топика

Конфигурации топика имеют как серверное значение по умолчанию, так и опциональное переопределение для каждого топика (per-topic). Если конфигурация для топика не указана, используется серверное значение по умолчанию. Переопределение можно установить во время создания топика, задав один или несколько параметров --config.

В этом примере создаётся топик с именем my-topic с настраиваемым максимальным размером сообщения и скоростью сброса:

|  |
| --- |
| > bin/kafka-topics.sh --bootstrap-server localhost:9092 --create --topic my-topic --partitions 1 \  --replication-factor 1 --config max.message.bytes=64000 --config flush.messages=1 |

Переопределения также можно изменить или установить позже с помощью команды alter configs.

В этом примере обновляется максимальный размер сообщения для топика my-topic:

|  |
| --- |
| > bin/kafka-configs.sh --bootstrap-server localhost:9092 --entity-type topics --entity-name my-topic  --alter --add-config max.message.bytes=128000 |

Чтобы проверить переопределения, установленные в топике, вы можете отправить:

|  |
| --- |
| > bin/kafka-configs.sh --bootstrap-server localhost:9092 --entity-type topics --entity-name my-topic --describe |

Чтобы удалить переопределение, вы можете отправить:

|  |
| --- |
| > bin/kafka-configs.sh --bootstrap-server localhost:9092 --entity-type topics --entity-name my-topic  --alter --delete-config max.message.bytes |

Конфигурации топика и их значения по умолчанию представлены в таблице ниже в алфавитном порядке по наименованию конфигурации.

Конфигурация сервера по умолчанию для указанного свойства указана в столбце **Свойство сервера по умолчанию**. Данное значение конфигурации сервера по умолчанию применяется к топику только в том случае, если для него не установлено явного переопределения конфигурации топика.

Таблица — Конфигурации топика

| Конфигурация | Тип | Значение по умолчанию | Действительные значения | Свойство сервера по умолчанию | Важность | Описание |
| --- | --- | --- | --- | --- | --- | --- |
| cleanup.policy | list | delete | [compact, delete] | log.cleanup.policy | medium | Политика хранения для использования старых сегментов логов. Значение либо delete, либо compact, либо оба значения. Политика по умолчанию (delete) отбрасывает старые сегменты по достижении срока их хранения или предельного размера. Параметр compact включает сжатие лога по топику. |
| compression.type | string | producer | [uncompressed, zstd, lz4, snappy, gzip, producer] | compression.type | medium | Указывает окончательный тип сжатия для укзанного топика. Данная конфигурация принимает стандартные кодеки сжатия (gzip, snappy, lz4, zstd). Кроме того, он принимает значение uncompressed (что эквивалентно отсутствию сжатия) и producer (что означает сохранение исходного кодека сжатия, установленного поставщиком). |
| delete.retention.ms | long | 86400000 (1 день) | [0,…] | log.cleaner.delete.retention.ms | medium | Продолжительность хранения данных, маркированных на удаление, сжатых топиков в логах. Установленный параметр также даёт ограничение на время, в течение которого потребитель должен завершить чтение, если он начинает со смещения 0, чтобы гарантировать, что он получит валидный снапшот последнего этапа (иначе маркированные данные могут быть удалены прежде, чем завершится их сканирование). |
| file.delete.delay.ms | long | 60000 (1 минута) | [0,…] | log.segment.delete.delay.ms | medium | Время ожидания перед удалением файла из файловой системы. |
| flush.messages | long | 9223372036854775807 | [0,…] | log.flush.interval.messages | medium | Интервал, с которым выполняется принудительная синхронизация данных, записываемых в лог. Например, если было установлено значение 1, синхронизация будет выполняться после каждого сообщения; если значение 5, синхронизация будет выполняться через каждые 5 сообщениц. Как правило, рекомендуется не устанавливать данный параметр, а использовать репликацию для обеспечения надёжности и разрешить фоновую очистку операционной системы, поскольку это более эффективно. Данный параметр можно переопределить для каждого топика. |
| flush.ms | long | 9223372036854775807 | [0,…] | log.flush.interval.ms | medium | Временной интервал, в течение которого принудительно выполняется синхронизация данных, записываемых в лог. Например, если было установлено значение 1000, принудительная синхронизация будет выполняться по прошествии 1000 мс. Как правило, рекомендуется не устанавливать данный параметр, а использовать репликацию для обеспечения надёжности и разрешить фоновую очистку операционной системы, поскольку это более эффективно. Данный параметр можно переопределить для каждого топика. |
| follower.replication.throttled.replicas | list | "" | [partitionId]:[brokerId],[partitionId]:[brokerId],… | follower.replication.throttled.replicas | medium | Перечень реплик, для которых репликация логов должна регулироваться на стороне подписчика. Список должен описывать набор реплик в форме [PartitionId]:[BrokerId],[PartitionId]:[BrokerId],…. Также можно использовать подстановочный символ \* для ограничения всех реплик для выбранного топика. |
| index.interval.bytes | int | 4096 (4 КиБ) | [0,…] | log.index.interval.bytes | medium | Периодичность добавления индексной записи в индекс смещения. Значение по умолчанию гарантирует, что сообщение индексируется примерно каждые 4096 байт. Более частая индексация позволяет читать ближе к точной позиции в логе, но увеличивает индекс. Рекомендуется не менять значение по умолчанию. |
| leader.replication.throttled.replicas | list | "" | [partitionId]:[brokerId],[partitionId]:[brokerId],… | leader.replication.throttled.replicas | medium | Перечень реплик, для которых репликация логов должна регулироваться на стороне лидера. Список должен описывать набор реплик в форме [PartitionId]:[BrokerId],[PartitionId]:[BrokerId],…. Также можно использовать подстановочный символ \* для ограничения всех реплик для выбранного топика. |
| max.compaction.lag.ms | long | 9223372036854775807 | [1,…] | log.cleaner.max.compaction.lag.ms | medium | Максимальное время, в течение которого сообщение не подлежит сжатию в журнале. Применимо только для логов, для которых включено сжатие. |
| max.message.bytes | int | 1048588 | [0,…] | message.max.bytes | medium | Максимальный размер пакета записей, разрешённый Kafka (после сжатия, если сжатие включено). Если размер был увеличен и есть потребители старше 0.10.2, размер выборки потребителей также должен быть увеличен, чтобы они могли получать пакеты записей установленного размера. В последней версии формата сообщений записи всегда сгруппированы в пакеты для повышения эффективности. В предыдущих версиях формата сообщений несжатые записи не группировались в пакеты, и в этом случае это ограничение применяется только к одной записи. |
| message.downconversion.enable | boolean | true | — | log.message.downconversion.enable | low | Данная конфигурация определяет, разрешено ли понижающее преобразование форматов сообщений для удовлетворения запросов на потребление. Если установлено значение false, брокер не будет выполнять понижающее преобразование для потребителей, ожидающих более старого формата сообщения. Брокер отвечает ошибкой UNSUPPORTED\_VERSION для запросов на потребление от таких старых клиентов. Данная конфигурация не применяется к любому преобразованию формата сообщения, которое может потребоваться для репликации подписчикам. |
| message.format.version | string | 2.8-IV1 | [0.8.0, 0.8.1, 0.8.2, 0.9.0, 0.10.0-IV0, 0.10.0-IV1, 0.10.1-IV0, 0.10.1-IV1, 0.10.1-IV2, 0.10.2-IV0, 0.11.0-IV0, 0.11.0-IV1, 0.11.0-IV2, 1.0-IV0, 1.1-IV0, 2.0-IV0, 2.0-IV1, 2.1-IV0, 2.1-IV1, 2.1-IV2, 2.2-IV0, 2.2-IV1, 2.3-IV0, 2.3-IV1, 2.4-IV0, 2.4-IV1, 2.5-IV0, 2.6-IV0, 2.7-IV0, 2.7-IV1, 2.7-IV2, 2.8-IV0, 2.8-IV1] | log.message.format.version | medium | Установка версии формата сообщения, которую брокер будет использовать для добавления сообщений в логи. Значение должно быть валидным ApiVersion. Вот несколько примеров: 0.8.2, 0.9.0.0, 0.10.0. Устанавливая конкретную версию формата сообщения, пользователь удостоверяет, что все существующие сообщения на диске меньше или равны указанной версии. Неправильная установка данного значения приведёт к тому, что потребители более старых версий перестанут работать, поскольку они будут получать сообщения в формате, который им не понятен. |
| message.timestamp.difference.max.ms | long | 9223372036854775807 | [0,…] | log.message.timestamp.difference.max.ms | medium | Максимально допустимая разница между отметкой времени, когда брокер получает сообщение, и отметкой времени, указанной в сообщении. Если message.timestamp.type=CreateTime, сообщение будет отклонено, если разница во времени превышает этот порог. Данная конфигурация игнорируется, если message.timestamp.type=LogAppendTime. |
| message.timestamp.type | string | CreateTime | [CreateTime, LogAppendTime] | log.message.timestamp.type | medium | Определяет, является ли метка времени в сообщении временем создания сообщения или временем добавления в лог. Значение должно быть либо CreateTime, либо LogAppendTime. |
| min.cleanable.dirty.ratio | double | 0.5 | [0,…,1] | log.cleaner.min.cleanable.ratio | medium | Частота очистки лога (при условии, что сжатие логавключено). По умолчанию избегается очистка лога, в котором сжато более 50% данных. Данное соотношение ограничивает максимальное пространство, выделенное в логе под дубликаты (при установленных 50% максимум 50% журнала могут быть дубликатами). Установленное более высокое значение соотношения означает, что очистки являются более эффективными, но место в логах будет неэффективно использоваться. Если также указаны конфигурации max.compaction.lag.ms или min.compaction.lag.ms, то то механизм сжатия логов считает, что лог подходит для сжатия, как только:  был достигнут установленный порог, и в логе были несжатые записи в течение, по крайней мере, времени, установленного в свойстве min.compaction.lag.ms;  если в логе были несжатые записи в течение времени, установленного в свойстве max.compaction.lag.ms. |
| min.compaction.lag.ms | long | 0 | [0,…] | log.cleaner.min.compaction.lag.ms | medium | Минимальное время, в течение которого сообщение остаётся в логе без сжатия. Применимо только для логов, для которых установлено сжатие. |
| min.insync.replicas | int | 1 | [1,…] | min.insync.replicas | medium | Когда поставщик устанавливает для свойства acks значение all (или -1), min.insync.replicas указывает минимальное количество реплик, которые должны подтвердить запись, чтобы запись считалась успешной. Если этот минимум не может быть соблюден, поставщик вызовет исключение (NotEnoughReplicas или NotEnoughReplicasAfterAppend).  Совместное использование min.insync.replicas и acks обеспечивает бОльшую надёжность. Типичный сценарий — создать топик с коэффициентом репликации равным 3, установить min.insync.replicas равным 2 и acks равным all. Установленные настройки гарантирует, что поставщик вызовет исключение, если большинство реплик не получит запись. |
| preallocate | boolean | false | — | log.preallocate | medium | Значение true, если файл предварительно размещается на диске при создании нового сегмента лога. |
| retention.bytes | long | -1 | — | log.retention.bytes | medium | Максимальный размер, до которого может вырасти партиция (состоящая из сегментов лога), прежде чем будут отбрасываться старые сегменты лога для освобождения места, если используется политика хранения delete. По умолчанию ограничений по размеру нет, только ограничение по времени. Поскольку это ограничение применяется на уровне партиций, умножьте его на количество партиций, чтобы вычислить хранение топика в байтах. |
| retention.ms | long | 604800000 (7 дней) | [-1,…] | log.retention.ms | medium | Максимальное время, в течение которого хранится лог, прежде чем будут отброшены старые сегменты лога, чтобы освободить место, если используется политика хранения delete. Значение представляет собой SLA о том, как быстро потребители должны прочитать свои данные. Если установлено значение -1, ограничение по времени не применяется. |
| segment.bytes | int | 1073741824 (1 ГиБ) | [14,…] | log.segment.bytes | medium | Размер файла сегмента для лога. Хранение и очистка всегда выполняются по одному файлу, поэтому больший размер сегмента означает меньшее количество файлов, но меньший контроль над хранением. |
| segment.index.bytes | int | 10485760 (10 МиБ) | [0,…] | log.index.size.max.bytes | medium | Размер индекса, который маппит смещения на позиции в файлах. Данный индексный файл выделяется заранее и сжимается только после просмотра лога. Обычно изменять установленное значение не требуется. |
| segment.jitter.ms | long | 0 | [0,…] | log.roll.jitter.ms | medium | Максимальный случайный джиттер, вычитаемый из запланированного времени прокрутки сегмента, чтобы избежать проблемы сегментов thundering herds. |
| segment.ms | long | 604800000 (7 дней) | [1,…] | log.roll.ms | medium | Период времени, по истечении которого Kafka будет принудительно запускать сжатие лога, даже если файл сегмента не заполнен, чтобы гарантировать сохранение и сжатие устаревших данных. |
| unclean.leader.election.enable | boolean | false | — | unclean.leader.election.enable | medium | Указывает, следует ли включать реплики, не входящие в набор ISR, для выбора лидером в крайнем случае, даже если это может привести к потере данных. |

Конфигурации поставщика

Конфигурации producer (поставщика) и их значения по умолчанию представлены в таблице ниже в алфавитном порядке по наименованию конфигурации.

Таблица — Конфигурации поставщика

| Конфигурация | Тип | Значение по умолчанию | Действительные значения | Важность | Описание |
| --- | --- | --- | --- | --- | --- |
| acks | string | 1 | [all, -1, 0, 1] | high | Количество подтверждений, которые поставщик требует, чтобы лидер получил, прежде чем считать запрос завершённым. Конфигурация контролирует устойчивость отправляемых записей. Допускаются следующие значения:  acks=0 — поставщик не будет ждать подтверждения от сервера. Запись будет немедленно добавлена ​​в буфер сокета и считаться отправленной. В этом случае нельзя гарантировать, что сервер получит запись, и конфигурация повторных попыток не вступит в силу (поскольку клиент обычно не узнает о каких-либо сбоях). Смещение, возвращаемое для каждой записи, всегда будет равно -1.  acks=1 — лидер будет записывать запись в свой локальный журнал, но будет отвечать, не дожидаясь полного подтверждения от всех подписчиков. В этом случае, если лидер выйдет из строя сразу после подтверждения записи, но до того, как подписчики реплицируют её, запись будет потеряна.  acks=all — лидер будет ждать, пока полный набор синхронизированных реплик подтвердит запись. Это гарантирует, что запись не будет потеряна, пока остается жива хотя бы одна синхронизированная реплика. Это самая надёжная доступная гарантия. Данное значение эквивалентно настройке acks=-1. |
| batch.size | int | 16384 | [0,…] | medium | Поставщик будет пытаться объединить пакеты в меньшее количество запросов всякий раз, когда несколько пакетов отправляются в одну и ту же партицию. Такое объединение помогает повысить производительность как на клиенте, так и на сервере. Данная конфигурация управляет размером пакета по умолчанию в байтах.  Попытки пакетирования записей больше установленного размера предприниматься не будут.  Запросы, отправленные брокерам, будут содержать несколько пакетов, по одному для каждой партиции, с данными, доступными для отправки.  Маленький размер пакета делает пакетирование менее общим и может снизить пропускную способность (нулевой размер пакета полностью отключит пакетирование). Очень большой размер пакета может использовать память немного более расточительно, поскольку всегда будет выделяться буфер указанного размера пакета в ожидании дополнительных записей. |
| bootstrap.servers | list | "" | non-null string | high | Перечень пар хост/порт, используемых для установления начального подключения к кластеру Kafka. Клиент будет использовать все серверы независимо от того, какие серверы указаны здесь для начальной подключения — данный перечень влияет только на инициирующие хосты, используемые для обнаружения полного набора серверов. Данный перечень должен быть в формате: host1:port1,host2:port2,…. Поскольку эти серверы используются только для начального соединения, чтобы обнаружить полное членство в кластере (которое может динамически меняться), данный перечень не обязательно должен содержать полный набор серверов. |
| buffer.memory | long | 33554432 | [0,…] | high | Общий объём памяти в байтах, который поставщик может использовать для буферизации записей, ожидающих отправки на сервер. Если записи отправляются быстрее, чем они могут быть доставлены на сервер, поставщик блокирует свойство max.block.ms, после чего генерирует исключение.  Данный параметр должен примерно соответствовать общей памяти, которую будет использовать поставщик, но не является жёсткой границей, поскольку не вся память, которую использует поставщик, используется для буферизации. Некоторая дополнительная память будет использоваться для сжатия (если сжатие включено), а также для поддержки запросов in-flight. |
| client.dns.lookup | string | use\_all\_dns\_ips | [default, use\_all\_dns\_ips, resolve\_canonical\_bootstrap\_servers\_only] | medium | Параметр управляет тем, как клиент использует поиск DNS. Если установлено значение use\_all\_dns\_ips, клиент будеет последовательно подключаться к каждому возвращённому IP-адресу до тех пор, пока не будет установлено успешное соединение. После отключения используется следующий IP-адрес. После того, как все IP-адреса были использованы один раз, клиент снова разрешает IP-адреса из имени хоста (поиск DNS-имени как JVM, так и кэша ОС).  Если установлено значение resolve\_canonical\_bootstrap\_servers\_only, разрешается каждый адрес начальной загрузки в списке канонических имён. После фазы начальной загрузки поиск DNS осуществляется так, как при установленном значении use\_all\_dns\_ips.  Если установлено значение default (не рекомендуется), осуществляется попытка подключиться к первому IP-адресу, возвращённому поиском, даже если поиск вернёт несколько IP-адресов. |
| client.id | string | "" | — | medium | Строка идентификатора, передаваемая на сервер при выполнении запросов. Это делается для того, чтобы иметь возможность отслеживать источник запросов помимо IP-адреса/порта, позволяя включать логическое имя приложения в лог запросов на стороне сервера. |
| compression.type | string | none | [none, gzip, snappy, lz4, zstd] | high | Тип сжатия для всех данных, созданных поставщиком. По умолчанию установлено значение none (т.е. без сжатия). Допустимые значения: none, gzip, snappy, lz4 или zstd. Сжатие производится над полным пакетом данных, поэтому эффективность пакетирования также влияет на степень сжатия (большее пакетирование означает лучшее сжатие). |
| connections.max.idle.ms | long | 540000 (9 минут) | — | medium | Неактивные подключения закрываются по истечении количества миллисекунд, указанного в данной конфигурации. |
| delivery.timeout.ms | int | 120000 (2 минуты) | [0,…] | medium | Верхняя граница времени для отчёта об успехе или неудаче после возврата вызова метода send(). Установленный параметр ограничивает общее время, на которое запись будет отложена перед отправкой, время ожидания подтверждения от брокера (если ожидается) и время, разрешённое для повторных сбоев отправки. Поставщик может отчитаться о неудаче отправить запись раньше, чем установленное значение конфигурации, если обнаружена неисправимая ошибка, количество повторных попыток исчерпано или запись добавлена в пакет, который достиг более раннего срока истечения доставки. Значение даноой конфигурации должно быть больше или равно сумме значений свойств request.timeout.ms и linger.ms. |
| enable.idempotence | boolean | false | — | low | Если установлено значение true, поставщик гарантирует, что в поток записывается ровно одна копия каждого сообщения. Если false, поставщик будет повторять попытки из-за сбоев брокера и т.д., поэтому могут быть записаны дубликаты повторного сообщения в поток. Обратите внимание, что для включения идемпотентности требуется, чтобы значение свойства max.in.flight.requests.per.connection было меньше или равно 5, количество попыток было больше 0, а для acks должно быть установлено all. Если эти значения явно не установлены пользователем, будут выбраны подходящие значения. Если установлены несовместимые значения, будет выброшено исключение ConfigException. |
| interceptor.classes | list | "" | non-null string | low | Перечень классов для использования в качестве перехватчиков (interceptors). Имплементация интерфейса org.apache.kafka.clients.producer.ProducerInterceptor позволяет перехватывать (и, возможно, изменять) записи, полученные поставщиком, до их публикации в кластере Kafka. По умолчанию установлено отсутствие перехватчиков. |
| key.serializer | class | — | — | high | Класс сериализатора (Serializer) для ключа, который имплементирует интерфейс org.apache.kafka.common.serialization.Serializer. |
| linger.ms | long | 0 | [0,…] | medium | Поставщик группирует все записи, поступающие между передачами запроса, в один пакетный запрос. Обычно это происходит только под нагрузкой, когда записи приходят быстрее, чем могут быть отправлены. Однако в некоторых случаях клиент может захотеть уменьшить количество запросов даже при умеренной нагрузке. Данный параметр выполняет это путём добавления небольшого количества искусственной задержки, то есть вместо немедленной отправки записи поставщик будет ждать до заданной задержки, чтобы разрешить отправку других записей, чтобы отправки могли быть объединены вместе в пакет. Данную конфигурацию можно рассматривать как аналог алгоритма Nagle в TCP. Этот параметр задаёт верхнюю границу задержки для пакетной обработки: как только будут получены записи для партиции размером batch.size, они будут отправлены немедленно, независимо от установленного значения данного параметра, однако, если для партиции будет накоплено меньше установленного количества байт, отправка будет задерживаться на указанное время, ожидая появления новых записей. По умолчанию значение параметра равно 0 (т.е. без задержки). Например, установка linger.ms=5 приведёт к уменьшению количества отправленных запросов, но добавит до 5 мс задержки к записям, отправляемым при отсутствии нагрузки. |
| max.block.ms | long | 60000 (1 минута) | [0,…] | medium | Конфигурация определяет, как долго методы KafkaProducer send(), partitionsFor(), initTransactions(), sendOffsetsToTransaction(), commitTransaction() и abortTransaction() будут блокироваться. Для send() данный тайм-аут ограничивает общее время ожидания как выборки метаданных, так и выделения буфера (блокировка в предоставляемых пользователем сериализаторах или партиционерах не засчитывается в данный тайм-аут). Для partitionsFor() данный тайм-аут ограничивает время ожидания метаданных, если они недоступны. Связанные с транзакцией методы всегда блокируются, но могут отключиться по тайм-ауту, если координатор транзакции не может быть обнаружен или не ответил в течение тайм-аута. |
| max.in.flight.requests.per.connection | int | 5 | [1,…] | low | Максимальное количество неподтверждённых запросов, которые клиент отправит в одном подключении до блокировки. Обратите внимание, что если для данного параметра установлено значение больше 1 и есть неудачные отправки, существует риск переупорядочения сообщений из-за повторных попыток (если повторные попытки разрешены). |
| max.request.size | int | 1048576 | [0,…] | medium | Максимальный размер запроса в байтах. Данный параметр ограничивает количество пакетов записей, которые поставщик отправляет в одном запросе, чтобы избежать отправки огромных запросов. Это также эффективное ограничение максимального размера пакета несжатой записи. Обратите внимание, что сервер имеет собственное ограничение на размер пакета записи (после сжатия, если сжатие включено), которое может отличаться от данного. |
| metadata.max.age.ms | long | 300000 (5 минут) | [0,…] | low | Период времени в миллисекундах, по истечении которого метаданные будут принудительно обновлены, даже если отсутствуют какие-либо изменения в лидере партиции, чтобы упреждающе обнаруживать новых брокеров или партиции. |
| metadata.max.idle.ms | long | 300000 (5 минут) | [5000,…] | low | Параметр определяет, как долго поставщик будет кэшировать метаданные для пустого топика. Если время, прошедшее с момента последнего создания топика, превышает продолжительность простоя метаданных, то метаданные топика забываются, и следующий доступ к ней приведёт к принудительному запросу на выборку метаданных. |
| metric.reporters | list | "" | non-null string | low | Перечень классов для использования в качестве репортеров метрик (metrics reporters). Имплементация интерфейса org.apache.kafka.common.metrics.MetricsReporter позволяет подключать классы, которые будут получать уведомления о создании новой метрики. JmxReporter всегда включён для регистрации статистики JMX. |
| metrics.num.samples | int | 2 | [1,…] | low | Количество образцов, поддерживаемых для вычисления метрик. |
| metrics.recording.level | string | INFO | [INFO, DEBUG, TRACE] | low | Самый высокий уровень записи метрик. |
| metrics.sample.window.ms | long | 30000 (30 секунд) | [0,…] | low | Временной промежуток, в течение которого вычисляется образец метрики. |
| partitioner.class | class | org.apache.kafka.clients.producer.internals.DefaultPartitioner | — | medium | Класс партиционера (Partitioner), имплементирующий интерфейс org.apache.kafka.clients.producer.Partitioner. |
| receive.buffer.bytes | int | 32768 (32 КиБ) | [-1,…] | medium | Размер приёмного буфера TCP (SO\_RCVBUF) для использования при чтении данных. Если установлено значение -1, будет использоваться значение по умолчанию ОС. |
| reconnect.backoff.max.ms | long | 1000 (1 секунда) | [0,…] | low | Максимальное время ожидания в миллисекундах при повторном подключении к брокеру, к которому неоднократно не удавалось подключиться. Если значение установлено, отсрочка для каждого хоста будет экспоненциально увеличиваться при каждом последующем сбое соединения, вплоть до установленного максимума. После расчёта увеличения отсрочки добавляется 20% случайного джиттера, чтобы избежать штормов соединения. |
| reconnect.backoff.ms | long | 50 | [0,…] | low | Базовое время ожидания перед попыткой повторного подключения к данному хосту. Это позволяет избежать многократного подключения к хосту в замкнутом цикле. Установленная отсрочка применяется ко всем попыткам подключения клиента к брокеру. |
| request.timeout.ms | int | 30000 (30 секунд) | [0,…] | medium | Конфигурация контролирует максимальное количество времени, в течение которого клиент будет ждать ответа на запрос. Если ответ не будет получен до истечения тайм-аута, клиент повторно отправит запрос, если необходимо, или запрос завершится неудачей, если количество повторных попыток исчерпано. Значение должно быть больше, чем replica.lag.time.max.ms (конфигурация брокера), чтобы уменьшить вероятность дублирования сообщений из-за ненужных повторных попыток поставщика. |
| retries | int | 2147483647 | [0,…,2147483647] | high | Установка значения больше нуля приведёт к тому, что клиент повторно отправит любую запись, отправка которой не удалась с потенциально временной ошибкой. Обратите внимание, что эта повторная попытка ничем не отличается от повторной отправки записи клиентом после получения ошибки. Разрешение повторных попыток без установки значения 1 свойству max.in.flight.requests.per.connection потенциально изменит порядок записей, потому что, если два пакета отправляются в одну партицию и отправка первого пакета не удастся и повторится, а отправка второго будет успешна, тогда записи второго пакета можут появиться первыми. Кроме того, обратите внимание, что запросы на создание будут неудачными до того, как будет исчерпано количество повторных попыток, если тайм-аут, настроенный в свойстве delivery.timeout.ms, истекает раньше, чем будет получено успешное подтверждение. Обычно пользователи предпочитают не задавать данную конфигурацию и вместо этого использовать свойство delivery.timeout.ms для управления поведением повторных попыток. |
| retry.backoff.ms | long | 100 | [0,…] | low | Время ожидания перед попыткой повторить неудавшийся запрос к данной партиции топика. Это позволяет избежать повторной отправки запросов в замкнутом цикле при некоторых сценариях сбоя. |
| sasl.client.callback.handler.class | class | null | — | medium | Полное имя класса обработчика обратного вызова SASL-клиента, имплементирующего интерфейс AuthenticateCallbackHandler. |
| sasl.jaas.config | password | null | — | medium | Параметры контекста входа JAAS для SASL-соединений в формате, используемом файлами конфигурации JAAS. [Здесь](http://docs.oracle.com/javase/8/docs/technotes/guides/security/jgss/tutorials/LoginConfigFile.html) описан формат файла конфигурации JAAS. Формат для значения: loginModuleClass controlFlag (optionName=optionValue)\*;. Для брокеров конфигурация должна начинаться с префикса слушателя и имени SASL-механизма в нижнем регистре. Например, listener.name.sasl\_ssl.scram-sha-256.sasl.jaas.config=com.example.ScramLoginModule required; |
| sasl.kerberos.kinit.cmd | string | /usr/bin/kinit | — | low | Путь к команде Kerberos kinit. |
| sasl.kerberos.min.time.before.relogin | long | 60000 | — | low | Время ожидания авторизации потока между попытками обновления. |
| sasl.kerberos.service.name | string | null | — | medium | Имя принципала Kerberos, под которым работает Kafka. Имя можно определить либо в конфигурации Kafka JAAS, либо в конфигурации Kafka. |
| sasl.kerberos.ticket.renew.jitter | double | 0.05 | — | low | Процент случайного джиттера, добавленный ко времени возобновления. |
| sasl.kerberos.ticket.renew.window.factor | double | 0.8 | — | low | Поток входа в систему будет находиться в спящем режиме до тех пор, пока не будет достигнут установленный коэффициент окна времени с момента последнего обновления до истечения срока действия тикета, после чего он попытается обновить тикет. |
| sasl.login.callback.handler.class | class | null | — | medium | Полное имя класса обработчика обратного вызова входа SASL, имплементирующего интерфейс AuthenticateCallbackHandler. Для брокеров конфигурация обработчика обратного вызова при входе в систему должна иметь префикс с префиксом слушателя и именем SASL-механизма в нижнем регистре. Например, listener.name.sasl\_ssl.scram-sha-256.sasl.login.callback.handler.class = com.example.CustomScramLoginCallbackHandler. |
| sasl.login.refresh.buffer.seconds | short | 300 | [0,…,3600] | low | Время буферизации до истечения срока действия учётных данных, которое необходимо поддерживать при обновлении учётных данных, в секундах. Если в противном случае обновление произошло бы ближе к истечению срока действия, чем количество секунд буфера, то обновление будет перемещено вверх, чтобы сохранить как можно большую часть времени буфера. Допустимые значения от 0 до 3600 (1 час). Если значение не указано, используется значение по умолчанию 300 (5 минут). Установленное значение и значение sasl.login.refresh.min.period.seconds игнорируются, если их сумма превышает оставшееся время жизни учётных данных. В настоящее время относится только к OAUTHBEARER. |
| sasl.login.refresh.min.period.seconds | short | 60 | [0,…,900] | low | Желаемое минимальное время ожидания потока обновления входа в систему перед обновлением учётных данных, в секундах. Допустимые значения от 0 до 900 (15 минут). Если значение не указано, используется значение по умолчанию 60 (1 минута). Установленное значение и значение sasl.login.refresh.buffer.seconds игнорируются, если их сумма превышает оставшееся время жизни учётных данных. В настоящее время относится только к OAUTHBEARER. |
| sasl.login.refresh.window.factor | double | 0.8 | [0.5,…,1.0] | low | Поток обновления входа в систему будет находиться в спящем режиме до тех пор, пока не будет достигнут установленный коэффициент окна относительно времени жизни учётных данных, после чего он попытается обновить учётные данные. Допустимые значения от 0.5 (50%) до 1.0 (100%) включительно. Если значение не указано, используется значение по умолчанию 0.8 (80%). В настоящее время относится только к OAUTHBEARER. |
| sasl.login.refresh.window.jitter | double | 0.05 | [0.0,…,0.25] | low | Максимальное количество случайного джиттера относительно времени жизни учётных данных, которое добавляется ко времени ожидания потока обновления входа в систему. Допустимые значения от 0 до 0.25 (25%) включительно. Если значение не указано, используется значение по умолчанию 0,05 (5%). В настоящее время относится только к OAUTHBEARER. |
| sasl.mechanism | string | GSSAPI | — | medium | SASL-механизм, используемый для клиентских подключений. Это может быть любой механизм, для которого доступен поставщик безопасности. GSSAPI — значение по умолчанию. |
| security.protocol | string | PLAINTEXT | — | medium | Протокол, используемый для взаимодействия с брокерами. Допустимые значения: PLAINTEXT, SSL, SASL\_PLAINTEXT, SASL\_SSL. |
| security.providers | string | null | — | low | Перечень настраиваемых классов создателей, каждый из которых возвращает поставщика, имплементирующего алгоритмы безопасности. Установленный классы должны имплементировать интерфейс org.apache.kafka.common.security.auth.SecurityProviderCreator. |
| send.buffer.bytes | int | 131072 (128 КиБ) | [-1,…] | medium | Размер буфера отправки TCP (SO\_SNDBUF), используемого при отправке данных. Если установлено значение -1, будет использоваться значение по умолчанию ОС. |
| socket.connection.setup.timeout.max.ms | long | 30000 (30 секунд) | — | medium | Максимальное время, в течение которого клиент будет ожидать установления соединения с сокетом. Тайм-аут установки соединения будет экспоненциально увеличиваться при каждом последующем сбое соединения до установленного максимума. Чтобы избежать штормов соединений, к тайм-ауту будет применён коэффициент рандомизации 0.2, в результате чего получится случайный диапазон от 20% ниже до 20% выше вычисленного значения. |
| socket.connection.setup.timeout.ms | long | 10000 (10 секунд) | — | medium | Время, в течение которого клиент будет ожидать установления соединения с сокетом. Если соединение не будет установлено до истечения тайм-аута, клиенты закроют канал сокета. |
| ssl.cipher.suites | list | null | — | low | Перечень комплектов шифров. Это именованная комбинация алгоритма аутентификации, шифрования, MAC и обмена ключами, используемая для согласования настроек безопасности для сетевого подключения с использованием сетевого протокола TLS или SSL. По умолчанию поддерживаются все доступные наборы шифров. |
| ssl.enabled.protocols | list | TLSv1.2 | — | medium | Перечень протоколов, разрешённых для SSL-соединений. По умолчанию используется TLSv1.2, TLSv1.3 при работе с Java 11 или новее, в противном случае — TLSv1.2. Со значением по умолчанию для Java 11 клиенты и серверы предпочтут TLSv1.3, если оба поддерживают его, и откатятся к TLSv1.2 в противном случае (при условии, что оба поддерживают как минимум TLSv1.2). Это значение по умолчанию подходит для большинства случаев. См. также информацию по конфигурации ssl.protocol. |
| ssl.endpoint.identification.algorithm | string | https | — | low | Алгоритм идентификации конечной точки для валидации имени хоста сервера с помощью сертификата сервера. |
| ssl.engine.factory.class | class | null | — | low | Класс типа org.apache.kafka.common.security.auth.SslEngineFactory для предоставления объектов SSLEngine. Значение по умолчанию org.apache.kafka.common.security.ssl.DefaultSslEngineFactory. |
| ssl.key.password | password | null | — | high | Алгоритм идентификации конечной точки для валидации имени хоста сервера с помощью сертификата сервера. |
| ssl.keymanager.algorithm | string | SunX509 | — | low | Алгоритм, используемый службой управления ключами для SSL-соединений. Значение по умолчанию — это алгоритм службы, настроенный Java Virtual Machine. |
| ssl.keystore.certificate.chain | password | null | — | high | Цепочка сертификатов в формате, указанном свойством ssl.keystore.type. По умолчанию служба SSL-механизма поддерживает только формат PEM с перечнем сертификатов X.509. |
| ssl.keystore.key | password | null | — | high | Закрытый ключ в формате, указанном свойством ssl.keystore.type. По умолчанию служба SSL-механизма поддерживает только формат PEM с ключами PKCS#8. Если ключ зашифрован, пароль ключа должен быть указан с помощью ssl.key.password. |
| ssl.keystore.location | string | null | — | high | Расположение файла хранилища ключей. Свойство опционально для клиента, и может использоваться для двусторонней аутентификации клиента. |
| ssl.keystore.password | password | null | — | high | Пароль хранилища для файла хранилища ключей. Свойство опционально для клиента, и требуется только в том случае, если настроено свойство ssl.keystore.location. Пароль хранилища ключей не поддерживается для формата PEM. |
| ssl.keystore.type | string | JKS | — | medium | Формат файла хранилища ключей. Свойство опционально для клиента. |
| ssl.protocol | string | TLSv1.2 | — | medium | SSL-протокол, используемый для генерации SSLContext. По умолчанию используется значение TLSv1.3 при работе с Java 11 или новее, в противном случае — TLSv1.2. Данное значение подходит для большинства случаев использования. Допустимые значения в последних JVM: TLSv1.2 и TLSv1.3. TLS, TLSv1.1, SSL, SSLv2 и SSLv3 могут поддерживаться в старых JVM, но их использование не рекомендуется из-за известных уязвимостей безопасности. Со значением по умолчанию для данной конфигурации и свойства ssl.enabled.protocols клиенты будут переходить на TLSv1.2, если сервер не поддерживает TLSv1.3. Если для данной конфигурации установлено значение TLSv1.2, клиенты не будут использовать TLSv1.3, даже если оно одно из значений в свойстве ssl.enabled.protocols, а сервер поддерживает только TLSv1.3. |
| ssl.provider | string | null | — | medium | Имя поставщика безопасности, используемого для SSL-соединений. Значение по умолчанию — это поставщик безопасности для JVM по умолчанию. |
| ssl.secure.random.implementation | string | null | — | low | Имплементация SecureRandom PRNG, используемая для операций шифрования SSL. |
| ssl.trustmanager.algorithm | string | PKIX | — | low | Алгоритм доверенной службы управления ключами для SSL-соединений. Значение по умолчанию — это алгоритм службы, настроенный для Java Virtual Machine. |
| ssl.truststore.certificates | password | null | — | high | Доверенные сертификаты в формате, указанном свойством ssl.truststore.type. Служба SSL-механизма по умолчанию поддерживает только формат PEM с сертификатами X.509. |
| ssl.truststore.location | string | null | — | high | Расположение файла хранилища доверенных сертификатов. |
| ssl.truststore.password | password | null | — | high | Пароль для файла доверенного хранилища. Если пароль не установлен, настроенный файл хранилища доверенных сертификатов по-прежнему будет использоваться, но проверка целостности будет отключена. Пароль доверенного хранилища не поддерживается для формата PEM. |
| ssl.truststore.type | string | JKS | — | medium | Формат файла хранилища доверенных сертификатов. |
| transaction.timeout.ms | int | 60000 (1 минута) | — | low | Максимальное время в мс, в течение которого координатор транзакции будет ожидать обновления статуса транзакции от поставщика перед тем, как упреждающе прервать текущую транзакцию. Если это значение больше, чем параметр transaction.max.timeout.ms в брокере, запрос упадёт с ошибкой InvalidTxnTimeoutException. |
| transactional.id | string | null | non-empty string | low | TransactionalId, используемый для транзакционной доставки. Это обеспечивает семантику надежности, которая охватывает несколько сеансов поставщиков, поскольку позволяет клиенту гарантировать, что транзакции, использующие один и тот же TransactionalId, были завершены до начала любых новых транзакций. Если TransactionalId не указан, то поставщик ограничен идемпотентной доставкой. Если TransactionalId настроен, подразумевается установка значения для enable.idempotence. По умолчанию TransactionId не настроен, что означает, что транзакции использовать нельзя. Обратите внимание, что по умолчанию для транзакций требуется кластер как минимум из трех брокеров, что является рекомендуемой настройкой для производственной среды; для разработки вы можете изменить это, настроив конфигурацию брокера transaction.state.log.replication.factor. |
| value.serializer | class | — | — | high | Класс сериализатора (Serializer) для значения, который имплементирует интерфейс org.apache.kafka.common.serialization.Serializer. |

Конфигурации потребителя

Конфигурации consumer (потребителя) и их значения по умолчанию представлены в таблице ниже в алфавитном порядке по наименованию конфигурации.

