Red Hat build of Debezium 2.5.4

在 RHEL 上安装 Debezium

用于 Red Hat build of Debezium 2.5.4 on Red Hat Enterprise Linux (RHEL)

Last Updated: 2024-04-19

Red Hat build of Debezium 2.5.4 在 RHEL 上安装 Debezium

用于 Red Hat build of Debezium 2.5.4 on Red Hat Enterprise Linux (RHEL)

法律通告

摘要

本指南论述了如何在带有 AMQ Streams 的 RHEL 上安装红帽构建的 Debezium。

目录

前言

使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。有关更多详情，请参阅我们的首席技术官 Chris Wright 提供的消息。

第 1 章 DEBEZIUM 概述

Red Hat Integration 的 Debezium 是一个用来捕获数据库操作的分布式平台，为行级操作创建数据更改事件记录，并将事件记录流传输到 Apache Kafka 主题。Debezium 基于 Apache Kafka 构建，已部署并与 AMQ Streams 集成。

Debezium 捕获数据库表的行级更改，并将对应的更改事件传递给 AMQ Streams。应用程序可以读取 *这些更改事件流*，并按发生更改事件的顺序访问更改事件。

Debezium 是 Debezium 用于 Red Hat Integration 的上游社区项目。

Debezium 具有多个用途，包括：

- 数据复制
- 更新缓存和搜索索引
- 简化单体式应用程序
- 数据集成
- 启用流查询

Debezium 为以下通用数据库提供 Apache Kafka 连接连接器：

- Db2
- MySQL
- MongoDB
- Oracle
- PostgreSQL
- SQL Server

第 2 章 在 RHEL 上安装 DEBEZIUM 连接器

通过使用连接器插件扩展 Kafka Connect，通过 AMQ Streams 安装 Debezium 连接器。部署 AMQ Streams 后，您可以通过 Kafka Connect 将 Debezium 部署为连接器配置。

2.1. KAFKA 主题创建建议

Debezium 将数据存储在多个 Apache Kafka 主题中。主题必须由管理员提前创建，或者您可以配置 Kafka Connect 以 自动配置主题。

以下列表描述了在创建主题时要考虑的限制和建议：

Debezium Db2、MySQL、Oracle 和 SQL Server 连接器的数据库架构历史记录主题

对于前面的每个连接器，都需要一个数据库架构历史记录主题。无论您手动创建数据库 schema 历史记录主题，请使用 Kafka 代理自动创建主题，或使用 Kafka Connect 创建主题，请确保主题配置了以下设置：

- 无限或非常长的保留。
- 在生产环境中至少为 3 个复制因素。
- 单个分区。

其他主题

- 当您启用 Kafka 日志压缩 以便只保存给定记录 *的最后* 更改事件时，在 Apache Kafka 中设置以下主题属性：
  - **min.compaction.lag.ms**
  - **delete.retention.ms**
    为确保主题消费者有足够的时间接收所有事件和删除标记，请指定超过接收器连接器期望的最大停机时间的值。例如，请考虑将更新应用到接收器连接器时可能会出现的停机时间。
- 在生产环境中复制。
- 单个分区。
  您可以放宽单个磁盘分区，但应用程序必须为数据库中的不同行处理超出顺序的事件。一行的事件仍然被完全排序。如果您使用多个分区，则默认行为是 Kafka 通过哈希密钥来确定分区。其他分区策略需要使用单一消息转换(SMT)来为每个记录设置分区号。

2.2. 规划 DEBEZIUM 连接器配置

在部署 Debezium 连接器前，请确定如何配置连接器。配置提供了指定连接器行为的信息，并允许 Debezium 连接到源数据库。

您可以将连接器配置指定为 JSON，当您准备好注册连接器时，您可以使用 **curl** 将配置提交到 Kafka Connect API 端点。

先决条件

- 部署源数据库，Debezium 连接器可以访问数据库。

- 您知道以下信息，连接器需要访问源数据库：
  - 数据库主机的名称或 IP 地址。
  - 用于连接到数据库的端口号。
  - 连接器可用于登录到数据库的帐户名称。
  - 数据库用户帐户的密码。
  - 数据库的名称。
  - 您希望连接器捕获信息的表名称。
  - 您希望连接器发出更改事件的 Kafka 代理的名称。
  - 您希望连接器将数据库历史记录信息发送到的 Kafka 主题的名称。

