ThoughtSpot

ThoughtSpot Node Setup Guide

Release 6.0

December, 2019

© COPYRIGHT 2015, 2019 THOUGHTSPOT, INC. ALL RIGHTS RESERVED.

910 Hermosa Court, Sunnyvale, California 94085

This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior consent in writing from ThoughtSpot, Inc.

All rights reserved. The ThoughtSpot products and related documentation are protected by U.S. and international copyright and intellectual property laws. ThoughtSpot and the ThoughtSpot logo are trademarks of ThoughtSpot, Inc. in the United States and certain other jurisdictions. ThoughtSpot, Inc. also uses numerous other registered and unregistered trademarks to identify its goods and services worldwide. All other marks used herein are the trademarks of their respective owners, and ThoughtSpot, Inc. claims no ownership in such marks.

Every effort was made to ensure the accuracy of this document. However, ThoughtSpot, Inc., makes no warranties with respect to this document and disclaims any implied warranties of merchantability and fitness for a particular purpose. ThoughtSpot, Inc. shall not be liable for any error or for incidental or consequential damages in connection with the furnishing, performance, or use of this document or examples herein. The information in this document is subject to change without notice.

Table of Contents

Introduction to administration 8

Sign-in credentials for administration 9

Understand the architecture

- Architectural components 12
- Data caching 13
- Authentication frameworks 15
- Data and object security 16
- Performance considerations 19

Installation and setup

- About installation and upgrades 21
- Set your locale 24
- Test connectivity between nodes 26
- Set the relay host for SMTP 27
- Set up custom calendars 29
- Configure internal authentication 34
- Configure SSL 37
- Configure SAML 41
- Enable Active Directory based access 44
- Integrate LDAP
  - About LDAP integration 46
  - Configure authentication through Active Directory 47
  - Add the SSL certificate for LDAP 50
  - Test the LDAP configuration 51
  - Sync users and groups from LDAP 52
- Configure NAS file system 56
- Set up monitoring 58
- Configure support services 60
- Network policies 66
- Configure load balancing and proxies 78
- Customize ThoughtSpot Help 80
- Customize look and feel 82
- Add the Slack integration 86

Load and manage data

Introduction to data management 87

Configure casing 89

Load CSV files with the UI 90

How to view a data schema 96

Plan the schema

About schema planning 102

Data types 105

Constraints 108

Sharding 112

Chasm traps 118

Build the schema

Schema building overview 121

Connect with TQL and create a schema 123

How to write a SQL script 125

Schema creation examples 127

Upload an SQL script 132

Change the schema

How to change a schema 134

Convert column data type 139

Load bulk data

Import CSV files with tsload 144

Use a script to load data 152

Delete a data source

Delete a data source (table) 157

Delete or change a table in TQL 161

Manage users and groups

Understand groups and privileges 163

Create, edit, or delete a group 169

Create, edit, or delete a user 183

Allow users to sign up 192

Security

Overview of security features 194

System security

Tools and processes 195

About third-party software 198
Installing third-party software 201
Data security
Data security 203
Share tables and columns 209
Share worksheets 211
Share a pinboard 213
Security for SpotIQ functions 217
Revoke access (unshare) 218
Row level security (RLS)
About row level security (RLS) 221
How Rule-Based RLS works 223
Set Rule-Based RLS 228
ThoughtSpot Lifecycle 231
Encryption of data in transit 237
System administration
Overview of System administration 239
Send logs when reporting problems 240
Set up recording for Replay Search 242
Upgrade a cluster 247
Backup and restore
Understand the backup strategies 248
Understand backup/snapshot schedules 251
Work with snapshots 256
Work with backups
Understand backup modes 260
Create a manual backup 263
Configure periodic backups 266
About restore operations 272
Improve search with modeling
About data modeling 273
Change a table's data model 275
Edit the system-wide data model 277
Data model settings
Overview of the settings 282

Set column name, description, and type 285
Set additive and aggregate values 287
Hide a column or define a synonym 291
Set columns to exclude from SpotIQ analyses 293
Manage suggestion indexing 295
Add a geographical data setting 301
Set number, date, currency formats 303
Change the Attribution Dimension setting 309