Таблица — Конфигурации потребителя

| Конфигурация | Тип | Значение по умолчанию | Действительные значения | Важность | Описание |
| --- | --- | --- | --- | --- | --- |
| allow.auto.create.topics | boolean | true | — | high | Разрешает автоматическое создание топика на брокере при подписке или назначении топика. Топик, на который подписывается, будет автоматически создан, только если брокер разрешит это, используя конфигурацию брокера auto.create.topics.enable. Для данной конфигурации необходимо установить значение false при использовании брокеров старше 0.11.0. |
| auto.commit.interval.ms | int | 5000 (5 секунд) | [0,…] | low | Частота в миллисекундах, с которой потребительские смещения автоматически коммитятся в Kafka, если для параметра enable.auto.commit установлено значение true. |
| auto.offset.reset | string | latest | [latest, earliest, none] | medium | Что делать, если в Kafka нет начального смещения или если текущее смещение больше не существует на сервере (например, потому что эти данные были удалены):  earliest — автоматически сбрасывает смещение до самого раннего смещения;  latest — автоматически сбрасывает смещение до последнего смещения;  none — генерировать исключение для потребителя, если для группы потребителя не найдено предыдущее смещение;  anything else — сбросить исключение для потребителя. |
| bootstrap.servers | list | "" | non-null string | high | Перечень пар хост/порт, используемых для установления начального подключения к кластеру Kafka. Клиент будет использовать все серверы независимо от того, какие серверы указаны здесь для начальной подключения, — данный перечень влияет только на инициирующие хосты, используемые для обнаружения полного набора серверов. Данный перечень должен быть в формате: host1:port1,host2:port2,…. Поскольку эти серверы используются только для начального соединения, чтобы обнаружить полное членство в кластере (которое может динамически меняться), данный перечень не обязательно должен содержать полный набор серверов (может потребоваться больше одного сервера, если какой-либо сервер выходит из строя). |
| check.crcs | boolean | true | — | low | Автоматическая проверка CRC32 считываемых записей. Установка параметра гарантирует, что сообщения не будут повреждены по сети или на диске. Данная проверка добавляет некоторые накладные расходы, поэтому её можно отключить в случаях, когда требуется максимальная производительность. |
| client.dns.lookup | string | use\_all\_dns\_ips | [default, use\_all\_dns\_ips, resolve\_canonical\_bootstrap\_servers\_only] | medium | Параметр управляет тем, как клиент использует поиск DNS. Если установлено значение use\_all\_dns\_ips, клиент будеет последовательно подключаться к каждому возвращённому IP-адресу до тех пор, пока не будет установлено успешное соединение. После отключения используется следующий IP-адрес. После того, как все IP-адреса были использованы один раз, клиент снова разрешает IP-адреса из имени хоста (поиск DNS-имени как JVM, так и кэша ОС).  Если установлено значение resolve\_canonical\_bootstrap\_servers\_only, разрешается каждый адрес начальной загрузки в списке канонических имён. После фазы начальной загрузки поиск DNS осуществляется так, как при установленном значении use\_all\_dns\_ips.  Если установлено значение default (не рекомендуется), осуществляется попытка подключиться к первому IP-адресу, возвращённому поиском, даже если поиск вернёт несколько IP-адресов. |
| client.id | string | "" | — | low | Строка идентификатора, передаваемая на сервер при выполнении запросов. Это делается для того, чтобы иметь возможность отслеживать источник запросов помимо IP-адреса/порта, позволяя включать логическое имя приложения в лог запросов на стороне сервера. |
| client.rack | string | "" | — | low | Идентификатор стойки для установленного клиента. Может быть указано любое строковое значение, указывающее, где физически находится этот клиент. Поддерживается вместе с конфигурацией брокера broker.rack. |
| connections.max.idle.ms | long | 540000 (9 минут) | — | medium | Неактивные подключения закрываются по истечении количества миллисекунд, указанного в данной конфигурации. |
| default.api.timeout.ms | int | 60000 (1 минута) | [0,…] | medium | Параметр задаёт тайм-аут (в миллисекундах) для клиентских API. Данная конфигурация используется в качестве тайм-аута по умолчанию для всех клиентских операций, для которых не указан параметр timeout. |
| enable.auto.commit | boolean | true | — | medium | Если установлено значение true, смещение потребителя будет периодически фиксироваться в фоновом режиме. |
| exclude.internal.topics | boolean | true | — | medium | Исключение из подписки внутренних топиков, соответствующих шаблону подписки. В таком случае необходимо явно подписаться на внутренний топик. |
| fetch.max.bytes | int | 52428800 (50 МиБ) | [0,…] | medium | Максимальный объём данных, который сервер должен вернуть для запроса на выборку. Записи выбираются в пакетах потребителем, и, если первый пакет записей в первой непустой партиции выборки больше, чем установленное значение, пакет записей всё равно будет возвращён, чтобы гарантировать, что потребитель может продолжить работу. Таким образом, установленное значение не является абсолютным максимумом. Максимальный размер пакета записей, принимаемый брокером, определяется через свойства message.max.bytes (конфигурация брокера) или max.message.bytes (конфигурация топика). Обратите внимание, что потребитель выполняет несколько выборок параллельно. |
| fetch.max.wait.ms | int | 500 | [0,…] | low | Максимальная продолжительность времени, в течение которой сервер будет блокировать ответ на запрос выборки, если данных недостаточно для немедленного удовлетворения требований, заданных свойством fetch.min.bytes. |
| fetch.min.bytes | int | 1 | [0,…] | high | Минимальное количество байт, ожидаемых для каждого ответа на запрос на выборку. Если данных недостаточно, запрос будет ждать накопления достаточного количества данных, прежде чем ответить на запрос. Значение по умолчанию, равное 1 байту, означает, что на запросы выборки ответят, как только станет доступен единственный байт данных или истечёт время ожидания запроса на получение данных. Установка данного параметра на значение больше 1 приведёт к тому, что сервер будет ждать накопления больших объёмов данных, что может немного улучшить пропускную способность сервера за счёт некоторой дополнительной задержки. |
| group.id | string | null | — | high | Уникальная строка, которая идентифицирует группу потребителей, к которой принадлежит этот потребитель. Данное свойство требуется, если потребитель использует либо функцию управления группами с помощью subscribe(topic), либо стратегию управления смещением на основе Kafka. |
| group.instance.id | string | null | — | medium | Уникальный идентификатор инстанса потребителя, предоставленный конечным пользователем. Разрешены только непустые строки. Если значение установлено, потребитель рассматривается как статический член, что означает, что только один инстанс с данным идентификатором разрешён в группе потребителей в любое время. Данное свойство можно использовать в сочетании с увеличенным тайм-аутом сеанса, чтобы избежать перебалансировки групп, вызванной временной недоступностью (например, перезапусками процесса). Если значение не установлено, потребитель присоединится к группе как динамический член, что является традиционным поведением. |
| heartbeat.interval.ms | int | 3000 (3 секунды) | — | high | Ожидаемое время между heartbeat-сообщениями для координатора потребителей при использовании средств управления группами Kafka. Heartbeat-сообщения используются для обеспечения того, чтобы сеанс потребителя оставался активным, и для облегчения перебалансировки, когда новые потребители присоединяются или покидают группу. Значение должно быть меньше, чем session.timeout.ms, но обычно не должно превышать 1/3 установленного для свойства значения. Его можно настроить еще ниже, чтобы контролировать ожидаемое время для нормальной перебалансировки. |
| interceptor.classes | list | "" | non-null string | low | Перечень классов для использования в качестве перехватчиков (interceptors). Имплементация интерфейса org.apache.kafka.clients.producer.ProducerInterceptor позволяет перехватывать (и, возможно, изменять) записи, полученные поставщиком, до их публикации в кластере Kafka. По умолчанию установлено отсутствие перехватчиков. |
| isolation.level | string | read\_uncommitted | [read\_committed, read\_uncommitted] | medium | Управляет чтением сообщений, записанных транзакционно. Если установлено значение read\_committed, метод consumer.poll() будет возвращать только зафиксированные транзакционные сообщения. Если установлено значение read\_uncommitted (по умолчанию), метод consumer.poll() будет возвращать все сообщения, даже транзакционные сообщения, которые были прерваны. Нетранзакционные сообщения будут возвращены безоговорочно в любом режиме.  Сообщения всегда будут возвращаться в порядке смещения. Следовательно, в режиме read\_committed, consumer.poll() будет возвращать сообщения только до последнего стабильного смещения (last stable offset, LSO), которое меньше смещения первой открытой транзакции. В частности, любые сообщения, появляющиеся после сообщений, относящихся к текущим транзакциям, будут удерживаться до тех пор, пока соответствующая транзакция не будет завершена. В результате потребители read\_committed не смогут считывать до верхнего водяного знака, когда есть транзакции in-flight.  Кроме того, в режиме read\_committed метод seekToEnd вернёт LSO. |
| key.deserializer | class | — | — | high | Класс десериализатора (Deserializer) для ключа, который имплементирует интерфейс org.apache.kafka.common.serialization.Deserializer. |
| max.partition.fetch.bytes | int | 1048576 (1 МиБ) | [0,…] | high | Максимальный объём данных для каждой партиции, который вернёт сервер. Записи загружаются потребителем пакетами. Если первый пакет записей в первой непустой партиции выборки превышает установленный предел, пакет всё равно будет возвращён, чтобы гарантировать, что потребитель может продолжить работу. Максимальный размер пакета записей, принимаемый брокером, определяется через message.max.bytes (конфигурация брокера) или max.message.bytes (конфигурация топика). См. fetch.max.bytes для ограничения размера запроса потребителя. |
| max.poll.interval.ms | int | 300000 (5 минут) | [1,…] | medium | Максимальная задержка между вызовами метода poll() при использовании управления группами потребителей. Свойство устанавливает верхнюю границу времени, в течение которого потребитель может бездействовать, прежде чем выбрать дополнительные записи. Если poll() не вызывается до истечения этого тайм-аута, то потребитель считается упавшим, и группа будет перебалансирована, чтобы переназначить партиции другому члену. Для потребителей, использующих ненулевое значение свойства group.instance.id, который достигает данного тайм-аута, партиции не будут немедленно переназначены. Вместо этого потребитель перестанет отправлять heartbeat-сообщения, и партиции будут переназначены после истечения срока, установленного в свойстве session.timeout.ms. Данная ситуация отражает поведение отключённого статического потребителя. |
| max.poll.records | int | 500 | [1,…] | medium | Максимальное количество записей, возвращаемых за один вызов метода poll(). Обратите внимание, что max.poll.records не влияет на базовое поведение выборки. Потребитель будет кэшировать записи из каждого запроса выборки и возвращать их постепенно из каждого опроса. |
| metadata.max.age.ms | long | 300000 (5 минут) | [0,…] | low | Период времени в миллисекундах, по истечении которого метаданные будут принудительно обновлены, даже если отсутствуют какие-либо изменения в лидере партиции, чтобы упреждающе обнаруживать новых брокеров или партиции. |
| metric.reporters | list | "" | non-null string | low | Перечень классов для использования в качестве репортеров метрик (metrics reporters). Имплементация интерфейса org.apache.kafka.common.metrics.MetricsReporter позволяет подключать классы, которые будут получать уведомления о создании новой метрики. JmxReporter всегда включён для регистрации статистики JMX. |
| metrics.num.samples | int | 2 | [1,…] | low | Количество образцов, поддерживаемых для вычисления метрик. |
| metrics.recording.level | string | INFO | [INFO, DEBUG, TRACE] | low | Самый высокий уровень записи метрик. |
| metrics.sample.window.ms | long | 30000 (30 секунд) | [0,…] | low | Временной промежуток, в течение которого вычисляется образец метрики. |
| partition.assignment.strategy | list | class org.apache.kafka.clients.consumer.RangeAssignor | non-null string | medium | Перечень имён классов или типов классов, упорядоченных по предпочтению, поддерживаемых стратегий назначения партиций, которые клиент будет использовать для распределения владения партициями между инстансами потребителя при использовании управления группами.  Доступные варианты:  org.apache.kafka.clients.consumer.RangeAssignor — распределение по умолчанию, который работает на уровне каждого топика;  org.apache.kafka.clients.consumer.RoundRobinAssignor — назначает партиции потребителям циклически;  org.apache.kafka.clients.consumer.StickyAssignor — гарантирует максимально сбалансированное назначение, сохраняя при этом как можно больше существующих назначений партиций;  org.apache.kafka.clients.consumer.CooperativeStickyAssignor — следует той же логике StickyAssignor, но разрешает совместную перебалансировку;  Имплементацияя интерфейса org.apache.kafka.clients.consumer.ConsumerPartitionAssignor позволяет подключать кастомную стратегию назначения. |
| receive.buffer.bytes | int | 65536 (64 КиБ) | [-1,…] | medium | Размер приёмного буфера TCP (SO\_RCVBUF) для использования при чтении данных. Если установлено значение -1, будет использоваться значение по умолчанию ОС. |
| reconnect.backoff.max.ms | long | 1000 (1 секунда) | [0,…] | low | Максимальное время ожидания в миллисекундах при повторном подключении к брокеру, к которому неоднократно не удавалось подключиться. Если значение установлено, отсрочка для каждого хоста будет экспоненциально увеличиваться при каждом последующем сбое соединения, вплоть до установленного максимума. После расчёта увеличения отсрочки добавляется 20% случайного джиттера, чтобы избежать штормов соединения. |
| reconnect.backoff.ms | long | 50 | [0,…] | low | Базовое время ожидания перед попыткой повторного подключения к данному хосту. Это позволяет избежать многократного подключения к хосту в замкнутом цикле. Установленная отсрочка применяется ко всем попыткам подключения клиента к брокеру. |
| request.timeout.ms | int | 30000 (30 секунд) | [0,…] | medium | Конфигурация контролирует максимальное количество времени, в течение которого клиент будет ждать ответа на запрос. Если ответ не будет получен до истечения тайм-аута, клиент повторно отправит запрос, если необходимо, или запрос завершится неудачей, если количество повторных попыток исчерпано. |
| retry.backoff.ms | long | 100 | [0,…] | low | Время ожидания перед попыткой повторить неудавшийся запрос к данной партиции топика. Это позволяет избежать повторной отправки запросов в замкнутом цикле при некоторых сценариях сбоя. |
| sasl.client.callback.handler.class | class | null | — | medium | Полное имя класса обработчика обратного вызова SASL-клиента, имплементирующего интерфейс AuthenticateCallbackHandler. |
| sasl.jaas.config | password | null | — | medium | Параметры контекста входа JAAS для SASL-соединений в формате, используемом файлами конфигурации JAAS. [Здесь](http://docs.oracle.com/javase/8/docs/technotes/guides/security/jgss/tutorials/LoginConfigFile.html) описан формат файла конфигурации JAAS. Формат для значения: loginModuleClass controlFlag (optionName=optionValue)\*;. Для брокеров конфигурация должна начинаться с префикса слушателя и имени SASL-механизма в нижнем регистре. Например, listener.name.sasl\_ssl.scram-sha-256.sasl.jaas.config=com.example.ScramLoginModule required; |
| sasl.kerberos.kinit.cmd | string | /usr/bin/kinit | — | low | Путь к команде Kerberos kinit. |
| sasl.kerberos.min.time.before.relogin | long | 60000 | — | low | Время ожидания авторизации потока между попытками обновления. |
| sasl.kerberos.service.name | string | null | — | medium | Имя принципала Kerberos, под которым работает Kafka. Имя можно определить либо в конфигурации Kafka JAAS, либо в конфигурации Kafka. |
| sasl.kerberos.ticket.renew.jitter | double | 0.05 | — | low | Процент случайного джиттера, добавленный ко времени возобновления. |
| sasl.kerberos.ticket.renew.window.factor | double | 0.8 | — | low | Поток входа в систему будет находиться в спящем режиме до тех пор, пока не будет достигнут установленный коэффициент окна времени с момента последнего обновления до истечения срока действия тикета, после чего он попытается обновить тикет. |
| sasl.login.callback.handler.class | class | null | — | medium | Полное имя класса обработчика обратного вызова входа SASL, имплементирующего интерфейс AuthenticateCallbackHandler. Для брокеров конфигурация обработчика обратного вызова при входе в систему должна иметь префикс с префиксом слушателя и именем SASL-механизма в нижнем регистре. Например, listener.name.sasl\_ssl.scram-sha-256.sasl.login.callback.handler.class = com.example.CustomScramLoginCallbackHandler. |
| sasl.login.class | class | null | — | medium | Полное имя класса, имплементирующего интерфейс входа в систему. Для брокеров конфигурация входа в систему должна начинаться с префикса слушателя и имени SASL-механизма в нижнем регистре. Например, listener.name.sasl\_ssl.scram-sha-256.sasl.login.class = com.example.CustomScramLogin. |
| sasl.login.refresh.buffer.seconds | short | 300 | [0,…,3600] | low | Время буферизации до истечения срока действия учётных данных, которое необходимо поддерживать при обновлении учётных данных, в секундах. Если в противном случае обновление произошло бы ближе к истечению срока действия, чем количество секунд буфера, то обновление будет перемещено вверх, чтобы сохранить как можно большую часть времени буфера. Допустимые значения от 0 до 3600 (1 час). Если значение не указано, используется значение по умолчанию 300 (5 минут). Установленное значение и значение sasl.login.refresh.min.period.seconds игнорируются, если их сумма превышает оставшееся время жизни учётных данных. В настоящее время относится только к OAUTHBEARER. |
| sasl.login.refresh.min.period.seconds | short | 60 | [0,…,900] | low | Желаемое минимальное время ожидания потока обновления входа в систему перед обновлением учётных данных, в секундах. Допустимые значения от 0 до 900 (15 минут). Если значение не указано, используется значение по умолчанию 60 (1 минута). Установленное значение и значение sasl.login.refresh.buffer.seconds игнорируются, если их сумма превышает оставшееся время жизни учётных данных. В настоящее время относится только к OAUTHBEARER. |
| sasl.login.refresh.window.factor | double | 0.8 | [0.5,…,1.0] | low | Поток обновления входа в систему будет находиться в спящем режиме до тех пор, пока не будет достигнут установленный коэффициент окна относительно времени жизни учётных данных, после чего он попытается обновить учётные данные. Допустимые значения от 0.5 (50%) до 1.0 (100%) включительно. Если значение не указано, используется значение по умолчанию 0.8 (80%). В настоящее время относится только к OAUTHBEARER. |
| sasl.login.refresh.window.jitter | double | 0.05 | [0.0,…,0.25] | low | Максимальное количество случайного джиттера относительно времени жизни учётных данных, которое добавляется ко времени ожидания потока обновления входа в систему. Допустимые значения от 0 до 0.25 (25%) включительно. Если значение не указано, используется значение по умолчанию 0,05 (5%). В настоящее время относится только к OAUTHBEARER. |
| sasl.mechanism | string | GSSAPI | — | medium | SASL-механизм, используемый для клиентских подключений. Это может быть любой механизм, для которого доступен поставщик безопасности. GSSAPI — значение по умолчанию. |
| security.protocol | string | PLAINTEXT | — | medium | Протокол, используемый для взаимодействия с брокерами. Допустимые значения: PLAINTEXT, SSL, SASL\_PLAINTEXT, SASL\_SSL. |
| security.providers | string | null | — | low | Перечень настраиваемых классов создателей, каждый из которых возвращает поставщика, имплементирующего алгоритмы безопасности. Установленный классы должны имплементировать интерфейс org.apache.kafka.common.security.auth.SecurityProviderCreator. |
| send.buffer.bytes | int | 131072 (128 КиБ) | [-1,…] | medium | Размер буфера отправки TCP (SO\_SNDBUF), используемого при отправке данных. Если установлено значение -1, будет использоваться значение по умолчанию ОС. |
| session.timeout.ms | int | 10000 (10 секунд) | — | high | Тайм-аут, используемый для обнаружения сбоев клиентов при использовании средства управления группами Kafka. Клиент отправляет периодические heartbeat-сообщения, чтобы показать брокеру свою работоспособность. Если до истечения тайм-аута сеанса брокер не получит heartbeat-сообщений, он удалит этого клиента из группы и инициирует перебалансировку. Обратите внимание, что значение должно быть в допустимом диапазоне, заданном в конфигурации брокера с помощью свойств group.min.session.timeout.ms и group.max.session.timeout.ms. |
| socket.connection.setup.timeout.max.ms | long | 30000 (30 секунд) | — | medium | Максимальное время, в течение которого клиент будет ожидать установления соединения с сокетом. Тайм-аут установки соединения будет экспоненциально увеличиваться при каждом последующем сбое соединения до установленного максимума. Чтобы избежать штормов соединений, к тайм-ауту будет применён коэффициент рандомизации 0.2, в результате чего получится случайный диапазон от 20% ниже до 20% выше вычисленного значения. |
| socket.connection.setup.timeout.ms | long | 10000 (10 секунд) | — | medium | Время, в течение которого клиент будет ожидать установления соединения с сокетом. Если соединение не будет установлено до истечения тайм-аута, клиенты закроют канал сокета. |
| ssl.cipher.suites | list | null | — | low | Перечень комплектов шифров. Это именованная комбинация алгоритма аутентификации, шифрования, MAC и обмена ключами, используемая для согласования настроек безопасности для сетевого подключения с использованием сетевого протокола TLS или SSL. По умолчанию поддерживаются все доступные наборы шифров. |
| ssl.enabled.protocols | list | TLSv1.2 | — | medium | Перечень протоколов, разрешённых для SSL-соединений. По умолчанию используется TLSv1.2, TLSv1.3 при работе с Java 11 или новее, в противном случае — TLSv1.2. Со значением по умолчанию для Java 11 клиенты и серверы предпочтут TLSv1.3, если оба поддерживают его, и откатятся к TLSv1.2 в противном случае (при условии, что оба поддерживают как минимум TLSv1.2). Это значение по умолчанию подходит для большинства случаев. См. также информацию по конфигурации ssl.protocol. |
| ssl.endpoint.identification.algorithm | string | https | — | low | Алгоритм идентификации конечной точки для валидации имени хоста сервера с помощью сертификата сервера. |
| ssl.engine.factory.class | class | null | — | low | Класс типа org.apache.kafka.common.security.auth.SslEngineFactory для предоставления объектов SSLEngine. Значение по умолчанию org.apache.kafka.common.security.ssl.DefaultSslEngineFactory. |
| ssl.key.password | password | null | — | high | Пароль закрытого ключа в файле хранилища ключей или ключ PEM, указанный в ssl.keystore.key. Свойство требуется для клиентов, только если настроена двусторонняя аутентификация. |
| ssl.keymanager.algorithm | string | SunX509 | — | low | Алгоритм, используемый службой управления ключами для SSL-соединений. Значение по умолчанию — это алгоритм службы, настроенный Java Virtual Machine. |
| ssl.keystore.certificate.chain | password | null | — | high | Цепочка сертификатов в формате, указанном свойством ssl.keystore.type. По умолчанию служба SSL-механизма поддерживает только формат PEM с перечнем сертификатов X.509. |
| ssl.keystore.key | password | null | — | high | Закрытый ключ в формате, указанном свойством ssl.keystore.type. По умолчанию служба SSL-механизма поддерживает только формат PEM с ключами PKCS#8. Если ключ зашифрован, пароль ключа должен быть указан с помощью ssl.key.password. |
| ssl.keystore.location | string | null | — | high | Расположение файла хранилища ключей. Свойство опционально для клиента, и может использоваться для двусторонней аутентификации клиента. |
| ssl.keystore.password | password | null | — | high | Пароль хранилища для файла хранилища ключей. Свойство опционально для клиента, и требуется только в том случае, если настроено свойство ssl.keystore.location. Пароль хранилища ключей не поддерживается для формата PEM. |
| ssl.keystore.type | string | JKS | — | medium | Формат файла хранилища ключей. Свойство опционально для клиента. |
| ssl.protocol | string | TLSv1.2 | — | medium | SSL-протокол, используемый для генерации SSLContext. По умолчанию используется значение TLSv1.3 при работе с Java 11 или новее, в противном случае — TLSv1.2. Данное значение подходит для большинства случаев использования. Допустимые значения в последних JVM: TLSv1.2 и TLSv1.3. TLS, TLSv1.1, SSL, SSLv2 и SSLv3 могут поддерживаться в старых JVM, но их использование не рекомендуется из-за известных уязвимостей безопасности. Со значением по умолчанию для данной конфигурации и свойства ssl.enabled.protocols клиенты будут переходить на TLSv1.2, если сервер не поддерживает TLSv1.3. Если для данной конфигурации установлено значение TLSv1.2, клиенты не будут использовать TLSv1.3, даже если оно одно из значений в свойстве ssl.enabled.protocols, а сервер поддерживает только TLSv1.3. |
| ssl.provider | string | null | — | medium | Имя поставщика безопасности, используемого для SSL-соединений. Значение по умолчанию — это поставщик безопасности для JVM по умолчанию. |
| ssl.secure.random.implementation | string | null | — | low | Имплементация SecureRandom PRNG, используемая для операций шифрования SSL. |
| ssl.trustmanager.algorithm | string | PKIX | — | low | Алгоритм доверенной службы управления ключами для SSL-соединений. Значение по умолчанию — это алгоритм службы, настроенный для Java Virtual Machine. |
| ssl.truststore.certificates | password | null | — | high | Доверенные сертификаты в формате, указанном свойством ssl.truststore.type. Служба SSL-механизма по умолчанию поддерживает только формат PEM с сертификатами X.509. |
| ssl.truststore.location | string | null | — | high | Расположение файла хранилища доверенных сертификатов. |
| ssl.truststore.password | password | null | — | high | Пароль для файла доверенного хранилища. Если пароль не установлен, настроенный файл хранилища доверенных сертификатов по-прежнему будет использоваться, но проверка целостности будет отключена. Пароль доверенного хранилища не поддерживается для формата PEM. |
| ssl.truststore.type | string | JKS | — | medium | Формат файла хранилища доверенных сертификатов. |
| value.deserializer | class | — | — | high | Класс десериализатора (Deserializer) для значения, который имплементирует интерфейс org.apache.kafka.common.serialization.Deserializer. |

Конфигурации подключения к Kafka

Конфигурации фреймворка Kafka Connect (подключения к Kafka) и их значения по умолчанию представлены в таблице ниже в алфавитном порядке по наименованию конфигурации.

Таблица — Конфигурации подключения к Kafka

| Конфигурация | Тип | Значение по умолчанию | Действительные значения | Важность | Описание |
| --- | --- | --- | --- | --- | --- |
| access.control.allow.methods | string | "" | — | low | Конфигурация устанавливает методы, поддерживаемые для запросов из разных источников, задав заголовок Access-Control-Allow-Methods. Значение по умолчанию заголовка Access-Control-Allow-Methods разрешает запросы из разных источников для GET, POST и HEAD. |
| access.control.allow.origin | string | "" | — | low | Значение для установки заголовка Access-Control-Allow-Origin для запросов REST API. Чтобы включить доступ между источниками, установите значение домена приложения, которому должен быть разрешен доступ к API, или значение \*, чтобы разрешить доступ из любого домена. Значение по умолчанию разрешает доступ только из домена по REST API. |
| admin.listeners | list | null | org.apache.kafka.connect.runtime.WorkerConfig$AdminListenersValidator@383534aa | low | Перечень URI, разделённых запятыми, которые будет прослушивать Admin REST API. Поддерживаемые протоколы — HTTP и HTTPS. Пустая или незаполненная строка в значении отключит данную функцию. По умолчанию используется обычный слушатель (заданный свойством listeners). |
| bootstrap.servers | list | localhost:9092 | — | high | Перечень пар хост/порт, используемых для установления начального подключения к кластеру Kafka. Клиент будет использовать все серверы независимо от того, какие серверы указаны здесь для начальной подключения, — данный перечень влияет только на инициирующие хосты, используемые для обнаружения полного набора серверов. Данный перечень должен быть в формате: host1:port1,host2:port2,…. Поскольку эти серверы используются только для начального соединения, чтобы обнаружить полное членство в кластере (которое может динамически меняться), данный перечень не обязательно должен содержать полный набор серверов (может потребоваться больше одного сервера, если какой-либо сервер выходит из строя). |
| client.dns.lookup | string | use\_all\_dns\_ips | [default, use\_all\_dns\_ips, resolve\_canonical\_bootstrap\_servers\_only] | medium | Параметр управляет тем, как клиент использует поиск DNS. Если установлено значение use\_all\_dns\_ips, клиент будеет последовательно подключаться к каждому возвращённому IP-адресу до тех пор, пока не будет установлено успешное соединение. После отключения используется следующий IP-адрес. После того, как все IP-адреса были использованы один раз, клиент снова разрешает IP-адреса из имени хоста (поиск DNS-имени как JVM, так и кэша ОС).  Если установлено значение resolve\_canonical\_bootstrap\_servers\_only, разрешается каждый адрес начальной загрузки в списке канонических имён. После фазы начальной загрузки поиск DNS осуществляется так, как при установленном значении use\_all\_dns\_ips.  Если установлено значение default (не рекомендуется), осуществляется попытка подключиться к первому IP-адресу, возвращённому поиском, даже если поиск вернёт несколько IP-адресов. |
| client.id | string | "" | — | low | Строка идентификатора, передаваемая на сервер при выполнении запросов. Это делается для того, чтобы иметь возможность отслеживать источник запросов помимо IP-адреса/порта, позволяя включать логическое имя приложения в лог запросов на стороне сервера. |
| config.providers | list | "" | — | low | Разделённые запятыми имена классов ConfigProvider, загружаемых и используемых в указанном порядке. Имплементация интерфейса ConfigProvider позволяет заменять ссылки на переменные в конфигурациях коннектора, например, для внешних секретов. |
| config.storage.replication.factor | short | 3 | — | low | Фактор репликации, используемый при создании топика для хранения конфигурации.  В качестве значения может быть положительное число, не превышающее количество брокеров в кластере Kafka, или -1, чтобы использовать брокера по умолчанию. |
| config.storage.topic | string | — | — | high | Название топика Kafka, в котором хранятся конфигурации коннекторов. |
| connect.protocol | string | sessioned | [eager, compatible, sessioned] | low | Режим совместимости для протокола Kafka Connect. |
| connections.max.idle.ms | long | 540000 (9 минут) | — | medium | Неактивные подключения закрываются по истечении количества миллисекунд, указанного в данной конфигурации. |
| connector.client.config.override.policy | string | None | — | medium | Имя класса или алиас имплементации ConnectorClientConfigOverridePolicy. Определяет, какие конфигурации клиентов могут быть отменены коннектором. Имплементация по умолчанию — None. Другие возможные политики в фреймворке включают All и Principal. |
| group.id | string | — | — | high | Уникальная строка, которая идентифицирует группу кластера Connect, к которой принадлежит данный обработчик. |
| header.converter | class | org.apache.kafka.connect.storage.SimpleHeaderConverter | — | low | Класс HeaderConverter, используемый для преобразования между форматом Kafka Connect и сериализованной формой, записанной в Kafka. Параметр контролирует формат значений заголовков в сообщениях, записываемых в Kafka или считываемых из него, и, поскольку он не зависит от коннекторов, он позволяет любому соединителю работать с любым форматом сериализации. Примеры распространённых форматов включают JSON и Avro. По умолчанию SimpleHeaderConverter используется для сериализации значений заголовков в строки и их десериализации путём вывода схем. |
| heartbeat.interval.ms | int | 3000 (3 секунды) | — | high | Ожидаемое время между heartbeat-сообщениями для координатора потребителей при использовании средств управления группами Kafka. Heartbeat-сообщения используются для обеспечения того, чтобы сеанс потребителя оставался активным, и для облегчения перебалансировки, когда новые потребители присоединяются или покидают группу. Значение должно быть меньше, чем session.timeout.ms, но обычно не должно превышать 1/3 установленного для свойства значения. Его можно настроить еще ниже, чтобы контролировать ожидаемое время для нормальной перебалансировки. |
| inter.worker.key.generation.algorithm | string | HmacSHA256 | — | low | Алгоритм, используемый для генерации ключей внутреннего запроса.  В качестве значения может быть указан любой алгоритм KeyGenerator, поддерживаемый обработчиком JVM. |
| inter.worker.key.size | int | null | — | low | Размер ключа, используемого для подписи внутренних запросов, в битах. Если установлено значение null, будет использоваться размер ключа по умолчанию для алгоритма генерации ключа. |
| inter.worker.key.ttl.ms | int | 3600000 (1 час) | [0,…,2147483647] | low | TTL сгенерированных ключей сеанса, используемых для внутренней валидации запроса (в миллисекундах). |
| inter.worker.signature.algorithm | string | HmacSHA256 | — | low | Алгоритм, используемый для подписи внутренних запросов.  В качестве значения может быть установлен любой алгоритм MAC, поддерживаемый обработчиком JVM. |
| inter.worker.verification.algorithms | list | HmacSHA256 | — | low | Перечень разрешённых алгоритмов для верификации внутренних запросов.  В качестве значения может быть установлен перечень из одного или нескольких алгоритмов MAC, каждый из которых поддерживается обработчиком JVM. |
| internal.key.converter | class | org.apache.kafka.connect.json.JsonConverter | — | low | Класс Converter, используемый для преобразования между форматом Kafka Connect и сериализованной формой, записанной в Kafka. Параметр контролирует формат ключей в сообщениях, записываемых в Kafka или считываемых из него, и, поскольку это не зависит от коннекторов, позволяет любому коннектору работать с любым форматом сериализации. Примеры распространённых форматов включают JSON и Avro. Данный параметр контролирует формат, используемый для внутренних данных, используемых фреймворком, таких как конфигурация и смещения, поэтому пользователи обычно могут использовать любую функционирующую имплементацию Converter.  Свойство устарело и будет удалено в следующей версии. |
| internal.value.converter | class | org.apache.kafka.connect.json.JsonConverter | — | low | Класс Converter, используемый для преобразования между форматом Kafka Connect и сериализованной формой, записанной в Kafka. Параметр контролирует формат значений в сообщениях, записываемых в Kafka или считываемых из него, и, поскольку он не зависит от коннекторов, он позволяет любому коннектору работать с любым форматом сериализации. Примеры распространённых форматов включают JSON и Avro. Данный параметр контролирует формат, используемый для внутренних данных, используемых фреймворком, таких как конфигурация и смещения, поэтому пользователи обычно могут использовать любую функционирующую имплементацию Converter.  Свойство устарело и будет удалено в следующей версии. |
| key.converter | class | — | — | high | Класс Converter, используемый для преобразования между форматом Kafka Connect и сериализованной формой, записанной в Kafka. Свойство контролирует формат ключей в сообщениях, записываемых в Kafka или считываемых из него, и, поскольку это не зависит от коннекторов, позволяет любому коннектору работать с любым форматом сериализации. Примеры распространённых форматов включают JSON и Avro. |
| listeners | list | null | — | low | Список разделённых запятыми URI, которые REST API будет прослушивать. Поддерживаемые протоколы — HTTP и HTTPS.  Укажите имя хоста как 0.0.0.0 для привязки ко всем интерфейсам.  Оставьте имя хоста пустым, чтобы привязать его к интерфейсу по умолчанию.  Примеры легальных списков слушателей — HTTP://myhost:8083,HTTPS://myhost:8084. |
| metadata.max.age.ms | long | 300000 (5 минут) | [0,…] | low | Период времени в миллисекундах, по истечении которого метаданные будут принудительно обновлены, даже если отсутствуют какие-либо изменения в лидере партиции, чтобы упреждающе обнаруживать новых брокеров или партиции. |
| metric.reporters | list | "" | — | low | Перечень классов для использования в качестве репортеров метрик (metrics reporters). Имплементация интерфейса org.apache.kafka.common.metrics.MetricsReporter позволяет подключать классы, которые будут получать уведомления о создании новой метрики. JmxReporter всегда включён для регистрации статистики JMX. |
| metrics.num.samples | int | 2 | [1,…] | low | Количество образцов, поддерживаемых для вычисления метрик. |
| metrics.recording.level | string | INFO | [INFO, DEBUG] | low | Самый высокий уровень записи метрик. |
| metrics.sample.window.ms | long | 30000 (30 секунд) | [0,…] | low | Временной промежуток, в течение которого вычисляется образец метрики. |
| offset.flush.interval.ms | long | 60000 (1 минута) | — | low | Интервал, в который осуществляется попытка коммита смещения для тасков. |
| offset.flush.timeout.ms | long | 5000 (5 секунд) | — | low | Максимальное количество миллисекунд ожидания, пока записи будут сброшены и данные смещения партиции будут закоммичены в хранилище смещения, прежде чем отменить процесс и восстановить данные смещения, которые будут закоммичены в будущей попытке. |
| offset.storage.partitions | int | 25 | — | low | Количество партиций, используемых при создании топика смещения хранилища.  В качестве значения может быть установлено положительное число или значение -1, чтобы использовать значение брокера по умолчанию. |
| offset.storage.replication.factor | short | 3 | — | low | Фактор репликации, используемый при создании топика смещения хранилища.  В качестве значения может быть установлено положительное число, не превышающее количество брокеров в кластере Kafka, или значение -1, чтобы использовать значение брокера по умолчанию. |
| offset.storage.topic | string | — | — | high | Название топика Kafka, в котором хранятся коннекторы смещений. |
| plugin.path | list | null | — | low | Перечень путей, разделённых запятыми, содержащих плагины (коннекторы, конвертеры, трансформации). Перечень должен состоять из каталогов верхнего уровня, которые включают любую комбинацию:  каталоги, сразу содержащие jar-файлы с плагинами и их зависимостями;  uber-jar-файлы с плагинами и их зависимостями;  каталоги, непосредственно содержащие структуру каталогов пакетов классов плагинов и их зависимостей.  Примечание: Симлинки будут использоваться для обнаружения зависимостей или плагинов.  Примеры: plugin.path=/usr/local/share/java,/usr/local/share/kafka/plugins,/opt/connectors  Не используйте переменные поставщика конфигурации в этом свойстве, поскольку необработанный путь используется сканером обработчика перед тем, как конфигурации поставщика будут инициализированы и использованы для замены переменных. |
| rebalance.timeout.ms | int | 60000 (1 минута) | — | high | Максимально допустимое время для каждого обработчика, чтобы присоединиться к группе, как только началась перебалансировка. По сути, значение является ограничением на количество времени, необходимое всем таскам для очистки любых ожидающих данных и фиксации смещений. Если тайм-аут превышен, обработчик будет удалён из группы, что вызовет сбои фиксации смещения. |
| receive.buffer.bytes | int | 32768 (32 КиБ) | [0,…] | medium | Размер приёмного буфера TCP (SO\_RCVBUF) для использования при чтении данных. Если установлено значение -1, будет использоваться значение по умолчанию ОС. |
| reconnect.backoff.max.ms | long | 1000 (1 секунда) | [0,…] | low | Максимальное время ожидания в миллисекундах при повторном подключении к брокеру, к которому неоднократно не удавалось подключиться. Если значение установлено, отсрочка для каждого хоста будет экспоненциально увеличиваться при каждом последующем сбое соединения, вплоть до установленного максимума. После расчёта увеличения отсрочки добавляется 20% случайного джиттера, чтобы избежать штормов соединения. |
| reconnect.backoff.ms | long | 50 | [0,…] | low | Базовое время ожидания перед попыткой повторного подключения к данному хосту. Это позволяет избежать многократного подключения к хосту в замкнутом цикле. Установленная отсрочка применяется ко всем попыткам подключения клиента к брокеру. |
| request.timeout.ms | int | 40000 (40 секунд) | [0,…] | medium | Конфигурация контролирует максимальное количество времени, в течение которого клиент будет ждать ответа на запрос. Если ответ не будет получен до истечения тайм-аута, клиент повторно отправит запрос, если необходимо, или запрос завершится неудачей, если количество повторных попыток исчерпано. |
| response.http.headers.config | string | "" | — | low | Правила для заголовков ответа HTTP REST API.  В качестве значения устанавливаются правила заголовков, разделённых запятыми, где каждое правило заголовка имеет формат [action] [header name]:[header value] и может быть заключено в двойные кавычки, если какая-либо часть правила заголовка содержит запятую. |
| rest.advertised.host.name | string | null | — | low | Если значение установлено, указанное в значении имя хоста будет выдано другим обработчикам для подключения. |
| rest.advertised.listener | string | null | — | low | Устанавливает объявленного слушателя (HTTP или HTTPS), который будет предоставлен другим обработчикам для использования. |
| rest.advertised.port | int | null | — | low | Если значение установлено, указанный в значении порт будет выдан другим обработчикам для подключения. |
| rest.extension.classes | list | "" | — | low | Разделённые запятыми имена классов ConnectRestExtension, загружаемых и вызываемых в указанном порядке. Имплементация интерфейса ConnectRestExtension позволяет внедрять в REST API Connect ресурсы, определённые пользователем, например, фильтры. Параметр обычно используется для добавления кастомных функций, таких как логирование, безопасность и прочее. |
| rest.host.name | string | null | — | low | Имя хоста для REST API. Если параметр установлен, он будет привязываться только к указанному интерфейсу. |
| rest.port | int | 8083 | — | low | Порт для REST API для прослушивания. |
| retry.backoff.ms | long | 100 | [0,…] | low | Время ожидания перед попыткой повторить неудавшийся запрос к данной партиции топика. Это позволяет избежать повторной отправки запросов в замкнутом цикле при некоторых сценариях сбоя. |
| sasl.client.callback.handler.class | class | null | — | medium | Полное имя класса обработчика обратного вызова SASL-клиента, имплементирующего интерфейс AuthenticateCallbackHandler. |
| sasl.jaas.config | password | null | — | medium | Параметры контекста входа JAAS для SASL-соединений в формате, используемом файлами конфигурации JAAS. [Здесь](http://docs.oracle.com/javase/8/docs/technotes/guides/security/jgss/tutorials/LoginConfigFile.html) описан формат файла конфигурации JAAS. Формат для значения: loginModuleClass controlFlag (optionName=optionValue)\*;. Для брокеров конфигурация должна начинаться с префикса слушателя и имени SASL-механизма в нижнем регистре. Например, listener.name.sasl\_ssl.scram-sha-256.sasl.jaas.config=com.example.ScramLoginModule required; |
| sasl.kerberos.kinit.cmd | string | /usr/bin/kinit | — | low | Путь к команде Kerberos kinit. |
| sasl.kerberos.service.name | string | null | — | medium | Имя принципала Kerberos, под которым работает Kafka. Имя можно определить либо в конфигурации Kafka JAAS, либо в конфигурации Kafka. |
| sasl.login.callback.handler.class | class | null | — | medium | Полное имя класса обработчика обратного вызова входа SASL, имплементирующего интерфейс AuthenticateCallbackHandler. Для брокеров конфигурация обработчика обратного вызова при входе в систему должна иметь префикс с префиксом слушателя и именем SASL-механизма в нижнем регистре. Например, listener.name.sasl\_ssl.scram-sha-256.sasl.login.callback.handler.class = com.example.CustomScramLoginCallbackHandler. |
| sasl.login.class | class | null | — | medium | Полное имя класса, имплементирующего интерфейс входа в систему. Для брокеров конфигурация входа в систему должна начинаться с префикса слушателя и имени SASL-механизма в нижнем регистре. Например, listener.name.sasl\_ssl.scram-sha-256.sasl.login.class = com.example.CustomScramLogin. |
| sasl.mechanism | string | GSSAPI | — | medium | SASL-механизм, используемый для клиентских подключений. Это может быть любой механизм, для которого доступен поставщик безопасности. GSSAPI — значение по умолчанию. |
| sasl.kerberos.min.time.before.relogin | long | 60000 | — | low | Время ожидания авторизации потока между попытками обновления. |
| sasl.kerberos.ticket.renew.jitter | double | 0.05 | — | low | Процент случайного джиттера, добавленный ко времени возобновления. |
| sasl.kerberos.ticket.renew.window.factor | double | 0.8 | — | low | Поток входа в систему будет находиться в спящем режиме до тех пор, пока не будет достигнут установленный коэффициент окна времени с момента последнего обновления до истечения срока действия тикета, после чего он попытается обновить тикет. |
| sasl.login.refresh.buffer.seconds | short | 300 | [0,…,3600] | low | Время буферизации до истечения срока действия учётных данных, которое необходимо поддерживать при обновлении учётных данных, в секундах. Если в противном случае обновление произошло бы ближе к истечению срока действия, чем количество секунд буфера, то обновление будет перемещено вверх, чтобы сохранить как можно большую часть времени буфера. Допустимые значения от 0 до 3600 (1 час). Если значение не указано, используется значение по умолчанию 300 (5 минут). Установленное значение и значение sasl.login.refresh.min.period.seconds игнорируются, если их сумма превышает оставшееся время жизни учётных данных. В настоящее время относится только к OAUTHBEARER. |
| sasl.login.refresh.min.period.seconds | short | 60 | [0,…,900] | low | Желаемое минимальное время ожидания потока обновления входа в систему перед обновлением учётных данных, в секундах. Допустимые значения от 0 до 900 (15 минут). Если значение не указано, используется значение по умолчанию 60 (1 минута). Установленное значение и значение sasl.login.refresh.buffer.seconds игнорируются, если их сумма превышает оставшееся время жизни учётных данных. В настоящее время относится только к OAUTHBEARER. |
| sasl.login.refresh.window.factor | double | 0.8 | [0.5,…,1.0] | low | Поток обновления входа в систему будет находиться в спящем режиме до тех пор, пока не будет достигнут установленный коэффициент окна относительно времени жизни учётных данных, после чего он попытается обновить учётные данные. Допустимые значения от 0.5 (50%) до 1.0 (100%) включительно. Если значение не указано, используется значение по умолчанию 0.8 (80%). В настоящее время относится только к OAUTHBEARER. |
| sasl.login.refresh.window.jitter | double | 0.05 | [0.0,…,0.25] | low | Максимальное количество случайного джиттера относительно времени жизни учётных данных, которое добавляется ко времени ожидания потока обновления входа в систему. Допустимые значения от 0 до 0.25 (25%) включительно. Если значение не указано, используется значение по умолчанию 0,05 (5%). В настоящее время относится только к OAUTHBEARER. |
| scheduled.rebalance.max.delay.ms | int | 300000 (5 минут) | [0,…,2147483647] | low | Максимальная задержка, которая запланирована, чтобы дождаться возвращения одного или нескольких ушедших обработчиков перед перебалансировкой и переназначением их коннекторов и тасков группе. В установленный период коннекторы и таски ушедших обработчиков остаются неназначенными. |
| security.protocol | string | PLAINTEXT | — | medium | Протокол, используемый для взаимодействия с брокерами. Допустимые значения: PLAINTEXT, SSL, SASL\_PLAINTEXT, SASL\_SSL. |
| send.buffer.bytes | int | 131072 (128 КиБ) | [0,…] | medium | Размер буфера отправки TCP (SO\_SNDBUF), используемого при отправке данных. Если установлено значение -1, будет использоваться значение по умолчанию ОС. |
| session.timeout.ms | int | 10000 (10 секунд) | — | high | Тайм-аут, используемый для обнаружения сбоев обработчиков. Обработчик отправляет периодические heartbeat-сообщения, чтобы показать брокеру свою работоспособность. Если до истечения тайм-аута сеанса брокер не получит heartbeat-сообщений, он удалит этого обработчика из группы и инициирует перебалансировку. Обратите внимание, что значение должно быть в допустимом диапазоне, заданном в конфигурации брокера с помощью свойств group.min.session.timeout.ms и group.max.session.timeout.ms. |
| socket.connection.setup.timeout.max.ms | long | 30000 (30 секунд) | — | low | Максимальное время, в течение которого клиент будет ожидать установления соединения с сокетом. Тайм-аут установки соединения будет экспоненциально увеличиваться при каждом последующем сбое соединения до установленного максимума. Чтобы избежать штормов соединений, к тайм-ауту будет применён коэффициент рандомизации 0.2, в результате чего получится случайный диапазон от 20% ниже до 20% выше вычисленного значения. |
| socket.connection.setup.timeout.ms | long | 10000 (10 секунд) | [0,…] | low | Время, в течение которого клиент будет ожидать установления соединения с сокетом. Если соединение не будет установлено до истечения тайм-аута, клиенты закроют канал сокета. |
| ssl.cipher.suites | list | null | — | low | Перечень комплектов шифров. Это именованная комбинация алгоритма аутентификации, шифрования, MAC и обмена ключами, используемая для согласования настроек безопасности для сетевого подключения с использованием сетевого протокола TLS или SSL. По умолчанию поддерживаются все доступные наборы шифров. |
| ssl.client.auth | string | none | — | low | Настройка брокера Kafka для запроса аутентификации клиента. Следующие настройки являются общими:   * ssl.client.auth=required — требуется аутентификация клиента; * ssl.client.auth=requested — аутентификация клиента не является обязательной. В отличие от значения required, если этот параметр установлен, клиент может отказаться от предоставления аутентификационной информации о себе; * ssl.client.auth=none — аутентификация клиента не требуется. |
| ssl.enabled.protocols | list | TLSv1.2 | — | medium | Перечень протоколов, разрешённых для SSL-соединений. По умолчанию используется TLSv1.2, TLSv1.3 при работе с Java 11 или новее, в противном случае — TLSv1.2. Со значением по умолчанию для Java 11 клиенты и серверы предпочтут TLSv1.3, если оба поддерживают его, и откатятся к TLSv1.2 в противном случае (при условии, что оба поддерживают как минимум TLSv1.2). Это значение по умолчанию подходит для большинства случаев. См. также информацию по конфигурации ssl.protocol. |
| ssl.endpoint.identification.algorithm | string | https | — | low | Алгоритм идентификации конечной точки для валидации имени хоста сервера с помощью сертификата сервера. |
| ssl.engine.factory.class | class | null | — | low | Класс типа org.apache.kafka.common.security.auth.SslEngineFactory для предоставления объектов SSLEngine. Значение по умолчанию org.apache.kafka.common.security.ssl.DefaultSslEngineFactory. |
| ssl.key.password | password | null | — | high | Пароль закрытого ключа в файле хранилища ключей или ключ PEM, указанный в ssl.keystore.key. Свойство требуется для клиентов, только если настроена двусторонняя аутентификация. |
| ssl.keymanager.algorithm | string | SunX509 | — | low | Алгоритм, используемый службой управления ключами для SSL-соединений. Значение по умолчанию — это алгоритм службы, настроенный Java Virtual Machine. |
| ssl.keystore.certificate.chain | password | null | — | high | Цепочка сертификатов в формате, указанном свойством ssl.keystore.type. По умолчанию служба SSL-механизма поддерживает только формат PEM с перечнем сертификатов X.509. |
| ssl.keystore.key | password | null | — | high | Закрытый ключ в формате, указанном свойством ssl.keystore.type. По умолчанию служба SSL-механизма поддерживает только формат PEM с ключами PKCS#8. Если ключ зашифрован, пароль ключа должен быть указан с помощью ssl.key.password. |
| ssl.keystore.location | string | null | — | high | Расположение файла хранилища ключей. Свойство опционально для клиента, и может использоваться для двусторонней аутентификации клиента. |
| ssl.keystore.password | password | null | — | high | Пароль хранилища для файла хранилища ключей. Свойство опционально для клиента, и требуется только в том случае, если настроено свойство ssl.keystore.location. Пароль хранилища ключей не поддерживается для формата PEM. |
| ssl.keystore.type | string | JKS | — | medium | Формат файла хранилища ключей. Свойство опционально для клиента. |
| ssl.protocol | string | TLSv1.2 | — | medium | SSL-протокол, используемый для генерации SSLContext. По умолчанию используется значение TLSv1.3 при работе с Java 11 или новее, в противном случае — TLSv1.2. Данное значение подходит для большинства случаев использования. Допустимые значения в последних JVM: TLSv1.2 и TLSv1.3. TLS, TLSv1.1, SSL, SSLv2 и SSLv3 могут поддерживаться в старых JVM, но их использование не рекомендуется из-за известных уязвимостей безопасности. Со значением по умолчанию для данной конфигурации и свойства ssl.enabled.protocols клиенты будут переходить на TLSv1.2, если сервер не поддерживает TLSv1.3. Если для данной конфигурации установлено значение TLSv1.2, клиенты не будут использовать TLSv1.3, даже если оно одно из значений в свойстве ssl.enabled.protocols, а сервер поддерживает только TLSv1.3. |
| ssl.provider | string | null | — | medium | Имя поставщика безопасности, используемого для SSL-соединений. Значение по умолчанию — это поставщик безопасности для JVM по умолчанию. |
| ssl.secure.random.implementation | string | null | — | low | Имплементация SecureRandom PRNG, используемая для операций шифрования SSL. |
| ssl.trustmanager.algorithm | string | PKIX | — | low | Алгоритм доверенной службы управления ключами для SSL-соединений. Значение по умолчанию — это алгоритм службы, настроенный для Java Virtual Machine. |
| ssl.truststore.certificates | password | null | — | high | Доверенные сертификаты в формате, указанном свойством ssl.truststore.type. Служба SSL-механизма по умолчанию поддерживает только формат PEM с сертификатами X.509. |
| ssl.truststore.location | string | null | — | high | Расположение файла хранилища доверенных сертификатов. |
| ssl.truststore.password | password | null | — | high | Пароль для файла доверенного хранилища. Если пароль не установлен, настроенный файл хранилища доверенных сертификатов по-прежнему будет использоваться, но проверка целостности будет отключена. Пароль доверенного хранилища не поддерживается для формата PEM. |
| ssl.truststore.type | string | JKS | — | medium | Формат файла хранилища доверенных сертификатов. |
| status.storage.partitions | int | 5 | — | low | Количество партиций, использованных при создании топика хранилища статусов.  В качестве значения может быть установлено положительное число или -1, чтобы использовать значение брокера по умолчанию. |
| status.storage.replication.factor | short | 3 | — | low | Фактор репликации, используемый при создании топика хранилища статусов.  В качестве значения может быть установлено положительное число, не превышающее количество брокеров в кластере Kafka, или -1, чтобы использовать значение брокера по умолчанию. |
| status.storage.topic | string | — | — | high | Название темы Kafka, в которой хранятся коннектор и статус таска. |
| task.shutdown.graceful.timeout.ms | long | 5000 (5 секунд) | — | low | Время ожидания корректного завершения тасков. Параметр устанавливает общее количество времени, а не на отдельный таск. У всех тасков завершение работы срабатывает по триггеру, затем они ожидаются последовательно. |
| topic.creation.enable | boolean | true | — | low | Разрешение автоматического создания топиков, используемых коннекторами источников, если коннекторы источников сконфигурированы с использованием свойств topic.creation.. Каждый таск будет использовать клиент администратора для создания своих топиков и не будет зависеть от брокеров Kafka для автоматического создания топиков. |
| topic.tracking.allow.reset | boolean | true | — | low | Если установлено значение true, это позволяет запросам пользователей сбрасывать набор активных топиков для каждого коннектора. |
| topic.tracking.enable | boolean | true | — | low | Включение отслеживания набора активных топиков для каждого коннектора во время выполнения. |
| value.converter | class | — | — | high | Класс Converter, используемый для преобразования между форматом Kafka Connect и сериализованной формой, записанной в Kafka. Свойство контролирует формат значений в сообщениях, записываемых в Kafka или считываемых из него, и, поскольку он не зависит от коннекторов, он позволяет любому коннектору работать с любым форматом сериализации. Примеры распространённых форматов включают JSON и Avro. |
| worker.sync.timeout.ms | int | 3000 (3 секунды) | — | medium | Когда обработчик не синхронизирован с другими обработчиками и ему необходимо повторно синхронизировать конфигурации, будет произведена задержка до установленного времени, прежде чем отказаться, покинуть группу и дождаться периода отсрочки перед повторным присоединением. |
| worker.unsync.backoff.ms | int | 300000 (5 минут) | — | medium | Если обработчик не синхронизирован с другими обработчиками и падает, не догнав значение worker.sync.timeout.ms, он покидает кластер Connect на установленное время до повторного присоединения. |

Конфигурации коннектора источника

Конфигурации коннектора источника и их значения по умолчанию представлены в таблице ниже в алфавитном порядке по наименованию конфигурации.