流程

- 以 JSON 格式指定要应用到 Debezium 连接器的配置。
  以下示例显示了 Debezium MySQL 连接器的简单配置：

```
{
 "name": "inventory-connector", 1
 "config": {
  "connector.class": "io.debezium.connector.mysql.MySqlConnector", 2
  "tasks.max": "1", 3
  "database.hostname": "mysql", 4
  "database.port": "3306", 5
  "database.user": "debezium", 6
  "database.password": "dbz", 7
  "database.server.id": "184054", 8
  "topic.prefix": "dbserver1", 9
  "table.include.list": "public.inventory", 10
  "schema.history.internal.kafka.bootstrap.servers": "kafka:9092", 11
  "schema.history.internal.kafka.topic": "dbhistory.inventory" 12
 }
}
```

1. 使用 Kafka Connect 集群注册的连接器名称。
2. 连接器类的名称。
3. 可以同时操作的任务数量。一次仅应运行一个任务。
4. 主机数据库实例的主机名或 IP 地址。
5. 数据库实例的端口号。
6. Debezium 通过它连接到数据库的用户帐户名称。
7. 数据库用户帐户的密码。
8. 连接器的唯一数字 ID。
   此属性仅用于 MySQL 连接器。

此属性仅用于 MySQL 连接器。

9 用作数据库服务器的逻辑标识符或连接器捕获更改的服务器集群的字符串。指定的字符串指定一个命名空间。当使用 Avro converter 时，Debezium 将此名称添加到连接器写入的每个 Kafka 主题以及 Kafka Connect 模式的名称，以及相应的 Avro 模式的命名空间。

10 连接器捕获更改事件的表列表。

11 连接器发送数据库 schema 历史记录的 Kafka 代理的名称。指定代理还会接收连接器发出的更改事件。

12 存储 schema 历史记录的 Kafka 主题的名称。
在连接器重启后，连接器会从停止的时间点恢复读取数据库日志，为在它离线时发生的任何事务发出事件。在连接器将更改事件写入 Kafka 之前，它会检查 schema 历史记录，然后应用原始事务发生时生效的 schema。

附加信息

- 有关您可以为每种连接器设置的配置属性的信息，请参阅 Debezium 用户指南中的 连接器的部署文档。

后续步骤

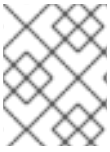
第 2.3 节 “在 Red Hat Enterprise Linux 中使用 AMQ Streams 部署 Debezium”。

2.3. 在 RED HAT ENTERPRISE LINUX 中使用 AMQ STREAMS 部署 DEBEZIUM

这个步骤描述了如何在 Red Hat Enterprise Linux 中为 Debezium 设置连接器。连接器使用 Apache Kafka Connect 部署到 AMQ Streams 集群，这是在 Apache Kafka 和外部系统间流传输数据的框架。Kafka Connect 必须以分布式模式运行，而不是独立模式。

先决条件

- 用于您需要部署 Debezium 的主机环境运行 Red Hat Enterprise Linux、AMQ Streams 和 Java（以一个支持的配置）。
  - 有关如何安装 AMQ Streams 的详情，请参考 安装 AMQ Streams。
  - 有关如何安装包含单个 ZooKeeper 节点的基本非生产环境 AMQ Streams 集群以及单个 Kafka 节点的详情，请参考 运行单一节点 AMQ Streams 集群。



注意

如果您正在运行早期版本的 AMQ Streams，您必须首先升级到 AMQ Streams 2.6。有关升级过程的详情，请参考 AMQ Streams 和 Kafka 升级。

- 主机上具有管理权限(**sudo** 访问权限)。
- Apache ZooKeeper 和 Apache Kafka 代理正在运行。
- Kafka Connect 以分布式模式运行，而不是在独立模式中运行。
- 您知道安装 AMQ Streams 时创建的 **kafka** 用户的凭证。

- 部署源数据库，部署 Debezium 的主机可以访问数据库。
- 您知道如何配置连接器。

步骤

1. 从 Red Hat Integration 下载站点下载 Debezium 连接器或连接器。例如，要将 Debezium 与 MySQL 数据库搭配使用，请下载 **Debezium 2.5.4 MySQL Connector**。
2. 在部署 AMQ Streams 的 Red Hat Enterprise Linux 主机上，打开终端窗口并在 **/opt/kafka** 中创建 **connector-plugins** 目录（如果它尚不存在）：

```
$ sudo mkdir /opt/kafka/connector-plugins
```

3. 输入以下命令将您下载到 **/opt/kafka/connector-plugins** 目录的 Debezium 连接器存档的内容。

```
$ sudo unzip debezium-connector-mysql-2.5.4.Final.zip -d /opt/kafka/connector-plugins
```

4. 对您要安装的每个连接器重复步骤 1-3。
5. 在一个终端窗口中，以 **kafka** 用户身份进行登录：

```
$ su - kafka
$ Password:
```

6. 如果正在运行，停止 Kafka Connect 进程。
   a. 输入以下命令检查 Kafka Connect 是否在分布式模式下运行：

```
$ jcmd | grep ConnectDistributed
```

