Debezium

дистрибутив, деплой и сравнение с Oracle GoldenGate

- Debezium для CDC-репликации Oracle → Oracle

Debezium для CDC-репликации Oracle → Oracle

Материал собран по официальным источникам Debezium (blog, GitHub, документация), официальной документации Oracle GoldenGate и обсуждениям в issue-трекере/форуме проекта.

1. Где брать дистрибутив

- **Maven-артефакт:** `io.debezium:debezium-connector-oracle` — доступен на Maven Central / Sonatype (central.sonatype.com, mvnrepository.com). Актуальная на середину 2026 линейка — 3.6.x.
- **Docker-образы:** с релиза **Debezium 3.0.0.Final** новые образы (2.7.x и 3.x+) публикуются **только на quay.io/debezium**; на Docker Hub (docker.io) остаются исторические образы 1.x/2.x вплоть до 3.0.0.Final включительно, новых версий там больше не появляется — официально анонсировано в блоге Debezium (debezium.io/blog/2024/09/18/quay-io-reminder).
- **GitHub:** архивы плагина коннектора (tar.gz с jar-файлами для Kafka Connect plugin path) публикуются в релизах репозитория `debezium/debezium`.
- **Актуальная версия:** финальный релиз **Debezium 3.6** вышел 1 июля 2026 (debezium.io/blog/2026/07/01/debezium-3-6-final-release), предыдущая стабильная серия — 3.5 (последний патч 3.5.2.Final, 2 июня 2026, debezium.io/blog/2026/06/02/debezium-3-5-2-final-released). Номера версий стоит перепроверить на debezium.io/releases непосредственно перед установкой — релизы выходят часто.

2. Краткая инструкция по развёртыванию Oracle → Oracle

Debezium — это **только источник CDC** (capture + публикация в Kafka), а не готовый инструмент репликации. Доставки «из коробки» в целевую Oracle-БД у Debezium нет — нужен отдельный sink-коннектор.

2.1. Подготовка источника Oracle

- **ARCHIVELOG mode обязателен** — без него LogMiner не сможет работать.
- **Supplemental logging** — нужен минимум database-level supplemental logging; для полноценной фиксации значений колонок в redo-логах дополнительно нужен table-level supplemental logging уровня `ALL COLUMNS`. Начиная с **Debezium 3.4** коннектор распознаёт этот табличный режим как эквивалент минимального, поэтому нужен один из двух вариантов, а не оба одновременно (тикет DBZ-7341).
- **Права пользователя коннектора** — отдельный технический пользователь с грантами вида `CREATE TABLE`, `CREATE SEQUENCE`, `LOGMINING`, `SELECT ANY TRANSACTION`, `SELECT ANY DICTIONARY` и доступом к представлениям LogMiner.
- **Multitenant (CDB/PDB)** — если Oracle работает в архитектуре CDB/PDB, обязательно нужно задать `database.pdb.name`; без этого параметра коннектор майнит только корневую CDB и пропускает изменения в PDB.

2.2. Компоненты инфраструктуры

1. Кластер Kafka + Kafka Connect (distributed mode) — стандартная точка развёртывания коннекторов Debezium.
2. Плагин `debezium-connector-oracle` — устанавливается в plugin path воркеров Kafka Connect (из образа quay.io/debezium или архива с GitHub Releases).
3. **Sink для доставки в целевую Oracle** — на выбор:
   - официальный **`debezium-connector-jdbc`** (часть монорепозитория Debezium, `debezium.io/documentation/reference/stable/connectors/jdbc.html`) — «понимает» нативную структуру событий Debezium без дополнительных трансформаций;
   - сторонний (не Debezium) **Confluent Kafka Connect JDBC Sink Connector** — тоже официально поддерживает Oracle как target через MERGE-upsert, но с задокументированным ограничением: upsert ломается на первичном ключе типа `CHAR`, работает на `VARCHAR2` (docs.confluent.io/kafka-connectors/jdbc).

2.3. Ключевые параметры конфигурации source-коннектора

- `database.connection.adapter` — режим захвата:
  - `logminer` (по умолчанию) — LogMiner в режиме некоммиченных изменений, Debezium сам буферизует транзакции;
  - `logminer_unbuffered` — LogMiner отдаёт только закоммиченные изменения, экономит память коннектора, но перекладывает нагрузку на PGA источника (проблема для больших транзакций);
  - `olr` — через сторонний open-source OpenLogReplicator;
  - `xstream` — через Oracle XStream API, **требует лицензии на продукт Oracle GoldenGate** для production-использования, даже если сам GoldenGate физически не установлен (README debezium-connector-oracle).
- `log.mining.strategy`:
  - `redo_log_catalog` (по умолчанию) — самый надёжный вариант отслеживания DDL вперемешку с DML (LogMiner интерполирует между дамп-снимками словаря данных), но самый дорогой — форсирует переключение redo-логов и генерирует больше архивных логов;

    - `online_catalog` — заметно быстрее, но не отслеживает DDL-изменения; подходит, если схема таблиц стабильна.
- Данные о `snapshot.mode` и точный список привилегий для XStream в собранном пуле не были независимо подтверждены до высокой степени детализации — перед продакшен-настройкой сверяйтесь напрямую с `debezium.io/documentation/reference/stable/connectors/oracle.html`.