Link tables using relationships

Link tables using relationships 312
Delete a relationship 316
Use stickers 318

Simplify search with worksheets

Create and use worksheets 322
Edit a worksheet 329
Create a formula in a worksheet 331
Create worksheet filters 334
How the worksheet join rule works 339
Change join rule or RLS for a worksheet 341
Create a join relationship 343
Modify joins between Worksheet Tables 347
Delete Worksheets or Tables 350
Migrate or restore Worksheets 353
Worksheet YAML specification 357

Set up SearchIQ

Enable SearchIQ 365
Optimize SearchIQ 367
Set entity categories for SearchIQ 369
Change SearchIQ mappings 371

Manage scheduled jobs

Job management (scheduled pinboards) 374
Scheduled pinboards management 377

Monitoring

Introduction to monitoring 379
Overview board 381

Data board 394
Cluster Manager board 397
Alerts and Events board 400
System worksheets 402
System pinboards 404

Troubleshooting

About troubleshooting 406
Get your configuration and logs 407
Upload logs to ThoughtSpot Support 411
Network connectivity issues 413
Check the timezone 414
Browser untrusted connection error 415
Characters not displaying correctly 416
Clear the browser cache 417
Cannot open a saved answer that contains a formula 420
Data loading too slowly 423
Search results contain too many blanks 424

Overview 425

Hardware appliance

Hardware appliance overview 426
Supported hardware 429
Cable networking 432
Set up and start the appliance 437

Cloud

Cloud overview 441

AWS

Configuration options 444
Set up ThoughtSpot in AWS 447

Azure

Configuration options 454
Set up ThoughtSpot in Azure 455

GCP

Configuration options 466
Set up ThoughtSpot in GCP 468

VMware

Configuration overview 479
Set up ThoughtSpot in VMware 482
Network policies 488
Contact support 500
Introduction to Data Integration 502
Embrace
Overview 504
JDBC and ODBC setup prerequisites 507
ODBC driver client
ODBC driver overview 508
ODBC on Windows
Install the ODBC driver on Windows 512
Configure multiple connections on Windows 522
Deploy SSL with ODBC on Windows 527
Set up the ODBC driver for SSIS 534
Install the ODBC driver on Linux 545
Best Practices for Using ODBC 556
JDBC driver client
JDBC driver overview 557
Use the JDBC driver 559
Set up the JDBC driver for Pentaho 565
Troubleshooting
Troubleshooting Data Integrations 579
Enable ODBC logs 580
Enable JDBC logs 586
Schema not found error with ODBC 587
How to improve throughput 589
ODBC tracing on Windows 590
Reference
Supported SQL commands 592
Connection configuration 594
Keyword reference 597
TQL reference 605

tsload flag reference 618

tscli command reference 621

Date and time formats reference 666

Row level security rules reference 670

Formula function reference 683

Alert codes reference 701

User action codes reference 710

Frequently asked questions 712

Introduction to administration

Summary: This guide covers all topics of special interest to application administrators.

This guide provides information for application administrators, or users with administrative access privileges [See page 9].

Before addressing the major components of this guide, we recommend that you familiarize yourself with the general top-level architecture [See page 12] of the ThoughtSpot service.

Administrators are responsible for many facets of the ThoughtSpot service. They are most frequently in charge of these common processes:

- Installation and setup of ThoughtSpot [See page 21]
- Loading and managing data [See page 87]
- Managing users and groups [See page 163]
- Security [See page 194]
- System administration [See page 239]
- Backup and Restore [See page 0]

Additionally, administrators are often involved in the following workflows:

- Data modeling [See page 273]
- Using worksheets [See page 322] to simplify search
- Using views [See page 0] for 'stacked' search; note that starting with Release 5.2, you can accomplish some aspects of search stacking by using the `IN` keyword [See page 0].
- **Beta** Enabling SearchIQ [See page 0], ThoughtSpot's natural language search.
- Managing scheduled jobs [See page 374]
- Monitoring system health [See page 379]
- Troubleshooting [See page 406]

Sign-in credentials for administration

> **Summary:** You must have administrative access to perform various administrative tasks.