   如果进程正在运行，命令会返回进程 ID，例如：

```
18514 org.apache.kafka.connect.cli.ConnectDistributed /opt/kafka/config/connect-distributed.properties
```

   b. 使用进程 ID 输入 **kill** 命令来停止该进程，例如：

```
$ kill 18514
```

7. 编辑 **/opt/kafka/config/** 中的 **connect-distributed.properties** 文件，并将 **plugin.path** 的值设置为 Debezium 连接器插件的父目录的位置：

```
plugin.path=/opt/kafka/connector-plugins
```

8. 以分布式模式启动 Kafka 连接。

```
$ /opt/kafka/bin/connect-distributed.sh /opt/kafka/config/connect-distributed.properties
```

9. 在 Kafka Connect 运行后，使用 Kafka Connect API 来注册连接器。
   输入 **curl** 命令，以提交 **POST** 请求，将您在 第 2.2 节 “规划 Debezium 连接器配置” 中指定的连接器配置 JSON 发送到 **localhost:8083/connectors** 的 Kafka Connect REST API 端点。
   例如：

```
curl -i -X POST -H "Accept:application/json" -H "Content-Type:application/json"
http://localhost:8083/connectors/ -d '{"name": "inventory-connector", "config":
{"connector.class": "io.debezium.connector.mysql.MySqlConnector",
"tasks.max": "1",
"database.hostname": "mysql",
"database.port": "3306",
"database.user": "debezium",
"database.password": "dbz",
"database.server.id": "184054",
"topic.prefix": "dbserver1",
"table.include.list": "public.inventory",
"schema.history.internal.kafka.bootstrap.servers": "kafka:9092",
"schema.history.internal.kafka.topic": "dbhistory.inventory" }
}'
```

要注册多个连接器，请为每个连接器提交一个单独的请求。

10. 重启 Kafka Connect 来实现您的更改。
当 Kafka Connect 启动时，它会从 **connector-plugins** 目录中加载配置的 Debezium 连接器。

    完成配置后，部署的连接器连接到源数据库，并为每个插入、更新或删除的行或文档生成事件。

11. 为每个 Kafka Connect worker 节点重复步骤 5-10。

后续步骤

验证部署。

其他资源

- 添加连接器插件

2.4. 验证部署

连接器启动后，它会执行配置的数据库的快照，并为您指定的每个表创建主题。

先决条件

- 根据 第 2.3 节 “在 Red Hat Enterprise Linux 中使用 AMQ Streams 部署 Debezium” 中的说明，您在 Red Hat Enterprise Linux 上部署了连接器。

1. 在主机上的终端窗口中输入以下命令请求 Kafka Connect API 中的连接器列表：

   ```
   $ curl -H "Accept:application/json" localhost:8083/connectors/
   ```

   查询返回部署的连接器的名称，例如：

   ```
   ["inventory-connector"]
   ```

2. 在主机上的终端窗口中输入以下命令查看连接器正在运行的任务：

   ```
   $ curl -i -X GET -H "Accept:application/json" localhost:8083/connectors/inventory-
   connector
   ```

该命令返回类似以下示例的输出：

```
HTTP/1.1 200 OK
Date: Thu, 06 Feb 2020 22:12:03 GMT
Content-Type: application/json
Content-Length: 531
Server: Jetty(9.4.20.v20190813)

{
  "name": "inventory-connector",
  ...
  "tasks": [
    {
      "connector": "inventory-connector",
      "task": 0
    }
  ]
}
```

3. 显示 Kafka 集群中的主题列表。
在终端窗口中进入 **/opt/kafka/bin/** 并运行以下 shell 脚本：

```
./kafka-topics.sh --bootstrap-server=localhost:9092 --list
```

Kafka 代理返回连接器创建的主题列表。可用的主题取决于连接器的 **snapshot.mode**、**snapshot.include.collection.list** 和 **table.include.list** 配置属性的设置。默认情况下，连接器会为数据库中的每个非系统表创建一个主题。

4. 查看主题的内容。
在终端窗口中，进入到 **/opt/kafka/bin/**，并运行 **kafka-console-consumer.sh** shell 脚本，以显示上一命令返回的其中一个主题的内容：
例如：

```
./kafka-console-consumer.sh \
>     --bootstrap-server localhost:9092 \
>     --from-beginning \
>     --property print.key=true \
>     --topic=dbserver1.inventory.products_on_hand
```