Таблица — Конфигурации коннектора источника

| Конфигурация | Тип | Значение по умолчанию | Действительные значения | Важность | Описание |
| --- | --- | --- | --- | --- | --- |
| config.action.reload | string | restart | [none, restart] | low | Действие, которое Connect должен выполнять с коннектором, когда изменения во внешних поставщиках конфигурации приводят к изменению свойств конфигурации коннектора. Значение none означает, что Connect ничего не будет делать. Значение restart указывает, что Connect следует перезапустить/перезагрузить коннектор с обновлёнными свойствами конфигурации. Фактически перезапуск может быть запланирован в будущем, если внешний поставщик конфигурации указывает, что значение конфигурации истекает в будущем. |
| connector.class | string | — | — | high | Имя или алиас класса для данного коннектора. Параметр должен быть субклассом org.apache.kafka.connect.connector.Connector. Если коннектором является org.apache.kafka.connect.file.FileStreamSinkConnector, вы можете указать данного полное имя или использовать FileStreamSink или FileStreamSinkConnector, чтобы сделать конфигурацию немного короче. |
| errors.log.enable | boolean | false | — | medium | Если установлено значение true, записывается каждая ошибка, а также сведения о неудачных операциях и проблемных записях в лог приложения Connect. По умолчанию установлено значение false, поэтому записываются только недопустимые ошибки. |
| errors.log.include.messages | boolean | false | — | medium | Параметр определяет, следует ли включать в лог Connect запись, которая привела к сбою. По умолчанию установлено значение false, которое предотвращает запись ключей, значений и заголовков записи в файлы лога, хотя некоторая информация, такая как топик и номер партиции, всё равно будет логироваться. |
| errors.retry.timeout | long | 0 | — | medium | Максимальная продолжительность повторной попытки неудачной операции (в миллисекундах). Значение по умолчанию — 0, что означает, что повторных попыток предприниматься не будет. Значение -1 означает, что будет предприниматься бесконечное количество попыток. |
| errors.retry.delay.max.ms | long | 60000 (1 минута) | — | medium | Максимальная продолжительность в миллисекундах между последовательными попытками повтора. По достижении установленого предела к задержке будет добавлен джиттер, чтобы предотвратить проблемы thundering herd. |
| errors.tolerance | string | none | [none, all] | medium | Поведение для допуска ошибок во время операций коннектора. Значение none является значением по умолчанию и сигнализирует о том, что любая ошибка приведёт к немедленному сбою таска коннектора. Значение all изменяет поведение, чтобы пропускать проблемные записи. |
| header.converter | class | null | — | low | Класс HeaderConverter, используемый для преобразования между форматом Kafka Connect и сериализованной формой, записанной в Kafka. Параметр контролирует формат значений заголовков в сообщениях, записываемых в Kafka или считываемых из него, и, поскольку он не зависит от коннекторов, он позволяет любому соединителю работать с любым форматом сериализации. Примеры распространённых форматов включают JSON и Avro. По умолчанию SimpleHeaderConverter используется для сериализации значений заголовков в строки и их десериализации путём вывода схем. |
| key.converter | class | null | — | low | Класс Converter, используемый для преобразования между форматом Kafka Connect и сериализованной формой, записанной в Kafka. Свойство контролирует формат ключей в сообщениях, записываемых в Kafka или считываемых из него, и, поскольку это не зависит от коннекторов, позволяет любому коннектору работать с любым форматом сериализации. Примеры распространённых форматов включают JSON и Avro. |
| name | string | — | — | high | Глобально уникальное имя, которое будет использоваться для данного коннектора.  В качестве значения указывается непустая строка без управляющих символов ISO. |
| predicates | list | "" | — | low | Алиасы для предикатов, используемых трансформациями.  В качестве значений должна быть установлена ненулевая строка, и значение должно быть уникальным алиасом предикатов. |
| tasks.max | int | 1 | [1,…] | high | Максимальное количество тасков для данного коннектора. |
| topic.creation.groups | list | "" | — | low | Группы конфигураций для топиков, созданные коннекторами источника.  В качестве значений должна быть установлена ненулевая строка, и значение должно быть уникальной группой создания топика. |
| transforms | list | "" | — | low | Алиасы трансформаций, применяемых к записям.  В качестве значений должна быть установлена ненулевая строка, и значение должно быть уникальным алиасом трансформаций. |
| value.converter | class | null | — | low | Класс Converter, используемый для преобразования между форматом Kafka Connect и сериализованной формой, записанной в Kafka. Свойство контролирует формат значений в сообщениях, записываемых в Kafka или считываемых из него, и, поскольку он не зависит от коннекторов, он позволяет любому коннектору работать с любым форматом сериализации. Примеры распространённых форматов включают JSON и Avro. |

Конфигурации sink-коннектора

Конфигурации sink-коннектора источника и их значения по умолчанию представлены в таблице ниже в алфавитном порядке по наименованию конфигурации.

Таблица — Конфигурации sink-коннектора

| Конфигурация | Тип | Значение по умолчанию | Действительные значения | Важность | Описание |
| --- | --- | --- | --- | --- | --- |
| config.action.reload | string | restart | [none, restart] | low | Действие, которое Connect должен выполнять с коннектором, когда изменения во внешних поставщиках конфигурации приводят к изменению свойств конфигурации коннектора. Значение none означает, что Connect ничего не будет делать. Значение restart указывает, что Connect следует перезапустить/перезагрузить коннектор с обновлёнными свойствами конфигурации. Фактически перезапуск может быть запланирован в будущем, если внешний поставщик конфигурации указывает, что значение конфигурации истекает в будущем. |
| connector.class | string | — | — | high | Имя или алиас класса для данного коннектора. Параметр должен быть субклассом org.apache.kafka.connect.connector.Connector. Если коннектором является org.apache.kafka.connect.file.FileStreamSinkConnector, вы можете указать данного полное имя или использовать FileStreamSink или FileStreamSinkConnector, чтобы сделать конфигурацию немного короче. |
| errors.deadletterqueue.context.headers.enable | boolean | false | — | medium | Если установлено значение true, добавляются заголовки, содержащие контекст ошибки, к сообщениям, записанным в очередь недоставленных сообщений (DLQ). Чтобы избежать конфликтов с заголовками из исходной записи, все ключи заголовка контекста ошибки начинаются с \_\_connect.errors. |
| errors.deadletterqueue.topic.name | string | "" | — | medium | Имя топика, который будет использоваться в качестве очереди недоставленных сообщений (dead letter queue, DLQ) для сообщений, которые приводят к ошибке при обработке данным sink-коннектором, или его трансформациями или конвертерами. Название топика по умолчанию пустое, что означает, что сообщения не записываются в DLQ. |
| errors.deadletterqueue.topic.replication.factor | short | 3 | — | medium | Фактор репликации, используемый для создания топика очереди недоставленных сообщений (DLQ), когда он ещё не существует. |
| errors.log.enable | boolean | false | — | medium | Если установлено значение true, записывается каждая ошибка, а также сведения о неудачных операциях и проблемных записях в лог приложения Connect. По умолчанию установлено значение false, поэтому записываются только недопустимые ошибки. |
| errors.log.include.messages | boolean | false | — | medium | Параметр определяет, следует ли включать в лог Connect запись, которая привела к сбою. По умолчанию установлено значение false, которое предотвращает запись ключей, значений и заголовков записи в файлы лога, хотя некоторая информация, такая как топик и номер партиции, всё равно будет логироваться. |
| errors.retry.delay.max.ms | long | 60000 (1 минута) | — | medium | Максимальная продолжительность в миллисекундах между последовательными попытками повтора. По достижении установленого предела к задержке будет добавлен джиттер, чтобы предотвратить проблемы thundering herd. |
| errors.retry.timeout | long | 0 | — | medium | Максимальная продолжительность повторной попытки неудачной операции (в миллисекундах). Значение по умолчанию — 0, что означает, что повторных попыток предприниматься не будет. Значение -1 означает, что будет предприниматься бесконечное количество попыток. |
| errors.tolerance | string | none | [none, all] | medium | Поведение для допуска ошибок во время операций коннектора. Значение none является значением по умолчанию и сигнализирует о том, что любая ошибка приведёт к немедленному сбою таска коннектора. Значение all изменяет поведение, чтобы пропускать проблемные записи. |
| header.converter | class | null | — | low | Класс HeaderConverter, используемый для преобразования между форматом Kafka Connect и сериализованной формой, записанной в Kafka. Параметр контролирует формат значений заголовков в сообщениях, записываемых в Kafka или считываемых из него, и, поскольку он не зависит от коннекторов, он позволяет любому соединителю работать с любым форматом сериализации. Примеры распространённых форматов включают JSON и Avro. По умолчанию SimpleHeaderConverter используется для сериализации значений заголовков в строки и их десериализации путём вывода схем. |
| key.converter | class | null | — | low | Класс Converter, используемый для преобразования между форматом Kafka Connect и сериализованной формой, записанной в Kafka. Свойство контролирует формат ключей в сообщениях, записываемых в Kafka или считываемых из него, и, поскольку это не зависит от коннекторов, позволяет любому коннектору работать с любым форматом сериализации. Примеры распространённых форматов включают JSON и Avro. |
| name | string | — | — | high | Глобально уникальное имя, которое будет использоваться для данного коннектора.  В качестве значения указывается непустая строка без управляющих символов ISO. |
| predicates | list | "" | — | low | Алиасы для предикатов, используемых трансформациями.  В качестве значений должна быть установлена ненулевая строка, и значение должно быть уникальным алиасом предикатов. |
| tasks.max | int | 1 | [1,…] | high | Максимальное количество тасков для данного коннектора. |
| topics | list | "" | — | high | Перечень топиков для потребления, разделённый запятыми. |
| topics.regex | string | "" | — | high | Регулярное выражение, задающее топики для потребления. Внутри регулярное выражение компилируется в java.util.regex.Pattern. Должен быть указан только один параметр из параметров topics или topics.regex.  В качестве значения указывается только валидное регулярное выражение. |
| transforms | list | "" | — | low | Алиасы трансформаций, применяемых к записям.  В качестве значений должна быть установлена ненулевая строка, и значение должно быть уникальным алиасом трансформаций. |
| value.converter | class | null | — | low | Класс Converter, используемый для преобразования между форматом Kafka Connect и сериализованной формой, записанной в Kafka. Свойство контролирует формат значений в сообщениях, записываемых в Kafka или считываемых из него, и, поскольку он не зависит от коннекторов, он позволяет любому коннектору работать с любым форматом сериализации. Примеры распространённых форматов включают JSON и Avro. |

Конфигурации стримов

Конфигурации клиентской библиотеки Kafka Streams (стримов Kafka) и их значения по умолчанию представлены в таблице ниже в алфавитном порядке по наименованию конфигурации.

Таблица — Конфигурации стримов

| Конфигурация | Тип | Значение по умолчанию | Действительные значения | Важность | Описание |
| --- | --- | --- | --- | --- | --- |
| acceptable.recovery.lag | long | 10000 | [0,…] | medium | Максимально допустимое отставание (количество смещений, которое нужно догнать) для клиента, которое считается догоняющим для активного таска. Значение должно соответствовать времени восстановления менее минуты для данной рабочей нагрузки. Значение должно быть не менее 0. |
| application.id | string | — | — | high | Идентификатор приложения потоковой обработки. Значение должно быть уникальным в кластере Kafka. Параметр используется как:  префикс идентификатора клиента по умолчанию;  идентификатор группы для управления членством;  префикс лога изменений топика. |
| application.server | string | "" | — | low | Пара host:port, указывающая на определяемую пользователем конечную точку, которая может использоваться для обнаружения хранилища статусов и интерактивных запросов в данном инстансе KafkaStreams. |
| bootstrap.servers | list | — | — | high | Перечень пар хост/порт, используемых для установления начального подключения к кластеру Kafka. Клиент будет использовать все серверы независимо от того, какие серверы указаны здесь для начальной подключения, — данный перечень влияет только на инициирующие хосты, используемые для обнаружения полного набора серверов. Данный перечень должен быть в формате: host1:port1,host2:port2,…. Поскольку эти серверы используются только для начального соединения, чтобы обнаружить полное членство в кластере (которое может динамически меняться), данный перечень не обязательно должен содержать полный набор серверов (может потребоваться больше одного сервера, если какой-либо сервер выходит из строя). |
| buffered.records.per.partition | int | 1000 | — | low | Максимальное количество записей для буфера каждой партиции. |
| built.in.metrics.version | string | latest | [0.10.0-2.4, latest] | low | Версия используемых встроенных метрик. |
| cache.max.bytes.buffering | long | 10485760 | [0,…] | medium | Максимальное количество байт памяти, которое будет использоваться для буферизации во всех потоках. |
| client.id | string | "" | — | medium | Строка префикса идентификатора, используемая для идентификаторов клиентов внутреннего потребителя, поставщика и потребителя восстановления, с шаблоном -StreamThread--. |
| commit.interval.ms | long | 30000 (30 секунд) | [0,…] | low | Частота в миллисекундах, с которой сохраняется позиция процессора.  Обратите внимание, если для свойства processing.guarantee задано значение exactly\_once, значение по умолчанию будет равно 100, в противном случае значением по умолчанию будет 30000. |
| connections.max.idle.ms | long | 540000 (9 минут) | — | low | Неактивные подключения закрываются по истечении количества миллисекунд, указанного в данной конфигурации. |
| default.deserialization.exception.handler | class | org.apache.kafka.streams.errors.LogAndFailExceptionHandler | — | medium | Класс обработки исключений, имплементирующий интерфейс org.apache.kafka.streams.errors.DeserializationExceptionHandler. |
| default.key.serde | class | org.apache.kafka.common.serialization.Serdes$ByteArraySerde | — | medium | Класс сериализатора/десериализатора по умолчанию для ключа, который имплементирует интерфейс org.apache.kafka.common.serialization.Serde. Обратите внимание, когда используется оконный класс serde, необходимо установить внутренний класс serde, который имплементирует интерфейс org.apache.kafka.common.serialization.Serde через default.windowed.key.serde.inner или default.windowed.value.serde.inner. |
| default.production.exception.handler | class | org.apache.kafka.streams.errors.DefaultProductionExceptionHandler | — | medium | Класс обработки исключений, имплементирующий интерфейс org.apache.kafka.streams.errors.ProductionExceptionHandler. |
| default.timestamp.extractor | class | org.apache.kafka.streams.processor.FailOnInvalidTimestamp | — | medium | Класс извлечения метки времени по умолчанию, который имплементирует интерфейс org.apache.kafka.streams.processor.TimestampExtractor. |
| default.value.serde | class | org.apache.kafka.common.serialization.Serdes$ByteArraySerde | — | medium | Класс сериализатора/десериализатора по умолчанию для значения, который имплементирует интерфейс org.apache.kafka.common.serialization.Serde. Обратите внимание, когда используется оконный класс serde, необходимо установить внутренний класс serde, который имплементирует интерфейс org.apache.kafka.common.serialization.Serde через default.windowed.key.serde.inner или default.windowed.value.serde.inner. |
| default.windowed.key.serde.inner | class | null | — | medium | Сериализатор/десериализатор по умолчанию для внутреннего класса оконного ключа. Должен имплементировать интерфейс org.apache.kafka.common.serialization.Serde. |
| default.windowed.value.serde.inner | class | null | — | medium | Сериализатор/десериализатор по умолчанию для внутреннего класса оконного значения. Должен имплементировать интерфейс org.apache.kafka.common.serialization.Serde. |
| max.task.idle.ms | long | 0 | — | medium | Максимальное количество времени в миллисекундах, в течение которого таск стрима будет оставаться в режиме ожидания, если не все его буферы партиций содержат записи, чтобы избежать потенциальной обработки записей вне очереди в нескольких входных стримах. |
| max.warmup.replicas | int | 2 | [1,…] | medium | Максимальное количество реплик (дополнительных резервных серверов сверх установленного значения в свойстве num.standbys), которые могут быть назначены сразу для того, чтобы таск оставался доступным на одном инстансе, пока он разогревается на другом инстансе, которому он был переназначен. Параметр используется для ограничения количества дополнительного трафика брокера и статуса кластера, которые могут использоваться для обеспечения высокой доступности. Значение должно быть не менее 1. |
| metadata.max.age.ms | long | 300000 (5 минут) | [0,…] | low | Период времени в миллисекундах, по истечении которого метаданные будут принудительно обновлены, даже если отсутствуют какие-либо изменения в лидере партиции, чтобы упреждающе обнаруживать новых брокеров или партиции. |
| metric.reporters | list | "" | — | low | Перечень классов для использования в качестве репортеров метрик (metrics reporters). Имплементация интерфейса org.apache.kafka.common.metrics.MetricsReporter позволяет подключать классы, которые будут получать уведомления о создании новой метрики. JmxReporter всегда включён для регистрации статистики JMX. |
| metrics.num.samples | int | 2 | [1,…] | low | Количество образцов, поддерживаемых для вычисления метрик. |
| metrics.recording.level | string | INFO | [INFO, DEBUG, TRACE] | low | Самый высокий уровень записи метрик. |
| metrics.sample.window.ms | long | 30000 (30 секунд) | [0,…] | low | Временной промежуток, в течение которого вычисляется образец метрики. |
| num.standby.replicas | int | 0 | — | medium | Количество резервных реплик для каждого таска. |
| num.stream.threads | int | 1 | — | medium | Количество потоков для выполнения обработки стримов. |
| partition.grouper | class | org.apache.kafka.streams.processor.DefaultPartitionGrouper | — | low | Класс группировщика (grouper) партиций, имплементирующий интерфейс org.apache.kafka.streams.processor.PartitionGrouper.  Внимание: данная конфигурация устарела и будет удалена в версии 3.0.0. |
| poll.ms | long | 100 | — | low | Время в миллисекундах, в течение которого блокируется ожидание ввода. |
| probing.rebalance.interval.ms | long | 600000 (10 минут) | [60000,…] | low | Максимальное время ожидания в миллисекундах перед запуском перебалансировки для проверки warmup-реплик, которые закончили warmup и готовы стать активными. Probing-перебалансировка будет запускаться до тех пор, пока назначение не будет сбалансировано. Значение должно быть не менее 1 минуты. |
| processing.guarantee | string | at\_least\_once | [at\_least\_once, exactly\_once, exactly\_once\_beta] | medium | Гарантия обработки, которую следует использовать. Возможные значения: at\_least\_once (по умолчанию), even\_once (требуются брокеры версии 0.11.0 или выше) и even\_once\_beta (требуются брокеры версии 2.5 или выше). Обратите внимание, что для однократной обработки требуется кластер из как минимум трёх брокеров по умолчанию, что является рекомендуемой настройкой для производственной среды; для среды разработки вы можете изменить поведение, настроив параметры брокера transaction.state.log.replication.factor и transaction.state.log.min.isr. |
| receive.buffer.bytes | int | 32768 (32 КиБ) | [-1,…] | medium | Размер приёмного буфера TCP (SO\_RCVBUF) для использования при чтении данных. Если установлено значение -1, будет использоваться значение по умолчанию ОС. |
| reconnect.backoff.max.ms | long | 1000 (1 секунда) | [0,…] | low | Максимальное время ожидания в миллисекундах при повторном подключении к брокеру, к которому неоднократно не удавалось подключиться. Если значение установлено, отсрочка для каждого хоста будет экспоненциально увеличиваться при каждом последующем сбое соединения, вплоть до установленного максимума. После расчёта увеличения отсрочки добавляется 20% случайного джиттера, чтобы избежать штормов соединения. |
| reconnect.backoff.ms | long | 50 | [0,…] | low | Базовое время ожидания перед попыткой повторного подключения к данному хосту. Это позволяет избежать многократного подключения к хосту в замкнутом цикле. Установленная отсрочка применяется ко всем попыткам подключения клиента к брокеру. |
| replication.factor | int | 1 | — | high | Фактор репликации для лога изменений топиков и репартиционирования топиков, созданных приложением потоковой обработки. Если ваш кластер брокеров имеет версию 2.4 или новее, вы можете установить -1, чтобы использовать значение фактора репликации брокера по умолчанию. |
| request.timeout.ms | int | 40000 (40 секунд) | [0,…] | low | Конфигурация контролирует максимальное количество времени, в течение которого клиент будет ждать ответа на запрос. Если ответ не будет получен до истечения тайм-аута, клиент повторно отправит запрос, если необходимо, или запрос завершится неудачей, если количество повторных попыток исчерпано. |
| retries | int | 2147483647 | [0,…,2147483647] | low | Установка значения больше нуля приведёт к тому, что клиент повторно отправит любую запись, отправка которой не удалась с потенциально временной ошибкой. Рекомендуется установить значение либо равным нулю, либо MAX\_VALUE и использовать соответствующие параметры тайм-аута, чтобы контролировать, как долго клиент должен повторять запрос. |
| retry.backoff.ms | long | 100 | [0,…] | low | Время ожидания перед попыткой повторить неудавшийся запрос к данной партиции топика. Это позволяет избежать повторной отправки запросов в замкнутом цикле при некоторых сценариях сбоя. |
| rocksdb.config.setter | class | null | — | low | Класс установки конфигурации Rocks DB или имя класса, имплементирующий интерфейс org.apache.kafka.streams.state.RocksDBConfigSetter. |
| security.protocol | string | PLAINTEXT | — | medium | Протокол, используемый для взаимодействия с брокерами. Допустимые значения: PLAINTEXT, SSL, SASL\_PLAINTEXT, SASL\_SSL. |
| send.buffer.bytes | int | 131072 (128 КиБ) | [-1,…] | low | Размер буфера отправки TCP (SO\_SNDBUF), используемого при отправке данных. Если установлено значение -1, будет использоваться значение по умолчанию ОС. |
| state.cleanup.delay.ms | long | 600000 (10 минут) | — | low | Время в миллисекундах ожидания перед удалением статуса после миграции партиции. Будут удалены только каталоги статусов, которые не были изменены как минимум для state.cleanup.delay.ms. |
| state.dir | string | /tmp/kafka-streams | — | high | Расположение каталога для хранилища статусов. Данный путь должен быть уникальным для каждого инстанса потока, использующего одну и ту же базовую файловую систему. |
| task.timeout.ms | long | 300000 (5 минут) | [0,…] | medium | Максимальное количество времени в миллисекундах, в течение которого таск может зависнуть из-за внутренних ошибок и повторных попыток, пока не возникнет ошибка. При тайм-ауте 0 мс таск вызовет ошибку при первой внутренней ошибке. Для любого тайм-аута больше 0 мс задача повторит попытку хотя бы один раз, прежде чем возникнет ошибка. |
| topology.optimization | string | none | [none, all] | medium | Конфигурация, сообщающая Kafka Streams, следует ли оптимизировать топологию. По умолчанию отключена. |
| upgrade.from | string | null | [null, 0.10.0, 0.10.1, 0.10.2, 0.11.0, 1.0, 1.1, 2.0, 2.1, 2.2, 2.3] | low | Позволяет выполнять обновление обратно совместимым способом. Это необходимо при обновлении с [0.10.0, 1.1] до 2.0+ или при обновлении с [2.0, 2.3] до 2.4+. При обновлении с 2.4 до более новой версии указывать данную конфигурацию не требуется. По умолчанию установлено значение null. Допустимые значения: 0.10.0, 0.10.1, 0.10.2, 0.11.0, 1.0, 1.1, 2.0, 2.1, 2.2, 2.3 (для обновления с соответствующей старой версии). |
| window.size.ms | long | null | — | low | Устанавливает размер окна для десериализатора для расчёта времени окончания окна. |
| windowstore.changelog.additional.retention.ms | long | 86400000 (1 день) | — | low | Добавление maintainMs, чтобы исключить риск преждевременного удаления данных из лога. Позволяет осуществлять отставание часов. Значение по умолчанию 1 день. |

Конфигурации администратора

Конфигурации клиентской библиотеки Kafka Admin и их значения по умолчанию представлены в таблице ниже в алфавитном порядке по наименованию конфигурации.

Таблица — Конфигурации администратора

| Конфигурация | Тип | Значение по умолчанию | Действительные значения | Важность | Описание |
| --- | --- | --- | --- | --- | --- |
| bootstrap.servers | list | — | — | high | Перечень пар хост/порт, используемых для установления начального подключения к кластеру Kafka. Клиент будет использовать все серверы независимо от того, какие серверы указаны здесь для начальной подключения, — данный перечень влияет только на инициирующие хосты, используемые для обнаружения полного набора серверов. Данный перечень должен быть в формате: host1:port1,host2:port2,…. Поскольку эти серверы используются только для начального соединения, чтобы обнаружить полное членство в кластере (которое может динамически меняться), данный перечень не обязательно должен содержать полный набор серверов (может потребоваться больше одного сервера, если какой-либо сервер выходит из строя). |
| client.dns.lookup | string | use\_all\_dns\_ips | [default, use\_all\_dns\_ips, resolve\_canonical\_bootstrap\_servers\_only] | medium | Параметр управляет тем, как клиент использует поиск DNS. Если установлено значение use\_all\_dns\_ips, клиент будеет последовательно подключаться к каждому возвращённому IP-адресу до тех пор, пока не будет установлено успешное соединение. После отключения используется следующий IP-адрес. После того, как все IP-адреса были использованы один раз, клиент снова разрешает IP-адреса из имени хоста (поиск DNS-имени как JVM, так и кэша ОС).  Если установлено значение resolve\_canonical\_bootstrap\_servers\_only, разрешается каждый адрес начальной загрузки в списке канонических имён. После фазы начальной загрузки поиск DNS осуществляется так, как при установленном значении use\_all\_dns\_ips.  Если установлено значение default (не рекомендуется), осуществляется попытка подключиться к первому IP-адресу, возвращённому поиском, даже если поиск вернёт несколько IP-адресов. |
| client.id | string | "" | — | medium | Строка идентификатора, передаваемая на сервер при выполнении запросов. Это делается для того, чтобы иметь возможность отслеживать источник запросов помимо IP-адреса/порта, позволяя включать логическое имя приложения в лог запросов на стороне сервера. |
| connections.max.idle.ms | long | 300000 (5 минут) | — | medium | Неактивные подключения закрываются по истечении количества миллисекунд, указанного в данной конфигурации. |
| default.api.timeout.ms | int | 60000 (1 минута) | [0,…] | medium | Параметр задаёт тайм-аут (в миллисекундах) для клиентских API. Данная конфигурация используется в качестве тайм-аута по умолчанию для всех клиентских операций, для которых не указан параметр timeout. |
| metadata.max.age.ms | long | 300000 (5 минут) | [0,…] | low | Период времени в миллисекундах, по истечении которого метаданные будут принудительно обновлены, даже если отсутствуют какие-либо изменения в лидере партиции, чтобы упреждающе обнаруживать новых брокеров или партиции. |
| metric.reporters | list | "" | — | low | Перечень классов для использования в качестве репортеров метрик (metrics reporters). Имплементация интерфейса org.apache.kafka.common.metrics.MetricsReporter позволяет подключать классы, которые будут получать уведомления о создании новой метрики. JmxReporter всегда включён для регистрации статистики JMX. |
| metrics.num.samples | int | 2 | [1,…] | low | Количество образцов, поддерживаемых для вычисления метрик. |
| metrics.recording.level | string | INFO | [INFO, DEBUG, TRACE] | low | Самый высокий уровень записи метрик. |
| metrics.sample.window.ms | long | 30000 (30 секунд) | [0,…] | low | Временной промежуток, в течение которого вычисляется образец метрики. |
| receive.buffer.bytes | int | 65536 (64 КиБ) | [-1,…] | medium | Размер приёмного буфера TCP (SO\_RCVBUF) для использования при чтении данных. Если установлено значение -1, будет использоваться значение по умолчанию ОС. |
| reconnect.backoff.max.ms | long | 1000 (1 секунда) | [0,…] | low | Максимальное время ожидания в миллисекундах при повторном подключении к брокеру, к которому неоднократно не удавалось подключиться. Если значение установлено, отсрочка для каждого хоста будет экспоненциально увеличиваться при каждом последующем сбое соединения, вплоть до установленного максимума. После расчёта увеличения отсрочки добавляется 20% случайного джиттера, чтобы избежать штормов соединения. |
| reconnect.backoff.ms | long | 50 | [0,…] | low | Базовое время ожидания перед попыткой повторного подключения к данному хосту. Это позволяет избежать многократного подключения к хосту в замкнутом цикле. Установленная отсрочка применяется ко всем попыткам подключения клиента к брокеру. |
| request.timeout.ms | int | 30000 (30 секунд) | [0,…] | medium | Конфигурация контролирует максимальное количество времени, в течение которого клиент будет ждать ответа на запрос. Если ответ не будет получен до истечения тайм-аута, клиент повторно отправит запрос, если необходимо, или запрос завершится неудачей, если количество повторных попыток исчерпано. |
| retries | int | 2147483647 | [0,…,2147483647] | low | Установка значения больше нуля приведёт к тому, что клиент повторно отправит любую запись, отправка которой не удалась с потенциально временной ошибкой. Рекомендуется установить значение либо равным нулю, либо MAX\_VALUE и использовать соответствующие параметры тайм-аута, чтобы контролировать, как долго клиент должен повторять запрос. |
| retry.backoff.ms | long | 100 | [0,…] | low | Время ожидания перед попыткой повторить неудавшийся запрос к данной партиции топика. Это позволяет избежать повторной отправки запросов в замкнутом цикле при некоторых сценариях сбоя. |
| sasl.client.callback.handler.class | class | null | — | medium | Полное имя класса обработчика обратного вызова SASL-клиента, имплементирующего интерфейс AuthenticateCallbackHandler. |
| sasl.jaas.config | password | null | — | medium | Параметры контекста входа JAAS для SASL-соединений в формате, используемом файлами конфигурации JAAS. [Здесь](http://docs.oracle.com/javase/8/docs/technotes/guides/security/jgss/tutorials/LoginConfigFile.html) описан формат файла конфигурации JAAS. Формат для значения: loginModuleClass controlFlag (optionName=optionValue)\*;. Для брокеров конфигурация должна начинаться с префикса слушателя и имени SASL-механизма в нижнем регистре. Например, listener.name.sasl\_ssl.scram-sha-256.sasl.jaas.config=com.example.ScramLoginModule required; |
| sasl.kerberos.kinit.cmd | string | /usr/bin/kinit | — | low | Путь к команде Kerberos kinit. |
| sasl.kerberos.min.time.before.relogin | long | 60000 | — | low | Время ожидания авторизации потока между попытками обновления. |
| sasl.kerberos.service.name | string | null | — | medium | Имя принципала Kerberos, под которым работает Kafka. Имя можно определить либо в конфигурации Kafka JAAS, либо в конфигурации Kafka. |
| sasl.kerberos.ticket.renew.jitter | double | 0.05 | — | low | Процент случайного джиттера, добавленный ко времени возобновления. |
| sasl.kerberos.ticket.renew.window.factor | double | 0.8 | — | low | Поток входа в систему будет находиться в спящем режиме до тех пор, пока не будет достигнут установленный коэффициент окна времени с момента последнего обновления до истечения срока действия тикета, после чего он попытается обновить тикет. |
| sasl.login.callback.handler.class | class | null | — | medium | Полное имя класса обработчика обратного вызова входа SASL, имплементирующего интерфейс AuthenticateCallbackHandler. Для брокеров конфигурация обработчика обратного вызова при входе в систему должна иметь префикс с префиксом слушателя и именем SASL-механизма в нижнем регистре. Например, listener.name.sasl\_ssl.scram-sha-256.sasl.login.callback.handler.class = com.example.CustomScramLoginCallbackHandler. |
| sasl.login.class | class | null | — | medium | Полное имя класса, имплементирующего интерфейс входа в систему. Для брокеров конфигурация входа в систему должна начинаться с префикса слушателя и имени SASL-механизма в нижнем регистре. Например, listener.name.sasl\_ssl.scram-sha-256.sasl.login.class = com.example.CustomScramLogin. |
| sasl.login.refresh.buffer.seconds | short | 300 | [0,…,3600] | low | Время буферизации до истечения срока действия учётных данных, которое необходимо поддерживать при обновлении учётных данных, в секундах. Если в противном случае обновление произошло бы ближе к истечению срока действия, чем количество секунд буфера, то обновление будет перемещено вверх, чтобы сохранить как можно большую часть времени буфера. Допустимые значения от 0 до 3600 (1 час). Если значение не указано, используется значение по умолчанию 300 (5 минут). Установленное значение и значение sasl.login.refresh.min.period.seconds игнорируются, если их сумма превышает оставшееся время жизни учётных данных. В настоящее время относится только к OAUTHBEARER. |
| sasl.login.refresh.min.period.seconds | short | 60 | [0,…,900] | low | Желаемое минимальное время ожидания потока обновления входа в систему перед обновлением учётных данных, в секундах. Допустимые значения от 0 до 900 (15 минут). Если значение не указано, используется значение по умолчанию 60 (1 минута). Установленное значение и значение sasl.login.refresh.buffer.seconds игнорируются, если их сумма превышает оставшееся время жизни учётных данных. В настоящее время относится только к OAUTHBEARER. |
| sasl.login.refresh.window.factor | double | 0.8 | [0.5,…,1.0] | low | Поток обновления входа в систему будет находиться в спящем режиме до тех пор, пока не будет достигнут установленный коэффициент окна относительно времени жизни учётных данных, после чего он попытается обновить учётные данные. Допустимые значения от 0.5 (50%) до 1.0 (100%) включительно. Если значение не указано, используется значение по умолчанию 0.8 (80%). В настоящее время относится только к OAUTHBEARER. |
| sasl.login.refresh.window.jitter | double | 0.05 | [0.0,…,0.25] | low | Максимальное количество случайного джиттера относительно времени жизни учётных данных, которое добавляется ко времени ожидания потока обновления входа в систему. Допустимые значения от 0 до 0.25 (25%) включительно. Если значение не указано, используется значение по умолчанию 0,05 (5%). В настоящее время относится только к OAUTHBEARER. |
| sasl.mechanism | string | GSSAPI | — | medium | SASL-механизм, используемый для клиентских подключений. Это может быть любой механизм, для которого доступен поставщик безопасности. GSSAPI — значение по умолчанию. |
| security.protocol | string | PLAINTEXT | — | medium | Протокол, используемый для взаимодействия с брокерами. Допустимые значения: PLAINTEXT, SSL, SASL\_PLAINTEXT, SASL\_SSL. |
| security.providers | string | null | — | low | Перечень настраиваемых классов создателей, каждый из которых возвращает поставщика, имплементирующего алгоритмы безопасности. Установленный классы должны имплементировать интерфейс org.apache.kafka.common.security.auth.SecurityProviderCreator. |
| send.buffer.bytes | int | 131072 (128 КиБ) | [-1,…] | medium | Размер буфера отправки TCP (SO\_SNDBUF), используемого при отправке данных. Если установлено значение -1, будет использоваться значение по умолчанию ОС. |
| socket.connection.setup.timeout.max.ms | long | 30000 (30 секунд) | — | medium | Максимальное время, в течение которого клиент будет ожидать установления соединения с сокетом. Тайм-аут установки соединения будет экспоненциально увеличиваться при каждом последующем сбое соединения до установленного максимума. Чтобы избежать штормов соединений, к тайм-ауту будет применён коэффициент рандомизации 0.2, в результате чего получится случайный диапазон от 20% ниже до 20% выше вычисленного значения. |
| socket.connection.setup.timeout.ms | long | 10000 (10 секунд) | — | medium | Время, в течение которого клиент будет ожидать установления соединения с сокетом. Если соединение не будет установлено до истечения тайм-аута, клиенты закроют канал сокета. |
| ssl.cipher.suites | list | null | — | low | Перечень комплектов шифров. Это именованная комбинация алгоритма аутентификации, шифрования, MAC и обмена ключами, используемая для согласования настроек безопасности для сетевого подключения с использованием сетевого протокола TLS или SSL. По умолчанию поддерживаются все доступные наборы шифров. |
| ssl.enabled.protocols | list | TLSv1.2 | — | medium | Перечень протоколов, разрешённых для SSL-соединений. По умолчанию используется TLSv1.2, TLSv1.3 при работе с Java 11 или новее, в противном случае — TLSv1.2. Со значением по умолчанию для Java 11 клиенты и серверы предпочтут TLSv1.3, если оба поддерживают его, и откатятся к TLSv1.2 в противном случае (при условии, что оба поддерживают как минимум TLSv1.2). Это значение по умолчанию подходит для большинства случаев. См. также информацию по конфигурации ssl.protocol. |
| ssl.endpoint.identification.algorithm | string | https | — | low | Алгоритм идентификации конечной точки для валидации имени хоста сервера с помощью сертификата сервера. |
| ssl.engine.factory.class | class | null | — | low | Класс типа org.apache.kafka.common.security.auth.SslEngineFactory для предоставления объектов SSLEngine. Значение по умолчанию org.apache.kafka.common.security.ssl.DefaultSslEngineFactory. |
| ssl.key.password | password | null | — | high | Пароль закрытого ключа в файле хранилища ключей или ключ PEM, указанный в ssl.keystore.key. Свойство требуется для клиентов, только если настроена двусторонняя аутентификация. |
| ssl.keymanager.algorithm | string | SunX509 | — | low | Алгоритм, используемый службой управления ключами для SSL-соединений. Значение по умолчанию — это алгоритм службы, настроенный Java Virtual Machine. |
| ssl.keystore.type | string | JKS | — | medium | Формат файла хранилища ключей. Свойство опционально для клиента. |
| ssl.keystore.certificate.chain | password | null | — | high | Цепочка сертификатов в формате, указанном свойством ssl.keystore.type. По умолчанию служба SSL-механизма поддерживает только формат PEM с перечнем сертификатов X.509. |
| ssl.keystore.key | password | null | — | high | Закрытый ключ в формате, указанном свойством ssl.keystore.type. По умолчанию служба SSL-механизма поддерживает только формат PEM с ключами PKCS#8. Если ключ зашифрован, пароль ключа должен быть указан с помощью ssl.key.password. |
| ssl.keystore.location | string | null | — | high | Расположение файла хранилища ключей. Свойство опционально для клиента, и может использоваться для двусторонней аутентификации клиента. |
| ssl.keystore.password | password | null | — | high | Пароль хранилища для файла хранилища ключей. Свойство опционально для клиента, и требуется только в том случае, если настроено свойство ssl.keystore.location. Пароль хранилища ключей не поддерживается для формата PEM. |
| ssl.protocol | string | TLSv1.2 | — | medium | SSL-протокол, используемый для генерации SSLContext. По умолчанию используется значение TLSv1.3 при работе с Java 11 или новее, в противном случае — TLSv1.2. Данное значение подходит для большинства случаев использования. Допустимые значения в последних JVM: TLSv1.2 и TLSv1.3. TLS, TLSv1.1, SSL, SSLv2 и SSLv3 могут поддерживаться в старых JVM, но их использование не рекомендуется из-за известных уязвимостей безопасности. Со значением по умолчанию для данной конфигурации и свойства ssl.enabled.protocols клиенты будут переходить на TLSv1.2, если сервер не поддерживает TLSv1.3. Если для данной конфигурации установлено значение TLSv1.2, клиенты не будут использовать TLSv1.3, даже если оно одно из значений в свойстве ssl.enabled.protocols, а сервер поддерживает только TLSv1.3. |
| ssl.provider | string | null | — | medium | Имя поставщика безопасности, используемого для SSL-соединений. Значение по умолчанию — это поставщик безопасности для JVM по умолчанию. |
| ssl.secure.random.implementation | string | null | — | low | Имплементация SecureRandom PRNG, используемая для операций шифрования SSL. |
| ssl.trustmanager.algorithm | string | PKIX | — | low | Алгоритм доверенной службы управления ключами для SSL-соединений. Значение по умолчанию — это алгоритм службы, настроенный для Java Virtual Machine. |
| ssl.truststore.certificates | password | null | — | high | Доверенные сертификаты в формате, указанном свойством ssl.truststore.type. Служба SSL-механизма по умолчанию поддерживает только формат PEM с сертификатами X.509. |
| ssl.truststore.location | string | null | — | high | Расположение файла хранилища доверенных сертификатов. |
| ssl.truststore.password | password | null | — | high | Пароль для файла доверенного хранилища. Если пароль не установлен, настроенный файл хранилища доверенных сертификатов по-прежнему будет использоваться, но проверка целостности будет отключена. Пароль доверенного хранилища не поддерживается для формата PEM. |
| ssl.truststore.type | string | JKS | — | medium | Формат файла хранилища доверенных сертификатов. |

Основные операции Kafka

В данном разделе рассмотрены наиболее распространённые операции Kafka. Все утилиты, рассмотренные в данном разделе, доступны в каталоге bin/ дистрибутива Kafka, и каждая утилита выводит сведения обо всех возможных параметрах командной строки, если она запускается без аргументов.

Добавление и удаление топиков

Топики можно создавать вручную, либо автоматически при первой публикации данных в несуществующем топике.

Если установлено автоматическое создание топиков, вы можете настроить конфигурации топиков по умолчанию, которые будут использоваться для автоматического создания топиков.

Топики добавляются и изменяются с использованием утилиты топиков:

|  |
| --- |
| > bin/kafka-topics.sh --bootstrap-server broker\_host:port --create --topic my\_topic\_name \  --partitions 20 --replication-factor 3 --config x=y |

Фактор репликации определяет, сколько серверов будет реплицировать каждое записанное сообщение. Если установлено значение фактора репликации 3, то до 2 серверов могут выйти из строя, прежде чем доступ к своим данным будет потерян. Рекомендуется использовать фактор репликации 2 или 3, чтобы вы могли прозрачно переключать машины, не прерывая потребление данных.

Счётчик партиций определяет количество логов, в которые будет шардироваться топик. Количество партиций влияет на несколько факторов. Во-первых, каждая партиция должна полностью уместиться на одном сервере. Таким образом, если у вас 20 партиций, полный набор данных (а также их чтение и запись) будет обрабатываться не более чем 20 серверами (не считая реплик). Наконец, количество партиций влияет на максимальный параллелизм ваших потребителей.

Лог сегментированных партиций размещается в собственной директории в каталоге логов Kafka. Имена таких папок состоят из имени топика, дополненного тире (-), и идентификатора партиции. Поскольку типичное имя папки не может быть длиннее 255 символов, на длину имён топиков применяется ограничение. Предполагается, что количество партиций никогда не превысит 100 000. Таким образом, имена топиков не могут быть длиннее 249 символов. Данная длина оставляет достаточно места в имени папки для тире и потенциально 5-значного идентификатора партиции.

Конфигурации, добавленные в командной строке, переопределяют настройки сервера по умолчанию для таких параметров, как продолжительность хранения данных. Полный набор конфигураций топика представлен в п. 1.3.

Изменение топиков

Изменение конфигурации топика и его партиционирования производится с использованием утилиты топиков.

Чтобы добавить партицию, выполните:

|  |
| --- |
| > bin/kafka-topics.sh --bootstrap-server broker\_host:port --alter --topic my\_topic\_name \  --partitions 40 |

Имейте в виду, что одним из вариантов использования партиций является семантическое партиционирование данных, а добавление партиций не меняет партиционирование существующих данных, поэтому это может нарушать потребление данных потребителями, если они зависят от данной партиции. То есть, если данные партиционированы с использованием hash(key) % number\_of\_partitions, тогда данное партиционирование потенциально будет перетасовано путём добавления партиций, но Kafka не будет пытаться автоматически перераспределять данные каким-либо образом.

Чтобы добавить конфигурацию, выполните:

|  |
| --- |
| > bin/kafka-configs.sh --bootstrap-server broker\_host:port --entity-type topics --entity-name my\_topic\_name --alter --add-config x=y |

Чтобы удалить конфигурацию, выполните:

|  |
| --- |
| > bin/kafka-configs.sh --bootstrap-server broker\_host:port --entity-type topics --entity-name my\_topic\_name --alter --delete-config x |

Чтобы удалить топик, выполните:

|  |
| --- |
| > bin/kafka-topics.sh --bootstrap-server broker\_host:port --delete --topic my\_topic\_name |

В настоящее время Kafka не поддерживает уменьшение количества партиций для топика.

Плавное выключение

Кластер Kafka автоматически обнаруживает любое отключение или сбой брокера и выбирает новых лидеров для партиций на данной машине. Это произойдёт, если сервер выйдет из строя или будет намеренно отключён для обслуживания или изменения конфигурации. В последних случаях Kafka поддерживает более плавный механизм остановки сервера, чем просто его отключение.

