



Red Hat Integration 2022.Q2

Installing Debezium on RHEL

For use with Debezium 1.7 on Red Hat Enterprise Linux (RHEL)

Red Hat Integration 2022.Q2 Installing Debezium on RHEL

For use with Debezium 1.7 on Red Hat Enterprise Linux (RHEL)

Legal Notice

Copyright © 2022 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This guide describes how to install Red Hat Debezium on RHEL with AMQ Streams.

Table of Contents

PREFACE	3
MAKING OPEN SOURCE MORE INCLUSIVE	3
CHAPTER 1. DEBEZIUM OVERVIEW	4
CHAPTER 2. INSTALLING DEBEZIUM CONNECTORS ON RHEL	5
2.1. PREREQUISITES	5
2.2. KAFKA TOPIC CREATION RECOMMENDATIONS	5
2.3. DEPLOYING DEBEZIUM WITH AMQ STREAMS ON RHEL	6
APPENDIX A. USING YOUR SUBSCRIPTION	8
Accessing your account	8
Activating a subscription	8
Downloading zip and tar files	8

PREFACE

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see [our CTO Chris Wright's message](#).

CHAPTER 1. DEBEZIUM OVERVIEW

Debezium for Red Hat Integration is a distributed platform that captures database operations, creates data change event records for row-level operations, and streams change event records to Apache Kafka topics. Debezium is built on Apache Kafka and is deployed and integrated with AMQ Streams.

Debezium captures row-level changes to a database table and passes corresponding change events to AMQ Streams. Applications can read these *change event streams* and access the change events in the order in which they occurred.

[Debezium](#) is the upstream community project for Debezium for Red Hat Integration.

Debezium has multiple uses, including:

- Data replication
- Updating caches and search indexes
- Simplifying monolithic applications
- Data integration
- Enabling streaming queries

Debezium provides connectors (based on Kafka Connect) for the following common databases:

- [Db2](#)
- [MySQL](#)
- [MongoDB](#)
- [Oracle](#) (Technology Preview)
- [PostgreSQL](#)
- [SQL Server](#)

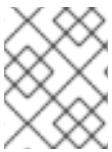
CHAPTER 2. INSTALLING DEBEZIUM CONNECTORS ON RHEL

Install Debezium connectors through AMQ Streams by extending Kafka Connect with connector plugins. Following a deployment of AMQ Streams, you can deploy Debezium as a connector configuration through Kafka Connect.

2.1. PREREQUISITES

A Debezium installation requires the following:

- Red Hat Enterprise Linux is running.
- Administrative privileges (**sudo** access).
- AMQ Streams 2.0 on Red Hat Enterprise Linux is installed on the host computer.
- Credentials for the **kafka** user that was created when AMQ Streams was installed.
- An AMQ Streams cluster is running.
 - For instructions on running a basic, non-production AMQ Streams cluster that contains a single ZooKeeper node, and a single Kafka node, see [Running a single node AMQ Streams cluster](#).



NOTE

If you have an earlier version of AMQ Streams, you must first upgrade to AMQ Streams 2.0. For upgrade instructions, see [AMQ Streams and Kafka upgrades](#).

Additional resources

- For information about the supported configuration for running Debezium on Red Hat Enterprise Linux, see the [Debezium Supported Configurations page](#).
- For more information about how to install AMQ Streams, see [Installing AMQ Streams](#).

2.2. KAFKA TOPIC CREATION RECOMMENDATIONS

Debezium stores data in multiple Apache Kafka topics. The topics must either be created in advance by an administrator, or you can configure Kafka Connect to [configure topics automatically](#).

The following list describes limitations and recommendations to consider when creating topics:

Database history topics for MySQL, SQL Server, Db2, and Oracle connectors

- Infinite or very long retention.
- Replication factor of at least three in production environments.
- Single partition.