对于主题中的每个事件，命令会返回类似以下示例的信息：

例 2.1. Debezium 更改事件的内容

```
{"schema":{"type":"struct","fields":
[{"type":"int32","optional":false,"field":"product_id"}],"optional":false,"name":"dbserver1.in
ventory.products_on_hand.Key"},"payload":{"product_id":101}} {"schema":
{"type":"struct","fields":[{"type":"struct","fields":
[{"type":"int32","optional":false,"field":"product_id"},
{"type":"int32","optional":false,"field":"quantity"}],"optional":true,"name":"dbserver1.inven
tory.products_on_hand.Value","field":"before"},{"type":"struct","fields":
[{"type":"int32","optional":false,"field":"product_id"},
{"type":"int32","optional":false,"field":"quantity"}],"optional":true,"name":"dbserver1.inven
tory.products_on_hand.Value","field":"after"},{"type":"struct","fields":
[{"type":"string","optional":false,"field":"version"},
```

{"type":"string","optional":false,"field":"connector"},
{"type":"string","optional":false,"field":"name"},
{"type":"int64","optional":false,"field":"ts_ms"},
{"type":"string","optional":true,"name":"io.debezium.data.Enum","version":1,"parameters":{"allowed":"true,last,false"},"default":"false","field":"snapshot"},
{"type":"string","optional":false,"field":"db"},
{"type":"string","optional":true,"field":"sequence"},
{"type":"string","optional":true,"field":"table"},
{"type":"int64","optional":false,"field":"server_id"},
{"type":"string","optional":true,"field":"gtid"},{"type":"string","optional":false,"field":"file"},
{"type":"int64","optional":false,"field":"pos"},
{"type":"int32","optional":false,"field":"row"},
{"type":"int64","optional":true,"field":"thread"},
{"type":"string","optional":true,"field":"query"}],"optional":false,"name":"io.debezium.connector.mysql.Source","field":"source"},{"type":"string","optional":false,"field":"op"},
{"type":"int64","optional":true,"field":"ts_ms"},{"type":"struct","fields":
[{"type":"string","optional":false,"field":"id"},
{"type":"int64","optional":false,"field":"total_order"},
{"type":"int64","optional":false,"field":"data_collection_order"}],"optional":true,"field":"transaction"}],"optional":false,"name":"dbserver1.inventory.products_on_hand.Envelope"},
"payload":{"before":null,"after":{"product_id":101,"quantity":3},"source":
{"version":"2.5.4.Final-redhat-
00001","connector":"mysql","name":"inventory_connector_mysql","ts_ms":1638985247805,"snapshot":"true","db":"inventory","sequence":null,"table":"products_on_hand","server_id":0,"gtid":null,"file":"mysql-
bin.000003","pos":156,"row":0,"thread":null,"query":null},"op":"r","ts_ms":1638985247805,"transaction":null}}

在前面的示例中，**有效负载** 值显示连接器快照从表 **inventory.products_on_hand** 生成读取 (**op" ="r"**)事件。**product_id** 记录的 **"before"** 状态为 **null**，表示该记录不存在之前的值。**"after"** 状态对于 **product_id** 为 **101** 的项目的 **quantity** 显示为 **3**。

后续步骤

有关每个连接器可用的配置设置的信息，并了解如何配置源数据库以启用更改数据捕获，请参阅 Debezium 用户指南。

2.5. 更新 KAFKA CONNECT 集群中的 DEBEZIUM 连接器插件

要替换在 Red Hat Enterprise Linux 上部署的 Debezium 连接器版本，您需要更新连接器插件。

步骤

1. 下载您要从 Red Hat Integration 下载站点 替换的 Debezium 连接器插件的副本。
2. 将 Debezium 连接器存档的内容提取到 **/opt/kafka/connector-plugins** 目录中。

```
$ sudo unzip debezium-connector-mysql-2.5.4.Final.zip -d /opt/kafka/connector-plugins
```

3. 重启 Kafka Connect。

附录 A. 使用您的订阅

Debezium 通过软件订阅提供。要管理您的订阅，请访问红帽客户门户中的帐户。

访问您的帐户

1. 转至 access.redhat.com。
2. 如果您还没有帐户，请创建一个帐户。
3. 登录到您的帐户。

激活订阅

1. 转至 access.redhat.com。
2. 导航到 **Subscriptions**。
3. 导航到 **激活订阅** 并输入您的 16 位激活号。

下载 zip 和 tar 文件

要访问 zip 或 tar 文件，请使用客户门户网站查找下载的相关文件。如果您使用 RPM 软件包，则不需要这一步。

1. 打开浏览器并登录红帽客户门户网站 **产品下载页面**，网址为 access.redhat.com/downloads。
2. 向下滚动到 INTEGRATION **和 AUTOMATION**。
3. 点 **Red Hat Integration** 以显示 Red Hat Integration 下载页面。
4. 单击组件的 **Download** 链接。

更新于 2024-04-19