При плавной остановке (graceful shutdown) сервер использует две оптимизации:

1. Он синхронизирует все свои логи с диском, чтобы избежать необходимости восстановления логов при перезапуске (т.е. валидации контрольной суммы для всех сообщений в конце лога). Восстановление лога требует времени, поэтому оптимизация ускоряет преднамеренный перезапуск.

Он мигрирует любые партиции, для которых сервер является лидером, на другие реплики до завершения работы. Данная оптимизация ускоряет передачу лидерства и сводит к минимуму время недоступности каждой партиции до нескольких миллисекунд.

Синхронизация логов будет происходить автоматически всякий раз, когда сервер останавливается, кроме аппаратного сбоя, но для контролируемой миграции лидерства требуется использование специальной настройки:

|  |
| --- |
| controlled.shutdown.enable=true |

Обратите внимание, что управляемое завершение работы будет успешным только в том случае, если все партиции, размещённые на брокере, имеют реплики (т. е. фактор репликации больше 1 и хотя бы одна из этих реплик активна). Рекомендуется данная настройка, поскольку завершение работы последней реплики сделает партицию топика недоступной.

Балансировка лидерства

Каждый раз, когда брокер останавливается или выходит из строя, лидерство партициями такого брокера передаётся другим репликам. Когда брокер перезапускается, он будет только подписчиком для всех своих партиций, то есть он не будет использоваться для чтения и записи клиентами.

Чтобы избежать данного дисбаланса, Kafka имеет понятие предпочтительных реплик. Если перечень реплик для партиции равен 1, 5, 9, то нода 1 предпочтительнее в качестве лидера чем ноды 5 или 9, потому что она находится раньше в перечне реплик. По умолчанию кластер Kafka пытается возобновить лидерство для восстановленных реплик. Данное поведение настраивается с помощью:

|  |
| --- |
| auto.leader.rebalance.enable=true |

Вы также можете установить это значение в false, но тогда вам нужно будет вручную возобновить лидерство для восстановленных реплик, выполнив команду:

|  |
| --- |
| > bin/kafka-preferred-replica-election.sh --bootstrap-server broker\_host:port |

Балансировка реплик в стойках

Если в настройках установлена информация о стойках, то Kafka будет распределять реплики одной и той же партиции по разным стойкам. Данное поведение расширяет гарантии, которые Kafka предоставляет на случай отказа брокера, чтобы покрыть отказ стойки, снижая риск потери данных, если все брокеры на стойке выйдут из строя сразу. Данную функцию также можно применить к другим группам брокеров, таким как зоны доступности в EC2.

Вы можете указать, что брокер принадлежит к определённой стойке, добавив свойство в конфигурацию брокера:

|  |
| --- |
| broker.rack=my-rack-id |

Когда создаётся, изменяется топик или реплики перераспределяются, ограничение стойки будет соблюдаться, гарантируя, что реплики охватывают столько стоек, сколько они могут (партиция будет охватывать min(#racks, replication-factor) разных стоек).

Алгоритм, используемый для назначения реплик брокерам, гарантирует, что количество лидеров на одного брокера будет постоянным, независимо от того, как брокеры распределяются по стойкам. Это обеспечивает сбалансированную пропускную способность.

Однако, если стойкам назначено разное количество брокеров, назначение реплик не будет равномерным. Стойки с меньшим количеством брокеров получат больше реплик, что означает, что они будут использовать больше хранилища и вкладывать больше ресурсов в репликацию. Следовательно, разумно настроить равное количество брокеров на стойку.

Проверка позиции потребителя

Иногда полезно увидеть позицию ваших потребителей. У Kafka имеется утилита, которая покажет позицию всех потребителей в группе потребителей, а также то, насколько далеко они отстают от конца лога. Запуск данной утилиты в группе потребителей с именем my-group, использующей топик с именем my-topic, будет выглядеть следующим образом:

|  |
| --- |
| > bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --describe --group my-group  TOPIC PARTITION CURRENT-OFFSET LOG-END-OFFSET LAG CONSUMER-ID HOST CLIENT-ID  my-topic 0 2 4 2 consumer-1-029af89c-873c-4751-a720-cefd41a669d6 /127.0.0.1 consumer-1  my-topic 1 2 3 1 consumer-1-029af89c-873c-4751-a720-cefd41a669d6 /127.0.0.1 consumer-1  my-topic 2 2 3 1 consumer-2-42c1abd4-e3b2-425d-a8bb-e1ea49b29bb2 /127.0.0.1 consumer-2 |

Управление группами потребителей

С помощью утилиты ConsumerGroupCommand можно выводить перечень, описывать или удалять группы потребителей. Группа потребителей может быть удалена вручную или автоматически по истечении последнего зафиксированного смещения для этой группы. Ручное удаление работает только в том случае, если в группе нет активных участников. Например, чтобы перечислить все группы потребителей по всем топикам, выполните:

|  |
| --- |
| > bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --list  test-consumer-group |

Чтобы просмотреть смещения, как упоминалось ранее, мы «описываем» группу потребителей следующим образом:

|  |
| --- |
| > bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --describe --group my-group  TOPIC PARTITION CURRENT-OFFSET LOG-END-OFFSET LAG CONSUMER-ID HOST CLIENT-ID  topic3 0 241019 395308 154289 consumer2-e76ea8c3-5d30-4299-9005-47eb41f3d3c4 /127.0.0.1 consumer2  topic2 1 520678 803288 282610 consumer2-e76ea8c3-5d30-4299-9005-47eb41f3d3c4 /127.0.0.1 consumer2  topic3 1 241018 398817 157799 consumer2-e76ea8c3-5d30-4299-9005-47eb41f3d3c4 /127.0.0.1 consumer2  topic1 0 854144 855809 1665 consumer1-3fc8d6f1-581a-4472-bdf3-3515b4aee8c1 /127.0.0.1 consumer1  topic2 0 460537 803290 342753 consumer1-3fc8d6f1-581a-4472-bdf3-3515b4aee8c1 /127.0.0.1 consumer1  topic3 2 243655 398812 155157 consumer4-117fe4d3-c6c1-4178-8ee9-eb4a3954bee0 /127.0.0.1 consumer4 |

Существует ряд дополнительных параметров «описания», которые можно использовать для предоставления более подробной информации о группе потребителей:

--members — этот параметр предоставляет перечень всех активных членов группы потребителей;

|  |
| --- |
| > bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --describe --group my-group --members  CONSUMER-ID HOST CLIENT-ID #PARTITIONS  consumer1-3fc8d6f1-581a-4472-bdf3-3515b4aee8c1 /127.0.0.1 consumer1 2  consumer4-117fe4d3-c6c1-4178-8ee9-eb4a3954bee0 /127.0.0.1 consumer4 1  consumer2-e76ea8c3-5d30-4299-9005-47eb41f3d3c4 /127.0.0.1 consumer2 3  consumer3-ecea43e4-1f01-479f-8349-f9130b75d8ee /127.0.0.1 consumer3 0 |

--members --verbose — помимо информации, сообщаемой параметром --members, данный параметр также предоставляет партиции, назначенные каждому участнику;

|  |
| --- |
| > bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --describe --group my-group --members --verbose  CONSUMER-ID HOST CLIENT-ID #PARTITIONS ASSIGNMENT  consumer1-3fc8d6f1-581a-4472-bdf3-3515b4aee8c1 /127.0.0.1 consumer1 2 topic1(0), topic2(0)  consumer4-117fe4d3-c6c1-4178-8ee9-eb4a3954bee0 /127.0.0.1 consumer4 1 topic3(2)  consumer2-e76ea8c3-5d30-4299-9005-47eb41f3d3c4 /127.0.0.1 consumer2 3 topic2(1), topic3(0,1)  consumer3-ecea43e4-1f01-479f-8349-f9130b75d8ee /127.0.0.1 consumer3 0 - |

--offsets — данный параметр описания по умолчанию, который обеспечивает тот же вывод, что и параметр --describe;

--state — данный параметр предоставляет полезную информацию на уровне группы.

|  |
| --- |
| > bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --describe --group my-group --state  COORDINATOR (ID) ASSIGNMENT-STRATEGY STATE #MEMBERS  localhost:9092 (0) range Stable 4 |

Чтобы вручную удалить одну или несколько групп потребителей, можно использовать параметр --delete:

|  |
| --- |
| > bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --delete --group my-group --group my-other-group  Deletion of requested consumer groups ('my-group', 'my-other-group') was successful. |

Чтобы сбросить смещения группы потребителей, можно использовать параметр --reset-offsets. Данный параметр поддерживает только одну группу потребителей одновременно. Для параметра необходимо определить параметры: --all-themes или --topic. Должен быть выбран один параметров, если вы не используете сценарий --from-file. Кроме того, сначала убедитесь, что инстансы потребителей неактивны.

Утилита имеет 3 исполняющих параметра:

по умолчанию — для отображения сбрасываемых смещений;

--execute — для выполнения процесса --reset-offsets;

--export — для экспорта результатов в формат CSV.

--reset-offsets также имеет следующие сценарии на выбор (должен быть выбран хотя бы один сценарий):

--to-datetime <String: datetime> — сбрасывает смещения на смещения от datetime. Формат: 'YYYY-MM-DDTHH:mm:SS.sss';

--to-early — сбрасывает смещения до самого раннего смещения;

--to-latest — сбрасывает смещения до последнего смещения;

--shift-by <Long: number-of-offsets> — сбрасывает смещения, сдвигая текущее смещение на n, где n может быть положительным или отрицательным;

--from-file — сбрасывает смещения до значений, определённых в файле CSV;

--to-current — сбрасывает смещения до текущего смещения;

--by-duration <String: duration> — сбрасывает смещения на смещение по продолжительности от текущей метки времени. Формат: PnDTnHnMnS;

--to-offset — сбрасывает смещения до определённого смещения.

Обратите внимание, что смещения вне диапазона будут скорректированы до доступного конца смещения. Например, если конец смещения равен 10, а запрос сдвига смещения равен 15, тогда фактически будет выбрано смещение 10.

Например, чтобы сбросить смещения группы потребителей до последнего смещения, выполните:

|  |
| --- |
| > bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --reset-offsets --group consumergroup1 --topic topic1 --to-latest  TOPIC PARTITION NEW-OFFSET  topic1 0 0 |

Если вы используете устаревшего высокоуровневого потребителя и храните метаданные группы в ZooKeeper (т.е. offsets.storage=zookeeper), передавайте --zookeeper вместо --bootstrap-server:

|  |
| --- |
| > bin/kafka-consumer-groups.sh --zookeeper localhost:2181 --list |

Расширение кластера

Добавить серверы в кластер Kafka легко, просто назначьте им уникальный идентификатор брокера и запустите Kafka на своих новых серверах. Однако этим новым серверам не будут автоматически назначены какие-либо партиции данных, поэтому, если партиции не будут перенесены на них, они не будут выполнять никакой работы, пока не будут созданы новые топики. Поэтому обычно, когда вы добавляете машины в свой кластер, вам нужно перенести на эти машины некоторые существующие данные.

Процесс миграции данных запускается вручную, но полностью автоматизирован. Внутри автоматизации Kafka добавляет новый сервер в качестве подписчика партиции, которую он переносит, и позволяет ему полностью реплицировать существующие данные в этой партиции. Когда новый сервер полностью реплицирует содержимое данной партиции и присоединится к синхронизированной реплике, одна из существующих реплик удалит данные своей партиции.

Утилиту переназначения партиций можно использовать для перемещения партиций между брокерами. Идеальное распределение партиций обеспечит равномерную загрузку данных и размеры партиций для всех брокеров. Утилита переназначения партиций не имеет возможности автоматически изучать распределение данных в кластере Kafka и перемещать партиции для достижения равномерного распределения нагрузки. Таким образом, администратор должен выяснить, какие топики или партиции следует переместить.

Утилита переназначения партиций может работать в 3 взаимоисключающих режимах:

--generate — в данном режиме, передавая перечень топиков и перечень брокеров, утилита генерирует переназначение кандидата, чтобы переместить все партиции указанных топиков новым брокерам. Данный параметр просто обеспечивает удобный способ создания плана переназначения партиций с учётом перечня топиков и целевых брокеров;

--execute — в данном режиме инструмент запускает переназначение партиций на основе предоставленного пользователем плана переназначения. (используя параметр --reassignment-json-file). Это может быть индивидуальный план переназначения, созданный администратором вручную, или план, предоставленный с помощью параметра --generate;

--verify — в данном режиме инструмент верифицирует статус переназначения для всех партиций, перечисленных во время последнего --execute. Статус может быть успешно завершён, не пройден или выполняется.

Автоматическая миграция данных на новые машины

Утилиту переназначения партиций можно использовать для перемещения некоторых топиков из текущего набора брокеров к вновь добавленным брокерам. Обычно это полезно при расширении существующего кластера, поскольку легче перемещать целые партиции в новый набор брокеров, чем перемещать по одной партиции за раз. Для этого пользователь должен предоставить утилите перечень топиков, которые следует переместить в новый набор брокеров, и целевой перечень новых брокеров. Затем утилита равномерно распределяет все партиции для данного перечня топиков по новому набору брокеров. Во время этого перемещения фактор репликации топика остаётся постоянным. Фактически реплики для всех партиций входного перечня топиков перемещаются из старого набора брокеров во вновь добавленные брокеры.

Например, в следующем примере все партиции для топиков foo1, foo2 будут перемещены в новый набор брокеров 5, 6. В конце данного перемещения все партиции для топиков foo1 и foo2 будут существовать только на брокерах 5, 6.

Поскольку утилита принимает входной перечень топиков в виде файла json, сначала необходимо определить топики, которые необходимо переместить, и создать файл json следующим образом:

|  |
| --- |
| > cat topics-to-move.json  {"topics": [{"topic": "foo1"},  {"topic": "foo2"}],  "version":1  } |

После создания файла json используйте инструмент переназначения партиций, чтобы сгенерировать назначение кандидата:

|  |
| --- |
| > bin/kafka-reassign-partitions.sh --bootstrap-server localhost:9092 --topics-to-move-json-file topics-to-move.json --broker-list "5,6" --generate  Current partition replica assignment  {"version":1,  "partitions":[{"topic":"foo1","partition":2,"replicas":[1,2]},  {"topic":"foo1","partition":0,"replicas":[3,4]},  {"topic":"foo2","partition":2,"replicas":[1,2]},  {"topic":"foo2","partition":0,"replicas":[3,4]},  {"topic":"foo1","partition":1,"replicas":[2,3]},  {"topic":"foo2","partition":1,"replicas":[2,3]}]  }  Proposed partition reassignment configuration  {"version":1,  "partitions":[{"topic":"foo1","partition":2,"replicas":[5,6]},  {"topic":"foo1","partition":0,"replicas":[5,6]},  {"topic":"foo2","partition":2,"replicas":[5,6]},  {"topic":"foo2","partition":0,"replicas":[5,6]},  {"topic":"foo1","partition":1,"replicas":[5,6]},  {"topic":"foo2","partition":1,"replicas":[5,6]}]  } |

Утилита генерирует назначение кандидата, которое переместит все партиции из топиков foo1, foo2 на брокеры 5, 6. Обратите внимание, что на данном этапе перемещение партиций ещё не началось, оно просто сообщает вам текущее назначение и предлагаемое новое назначение. Текущее назначение следует сохранить на случай, если вы захотите вернуться к нему. Новое назначение следует сохранить в файле json (например, expand-cluster-reassignment.json) для ввода в утилиту с параметром --execute следующим образом:

|  |
| --- |
| > bin/kafka-reassign-partitions.sh --bootstrap-server localhost:9092 --reassignment-json-file expand-cluster-reassignment.json --execute  Current partition replica assignment  {"version":1,  "partitions":[{"topic":"foo1","partition":2,"replicas":[1,2]},  {"topic":"foo1","partition":0,"replicas":[3,4]},  {"topic":"foo2","partition":2,"replicas":[1,2]},  {"topic":"foo2","partition":0,"replicas":[3,4]},  {"topic":"foo1","partition":1,"replicas":[2,3]},  {"topic":"foo2","partition":1,"replicas":[2,3]}]  }  Save this to use as the --reassignment-json-file option during rollback  Successfully started reassignment of partitions  {"version":1,  "partitions":[{"topic":"foo1","partition":2,"replicas":[5,6]},  {"topic":"foo1","partition":0,"replicas":[5,6]},  {"topic":"foo2","partition":2,"replicas":[5,6]},  {"topic":"foo2","partition":0,"replicas":[5,6]},  {"topic":"foo1","partition":1,"replicas":[5,6]},  {"topic":"foo2","partition":1,"replicas":[5,6]}]  } |

Наконец, параметр --verify можно использовать с утилитой для проверки статуса переназначения партиций. Обратите внимание, что тот же файл expand-cluster-reassignment.json (используемый с параметром --execute) следует использовать с параметром --verify:

|  |
| --- |
| > bin/kafka-reassign-partitions.sh --bootstrap-server localhost:9092 --reassignment-json-file expand-cluster-reassignment.json --verify  Status of partition reassignment:  Reassignment of partition [foo1,0] completed successfully  Reassignment of partition [foo1,1] is in progress  Reassignment of partition [foo1,2] is in progress  Reassignment of partition [foo2,0] completed successfully  Reassignment of partition [foo2,1] completed successfully  Reassignment of partition [foo2,2] completed successfully |

Кастомное назначение партиций и миграция

Утилиту переназначения партиций также можно использовать для выборочного перемещения реплик партиций на определённый набор брокеров. При использовании предполагается, что пользователь знает план переназначения и не требует, чтобы утилита генерировал переназначение кандидата, т.е. фактически пропускает шаг --generate и сразу переходит к шагу --execute.

Например, в следующем примере партиция 0 топика foo1 перемещается на брокеры 5, 6, а партиция 1 топика foo2 — на брокеры 2, 3.

Первым шагом является создание пользовательского плана переназначения в файле json:

|  |
| --- |
| > cat custom-reassignment.json  {"version":1,"partitions":[{"topic":"foo1","partition":0,"replicas":[5,6]},{"topic":"foo2","partition":1,"replicas":[2,3]}]} |

Затем используйте файл json с параметром --execute, чтобы начать процесс переназначения:

|  |
| --- |
| > bin/kafka-reassign-partitions.sh --bootstrap-server localhost:9092 --reassignment-json-file custom-reassignment.json --execute  Current partition replica assignment  {"version":1,  "partitions":[{"topic":"foo1","partition":0,"replicas":[1,2]},  {"topic":"foo2","partition":1,"replicas":[3,4]}]  }  Save this to use as the --reassignment-json-file option during rollback  Successfully started reassignment of partitions  {"version":1,  "partitions":[{"topic":"foo1","partition":0,"replicas":[5,6]},  {"topic":"foo2","partition":1,"replicas":[2,3]}]  } |

Параметр --verify можно использовать с утилитой для проверки статуса переназначения партиций. Обратите внимание, что тот же custom-reassignment.json (используемый с параметром --execute) следует использовать с параметром --verify:

|  |
| --- |
| > bin/kafka-reassign-partitions.sh --bootstrap-server localhost:9092 --reassignment-json-file custom-reassignment.json --verify  Status of partition reassignment:  Reassignment of partition [foo1,0] completed successfully  Reassignment of partition [foo2,1] completed successfully |

Вывод брокеров из эксплуатации

Утилита переназначения партиций не имеет возможности автоматически создавать план переназначения для выводимых брокеров. Таким образом, администратор должен создать план переназначения, чтобы переместить реплику всех партиций, размещённых на брокере, который будет выведен из эксплуатации, на другие брокеры. Обратите внимание, что такое переназначение должно гарантировать, что все реплики не будут перемещены с выводимого брокера только к одному другому брокеру.

Увеличение фактора репликации

Чтобы увеличить фактор репликации существующей партиции, укажите дополнительные реплики в json-файле кастомного переназначения и используйте его с параметром --execute, чтобы увеличить фактор репликации указанных партиций.

Например, в следующем примере фактор репликации партиции 0 топика foo увеличивается с 1 до 3. Перед увеличением фактора репликации единственная реплика партиций располагалась на брокере 5. В рамках увеличения фактора репликации мы добавим дополнительные реплики на брокеры 6 и 7.

Первым шагом является создание кастомного плана переназначения в файле json (например, increase-replication-factor.json):

|  |
| --- |
| > cat increase-replication-factor.json  {"version":1,  "partitions":[{"topic":"foo","partition":0,"replicas":[5,6,7]}]} |

Затем используйте файл json с параметром --execute, чтобы начать процесс переназначения:

|  |
| --- |
| > bin/kafka-reassign-partitions.sh --bootstrap-server localhost:9092 --reassignment-json-file increase-replication-factor.json --execute  Current partition replica assignment  {"version":1,  "partitions":[{"topic":"foo","partition":0,"replicas":[5]}]}  Save this to use as the --reassignment-json-file option during rollback  Successfully started reassignment of partitions  {"version":1,  "partitions":[{"topic":"foo","partition":0,"replicas":[5,6,7]}]} |

Можно использовать параметр --verify с утилитой для проверки статуса переназначения партиций. Обратите внимание, что тот же increase-replication-factor.json (используемый с параметром --execute) следует использовать с параметром --verify:

|  |
| --- |
| > bin/kafka-reassign-partitions.sh --bootstrap-server localhost:9092 --reassignment-json-file increase-replication-factor.json --verify  Status of partition reassignment:  Reassignment of partition [foo,0] completed successfully |

Вы также можете проверить увеличение фактора репликации с помощью утилиты kafka-topics:

|  |
| --- |
| > bin/kafka-topics.sh --bootstrap-server localhost:9092 --topic foo --describe  Topic:foo PartitionCount:1 ReplicationFactor:3 Configs:  Topic: foo Partition: 0 Leader: 5 Replicas: 5,6,7 Isr: 5,6,7 |

Дросселирование полосы пропускания во время миграции данных

Kafka позволяет вам ограничивать трафик репликации, устанавливая верхнюю границу полосы пропускания, используемой для миграции реплик с машины на машину. Такое ограничение может быть полезно при перебалансировке кластера, начальной загрузке нового брокера или добавлении или удалении брокеров, поскольку данные операции с большим объёмом данных могут ограничивать влияние на пользователей.

Есть два интерфейса, которые можно использовать для включения дросселирования. Самым простым и безопасным является применение дросселирования при вызове kafka-reassign-partitions.sh, но kafka-configs.sh также можно использовать для просмотра и изменения значений дросселирования напрямую.

Так, например, если бы вы выполнили перебалансировку с помощью приведённой ниже команды, она мигрировала бы партиции со скоростью не более 50 МБ/с.

|  |
| --- |
| $ bin/kafka-reassign-partitions.sh --bootstrap-server localhost:9092 --execute --reassignment-json-file bigger-cluster.json --throttle 50000000 |

Когда вы выполните этот скрипт, вы увидите, что дросселирование применено:

|  |
| --- |
| The throttle limit was set to 50000000 B/s  Successfully started reassignment of partitions. |

Если вы хотите изменить дросселирование во время перебалансировки, скажем, чтобы увеличить пропускную способность, чтобы он завершился быстрее, вы можете сделать это, повторно запустив команду --execute, передав тот же json-файл переназначения:

|  |
| --- |
| $ bin/kafka-reassign-partitions.sh --bootstrap-server localhost:9092 --execute --reassignment-json-file bigger-cluster.json --throttle 700000000  There is an existing assignment running.  The throttle limit was set to 700000000 B/s |

После завершения перебалансировки администратор может проверить статус перебалансировки с помощью параметра --verify. Если перебалансировка завершена, дросселирование будет выключено с помощью команды --verify. Важно, чтобы администраторы своевременно выключали дросселирование после завершения перебалансировки, выполнив команду с параметром --verify. Если не выключить дросселирование, то дальнейшая репликация будет производиться по дросселированному трафику.

Когда параметр --verify выполнен и переназначение завершено, скрипт подтвердит, что дросселирование отключено:

|  |
| --- |
| > bin/kafka-reassign-partitions.sh --bootstrap-server localhost:9092 --verify --reassignment-json-file bigger-cluster.json  Status of partition reassignment:  Reassignment of partition [my-topic,1] completed successfully  Reassignment of partition [mytopic,0] completed successfully  Throttle was removed. |

Администратор также может провалидировать назначенные конфигурации с помощью kafka-configs.sh. Для управления процессом дросселирования используются две пары конфигураций дросселирования. Первая пара относится к непосредственному значению дросселирования. Она настраивается на уровне брокера с использованием динамических свойств:

|  |
| --- |
| leader.replication.throttled.rate  follower.replication.throttled.rate |

Следующая пара является конфигурационной парой перечисленных наборов дросселируемых реплик, которые конфигурируются для каждого топика:

|  |
| --- |
| leader.replication.throttled.replicas  follower.replication.throttled.replicas |

Значения всех четырёх конфигураций автоматически назначаются kafka-reassign-partitions.sh.

Чтобы просмотреть конфигурацию дросселирования, выполните:

|  |
| --- |
| > bin/kafka-configs.sh --describe --bootstrap-server localhost:9092 --entity-type brokers  Configs for brokers '2' are leader.replication.throttled.rate=700000000,follower.replication.throttled.rate=700000000  Configs for brokers '1' are leader.replication.throttled.rate=700000000,follower.replication.throttled.rate=700000000 |

Ответ показывает, что дросселирование применяется как к лидеру, так и к подписчикам протокола репликации. По умолчанию обеим сторонам назначается одно и то же значение дросселированной пропускной способности.

Чтобы просмотреть перечень дросселированных реплик, выполните:

|  |
| --- |
| > bin/kafka-configs.sh --describe --bootstrap-server localhost:9092 --entity-type topics  Configs for topic 'my-topic' are leader.replication.throttled.replicas=1:102,0:101,  follower.replication.throttled.replicas=1:101,0:102 |

Здесь мы видим, что дросселирование лидера применяется к партиции 1 на брокере 102 и партиции 0 на брокере 101. Аналогичным образом дросселирование подписчика применяется к партиции 1 на брокере 101 и партиции 0 на брокере 102.

По умолчанию kafka-reassign-partitions.sh применяет дросселирование лидера ко всем репликам, существующим до перебалансировки, любая из которых может быть лидером. Также применяется дросселирование подписчиков ко всем пунктам назначения перемещения. Таким образом, если имеется партиция с репликами на брокерах 101, 102, переназначенная на брокеры 102, 103, дросселирование лидера для данной партиции будет применено к брокерам 101, 102, а дросселирование подписчиков будет применено только к брокеру 103.

При необходимости вы также можете использовать переключатель --alter в kafka-configs.sh, чтобы вручную изменить настройки дросселирования.

Безопасное использование дросселируемой репликации

Следует соблюдать осторожность при использовании дросселируемой репликации. В частности:

1. Отключить дросселирование по завершению репликации.

Дросселирование следует своевременно отключать после завершения переназначения (с помощью kafka-reassign-partitions.sh --verify).

Удостовериться в прогрессе репликации.

Если значение дросселирования установлено слишком низкое по сравнению со скоростью входящей записи, репликация может не продвигаться.

Это происходит, когда max(BytesInPerSec) > throttle.

Где BytesInPerSec — это метрика, которая отслеживает пропускную способность записи поставщиков в каждый брокер. throttle — установленное значение дросселирования.

Администратор может отслеживать прогресс репликации во время перебалансировки, используя метрику:

|  |
| --- |
| kafka.server:type=FetcherLagMetrics,name=ConsumerLag,clientId=([-.\w]+),topic=([-.\w]+),partition=([0-9]+) |

Во время репликации задержка должна постоянно уменьшаться. Если показатель не уменьшается, администратор должен увеличить пропускную способность дросселирования, как описано выше в разделе.

Установка квот

Переопределение квот и значений по умолчанию могут быть настроены на уровнях (user, client-id), user или client-id. По умолчанию клиенты получают безлимитную квоту. Можно установить собственные квоты для каждой группы (user, client-id), user или client-id.

Настройте индивидуальную квоту для (user=user1, client-id=clientA):

|  |
| --- |
| > bin/kafka-configs.sh --bootstrap-server localhost:9092 --alter --add-config 'producer\_byte\_rate=1024,consumer\_byte\_rate=2048,request\_percentage=200' --entity-type users --entity-name user1 --entity-type clients --entity-name clientA  Updated config for entity: user-principal 'user1', client-id 'clientA'. |

Настройте кастомную квоту для user=user1:

|  |
| --- |
| > bin/kafka-configs.sh --bootstrap-server localhost:9092 --alter --add-config 'producer\_byte\_rate=1024,consumer\_byte\_rate=2048,request\_percentage=200' --entity-type users --entity-name user1  Updated config for entity: user-principal 'user1'. |

Настройте кастомную квоту для client-id=clientA:

|  |
| --- |
| > bin/kafka-configs.sh --bootstrap-server localhost:9092 --alter --add-config 'producer\_byte\_rate=1024,consumer\_byte\_rate=2048,request\_percentage=200' --entity-type clients --entity-name clientA  Updated config for entity: client-id 'clientA'. |

Можно установить квоты по умолчанию для каждой группы (user, client-id), user или client-id, указав параметр --entity-default вместо --entity-name.

Настройте квоту client-id по умолчанию для user=userA:

|  |
| --- |
| > bin/kafka-configs.sh --bootstrap-server localhost:9092 --alter --add-config 'producer\_byte\_rate=1024,consumer\_byte\_rate=2048,request\_percentage=200' --entity-type users --entity-name user1 --entity-type clients --entity-default  Updated config for entity: user-principal 'user1', default client-id. |

Настройте квоту по умолчанию для user:

|  |
| --- |
| > bin/kafka-configs.sh --bootstrap-server localhost:9092 --alter --add-config 'producer\_byte\_rate=1024,consumer\_byte\_rate=2048,request\_percentage=200' --entity-type users --entity-default  Updated config for entity: default user-principal. |

Настройте квоту по умолчанию для client-id:

|  |
| --- |
| > bin/kafka-configs.sh --bootstrap-server localhost:9092 --alter --add-config 'producer\_byte\_rate=1024,consumer\_byte\_rate=2048,request\_percentage=200' --entity-type clients --entity-default  Updated config for entity: default client-id. |

Вот как описать квоту для выбранных (user, client-id):

|  |
| --- |
| > bin/kafka-configs.sh --bootstrap-server localhost:9092 --describe --entity-type users --entity-name user1 --entity-type clients --entity-name clientA  Configs for user-principal 'user1', client-id 'clientA' are producer\_byte\_rate=1024,consumer\_byte\_rate=2048,request\_percentage=200 |

Описать квоту для выбранного user:

|  |
| --- |
| > bin/kafka-configs.sh --bootstrap-server localhost:9092 --describe --entity-type users --entity-name user1  Configs for user-principal 'user1' are producer\_byte\_rate=1024,consumer\_byte\_rate=2048,request\_percentage=200 |

Описать квоту для выбранного client-id:

|  |
| --- |
| > bin/kafka-configs.sh --bootstrap-server localhost:9092 --describe --entity-type clients --entity-name clientA  Configs for client-id 'clientA' are producer\_byte\_rate=1024,consumer\_byte\_rate=2048,request\_percentage=200 |

Если имя сущности не указано, описываются все сущности указанного типа. Например, описать всех user:

|  |
| --- |
| > bin/kafka-configs.sh --bootstrap-server localhost:9092 --describe --entity-type users  Configs for user-principal 'user1' are producer\_byte\_rate=1024,consumer\_byte\_rate=2048,request\_percentage=200  Configs for default user-principal are producer\_byte\_rate=1024,consumer\_byte\_rate=2048,request\_percentage=200 |

Аналогично для (user, client-id):

|  |
| --- |
| > bin/kafka-configs.sh --bootstrap-server localhost:9092 --describe --entity-type users --entity-type clients  Configs for user-principal 'user1', default client-id are producer\_byte\_rate=1024,consumer\_byte\_rate=2048,request\_percentage=200  Configs for user-principal 'user1', client-id 'clientA' are producer\_byte\_rate=1024,consumer\_byte\_rate=2048,request\_percentage=200 |

Можно установить квоты по умолчанию, которые применяются ко всем client-id, установив данные конфигурации на брокерах. Данные свойства применяются только в том случае, если в Zookeeper не настроены переопределения квот или значения по умолчанию. По умолчанию каждый client-id получает неограниченную квоту. Следующая команда устанавливает квоту по умолчанию для каждого поставщика и потребителя с client-id равной 10 МБ/с.

|  |
| --- |
| quota.producer.default=10485760  quota.consumer.default=10485760 |

Обратите внимание, что данные свойства устарели и могут быть удалены в будущем выпуске. Значения по умолчанию, настроенные с помощью kafka-configs.sh, имеют приоритет над этими свойствами.

АДМИНИСТРИРОВАНИЕ NIFI

Конфигурация порта

NiFi

В следующей таблице перечислены порты по умолчанию, используемые NiFi, и соответствующее свойство в файле nifi.properties.

Таблица 10 — Конфигурация портов NiFi

| Функция | Свойство | Значение по умолчанию |
| --- | --- | --- |
| Порт HTTPS | nifi.web.https.port | 8443 |
| Порт удалённого входа\* | nifi.remote.input.socket.port | 10443 |
| Порт протокола ноды кластера\* | nifi.cluster.node.protocol.port | 11443 |
| Порт балансировки нагрузки ноды кластера | nifi.cluster.node.load.balance.port | 6342 |
| Порт пересылки веб-HTTP | nifi.web.http.port.forwarding | — |

Порты, отмеченные звездочкой (\*), имеют значения свойств, которые по умолчанию пусты в nifi.properties. Значения, показанные в таблице, являются значениями по умолчанию для этих портов, когда TLS Toolkit используется для создания nifi.properties для защищённого инстанса NiFi. Порт центра сертификации по умолчанию, используемый TLS Toolkit, — 9443.

Встроенный ZooKeeper

В следующей таблице перечислены порты по умолчанию, используемые встроенным сервером ZooKeeper, и соответствующее свойство в файле zookeeper.properties.

Таблица 11 — Конфигурация портов встроенного ZooKeeper

| Функция | Свойство | Значение по умолчанию |
| --- | --- | --- |
| Порт клиента ZooKeeper (устарело: порт клиента больше не указывается в отдельной строке с NiFi 1.10.x) | clientPort | 2181 |
| Порты выбранных кворума и лидера сервера ZooKeeper | server.1 | — |

Примеры с комментариями для портов сервера ZooKeeper включены в файл zookeeper.properties в форме server.N=nifi-nodeN-hostname:2888:3888;2181.

Конфигурация безопасности

NiFi предоставляет несколько различных вариантов конфигурации в целях безопасности. Наиболее важные свойства находятся под заголовком security properties в файле nifi.properties. Для безопасной работы необходимо установить свойства, представленные в таблице ниже.

Таблица 12 — Свойства безопасности

| Свойство | Описание |
| --- | --- |
| nifi.security.keystore | Имя файла Keystore, содержащего закрытый ключ сервера. |
| nifi.security.keystoreType | Тип Keystore. Должен быть PKCS12, JKS или BCFKS. JKS является предпочтительным типом, файлы BCFKS и PKCS12 будут загружены с помощью поставщика BouncyCastle. |
| nifi.security.keystorePasswd | Пароль для Keystore. |
| nifi.security.keyPasswd | Пароль для сертификата в Keystore. Если не установлен, будет использоваться значение nifi.security.keystorePasswd. |
| nifi.security.truststore | Имя файла Truststore, которое будет использоваться для авторизации тех, кто подключается к NiFi. Защищённый инстанс без Truststore будет отклонять все входящие соединения. |
| nifi.security.truststoreType | Тип Truststore. Должен быть PKCS12, JKS или BCFKS. JKS является предпочтительным типом, файлы BCFKS и PKCS12 будут загружены с помощью поставщика BouncyCastle. |
| nifi.security.truststorePasswd | Пароль для Truststore. |

После настройки вышеуказанных свойств мы можем разрешить доступ к пользовательскому интерфейсу через HTTPS вместо HTTP. Это достигается установкой свойств nifi.web.https.host и nifi.web.https.port. Свойство nifi.web.https.host указывает, на каком имени хоста должен работать сервер. Если желательно, чтобы интерфейс HTTPS был доступен со всех сетевых интерфейсов, следует использовать значение 0.0.0.0. Чтобы администраторы могли настроить приложение для работы только на определённых сетевых интерфейсах, можно указать свойства nifi.web.http.network.interface\* или nifi.web.https.network.interface\*.

|  |
| --- |
| Примечание. |
| При включении HTTPS важно отключить свойство nifi.web.http.port. NiFi поддерживает работу только по HTTP или HTTPS, но не по обоим одновременно. |

Веб-сервер NiFi требует (REQUIRE) аутентификации клиента на основе сертификата для пользователей, осуществляющих доступ к пользовательскому интерфейсу, если не настроен альтернативный механизм аутентификации, который требует одностороннего SSL (например, LDAP, OpenId Connect и т.д.). Включение альтернативного механизма аутентификации настроит веб-сервер для аутентификации клиента на основе сертификатов WANT. Это позволит ему поддерживать пользователей с сертификатами, а те, у кого нет сертификатов, могут входить в систему с учётными данными.

Теперь, когда пользовательский интерфейс защищён, мы можем легко защитить соединения Site-to-Site, а также связь внутри кластера. Это достигается установкой для свойств nifi.remote.input.secure и nifi.cluster.protocol.is.secure соответственно значения true. Эти коммуникации всегда REQUIRE двустороннего SSL, поскольку ноды будут использовать свои настроенные keystore/truststore для аутентификации.

Автоматическое обновление механизма веб-контекста SSL NiFi можно включить с помощью свойств, представленных в таблице ниже.

Таблица 13 — Свойства безопасности

| Свойство | Описание |
| --- | --- |
| nifi.security.autoreload.enabled | Указывает, должен ли механизм контекста SSL автоматически перезагружаться при обнаружении обновлений в keystore/truststore. По умолчанию установлено значение false. |
| nifi.security.autoreload.interval | Задаёт интервал, с которым keystore/truststore проверяются на наличие обновлений. Применяется, только если для nifi.security.autoreload.enabled установлено значение true. Значение по умолчанию — 10 секунд. |

Как только для свойства nifi.security.autoreload.enabled установлено значение true, любые допустимые изменения в настроенном keystore/truststore вызовут перезагрузку механизма контекста SSL NiFi, позволяя клиентам принимать изменения. Это предназначено для обновления сертификатов с истёкшим сроком действия в keystore и добавления новых доверенных сертификатов в truststore без перезапуска сервера NiFi.

Изменения каких-либо свойств nifi.security.keystore\* или nifi.security.truststore\* не будут улавливаться логикой автоматического обновления, которая предполагает, что пароли и пути к хранилищам останутся прежними.

Набор инструментов для генерации TLS

Чтобы упростить безопасную настройку NiFi, вы можете использовать утилиту командной строки tls-toolkit для автоматической генерации необходимых keystore, truststore и соответствующих файлов конфигурации. Это особенно полезно для защиты нескольких нод NiFi, что может быть утомительным и подверженным ошибкам процессом.

Wildcat-сертификаты

Wildcat-сертификаты (т.е. двум узлам node1.nifi.apache.org и node2.nifi.apache.org назначается один и тот же сертификат с записью CN или SAN \*.nifi.apache.org) официально не поддерживаются и не рекомендуются. Использование wildcat-сертификатов имеет множество недостатков, и и приемлемо только, если каждый сертификат поддерживает дополнительную уникальную запись SAN и запись CN.

Возможные проблемы с wildcat-сертификатами:

Во многих местах кодовой базы при взаимодействии кластера используются идентификаторы сертификатов много раз для идентификации ноды, и если сертификат просто представляет собой DN с подстановочными знаками, это не относится к конкретной ноде.

Администраторам может потребоваться указать кастомный идентификатор ноды в authorizers.xml для \*.nifi.apache.org, поскольку все действия прокси разрешаются только в DN сертификата.

Администраторы не могут отслеживать, на какой ноде было выполнено действие, потому что все они разрешают один и тот же DN.

Администраторы, запускающие несколько инстансов на одном компьютере, используя разные порты для их идентификации, могут случайно поместить имя хоста node1 в порт node2, и адрес будет разрешён, потому что он использует тот же сертификат, но обработчик заголовка хоста заблокирует его, потому что имя хоста node1 не указан как приемлемый хост для инстанса node2.

Если wildcat-сертификат скомпрометирован, скомпрометированы все ноды.

Режимы работы

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

Автономный (Standalone) — генерирует центры сертификации, хранилища ключей, доверенные хранилища и файлы nifi.properties одной командой.

Клиент/сервер (Client/Server) — использует сервер центра сертификации, который принимает запросы на подпись сертификатов от клиентов, подписывает их и отправляет полученные сертификаты обратно. И клиент, и сервер проверяют личность друг друга с помощью общего секрета.

Режим Standalone

Режим Standalone вызывается запуском ./bin/tls-toolkit.sh standalone или bin\tls-toolkit.sh standalone.

Чтобы показать справку:

|  |
| --- |
| ./bin/tls-toolkit.sh standalone -h |

Доступны следующие параметры:

-a, --keyAlgorithm <arg> — алгоритм, используемый для сгенерированных ключей (по умолчанию — RSA);

--additionalCACertificate <arg> — путь к дополнительному сертификату CA (используется для подписи сертификата CA набора инструментов) в формате PEM, если необходимо;

-B, --clientCertPassword <arg> — пароль для сертификата клиента. Должно быть одно значение или одно значение для каждого клиентского DN (автоматически генерируется, если не указано);

-c, --certificateAuthorityHostname <arg> — имя хоста центра сертификации NiFi (по умолчанию — localhost);

-C, --clientCertDn <arg> — сгенерировать сертификат клиента, подходящий для использования в браузере с указанным DN (можно указать несколько раз);

-d, --days <arg> — количество дней, в течение которых выданный сертификат должен быть действителен (по умолчанию — 825);

-f, --nifiPropertiesFile <arg> — базовый файл nifi.properties для обновления (будет использоваться встроенный файл, идентичный тому, который используется при установке NiFi по умолчанию, если не указан);

-g, --differentKeyAndKeystorePasswords — использовать разные сгенерированные пароли для ключа и keystore;

-G, --globalPortSequence <arg> — использовать последовательные порты, которые вычисляются для всех хостов в соответствии с предоставленными выражениями имён хоста (может быть указан несколько раз, должен быть одинаковым от запуска до запуска);

-h, --help — вывести справку и выйти;

-k, --keySize <arg> — количество бит для сгенерированных ключей (по умолчанию — 2048);

-K, --keyPassword <arg> — пароль ключа. Должно быть одно значение или одно значение для каждого хоста (автоматически генерируется, если не указано);

-n, --hostnames <arg> — перечень имён хостов, разделённых запятыми;

--nifiDnPrefix <arg> — строка, добавляемая к именам хостов при определении DN (по умолчанию — CN=);

--nifiDnSuffix <arg> — строка, добавляемая к именам хостов при определении DN (по умолчанию — , OU=NIFI);

-o, --outputDirectory <arg> — каталог для вывода keystore, truststore, файлов конфигурации (по умолчанию — ../bin);

-O, --isOverwrite — перезаписать существующий вывод хоста;

-P, --trustStorePassword <arg> — пароль truststore. Должно быть одно значение или одно значение для каждого хоста (автоматически генерируется, если не указано);

-s, --signingAlgorithm <arg> — алгоритм, используемый для подписи сертификатов (по умолчанию — SHA256WITHRSA);

-S, --keyStorePassword <arg> — пароль keystore. Должно быть одно значение или одно значение для каждого хоста (автоматически генерируется, если не указано);

--subjectAlternativeNames <arg> — разделённый запятыми перечень доменов для использования в качестве альтернативных имён субъектов в сертификате;

-T, --keyStoreType <arg> — тип keystore для генерации (по умолчанию — jks).

Шаблоны Hostname и Subject Alternative Name:

Квадратные скобки можно использовать, чтобы легко указать диапазон имён хостов или альтернативных имён субъектов. Пример: [01-20];

Круглые скобки могут использоваться, чтобы указать, что на данном хосте(ах) будет работать более одного инстанса NiFi. Пример: (5).

Примеры:

1. Создать 4 набора keystore, truststore, nifi.properties для localhost вместе с сертификатом клиента с заданным DN:

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

Создать keystore, truststore, nifi.properties для 10 имён хостов NiFi в каждом из 4 поддоменов:

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

Создать 2 набора keystore, truststore, nifi.properties для 10 имён хостов NiFi в каждом из 4 поддоменов вместе с сертификатом клиента с заданным DN:

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

Та же команда с диапазоном альтернативных имён субъектов:

|  |
| --- |
| bin/tls-toolkit.sh standalone -n 'nifi[01-10].subdomain[1-4].domain(2)' -C 'CN=username,OU=NIFI' --subjectAlternativeNames 'nifi[21-30].other[2-5].example.com(2)' |

Режим Client/Server

В режиме Client/Server для выдачи сертификатов используется центр сертификации (Certificate Authority, CA). Центр сертификации можно остановить, когда вы не подключаете ноды к сети.

Сервер

Режим сервера CA вызывается запуском ./bin/tls-toolkit.sh server или bin\tls-toolkit.sh server.

Чтобы показать справку:

|  |
| --- |
| ./bin/tls-toolkit.sh server -h |

Доступны следующие параметры:

-a, --keyAlgorithm <arg> — алгоритм, используемый для сгенерированных ключей (по умолчанию — RSA);

--configJsonIn <arg> — место, откуда читается информация о конфигурации (по умолчанию значение configJson), подразумевает useConfigJson, если установлено (по умолчанию — значение configJson);

-d, --days <arg> — количество дней, в течение которых выданный сертификат должен быть действителен (по умолчанию — 825);

-D, --dn <arg> — DNS, используемый для сертификата CA (по умолчанию — CN=YOUR\_CA\_HOSTNAME,OU=NIFI);

-f, --configJson <arg> — место для записи информации о конфигурации (по умолчанию — config.json);

-F, --useConfigJson — флаг, указывающий, что вся конфигурация считывается из configJson для облегчения автоматического использования (в противном случае configJson будет только записываться в неё);

-g, --differentKeyAndKeystorePasswords — использовать разные сгенерированные пароли для ключа и keystore;

-h, --help — вывести справку и выйти;

-k, --keySize <arg> — количество бит для сгенерированных ключей (по умолчанию — 2048);

-p, --PORT <arg> — порт для центра сертификации для прослушивания (по умолчанию — 9443);

-s, --signingAlgorithm <arg> — алгоритм, используемый для подписи сертификатов (по умолчанию — SHA256WITHRSA);

-T, --keyStoreType <arg> — тип keystore для генерации (по умолчанию — jks);

-t, --token <arg> — токен, используемый для предотвращения MITM (требуется и должен быть таким же, как тот, который используется клиентами).