You can access ThoughtSpot through SSH at the command prompt, and from a Web browser.

Administrative access

Each ThoughtSpot cluster has three default users. Contact your ThoughtSpot support team to get the passwords.

Type	Username	Description
Shell user	`admin`	For work that requires `sudo` or `root` privileges Not for application login Logs for this user are in `/usr/local/scaligent/logs` directory
Shell user	`thoughtspot`	For command-line work that does not `sudo` or `root` privileges Can use `tsload` , `tql` , and check the cluster status Not for application login Logs for this user are in `/tmp` directory
Application user	`tsadmin`	Access through a Web browser

Both the `admin` [See page 9] and `thoughtspot` [See page 9] user can SSH into the cluster. After authenticating, either user can use and and all of the following utilities:

- tscli [See page 621]; `thoughtspot` [See page 9] user cannot use commands that require `sudo` or `root` privileges
- tsload [See page 605]
- tql [See page 144]

SSH to the appliance

To perform basic administration such as checking network connectivity, starting and stopping services, and setting up email, log in remotely as the Linux administrator user “admin”. To log in with SSH from any machine, you can use the command shell or a utility like Putty.

In the following procedure, replace `<hostname_or_IP>` with the hostname or IP address of a node in ThoughtSpot. The default SSH port (22) will be used.

1. Log in to a client machine and open a command prompt.
2. Issue the SSH command, specifying the IP address or hostname of the ThoughtSpot instance:

```
ssh admin@<hostname_or_IP>
```

3. Enter the password for the admin user.

Sign in to the ThoughtSpot application

To set up and explore your data, access the ThoughtSpot application from a standard Web browser, using a username and password.

Before accessing ThoughtSpot, you need the following:

- ❏ The Web address (IP address or server name) for ThoughtSpot
- ❏ A network connection
- ❏ A Web browser
- ❏ A username and password for ThoughtSpot

ThoughtSpot supports the following Web browsers:

Firefox

68.x, 69.x, and later

Chrome

76.x, 77.x, and later

Internet Explorer

11.x, and later

Edge

44.x, and later

Safari

13.x, and later

> ☑ **Tip:** We support, but do not recommend, the use of the Internet Explorer.
>
> Depending on your environment, you can experience performance or UI issues.

To sign in to ThoughtSpot from a browser, follow these steps:

1. Open the browser and type in the Web address for ThoughtSpot:

   `http://<hostname_or_IP>`

2. Enter your username and password, and click **Sign in**.

Architecture components

> **Summary:** To implement ThoughtSpot it is important to understand where it sits within your overall analytics architecture and how it provides data to end users.

ThoughtSpot consists of a cluster of one or more nodes, acting together to provide analytic answers to business questions. As such, there are only a few integration points with ThoughtSpot on your network. The major components in a ThoughtSpot cluster are:

ThoughtSpot can handle a wide variety of different data sources. ThoughtSpot does all analysis against data in memory to help achieve fast results across millions and billions of records of data. ThoughtSpot caches the data in order to process it.

The ThoughtSpot appliance can be a physical appliance that ThoughtSpot ships, one or more AWS instances that are clustered together, or one or more VMware instances that are clustered together. From an external interface, regardless of the appliance type, the appliance appears to be a single instance.

For authentication (logging in), some source of user information is required. These define the login requirements and access control groups. Users must access the data from a supported browser to view saved content, or perform searched-based analytics. Finally, it is recommended that you have some sort of networked attached storage for storing backups in case of hardware failure.

Data Caching

> **Summary:** ThoughtSpot does all analysis against data in memory to help achieve fast results across millions and billions of records of data.

ThoughtSpot caches data as relational tables in memory. The tables can be sourced from different data sources and joined together. ThoughtSpot has several approaches for getting data into the cluster.

