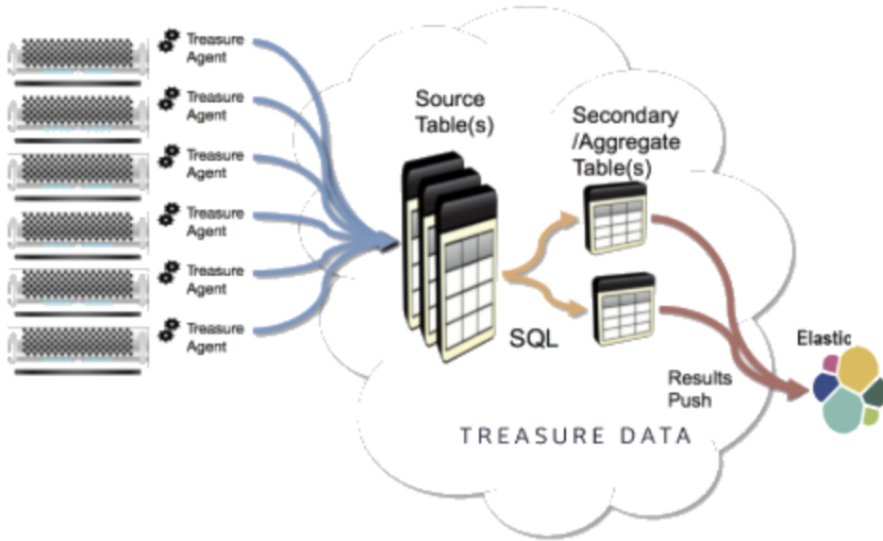


# Elastic Cloud Export Integration

Send job results directly to your Elastic Cloud instance using this connector.



This topic includes:

- [Supported](#)
- [Prerequisites](#)
- [Define the Data Export from Treasure Data](#)
- [Validate Your Export Data within the Elastic Cloud Instance](#)
- [Tune Timeout Exceptions](#)

## Supported

- This export supports “basic authentication” including “Security”(formally “Shield”) of Elastic Cloud.
- The query result doesn’t support LDAP and Active Directory that are provided by “Security”.
- Elastic Cloud result output supports TCP/9200 as a default. Elastic Cloud provides different a port for every user.
- This export supports “Security” for Elastic Cloud with Elastic Cloud Result Output.
- LDAP or other authentication methods are not supported.

## Prerequisites

- Basic knowledge of Treasure Data, including the TD [Toolbelt](#).
- Data imported into Treasure Data, that you want to export into Elastic Cloud.
- Working knowledge of SQL, Hive, or Presto.
- A working Elastic Cloud instance. Recommend version 2.0 or greater.
- You can use this with your own Elastic Cloud instance running in your environment.

Also, a knowledge of the following Elastic Cloud hierarchy is helpful:

Term	Description	Description of Value to Specify
Cluster	A collection of one or more servers (nodes) that collectively hold and provides search and indexing functionality for your entire dataset.	
Node	A single server that is part of (or all of) your cluster.	<ul style="list-style-type: none"> <li>• comma-separated list of nodes</li> </ul>

Index	This is analogous to a database. An index is a collection of documents with somewhat similar characteristics.	<ul style="list-style-type: none"> <li>the name of the index</li> </ul>
Type	This is analogous to a table. One or more types is defined within an index. A type is a logical category or partition of your index.	<ul style="list-style-type: none"> <li>the name of the type</li> </ul>
ID	A column containing each name for each row/record. In Elastic Cloud result export, this setting is optional.	<ul style="list-style-type: none"> <li>(optional) the name of the ID column</li> </ul>

For more information, go to the [Elastic Cloud documentation](#).

## Define the Data Export from Treasure Data

1. Complete the instructions in [Creating a Destination Integration](#).
2. Navigate to **Data Workbench > Queries**.
3. Select a query for which you would like to export data.
4. Run the query to validate the result set.
5. Select **Export Results To**.
6. Use the selection dialog to select your destination connection. For example:

### Choose Integration ✕

Use Existing Integration

Search...

00_2977_box_connection_1	box
00_297_box_connection_2	box
00_mailpublisher_shirai	mail_publisher_smart

7. Define any additional Export Results details.  
For example:

### Export Results ✕

Mode:

Index:   
Like a 'database' in a relational database. In replace mode, this name will be used as the alias name. Index must not already exist

Index type:   
Like a 'table' in a relational database

ID:   
Column name in your schema that you want to set as document\_id

Batch size (actions):   
Flush when this many actions are buffered

Batch size (bytes):   
Flush when this many bytes are buffered

Export null when column is empty

Delete
Done

8. Select **Done**.
9. Run your query.
10. Validate that your data moved to the destination you specified.

For example, open your Google sheet file and validate that is populated with data.

When you execute your query, Treasure Data query result is imported into Elastic Cloud.

## Validate Your Export Data within the Elastic Cloud Instance

You can sanity check the data on your elastic search index with a simple query. Assuming the IP and port on your Elastic Cloud instance are `example.com:9200`, the following command can dump all your data to a file:

```
$ curl -XGET -i 'http://example.com:9200/*/_search' --user <username>:<password> > dump.txt
```

The result is a JSON file with the column names, column types, and content according to the data you've previously exported there. An example of what an Elastic Cloud query might output is as follows:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 2283

{"took":4,"timed_out":false,"_shards":{"total":15,"successful":15,"failed":0},"hits":{"total":100024,"max_score":1.0,"hits":[{"_index":"embulk_20160205-141457","_type":"embulk_type","_id":"AVKxyShGu46fqokIoDTf","_score":1...
```

## Tune Timeout Exceptions

Increasing `Bulk actions` and `Bulk size` helps increase the records in every insert requests and reduces the HTTP requests. If you don't get good results, consider upgrading your instance specs.