Клиент

Клиент может использоваться для запроса новых сертификатов из CA. Клиентская утилита генерирует пару ключей и запрос на подпись сертификата (Certificate Signing Request, CSR) и отправляет CSR в центр сертификации.

Режим клиента CA вызывается при запуске ./bin/tls-toolkit.sh client или bin\tls-toolkit.sh client.

Чтобы показать справку:

|  |
| --- |
| ./bin/tls-toolkit.sh client -h |

Доступны следующие параметры:

-a, --keyAlgorithm <arg> — алгоритм, используемый для сгенерированных ключей (по умолчанию — RSA);

-c, --certificateAuthorityHostname <arg> — имя хоста центра сертификации NiFi (по умолчанию — localhost);

-C, --certificateDirectory <arg> — каталог для записи сертификата CA (по умолчанию — .);

--configJsonIn <arg> — место, откуда читается информация о конфигурации, подразумевает useConfigJson, если установлено (по умолчанию — configJson);

-D, --dn <arg> — DN, который будет использоваться для сертификата клиента (по умолчанию — CN=<localhost name>,OU=NIFI) (автоматически заполняется инструментом);

-f, --configJson <arg> — место для записи информации о конфигурации (по умолчанию — config.json);

-F, --useConfigJson — флаг, указывающий, что вся конфигурация считывается из configJson для облегчения автоматического использования (в противном случае configJson будет записываться только в неё);

-g, --differentKeyAndKeystorePasswords — использовать разные сгенерированные пароли для ключа и keystore;

-h, --help — вывести справку и выйти;

-k, --keySize <arg> — количество бит для сгенерированных ключей (по умолчанию — 2048);

-p, --PORT <arg> — порт для центра сертификации для прослушивания (по умолчанию — 9443);

--subjectAlternativeNames <arg> — разделённый запятыми перечень доменов для использования в качестве альтернативных имён субъектов в сертификате;

-T, --keyStoreType <arg> — тип keystore для генерации (по умолчанию — jks).

-t, --token <arg> — токен, используемый для предотвращения MITM (требуется и должен быть таким же, как тот, который используется СА).

После запуска клиента у вас будет сертификат СА, keystore, truststore и config.json с информацией о них, а также их паролями.

Для сертификата клиента, который можно легко импортировать в браузер, укажите -T PKCS12.

Аутентификация пользователя

NiFi поддерживает аутентификацию пользователей с помощью клиентских сертификатов, имени пользователя и пароля, через Apache Knox или через [OpenId Connect](http://openid.net/connect).

Аутентификация по имени/паролю пользователя выполняется Login Identity Provider. Login Identity Provider — это подключаемый механизм для аутентификации пользователей с помощью их имени и пароля. Какой Login Identity Provider использовать, настраивается в файле nifi.properties. В настоящее время NiFi предлагает аутентификацию по имени/паролю пользователя с параметрами Login Identity Provider для одного пользователя (Single User), облегчённого протокола доступа к каталогам (Lightweight Directory Access Protocol, LDAP) и Kerberos.

Свойство nifi.login.identity.provider.configuration.file указывает файл конфигурации для Login Identity Provider. По умолчанию для данного свойства установлено ./conf/login-identity-providers.xml.

Свойство nifi.security.user.login.identity.provider указывает, какой из настроенных Login Identity Provider следует использовать. Значение данного свойства по умолчанию — single-user-provider, поддерживающий аутентификацию со сгенерированным именем пользователя и паролем.

Во время аутентификации OpenId Connect NiFi будет перенаправлять пользователей для входа в систему с провайдером, прежде чем вернуться в NiFi. Затем NiFi позвонит поставщику, чтобы узнать личность пользователя.

Во время аутентификации Apache Knox NiFi перенаправляет пользователей для входа в Apache Knox, прежде чем вернуться в NiFi. NiFi проверит токен Apache Knox во время аутентификации.

|  |
| --- |
| Примечание. |
| NiFi можно настроить только для имени/пароля пользователя, OpenId Connect или Apache Knox. Он не поддерживает одновременный запуск каждого из них. NiFi потребует клиентские сертификаты для аутентификации пользователей по HTTPS, если ни один из них не настроен. |

Пользователь не может анонимно аутентифицироваться с помощью защищённого инстанса NiFi, если для параметра nifi.security.allow.anonymous.authentication установлено значение true. В этом случае NiFi также должен быть настроен с Authorizer, который поддерживает авторизацию анонимного пользователя. В настоящее время NiFi не поставляется с Authorizer, поддерживающими это.

При настройке nifi.security.allow.anonymous.authentication следует учитывать три сценария. Когда пользователь напрямую вызывает конечную точку без попытки аутентификации, тогда nifi.security.allow.anonymous.authentication будет контролировать, аутентифицирован ли запрос или отклонен. Два других сценария — это когда запрос проксируется. Запрос может быть проксирован нодой NiFi (например, нодой в кластере NiFi) или отдельным прокси, который передаёт запрос анонимному пользователю. В этих сценариях прокси-сервера nifi.security.allow.anonymous.authentication будет контролировать, аутентифицирован ли запрос или отклонен. Во всех трёх сценариях, если запрос аутентифицирован, он впоследствии будет подвергнут обычной авторизации на основе запрошенного ресурса.

|  |
| --- |
| Примечание. |
| NiFi не выполняет аутентификацию пользователя через HTTP. Используя HTTP, всем пользователям будут предоставлены все роли. |

Single User

Single User Login Identity Provider по умолчанию поддерживает автоматическую генерацию учётных данных для имени и пароля пользователя.

Сгенерированное имя пользователя будет случайным UUID, состоящим из 36 символов. Сгенерированный пароль будет представлять собой случайную строку, состоящую из 32 символов и сохраненную с использованием хэширования bcrypt.

Конфигурация по умолчанию в nifi.properties включает аутентификацию Single User:

|  |
| --- |
| nifi.security.user.login.identity.provider=single-user-provider |

По умолчанию файл login-identity-sizes.xml включает бланк определения поставщика:

|  |
| --- |
| <provider>  <identifier>single-user-provider</identifier>  <class>org.apache.nifi.authentication.single.user.SingleUserLoginIdentityProvider</class>  <property name="Username"/>  <property name="Password"/>  </provider> |

Для изменения имени пользователя и пароля можно использовать следующую команду:

|  |
| --- |
| $ ./bin/nifi.sh set-single-user-credentials <username> <password> |

LDAP

Ниже приведены пример и описание настройки Login Identity Provider, который интегрируется с Directory Server для аутентификации пользователей.

Установите следующие параметры в nifi.properties, чтобы включить аутентификацию по имени и паролю пользователя LDAP:

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

Измените login-identity-providers.xml, чтобы включить ldap-provider. Вот образец, представленный в файле:

|  |
| --- |
| <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> |

ldap-provider имеет свойства, представленные в таблице ниже.

Таблица 14 — Свойства ldap-provider

| Свойство | Описание |
| --- | --- |
| Authentication Strategy | Как аутентифицируется соединение с сервером LDAP. Возможные значения: ANONYMOUS, SIMPLE, LDAPS или START\_TLS. |
| Manager DN | DN менеджера, который используется для привязки к серверу LDAP для поиска пользователей. |
| Manager Password | Пароль менеджера, который используется для привязки к серверу LDAP для поиска пользователей. |
| TLS - Keystore | Путь к Keystore, который используется при подключении к LDAP с помощью LDAPS или START\_TLS. |
| TLS - Keystore Password | Пароль для Keystore, который используется при подключении к LDAP с помощью LDAPS или START\_TLS. |
| TLS - Keystore Type | Тип Keystore, который используется при подключении к LDAP с использованием LDAPS или START\_TLS (например, JKS или PKCS12). |
| TLS - Truststore | Путь к Truststore, который используется при подключении к LDAP с помощью LDAPS или START\_TLS. |
| TLS - Truststore Password | Пароль для Truststore, который используется при подключении к LDAP с помощью LDAPS или START\_TLS. |
| TLS - Truststore Type | Тип Truststore, который используется при подключении к LDAP с использованием LDAPS или START\_TLS (например, JKS или PKCS12). |
| TLS - Client Auth | Политика аутентификации клиента при подключении к LDAP с помощью LDAPS или START\_TLS. Возможные значения — REQUIRED, WANT, NONE. |
| TLS - Protocol | Протокол для использования при подключении к LDAP с помощью LDAPS или START\_TLS. Например, TLS, TLSv1.1, TLSv1.2 и т.д. |
| TLS - Shutdown Gracefully | Указывает, следует ли корректно завершить работу TLS перед закрытием целевого контекста. По умолчанию — false. |
| Referral Strategy | Стратегия работы с рефералами. Возможные значения — FOLLOW, IGNORE, THROW. |
| Connect Timeout | Продолжительность тайм-аута подключения. Т.е. 10 секунд. |
| Read Timeout | Продолжительность тайм-аута чтения. Т.е. 10 секунд. |
| Url | Перечень URL-адресов серверов LDAP, разделённых пробелами (например, ldap://<hostname>:<port>). |
| User Search Base | Базовое DN для поиска пользователей (например, CN=Users,DC=example,DC=com). |
| User Search Filter | Фильтр для поиска пользователей по User Search Base. (т.е. sAMAccountName={0}). Имя, указанное пользователем, вставляется в {0}. |
| Identity Strategy | Стратегия идентификации пользователей. Возможные значения — USE\_DN и USE\_USERNAME. Функциональность по умолчанию, если это свойство отсутствует, — USE\_DN, чтобы сохранить обратную совместимость. USE\_DN будет использовать полное DN записи пользователя, если это возможно. USE\_USERNAME будет использовать имя пользователя, под которым вошёл в систему. |
| Authentication Expiration | Продолжительность, в течение которой действительна аутентификация пользователя. Если пользователь никогда не выходит из системы, он должен будет снова войти в систему по истечении установленного времени. |

|  |
| --- |
| Примечание. |
| Чтобы изменения, внесенные в nifi.properties и login-identity-sizes.xml, вступили в силу, необходимо перезапустить NiFi. Если NiFi кластеризован, файлы конфигурации должны быть одинаковыми на всех нодах. |

Kerberos

Ниже приведены пример и описание настройки Login Identity Provider, который интегрируется с Kerberos Key Distribution Center (KDC) для аутентификации пользователей.

Установите следующие параметры в nifi.properties, чтобы включить аутентификацию по имени пользователя и паролю Kerberos:

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

Измените login-identity-providers.xml, чтобы включить kerberos-provider. Вот образец, представленный в файле:

|  |
| --- |
| <provider>  <identifier>kerberos-provider</identifier>  <class>org.apache.nifi.kerberos.KerberosProvider</class>  <property name="Default Realm">NIFI.APACHE.ORG</property>  <property name="Authentication Expiration">12 hours</property>  </provider> |

kerberos-provider имеет свойства, представленные в таблице ниже.

Таблица 15 — Свойства kerberos-provider

| Свойство | Описание |
| --- | --- |
| Default Realm | Область по умолчанию, предоставляемая, когда пользователь вводит неполного принципала-пользователя (например, NIFI.APACHE.ORG). |
| Authentication Expiration | Продолжительность, в течение которой действительна аутентификация пользователя. Если пользователь никогда не выходит из системы, он должен будет снова войти в систему по истечении установленного времени. |

|  |
| --- |
| Примечание. |
| Чтобы изменения, внесенные в nifi.properties и login-identity-sizes.xml, вступили в силу, необходимо перезапустить NiFi. Если NiFi кластеризован, файлы конфигурации должны быть одинаковыми на всех нодах. |

OpenId Connect

Чтобы включить аутентификацию через OpenId Connect, в nifi.properties необходимо настроить свойства, представленные в таблице ниже.

Таблица 16 — Свойства OpenId Connect

| Свойство | Описание |
| --- | --- |
| nifi.security.user.oidc.discovery.url | URL обнаружения для желаемого OpenId Connect Provider (<http://openid.net/specs/openid-connect-discovery-1_0.html>). |
| nifi.security.user.oidc.connect.timeout | Тайм-аут подключения при обмене данными с OpenId Connect Provider. |
| nifi.security.user.oidc.read.timeout | Тайм-аут чтения при обмене данными с OpenId Connect Provider. |
| nifi.security.user.oidc.client.id | Идентификатор клиента для NiFi после регистрации у OpenId Connect Provider. |
| nifi.security.user.oidc.client.secret | Секрет клиента для NiFi после регистрации у OpenId Connect Provider. |
| nifi.security.user.oidc.preferred.jwsalgorithm | Предпочтительный алгоритм валидации токенов идентификации. Если это значение пустое, по умолчанию будет использоваться RS256, который должен поддерживаться OpenId Connect Provider в соответствии со спецификацией. Если это значение HS256, HS384 или HS512, NiFi пытается валидировать токены, защищённые HMAC, используя указанный секрет клиента. Если это значение равно none, NiFi пытается валидировать незащищённые/простые токены. Другие значения этого алгоритма будут пытаться анализировать как алгоритм RSA или EC, который будет использоваться вместе с JSON Web Key (JWK), предоставленным через jwks\_uri в метаданных, найденных по URL-адресу обнаружения. |
| nifi.security.user.oidc.additional.scopes | Области, разделённые запятыми, отправляются OpenId Connect Provider в дополнение к openid и email. |
| nifi.security.user.oidc.claim.identifying.user | Заявление, которое идентифицирует пользователя, который должен войти в систему. По умолчанию — email. Может потребоваться запросить через nifi.security.user.oidc.additional.scopes перед использованием. |
| nifi.security.user.oidc.fallback.claims.identifying.user | Разделённые запятыми возможные резервные утверждения, используемые для идентификации пользователя в случае, если утверждение nifi.security.user.oidc.claim.identifying.user отсутствует для пользователя, вошедшего в систему. |

SAML

Чтобы включить аутентификацию через SAML, в nifi.properties необходимо настроить свойства, представленные в таблице ниже.

Таблица 17 — Свойства SAML

| Свойство | Описание |
| --- | --- |
| nifi.security.user.saml.idp.metadata.url | URL-адрес для получения метаданных поставщика удостоверений. Метаданные можно получить от поставщика удостоверений через http:// или https://, или на локальный файл можно ссылаться с помощью file://. |
| nifi.security.user.saml.sp.entity.id | Идентификатор объекта поставщика услуг (например, NiFi). Это значение будет использоваться в качестве эмитента для запросов аутентификации SAML и должно быть действительным URI. В некоторых случаях идентификатор объекта поставщика услуг должен быть зарегистрирован заранее у поставщика удостоверений. |
| nifi.security.user.saml.identity.attribute.name | Имя атрибута утверждения SAML, содержащего личность пользователя. Это свойство является необязательным и, если не указано или если атрибут не найден, будет использоваться NameID Subject. |
| nifi.security.user.saml.group.attribute.name | Имя атрибута утверждения SAML, содержащего имена групп, к которым принадлежит пользователь. Это свойство является необязательным, но если оно заполнено, группы будут переданы в процесс авторизации. |
| nifi.security.user.saml.metadata.signing.enabled | Включает подпись сгенерированных метаданных поставщика услуг. |
| nifi.security.user.saml.request.signing.enabled | Управляет значением AuthnRequestsSigned в сгенерированных метаданных поставщика услуг из nifi-api/access/saml/metadata. Это указывает на то, что поставщик услуг (например, NiFi) не должен подписывать запросы аутентификации, отправленные поставщику удостоверений, но запросы, возможно, все равно необходимо подписывать, если поставщик удостоверений указывает WantAuthnRequestSigned=true. |
| nifi.security.user.saml.want.assertions.signed | Управляет значением WantAssertionsSigned в сгенерированных метаданных поставщика услуг из nifi-api/access/saml/metadata. Это указывает на то, что поставщик удостоверений должен подписывать утверждения, но некоторые поставщики удостоверений могут предоставлять свою собственную конфигурацию для контроля того, подписываются ли утверждения. |
| nifi.security.user.saml.signature.algorithm | Алгоритм, используемый при подписании сообщений SAML. Обратитесь к [Open SAML Signature Constants](https://git.shibboleth.net/view/?p=java-xmltooling.git;a=blob;f=src/main/java/org/opensaml/xml/signature/SignatureConstants.java) для получения списка допустимых значений. Если не указано, по умолчанию будет использоваться SHA-256. |
| nifi.security.user.saml.signature.digest.algorithm | Алгоритм дайджеста для использования при подписании сообщений SAML. Обратитесь к [Open SAML Signature Constants](https://git.shibboleth.net/view/?p=java-xmltooling.git;a=blob;f=src/main/java/org/opensaml/xml/signature/SignatureConstants.java) для получения списка допустимых значений. Если не указано, по умолчанию будет использоваться SHA-256. |
| nifi.security.user.saml.message.logging.enabled | Включает регистрацию сообщений SAML для целей отладки. |
| nifi.security.user.saml.authentication.expiration | Истечение срока действия NiFi JWT, который будет создан в результате успешного ответа аутентификации SAML. |
| nifi.security.user.saml.single.logout.enabled | Включает SAML SingleLogout, который вызывает выход из системы NiFi для выхода из системы провайдера идентификации. По умолчанию при выходе из системы NiFi удаляется только NiFi JWT. |
| nifi.security.user.saml.http.client.truststore.strateg | Стратегия хранилища доверенных сертификатов, когда URL-адрес метаданных IDP начинается с https. Значение JDK указывает на использование хранилища доверенных сертификатов JDK по умолчанию. Значение NIFI указывает на использование хранилища доверенных сертификатов, указанного в nifi.security.truststore. |
| nifi.security.user.saml.http.client.connect.timeout | Тайм-аут соединения при обмене данными с SAML IDP. |
| nifi.security.user.saml.http.client.read.timeout | Тайм-аут чтения при обмене данными с SAML IDP. |

Apache Knox

Чтобы включить аутентификацию через Apache Knox, в nifi.properties необходимо настроить свойства, представленные в таблице ниже.

Таблица 18 — Свойства 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, который будет использоваться для проверки подписей токенов аутентификации в HTTP Cookie. |
| nifi.security.user.knox.cookieName | Имя HTTP-файла cookie, который Apache Knox сгенерирует после успешного входа в систему. |
| nifi.security.user.knox.audiences | Опционально. Перечень разрешённых подключений разделён запятыми. Если значение установлено, подключение должен присутствовать в данном перечне. Подключения, заполняемые токеном, можно настроить в Knox. |

Системные свойства

Файл nifi.properties в каталоге conf является основным файлом конфигурации для управления работой NiFi. В этом разделе представлен обзор свойств в этом файле и их параметров настройки.

|  |
| --- |
| Примечание. |
| Значения для периодов времени и размеров данных должны включать единицу измерения, например 10 secs или 10 MB, а не просто 10. |

|  |
| --- |
| Примечание. |
| После внесения изменений в nifi.properties перезапустите NiFi, чтобы изменения вступили в силу. |

Рекомендации по обновлению

Содержимое файла nifi.properties относительно стабильно, но может меняться от версии к версии. Всегда рекомендуется просматривать данный файл при обновлении и обращать внимание на любые изменения.

Рассмотрите возможность настройки элементов, отмеченных звездочкой (\*) ниже, таким образом, чтобы было проще выполнить обновление. Например, измените конфигурации каталога по умолчанию на расположение вне основной корневой установки. Таким образом, эти элементы могут оставаться в своём настроенном месте при обновлении, позволяя NiFi находить все репозитории и файлы конфигурации и продолжать работу с того места, где они были остановлены, как только старая версия останавливается и запускается новая версия. Кроме того, администратор может повторно использовать этот файл nifi.properties и любые другие файлы конфигурации без необходимости заново настраивать их каждый раз при обновлении.

Основные свойства

Первый раздел файла nifi.properties предназначен для основных свойств. Эти свойства применяются к основной структуре в целом.

Таблица 19 — Основные свойства nifi.properties

| Свойство | Описание |
| --- | --- |
| nifi.flow.configuration.file\* | Расположение файла конфигурации flow (т.е. файла, который содержит то, что в настоящее время отображается на графе NiFi). Значение по умолчанию — ./conf/flow.xml.gz. |
| nifi.flow.configuration.archive.enabled\* | Указывает, создает ли NiFi резервную копию flow автоматически при обновлении flow. Значение по умолчанию true. |
| nifi.flow.configuration.archive.dir\* | Расположение каталога архива, в котором сохраняются резервные копии файла flow.xml. Значение по умолчанию — ./conf/archive. NiFi удаляет старые архивные файлы, чтобы ограничить использование диска в зависимости от продолжительности жизни, общего размера и количества файлов в архиве, как указано в свойствах nifi.flow.configuration.archive.max.time, max.storage и max.count соответственно. Если ни одно из этих ограничений для архивирования не указано, NiFi использует условия по умолчанию, то есть 30 days для max.time и 500 MB для max.storage.  Этот механизм очистки учитывает только автоматически созданные заархивированные файлы flow.xml. Если в этом каталоге архива есть другие файлы или каталоги, NiFi проигнорирует их. Автоматически созданные архивы имеют имя файла с префиксом метки времени в формате ISO 8601, за которым следует <original-filename>. То есть <year><month><day>T<hour><minute><second>+<timezone offset>\_<original filename>. Например, 20160706T160719+0900\_flow.xml.gz. NiFi проверяет имена файлов при очистке каталога архива. Если вы хотите сохранить определённый архив в этом каталоге, не беспокоясь о его удалении NiFi, вы можете сделать это, скопировав его с другим шаблоном имени файла. |
| nifi.flow.configuration.archive.max.time\* | Срок службы заархивированных файлов flow.xml. Если это свойство указано, NiFi удалит архивные файлы с истёкшим сроком действия при обновлении файла flow.xml. Срок действия определяется на основе текущего системного времени и последней изменённой отметки времени заархивированного файла flow.xml. Если в nifi.properties не указано никаких ограничений архива, NiFi удаляет архивы старше 30 days. |
| nifi.flow.configuration.archive.max.storage\* | Общий размер данных, разрешённый для заархивированных файлов flow.xml. NiFi будет удалять самые старые архивные файлы до тех пор, пока общий размер архивных файлов не станет меньше, чем это значение конфигурации, если это свойство указано. Если в nifi.properties не указано никаких ограничений архива, NiFi использует для этого 500 MB. |
| nifi.flow.configuration.archive.max.count\* | Допустимое количество архивных файлов. NiFi удалит самые старые архивные файлы, чтобы можно было сохранить только N последних архивов, если это свойство указано. |
| nifi.flowcontroller.autoResumeState | Указывает, должны ли при перезапуске компоненты на графе NiFi вернуться в своё последнее состояние. Значение по умолчанию — true. |
| nifi.flowcontroller.graceful.shutdown.period | Указывает период выключения. Значение по умолчанию — 10 secs. |
| nifi.flowservice.writedelay.interval | Когда в файл flow.xml вносится много изменений, это свойство указывает, как долго ждать перед записью изменений, чтобы внести изменения в одну запись. Значение по умолчанию — 500 ms. |
| nifi.administrative.yield.duration | Если компонент позволяет избежать неожиданного исключения, это считается ошибкой. В результате фреймворк приостанавливает (или административно уступает) компонент на это время. Это сделано для того, чтобы компонент не использовал огромное количество системных ресурсов, поскольку известно, что он имеет проблемы в существующем состоянии. Значение по умолчанию — 30 secs. |
| nifi.bored.yield.duration | Когда у компонента нет работы, это время, в течение которого он будет ждать, прежде чем проверять, есть ли у него новые данные для работы. Таким образом, он не использует ресурсы ЦП, слишком часто проверяя наличие новой работы. При установке этого свойства имейте в виду, что это может добавить дополнительную задержку для компонентов, у которых нет постоянной работы, поскольку, когда они перейдут в это «скучающее» состояние, они будут ждать это количество времени, прежде чем проверять наличие дополнительной работы. Значение по умолчанию — 10 ms. |
| nifi.queue.backpressure.count | При рисовании нового соединения между двумя компонентами это значение по умолчанию для порогового значения объекта обратного давления этого соединения. По умолчанию 10000, и значение должно быть целым числом. |
| nifi.queue.backpressure.size | При рисовании нового соединения между двумя компонентами это значение по умолчанию для порогового значения размера данных обратного давления для этого соединения. Значение по умолчанию — 1 GB, и значение должно быть размером данных, включая единицу измерения. |
| nifi.authorizer.configuration.file\* | Это расположение файла, который определяет, как определяются авторизаторы. Значение по умолчанию — ./conf/authorizers.xml. |
| nifi.login.identity.provider.configuration.file\* | Это расположение файла, который определяет, как выполняется аутентификация по имени пользователя и паролю. Этот файл учитывается, только если nifi.security.user.login.identity.provider настроен с идентификатором поставщика. Значение по умолчанию — ./conf/login-identity-providers.xml. |
| nifi.templates.directory\* | Это расположение каталога, в котором сохраняются шаблоны flow (только для обратной совместимости). Шаблоны хранятся в flow.xml.gz, начиная с версии NiFi 1.0. Каталог шаблонов можно использовать для (массового) импорта шаблонов в flow.xml.gz автоматически при запуске NiFi. Значение по умолчанию — ./conf/templates. |
| nifi.ui.banner.text | Это текст баннера, который можно настроить для отображения в верхней части пользовательского интерфейса. По умолчанию он пуст. |
| nifi.ui.autorefresh.interval | Интервал автоматического обновления пользовательского интерфейса. Значение по умолчанию — 30 secs. |
| nifi.nar.library.directory | Расположение nar-библиотеки. Значение по умолчанию — ./lib, и его, вероятно, следует оставить как есть.  **Примечание**. Дополнительные каталоги библиотеки можно указать с помощью префикса nifi.nar.library.directory. с уникальными суффиксами и отдельными путями в качестве значений.  Например, чтобы предоставить два дополнительных местоположения библиотеки, пользователь может также указать дополнительные свойства с ключами:  nifi.nar.library.directory.lib1=/nars/lib1  nifi.nar.library.directory.lib2=/nars/lib2  Предоставляется три общих местоположения, включая nifi.nar.library.directory. |
| nifi.nar.working.directory | Расположение рабочего nar-каталога. Значение по умолчанию — ./work/nar, и его, вероятно, следует оставить как есть. |
| nifi.documentation.working.directory | Рабочий каталог документации. Значение по умолчанию — ./work/docs/components, и его, вероятно, следует оставить как есть. |
| nifi.processor.scheduling.timeout | Время ожидания завершения операции жизненного цикла процессора (@OnScheduled и @OnUnscheduled) до того, как можно будет вызвать другую операцию жизненного цикла (например, остановку). Значение по умолчанию — 1 min. |

Управление состояниями

Раздел управления состояниями файла свойств предоставляет механизм для настройки локальных и общекластерных механизмов сохранения состояния компонентов.

Таблица 20 — Свойства управления статусами

| Свойство | Описание |
| --- | --- |
| nifi.state.management.configuration.file | XML-файл, содержащий конфигурацию для локальных и общекластерных State Providers. Значение по умолчанию — ./conf/state-management.xml. |
| nifi.state.management.provider.local | Идентификатор используемого Local State Provider. Установленное значение должно соответствовать значению элемента id одного из элементов local-provider в файле state-management.xml. |
| nifi.state.management.provider.cluster | Идентификатор используемого Cluster State Provider. Установленное значение должно соответствовать значению элемента id одного из элементов cluster-provider в файле state-management.xml. Значение игнорируется, если кластеризация отсутствует, но требуется для нод в кластере. |
| nifi.state.management.embedded.zookeeper.start | Определяет, должен ли этот инстанс NiFi запускать встроенный сервер ZooKeeper. Используется вместе с ZooKeeperStateProvider. |
| nifi.state.management.embedded.zookeeper.properties | Задаёт файл свойств, содержащий конфигурацию для запущенного встроенного сервера ZooKeeper (если для свойства nifi.state.management.embedded.zookeeper.start установлено значение true). |

Конфигурация H2

Раздел конфигурации H2 определяет настройки для базы данных H2, которая отслеживает доступ пользователей и историю контроллера flow.

Таблица 21 — Свойства конфигурации H2

| Свойство | Описание |
| --- | --- |
| nifi.database.directory\* | Расположение каталога базы данных H2. Значение по умолчанию — ./database\_repository. |
| nifi.h2.url.append | Свойство определяет дополнительные аргументы для добавления в строку подключения для базы данных H2. Следует использовать значение по умолчанию, и его не следует изменять. Это — ;LOCK\_TIMEOUT=25000;WRITE\_DELAY=0;AUTO\_SERVER=FALSE. |

Репозиторий FlowFile

Репозиторий FlowFile отслеживает атрибуты и текущее состояние каждого FlowFile в системе. По умолчанию этот репозиторий устанавливается в том же корневом каталоге установки, что и все другие репозитории; однако рекомендуется настроить его на отдельном диске, если это возможно.

В настоящее время существует три реализации репозитория FlowFile, которые подробно описаны ниже.

Таблица 22 — Свойства репозитория FlowFile

| Свойство | Описание |
| --- | --- |
| nifi.flowfile.repository.implementation | Имплементация репозитория FlowFile. Значение по умолчанию — org.apache.nifi.controller.repository.WriteAheadFlowFileRepository. Другие текущие параметры — org.apache.nifi.controller.repository.VolatileFlowFileRepository и org.apache.nifi.controller.repository.RocksDBFlowFileRepository. |

|  |
| --- |
| Примечание. |
| Переключение имплементация репозитория должно выполняться только на инстансе с нулевыми очередями FlowFiles и должно выполняться только с осторожностью. |

Репозиторий Write Ahead FlowFile

WriteAheadFlowFileRepository — это имплементация по умолчанию. Он сохраняет FlowFiles на диске и при желании может быть настроен для синхронизации всех изменений на диске. Он очень дорогой и может значительно снизить производительность NiFi. Однако, если установлено false, существует вероятность потери данных в случае внезапного отключения питания или сбоя операционной системы. Значение по умолчанию — false.

Таблица 23 — Свойства репозитория Write Ahead FlowFile

| Свойство | Описание |
| --- | --- |
| nifi.flowfile.repository.wal.implementation | Если имплементация репозитория настроена на использование WriteAheadFlowFileRepository, это свойство можно использовать, чтобы указать, какую имплементацию Write-Ahead Log следует использовать. Значение по умолчанию — org.apache.nifi.wali.SequentialAccessWriteAheadLog. Эта версия лога упреждающей записи была добавлена ​​в Apache NiFi версии 1.6.0 и была разработана для решения проблемы, которая существует в более старой имплементации. В случае отключения питания или сбоя операционной системы старая имплементация могла неправильно восстанавливать FlowFiles. Это могло потенциально привести к неправильным атрибутам или содержимому, назначенным для FlowFile при перезапуске, после потери питания или сбоя ОС. Однако при желании можно выбрать вариант использования предыдущей имплементации и принять этот риск (например, если в новой имплементации возникнет непредвиденная ошибка). Для этого задайте для данного свойства значение org.wali.MinimalLockingWriteAheadLog. Другая доступная имплементация — org.apache.nifi.wali.EncryptedSequentialAccessWriteAheadLog. Если значение данного свойства изменяется, при перезапуске NiFi всё равно будет восстанавливать записи, записанные с использованием ранее настроенного репозитория, и удалять файлы, записанные ранее настроенной имплементацией. |
| nifi.flowfile.repository.directory\* | Расположение репозитория FlowFile. Значение по умолчанию — ./flowfile\_repository. |
| nifi.flowfile.repository.checkpoint.interval | Интервал контрольной точки репозитория FlowFile. Значение по умолчанию — 2 mins. |
| nifi.flowfile.repository.always.sync | Если установлено значение true, любые изменения в репозитории будут синхронизироваться с диском, а это означает, что NiFi попросит операционную систему не кэшировать информацию. Это очень дорого и может значительно снизить производительность NiFi. Однако, если установлено false, существует вероятность потери данных в случае внезапного отключения питания или сбоя операционной системы. Значение по умолчанию false. |

Репозиторий Encrypted Write Ahead FlowFile

Все свойства, определённые выше в п. 2.4.5.1, применяются. Здесь перечислены только свойства, относящиеся к шифрованию.

|  |
| --- |
| Примечание. |
| В отличие от зашифрованных content и provenance репозиториев, здесь не меняется имплементация репозитория, а только базовая имплементация лога упреждающей записи. Это обеспечивает более чёткое разделение и большую гибкость при выборе имплементации. Для включения шифрования необходимо изменить свойство nifi.flowfile.repository.wal.implementation. |

Таблица 24 — Свойства зашифрованного репозитория Write Ahead FlowFile

| Свойство | Описание |
| --- | --- |
| nifi.flowfile.repository.encryption.key.provider.implementation | Это полное имя класса поставщика ключей (key provider). Поставщик ключей — это интерфейс хранилища данных для доступа к ключу шифрования для защиты заявок на контент. В настоящее время существует три имплементации: StaticKeyProvider, который считывает ключ непосредственно из nifi.properties, FileBasedKeyProvider, который считывает ключи из зашифрованного файла, и KeyStoreKeyProvider, который считывает ключи из стандартного java.security.KeyStore. |
| nifi.flowfile.repository.encryption.key.provider.location | Путь к ресурсу определения ключа (пустой для StaticKeyProvider, ./keys.nkp или аналогичный путь для FileBasedKeyProvider). Для будущих поставщиков, таких как HSM, это может быть строка подключения или URL-адрес. |
| nifi.flowfile.repository.encryption.key.provider.password | Пароль, используемый для расшифровки ресурса определения ключа, например keystore для KeyStoreKeyProvider. |
| nifi.flowfile.repository.encryption.key.id | Идентификатор активного ключа, который будет использоваться для шифрования (например, Key1). |
| nifi.flowfile.repository.encryption.key | Ключ, который нужно использовать для StaticKeyProvider. Формат ключа закодирован в шестнадцатеричном формате (0123456789ABCDEFFEDCBA98765432100123456789ABCDEFFEDCBA9876543210), но также может быть зашифрован с помощью инструмента ./encrypt-config.sh в NiFi Toolkit. |
| nifi.flowfile.repository.encryption.key.id.\* | Позволяет указать дополнительные ключи для StaticKeyProvider. Например, строка nifi.flowfile.repository.encryption.key.id.Key2=012…​210 предоставит доступный ключ Key2. |

Самая простая конфигурация представлена ниже:

|  |
| --- |
| nifi.flowfile.repository.implementation=org.apache.nifi.controller.repository.WriteAheadFlowFileRepository  nifi.flowfile.repository.wal.implementation=org.apache.nifi.wali.EncryptedSequentialAccessWriteAheadLog  nifi.flowfile.repository.encryption.key.provider.implementation=org.apache.nifi.security.kms.StaticKeyProvider  nifi.flowfile.repository.encryption.key.provider.location=  nifi.flowfile.repository.encryption.key.id=Key1  nifi.flowfile.repository.encryption.key=0123456789ABCDEFFEDCBA98765432100123456789ABCDEFFEDCBA9876543210 |

Репозиторий Volatile FlowFile

Данная имплементация хранит FlowFiles в памяти, а не на диске. Это может привести к потере данных в случае сбоя питания/машины или перезапуска NiFi. Чтобы использовать данную имплементацию, установите для nifi.flowfile.repository.implementation значение org.apache.nifi.controller.repository.VolatileFlowFileRepository.

Репозиторий RocksDB FlowFile

Данная имплементация использует хранилище key-value RocksDB. Она использует периодическую синхронизацию, чтобы гарантировать, что созданные или полученные данные не будут потеряны (если для nifi.flowfile.repository.rocksdb.accept.data.loss установлено значение false). В случае сбоя (например, потери питания) работа, выполняемая с FlowFiles через систему (например, маршрутизация и трансформация), всё равно может быть потеряна. В частности, запись этих действий может быть потеряна, что приведёт к возврату затронутых FlowFiles в предыдущее допустимое состояние. Оттуда они возобновят свой путь через flow в обычном режиме. Данная гарантия предоставляется за счёт задержки операций по добавлению новых данных в систему. Эта задержка настраивается (как nifi.flowfile.repository.rocksdb.sync.period) и может быть настроена для отдельной системы.

Параметры конфигурации для данного репозитория делятся на две категории: NiFi-centric and RocksDB-centric. Настройки NiFi-centric связаны с операциями репозитория FlowFile и его взаимодействием с NiFi. Настройки RocksDB-centric напрямую связаны с настройками базового репозитория RocksDB. Более подробную информацию об этих настройках можно найти в [документации RocksDB](https://github.com/facebook/rocksdb/wiki/RocksJava-Basics).

|  |
| --- |
| Примечание. |
| Пользователи Windows должны убедиться, что Microsoft Visual C ++ 2015 Redistributable установлен, чтобы этот репозиторий работал. См. [документацию](https://github.com/facebook/rocksdb/wiki/RocksJava-Basics#maven-windows) для получения дополнительных сведений. |

Чтобы использовать данную имплементацию, установите для nifi.flowfile.repository.implementation значение org.apache.nifi.controller.repository.RocksDBFlowFileRepository.

Таблица 25 — Свойства конфигурации NiFi-centric

| Свойство | Описание |
| --- | --- |
| nifi.flowfile.repository.directory | Расположение репозитория FlowFile. Значение по умолчанию — ./flowfile\_repository. |
| nifi.flowfile.repository.rocksdb.sync.warning.period | Как часто логировать предупреждения, если синхронизация невозможна. Значение по умолчанию — 30 секунд. |
| nifi.flowfile.repository.rocksdb.claim.cleanup.period | Как часто помечать заявки на контент как разрушаемые (чтобы их можно было удалить из репозитория content). Значение по умолчанию — 30 секунд. |
| nifi.flowfile.repository.rocksdb.deserialization.threads | Сколько потоков использовать при запуске, восстанавливая состояние FlowFile. Значение по умолчанию — 16. |
| nifi.flowfile.repository.rocksdb.deserialization.buffer.size | Размер буфера для использования при запуске восстановления состояния FlowFile. Значение по умолчанию — 1000. |
| nifi.flowfile.repository.rocksdb.sync.period | Частота, с которой выполняется синхронизация с диском. Это максимальный период, в течение которого операция создания данных может блокироваться, если nifi.flowfile.repository.rocksdb.accept.data.loss имеет значение false. Значение по умолчанию — 10 миллисекунд. |
| nifi.flowfile.repository.rocksdb.accept.data.loss | Принимать ли потерю полученных/созданных данных. Установка этого значения в значение true увеличивает пропускную способность, если потеря данных допустима. Значение по умолчанию — false. |
| nifi.flowfile.repository.rocksdb.enable.stall.stop | Разрешить ли остановку записи в репозиторий на основе настроенных ограничений. Включение этой функции позволяет системе защитить себя, ограничивая (откладывая или запрещая) операции, которые увеличивают общее количество файлов FlowFile на ноде, чтобы предотвратить перегрузку системы. Значение по умолчанию — false. |
| nifi.flowfile.repository.rocksdb.stall.period | Период времени для остановки при обнаружении указанных критериев. Значение по умолчанию — 100 миллисекунд. |
| nifi.flowfile.repository.rocksdb.stall.flowfile.count | Счётчик FlowFile, с которого начинается задержка записи в репозиторий. Значение по умолчанию — 800000. |
| nifi.flowfile.repository.rocksdb.stall.heap.usage.percent | Использование heap, при котором начинается остановка записи в репозиторий. Значение по умолчанию — 95%. |
| nifi.flowfile.repository.rocksdb.stop.flowfile.count | Счётчик FlowFile, при котором начинается остановка создания новых FlowFiles. Значение по умолчанию — 1100000. |
| nifi.flowfile.repository.rocksdb.stop.heap.usage.percent | Использование heap, при котором начинается остановка создания новых FlowFiles. Значение по умолчанию — 99.9%. |
| nifi.flowfile.repository.rocksdb.remove.orphaned.flowfiles.on.startup | Разрешить ли репозиторию удалять FlowFiles, которые он не может идентифицировать при запуске. Поскольку это часто является результатом ошибки конфигурации или синхронизации, по умолчанию она отключена. Её следует включать только в том случае, если вы абсолютно уверены, что хотите потерять данные, о которых идет речь. Значение по умолчанию — false. |
| nifi.flowfile.repository.rocksdb.enable.recovery.mode | Включать ли "режим восстановления". Это ограничивает количество FlowFiles, загружаемых в граф за один раз, при этом фактически не удаляя какие-либо FlowFiles (или контент) из системы. Это позволяет восстановить систему, в которой при запуске возникают ошибки OutOfMemory или подобные. Это не должно быть включено, если это не необходимо для восстановления системы, и должно быть отключено, как только это будет выполнено.  Внимание. В режиме восстановления не вносите изменения в граф. Изменения в графе могут привести к невозможности дальнейшего восстановления FlowFiles из репозитория. Значение по умолчанию — false. |
| nifi.flowfile.repository.rocksdb.recovery.mode.flowfile.count | Количество FlowFiles для загрузки в граф в «режиме восстановления». Когда FlowFiles покидает систему, дополнительные FlowFiles будут загружаться до установленного предела. Данный параметр не предотвращает попадание FlowFiles в систему обычными средствами. Значение по умолчанию — 5000. |

Таблица 26 — Свойства конфигурации RocksDB-centric

| Свойство | Описание |
| --- | --- |
| nifi.flowfile.repository.rocksdb.parallel.threads | Количество потоков, используемых для фоновой очистки и уплотнения. Хорошее значение — это количество ядер. См. RockDB DBOptions.setIncreaseParallelism() для получения дополнительной информации. Значение по умолчанию — 8. |
| nifi.flowfile.repository.rocksdb.max.write.buffer.number | Максимальное количество буферов записи, создаваемых в памяти. Дополнительные сведения см. в ColumnFamilyOptions.setMaxWriteBufferNumber()/max\_write\_buffer\_number. Значение по умолчанию — 4. |
| nifi.flowfile.repository.rocksdb.write.buffer.size | Объём данных, которые необходимо накапливать в памяти перед преобразованием в отсортированный на диске файл. Большие значения увеличивают производительность, особенно при массовых загрузках. Одновременно в памяти может храниться до max\_write\_buffer\_number буферов записи, поэтому вы можете настроить этот параметр для управления использованием памяти. См. RockDB ColumnFamilyOptions.setWriteBufferSize()/write\_buffer\_size для получения дополнительной информации. Значение по умолчанию — 256 MB. |
| nifi.flowfile.repository.rocksdb.level.0.slowdown.writes.trigger | Мягкое ограничение на количество файлов нулевого уровня. На этом этапе записи замедляются. Значение меньше 0 означает, что замедление записи не будет вызвано количеством файлов на уровне 0. См. RocksDB ColumnFamilyOptions.setLevel0SlowdownWritesTrigger()/level0\_slowdown\_writes\_trigger для получения дополнительной информации. Значение по умолчанию — 20. |
| nifi.flowfile.repository.rocksdb.level.0.stop.writes.trigger | Максимальное количество файлов нулевого уровня. На этом этапе запись будет остановлена. См. RocksDB ColumnFamilyOptions.setLevel0StopWritesTrigger()/level0\_stop\_writes\_trigger для получения дополнительной информации. Значение по умолчанию — 40. |
| nifi.flowfile.repository.rocksdb.delayed.write.bytes.per.second | Ограниченная скорость записи в БД при замедлении. RocksDB может решить замедлить работу ещё больше, если уплотнение ещё больше отстанет. См. RocksDB DBOptions.setDelayedWriteRate() для получения дополнительной информации. Значение по умолчанию — 16 MB. |
| nifi.flowfile.repository.rocksdb.max.background.flushes | Задаёт максимальное количество одновременных джобов фоновой очистки. См. RocksDB DBOptions.setMaxBackgroundFlush()/max\_background\_flush для получения дополнительной информации. Значение по умолчанию — 1. |
| nifi.flowfile.repository.rocksdb.max.background.compactions | Задаёт максимальное количество одновременных джобов фонового уплотнения. См. RocksDB DBOptions.setMaxBackgroundCompactions()/max\_background\_compactions для получения дополнительной информации. Значение по умолчанию — 1. |
| nifi.flowfile.repository.rocksdb.min.write.buffer.number.to.merge | Минимальное количество буферов записи, которые необходимо объединить перед записью в хранилище. См. RocksDB ColumnFamilyOptions.setMinWriteBufferNumberToMerge()/min\_write\_buffer\_number\_to\_merge для получения дополнительной информации. Значение по умолчанию — 1. |
| nifi.flowfile.repository.rocksdb.stat.dump.period | Период, в который выгрузить дамп rockdb.stats в лог. См. RocksDB DBOptions.setStatsDumpPeriodSec()/stats\_dump\_period\_sec для получения дополнительной информации. Значение по умолчанию — 600 sec. |

Управление swap

NiFi хранит информацию о FlowFile в памяти (JVM), но во время всплесков входящих данных информация о FlowFile может начать занимать такую большую часть JVM, что страдает производительность системы. Чтобы противодействовать этому эффекту, NiFi временно «переставляет» информацию FlowFile на диск, пока снова не станет доступным дополнительное пространство JVM. Эти свойства определяют, как происходит этот процесс.

Таблица 27 — Свойства Swap Management

| Свойство | Описание |
| --- | --- |
| nifi.swap.manager.implementation | Имплементация Swap Manager. Значение по умолчанию — org.apache.nifi.controller.FileSystemSwapManager. Существует альтернативная имплементация EncryptedFileSystemSwapManager, которая шифрует содержимое файла подкачки на диске. Ключ шифрования, настроенный для репозитория FlowFile, используется для шифрования с использованием алгоритма AES-GCM. |
| nifi.queue.swap.threshold | Порог очереди, при котором NiFi начинает перекачивать информацию FlowFile на диск. Значение по умолчанию — 20000. |

Репозиторий Content

Репозиторий Content содержит контент для всех FlowFiles в системе. По умолчанию он устанавливается в тот же корневой каталог установки, что и все другие репозитории; однако администраторы, вероятно, захотят настроить его на отдельном диске, если это возможно. По крайней мере, лучше всего, если репозиторий Content не находится на том же диске, что и репозиторий FlowFile. В dataflows, которые обрабатывают большой объём данных, репозиторий Content может заполнить диск, а репозиторий FlowFile, если он также находится на этом диске, может быть выведен из строя. Чтобы избежать данной ситуации, настройте эти репозитории на разных дисках.