JDBC and ODBC Drivers

ThoughtSpot provides a JDBC and ODBC driver that can be used to write data to ThoughtSpot. This is useful for customers who already have an existing ETL process or tool, and want to extend it to populate the ThoughtSpot cache.

JDBC and ODBC drivers are appropriate under the following circumstances:

- have an ETL load, such as Informatica, SSIS, and so on
- have available resources to create and manage ETL
- have smaller daily loads

tsload

You can use the `tsload` command line tool to bulk load delimited data with very high throughput. Finally, individual users can upload smaller (< 50MB) spreadsheets or delimited files.

We recommend the tsload approach in the following cases:

- initial data load
- JDBC or ODBC drivers are not available
- there are large recurring daily loads
- for higher throughput; this can add I/O costs

Choosing a Data Caching Strategy

The approach you choose depends on your environment and data needs. There are, of course, tradeoffs between different data caching options.

Many implementations use a variety of approaches. For example, a solution with a large amount of initial data and smaller daily increments might use tsload to load the initial data, and then use the JDBC driver with an ETL tool for incremental loads.

Authentication

> **Summary:** ThoughtSpot provides LDAP/AD, SAML, and ThoughtSpot login to authenticate users.

ThoughtSpot provides three ways to authenticate users LDAP/AD, SAML, and ThoughtSpot login. In general, ThoughtSpot recommends that you use LDAP/AD or SAML if possible since ThoughtSpot provides only basic authentication with no restrictions on passwords, timeouts, failed logins, etc.

The following table shows each of the options and the items to consider for each.

SAML	LDAP/AD	ThoughtSpot
• Use SAML for single sign-on authentication. • Can redirect from ThoughtSpot to SAML logins. • Recommended for portal integration. • Option to sync users and groups if stored in LDAP/AD.	• Configuration. • Users authenticate against LDAP or AD. • Option to sync users and groups with ThoughtSpot to manage group membership.	• User created and managed in ThoughtSpot. • No enterprise password control (expiration, password strength, etc.). • Only recommended when SAML and LDAP are not options.

All users and groups must be known to ThoughtSpot. If you are using LDAP/AD or SAML and don't create users in ThoughtSpot, a user is created when the user first logs in. However, this user is assigned to the `All` group and can only see content available for all users.

Groups are the primary way that security is managed. Groups are not automatically created. You can create groups and users manually or you must automate the assignment from a source system. ThoughtSpot has an assignment script that works with most LDAP / AD stores. It also has public APIs that you can use to sync users and groups between source systems and your ThoughtSpot appliance.

Data and object security

ThoughtSpot provides many features for protecting data.

Object Security

Object security controls what content users see within ThoughtSpot. Objects are tables, columns in tables, worksheets, pinboards, and saved answers.

Users gain access to objects when an object owner shares access with them. Owners can share with individual users or with entire groups, giving access to everyone within that group. Objects may be shared with edit or view-only options. A user can automatically share objects with anyone else in the groups to which they belong. This has implications on setting up privileges, and on applying row-level security.

Permissive Security Mode

The default Permissive Security mode of ThoughtSpot means that when someone shares an object with you, you can see all the data it uses, regardless of explicit permissions to the parent object data. You can see a shared pinboard without having access to its underlying worksheet or table.

Advanced Security Mode

ThoughtSpot's Advanced Security mode is opposite of the default permissive mode. Unless the user has explicit permissions to the entire stack of parent objects, they cannot see the data in the child object. For example, in a shared pinboard, you can see data only if you have explicit permissions to the relevant columns of the parent worksheet. Similarly, you can only see the data in a worksheet to which you have access if you have explicit permissions to its parent table object.

Work with your ThoughtSpot support team to enable the Advanced Security Mode on the relevant clusters.

Row level security (RLS)

Row level security controls what data a user can see in each shared piece of content. Even if a user has access to a worksheet, they can only see rows from the tables they have permission to see.

RLS applies at the table level, so it automatically extends to all worksheets, saved answers, and pinboards based on that table, every time. Also, in queries where there are tables with table filters, all joins are always enforced to avoid accidentally allowing users access to data they shouldn't see.