Other topics

- When you enable [Kafka log compaction](#) so that only the *last* change event for a given record is saved, set the following topic properties in Apache Kafka:

- [min.compaction.lag.ms](#)

- [delete.retention.ms](#)

To ensure that topic consumers have enough time to receive all events and delete markers, specify values for the preceding properties that are larger than the maximum downtime that you expect for your sink connectors. For example, consider the downtime that might occur when you apply updates to sink connectors.

- Replicated in production.

- Single partition.

You can relax the single partition rule, but your application must handle out-of-order events for different rows in the database. Events for a single row are still totally ordered. If you use multiple partitions, the default behavior is that Kafka determines the partition by hashing the key. Other partition strategies require the use of single message transformations (SMTs) to set the partition number for each record.

2.3. DEPLOYING DEBEZIUM WITH AMQ STREAMS ON RHEL

This procedure describes how to set up connectors for Debezium on Red Hat Enterprise Linux. Connectors are deployed to an AMQ Streams cluster using Apache Kafka Connect, a framework for streaming data between Apache Kafka and external systems. Kafka Connect must be run in distributed mode rather than standalone mode.

This procedure assumes that AMQ Streams is installed and ZooKeeper and Apache Kafka are running.

Procedure

1. Visit the [Red Hat Integration download site](#) on the Red Hat Customer Portal and download the Debezium connector or connectors that you want to use. For example, download the **Debezium 1.7 MySQL Connector** to use Debezium with a MySQL database.
2. In `/opt/kafka`, create the **connector-plugins** directory if not already created for other Kafka Connect plugins:

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

3. Extract the contents of the Debezium connector archive to the `/opt/kafka/connector-plugins` directory.

This example extracts the contents of the MySQL connector:

```
$ sudo unzip debezium-connector-mysql-1.7.2.Final.zip -d /opt/kafka/connector-plugins
```

4. Repeat the preceding steps for each connector that you want to install.
5. Switch to the **kafka** user:

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

6. Stop the Kafka Connect process if it is running.

- a. Check whether Kafka Connect is running in distributed mode by entering the following command:

```
$ jcmd | grep ConnectDistributed
```

If the process is running, the command returns the process ID, for example:

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

- b. Stop the process by entering the **kill** command with the process ID, for example,

```
$ kill 18514
```

7. Edit the **connect-distributed.properties** file in **/opt/kafka/config/** and specify the location of the Debezium connector:

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

8. Start Kafka Connect in distributed mode:

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

Kafka Connect runs. During startup, Debezium connectors are loaded from the **connector-plugins** directory.

9. Repeat steps 6–8 for each Kafka Connect worker node.

Additional resources

- [Kafka Connect in distributed mode](#)
- [Adding connector plugins](#)

Updating Kafka Connect

If you need to update your deployment, amend the Debezium connector JAR files in the **/opt/kafka/connector-plugins** directory, and then restart Kafka Connect.

Next Steps

The *Debezium User Guide* describes how to configure each connector and its source database for change data capture. After you complete the configuration, a connector will connect to the source database and produce events for each inserted, updated, and deleted row or document.

APPENDIX A. USING YOUR SUBSCRIPTION

Integration is provided through a software subscription. To manage your subscriptions, access your account at the Red Hat Customer Portal.

Accessing your account

1. Go to access.redhat.com.
2. If you do not already have an account, create one.
3. Log in to your account.

Activating a subscription

1. Go to access.redhat.com.
2. Navigate to **My Subscriptions**.
3. Navigate to **Activate a subscription** and enter your 16-digit activation number.

Downloading zip and tar files

To access zip or tar files, use the customer portal to find the relevant files for download. If you are using RPM packages, this step is not required.

1. Open a browser and log in to the Red Hat Customer Portal **Product Downloads** page at access.redhat.com/downloads.
2. Scroll down to **INTEGRATION AND AUTOMATION**.
3. Click **Red Hat Integration** to display the Red Hat Integration downloads page.
4. Click the **Download** link for your component.

Revised on 2022-05-23 09:57:55 UTC