Таблица 28 — Свойства репозитория Content

| Свойство | Описание |
| --- | --- |
| nifi.content.repository.implementation | Имплементация Content Repository. Значение по умолчанию — org.apache.nifi.controller.repository.FileSystemRepository, и его следует изменять только с осторожностью. Чтобы хранить содержимое flowfile в памяти, а не на диске (с риском потери данных в случае сбоя питания/машины), установите для этого свойства значение org.apache.nifi.controller.repository.VolatileContentRepository. |

Файловая система репозитория Content

Таблица 29 — Свойства файловой системы репозитория Content

| Свойство | Описание |
| --- | --- |
| nifi.content.repository.implementation | Имплементация Content Repository. Значение по умолчанию — org.apache.nifi.controller.repository.FileSystemRepository, и его следует изменять только с осторожностью. Чтобы хранить содержимое flowfile в памяти, а не на диске (с риском потери данных в случае сбоя питания/машины), установите для этого свойства значение org.apache.nifi.controller.repository.VolatileContentRepository. |
| nifi.content.claim.max.appendable.size | Максимальный размер заявки на контент. Значение по умолчанию — 1 MB. |
| nifi.content.repository.directory.default\* | Расположение репозитория Content. Значение по умолчанию — ./content\_repository.  Примечание. Можно указать несколько репозиториев Content с помощью префикса nifi.content.repository.directory. с уникальными суффиксами и отдельными путями в качестве значений.  Например, чтобы предоставить два дополнительных местоположения, которые будут действовать как часть репозитория Content, пользователь может также указать дополнительные свойства с ключами:  nifi.content.repository.directory.content1=/repos/content1  nifi.content.repository.directory.content2=/repos/content2  Итого всего три расположения, включая nifi.content.repository.directory.default. |
| nifi.content.repository.archive.max.retention.period | Если архивирование включено (см. nifi.content.repository.archive.enabled ниже), то это свойство указывает максимальное количество времени для хранения заархивированных данных. Значение по умолчанию — 12 hours. |
| nifi.content.repository.archive.max.usage.percentage | Если архивирование включено (см. nifi.content.repository.archive.enabled ниже), то это свойство должно иметь значение, указывающее процент использования диска репозитория Content, при котором заархивированные данные начинают удаляться. Если архив пуст, а использование диска репозитория Content превышает этот процент, то архивирование временно отключается. Архивирование возобновится, когда использование диска станет ниже установленного процента. Значение по умолчанию — 50%. |
| nifi.content.repository.archive.enabled | Чтобы включить архивирование контента, установите для него значение true и укажите значение для свойства nifi.content.repository.archive.max.usage.percentage выше. Архивирование контента позволяет пользовательскому интерфейсу просматривать или воспроизводить контент, который больше не находится в очереди dataflow. По умолчанию архивирование включено. |
| nifi.content.repository.always.sync | Если установлено значение true, любые изменения в репозитории будут синхронизироваться с диском, а это означает, что NiFi попросит операционную систему не кэшировать информацию. Это очень дорого и может значительно снизить производительность NiFi. Однако, если установлено false, существует вероятность потери данных в случае внезапного отключения питания или сбоя операционной системы. Значение по умолчанию — false. |
| nifi.content.viewer.url | URL-адрес веб-просмотрщика контента, если он имеется. По умолчанию он пуст. |

Encrypted файловая система репозитория Content

Все свойства, определённые выше в п. 2.4.7.1, применяются. Здесь перечислены только свойства, относящиеся к шифрованию.

Таблица 30 — Свойства зашифрованной файловой системы репозитория Content

| Свойство | Описание |
| --- | --- |
| nifi.content.repository.encryption.key.provider.implementation | Это полное имя класса поставщика ключей (key provider). Поставщик ключей — это интерфейс хранилища данных для доступа к ключу шифрования для защиты заявок на контент. В настоящее время существует три имплементации: StaticKeyProvider, который считывает ключ непосредственно из nifi.properties, FileBasedKeyProvider, который считывает ключи из зашифрованного файла, и KeyStoreKeyProvider, который считывает ключи из стандартного java.security.KeyStore. |
| nifi.content.repository.encryption.key.provider.location | Путь к ресурсу определения ключа (пустой для StaticKeyProvider, ./keys.nkp или аналогичный путь для FileBasedKeyProvider). Для будущих поставщиков, таких как HSM, это может быть строка подключения или URL-адрес. |
| nifi.content.repository.encryption.key.provider.password | Пароль, используемый для расшифровки ресурса определения ключа, например keystore для KeyStoreKeyProvider. |
| nifi.content.repository.encryption.key.id | Идентификатор активного ключа, который будет использоваться для шифрования (например, Key1). |
| nifi.content.repository.encryption.key | Ключ, который нужно использовать для StaticKeyProvider. Формат ключа закодирован в шестнадцатеричном формате (0123456789ABCDEFFEDCBA98765432100123456789ABCDEFFEDCBA9876543210), но также может быть зашифрован с помощью инструмента ./encrypt-config.sh в NiFi Toolkit. |
| nifi.content.repository.encryption.key.id.\* | Позволяет указать дополнительные ключи для StaticKeyProvider. Например, строка nifi.flowfile.repository.encryption.key.id.Key2=012…​210 предоставит доступный ключ Key2. |

Самая простая конфигурация представлена ниже:

|  |
| --- |
| nifi.content.repository.implementation=org.apache.nifi.controller.repository.crypto.EncryptedFileSystemRepository  nifi.content.repository.encryption.key.provider.implementation=org.apache.nifi.security.kms.StaticKeyProvider  nifi.content.repository.encryption.key.provider.location=  nifi.content.repository.encryption.key.id=Key1  nifi.content.repository.encryption.key=0123456789ABCDEFFEDCBA98765432100123456789ABCDEFFEDCBA9876543210 |

Репозиторий Volatile Content

Таблица 31 — Свойства репозитория Volatile Content

| Свойство | Описание |
| --- | --- |
| nifi.volatile.content.repository.max.size | Максимальный размер репозитория Content в памяти. Значение по умолчанию — 100 MB. |
| nifi.volatile.content.repository.block.size | Размер блока репозитория Content. Значение по умолчанию — 32 KB. |

Репозиторий Provenance

Репозиторий Provenance содержит информацию, относящуюся к Data Provenance. Следующие четыре раздела посвящены свойствам Provenance Repository.

Таблица 32 — Свойства репозитория Provenance

| Свойство | Описание |
| --- | --- |
| nifi.provenance.repository.implementation | Имплементация Provenance Repository. Значение по умолчанию — org.apache.nifi.provenance.WriteAheadProvenanceRepository. Также доступны три дополнительных репозитория. Чтобы хранить события Provenance в памяти, а не на диске (в этом случае все события будут потеряны при перезапуске, а события будут вытеснены в порядке FIFO), установите для этого свойства значение org.apache.nifi.provenance. .VolatileProvenanceRepository. Это оставляет настраиваемое количество событий Provenance в куче Java, поэтому количество событий, которые можно сохранить, очень ограничено.  Доступны третий и четвертый варианты: org.apache.nifi.provenance.PersistentProvenanceRepository и org.apache.nifi.provenance.EncryptedWriteAheadProvenanceRepository. PersistentProvenanceRepository изначально был написан с простой целью — сохранять события Provenance по мере их создания и предоставлять возможность последовательно повторять эти события. Позже возникла потребность в возможности сжатия данных, чтобы можно было хранить больше данных. После этого была добавлена ​​возможность индексировать и запрашивать данные. По мере того, как требования менялись с течением времени, репозиторий продолжал меняться без каких-либо серьезных изменений. При использовании в Инстансе NiFi, который отвечает за обработку больших объёмов небольших FlowFiles, PersistentProvenanceRepository может быстро стать узким местом. Затем был написан WriteAheadProvenanceRepository, обеспечивающий те же возможности, что и PersistentProvenanceRepository, но при этом обеспечивающий гораздо лучшую производительность. WriteAheadProvenanceRepository был добавлен в NiFi версии 1.2.0. С тех пор он оказался очень стабильным и надёжным, и поэтому стал реализацией по умолчанию. PersistentProvenanceRepository теперь считается устаревшим и больше не должен использоваться. При администрировании инстанса NiFi, который в настоящее время использует PersistentProvenanceRepository, настоятельно рекомендуется перейти на WriteAheadProvenanceRepository. Сделать это так же просто, как изменить значение свойства имплементации с org.apache.nifi.provenance.PersistentProvenanceRepository на org.apache.nifi.provenance.WriteAheadProvenanceRepository. Поскольку репозиторий Provenance имеет обратную совместимость, потери данных или функциональности не произойдет.  EncryptedWriteAheadProvenanceRepository основывается на WriteAheadProvenanceRepository и обеспечивает шифрование данных в состоянии покоя.  Примечание. WriteAheadProvenanceRepository будет использовать данные Provenance, хранящиеся в PersistentProvenanceRepository. Однако PersistentProvenanceRepository может быть не в состоянии прочитать данные, записанные WriteAheadProvenanceRepository. Следовательно, как только репозиторий Provenance будет изменён на использование WriteAheadProvenanceRepository, его нельзя будет изменить обратно на PersistentProvenanceRepository без удаления данных в репозитории Provenance. |

Репозиторий Write Ahead Provenance

Таблица 33 — Свойства репозитория Write Ahead Provenance

| Свойство | Описание |
| --- | --- |
| nifi.provenance.repository.directory.default\* | Расположение репозитория Provenance. Значение по умолчанию —./provenance\_repository.  Примечание. Несколько репозиториев Provenance можно указать с помощью префикса nifi.provenance.repository.directory. с уникальными суффиксами и отдельными путями в качестве значений.  Например, чтобы предоставить два дополнительных местоположения, которые будут действовать как часть репозитория Provenance, пользователь может также указать дополнительные свойства с ключами:  nifi.provenance.repository.directory.provenance1=/repos/provenance1  nifi.provenance.repository.directory.provenance2=/repos/provenance2  Итого всего три расположения, включая nifi.provenance.repository.directory.default. |
| nifi.provenance.repository.max.storage.time | Максимальное время хранения информации о Provenance данных. Значение по умолчанию — 24 hours. |
| nifi.provenance.repository.max.storage.size | Максимальный объём информации о Provenance данных для одновременного хранения. Значение по умолчанию — 10 GB. Возможность Data Provenance может потреблять много места для хранения, потому что хранится очень много данных. Для производственных сред значения 1-2 ТБ и более не редкость. Репозиторий будет записывать в один event file (или набор event file, если определено несколько мест хранения, как описано выше), пока файл событий не достигнет размера, определённого в свойстве nifi.provenance.repository.rollover.size. Затем он «перевернётся» и начнет записывать новые события в новый файл. Данные всегда удаляются из одного файла за раз, поэтому не рекомендуется записывать огромный объём данных в один event file, так как это предотвратит старение старых данных так же плавно. |
| nifi.provenance.repository.rollover.size | Объём данных для записи в один event file. Значение по умолчанию — 100 MB. Для производственных сред, в которых создаётся очень большой объём данных Provenance, значение 1 ГБ также очень разумно. |
| nifi.provenance.repository.query.threads | Количество потоков, используемых для запросов к репозиторию Provenance. Значение по умолчанию — 2. |
| nifi.provenance.repository.index.threads | Количество потоков, используемых для индексации событий Provenance, чтобы они были доступны для поиска. Значение по умолчанию — 2. Для потоков, которые работают с очень большим количеством FlowFiles, индексация событий Provenance может стать узким местом. Если это произойдёт, увеличение значения этого свойства может увеличить скорость, с которой репозиторий Provenance сможет обрабатывать эти записи, что приведёт к повышению общей пропускной способности. Рекомендуется использовать по крайней мере 1 поток на каждое место хранения (т.е. если есть 3 места хранения, следует использовать как минимум 3 потока). Для сред с высокой пропускной способностью, где доступно больше операций ввода-вывода ЦП и диска, может иметь смысл значительно увеличить это значение. Обычно превышение 2–4 потоков на одно место хранения не имеет смысла. Однако это можно настроить в зависимости от доступных ресурсов ЦП по сравнению с ресурсами ввода-вывода. |
| nifi.provenance.repository.compress.on.rollover | Указывает, следует ли сжимать информацию о Provenance при пролистывании event file. Значение по умолчанию — true. |
| nifi.provenance.repository.always.sync | Если установлено значение true, любые изменения в репозитории будут синхронизироваться с диском, а это означает, что NiFi попросит операционную систему не кэшировать информацию. Это очень дорого и может значительно снизить производительность NiFi. Однако, если установлено false, существует вероятность потери данных в случае внезапного отключения питания или сбоя операционной системы. Значение по умолчанию — false. |
| nifi.provenance.repository.indexed.fields | Перечень полей, разделённых запятыми, которые следует проиндексировать и сделать доступными для поиска. Поля, которые не проиндексированы, не будут доступны для поиска. Допустимые поля: EventType, FlowFileUUID, Filename, TransitURI, ProcessorID, AlternateIdentifierURI, Relationship, Details. Значение по умолчанию — EventType, FlowFileUUID, Filename, ProcessorID. |
| nifi.provenance.repository.indexed.attributes | Перечень атрибутов FlowFile, разделённых запятыми, которые следует проиндексировать и сделать доступными для поиска. По умолчанию он пуст. Но следует рассмотреть несколько хороших примеров: filename и mime.type, а также любых настраиваемых атрибутов, которые могут быть полезны для вашего варианта использования. |
| nifi.provenance.repository.index.shard.size | Репозиторий использует Lucene для выполнения функций индексации и поиска. Это значение указывает, насколько большим должен стать индекс Lucene, прежде чем репозиторий начнёт запись в новый индекс. Большие значения размера шарда приведут к большему использованию кучи Java при поиске в репозитории Provenance, но должны обеспечить лучшую производительность. Значение по умолчанию — 500 MB. Однако это связано с тем, что настройки по умолчанию настроены для очень маленьких сред, где большинство пользователей начинают использовать NiFi. Для производственных сред рекомендуется изменить это значение на 4–8 ГБ. Как только все события Provenance в индексе будут удалены из event files, индекс также будет уничтожен.  Примечание. Это значение должно быть меньше (не более половины) свойства nifi.provenance.repository.max.storage.size. |
| nifi.provenance.repository.max.attribute.length | Указывает максимальную длину, которой может быть атрибут FlowFile при извлечении события Provenance из репозитория. Если длина любого атрибута превышает это значение, она будет усечена при получении события. Значение по умолчанию — 65536. |
| nifi.provenance.repository.concurrent.merge.threads | Lucene создаёт несколько «сегментов» в индексе. Эти сегменты периодически объединяются, чтобы обеспечить более быстрое выполнение запросов. Это свойство указывает максимальное количество потоков, которые разрешено использовать для каждого из каталогов хранилища. Значение по умолчанию — 2. Для сред с высокой пропускной способностью рекомендуется устанавливать количество потоков индекса больше, чем количество потоков слияния\*количество мест хранения. Например, если имеется 2 места хранения и количество потоков индекса установлено равным 8, то количество потоков слияния, вероятно, должно быть меньше 4. Хотя это не критично, если количество потоков слияния должно быть больше чем это может привести к тому, что все потоки индекса будут использоваться для слияния, что приведёт к тому, что поток NiFi будет периодически останавливаться во время индексирования, в результате чего некоторые данные обрабатываются с гораздо большей задержкой, чем другие данные. |
| nifi.provenance.repository.warm.cache.frequency | Каждый раз, когда выполняется запрос Provenance, он должен сначала выполнять поиск в индексах Lucene (по крайней мере, в большинстве случаев — некоторые запросы выполняются часто, а результаты кэшируются, чтобы избежать поиска в индексах Lucene). Когда индекс Lucene открывается впервые, это может быть очень дорого и занять несколько секунд. Это усугубляется наличием множества разных индексов и может привести к тому, что запрос Provenance займет гораздо больше времени. После того, как индекс был открыт, дисковый кэш операционной системы обычно удерживает достаточно данных, чтобы повторно открыть индекс намного быстрее — по крайней мере, в течение определённого периода времени, пока дисковый кэш не вытеснит эти данные. Если это значение установлено, NiFi будет периодически открывать каждый индекс Lucene, а затем закрывать его, чтобы «подогреть» кэш. Это приведёт к гораздо более быстрым запросам, когда репозиторий Provenance большой. Однако, как и все великие вещи, за это приходится платить. «Подогрев» кэша требует некоторых ресурсов ЦП, но, что более важно, он вытеснит другие данные из дискового кэша операционной системы и приведёт к чтению (потенциально большого количества) данных с диска. Это может привести к снижению производительности NiFi. Однако, если NiFi работает в среде, где ЦП и диск не используются полностью, эта функция может привести к гораздо более быстрым запросам Provenance. Значение по умолчанию для этого свойства пустое (т.е. отключено). |

Репозиторий Encrypted Write Ahead Provenance

Все свойства, определённые выше в п. 2.4.8.1, применяются. Здесь перечислены только свойства, относящиеся к шифрованию.

Таблица 34 — Свойства репозитория Encrypted Write Ahead Provenance

| Свойство | Описание |
| --- | --- |
| nifi.provenance.repository.encryption.key.provider.implementation | Это полное имя класса поставщика ключей (key provider). Поставщик ключей — это интерфейс хранилища данных для доступа к ключу шифрования для защиты заявок на контент. В настоящее время существует три имплементации: StaticKeyProvider, который считывает ключ непосредственно из nifi.properties, FileBasedKeyProvider, который считывает ключи из зашифрованного файла, и KeyStoreKeyProvider, который считывает ключи из стандартного java.security.KeyStore. |
| nifi.provenance.repository.encryption.key.provider.location | Путь к ресурсу определения ключа (пустой для StaticKeyProvider, ./keys.nkp или аналогичный путь для FileBasedKeyProvider). Для будущих поставщиков, таких как HSM, это может быть строка подключения или URL-адрес. |
| nifi.provenance.repository.encryption.key.provider.password | Пароль, используемый для расшифровки ресурса определения ключа, например keystore для KeyStoreKeyProvider. |
| nifi.provenance.repository.encryption.key.id | Идентификатор активного ключа, который будет использоваться для шифрования (например, Key1). |
| nifi.provenance.repository.encryption.key | Ключ, который нужно использовать для StaticKeyProvider. Формат ключа закодирован в шестнадцатеричном формате (0123456789ABCDEFFEDCBA98765432100123456789ABCDEFFEDCBA9876543210), но также может быть зашифрован с помощью инструмента ./encrypt-config.sh в NiFi Toolkit. |
| nifi.provenance.repository.encryption.key.id.\* | Позволяет указать дополнительные ключи для StaticKeyProvider. Например, строка nifi.flowfile.repository.encryption.key.id.Key2=012…​210 предоставит доступный ключ Key2. |

Самая простая конфигурация представлена ниже:

|  |
| --- |
| nifi.provenance.repository.implementation=org.apache.nifi.provenance.EncryptedWriteAheadProvenanceRepository  nifi.provenance.repository.encryption.key.provider.implementation=org.apache.nifi.security.kms.StaticKeyProvider  nifi.provenance.repository.encryption.key.provider.location=  nifi.provenance.repository.encryption.key.id=Key1  nifi.provenance.repository.encryption.key=0123456789ABCDEFFEDCBA98765432100123456789ABCDEFFEDCBA9876543210 |

Репозиторий Persistent Provenance

Таблица 35 — Свойства репозитория Persistent Provenance

| Свойство | Описание |
| --- | --- |
| nifi.provenance.repository.directory.default\* | Расположение репозитория Provenance. Значение по умолчанию — ./provenance\_repository.  Примечание. Несколько репозиториев Provenance можно указать с помощью префикса nifi.provenance.repository.directory. с уникальными суффиксами и отдельными путями в качестве значений.  Например, чтобы предоставить два дополнительных местоположения, которые будут действовать как часть репозитория Provenance, пользователь может также указать дополнительные свойства с ключами:  nifi.provenance.repository.directory.provenance1=/repos/provenance1  nifi.provenance.repository.directory.provenance2=/repos/provenance2  Итого всего три расположения, включая nifi.provenance.repository.directory.default. |
| nifi.provenance.repository.max.storage.time | Максимальное время хранения информации о Provenance данных. Значение по умолчанию — 24 hours. |
| nifi.provenance.repository.max.storage.size | Максимальный объём информации о Provenance данных для одновременного хранения. Значение по умолчанию — 10 GB. |
| nifi.provenance.repository.rollover.time | Время ожидания перед переходом к последней информации о Provenance данных, чтобы она была доступна в пользовательском интерфейсе. Значение по умолчанию — 30 secs. |
| nifi.provenance.repository.rollover.size | Объём информации, которую нужно пролистывать за раз. Значение по умолчанию — 100 MB. |
| nifi.provenance.repository.query.threads | Количество потоков, используемых для запросов к репозиторию Provenance. Значение по умолчанию — 2. |
| nifi.provenance.repository.index.threads | Количество потоков, используемых для индексации событий Provenance, чтобы они были доступны для поиска. Значение по умолчанию — 2. Для flows, которые работают с очень большим количеством FlowFiles, индексация событий Provenance может стать узким местом. В этом случае появится бюллетень, указывающий, что «The rate of the dataflow is exceeding the provenance recording rate. Slowing down flow to accommodate.» (Скорость потока данных превышает скорость записи источника. Замедление потока для адаптации). Если это произойдет, увеличение значения данного свойства может увеличить скорость, с которой репозиторий Provenance сможет обрабатывать эти записи, что приведёт к повышению общей пропускной способности. |
| nifi.provenance.repository.compress.on.rollover | Указывает, следует ли сжимать Provenance-информацию при её переносе. Значение по умолчанию — true. |
| nifi.provenance.repository.always.sync | Если установлено значение true, любые изменения в репозитории будут синхронизироваться с диском, а это означает, что NiFi попросит операционную систему не кэшировать информацию. Это очень дорого и может значительно снизить производительность NiFi. Однако, если установлено false, существует вероятность потери данных при внезапном отключении питания или сбое операционной системы. Значение по умолчанию — false. |
| nifi.provenance.repository.journal.count | Количество файлов журнала, которые следует использовать для сериализации данных Provenance Event. Увеличение данного значения позволит большему количеству тасков одновременно обновлять репозиторий, но позже приведёт к более дорогостоящему слиянию файлов журнала. В идеале это значение должно быть равно количеству потоков, которые, как ожидается, будут одновременно обновлять репозиторий, но 16 обычно хорошо работают в обязательных средах. Значение по умолчанию — 16. |
| nifi.provenance.repository.indexed.fields | Перечень полей, разделённых запятыми, которые следует проиндексировать и сделать доступными для поиска. Поля, которые не проиндексированы, не будут доступны для поиска. Допустимые поля: EventType, FlowFileUUID, Filename, TransitURI, ProcessorID, AlternateIdentifierURI, Relationship, Details. Значение по умолчанию: EventType, FlowFileUUID, Filename, ProcessorID. |
| nifi.provenance.repository.indexed.attributes | Перечень атрибутов FlowFile, разделённых запятыми, которые следует проиндексировать и сделать доступными для поиска. По умолчанию он пуст. Но следует рассмотреть несколько хороших примеров: filename и mime.type, а также любых настраиваемых атрибутов, которые могут быть полезны для вашего варианта использования. |
| nifi.provenance.repository.index.shard.size | Большие значения размера шарда приведут к большему использованию кучи Java при поиске в репозитории Provenance, но должны обеспечить лучшую производительность. Значение по умолчанию — 500 MB. |
| nifi.provenance.repository.max.attribute.length | Указывает максимальную длину, которой может быть атрибут FlowFile при извлечении события Provenance из репозитория. Если длина любого атрибута превышает это значение, она будет усечена при получении события. Значение по умолчанию — 65536. |

Репозиторий Volatile Provenance

Таблица 36 — Свойства репозитория Volatile Provenance

| Свойство | Описание |
| --- | --- |
| nifi.provenance.repository.buffer.size | Размер буфера репозитория Provenance. Значение по умолчанию — 100000 provenance-событий. |

Репозиторий Status History

Репозиторий Status History содержит информацию для инструментов Component Status History и Node Status History в пользовательском интерфейсе. Следующие свойства определяют, как работают эти инструменты.

Таблица 37 — Свойства репозитория Status History

| Свойство | Описание |
| --- | --- |
| nifi.components.status.repository.implementation | Реализация репозитория Status History. Значение по умолчанию — org.apache.nifi.controller.status.history.VolatileComponentStatusRepository, который хранит историю состояний в памяти. org.apache.nifi.controller.status.history.EmbeddedQuestDbStatusHistoryRepository также поддерживается и хранит информацию истории состояния на диске, чтобы она была доступна при перезапусках и могла храниться в течение гораздо более длительных периодов времени. |
| nifi.components.status.snapshot.frequency | Значение указывает, как часто нужно делать снапшот истории состояний компонентов. Значение по умолчанию — 1 min. |

Репозиторий In memory Status History

Если значение свойства nifi.components.status.repository.implementation равно VolatileComponentStatusRepository, данные истории состояний будут сохранены в памяти. Если приложение остановится, вся собранная информация будет потеряна.

Параметры buffer.size и snapshot.frequency работают вместе, чтобы определить объём сохраняемых исторических данных. В качестве примера, чтобы настроить исторические данные за два дня со снапшотом точки данных, который создаётся каждые 5 минут, вы должны настроить snapshot.frequency равным 5 mins, а buffer.size равным 576. Объяснение примера: на каждые 60 минут создается 12 (60/5) окон снапшотов за этот период времени. Чтобы сохранять эти данные в течение 48 часов (12\*48), необходим размер буфера 576.

Таблица 38 — Свойства репозитория In memory Status History

| Свойство | Описание |
| --- | --- |
| nifi.components.status.repository.buffer.size | Задаёт размер буфера для репозитория Status History. Значение по умолчанию — 1440. |

Репозиторий Persistent Status History

Если значение свойства nifi.components.status.repository.implementation равно EmbeddedQuestDbStatusHistoryRepository, данные истории состояния будут постоянно храниться на диске. Данные будут храниться между перезапусками.

Таблица 39 — Свойства репозитория Persistent Status History

| Свойство | Описание |
| --- | --- |
| nifi.status.repository.questdb.persist.node.days | Количество дней, в течение которых будут храниться данные о состоянии ноды (например, свободное место на диске в репозитории, информация о сборке мусора и т.д.). Значение по умолчанию — 14. |
| nifi.status.repository.questdb.persist.component.days | Количество дней, в течение которых будут храниться данные о состоянии компонентов (т.е. статистика для каждого Processor, Connection и т.д.). Значение по умолчанию — 3. |
| nifi.status.repository.questdb.persist.location | Расположение репозитория Persistent Status History. Значение по умолчанию — ./status\_repository. |

Свойства Site to Site

Данные свойства определяют, как этот инстанс NiFi взаимодействует с удалёнными инстансами NiFi, когда Remote Process Groups настроены в dataflow. Remote Process Groups могут выбирать транспортный протокол из RAW и HTTP. Свойства, названные с помощью nifi.remote.input.socket.\*, специфичные для транспортного протокола RAW. Точно так же nifi.remote.input.http.\* — это свойства, специфичные для транспортного протокола HTTP.

Таблица 40 — Свойства Site to Site

| Свойство | Описание |
| --- | --- |
| nifi.remote.input.host | Имя хоста, которое будет выдано клиентам для подключения к этому инстансу NiFi для взаимодействия Site-to-Site. По умолчанию данное значение из InetAddress.getLocalHost().getHostName(). В UNIX-подобных операционных системах это обычно результат команды hostname. |
| nifi.remote.input.secure | Свойство указывает на то, должно ли быть защищено взаимодействие между данным инстансом NiFi и удалёнными инстансами NiFi. По умолчанию установлено значение false. Для безопасной работы между сайтами установите для свойства значение true. Также необходимо настроить многие другие свойства безопасности. |
| nifi.remote.input.socket.port | Порт удалённого входного сокета для взаимодействия Site-to-Site. По умолчанию он пуст, но он должен иметь значение, чтобы использовать сокет RAW в качестве транспортного протокола для Site-to-Site. |
| nifi.remote.input.http.enabled | Указывает, следует ли включать HTTP-соединение Site-to-Site на данном хосте. По умолчанию установлено значение true.  Использует ли клиент Site-to-Site HTTP или HTTPS, определяется nifi.remote.input.secure. Если установлено значение true, запросы отправляются по протоколу HTTPS на nifi.web.https.port. Если установлено значение false, HTTP-запросы отправляются на nifi.web.http.port. |
| nifi.remote.input.http.transaction.ttl | Указывает, как долго транзакция может оставаться активной на сервере. По умолчанию установлено 30 secs.  Если клиент Site-to-Site не перешёл к следующему действию по истечении установленного периода времени, транзакция отклоняется от удалённого инстанса NiFi. Например, когда клиент создаёт транзакцию, но не отправляет и не получает flow-файлы, или когда клиент отправляет или получает flow-файлы, но не подтверждает эту транзакцию. |
| nifi.remote.contents.cache.expiration | Определяет, как долго NiFi должен кэшировать информацию об удалённом инстансе NiFi при обмене данными через Site-to-Site. По умолчанию NiFi кэширует ответы от удалённой системы за 30 secs. Это позволяет NiFi избегать постоянных HTTP-запросов к удалённой системе, что особенно важно, когда данный инстанс NiFi имеет много инстансов Remote Process Groups. |

Свойства маршрутизации Site to Site для обратных прокси

Site-to-Site требует peer-to-peer взаимодействия между клиентом и удалённой нодой NiFi. Например, если удалённый кластер NiFi имеет 3 ноды (nifi0, nifi1 и nifi2), то клиентские запросы должны быть доступны каждой из этих удалённых нод.

Если кластер NiFi планируется для приёма/передачи данных от/к клиентам Site-to-Site через Интернет или файрвол компании, перед нодами кластера NiFi можно развернуть обратный прокси-сервер в качестве шлюза для маршрутизации клиентских запросов к вышестоящим нодам NiFi, чтобы уменьшить количество серверов и портов, которые должны быть открыты.

В такой среде ожидается, что один и тот же кластер NiFi будет доступен для клиентов Site-to-Site в той же сети. Типичным примером может быть отправка FlowFiles себе для распределения нагрузки между нодами кластера NiFi. В этом случае клиентские запросы должны направляться непосредственно на ноду, не проходя через обратный прокси-сервер.

Для поддержки таких развёртываний удалённым кластерам NiFi необходимо динамически предоставлять свои конечные точки Site-to-Site в зависимости от контекстов клиентских запросов. Следующие свойства настраивают, как одноранговые ноды должны быть доступны клиентам. Определение маршрутизации состоит из 4 свойств: when, hostname, port и secure, сгруппированных по protocol и name. Можно настроить несколько определений маршрутизации. protocol представляет собой транспортный протокол Site-to-Site, то есть RAW или HTTP.

Таблица 41 — Свойства маршрутизации Site to Site для обратных прокси

| Свойство | Описание |
| --- | --- |
| nifi.remote.route.{protocol}.{name}.when | Логическое значение, true или false. Определяет, следует ли использовать определение маршрутизации для данного имени. |
| nifi.remote.route.{protocol}.{name}.hostname | Укажите имя хоста, которое будет представлено клиентам Site-to-Site для дальнейшего взаимодействия. |
| nifi.remote.route.{protocol}.{name}.port | Укажите номер порта, который будет представлен клиентам Site-to-Site для дальнейшего взаимодействия. |
| nifi.remote.route.{protocol}.{name}.secure | Логическое значение, true или false. Укажите, должен ли доступ к удалённой ноде осуществляться через безопасный протокол. По умолчанию — false. |

Все вышеперечисленные свойства маршрутизации могут использовать NiFi Expression Language для вычисления описания целевой одноранговой ноды из контекста запроса. Доступные переменные представлены в таблице.

Таблица 42 — Переменные для свойств маршрутизации Site to Site для обратных прокси

| Переменная | Описание |
| --- | --- |
| s2s.{source|target}.hostname | Имя хоста источника, откуда пришёл запрос, и исходной цели. |
| s2s.{source|target}.port | То же, что и выше, но для портов. Исходный порт может быть бесполезен, поскольку это просто TCP-порт на стороне клиента. |
| s2s.{source|target}.secure | То же, что и выше, безопасно или нет. |
| s2s.protocol | Название используемого протокола Site-to-Site, RAW или HTTP. |
| s2s.request | Имя текущего типа запроса, SiteToSiteDetail или Peers. |
| HTTP request headers | На значения заголовка HTTP-запроса можно ссылаться по его имени. |

Последовательность протокола Site to Site

Для правильной настройки данных свойств потребуется некоторое понимание последовательности протокола Site-to-Site.

1. Клиент инициирует протокол Site-to-Site, отправляя HTTP(S)-запрос на указанный удалённый URL-адрес для получения информации Site-to-Site удалённого кластера. В частности, в /nifi-api/site-to-site. Этот запрос называется SiteToSiteDetail.

Удалённая нода NiFi отвечает своими портами ввода и вывода, а также номерами портов TCP для транспортных протоколов RAW и TCP.

Клиент отправляет другой запрос на получение удалённых одноранговых нод, используя номер порта TCP, возвращённый на шега 2. Из этого запроса для транспортного протокола RAW используется взаимодействие raw-сокетов, а HTTP продолжает использовать HTTP(S). Этот запрос называется Peers.

Удалённая нода NiFi отвечает перечнем доступных удалённых нод, содержащим имя хоста, порт, безопасность и рабочую нагрузку, такую как, например, количество файлов FlowFiles в очереди. С этого момента устанавливается дальнейшая связь между клиентом и удалённой нодой NiFi.

Клиент решает, какой ноде передавать данные, на основе информации о рабочей нагрузке.

Клиент отправляет запрос на создание транзакции удалённой ноде NiFi.

Удалённая нода NiFi принимает транзакцию.

Данные отправляются целевой ноде. Несколько пакетов данных могут быть отправлены в пакетном режиме.

Когда данных для отправки больше нет или они достигли предела пакета, транзакция подтверждается на обоих концах путём вычисления хэша CRC32 отправленных данных.

Транзакция совершается на обоих концах.

Свойства Web

Данные свойства относятся к пользовательскому веб-интерфейсу.

Таблица 43 — Свойства Web

| Свойство | Описание |
| --- | --- |
| nifi.web.http.host | Хост HTTP. Значение по умолчанию пустое. |
| nifi.web.http.port | Порт HTTP. Значение по умолчанию пустое. |
| nifi.web.http.port.forwarding | Порт, который пересылает входящие HTTP-запросы на nifi.web.http.host. Это свойство предназначено для использования с «переадресацией портов», когда NiFi должен быть запущен пользователем без полномочий root для большей безопасности, но при этом для прохождения через файрвол необходимо получить доступ через низкий порт. Например, чтобы открыть NiFi по протоколу HTTP на порту 80, но на самом деле прослушивать порт 8080, вам необходимо настроить переадресацию портов на уровне ОС, такую как iptables (Linux/Unix) или pfctl (macOS), которая перенаправляет запросы с 80 на 8080. Затем установите nifi.web.http.port как 8080, а nifi.web.http.port.forwarding как 80. По умолчанию он пуст. |
| nifi.web.http.network.interface\* | Имя сетевого интерфейса, к которому NiFi должен привязаться для HTTP-запросов. По умолчанию он пуст.  Примечание. Несколько сетевых интерфейсов можно указать с помощью префикса nifi.web.http.network.interface. с уникальными суффиксами и отдельными именами сетевых интерфейсов в качестве значений.  Например, чтобы предоставить два дополнительных сетевых интерфейса, пользователь может также указать дополнительные свойства с ключами:  nifi.web.http.network.interface.eth0=eth0  nifi.web.http.network.interface.eth1=eth1  Предоставляется три сетевых интерфейса, включая nifi.web.http.network.interface.default. |
| nifi.web.https.host | Хост HTTPS. Значение по умолчанию — 127.0.0.1. |
| nifi.web.https.port | Порт HTTPS. Значение по умолчанию — 8443. |
| nifi.web.https.port.forwarding | То же, что и nifi.web.http.port.forwarding, но с HTTPS для безопасного взаимодействия. По умолчанию он пуст. |
| nifi.web.https.ciphersuites.include | Наборы шифров, используемые для инициализации SSLContext порта HTTPS Jetty. Если не указано, используются значения по умолчанию SSLContext среды выполнения. |
| nifi.web.https.ciphersuites.exclude | Наборы шифров, которые не могут использоваться клиентом SSL для установления соединения с Jetty. Если не указано, используются значения по умолчанию SSLContext среды выполнения. |
| nifi.web.https.network.interface\* | Имя сетевого интерфейса, к которому NiFi должен привязаться для запросов HTTPS. По умолчанию он пуст.  Примечание. Несколько сетевых интерфейсов можно указать с помощью префикса nifi.web.https.network.interface. с уникальными суффиксами и отдельными именами сетевых интерфейсов в качестве значений.  Например, чтобы предоставить два дополнительных сетевых интерфейса, пользователь может также указать дополнительные свойства с ключами:  nifi.web.https.network.interface.eth0=eth0  nifi.web.https.network.interface.eth1=eth1  Предоставляется три сетевых интерфейса, включая nifi.web.https.network.interface.default. |
| nifi.web.jetty.working.directory | Расположение рабочего каталога Jetty. Значение по умолчанию — ./work/jetty. |
| nifi.web.jetty.threads | Количество потоков Jetty. Значение по умолчанию — 200. |
| nifi.web.max.header.size | Максимальный размер, разреш`нный для заголовков запросов и ответов. Значение по умолчанию — 16 KB. |
| nifi.web.proxy.host | Разделённый запятыми перечень разрешённых значений заголовка HTTP-хоста, которые следует учитывать, когда NiFi работает безопасно и будет получать запросы на другой host[:port], чем он привязан. Например, при работе в контейнере Docker или за прокси-сервером (например, localhost:18443, proxyhost:443). По умолчанию это значение пустое, что означает, что NiFi должен разрешать только запросы, отправленные на host[:port], к которому привязан NiFi. |
| nifi.web.proxy.context.path | Перечень разрешённых значений заголовков HTTP X-ProxyContextPath, X-Forwarded-Context или X-Forwarded-Prefix, разделённых запятыми. По умолчанию это значение пустое, что означает, что все запросы, содержащие путь контекста прокси, отклоняются. Настройка этого свойства разрешит запросы, в которых путь прокси-сервера содержится в данном перечне. |
| nifi.web.max.content.size | Максимальный размер (HTTP Content-Length) для запросов PUT и POST. Для обратной совместимости значение по умолчанию не установлено. Указание значения для этого свойства включает фильтр Content-Length для всех входящих запросов API (кроме взаимодействия Site-to-Site и кластера). Предлагаемое значение — 20 MB. |
| nifi.web.max.requests.per.second | Максимальное количество запросов от соединения в секунду. Запросы, превышающие это значение, сначала задерживаются, а затем ограничиваются. |
| nifi.web.request.ip.whitelist | Перечень IP-адресов, разделённых запятыми. Используется для указания IP-адресов клиентов, которые могут превышать максимальное количество запросов в секунду (nifi.web.max.requests.per.second). Не применяется к тайм-ауту веб-запроса. |
| nifi.web.request.timeout | Тайм-аут запроса для веб-запросов. Запросы, выполняющиеся дольше этого времени, будут принудительно завершены ответом HTTP 503 Service Unavailable. Значение по умолчанию — 60 secs. |

Свойства Security

Данные свойства относятся к различным функциям безопасности в NiFi.

Таблица 44 — Свойства Security

| Свойство | Описание |
| --- | --- |
| nifi.sensitive.props.key | Пароль, используемый для шифрования любых значений конфиденциальных свойств, настроенных в процессорах. По умолчанию значение свойства пустое, но системный администратор должен указать для него значение. Это может быть строка любой длины, хотя рекомендуемая минимальная длина составляет 10 символов. Имейте в виду, что после установки этого пароля и настройки одного или нескольких важных свойств процессора этот пароль не следует менять. |
| nifi.sensitive.props.algorithm | Алгоритм, используемый для шифрования конфиденциальных свойств. Значение по умолчанию — NIFI\_PBKDF2\_AES\_GCM\_256. |
| nifi.sensitive.props.provider | Поставщик конфиденциальной свойства. Значение по умолчанию — BC. |
| nifi.sensitive.props.additional.keys | Разделённый запятыми перечень свойств в nifi.properties, используемых для шифрования в дополнение к чувствительным свойствам по умолчанию (см. «Зашифрованные пароли в файлах конфигурации»). |
| nifi.security.autoreload.enabled | Указывает, должна ли фабрика контекста SSL автоматически перезагружаться при обнаружении обновлений в keystore и truststore. По умолчанию установлено значение false. |
| nifi.security.autoreload.interval | Задаёт интервал, с которым keystore и truststore проверяются на наличие обновлений. Применяется, только если для nifi.security.autoreload.enabled установлено значение true. Значение по умолчанию — 10 secs. |
| nifi.security.keystore\* | Полный путь и имя keystore. По умолчанию он пуст. |
| nifi.security.keystoreType | Тип keystore. По умолчанию он пуст. |
| nifi.security.keystorePasswd | Пароль keystore. По умолчанию он пуст. |
| nifi.security.keyPasswd | Ключевой пароль. По умолчанию он пуст. |
| nifi.security.truststore\* | Полный путь и имя truststore. По умолчанию он пуст. |
| nifi.security.truststoreType | Тип truststore. По умолчанию он пуст. |
| nifi.security.truststorePasswd | Пароль truststore. По умолчанию он пуст. |
| nifi.security.user.authorizer | Указывает, какие из настроенных авторизаторов в файле authorizers.xml использовать. По умолчанию — file-provider. |
| nifi.security.allow.anonymous.authentication | Разрешена ли анонимная аутентификация при работе через HTTPS. Если установлено значение true, клиентские сертификаты не требуются для подключения через TLS. |
| nifi.security.user.login.identity.provider | Указывает, какой тип поставщика удостоверений входа использовать. Значение по умолчанию пустое, может быть установлено на идентификатор от поставщика в файле, указанном в nifi.login.identity.provider.configuration.file. Установка этого свойства приведёт к тому, что NiFi будет поддерживать аутентификацию по имени пользователя и паролю. |
| nifi.security.ocsp.responder.url | Это URL-адрес ответчика Online Certificate Status Protocol (OCSP), если он используется. По умолчанию он пуст. |
| nifi.security.ocsp.responder.certificate | Это расположение сертификата ответчика OCSP, если он используется. По умолчанию он пуст. |

Свойства Identity Mapping

Данные свойства можно использовать для нормализации удостоверений пользователей. Если имплементировано, удостоверения, проверенные разными поставщиками удостоверений (сертификаты, LDAP, Kerberos), внутри NiFi обрабатываются одинаково. В результате можно избежать дублирования пользователей, а конфигурации для конкретных пользователей, такие как авторизация, необходимо настраивать только один раз для каждого пользователя.

Следующие примеры демонстрируют нормализацию DN из сертификатов и принципалов из Kerberos:

|  |
| --- |
| nifi.security.identity.mapping.pattern.dn=^CN=(.\*?), OU=(.\*?), O=(.\*?), L=(.\*?), ST=(.\*?), C=(.\*?)$  nifi.security.identity.mapping.value.dn=$1@$2  nifi.security.identity.mapping.transform.dn=NONE  nifi.security.identity.mapping.pattern.kerb=^(.\*?)/instance@(.\*?)$  nifi.security.identity.mapping.value.kerb=$1@$2  nifi.security.identity.mapping.transform.kerb=NONE |

Последний сегмент каждого свойства — это идентификатор, используемый для связывания шаблона со значением замены. Когда пользователь делает запрос к NiFi, его личность проверяется на соответствие каждому из этих шаблонов в лексикографическом порядке. Для первого совпадающего используется замена, указанная в свойстве nifi.security.identity.mapping.value.xxxx. Таким образом, логин с CN=localhost, OU=Apache NiFi, O=Apache, L=Santa Monica, ST=CA, C=US соответствует шаблону маппинга DN, указанному выше, и применяется значение маппинга DN $1@$2. Пользователь нормализован к localhost@Apache NiFi.

В дополнение к маппингу может применяться трансформация. Поддерживаемые версии: NONE (трансформация не применяется), LOWER (идентификатор в нижнем регистре) и UPPER (идентификатор в верхнем регистре). Если не указано, значение по умолчанию — NONE.

|  |
| --- |
| Примечание. |
| Данные маппинги также применяются к Initial Admin Identity, Cluster Node Identity и всем устаревшим пользователям в файле authorizers.xml, а также к пользователям, импортированным из LDAP. |

Имена групп также могут быть смаплены. В следующем примере будет принято существующее имя группы, но в нижнем регистре. Это может быть полезно при использовании вместе с внешним авторизатором.

|  |
| --- |
| nifi.security.group.mapping.pattern.anygroup=^(.\*)$  nifi.security.group.mapping.value.anygroup=$1  nifi.security.group.mapping.transform.anygroup=LOWER |

|  |
| --- |
| Примечание. |
| Данные маппинги применяются к любым устаревшим группам, указанным в authorizers.xml, а также к группам, импортированным из LDAP. |

Общие свойства кластера

При настройке кластера NiFi данные свойства должны быть настроены одинаково на всех нодах.

Таблица 45 — Общие свойства кластера

| Свойство | Описание |
| --- | --- |
| nifi.cluster.protocol.heartbeat.interval | Интервал, с которым ноды должны отправлять heartbeat-сообщения Cluster Coordinator. Значение по умолчанию — 5 secs. |
| nifi.cluster.protocol.heartbeat.missable.max | Максимальное количество heartbeat-сообщений, которое Cluster Coordinator может пропустить для ноды в кластере, прежде чем Cluster Coordinator обновит статус ноды на Disconnected. Значение по умолчанию — 8. |
| nifi.cluster.protocol.is.secure | Показывает, установлена ли безопасность для кластерного взаимодействия. Значение по умолчанию — false. |

Свойства нод кластера

Настройте данные свойства для нод кластера.

