Cloudera Runtime 7.3.3

# Troubleshooting Apache Impala

**Date published: 2020-11-30**
**Date modified: 2026-11-13**



**https://docs.cloudera.com/**

# Legal Notice

© Cloudera Inc. 2026. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

# Contents

**Troubleshooting common issues in Impala** ..... 4
**Using Breakpad Minidumps for Crash Reporting** ..... 4
**Dynamic memory for Impala control service** ..... 5
**Impala sender timed out waiting error** ..... 6

# Troubleshooting common issues in Impala

This topic describes the general troubleshooting procedures to diagnose some of the commonly encountered issues in Impala.

| Symptom | Explanation | Recommendation |
|---|---|---|
| Impala takes a long time to start. | Impala instances with large numbers of tables, partitions, or data files take longer to start because the metadata for these objects is broadcast to all `impalad` nodes and cached. | Adjust timeout and synchronicity settings. |
| Query rejected with the default pool-defined memory limit settings. | Some complex queries fail because the minimum memory reservation per host is greater than the memory available to the query for buffer reservations. | Increase VW t-shirt size so that there are more hosts in the executor group and less memory is needed per host. |
| Joins fail to complete. | There may be insufficient memory. During a join, data from the second, third, and so on sets to be joined is loaded into memory. If Impala chooses an inefficient join order or join mechanism, the query could exceed the total memory available. | Start by gathering statistics with the COMPUTE STATS statement for each table involved in the join.<br>Consider specifying the [SHUFFLE] hint so that data from the joined tables is split up between nodes rather than broadcast to each node.<br>If tuning at the SQL level is not sufficient, add more memory to your system or join smaller data sets. |
| Queries return incorrect results. | Impala metadata may be outdated after changes are performed in Hive. | After inserting data, adding a partition, or other operation in Hive, refresh the metadata for the table with the REFRESH statement. |
| Attempts to complete Impala tasks such as executing INSERT SELECT statements fail. The Impala logs include notes that files could not be opened due to permission denied. | This can be the result of permissions issues. For example, you could use the Hive shell as the hive user to create a table. After creating this table, you could attempt to complete some action, such as an INSERT SELECT on the table. Because the table was created using one user and the INSERT SELECT is attempted by another, this action may fail due to permissions issues. | Ensure the Impala user has sufficient permissions to the table that the Hive user created. |
| Impala fails to start up, with the `impalad` logs referring to errors connecting to the statestored service and attempts to re-register. | A large number of databases, tables, partitions, and so on can require metadata synchronization, particularly on startup, that takes longer than the default timeout for the statestore service. | Configure the statestore timeout value and possibly other settings related to the frequency of statestore updates and metadata loading. |

# Using Breakpad Minidumps for Crash Reporting

The breakpad project is an open-source framework for crash reporting. Impala can use breakpad to record stack information and register values when any of the Impala-related daemons crash due to an error such as SIGSEGV or unhandled exceptions. The dump files are much smaller than traditional core dump files. The dump mechanism itself uses very little memory, which improves reliability if the crash occurs while the system is low on memory.

### Using the Minidump Files for Problem Resolution

You can see in the Impala log files or in the Cloudera Manager charts for Impala when crash events occur that generate minidump files. Because each restart begins a new log file, the “crashed” message is always at or near the bottom of the log file. (There might be another later message if core dumps are also enabled.)

**Important:** If an Impala-related daemon experiences a crash due to an out-of-memory condition, it does not generate a minidump for that error.

Typically, you provide minidump files to Cloudera Support as part of problem resolution, in the same way that you might provide a core dump. The Send Diagnostic Data under the Support menu in Cloudera Manager guides you through the process of selecting a time period and volume of diagnostic data, then collects the data from all hosts and transmits the relevant information for you.

### Procedure

1. In Cloudera Manager, navigate to Impala service Configuration .
2. In the search field, type minidump.
3. Set the following fields to configure breakpad minidumps.
   - minidump_path: Turn on or off generation of the minidump files.

     By default, a minidump file is generated when an Impala-related daemon crashes.
   - minidump_path: Specify the Location for minidump files.

     By default, all minidump files are written to the following location on the host where a crash occurs: /var/log/impala-minidumps/*DAEMON_NAME*

     The minidump files for `impalad`, `catalogd`, and `statestored` are each written to a separate directory.

     If you specify a relative path for this setting, the value is interpreted relative to the default minidump_path directory.
   - max_minidumps: Specify the number of minidump files.

     Like any files used for logging or troubleshooting, consider limiting the number of minidump files, or removing unneeded ones, depending on the amount of free storage space on the hosts in the cluster.

     Because the minidump files are only used for problem resolution, you can remove any such files that are not needed to debug current issues.

     The default for this setting is 9. A zero or negative value is interpreted as “unlimited”.
4. Click Save Changes and restart Impala.
5. To provide minidump files to Cloudera Support as part of problem resolution, in Cloudera Manager, navigate to Support Send Diagnostic Data and follow the steps.