RLS requires three things:

- A table filter with a column (possibly in a joined table) that can be used to determine who can see a row, such as account id or tenant id.
- A group that can be associated with the row of data by name. For example, if the column is `account_id` and has values of `1`, `2`, `3`, users can be assigned to groups `group_1`, `group_2`, `group_3` and then only see their data.
- Users must be assigned to the group. If they are not assigned to a group that has access, they do not see any data.

Administrative users can always see all rows of data because RLS does not apply to them.

RLS supports a hierarchy of groups, which makes it possible to grant access to some users across multiple groups.

Keep in mind that users within a group can share with one another. If you put everyone in your organization into the same group for RLS, they can share with anyone in the company.

Column level security (CLS)

Column level security lets users see certain columns in a table, but not other columns. This can be accomplished by sharing a limited set of columns in a table with specific users or groups.

Because someone can share with anyone in the same group, they can potentially share restricted columns. For example, if a *Human Resources* repository has a column with salary information, and it appears in a worksheet, any *Human Resources* group member could create an answer with visible salary information and mistakenly share with someone outside of *Human Resources*. That 'outside' person now has access to the salary information. In such cases, we recommend that you work with your ThoughtSpot support team to enable the Advanced Security Mode on the relevant clusters.

System privileges

System privileges refer to what a user can do in ThoughtSpot. For example, can they upload or download data or share with all users. These privileges are defined on a group level and inherit downwards. So, if Group A had child groups Group B and Group C, then any privilege given to Group A is also available to Group B and Group C. What this often means is that separate sets of groups are required to manage privileges.

Performance considerations

> **Summary:** Make sure you understand the performance considerations in your installation.

ThoughtSpot configuration and licensing varies by memory availability. Other considerations also impact the performance of your solution. Because some solutions perform better than others, think about the following issues before implementation.

Each node in a ThoughtSpot cluster performs ideally when it has less than 250GB of data and fewer than 0.25 billion total rows of data. For more complex schemas, we recommend even fewer rows of data per node for optimal performance. To reduce the total amount of data and rows of data, you can limit the data range to the relevant years or months, or combine long and narrow tables into wider tables when possible.

Data Boundaries

Total rows in a result of a join can have an impact on perforamence. In general, we recommend that you have fewer than 10 billion rows in a many-to-many join. Also, consider these boundaries:

Description	Boundary
Maximum number of rows that can be downloaded	10M (default: 1M)
Size in CSV format	250GB per node
Total number of rows across all tables	250GB per node
Many-to-Many (Generic) join cardinality	2.5B per node
Load frequency	Once every hour

Worksheet Boundaries

Worksheets must have less than 1000 columns. For aggregated worksheets, follow these guidelines:

- Number of columns should be less than 50

- Number of rows should be less than 10 millions

You can use an ETL process to circumvent these limitations. Speak with ThoughtSpot Customer Support to learn more.

Aggregated Worksheets and Joins

To join an aggregated worksheet with a base table, you must configure your installation to allow this behavior.

- The aggregated worksheet cannot use more than 5 component tables.
- The number of rows in the final aggregated worksheet cannot be greater than 1000.

Chasm Trap Worksheets

For chasm trap scenarios where two or more fact tables join through a shared dimension, we recommend the following boundaries:

Description	Boundary
Maximum number of fact tables in a worksheet	5
Maximum number of shared dimensions	2
Maximum number of rows in a *non* co-sharded shared dimension table of chasm trap	1B
Maximum number of rows in a co-sharded shared dimension table of chasm trap	1B

Row-level Security Boundaries

Maximum number of unique RLS rules with search data suggestions should not exceed 15K.

Scheduled Pinboards

For ideal performance of scheduled pinboards, do not exceed 50 scheduled pinboard jobs.

About installation and upgrades

Summary: As administrator, you are responsible for setting up and configuring ThoughtSpot. This guide explains how. It will also assist you in troubleshooting some common problems, finding additional resources, and contacting ThoughtSpot.

Your ThoughtSpot application software is already installed for you in a ThoughtSpot appliance. This is true for both physical and virtual appliances.