Таблица 46 — Свойства нод кластера

| Свойство | Описание |
| --- | --- |
| nifi.cluster.is.node | Установите значение true, если инстанс является нодой в кластере. Значение по умолчанию — false. |
| nifi.cluster.node.address | Полный адрес ноды. По умолчанию он пуст. |
| nifi.cluster.node.protocol.port | Порт протокола ноды. По умолчанию он пуст. |
| nifi.cluster.node.protocol.threads | Количество потоков, которые следует использовать для взаимодействия с другими нодами в кластере. По умолчанию это свойство равно 10, но для больших кластеров это значение может быть увеличено. |
| nifi.cluster.node.protocol.max.threads | Максимальное количество потоков, которые следует использовать для взаимодействия с другими нодами в кластере. Значение этого свойства по умолчанию — 50. |
| nifi.cluster.node.event.history.size | Когда состояние ноды в кластере изменяется, генерируется событие, которое можно просмотреть на странице Cluster. Это значение указывает, сколько событий хранить в памяти для каждого ноды. Значение по умолчанию — 25. |
| nifi.cluster.node.connection.timeout | При подключении к другой ноде в кластере указывает, как долго данная нода должна ждать, прежде чем считать соединение сбойным. Значение по умолчанию — 5 secs. |
| nifi.cluster.node.read.timeout | При обмене данными с другой нодой в кластере указывает, как долго данная нода должна ждать получения информации от удалённой ноды, прежде чем считать связь с нодой сбойной. Значение по умолчанию — 5 secs. |
| nifi.cluster.node.max.concurrent.requests | Максимальное количество ожидающих веб-запросов, которые могут быть реплицированы на ноды в кластере. Если это количество запросов превышено, встроенный сервер Jetty вернёт ответ 409: Conflict. По умолчанию это свойство равно 100. |
| nifi.cluster.firewall.file | Расположение файла файрвола ноды. Это файл, который можно использовать для перечисления всех нод, которым разрешено подключаться к кластеру. Это обеспечивает дополнительный уровень безопасности. По умолчанию значение пустое, что означает, что файл файрвола не используется. |
| nifi.cluster.flow.election.max.wait.time | Задаёт время ожидания перед выбором flow в качестве «правильного» flow. Если количество проголосовавших нод равно количеству, заданному свойством nifi.cluster.flow.election.max.candidates, кластер не будет ждать так долго. Значение по умолчанию — 5 mins. Обратите внимание, что отсчёт времени начинается, как только подан первый голос. |
| nifi.cluster.flow.election.max.candidates | Указывает количество нод, необходимых в кластере, чтобы вызвать досрочный выбор flow. Это позволяет нодам в кластере избежать длительного ожидания перед началом обработки, если мы достигнем хотя бы этого количества нод в кластере. |
| nifi.cluster.load.balance.port | Задаёт порт для прослушивания входящих подключений для загрузки данных и балансировки нагрузки в кластере. Значение по умолчанию — 6342. |
| nifi.cluster.load.balance.host | Задаёт имя хоста для прослушивания входящих подключений для загрузки данных и балансировки нагрузки в кластере. Если не указано, по умолчанию будет использоваться значение, используемое свойством nifi.cluster.node.address. |
| nifi.cluster.load.balance.connections.per.node | Максимальное количество соединений для создания между этой нодой и каждой другой нодой в кластере. Например, если в кластере 5 нод и это значение установлено на 4, будет установлено до 20 подключений к сокетам для целей балансировки нагрузки (5 x 4 = 20). Значение по умолчанию — 1. |
| nifi.cluster.load.balance.max.thread.count | Максимальное количество потоков, используемых для передачи данных от данной ноды к другим нодам в кластере. В то время как данный поток может записывать только в один сокет за раз, один поток может обслуживать несколько соединений одновременно, потому что данное соединение может быть недоступно для чтения/записи в любой момент времени. Значение по умолчанию — 8, то есть до 8 потоков будут отвечать за передачу данных на другие ноды, независимо от того, сколько нод находится в кластере.  Примечание. Увеличение этого значения позволит использовать дополнительные потоки для взаимодействия с другими нодами в кластере и записи данных в репозитории Content и FlowFile. Однако, если для этого свойства установлено значение, большее, чем количество нод в кластере, умноженное на количество подключений на ноду (nifi.cluster.load.balance.connections.per.node), то дополнительных преимуществ не будет, и ресурсы будут потрачены впустую. |
| nifi.cluster.load.balance.comms.timeout | При обмене данными с другой нодой, если это количество времени проходит без каких-либо достижений при чтении или записи в сокет, будет выброшено исключение TimeoutException. Это приведёт к повторной попытке данных или их отправке на другую ноду в кластере, в зависимости от настроенной стратегии балансировки нагрузки. Значение по умолчанию — 30 secs. |

Свойства ZooKeeper

NiFi зависит от ZooKeeper для определения того, какая нода в кластере должна играть роль основной ноды (Primary Node), а какая — координатора кластера (Cluster Coordinator). Эти свойства необходимо настроить, чтобы NiFi мог присоединиться к кластеру.

Таблица 47 — Свойства ZooKeeper

| Свойство | Описание |
| --- | --- |
| nifi.zookeeper.connect.string | Строка подключения, необходимая для подключения к ZooKeeper. Это перечень пар hostname:port, разделённых запятыми. Например, localhost:2181,localhost:2182,localhost:2183. Он должен содержать перечень всех инстансов ZooKeeper в кворуме ZooKeeper. Это свойство должно быть указано для присоединения к кластеру и не имеет значения по умолчанию. |
| nifi.zookeeper.connect.timeout | Как долго ждать при подключении к ZooKeeper, прежде чем считать соединение сбойным. Значение по умолчанию — 3 secs. |
| nifi.zookeeper.session.timeout | Как долго ждать после потери соединения с ZooKeeper до истечения срока действия сеанса. Значение по умолчанию — 3 secs. |
| nifi.zookeeper.root.node | Корневой ZNode, который следует использовать в ZooKeeper. ZooKeeper предоставляет структуру, похожую на каталог, для хранения данных. Каждый «каталог» в этой структуре называется ZNode. Значение обозначает корневой ZNode или «каталог», который следует использовать для хранения данных. Значение по умолчанию — /root. Значение важно установить правильно, поскольку к какому кластеру пытается присоединиться инстанс NiFi, зависит от того, к какому инстансу ZooKeeper он подключается, и от указанной корневой ноды ZooKeeper. |
| nifi.zookeeper.client.secure | Следует ли получать доступ к ZooKeeper с помощью клиентского TLS. |
| nifi.zookeeper.security.keystore | Имя файла keystore, содержащего закрытый ключ для использования при взаимодействии с ZooKeeper. |
| nifi.zookeeper.security.keystoreType | Опционально. Тип keystore. Должен быть PKCS12, JKS или PEM. Если не указан, тип будет определяться по расширению файла (.p12, .jks, .pem). |
| nifi.zookeeper.security.keystorePasswd | Пароль для keystore. |
| nifi.zookeeper.security.truststore | Имя файла truststore, содержащего закрытый ключ для использования при взаимодействии с ZooKeeper. |
| nifi.zookeeper.security.truststoreType | Опционально. Тип truststore. Должен быть PKCS12, JKS или PEM. Если не указан, тип будет определяться по расширению файла (.p12, .jks, .pem). |
| nifi.zookeeper.security.truststorePasswd | Пароль для truststore. |

Свойства Kerberos

Таблица 48 — Свойства Kerberos

| Свойство | Описание |
| --- | --- |
| nifi.kerberos.krb5.file\* | Расположение файла krb5, если он используется. По умолчанию он пуст. В настоящее время для каждого инстанса NiFi разрешено указывать только один файл krb5, поэтому это свойство настроено здесь для поддержки SPNEGO и служб принципалов, а не в отдельных процессорах. При необходимости файл krb5 может поддерживать несколько областей. Пример: /etc/krb5.conf. |
| nifi.kerberos.service.principal\* | Имя принципала службы NiFi Kerberos, если используется. По умолчанию он пуст. Обратите внимание, что это свойство предназначено для NiFi для аутентификации в качестве клиента других систем. Пример: nifi/nifi.example.com или nifi/nifi.example.com@EXAMPLE.COM. |
| nifi.kerberos.service.keytab.location\* | Путь к файлу вкладки NiFi Kerberos, если она используется. По умолчанию он пуст. Обратите внимание, что это свойство предназначено для NiFi для аутентификации в качестве клиента других систем. Пример: /etc/nifi.keytab. |
| nifi.kerberos.spnego.principal\* | Имя принципала службы NiFi Kerberos, если используется. По умолчанию он пуст. Обратите внимание, что это свойство используется для аутентификации пользователей NiFi. Пример: HTTP/nifi.example.com или HTTP/nifi.example.com@EXAMPLE.COM. |
| nifi.kerberos.spnego.keytab.location\* | Путь к файлу вкладки NiFi Kerberos, если она используется. По умолчанию он пуст. Обратите внимание, что это свойство используется для аутентификации пользователей NiFi. Пример: /etc/http-nifi.keytab. |
| nifi.kerberos.spengo.authentication.expiration\* | Срок действия успешной аутентификации пользователя Kerberos, если она используется. Значение по умолчанию — 12 hours. |

Свойства Analytics

Данные свойства определяют поведение возможностей внутренней прогнозной аналитики NiFi, и должны быть настроены одинаково на всех нодах.

Таблица 49 — Свойства Analytics

| Свойство | Описание |
| --- | --- |
| nifi.analytics.predict.enabled | Это указывает, следует ли включать прогнозирование для кластера. По умолчанию — false. |
| nifi.analytics.predict.interval | Временной интервал, для которого должны быть сделаны аналитические прогнозы (например, насыщение очереди). Значение по умолчанию — 3 mins. |
| nifi.analytics.query.interval | Временной интервал для запроса прошлых наблюдений (например, последние 3 минуты снапшотов). Значение по умолчанию — 5 mins. Примечание. Это значение должно быть как минимум в 3 раза больше, чем nifi.components.status.snapshot.frequency, чтобы обеспечить получение достаточного количества наблюдений для прогнозов. |
| nifi.analytics.connection.model.implementation | Класс имплементации для модели аналитики состояния, используемой для прогнозирования подключений. Значение по умолчанию — org.apache.nifi.controller.status.analytics.models.OrdinaryLeastSquares. |
| nifi.analytics.connection.model.score.name | Имя типа скоринга, который следует использовать для оценки модели. Значение по умолчанию — rSquared. |
| nifi.analytics.connection.model.score.threshold | Пороговое значение для оценки (где оценка модели должна быть выше заданного порога). Значение по умолчанию — .90. |

Свойства Runtime Monitoring

Long-Running Task Monitor периодически проверяет потоки исполнителей процессора NiFi и создаёт логи предупреждений и сообщения бюллетеней для тех, которые работали в течение более длительного периода времени. Его можно использовать для обнаружения возможных зависших тасков процессора. Обратите внимание на влияние монитора тасков на производительность: он создаёт дамп потока для каждого запуска, что может повлиять на нормальное выполнение flow. Long-Running Task Monitor можно отключить, не указав значений для его свойств, и по умолчанию он будет отключён. Чтобы включить его, в свойствах nifi.monitor.long.running.task.schedule и nifi.monitor.long.running.task.threshold должны быть указаны допустимые периоды времени.

Таблица 50 — Свойства Runtime Monitoring

| Свойство | Описание |
| --- | --- |
| nifi.monitor.long.running.task.schedule | Период времени между последовательными запусками Long-Running Task Monitor (например, 1 min). |
| nifi.monitor.long.running.task.threshold | Период времени, по истечении которого таск считается длительной, т.е. зависшим (например, 5 mins). |

Свойства Custom

Чтобы настроить кастомные свойства для использования Expression Language NiFi:

1. Создайте кастомное свойство. Убедитесь, что:

Каждое кастомное свойство содержит отдельное значение свойства, поэтому оно не переопределяется существующими свойствами среды, свойствами системы или атрибутами FlowFile.

Каждая нода в кластерной среде настроена с одинаковыми кастомными свойствами.

Обновите nifi.variable.registry.properties, указав расположение файлов кастомных свойств:

Таблица 51 — Свойства Custom

| Свойство | Описание |
| --- | --- |
| nifi.variable.registry.properties | Перечень разделённых запятыми путей расположения файлов для одного или нескольких файлов кастомных свойств. |

Перезапустите свои инстансы NiFi, чтобы обновления были подхвачены.

Расположение процессоров

Доступные параметры конфигурации

NiFi предоставляет 3 варианта конфигурации для расположения процессоров. А именно:

|  |
| --- |
| nifi.nar.library.directory  nifi.nar.library.directory.<custom>  nifi.nar.library.autoload.directory |

|  |
| --- |
| Примечание. |
| Пути, заданные с помощью этих параметров, относятся к домашнему каталогу NiFi. Например, если домашний каталог NiFi — /var/lib/nifi, а каталог библиотек — ./lib, то окончательный путь — /var/lib/nifi/lib. |

Каталог nifi.nar.library.directory используется по умолчанию для поддерживаемых процессоров NiFi. Не рекомендуется использовать его для кастомных процессоров, так как они могут быть потеряны во время обновления NiFi. Например:

|  |
| --- |
| nifi.nar.library.directory=./lib |

nifi.nar.library.directory.<custom> позволяет администратору предоставить NiFi несколько произвольных путей для поиска кастомных процессоров. Уникальный идентификатор свойства должен добавлять свойство для каждого уникального пути. Например:

|  |
| --- |
| nifi.nar.library.directory.myCustomLibs=./my-custom-nars/lib  nifi.nar.library.directory.otherCustomLibs=./other-custom-nars/lib |

Каталог nifi.nar.library.autoload.directory используется функцией автозагрузки, где NiFi может автоматически загружать новые процессоры, добавленные в настроенный путь, без необходимости перезагрузки. Например:

|  |
| --- |
| nifi.nar.library.autoload.directory=./autoload/lib |

Установка кастомных процессоров

В этом разделе описывается исходный процесс установки кастомных процессоров, который требует перезапуска NiFi. Чтобы использовать функцию автозагрузки, см. п. 2.5.3.

Во-первых, мы настроим каталог для кастомных процессоров. Дополнительные сведения об этих параметрах конфигурации см. в п. 2.5.1.

|  |
| --- |
| nifi.nar.library.directory.myCustomLibs=./my-custom-nars/lib |

Убедитесь, что этот каталог существует и имеет соответствующие разрешения для пользователя и группы nifi.

Теперь мы должны поместить наш кастомный процессор nar в настроенный каталог. Настроенный каталог относится к домашнему каталогу NiFi. Например, скажем, что наш домашний каталог NiFi — /var/lib/nifi, мы поместим наш пользовательский процессор nar в /var/lib/nifi/my-custom-nars/lib.

Убедитесь, что файл имеет соответствующие разрешения для пользователя и группы nifi.

Перезапустите NiFi, и теперь кастомный процессор должен быть доступен при добавлении нового процессора в ваш flow.

Автозагрузка кастомных процессоров

В этом разделе описывается процесс использования функции автозагрузки для кастомных процессоров.

Чтобы использовать функцию автозагрузки, свойство nifi.nar.library.autoload.directory необходимо настроить так, чтобы оно указывало на желаемый каталог. По умолчанию оно указывает на ./extensions.

Например:

|  |
| --- |
| nifi.nar.library.autoload.directory=./extensions |

Убедитесь, что этот каталог существует и имеет соответствующие разрешения для пользователя и группы nifi.

Теперь мы должны поместить наш кастомный процессор nar в настроенный каталог. Настроенный каталог относится к домашнему каталогу NiFi. Например, скажем, что наш домашний каталог NiFi — /var/lib/nifi, мы поместим наш кастомный процессор nar в /var/lib/nifi/extensions.

Убедитесь, что файл имеет соответствующие разрешения для пользователя и группы nifi.

Обновите страницу браузера, и теперь кастомный процессор должен быть доступен при добавлении нового процессора в ваш flow.

Провайдеры NAR

NiFi поддерживает получение файлов NAR для функции автозагрузки из внешних источников. Это может быть достигнуто с помощью провайдеров NAR. Провайдер NAR служит связующим звеном между внешним хранилищем данных и NiFi.

При настройке провайдер NAR опрашивает внешний источник доступных файлов NAR и предлагает их фреймворку. Затем фреймворк извлекает новые файлы NAR и копирует их в каталог nifi.nar.library.autoload.directory для автозагрузки.

Провайдер NAR можно настроить, добавив свойство nifi.nar.library.provider. <providerName> .implementation со значением, содержащим правильный класс имплементации. Некоторым имплементациям могут потребоваться дополнительные свойства. Они определяются имплементацией и должны начинаться с префикса nifi.nar.library.provider. <providerName>..

<providerName> является произвольным и служит для сопоставления нескольких свойств для одного провайдера. Может быть установлено несколько провайдеров с разными <providerName>. В настоящее время NiFi поддерживает провайдера NAR на основе HDFS.

Провайдер HDFS NAR

Эта имплементация способна загружать файлы NAR из файловой системы HDFS.

Значение имплементации nifi.nar.library.provider.<providerName>. должно быть org.apache.nifi.nar.hadoop.HDFSNarProvider. Провайдером определяются дополнительные свойства, представленные в таблице.

Таблица 52 — Свойства провайдера HDFS NAR

| Свойство | Описание |
| --- | --- |
| resources | Перечень ресурсов HDFS через запятую. |
| source.directory | Исходный каталог файлов NAR в HDFS. Примечание: провайдер не проверяет файлы рекурсивно. |
| storage.location | Опционально. Если установлено, то место хранения, определённое в core-site.xml, будет перезаписано этим значением. |
| kerberos.principal | Опционально. Принципал Kerberos для аутентификации. |
| kerberos.keytab | Опционально. Вкладка Kerberos, связанная с принципалом. |
| kerberos.password | Опционально. Пароль Kerberos, связанный с принципалом. |

Пример конфигурации:

|  |
| --- |
| nifi.nar.library.provider.hdfs1.implementation=org.apache.nifi.nar.hadoop.HDFSNarProvider  nifi.nar.library.provider.hdfs1.resources=/etc/hadoop/core-site.xml  nifi.nar.library.provider.hdfs1.source.directory=/customNars |

|  |
| --- |
| nifi.nar.library.provider.hdfs2.implementation=org.apache.nifi.nar.hadoop.HDFSNarProvider  nifi.nar.library.provider.hdfs2.resources=/etc/hadoop/core-site.xml  nifi.nar.library.provider.hdfs2.source.directory=/other/dir/for/customNars |

АДМИНИСТРИРОВАНИЕ AIRFLOW

Справочник по конфигурации

Этот раздел содержит список всех доступных конфигураций Airflow, которые вы можете установить в файле airflow.cfg или с помощью переменных среды.

Раздел [core]

Таблица — Раздел [core]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| dags\_folder | string | {AIRFLOW\_HOME}/dags | AIRFLOW\_\_CORE\_\_DAGS\_FOLDER | Папка, в которой находятся ваши pipeline Airflow, скорее всего, подпапка в репозитории кода. Этот путь должен быть абсолютным. |
| hostname\_callable | string | socket.getfqdn | AIRFLOW\_\_CORE\_\_HOSTNAME\_CALLABLE | Имя хоста, указав путь к вызываемому объекту, который разрешит имя хоста. Формат — package.function.  Например, значение по умолчанию socket.getfqdn означает, что результат getfqdn() пакета socket будет использоваться в качестве имени хоста.  В указанной функции аргументы не требуются. Если в качестве имени хоста используется IP-адрес, используйте значение airflow.utils.net.get\_host\_ip\_address. |
| default\_timezone | string | utc | AIRFLOW\_\_CORE\_\_DEFAULT\_TIMEZONE | Часовой пояс по умолчанию в случае, если указанная дата является naive, может быть utc (по умолчанию), system или любой строкой часового пояса IANA (например, Europe/Amsterdam). |
| executor | string | SequentialExecutor | AIRFLOW\_\_CORE\_\_EXECUTOR | Класс исполнителя, который должен использовать Airflow. Возможные варианты: SequentialExecutor, LocalExecutor, CeleryExecutor, DaskExecutor, KubernetesExecutor, CeleryKubernetesExecutor или полный путь импорта к классу при использовании кастомного исполнителя. |
| sql\_alchemy\_conn | string | sqlite:///{AIRFLOW\_HOME}/airflow.db | AIRFLOW\_\_CORE\_\_SQL\_ALCHEMY\_CONN  AIRFLOW\_\_CORE\_\_SQL\_ALCHEMY\_CONN\_CMD  AIRFLOW\_\_CORE\_\_SQL\_ALCHEMY\_CONN\_SECRET | Строка подключения SqlAlchemy к базе данных метаданных. SqlAlchemy поддерживает множество различных движков баз данных. |
| sql\_engine\_encoding | string | utf-8 | AIRFLOW\_\_CORE\_\_SQL\_ENGINE\_ENCODING | Кодировка для баз данных. |
| sql\_engine\_collation\_for\_ids | string | None | AIRFLOW\_\_CORE\_\_SQL\_ENGINE\_COLLATION\_FOR\_IDS | Сопоставление для столбцов dag\_id, task\_id, key, если они имеют разную кодировку. Это особенно полезно в случае mysql с кодировкой utf8mb4, потому что первичные ключи для таблицы XCom имеют слишком большой размер, а sql\_engine\_collation\_for\_ids следует установить в utf8mb3\_general\_ci. |
| sql\_alchemy\_pool\_enabled | string | True | AIRFLOW\_\_CORE\_\_SQL\_ALCHEMY\_POOL\_ENABLED | Если SqlAlchemy должен объединить соединения с базой данных. |
| sql\_alchemy\_pool\_size | string | 5 | AIRFLOW\_\_CORE\_\_SQL\_ALCHEMY\_POOL\_SIZE | Размер пула SqlAlchemy — это максимальное количество подключений к базе данных в пуле. 0 означает отсутствие ограничения. |
| sql\_alchemy\_max\_overflow | string | 10 | AIRFLOW\_\_CORE\_\_SQL\_ALCHEMY\_MAX\_OVERFLOW | Максимальный размер overflow пула. Когда количество разрегистрированных соединений достигает размера, установленного в pool\_size, дополнительные соединения будут возвращены сверх этого предела. Когда эти дополнительные соединения возвращаются в пул, они отключаются и сбрасываются. Из этого следует, что общее количество одновременных подключений, которые разрешит пул, равно pool\_size + max\_overflow, а общее количество «спящих» подключений, которые разрешит пул, равно pool\_size. max\_overflow может быть установлен в -1, чтобы указать отсутствие ограничения overflow; на общее количество одновременных подключений не накладывается никаких ограничений. По умолчанию — 10. |
| sql\_alchemy\_pool\_recycle | string | 1800 | AIRFLOW\_\_CORE\_\_SQL\_ALCHEMY\_POOL\_RECYCLE | Повторный цикл пула SqlAlchemy — это количество секунд, в течение которых соединение может простаивать в пуле, прежде чем оно станет недействительным. Эта конфигурация не относится к sqlite. Если количество подключений к БД когда-либо будет превышено, более низкое значение конфигурации позволит системе быстрее восстанавливаться. |
| sql\_alchemy\_pool\_pre\_ping | string | True | AIRFLOW\_\_CORE\_\_SQL\_ALCHEMY\_POOL\_PRE\_PING | Проверка соединения в начале каждой проверки пула соединений. Обычно это простая инструкция вроде SELECT 1. |
| sql\_alchemy\_schema | string | '' | AIRFLOW\_\_CORE\_\_SQL\_ALCHEMY\_SCHEMA | Схема, используемая для базы данных метаданных. SqlAlchemy поддерживает базы данных с концепцией нескольких схем. |
| sql\_alchemy\_connect\_args | string | None | AIRFLOW\_\_CORE\_\_SQL\_ALCHEMY\_CONNECT\_ARGS | Путь импорта для аргументов подключения в SqlAlchemy. По умолчанию пустой dict. Это полезно, когда вы хотите настроить аргументы механизма базы данных, которые SqlAlchemy не будет анализировать в строке подключения. |
| parallelism | string | 32 | AIRFLOW\_\_CORE\_\_PARALLELISM | Это определяет максимальное количество инстансов тасков, которые могут выполняться одновременно в Airflow, независимо от количества планировщиков и обработчиков. Как правило, это значение отражает количество инстансов тасков с запущенным состоянием в базе данных метаданных. |
| dag\_concurrency | string | 16 | AIRFLOW\_\_CORE\_\_DAG\_CONCURRENCY | Максимальное количество инстансов тасков, которые могут выполняться одновременно в каждом DAG. Чтобы рассчитать количество тасков, которые выполняются одновременно для DAG, сложите количество выполняемых тасков для всех запусков DAG. Это настраивается на уровне DAG с concurrency, который по умолчанию имеет значение dag\_concurrency. |
| dags\_are\_paused\_at\_creation | string | True | AIRFLOW\_\_CORE\_\_DAGS\_ARE\_PAUSED\_AT\_CREATION | Приостановка DAG при создании по умолчанию. |
| max\_active\_runs\_per\_dag | string | 16 | AIRFLOW\_\_CORE\_\_MAX\_ACTIVE\_RUNS\_PER\_DAG | Максимальное количество активных запусков DAG отдельного DAG. Планировщик не будет создавать дополнительных запусков DAG, если достигнет предела. Это настраивается на уровне DAG с помощью max\_active\_runs, которое по умолчанию равно max\_active\_runs\_per\_dag. |
| load\_examples | string | True | AIRFLOW\_\_CORE\_\_LOAD\_EXAMPLES | Следует ли загружать примеры DAG, поставляемые с Airflow. Хорошо начать, но вы, вероятно, захотите установить для него значение False в производственной среде. |
| load\_default\_connections | string | True | AIRFLOW\_\_CORE\_\_LOAD\_DEFAULT\_CONNECTIONS | Следует ли загружать соединения по умолчанию, поставляемые с Airflow. Хорошо начать, но вы, вероятно, захотите установить для него значение False в производственной среде. |
| plugins\_folder | string | {AIRFLOW\_HOME}/plugins | AIRFLOW\_\_CORE\_\_PLUGINS\_FOLDER | Путь к папке, содержащей плагины Airflow. |
| execute\_tasks\_new\_python\_interpreter | boolean | False | AIRFLOW\_\_CORE\_\_EXECUTE\_TASKS\_NEW\_PYTHON\_INTERPRETER | Должны ли задачи выполняться через разветвление родительского процесса (False, более быстрый вариант) или путём создания нового процесса python (True медленный, но означает, что изменения плагинов сразу же улавливаются тасками). |
| fernet\_key | string | {FERNET\_KEY} | AIRFLOW\_\_CORE\_\_FERNET\_KEY  AIRFLOW\_\_CORE\_\_FERNET\_KEY\_CMD  AIRFLOW\_\_CORE\_\_FERNET\_KEY\_SECRET | Секретный ключ для сохранения паролей подключения в БД. |
| donot\_pickle | string | True | AIRFLOW\_\_CORE\_\_DONOT\_PICKLE | Следует ли отключать picking DAG. |
| dagbag\_import\_timeout | float | 30.0 | AIRFLOW\_\_CORE\_\_DAGBAG\_IMPORT\_TIMEOUT | Как долго до истечения времени ожидания импорта файла Python. |
| dagbag\_import\_error\_tracebacks | boolean | True | AIRFLOW\_\_CORE\_\_DAGBAG\_IMPORT\_ERROR\_TRACEBACKS | Должен отображать трассировку в пользовательском интерфейсе для ошибок импорта dagbag, а не только сообщение об исключении. |
| dagbag\_import\_error\_traceback\_depth | integer | 2 | AIRFLOW\_\_CORE\_\_DAGBAG\_IMPORT\_ERROR\_TRACEBACK\_DEPTH | Если отображаются трассировки, сколько записей из трассировки должно быть показано. |
| dag\_file\_processor\_timeout | string | 50 | AIRFLOW\_\_CORE\_\_DAG\_FILE\_PROCESSOR\_TIMEOUT | Как долго до истечения тайм-аута DagFileProcessor, который обрабатывает файл DAG. |
| task\_runner | string | StandardTaskRunner | AIRFLOW\_\_CORE\_\_TASK\_RUNNER | Класс, используемый для запуска инстансов тасков в подпроцессе. Возможные варианты: StandardTaskRunner, CgroupTaskRunner или полный путь импорта к классу при использовании кастомного средства выполнения тасков. |
| default\_impersonation | string | '' | AIRFLOW\_\_CORE\_\_DEFAULT\_IMPERSONATION | Если установлено, таски без аргумента run\_as\_user будут запускаться с этим пользователем.Может использоваться для понижения уровня пользователя sudo, запускающего Airflow, при выполнении тасков. |
| security | string | '' | AIRFLOW\_\_CORE\_\_SECURITY | Какой модуль безопасности использовать (например, kerberos). |
| unit\_test\_mode | string | False | AIRFLOW\_\_CORE\_\_UNIT\_TEST\_MODE | Включение режима unit-тестирования (многие параметры конфигурации перезаписываются тестовыми значениями во время выполнения). |
| enable\_xcom\_pickling | string | False | AIRFLOW\_\_CORE\_\_ENABLE\_XCOM\_PICKLING | Включать ли pickling для xcom (обратите внимание, что это небезопасно и допускает эксплойты RCE). |
| killed\_task\_cleanup\_time | string | 60 | AIRFLOW\_\_CORE\_\_KILLED\_TASK\_CLEANUP\_TIME | Когда таск завершается принудительно, это количество времени в секундах, за которое он должен очистить после отправки SIGTERM, прежде чем он будет SIGKILLED. |
| dag\_run\_conf\_overrides\_params | string | True | AIRFLOW\_\_CORE\_\_DAG\_RUN\_CONF\_OVERRIDES\_PARAMS | Следует ли переопределить параметры с помощью dag\_run.conf. Если вы передадите несколько пар ключ-значение через airflow dags backfill -c или airflow dags trigger -c, пары ключ-значение переопределят существующие в params. |
| dag\_discovery\_safe\_mode | string | True | AIRFLOW\_\_CORE\_\_DAG\_DISCOVERY\_SAFE\_MODE | При обнаружении DAG игнорировать любые файлы, которые не содержат строк DAG и airflow. |
| default\_task\_retries | string | 0 | AIRFLOW\_\_CORE\_\_DEFAULT\_TASK\_RETRIES | Количество повторных попыток для каждого таска по умолчанию. Может быть переопределено на уровне DAG или таска. |
| min\_serialized\_dag\_update\_interval | string | 30 | AIRFLOW\_\_CORE\_\_MIN\_SERIALIZED\_DAG\_UPDATE\_INTERVAL | Обновление сериализованного DAG не может быть быстрее минимального интервала для снижения скорости записи в базу данных. |
| min\_serialized\_dag\_fetch\_interval | string | 10 | AIRFLOW\_\_CORE\_\_MIN\_SERIALIZED\_DAG\_FETCH\_INTERVAL | Получение сериализованной группы DAG не может быть быстрее минимального интервала, чтобы снизить скорость чтения базы данных. Эта конфигурация контролирует, когда ваши DAG обновляются на веб-сервере. |
| store\_dag\_code | string | None | AIRFLOW\_\_CORE\_\_STORE\_DAG\_CODE | Сохранять ли код файлов DAG в БД. Если установлено значение True, веб-сервер считывает содержимое файла из базы данных вместо попытки доступа к файлам в папке DAG. (По умолчанию True). |
| max\_num\_rendered\_ti\_fields\_per\_task | integer | 30 | AIRFLOW\_\_CORE\_\_MAX\_NUM\_RENDERED\_TI\_FIELDS\_PER\_TASK | Максимальное количество отображаемых полей инстанса таска (полей шаблона) для каждого таска для хранения в базе данных. Все поля template\_fields для каждого инстанса таска хранятся в базе данных. Поддерживая это число небольшим, может возникнуть ошибка при попытке просмотреть вкладку Rendered в представлении TaskInstance для более старых тасков. |
| check\_slas | string | True | AIRFLOW\_\_CORE\_\_CHECK\_SLAS | При каждой dagrun проверять на соответствие определённым SLA. |
| xcom\_backend | string | airflow.models.xcom.BaseXCom | AIRFLOW\_\_CORE\_\_XCOM\_BACKEND | Путь к настраиваемому классу XCom, который будет использоваться для хранения и разрешения результатов операторов. Пример — path.to.CustomXCom. |
| lazy\_load\_plugins | boolean | True | AIRFLOW\_\_CORE\_\_LAZY\_LOAD\_PLUGINS | По умолчанию плагины Airflow загружаются лениво (загружаются только при необходимости). Установите значение False, если вы хотите загружать плагины всякий раз, когда airflow вызывается через cli или загружается из модуля. |
| lazy\_discover\_providers | boolean | True | AIRFLOW\_\_CORE\_\_LAZY\_DISCOVER\_PROVIDERS | По умолчанию провайдеры Airflow обнаруживаются лениво (обнаружение и импорт происходят только при необходимости). Установите значение False, если вы хотите обнаруживать поставщиков всякий раз, когда airflow вызывается через cli или загружается из модуля. |
| max\_db\_retries | integer | 3 | AIRFLOW\_\_CORE\_\_MAX\_DB\_RETRIES | Количество повторных попыток выполнения кода в случае ошибок работы БД. Не все транзакции будут повторяться, так как это может вызвать нежелательное состояние. В настоящее время он используется только в DagFileProcessor.process\_file для повторной попытки dagbag.sync\_to\_db. |
| hide\_sensitive\_var\_conn\_fields | boolean | True | AIRFLOW\_\_CORE\_\_HIDE\_SENSITIVE\_VAR\_CONN\_FIELDS | Скрыть чувствительные переменные или дополнительные json-ключи подключения из пользовательского интерфейса и логов тасков, если установлено значение True  (Пароли подключения всегда скрыты в логах) |
| sensitive\_var\_conn\_names | string | '' | AIRFLOW\_\_CORE\_\_SENSITIVE\_VAR\_CONN\_NAMES | Разделённый запятыми перчень особо чувтсвительных ключевых слов, которые следует искать в именах переменных или в дополнительном JSON соединения. |

Раздел [logging]

Таблица — Раздел [logging]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| base\_log\_folder | string | {AIRFLOW\_HOME}/logs | AIRFLOW\_\_LOGGING\_\_BASE\_LOG\_FOLDER | Папка, в которой Airflow должен хранить файлы лога. Этот путь должен быть абсолютным. |
| remote\_logging | string | False | AIRFLOW\_\_LOGGING\_\_REMOTE\_LOGGING | Airflow может хранить логи удалённо в AWS S3, Google Cloud Storage или Elastic Search. Установите значение True, если вы хотите включить удалённое ведение лога. |
| remote\_log\_conn\_id | string | '' | AIRFLOW\_\_LOGGING\_\_REMOTE\_LOG\_CONN\_ID | Пользователи должны предоставить идентификатор подключения Airflow, который обеспечивает доступ к хранилищу. |
| google\_key\_path | string | '' | AIRFLOW\_\_LOGGING\_\_GOOGLE\_KEY\_PATH | Путь к файлу JSON учётных данных Google. Если этот параметр не указан, будет использоваться авторизация на основе [учётных данных приложения по умолчанию](https://cloud.google.com/docs/authentication/production#finding_credentials_automatically). |
| remote\_base\_log\_folder | string | '' | AIRFLOW\_\_LOGGING\_\_REMOTE\_BASE\_LOG\_FOLDER | URL-адрес сегмента хранилища для удалённого ведения лога. Сегменты S3 должны начинаться с s3://. Группы лога Cloudwatch должны начинаться с cloudwatch://. Сегменты GCS должны начинаться с gs://. Сегменты WASB должны начинаться с wasb только для того, чтобы помочь Airflow выбрать правильный обработчик. Логи Stackdriver должны начинаться с stackdriver://. |
| encrypt\_s3\_logs | string | False | AIRFLOW\_\_LOGGING\_\_ENCRYPT\_S3\_LOGS | Использовать шифрование на стороне сервера для логов, хранящихся в S3. |
| logging\_level | string | INFO | AIRFLOW\_\_LOGGING\_\_LOGGING\_LEVEL | Уровень ведения лога.  Поддерживаемые значения: CRITICAL, ERROR, WARNING, INFO, DEBUG. |
| fab\_logging\_level | string | WARN | AIRFLOW\_\_LOGGING\_\_FAB\_LOGGING\_LEVEL | Уровень ведения лога для пользовательского интерфейса Flask-appbuilder.  Поддерживаемые значения: CRITICAL, ERROR, WARNING, INFO, DEBUG. |
| logging\_config\_class | string | '' | AIRFLOW\_\_LOGGING\_\_LOGGING\_CONFIG\_CLASS | Класс ведения лога. Укажите класс, который будет указывать конфигурацию ведения лога. Этот класс должен находиться в пути к классам python. Пример — my.path.default\_local\_settings.LOGGING\_CONFIG. |
| colored\_console\_log | string | True | AIRFLOW\_\_LOGGING\_\_COLORED\_CONSOLE\_LOG | Флаг для включения/выключения разноцветных логов в консоли. Окрасить логи можно, когда управляющий терминал является TTY. |
| colored\_log\_format | string | [%%(blue)s%%(asctime)s%%(reset)s] {%%(blue)s%%(filename)s:%%(reset)s%%(lineno)d} %%(log\_color)s%%(levelname)s%%(reset)s - %%(log\_color)s%%(message)s%%(reset)s | AIRFLOW\_\_LOGGING\_\_COLORED\_LOG\_FORMAT | Формат лога, когда включены разноцветные логи. |
| colored\_formatter\_class | string | airflow.utils.log.colored\_log.CustomTTYColoredFormatter | AIRFLOW\_\_LOGGING\_\_COLORED\_FORMATTER\_CLASS | — |
| log\_format | string | [%%(asctime)s] {%%(filename)s:%%(lineno)d} %%(levelname)s - %%(message)s | AIRFLOW\_\_LOGGING\_\_LOG\_FORMAT | Формат строки лога. |
| simple\_log\_format | string | %%(asctime)s %%(levelname)s - %%(message)s | AIRFLOW\_\_LOGGING\_\_SIMPLE\_LOG\_FORMAT | — |
| task\_log\_prefix\_template | string | '' | AIRFLOW\_\_LOGGING\_\_TASK\_LOG\_PREFIX\_TEMPLATE | Указывает шаблон префикса, как указано ниже, с помощью обработчика потока TaskHandlerWithCustomFormatter. Пример — {ti.dag\_id}-{ti.task\_id}-{execution\_date}-{try\_number}. |
| log\_filename\_template | string | {{ ti.dag\_id }}/{{ ti.task\_id }}/{{ ts }}/{{ try\_number }}.log | AIRFLOW\_\_LOGGING\_\_LOG\_FILENAME\_TEMPLATE | Форматирование того, как airflow генерирует имена/пути файлов для каждого запуска таска. |
| log\_processor\_filename\_template | string | {{ filename }}.log | AIRFLOW\_\_LOGGING\_\_LOG\_PROCESSOR\_FILENAME\_TEMPLATE | Форматирование того, как airflow генерирует имена файлов для лога. |
| dag\_processor\_manager\_log\_location | string | {AIRFLOW\_HOME}/logs/dag\_processor\_manager/dag\_processor\_manager.log | AIRFLOW\_\_LOGGING\_\_DAG\_PROCESSOR\_MANAGER\_LOG\_LOCATION | Полный путь к лог-файлу dag\_processor\_manager. |
| task\_log\_reader | string | task | AIRFLOW\_\_LOGGING\_\_TASK\_LOG\_READER | Имя обработчика для чтения логов инстанса таска. По умолчанию используется обработчик task. |
| extra\_loggers | string | '' | AIRFLOW\_\_LOGGING\_\_EXTRA\_LOGGERS | Перечень имён сторонних инструментов логирования, разделённых запятыми, которые будут настроены для вывода сообщений на консоли. Пример — connexion, sqlalchemy. |

Раздел [metrics]

Таблица — Раздел [metrics]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| statsd\_on | string | False | AIRFLOW\_\_METRICS\_\_STATSD\_ON | Включает отправку метрик в StatsD. |
| statsd\_host | string | localhost | AIRFLOW\_\_METRICS\_\_STATSD\_HOST | Хост StatsD. |
| statsd\_port | string | 8125 | AIRFLOW\_\_METRICS\_\_STATSD\_PORT | Порт StatsD. |
| statsd\_prefix | string | airflow | AIRFLOW\_\_METRICS\_\_STATSD\_PREFIX | Префикс StatsD. |
| statsd\_allow\_list | string | '' | AIRFLOW\_\_METRICS\_\_STATSD\_ALLOW\_LIST | Если вы не хотите отправлять все доступные метрики в StatsD, вы можете настроить разрешённый перечень префиксов (разделённых запятыми), чтобы отправлять только метрики, которые начинаются с элементов перечня (например: scheduler,executor,dagrun). |
| stat\_name\_handler | string | '' | AIRFLOW\_\_METRICS\_\_STAT\_NAME\_HANDLER | Функция, которая проверяет имя статистики statsd, при необходимости применяет изменения к имени статистики и возвращает преобразованное имя статистики.  Функция должна иметь следующую сигнатуру: def func\_name(stat\_name: str) -> str:. |
| statsd\_datadog\_enabled | string | False | AIRFLOW\_\_METRICS\_\_STATSD\_DATADOG\_ENABLED | Чтобы включить интеграцию datadog для отправки метрик airflow. |
| statsd\_datadog\_tags | string | '' | AIRFLOW\_\_METRICS\_\_STATSD\_DATADOG\_TAGS | Список тегов datadog, прикреплённых ко всем метрикам (например: key1:value1,key2:value2). |
| statsd\_custom\_client\_path | string | None | AIRFLOW\_\_METRICS\_\_STATSD\_CUSTOM\_CLIENT\_PATH | Если вы хотите использовать свой собственный клиент Statsd, укажите путь к соответствующему модулю ниже. Примечание. Путь к модулю должен существовать на вашем PYTHONPATH, чтобы Airflow мог его подобрать. |

Раздел [secrets]

Таблица — Раздел [secrets]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| backend | string | '' | AIRFLOW\_\_SECRETS\_\_BACKEND | Полное имя класса серверной части секретов для включения (будет предшествовать переменным env и metastore в пути поиска). Пример — airflow.providers.amazon.aws.secrets.systems\_manager.SystemsManagerParameterStoreBackend. |
| backend\_kwargs | string | '' | AIRFLOW\_\_SECRETS\_\_BACKEND\_KWARGS | Параметр backend\_kwargs загружается в словарь и передаётся в \_\_init\_\_ внутреннего класса секретов. Ожидается JSON. Пример для AWS Systems Manager ParameterStore: {{"connections\_prefix": "/airflow/connections", "profile\_name": "default"}}. |

Раздел [cli]

Таблица — Раздел [cli]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| api\_client | string | airflow.api.client.local\_client | AIRFLOW\_\_CLI\_\_API\_CLIENT | Каким образом клиент должен получить доступ к API. LocalClient будет использовать базу данных напрямую, а json\_client будет использовать api, запущенный на веб-сервере. |
| endpoint\_url | string | http://localhost:8080 | AIRFLOW\_\_CLI\_\_ENDPOINT\_URL | Если вы установите web\_server\_url\_prefix, не забудьте добавить его здесь, например: endpoint\_url = http://localhost:8080/myroot. api будет выглядеть так: http://localhost:8080/myroot/api/experimental/.... |

Раздел [debug]

Таблица — Раздел [debug]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| fail\_fast | string | False | AIRFLOW\_\_DEBUG\_\_FAIL\_FAST | Используется только с DebugExecutor. Если установлено значение True, DAG завершится ошибкой с первым неудачным таском. Полезно для отладки. |

Раздел [api]

Таблица — Раздел [api]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| enable\_experimental\_api | boolean | False | AIRFLOW\_\_API\_\_ENABLE\_EXPERIMENTAL\_API | Включает устаревший экспериментальный API. Обратите внимание, что эти API-интерфейсы не имеют контроля доступа. У аутентифицированного пользователя есть полный доступ. |
| auth\_backend | string | airflow.api.auth.backend.deny\_all | AIRFLOW\_\_API\_\_AUTH\_BACKEND | Как аутентифицировать пользователей API. airflow.api.auth.backend.default разрешает все запросы по историческим причинам. |
| maximum\_page\_limit | integer | 100 | AIRFLOW\_\_API\_\_MAXIMUM\_PAGE\_LIMIT | Используется для установки максимального количества страниц для запросов API. |
| fallback\_page\_limit | integer | 100 | AIRFLOW\_\_API\_\_FALLBACK\_PAGE\_LIMIT | Используется для установки лимита страниц по умолчанию, когда лимит равен нулю. Предел по умолчанию 100 установлен в спецификации OpenApi. Однако этот конкретный лимит по умолчанию работает только тогда, когда лимит установлен равным нулю (0) из запросов API. Если ограничение не указано, используется спецификация OpenApi по умолчанию. |
| google\_oauth2\_audience | string | '' | AIRFLOW\_\_API\_\_GOOGLE\_OAUTH2\_AUDIENCE | Целевая аудитория для учетных данных токена JWT, используемых для авторизации. Это значение должно совпадать на стороне клиента и сервера. Если пусто, аудитория не будет проверяться. Пример — project-id-random-value.apps.googleusercontent.com. |
| google\_key\_path | string | '' | AIRFLOW\_\_API\_\_GOOGLE\_KEY\_PATH | Путь к файлу ключа учётной записи Google Cloud Service (JSON). Если этот параметр не указан, будет использоваться авторизация на основе [учётных данных приложения по умолчанию](https://cloud.google.com/docs/authentication/production#finding_credentials_automatically). Пример — /files/service-account-json. |
| access\_control\_allow\_headers | string | '' | AIRFLOW\_\_API\_\_ACCESS\_CONTROL\_ALLOW\_HEADERS | Используется в ответ на предварительный запрос, чтобы указать, какие заголовки HTTP могут использоваться при выполнении фактического запроса. Этот заголовок является ответом сервера на заголовок браузера Access-Control-Request-Headers. |
| access\_control\_allow\_methods | string | '' | AIRFLOW\_\_API\_\_ACCESS\_CONTROL\_ALLOW\_METHODS | Задаёт метод или методы, разрешённые при доступе к ресурсу. |
| access\_control\_allow\_origin | string | '' | AIRFLOW\_\_API\_\_ACCESS\_CONTROL\_ALLOW\_ORIGIN | Указывает, может ли ответ быть передан с кодом запроса из данного источника. |

Раздел [lineage]

Таблица — Раздел [lineage]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| backend | string | '' | AIRFLOW\_\_LINEAGE\_\_BACKEND | Какой бэкэнд Lineage использовать. |

Раздел [atlas]

Таблица — Раздел [atlas]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| sasl\_enabled | string | False | AIRFLOW\_\_ATLAS\_\_SASL\_ENABLED | Включение SASL. |
| host | string | '' | AIRFLOW\_\_ATLAS\_\_HOST | Хост. |
| port | string | 21000 | AIRFLOW\_\_ATLAS\_\_PORT | Порт. |
| username | string | '' | AIRFLOW\_\_ATLAS\_\_USERNAME | Имя пользователя. |
| password | string | '' | AIRFLOW\_\_ATLAS\_\_PASSWORD  AIRFLOW\_\_ATLAS\_\_PASSWORD\_CMD  AIRFLOW\_\_ATLAS\_\_PASSWORD\_SECRET | Пароль. |

Раздел [operators]