# Dynamic memory for Impala control service

Understand and configure memory allocation for Impala's control service queue to avoid errors and ensure consistent behavior across daemons with varying memory limits.

### Condition

Impala daemon logs report errors when the control_service_queue_mem_limit is set to a smaller value:

```
impala-service-pool.cc: EndDataStream request on impala.DataStreamService fr
om xx.xx.xx.xx:xxxx dropped due to backpressure.
The service queue contains xx items out of a maximum of xxxxxxxxxx; memory
consumption is 50.05 MB.
```

In this message, the "maximum" value (e.g., xxxxxxxxxx or INT_MAX) indicates that the ControlService's RPC queue does not have an explicit limit based on the number of items. This queue's effective capacity is determined solely by its memory consumption, as controlled by the --control_service_queue_mem_limit parameter.

**Tip:** For "dropped due to backpressure" messages, the critical metric is memory consumption, which reflects the actual queue limit. The "maximum" item count, which is INT_MAX (2^31-1), is not the governing limit.

### Cause

Before the implementation of IMPALA-10343, the default value of control_service_queue_mem_limit was 50MB. After IMPALA-10343 (starting from versions 7.1.7 SP2, 7.1.8, and 7.1.9), the value became dynamic and is now set to 1% of mem_limit.

The control service queue memory limit became dynamic, adjusting based on the daemon's memory. To avoid unintentional reductions in memory allocation, use the --control_service_queue_mem_limit_floor_bytes flag. This ensures consistent behavior for daemons with smaller memory limits, while increasing the limit for those with larger memory.

### Remedy

### Procedure

1. Log in to **Cloudera Manager** as an administrator.
2. Navigate to Clusters Impala Configuration and search for Impala Daemon Command Line Argument Advanced Configuration Snippet (Safety Valve).
3. Add the below value:

```
--control_service_queue_mem_limit=XXXMB
--control_service_queue_mem_limit_floor_bytes=XXMB (default is 50MB)
```

4. The memory limit can be adjusted using the previously mentioned configurations to accommodate higher workloads. Ensure that you restart the Impala service for the changes to take effect.

# Impala sender timed out waiting error

Resolve Impala query failures during high system load by increasing the data stream sender timeout startup flag.

### Condition

Impala jobs fail under high system load with the following error:

```
Sender <IP Address> timed out waiting for receiver fragment
```

This happens because the Impala coordinator takes longer than the default 120-second timeout to dispatch query fragments to the executor nodes. The sender times out while waiting for a response from the receiver. You can adjust the timeout setting to mitigate this delay and prevent query failures.

Impala jobs stop and display the following error message, which indicates a timeout while the sender waits for the receiver fragment instance:

```
Query Status: Sender <IP Address>:53006 timed out waiting for receiver
 fragment instance: ed44b43eb5b13652:9bccaee300000041, dest node: 3
```

The ImpalaD logs also displays a related timeout error:

```
I0719 04:48:46.672668 59007 rpcz_store.cc:269] Call
 impala.DataStreamService.TransmitData from <IP Address>:53006 (request call
 id 948988) took 120721ms. Trace:
```

### Cause

When an Impala query starts, the coordinator dispatches query fragments to the Impala daemons that execute the query. The senders and receivers for the data exchange must connect during fragment startup. Under heavy load, the coordinator's dispatch of fragments can be delayed. The factors that contribute to high system load or delays include issues such as Java Virtual Machine (JVM) pauses, Key Distribution Center (KDC) slowness (when Kerberos (KRB5) authentication is used), system resource limitations, or network latency or disruptions.

If the time between the first batch being sent and the receiver initializing is longer than the --datastream_sender_timeout_ms threshold (default is 120,000 milliseconds or 120 seconds), the query fails with a sender timed out waiting for receiver fragment instance error.

### Remedy

### Procedure

1. Review the ImpalaD logs to confirm the timeout duration exceeds the default limit of 120,000 milliseconds (120 seconds).

   Look for log entries similar to the following:

   ```
   Sender <IP>:53006 timed out waiting for receiver fragment instance:
    <instance ID>, dest node: 3
   Call impala.DataStreamService.TransmitData from <masked IP>:53006 (request
    call id <call ID>) took 120721ms.
   ```

2. Identify factors that contribute to high system load or delays.

   This includes issues such as JVM pauses, KDC slowness (if KRB5 authentication is used), system resource limitations, or network latency or disruptions.

3. Modify the --datastream_sender_timeout_ms Impala startup flag to a higher value.

   For example, setting --datastream_sender_timeout_ms to 240000 (240 seconds) provides an extended period, which prevents sender-receiver timeouts.

4. Monitor the performance of Impala query execution after adjusting the timeout value to ensure the modification resolves the timeout error. Contact Cloudera Support or troubleshoot the cause of cluster delays if the timeout error message occurs again.