The ThoughtSpot software is updated by ThoughtSpot Support. ThoughtSpot Support will contact you to schedule an update when one becomes available.

As administrator, you are responsible for setting up and configuring ThoughtSpot. This guide explains how. It will also assist you in troubleshooting some common problems, finding additional resources, and contacting ThoughtSpot.

Display your current configuration

To perform the setup and configuration, you must first learn how to gain administrative access [See page 9].

1. Log into the ThoughtSpot cluster as the `admin` user.
2. Use the `tscli feature` subcommand to display your current configuration.

```
 $ tscli feature get-all-config
 +--------------------------------+----------
+---------------+
 |              NAME              |  STATUS  | CONFIGUR
ATION |
 +--------------------------------+----------
+---------------+
 | Firewall                       | Disabled
|               |
 | Saml                           | Disabled
|               |
 | Ldap                           | Disabled
|               |
 | CustomBranding                 | Disabled
|               |
 | CustomBrandingFontCustomization | Disabled
|               |
 | DataConnect                    | Disabled
|               |
 | RLS                            | Enabled
|               |
 | Callhome                       | Enabled
|               |
 | SSHTunnel                      | Enabled
|               |
 | Fileserver                     | Disabled
|               |
 +--------------------------------+----------
+---------------+
```

Related information

The following tasks are also available:

- Set your locale [See page 24]
- Test connectivity between nodes [See page 26]
- Set up a fiscal calendar year [See page 29]
- Integrate LDAP [See page 46]
- Set up monitoring [See page 58]
- Configure support services [See page 60]
- Network ports [See page 66]
- Configure load balancing and proxies [See page 78]

- Customize look and feel [See page 82]
- Add the Slack integration [See page 86]

Set your ThoughtSpot locale

Summary: You can change the language displayed in the application.

By default, the language that ThoughtSpot UI displays depends on the system locale. It is simple to change it using the **Profile** interface.

The **Language** selection specifies more than just the language: it sets the locale, which controls both the language choice and standard data formats for date and number. So, if you set French as the default locale in your profile settings, the interface updates to reflect this. Be sure to refresh your browser page.

For example, in the United States the number format for large numbers uses the comma thousands separator and a period decimals separator, and looks like this: `xxx,xxx.xx`. In most European countries, they use the reverse notation, with comma decimals separator and period thousands separator, like this: `xxx.xxx,xx`.

In addition to American English (*en-US*), ThoughtSpot supports the following locales:

Locale	Language
da-DK	Dansk
de-DE	Deutsche
en-AU	English (Australia)
en-CA	English (Canada)
en-GB	English (United Kingdom)
en-US	English (United States)
es-US	Español (latín)
es-ES	Español (España)
fr-CA	Français (Canada)
fr-FR	Français (France)
it-IT	Italiano

Locale	Language
nl-NL	Nederland (beta)
nb-NO	Norsk
pt-BR	Português (Brazil)
pt-PT	Português (Portugal)
fi-FI	Suomi
sv-SE	Svenska
zh-CN	中文(简体)
ja-JP	日本語

ThoughtSpot translates keywords, operators, and error messages. See the keyword reference for all supported languages [See page 597].

ThoughtSpot *DOES NOT* translate formulas, or metadata entered by the user. For example, if you name a visualization ‘Quarterly Sales’ in any variant of English and subsequently change the locale to a variant of French, the visualization remains ‘Quarterly Sales’ and does not become ‘Ventes trimestrielles’.

Test network connectivity between nodes

Summary: Verify your network is properly configured for the application.

This procedure tests the network connectivity between all ThoughtSpot nodes, and to the LAN. If you can perform these steps successfully, the network settings on ThoughtSpot are correct.

1. Log in to the Linux shell using SSH.
2. Ping each of the other nodes in the cluster.
3. Ping another machine that exists outside of the cluster, for example, a machine that you will use to stage data to be loaded.

If you cannot perform these tests successfully, there is a problem with the network setup. If the tests fail, check Network connectivity issues [See page 413].