Таблица — Раздел [operators]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| default\_owner | string | airflow | AIRFLOW\_\_OPERATORS\_\_DEFAULT\_OWNER | Владелец по умолчанию, назначаемый каждому новому оператору, если он не указан явно или не передан через default\_args. |
| default\_cpus | string | 1 | AIRFLOW\_\_OPERATORS\_\_DEFAULT\_CPUS | CPU по умолчанию. |
| default\_ram | string | 512 | AIRFLOW\_\_OPERATORS\_\_DEFAULT\_RAM | RAM по умолчанию. |
| default\_disk | string | 512 | AIRFLOW\_\_OPERATORS\_\_DEFAULT\_DISK | Ёмкость диска по умолчанию. |
| default\_gpus | string | 0 | AIRFLOW\_\_OPERATORS\_\_DEFAULT\_GPUS | — |
| default\_queue | string | default | AIRFLOW\_\_OPERATORS\_\_DEFAULT\_QUEUE | Очередь по умолчанию, которой назначаются таски и которую слушает этот обработчик. |
| allow\_illegal\_arguments | string | False | AIRFLOW\_\_OPERATORS\_\_ALLOW\_ILLEGAL\_ARGUMENTS | Разрешено передавать дополнительные/неиспользуемые аргументы (args, kwargs) оператору BaseOperator. Если установлено значение False, будет создано исключение, в противном случае будет отображаться только консольное сообщение. |

Раздел [hive]

Таблица — Раздел [hive]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| default\_hive\_mapred\_queue | string | '' | AIRFLOW\_\_HIVE\_\_DEFAULT\_HIVE\_MAPRED\_QUEUE | Очередь mapreduce по умолчанию для тасков HiveOperator. |
| mapred\_job\_name\_template | string | None | AIRFLOW\_\_HIVE\_\_MAPRED\_JOB\_NAME\_TEMPLATE | Шаблон для mapred\_job\_name в HiveOperator, поддерживает следующие именованные параметры hostname, dag\_id, task\_id, execution\_date. |

Раздел [webserver]

Таблица — Раздел [webserver]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| base\_url | string | http://localhost:8080 | AIRFLOW\_\_WEBSERVER\_\_BASE\_URL | Базовый URL-адрес вашего веб-сайта, т.к. airflow не может угадать, какой домен или имя вы используете. Это используется в автоматических электронных письмах, которые airflow отправляет по ссылкам на нужный веб-сервер. |
| default\_ui\_timezone | string | UTC | AIRFLOW\_\_WEBSERVER\_\_DEFAULT\_UI\_TIMEZONE | Часовой пояс по умолчанию для отображения всех дат в пользовательском интерфейсе может быть UTC, системной или любой строкой часового пояса IANA (например, Europe/Amsterdam). Если оставить пустым, будет использоваться значение по умолчанию core/default\_timezone. |
| web\_server\_host | string | 0.0.0.0 | AIRFLOW\_\_WEBSERVER\_\_WEB\_SERVER\_HOST | IP-адрес, указанный при запуске веб-сервера. |
| web\_server\_port | string | 8080 | AIRFLOW\_\_WEBSERVER\_\_WEB\_SERVER\_PORT | Порт для запуска веб-сервера. |
| web\_server\_ssl\_cert | string | '' | AIRFLOW\_\_WEBSERVER\_\_WEB\_SERVER\_SSL\_CERT | Пути к сертификату SSL и ключу для веб-сервера. Когда оба предоставлены, SSL будет включён. Это не меняет порт веб-сервера. |
| web\_server\_ssl\_key | string | '' | AIRFLOW\_\_WEBSERVER\_\_WEB\_SERVER\_SSL\_KEY | Пути к сертификату SSL и ключу для веб-сервера. Когда оба предоставлены, SSL будет включён. Это не меняет порт веб-сервера. |
| web\_server\_master\_timeout | string | 120 | AIRFLOW\_\_WEBSERVER\_\_WEB\_SERVER\_MASTER\_TIMEOUT | Количество секунд, в течение которых веб-сервер ждёт, прежде чем остановить мастера gunicorn, который не отвечает. |
| web\_server\_worker\_timeout | string | 120 | AIRFLOW\_\_WEBSERVER\_\_WEB\_SERVER\_WORKER\_TIMEOUT | Количество секунд, в течение которых веб-сервер gunicorn ожидает, прежде чем истечёт время ожидания для обработчика. |
| worker\_refresh\_batch\_size | string | 1 | AIRFLOW\_\_WEBSERVER\_\_WORKER\_REFRESH\_BATCH\_SIZE | Количество обработчиков, обновляемых за раз. Если установлено значение 0, обновление обработчика отключено. Если значение не равно нулю, airflow периодически обновляет обработчики веб-сервера, добавляя новых и останавливая старых. |
| worker\_refresh\_interval | string | 6000 | AIRFLOW\_\_WEBSERVER\_\_WORKER\_REFRESH\_INTERVAL | Количество секунд ожидания перед обновлением пакета обработчиков. |
| reload\_on\_plugin\_change | boolean | False | AIRFLOW\_\_WEBSERVER\_\_RELOAD\_ON\_PLUGIN\_CHANGE | Если установлено значение True, Airflow будет отслеживать файлы в каталоге plugins\_folder. Когда он обнаружит изменения, перезагрузит gunicorn. |
| secret\_key | string | {SECRET\_KEY} | AIRFLOW\_\_WEBSERVER\_\_SECRET\_KEY  AIRFLOW\_\_WEBSERVER\_\_SECRET\_KEY\_CMD  AIRFLOW\_\_WEBSERVER\_\_SECRET\_KEY\_SECRET | Секретный ключ, используемый для запуска вашего приложения flask. Он должен быть случайным насколько это возможно. Однако при запуске более 1 инстанса веб-сервера убедитесь, что все они используют один и тот же secret\_key, в противном случае один из них выдаст ошибку «CSRF session token is missing». |
| workers | string | 4 | AIRFLOW\_\_WEBSERVER\_\_WORKERS | Количество обработчиков для запуска веб-сервера Gunicorn. |
| worker\_class | string | sync | AIRFLOW\_\_WEBSERVER\_\_WORKER\_CLASS | Использование класса обработчиков Gunicorn. Возможные варианты: sync (по умолчанию), eventlet, gevent. |
| access\_logfile | string | - | AIRFLOW\_\_WEBSERVER\_\_ACCESS\_LOGFILE | Файлы лога для веб-сервера Gunicorn. Значение - означает логирование в stderr. |
| error\_logfile | string | - | AIRFLOW\_\_WEBSERVER\_\_ERROR\_LOGFILE | Файлы лога для веб-сервера Gunicorn. Значение - означает логирование в stderr. |
| access\_logformat | string | '' | AIRFLOW\_\_WEBSERVER\_\_ACCESS\_LOGFORMAT | Формат лога доступа к веб-серверу Gunicorn. Формат по умолчанию: %%(h)s %%(l)s %%(u)s %%(t)s “%%(r)s” %%(s)s %%(b)s “%%(f)s” “%%(a)s”. Документация — <https://docs.gunicorn.org/en/stable/settings.html#access-log-format>. |
| expose\_config | string | False | AIRFLOW\_\_WEBSERVER\_\_EXPOSE\_CONFIG | Размещение файла конфигурации на веб-сервере. |
| expose\_hostname | string | True | AIRFLOW\_\_WEBSERVER\_\_EXPOSE\_HOSTNAME | Размещение имени хоста на веб-сервере. |
| expose\_stacktrace | string | True | AIRFLOW\_\_WEBSERVER\_\_EXPOSE\_STACKTRACE | Размещение трассировки стека на веб-сервере. |
| dag\_default\_view | string | tree | AIRFLOW\_\_WEBSERVER\_\_DAG\_DEFAULT\_VIEW | Просмотр DAG по умолчанию. Доступные значения: tree, graph, duration, gantt, landing\_times. |
| dag\_orientation | string | LR | AIRFLOW\_\_WEBSERVER\_\_DAG\_ORIENTATION | Ориентация DAG по умолчанию. Доступные значения: LR (слева направо), TB (сверху вниз), RL (справа налево), BT (снизу вверх). |
| log\_fetch\_timeout\_sec | string | 5 | AIRFLOW\_\_WEBSERVER\_\_LOG\_FETCH\_TIMEOUT\_SEC | Время (в секундах), в течение которого веб-сервер будет ожидать первоначального «рукопожатия» при получении логов с другой машины обработчика. |
| log\_fetch\_delay\_sec | integer | 2 | AIRFLOW\_\_WEBSERVER\_\_LOG\_FETCH\_DELAY\_SEC | Интервал времени (в секундах) ожидания перед следующей загрузкой лога. |
| log\_auto\_tailing\_offset | integer | 30 | AIRFLOW\_\_WEBSERVER\_\_LOG\_AUTO\_TAILING\_OFFSET | Расстояние от низа страницы, чтобы включить автоматическое отслеживание «хвоста» лога. |
| log\_animation\_speed | integer | 1000 | AIRFLOW\_\_WEBSERVER\_\_LOG\_ANIMATION\_SPEED | Скорость анимации для автоматического отображения «хвоста» лога. |
| hide\_paused\_dags\_by\_default | string | False | AIRFLOW\_\_WEBSERVER\_\_HIDE\_PAUSED\_DAGS\_BY\_DEFAULT | По умолчанию веб-сервер показывает приостановленные DAG. Поменяйте на True, чтобы скрыть приостановленные DAG по умолчанию. |
| page\_size | string | 100 | AIRFLOW\_\_WEBSERVER\_\_PAGE\_SIZE | Единый размер страницы для всех представлений листинга в пользовательском интерфейсе. |
| navbar\_color | string | #fff | AIRFLOW\_\_WEBSERVER\_\_NAVBAR\_COLOR | Определение цвета панели навигации. |
| default\_dag\_run\_display\_number | string | 25 | AIRFLOW\_\_WEBSERVER\_\_DEFAULT\_DAG\_RUN\_DISPLAY\_NUMBER | dagrun по умолчанию для отображения в пользовательском интерфейсе. |
| enable\_proxy\_fix | boolean | False | AIRFLOW\_\_WEBSERVER\_\_ENABLE\_PROXY\_FIX | Включить промежуточное ПО werkzeug ProxyFix для обратного прокси. |
| proxy\_fix\_x\_for | integer | 1 | AIRFLOW\_\_WEBSERVER\_\_PROXY\_FIX\_X\_FOR | Количество значений, которым можно доверять для X-Forwarded-For. Подробнее: <https://werkzeug.palletsprojects.com/en/0.16.x/middleware/proxy_fix/>. |
| proxy\_fix\_x\_proto | integer | 1 | AIRFLOW\_\_WEBSERVER\_\_PROXY\_FIX\_X\_PROTO | Количество значений, которым можно доверять для X-Forwarded-Proto. |
| proxy\_fix\_x\_host | integer | 1 | AIRFLOW\_\_WEBSERVER\_\_PROXY\_FIX\_X\_HOST | Количество значений, которым можно доверять для X-Forwarded-Host. |
| proxy\_fix\_x\_port | integer | 1 | AIRFLOW\_\_WEBSERVER\_\_PROXY\_FIX\_X\_PORT | Количество значений, которым можно доверять для X-Forwarded-Port. |
| proxy\_fix\_x\_prefix | integer | 1 | AIRFLOW\_\_WEBSERVER\_\_PROXY\_FIX\_X\_PREFIX | Количество значений, которым можно доверять для X-Forwarded-Prefix. |
| cookie\_secure | string | False | AIRFLOW\_\_WEBSERVER\_\_COOKIE\_SECURE | Установить флаг безопасности для файла cookie сеанса. |
| cookie\_samesite | string | Lax | AIRFLOW\_\_WEBSERVER\_\_COOKIE\_SAMESITE | Установить политику одного сайта для файла cookie сеанса. |
| default\_wrap | boolean | False | AIRFLOW\_\_WEBSERVER\_\_DEFAULT\_WRAP | Настройка по умолчанию для переключения представления кода DAG и лога TI. |
| x\_frame\_enabled | boolean | True | AIRFLOW\_\_WEBSERVER\_\_X\_FRAME\_ENABLED | Разрешить отображение пользовательского интерфейса в кадре. |
| analytics\_tool | string | None | AIRFLOW\_\_WEBSERVER\_\_ANALYTICS\_TOOL | Отправлять анонимные действия пользователей в свой инструмент аналитики на выбор: google\_analytics, segment или metarouter. |
| analytics\_id | string | None | AIRFLOW\_\_WEBSERVER\_\_ANALYTICS\_ID | Уникальный ID вашей учётной записи в инструменте аналитики. |
| show\_recent\_stats\_for\_completed\_runs | boolean | True | AIRFLOW\_\_WEBSERVER\_\_SHOW\_RECENT\_STATS\_FOR\_COMPLETED\_RUNS | Статистика Recent Tasks будет отображаться для старых DagRuns, если она установлена. |
| update\_fab\_perms | string | True | AIRFLOW\_\_WEBSERVER\_\_UPDATE\_FAB\_PERMS | Обновление разрешений FAB и синхронизация ролей менеджера безопасности при запуске веб-сервера. |
| session\_lifetime\_minutes | integer | 43200 | AIRFLOW\_\_WEBSERVER\_\_SESSION\_LIFETIME\_MINUTES | Время жизни cookie пользовательского интерфейса в минутах. Пользователь выйдет из пользовательского интерфейса после того, как session\_lifetime минут бездействия. |
| instance\_name | string | None | AIRFLOW\_\_WEBSERVER\_\_INSTANCE\_NAME | Устанавливает настраиваемый заголовок страницы для обзорной страницы DAG и заголовок сайта для всех страниц. |

Раздел [email]

Таблица — Раздел [email]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| email\_backend | string | airflow.utils.email.send\_email\_smtp | AIRFLOW\_\_EMAIL\_\_EMAIL\_BACKEND | Серверная часть электронной почты для использования. |
| email\_conn\_id | string | smtp\_default | AIRFLOW\_\_EMAIL\_\_EMAIL\_CONN\_ID | Соединение электрнной почты для использования |
| default\_email\_on\_retry | boolean | True | AIRFLOW\_\_EMAIL\_\_DEFAULT\_EMAIL\_ON\_RETRY | Следует ли отправлять оповещения по электронной почте при повторной попытке таска. |
| default\_email\_on\_failure | boolean | True | AIRFLOW\_\_EMAIL\_\_DEFAULT\_EMAIL\_ON\_FAILURE | Следует ли отправлять оповещения по электронной почте при сбое таска. |
| subject\_template | string | None | AIRFLOW\_\_EMAIL\_\_SUBJECT\_TEMPLATE | Файл, который будет использоваться в качестве шаблона для темы электронного письма (который будет отображаться с помощью Jinja2). Если не установлен, Airflow использует базовый шаблон. Пример — /path/to/my\_subject\_template\_file. |
| html\_content\_template | string | None | AIRFLOW\_\_EMAIL\_\_HTML\_CONTENT\_TEMPLATE | Файл, который будет использоваться в качестве шаблона для содержимого электронной почты (который будет отображаться с помощью Jinja2). Если не установлен, Airflow использует базовый шаблон. Пример — /path/to/my\_html\_content\_template\_file. |

Раздел [smtp]

Таблица — Раздел [smtp]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| smtp\_host | string | localhost | AIRFLOW\_\_SMTP\_\_SMTP\_HOST | — |
| smtp\_starttls | string | True | AIRFLOW\_\_SMTP\_\_SMTP\_STARTTLS | — |
| smtp\_ssl | string | False | AIRFLOW\_\_SMTP\_\_SMTP\_SSL | — |
| smtp\_user | string | None | AIRFLOW\_\_SMTP\_\_SMTP\_USER | Пример — airflow. |
| smtp\_password | string | None | AIRFLOW\_\_SMTP\_\_SMTP\_PASSWORD  AIRFLOW\_\_SMTP\_\_SMTP\_PASSWORD\_CMD  AIRFLOW\_\_SMTP\_\_SMTP\_PASSWORD\_SECRET | Пример — airflow. |
| smtp\_port | string | 25 | AIRFLOW\_\_SMTP\_\_SMTP\_PORT | — |
| smtp\_mail\_from | string | airflow@example.com | AIRFLOW\_\_SMTP\_\_SMTP\_MAIL\_FROM | — |
| smtp\_timeout | integer | 30 | AIRFLOW\_\_SMTP\_\_SMTP\_TIMEOUT | — |
| smtp\_retry\_limit | integer | 5 | AIRFLOW\_\_SMTP\_\_SMTP\_RETRY\_LIMIT | — |

Раздел [sentry]

Интеграция с Sentry (<https://docs.sentry.io>). Здесь вы можете указать дополнительные параметры конфигурации на основе платформы Python. См. <Https://docs.sentry.io/error-reporting/configuration/?platform=python>. Неподдерживаемые параметры: integrations, in\_app\_include, in\_app\_exclude, ignore\_errors, before\_breadcrumb, before\_send, transport.

Таблица — Раздел [sentry]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| sentry\_on | string | False | AIRFLOW\_\_SENTRY\_\_SENTRY\_ON | Включить отправку отчётов об ошибках в Sentry. |
| sentry\_dsn | string | '' | AIRFLOW\_\_SENTRY\_\_SENTRY\_DSN | — |

Раздел [celery\_kubernetes\_executor]

Этот раздел применяется только в том случае, если вы используете CeleryKubernetesExecutor в разделе [core] выше.

Таблица 68 — Раздел [celery\_kubernetes\_executor]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| kubernetes\_queue | string | kubernetes | AIRFLOW\_\_CELERY\_KUBERNETES\_EXECUTOR\_\_KUBERNETES\_QUEUE | Определяет, когда отправлять таск в KubernetesExecutor при использовании CeleryKubernetesExecutor. Когда очередь таска имеет значение kubernetes\_queue (kubernetes по умолчанию), таск выполняется через KubernetesExecutor, в противном случае — через CeleryExecutor. |

Раздел [celery]

Этот раздел применяется только в том случае, если вы используете CeleryExecutor в разделе [core] выше.

Таблица 69 — Раздел [celery]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| celery\_app\_name | string | airflow.executors.celery\_executor | AIRFLOW\_\_CELERY\_\_CELERY\_APP\_NAME | Название приложения, которое будет использовать celery. |
| worker\_concurrency | string | 16 | AIRFLOW\_\_CELERY\_\_WORKER\_CONCURRENCY | Параллелизм, который будет использоваться при запуске процессов обработчиков с помощью команды airflow celery worker. Это определяет количество инстансов таска, которые будет выполнять обработчик, поэтому размер ваших обработчиков зависит от ресурсов в вашем поле обработчика и характера ваших тасков. |
| worker\_autoscale | string | None | AIRFLOW\_\_CELERY\_\_WORKER\_AUTOSCALE | Максимальный и минимальный параллелизм, который будет использоваться при запуске обработчиков с помощью команды airflow celery worker (всегда сохраняйте минимальное количество процессов, но при необходимости увеличивайте до максимума). Обратите внимание, что значение должно быть max\_concurrency,min\_concurrency. Выберите эти числа в зависимости от ресурсов на в поле обработчиков и характера таска. Если доступна опция автомасштабирования, worker\_concurrency игнорируется. <http://docs.celeryproject.org/en/latest/reference/celery.bin.worker.html#cmdoption-celery-worker-autoscale>  Пример — 16,12. |
| worker\_prefetch\_multiplier | integer | None | AIRFLOW\_\_CELERY\_\_WORKER\_PREFETCH\_MULTIPLIER | Используется для увеличения количества тасков, которые выполняют предварительную выборку, что может повысить производительность. Количество процессов, умноженное на worker\_prefetch\_multiplier, — это количество тасков, которые предварительно выбираются обработчиками. Значение больше 1 может привести к излишней блокировке тасков, если есть несколько обработчиков, и один обработчик предварительно выбирает таски, которые находятся за долго выполняющимися тасками, в то время как другой обработчик имеет неиспользуемые процессы, которые не могут обрабатывать уже заявленные заблокированные таски. <https://docs.celeryproject.org/en/stable/userguide/optimizing.html#prefetch-limits>  Пример — 1. |
| worker\_log\_server\_port | string | 8793 | AIRFLOW\_\_CELERY\_\_WORKER\_LOG\_SERVER\_PORT | Когда вы запускаете обработчика Airflow, Airflow запускает крошечный подпроцесс веб-сервера для обслуживания локальных файлов лога обработчика на основном веб-сервере Airflow, который затем создаёт страницы и отправляет их пользователям. Это определяет порт, на котором обслуживаются логи. Он должен быть неиспользованным и открываться видимым с основного веб-сервера для подключения к обработчикам. |
| worker\_umask | string | 0o077 | AIRFLOW\_\_CELERY\_\_WORKER\_UMASK | Umask, который будет использоваться при запуске airflow celery worker в режиме демона. Это контролирует маску режима создания файла, которая определяет начальное значение битов прав доступа к файлам для вновь создаваемых файлов. |
| broker\_url | string | redis://redis:6379/0 | AIRFLOW\_\_CELERY\_\_BROKER\_URL  AIRFLOW\_\_CELERY\_\_BROKER\_URL\_CMD  AIRFLOW\_\_CELERY\_\_BROKER\_URL\_SECRET | URL-адрес брокера Celery. Celery поддерживает RabbitMQ, Redis и экспериментально базу данных sqlalchemy. |
| result\_backend | string | db+postgresql://postgres:airflow@postgres/airflow | AIRFLOW\_\_CELERY\_\_RESULT\_BACKEND  AIRFLOW\_\_CELERY\_\_RESULT\_BACKEND\_CMD  AIRFLOW\_\_CELERY\_\_RESULT\_BACKEND\_SECRET | Celery result\_backend. По завершении работы джоба необходимо обновить его метаданные. Поэтому он отправит сообщение в шину сообщений или вставит его в базу данных (в зависимости от серверной части). Этот статус используется планировщиком для обновления состояния таска. Настоятельно рекомендуется использовать базу данных <http://docs.celeryproject.org/en/latest/userguide/configuration.html#task-result-backend-settings>. |
| flower\_host | string | 0.0.0.0 | AIRFLOW\_\_CELERY\_\_FLOWER\_HOST | Celery Flower — это приятный пользовательский интерфейс для Celery. У Airflow есть ярлык, чтобы запустить airflow celery flower. Конфигурация определяет IP-адрес, на котором работает Celery Flower. |
| flower\_url\_prefix | string | '' | AIRFLOW\_\_CELERY\_\_FLOWER\_URL\_PREFIX | Корневой URL-адрес для Flower. Пример — /flower. |
| flower\_port | string | 5555 | AIRFLOW\_\_CELERY\_\_FLOWER\_PORT | Это определяет порт, на котором работает Celery Flower. |
| flower\_basic\_auth | string | '' | AIRFLOW\_\_CELERY\_\_FLOWER\_BASIC\_AUTH  AIRFLOW\_\_CELERY\_\_FLOWER\_BASIC\_AUTH\_CMD  AIRFLOW\_\_CELERY\_\_FLOWER\_BASIC\_AUTH\_SECRET | Securing Flowerс помощью базовой проверки подлинности принимает пары user:password, разделённые запятыми. Пример — user1:password1,user2:password2. |
| sync\_parallelism | string | 0 | AIRFLOW\_\_CELERY\_\_SYNC\_PARALLELISM | Сколько процессов использует CeleryExecutor для синхронизации состояния таска. 0 означает использование max(1,количество ядер-1) процессов. |
| celery\_config\_options | string | airflow.config\_templates.default\_celery.DEFAULT\_CELERY\_CONFIG | AIRFLOW\_\_CELERY\_\_CELERY\_CONFIG\_OPTIONS | Путь импорта для параметров конфигурации Celery. |
| ssl\_active | string | False | AIRFLOW\_\_CELERY\_\_SSL\_ACTIVE | Включение SSL. |
| ssl\_key | string | '' | AIRFLOW\_\_CELERY\_\_SSL\_KEY | Ключ SSL. |
| ssl\_cert | string | '' | AIRFLOW\_\_CELERY\_\_SSL\_CERT | Сертификат SSL. |
| ssl\_cacert | string | '' | AIRFLOW\_\_CELERY\_\_SSL\_CACERT | — |
| pool | string | prefork | AIRFLOW\_\_CELERY\_\_POOL | Имплементация Celery Pool. Доступны следующие варианты: prefork (по умолчанию), eventlet, gevent или solo. См. <https://docs.celeryproject.org/en/latest/userguide/workers.html#concurrency>, <https://docs.celeryproject.org/en/latest/userguide/concurrency/eventlet.html>. |
| operation\_timeout | float | 1.0 | AIRFLOW\_\_CELERY\_\_OPERATION\_TIMEOUT | Количество секунд ожидания перед истечением времени ожидания операций send\_task\_to\_executor или fetch\_celery\_task\_state. |
| task\_track\_started | boolean | True | AIRFLOW\_\_CELERY\_\_TASK\_TRACK\_STARTED | Таск Celery сообщит о своем статусе как started, когда таск выполняется обработчиком. Это используется в Airflow для отслеживания запущенных тасков, и если планировщик перезапущен или запущен в режиме высокой доступности, он может принять потерянные таски, запущенные предыдущим SchedulerJob. |
| task\_adoption\_timeout | integer | 600 | AIRFLOW\_\_CELERY\_\_TASK\_ADOPTION\_TIMEOUT | Время в секундах, по истечении которого принятые таски очищаются CeleryExecutor. Это полезно для удаления застрявших тасков. |
| task\_publish\_max\_retries | integer | 3 | AIRFLOW\_\_CELERY\_\_TASK\_PUBLISH\_MAX\_RETRIES | Максимальное количество повторных попыток публикации сообщений о тасках брокеру при сбое из-за ошибки AirflowTaskTimeout до отказа и пометки таска как сбойного. |
| worker\_precheck | string | False | AIRFLOW\_\_CELERY\_\_WORKER\_PRECHECK | Проверка инициализации обработчика для валидации подключения к базе данных метаданных. |

Раздел [celery\_broker\_transport\_options]

Этот раздел предназначен для указания параметров, которые могут быть переданы базовому транспорту брокера celery. См. <Http://docs.celeryproject.org/en/latest/userguide/configuration.html#std:setting-broker_transport_options>.

Таблица 70 — Раздел [celery\_broker\_transport\_options]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| visibility\_timeout | string | None | AIRFLOW\_\_CELERY\_BROKER\_TRANSPORT\_OPTIONS\_\_VISIBILITY\_TIMEOUT | Тайм-аут видимости определяет количество секунд, в течение которых обработчик подтвердит выполнение таска, прежде чем сообщение будет повторно доставлено другому обработчику. Обязательно увеличьте тайм-аут видимости, чтобы он совпадал со временем самого длительного расчётного времени прибытия, которое вы планируете использовать. visibility\_timeout поддерживается только брокерами Celery Redis и SQS. См. <http://docs.celeryproject.org/en/master/userguide/configuration.html#std:setting-broker_transport_options>.  Пример — 21600. |

Раздел [dask]

Этот раздел применяется только в том случае, если вы используете DaskExecutor в разделе [core] выше.

Таблица 71 — Раздел [dask]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| cluster\_address | string | 127.0.0.1:8786 | AIRFLOW\_\_DASK\_\_CLUSTER\_ADDRESS | IP-адрес и порт планировщика кластера Dask. |
| tls\_ca | string | '' | AIRFLOW\_\_DASK\_\_TLS\_CA | Настройки TLS / SSL для доступа к защищенному планировщику Dask. |
| tls\_cert | string | '' | AIRFLOW\_\_DASK\_\_TLS\_CERT | — |
| tls\_key | string | '' | AIRFLOW\_\_DASK\_\_TLS\_KEY | — |

Раздел [scheduler]

Таблица 72 — Раздел [scheduler]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| job\_heartbeat\_sec | string | 5 | AIRFLOW\_\_SCHEDULER\_\_JOB\_HEARTBEAT\_SEC | Инстансы тасков прослушивают внешний сигнал kill (когда вы удаляете таски из интерфейса командной строки или пользовательского интерфейса), это определяет частоту, с которой они должны прослушивать (в секундах). |
| clean\_tis\_without\_dagrun\_interval | float | 15.0 | AIRFLOW\_\_SCHEDULER\_\_CLEAN\_TIS\_WITHOUT\_DAGRUN\_INTERVAL | Как часто (в секундах) проверять и приводить в порядок running TaskInstancess, у которых больше нет подходящего DagRun. |
| scheduler\_heartbeat\_sec | string | 5 | AIRFLOW\_\_SCHEDULER\_\_SCHEDULER\_HEARTBEAT\_SEC | Планировщик постоянно пытается запускать новые таски. Это определяет, как часто планировщик должен запускаться (в секундах). |
| num\_runs | string | -1 | AIRFLOW\_\_SCHEDULER\_\_NUM\_RUNS | Количество попыток запланировать каждый файл DAG.  -1 означает неограниченное количество. |
| processor\_poll\_interval | string | 1 | AIRFLOW\_\_SCHEDULER\_\_PROCESSOR\_POLL\_INTERVAL | Количество секунд ожидания между последовательной обработкой файла DAG. |
| min\_file\_process\_interval | string | 30 | AIRFLOW\_\_SCHEDULER\_\_MIN\_FILE\_PROCESS\_INTERVAL | Количество секунд, по истечении которых выполняется анализ файла DAG. Файл DAG анализируется каждые min\_file\_process\_interval количество секунд. Обновления для DAG отражаются по истечении этого интервала. Удержание этого числа на низком уровне увеличит загрузку ЦП. |
| dag\_dir\_list\_interval | string | 300 | AIRFLOW\_\_SCHEDULER\_\_DAG\_DIR\_LIST\_INTERVAL | Как часто (в секундах) сканировать каталог DAG на наличие новых файлов. По умолчанию 5 минут. |
| print\_stats\_interval | string | 30 | AIRFLOW\_\_SCHEDULER\_\_PRINT\_STATS\_INTERVAL | Как часто следует выводить статистику в логи. Установка значения 0 отключит печать статистики. |
| pool\_metrics\_interval | float | 5.0 | AIRFLOW\_\_SCHEDULER\_\_POOL\_METRICS\_INTERVAL | Как часто (в секундах) статистика использования пула должна отправляться в statsd (если statsd\_on включён). |
| scheduler\_health\_check\_threshold | string | 30 | AIRFLOW\_\_SCHEDULER\_\_SCHEDULER\_HEALTH\_CHECK\_THRESHOLD | Если последнее heartbeat-сообщение планировщика произошло более чем scheduler\_health\_check\_threshold назад (в секундах), планировщик считается неработоспособным. Это используется проверкой работоспособности в конечной точке /health. |
| orphaned\_tasks\_check\_interval | float | 300.0 | AIRFLOW\_\_SCHEDULER\_\_ORPHANED\_TASKS\_CHECK\_INTERVAL | Как часто (в секундах) планировщик должен проверять потерянные задачи и SchedulerJobs. |
| child\_process\_log\_directory | string | {AIRFLOW\_HOME}/logs/scheduler | AIRFLOW\_\_SCHEDULER\_\_CHILD\_PROCESS\_LOG\_DIRECTORY | — |
| scheduler\_zombie\_task\_threshold | string | 300 | AIRFLOW\_\_SCHEDULER\_\_SCHEDULER\_ZOMBIE\_TASK\_THRESHOLD | Джобы локальных тасков периодически отправляют heartbeat-сообщение в базу данных. Если в течение этого количества секунд джоб не отправляет heartbeat-сообщение, планировщик пометит связанный инстанс таска как сбойный и перепланирует таск. |
| catchup\_by\_default | string | True | AIRFLOW\_\_SCHEDULER\_\_CATCHUP\_BY\_DEFAULT | Отключает отслеживание планировщика, установив для него значение False. Поведение по умолчанию не изменилось, и Command Line Backfills по-прежнему работают, но планировщик не будет выполнять отслеживание планировщика, если оно имеет значение False, однако его можно установить для каждого DAG в определении DAG (catchup). |
| max\_tis\_per\_query | string | 512 | AIRFLOW\_\_SCHEDULER\_\_MAX\_TIS\_PER\_QUERY | Это изменяет размер пакета запросов в основном цикле планирования. Если оно слишком велико, на производительность SQL-запроса может повлиять одно или несколько из следующего:  возврат к полному сканированию таблицы;  сложность предиката запроса;  чрезмерная блокировка  Кроме того, вы можете достичь максимально допустимой длины запроса для вашей базы данных. Установите 0, чтобы не было ограничений (не рекомендуется). |
| use\_row\_level\_locking | boolean | True | AIRFLOW\_\_SCHEDULER\_\_USE\_ROW\_LEVEL\_LOCKING | Должен ли планировщик выдаёт SELECT ... FOR UPDATE в соответствующих запросах. Если установлено значение False, вам не следует запускать более одного планировщика одновременно. |
| max\_dagruns\_to\_create\_per\_loop | string | None | AIRFLOW\_\_SCHEDULER\_\_MAX\_DAGRUNS\_TO\_CREATE\_PER\_LOOP | Максимальное количество DAG для создания DagRuns для цикла планировщика. По умолчанию — 10. |
| max\_dagruns\_per\_loop\_to\_schedule | string | None | AIRFLOW\_\_SCHEDULER\_\_MAX\_DAGRUNS\_PER\_LOOP\_TO\_SCHEDULE | Сколько DagRuns должен проверить (и заблокировать) планировщик при планировании и постановке тасков в очередь. По умолчанию — 20. |
| schedule\_after\_task\_execution | boolean | None | AIRFLOW\_\_SCHEDULER\_\_SCHEDULE\_AFTER\_TASK\_EXECUTION | Если процесс диспетчера тасков выполняет «мини-планировщик», чтобы попытаться запланировать больше тасков той же группы обеспечения доступности баз данных. Если оставить этот параметр включённым, таски в тех же DAG будут выполняться быстрее, но в некоторых случаях это может привести к тому, что другие DAG-файлы не будут работать. По умолчанию — True. |
| parsing\_processes | string | 2 | AIRFLOW\_\_SCHEDULER\_\_PARSING\_PROCESSES | Планировщик может запускать несколько процессов параллельно для анализа DAG. Это определяет, сколько процессов будет запущено. |
| file\_parsing\_sort\_mode | string | modified\_time | AIRFLOW\_\_SCHEDULER\_\_FILE\_PARSING\_SORT\_MODE | Один из вариантов: modified\_time, random\_seeded\_by\_host и alphabetical. Планировщик перечислит и отсортирует файлы DAG, чтобы определить порядок анализа.  modified\_time — сортировать файлы по времени изменения. Это полезно в больших масштабах, чтобы сначала проанализировать недавно изменённые DAG;  random\_seeded\_by\_host — произвольная сортировка по нескольким планировщикам, но в том же порядке на одном и том же хосте. Это полезно при работе с планировщиком в режиме высокой доступности, когда каждый планировщик может анализировать разные файлы DAG;  alphabetical — сортировка по имени файла. |
| use\_job\_schedule | string | True | AIRFLOW\_\_SCHEDULER\_\_USE\_JOB\_SCHEDULE | Отключает планировщик для использования интервалов cron, установив для него значение False. DAG, отправленные вручную в веб-интерфейсе или с помощью trigger\_dag, по-прежнему будут работать. |
| allow\_trigger\_in\_future | string | False | AIRFLOW\_\_SCHEDULER\_\_ALLOW\_TRIGGER\_IN\_FUTURE | Разрешить запускаемые извне DagRuns для дат выполнения в будущем. Имеет эффект, только если для schedule\_interval установлено значение None в DAG. |
| dependency\_detector | string | airflow.serialization.serialized\_objects.DependencyDetector | AIRFLOW\_\_SCHEDULER\_\_DEPENDENCY\_DETECTOR | Класс детектора зависимостей DAG для использования. |

Раздел [kerberos]

Таблица 73 — Раздел [kerberos]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| ccache | string | /tmp/airflow\_krb5\_ccache | AIRFLOW\_\_KERBEROS\_\_CCACHE | — |
| principal | string | airflow | AIRFLOW\_\_KERBEROS\_\_PRINCIPAL | Дополняется с помощью fqdn. |
| reinit\_frequency | string | 3600 | AIRFLOW\_\_KERBEROS\_\_REINIT\_FREQUENCY | — |
| kinit\_path | string | kinit | AIRFLOW\_\_KERBEROS\_\_KINIT\_PATH | — |
| keytab | string | airflow.keytab | AIRFLOW\_\_KERBEROS\_\_KEYTAB | — |

Раздел [github\_enterprise]

Таблица 74 — Раздел [github\_enterprise]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| api\_rev | string | v3 | AIRFLOW\_\_GITHUB\_ENTERPRISE\_\_API\_REV | — |

Раздел [elasticsearch]

Таблица 75 — Раздел [elasticsearch]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| host | string | '' | AIRFLOW\_\_ELASTICSEARCH\_\_HOST | Хост Elasticsearch. |
| log\_id\_template | string | {dag\_id}-{task\_id}-{execution\_date}-{try\_number} | AIRFLOW\_\_ELASTICSEARCH\_\_LOG\_ID\_TEMPLATE | Формат log\_id, который используется для запроса логов заданных тасков. |
| end\_of\_log\_mark | string | end\_of\_log | AIRFLOW\_\_ELASTICSEARCH\_\_END\_OF\_LOG\_MARK | Используется для обозначения конца потока лога для таска. |
| frontend | string | '' | AIRFLOW\_\_ELASTICSEARCH\_\_FRONTEND | Квалифицированный URL-адрес для внешнего интерфейса elasticsearch (например, Kibana) с аргументом шаблона для log\_id. Код будет создавать log\_id с использованием шаблона log\_id из аргумента выше.  Примечание. Код автоматически добавит префикс https://, не включайте его здесь. |
| write\_stdout | string | False | AIRFLOW\_\_ELASTICSEARCH\_\_WRITE\_STDOUT | Записывать логи тасков в stdout обработчика, а не в файлы по умолчанию. |
| json\_format | string | False | AIRFLOW\_\_ELASTICSEARCH\_\_JSON\_FORMAT | Вместо форматирования лога по умолчанию записывать строки лога как JSON. |
| json\_fields | string | asctime, filename, lineno, levelname, message | AIRFLOW\_\_ELASTICSEARCH\_\_JSON\_FIELDS | Поля лога, которые также необходимо прикрепить к выходным данным json, если они включены. |
| host\_field | string | host | AIRFLOW\_\_ELASTICSEARCH\_\_HOST\_FIELD | Поле, в котором хранится имя хоста (обычно host или host.name). |
| offset\_field | string | offset | AIRFLOW\_\_ELASTICSEARCH\_\_OFFSET\_FIELD | Поле, в котором хранится смещение (обычно либо offset, либо log.offset). |

Раздел [elasticsearch\_configs]

Таблица 76 — Раздел [elasticsearch\_configs]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| use\_ssl | string | False | AIRFLOW\_\_ELASTICSEARCH\_CONFIGS\_\_USE\_SSL | — |
| verify\_certs | string | True | AIRFLOW\_\_ELASTICSEARCH\_CONFIGS\_\_VERIFY\_CERTS | — |

Раздел [kubernetes]

Таблица 77 — Раздел [kubernetes]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| pod\_template\_file | string | '' | AIRFLOW\_\_KUBERNETES\_\_POD\_TEMPLATE\_FILE | Путь к файлу пода YAML. Если установлено, все остальные поля, относящиеся к kubernetes, игнорируются. |
| worker\_container\_repository | string | '' | AIRFLOW\_\_KUBERNETES\_\_WORKER\_CONTAINER\_REPOSITORY | Репозиторий образа Kubernetes для запуска обработчика. |
| worker\_container\_tag | string | '' | AIRFLOW\_\_KUBERNETES\_\_WORKER\_CONTAINER\_TAG | Тег образа Kubernetes для запуска обработчика. |
| namespace | string | default | AIRFLOW\_\_KUBERNETES\_\_NAMESPACE | Пространство имён Kubernetes, в котором должны быть созданы обработчики airflow. По умолчанию — default. |
| delete\_worker\_pods | string | True | AIRFLOW\_\_KUBERNETES\_\_DELETE\_WORKER\_PODS | Если True, все поды обработчиков будут удалены при завершении работы. |
| delete\_worker\_pods\_on\_failure | string | False | AIRFLOW\_\_KUBERNETES\_\_DELETE\_WORKER\_PODS\_ON\_FAILURE | Если False (а delete\_worker\_pods — True), отказавшие поды обработчиков не будут удалены, чтобы пользователи могли их исследовать. Это предотвращает удаление подов обработчиков только в случае сбоя самого обработчика, а не в случае сбоя выполняемого им таска. |
| worker\_pods\_creation\_batch\_size | string | 1 | AIRFLOW\_\_KUBERNETES\_\_WORKER\_PODS\_CREATION\_BATCH\_SIZE | Количество вызовов создания Kubernetes Worker Pod на цикл планировщика. Обратите внимание, что текущее значение по умолчанию 1 будет запускать только один под за heartbeat-такт. Настоятельно рекомендуется, чтобы пользователи увеличивали это число в соответствии с допуском их кластера Kubernetes для лучшей производительности. |
| multi\_namespace\_mode | boolean | False | AIRFLOW\_\_KUBERNETES\_\_MULTI\_NAMESPACE\_MODE | Позволяет пользователям запускать поды в нескольких пространствах имён. Потребуется создание кластерной роли для планировщика. |
| in\_cluster | string | True | AIRFLOW\_\_KUBERNETES\_\_IN\_CLUSTER | Используйте служебную учетную запись, которую Kubernetes предоставляет подам для подключения к кластеру Kubernetes. Он предназначен для клиентов, которые планируют работать внутри пода, работающего на Kubernetes. Он вызовет исключение, если будет вызван из процесса, не работающего в среде Kubernetes. |
| cluster\_context | string | None | AIRFLOW\_\_KUBERNETES\_\_CLUSTER\_CONTEXT | При запуске с in\_cluster=False измените параметры cluster\_context или config\_file по умолчанию на клиент Kubernetes. Оставьте это поле пустым, чтобы использовать поведение по умолчанию, такое как у kubectl. |
| config\_file | string | None | AIRFLOW\_\_KUBERNETES\_\_CONFIG\_FILE | Путь к конфигурационному файлу kubernetes, который будет использоваться, если для параметра in\_cluster установлено значение False. |
| kube\_client\_request\_args | string | '' | AIRFLOW\_\_KUBERNETES\_\_KUBE\_CLIENT\_REQUEST\_ARGS | Параметры ключевого слова, передаваемые при вызове методов core\_v1\_api клиента kubernetes из Kubernetes Executor, предоставляемых в виде однострочной строки словаря JSON. Список поддерживаемых параметров одинаков для всех core\_v1\_apis, поэтому для всех apis используется одна переменная конфигурации. См. <https://raw.githubusercontent.com/kubernetes-client/python/41f11a09995efcd0142e25946adc7591431bfb2f/kubernetes/client/api/core_v1_api.py>. |
| delete\_option\_kwargs | string | '' | AIRFLOW\_\_KUBERNETES\_\_DELETE\_OPTION\_KWARGS | Необязательные аргументы ключевого слова для передачи delete\_namespaced\_pod клиентскому методу kubernetes core\_v1\_api при использовании Kubernetes Executor. Это должен быть объект, который может содержать любые параметры, перечисленные в классе v1DeleteOptions, определенном здесь: <https://github.com/kubernetes-client/python/blob/41f11a09995efcd0142e25946adc7591431bfb2f/kubernetes/client/models/v1_delete>. Пример — {"grace\_period\_seconds": 10}. |
| enable\_tcp\_keepalive | boolean | True | AIRFLOW\_\_KUBERNETES\_\_ENABLE\_TCP\_KEEPALIVE | Включает механизм поддержки активности TCP. Это предотвращает зависание запросов Kubernetes API на неопределенное время, когда время простоя подключения истекло для таких сервисов, как облачные балансировщики нагрузки или брандмауэры. |
| tcp\_keep\_idle | integer | 120 | AIRFLOW\_\_KUBERNETES\_\_TCP\_KEEP\_IDLE | Когда опция enable\_tcp\_keepalive включена, TCP проверяет соединение, которое неактивно в течение tcp\_keep\_idle секунд. |
| tcp\_keep\_intvl | integer | 30 | AIRFLOW\_\_KUBERNETES\_\_TCP\_KEEP\_INTVL | Когда включена опция enable\_tcp\_keepalive, если Kubernetes API не отвечает на зонд проверки активности, TCP повторно передает зонд через tcp\_keep\_intvl секунд. |
| tcp\_keep\_cnt | integer | 6 | AIRFLOW\_\_KUBERNETES\_\_TCP\_KEEP\_CNT | Когда включена опция enable\_tcp\_keepalive, если Kubernetes API не отвечает на зонд поддержки активности, TCP повторно передает зонд tcp\_keep\_cnt количество раз, прежде чем соединение будет считаться разорванным. |
| verify\_ssl | boolean | True | AIRFLOW\_\_KUBERNETES\_\_VERIFY\_SSL | Установите значение false, чтобы пропустить проверку SSL-сертификата клиента Python Kubernetes. |
| worker\_pods\_pending\_timeout | integer | 300 | AIRFLOW\_\_KUBERNETES\_\_WORKER\_PODS\_PENDING\_TIMEOUT | Как долго в секундах обработчик может находиться в состоянии ожидания, прежде чем он будет признан сбойным. |
| worker\_pods\_pending\_timeout\_check\_interval | integer | 120 | AIRFLOW\_\_KUBERNETES\_\_WORKER\_PODS\_PENDING\_TIMEOUT\_CHECK\_INTERVAL | Как часто (в секундах) проверять, не превысили ли ожидающие обработчики тайм-ауты. |
| worker\_pods\_pending\_timeout\_batch\_size | integer | 100 | AIRFLOW\_\_KUBERNETES\_\_WORKER\_PODS\_PENDING\_TIMEOUT\_BATCH\_SIZE | Сколько ожидающих подов нужно проверять на нарушение тайм-аута в каждом интервале проверки. Вы можете захотеть увеличить это значение, если у вас очень большой кластер и / или вы используете multi\_namespace\_mode. |

Раздел [smart\_sensor]

Таблица 78 — Раздел [smart\_sensor]

| Конфигурация | Тип | Значение по умолчанию | Переменная среды | Описание |
| --- | --- | --- | --- | --- |
| use\_smart\_sensor | boolean | False | AIRFLOW\_\_SMART\_SENSOR\_\_USE\_SMART\_SENSOR | Когда use\_smart\_sensor имеет значение True, Airflow перенаправляет несколько квалифицированных тасков с сенсором в таск с интеллектуальным сенсором. |
| shard\_code\_upper\_limit | integer | 10000 | AIRFLOW\_\_SMART\_SENSOR\_\_SHARD\_CODE\_UPPER\_LIMIT | shard\_code\_upper\_limit — верхний предел значения shard\_code. shard\_code генерируется hashcode % shard\_code\_upper\_limit. |
| shards | integer | 5 | AIRFLOW\_\_SMART\_SENSOR\_\_SHARDS | Количество запущенных процессов с интеллектуальным сенсором для каждой службы. |
| sensors\_enabled | string | NamedHivePartitionSensor | AIRFLOW\_\_SMART\_SENSOR\_\_SENSORS\_ENABLED | Поддержка классов сенсоров, разделённых запятыми, в smart\_sensor. |