

Cloudera Runtime 7.3.2

## Troubleshooting Apache Atlas

Date published: 2020-07-28

Date modified: 2026-03-31

The Cloudera logo is displayed in a bold, orange, sans-serif font. The word "CLOUDERA" is written in all caps, with a stylized 'E' that has a horizontal bar extending to the right.

<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

<b>Atlas index repair configuration.....</b>	<b>4</b>
<b>Activating Concurrent Message Ingestion.....</b>	<b>5</b>
<b>Real-time update of entity details.....</b>	<b>5</b>

## Atlas index repair configuration

You can use reindexing to troubleshoot Apache Atlas basic search inconsistency.

### Rebuilding the whole Atlas index

In your Cloudera Manager instance running the Atlas service, add the following in Atlas Server Advanced Configuration Snippet (Safety Valve) for conf/atlas-application.properties.

```
atlas.rebuild.index=true
```

```
atlas.patch.numWorkers=3
```

```
atlas.patch.batchSize=300
```

Later, restart the Atlas Service.



#### Attention:

- You must revert back this configuration once the reindexing is completed, else the reindexing takes place on every restart.
- The reindexing process will be done during Atlas restart, so Atlas will not be reachable till reindexing process is completed.
- The time taken for reindexing depends upon the amount of data.

### Rebuilding the index for particular GUID

Incorrect search results related to a particular GUID can be repaired by limiting the reindex to that element.

```
atlas-index-repair/repair_index.py [-g <***GUID***>]
```



#### Note:

Atlas will use REST APIs to fetch the entity, which will need the correct authentication mechanism to be specified based on the installation.

For an Atlas installation with username and password use the following:

```
atlas-index-repair/repair_index.py [-g <***GUID***>] [-u <***USER***>] [-p <***PASSWORD***>] *  
guid: [optional]
```

Example:

```
atlas-index-repair/repair_index.py -u admin -p admin123 -g 13d77457-2a45-4e92-ad53-a172c7cb70a5
```

For Atlas installations using Kerberos as authentication mode, use the following:

```
kinit -kt /etc/security/keytabs/atlas.service.keytab atlas/fqdn@DOMAIN
```

Example:

```
kinit -kt /etc/security/keytabs/atlas.service.keytab atlas/fqdn@EXAMPLE.com
```

```
atlas-index-repair/repair_index.py -g 13d77457-2a45-4e92-ad53-a172c7cb70a5
```



**Note:** In case of many affected entities, it is recommended to rebuild the whole index instead.

## Activating Concurrent Message Ingestion

Atlas hook message processing can take a long time with a high number of messages, greatly increasing wait times. This can be improved by turning on Concurrent Message Ingestion.

Atlas Hook message processing has linear complexity for consuming the messages. This can lead to the following:

- Enforcing authorization policies takes longer.
- Metadata showing up in Atlas takes an unpredictable amount of time.

Concurrent Message Ingestion can offset these issues by enabling the following after determining dependencies within incoming messages:

- Dependent messages are processed serially.
- Messages without dependencies are processed concurrently.

### Enabling Concurrent Message Ingestion

1. Go to Cloudera Manager Clusters Atlas Configuration .
2. Add the following property `atlas.notifications.concurrent=true`.

## Real-time update of entity details

Atlas cannot update row numbers for entities unless a Data Definition Language operation is made on the entity.

Atlas can only monitor events caused by the DDL commands. This means that Atlas is not updated when Data Manipulation Language (DML) operations (SELECT/INSERT/UPDATE/DELETE) cause changes in Hive. Details not updated include also the operational properties, for example, **lastAccessTime**, **last\_modified\_by**, **last\_modified\_time**. The displayed details in Atlas (for example, number of rows) can be triggered by Data Definition Language (DDL) operations (CREATE/ALTER/DROP).

Figure 1: Hive table without any rows in Atlas

The screenshot displays the Apache Atlas web interface for a Hive table named 'raw\_bookings'. The interface is split into a left sidebar and a main content area. The sidebar contains search filters and navigation options. The main content area shows the table's details, including technical properties, user-defined properties, labels, and business metadata. The 'Technical properties' section is expanded, showing a 'numRows' property with a value of '0', which is highlighted with a red box. Other properties include 'columns', 'createTime', 'db', 'lastAccessTime', 'name', 'owner', and 'parameters'.

You can check the number of rows in Hue with the **Table Browser**.

The screenshot shows the Hue Table Browser interface. The breadcrumb path is "Databases > airline\_operations > raw\_bookings". The "STATS" section shows "Rows: 20" highlighted in a red box. Below, the "SCHEMA" section displays a table with 9 columns and sample data.

Column (9)	Type	Description	Sample
flight_id_raw	string		BUD-JFK-20250828
passenger_name_raw	string		John Smith
seat_assignment_raw	string		14A
aircraft_model_raw	string		Boeing 787-9
origin_airport_raw	string		BUD
destination_airport_raw	string		JFK
age_raw	int		45
ticket_price_usd_raw	decimal...		1250.75
passport_number_raw	string		583472910

You can also check the number of rows by using the COUNT query.

The screenshot shows the Hue Hive query editor. A query is entered: `SELECT COUNT(*) FROM airline_operations.raw_bookings;`. The results pane shows a single row with the value 20.

```

1 | SELECT COUNT(*) FROM airline_operations.raw_bookings;
-----
1 | 20
    
```



**Note:** Running an ALTER operation on the specific table, for example adding new column, triggers the recalculation of rows in Atlas' **parameters** field.