Set the relay host for SMTP (email)

Summary: ThoughtSpot uses emails to send critical notifications to ThoughtSpot Support. A relay host for SMTP traffic routes the alert and notification emails coming from ThoughtSpot through an SMTP email server.

Configure using tscli

Set up SMTP rules to send critical email notifications to ThoughtSpot Support.

Set up the relay Host

To set up a relay host:

1. Log in to the Linux shell using SSH.
2. Issue the setup command, providing the IP address of the relay host:

```
$ tscli smtp set-relayhost <IP_address>
```

3. Verify your settings:

```
$ tscli smtp show-relayhost
```

4. Verify that email is working.

Configure an email to receive alerts

ThoughtSpot sends alerts to the email address specified during installation. If no email address was entered, no alerts are sent. You should add an email to receive alerts by issuing:

```
$ tscli monitoring set-config --email <your_email>
```

To send to multiple emails, provide a comma-separated list with no spaces.

Verify the relay with an email

Check if the email settings are working properly by using this procedure.

1. Log in to the Linux shell using SSH.
2. Try sending an email to yourself by issuing:

```
$ echo | mail -s Hello <your_email>
```

3. If you receive the email at the address(es) you supplied, email is working correctly.

Set up custom calendars

Summary: With a custom calendar, you can choose when the year, quarter, or week starts, and search using date-related keywords.

You can create custom calendars for different purposes within your company, including:

- Retail calendars, like 4-4-5, 4-5-4, or 5-4-4
- Fiscal calendars, where the year or quarter starts on a different date each year
- Pharmaceutical calendars, where the week runs from Friday through Thursday
- Calendars in different languages

Options

In a custom calendar, you can set the following:

- The starting date of a year
- The starting dates of quarters
- The starting day of the week
- The words used for the days of the week, months and quarters

Search features

You can use your custom calendar to search the following ways:

- Using date keywords, like `this quarter` and `q3`
- Using date formulas with the `fiscal` option specified [See page 0]
- Overriding the calendar used in the search bar by typing your custom calendar's name

Limitations

- Maximum scope of the date dimension is 90 years
- Maximum length of a calendar year is 12 months

Setting up a custom calendar

To set up a custom calendar for your cluster, you must do the following:

1. Enable the custom calendar feature.
2. Generate a calendar template.
3. Edit the calendar template.
4. Add the custom calendar to your cluster.

Enable the custom calendar feature

To enable the custom calendar feature for your cluster, contact ThoughtSpot Support [See page 0].

Generate a calendar template

Using a calendar template as your starting point ensures that you use a format that is compatible with ThoughtSpot.

To generate a calendar template, do the following:

1. SSH as admin into your ThoughtSpot cluster: `ssh admin@<cluster-ip-address or hostname>`.

2. Run the `tscli calendar generate` command using the following syntax:

   `tscli calendar generate --name <calendar_name> --start_date <MM/DD/YYYY> --end_date <MM/DD/YYYY> --username tsadmin`

   Example: `tscli calendar generate --name my_calendar --start_date 07/01/2019 --end_date 06/30/2020 --username tsadmin`

   This generates a calendar template file in .csv format. In the previous example: **my_calendar.csv**.

3. Exit your SSH session.

Edit the calendar template

To use the template you generated as your custom calendar, some editing is required.

1. Download the .csv file to your computer using following syntax:

   `scp admin@<cluster-ip-address>:/home/admin/<calendar_name>.csv /<Local directory on your computer>/.`

   Example (on Mac OS): `scp admin@172.18.144.217:/home/admin/my_calendar.csv /Users/john.smith/Desktop/.`

2. Open the .csv file in a text editor or spreadsheet program and edit the file to ensure the date and quarter columns are formatted correctly:
   - The Date column must use the format: **MM/DD/YYYY**. No other formats are supported.
   - The Quarter column must display the correct quarter number for each day of the year.

     > ℹ **Note:** By default, a generated calendar displays quarter numbers based on the Gregorian calendar (which starts on January 1st). If your custom calendar begins any other date, you must adjust the quarter numbers to align with your calendar. For example: If your custom calendar begins on April 1st, the calendar would incorrectly show April, May and June as quarter 2. You would need to correct this to indicate those months are quarter 1 and correct the subsequent months to have the correct quarter.