3. Сравнение с Oracle GoldenGate

	Oracle GoldenGate	Debezium (Oracle connector)
Архитектура	Единый коммерческий продукт: **Extract** читает redo/архивные логи и пишет в **Trail**-файлы; **Replicat** читает Trail и применяет изменения на target — либо через OCI (non-integrated), либо через LCR и inbound server (integrated mode) (docs.oracle.com — GoldenGate processes)	Только capture: коннектор публикует события в Kafka-топик. Доставка на target — отдельный, независимо настраиваемый sink-коннектор
Доставка на target	Встроена (Replicat)	Не встроена — нужен `debezium-connector-jdbc` или сторонний JDBC sink
Лицензирование	Платный продукт Oracle	Debezium — open source (Apache 2.0), но режим **XStream** внутри Oracle-коннектора Debezium требует лицензии GoldenGate
DDL	Integrated Replicat применяет DDL напрямую на target	Debezium парсит DDL из redo-потока, обновляет схему в памяти и пишет её в отдельный Kafka-топик schema history; при рестарте схема восстанавливается разбором этого топика с начала до точки рестарта
LOB/XMLType	Официально документировано Oracle как часть поддерживаемых типов GoldenGate (docs.oracle.com — data types) — детального сравнения в собранных источниках нет	CLOB/NCLOB/BLOB: неизменившееся значение не попадает в событие (плейсхолдер вместо значения). XMLType добавлен в 2.4, требует `lob.enabled=true`; были баги парсинга для не-binary storage моделей XMLType (CLOB/object-relational), исправлены позже
Производительность/мониторинг	открытый вопрос	JMX-метрики (`OffsetScn` и др.); trade-off `redo_log_catalog` vs `online_catalog` описан выше

4. Трудности при переходе с GoldenGate на Debezium

1. **Нет доставки на target «из коробки».** Придётся отдельно разворачивать и эксплуатировать JDBC sink-коннектор — это отдельный компонент со своим жизненным циклом, ограничениями (например, CHAR-PK у Confluent JDBC Sink) и

мониторингом.

2. **XStream не избавляет от лицензии GoldenGate.** Если ради производительности выбрать адаптер `xstream` вместо LogMiner, лицензия на GoldenGate всё равно нужна — миграция «с GoldenGate на Debezium ради экономии на лицензии» работает только при использовании LogMiner-адаптера.
3. **Trade-off LogMiner-стратегий.** `redo_log_catalog` (надёжное отслеживание DDL) против `online_catalog` (быстрее, но без DDL) — нужно явно решить, что важнее для конкретной репликации.
4. **Долгие транзакции.** В буферизованном режиме (`logminer`) долгие транзакции накапливаются в памяти коннектора; `logminer_unbuffered` снимает эту нагрузку с Debezium, но перекладывает её на PGA источника.
5. **Multitenant-архитектура (CDB/PDB)** добавляет обязательный параметр `database.pdb.name` и усложняет настройку прав и supplemental logging — GoldenGate решает это иначе, и при миграции это нужно перепроверять отдельно.
6. **LOB/XMLType — источник специфичных багов.** Известны нетривиальные проблемы с представлением XMLType в зависимости от storage-модели (CLOB / object-relational / binary XML), а также сочетание XMLType с generated columns — стоит тестировать на репрезентативной схеме перед переносом продакшена.
7. **Механика рестарта по SCN и зависимость от архивных логов**
   Debezium хранит в offset **два SCN**: *low watermark* (`scn`) — безопасная точка возобновления, как правило указывающая на начало самой старой ещё не закоммиченной транзакции на момент остановки, и *high watermark* (`commit_scn`) — позиция последней эмитированной транзакции (Debezium for Oracle — Part 3). При старте коннектор сравнивает low watermark с самым старым доступным на источнике архивным логом. Если этот лог начинается **позже** сохранённого SCN, коннектор падает с ошибкой `None of the log files contains offset SCN` и требует полного ре-снапшота (troubleshooting-разбор Debezium, июль 2025).
   Отсюда прямо следует наблюдаемый на практике эффект: если на момент остановки капчуринга существовала долгая незакоммиченная транзакция (или коннектор просто долго простаивал), low watermark может оказаться далеко в прошлом — и тогда для успешного рестарта на источнике нужны архивные логи, покрывающие этот старый SCN, а не только логи «с последней обработанной точки». Политика retention архивных логов на источнике должна учитывать этот сценарий — иначе логи будут вычищены раньше, чем понадобятся, и единственным выходом останется полный ре-снапшот данных.
   В Oracle RAC добавляется требование иметь логи по **всем redo-thread'ам** для восстановления глобального порядка транзакций; с версии Debezium 2.7 есть проверка консистентности логов (Redo Thread Consistency check), детектирующая разрывы между threads (единственный источник на это — блог Debezium, независимое подтверждение не найдено, поэтому уровень доверия — средний).
   Отдельно задокументирован конкретный **регрессионный баг в версиях 3.4.0/3.4.1**: коннектор мог «залипать» на SCN, даже если нужный лог физически присутствовал на диске в архиве — подтверждено мейнтейнером Debezium (Chris Cranford) в официальной ветке форума, исправлено в **3.4.3.Final** (groups.google.com/g/debezium). Это отдельный повод при тестировании поведения рестарта фиксировать версию коннектора и не путать баг конкретного релиза с

общей архитектурной особенностью.