   - (Optional) To enhance searchability, ThoughtSpot recommends adding a “Q” before each quarter number. Example: **Q1**. If adapting the calendar to different language, use the appropriate letter in place of “Q”.
   - Make any other changes needed to the calendar (like translating months or days into a different language.)

Example calendar with the fiscal year beginning on April 1:

date	day_of_week	month	quarter	year	day_number_of_week	week_number_of_month	week_number_of_quarter	week_number_of_year	is_weekend
4/1/19	monday	april	Q 1	2019	1	1	1	1	FALSE
4/2/19	tuesday	april	Q 1	2019	2	1	1	1	FALSE
4/3/19	wednesday	april	Q 1	2019	3	1	1	1	FALSE
4/4/19	thursday	april	Q 1	2019	4	1	1	1	FALSE
4/5/19	friday	april	Q 1	2019	5	1	1	1	FALSE
4/6/19	saturday	april	Q 1	2019	6	1	1	1	TRUE
4/7/19	sunday	april	Q 1	2019	7	2	1	1	TRUE
4/8/19	monday	april	Q 1	2019	1	2	2	2	FALSE
4/9/19	tuesday	april	Q 1	2019	2	2	2	2	FALSE
4/10/19	wednesday	april	Q 1	2019	3	2	2	2	FALSE
4/11/19	thursday	april	Q 1	2019	4	2	2	2	FALSE
4/12/19	friday	april	Q 1	2019	5	2	2	2	FALSE
4/13/19	saturday	april	Q 1	2019	6	2	2	2	TRUE
4/14/19	sunday	april	Q 1	2019	7	3	2	2	TRUE
4/15/19	monday	april	Q 1	2019	1	3	3	3	FALSE
4/16/19	tuesday	april	Q 1	2019	2	3	3	3	FALSE
4/17/19	wednesday	april	Q 1	2019	3	3	3	3	FALSE

3. Save your calendar template as a UTF-encoded .csv file with UNIX line breaks.

> ℹ **Note:** Saving the file with UNIX line breaks, ensures there are no carriage returns (^M characters) in the file which prevent you from using your calendar in ThoughtSpot. Microsoft Excel, for example, adds carriage returns. The easiest way to remove carriage returns is to open your .csv file in a text editor, and save it as a .csv with UNIX line breaks.

Add the custom calendar to your cluster

To use your edited calendar template as a custom calendar, you must upload it to your cluster and use it to create a calendar in ThoughtSpot.

1. Upload the .csv file to your ThoughtSpot cluster using the following syntax:

   ```
   scp /<Local directory on your computer>/<calendar_template_name>.csv
   admin@<cluster-ip-address>:/home/admin/
   ```

   Example (on Mac OS): `scp /Users/john.smith/Desktop/my_calendar.csv admin@172.18.144.217:/home/admin`

2. SSH as admin into your ThoughtSpot cluster: `ssh admin@<cluster-ip-address or hostname>` .

3. Run the `tscli calendar create` command using the following syntax: `tscli calendar create --file_path /home/admin/<calendar_template_name>.csv --name <calendar name> --username tsadmin`

   Example: `tscli calendar create --file_path /home/admin/my_calendar.csv --name my_calendar --username tsadmin`

(Optional) Set a custom calendar as the default calendar for your cluster

To set your custom calendar as the default calendar for your cluster, contact ThoughtSpot Support [See page 0].

Setting a worksheet, table or view to use your custom calendar

If you don't set your custom calendar as the default for your cluster, you must do the following to use your calendar:

1. Sign in to your ThoughtSpot cluster and click **DATA**.

2. On the DATA page, click the name of a worksheet, table or view in which you want to use your custom calendar.

3. Under COLUMN NAME, find a column that uses the DATE or DATE_TIME data type where you want to use your custom calendar and scroll right until you see the CALENDAR TYPE column.

   > ℹ **Note:** The column must use the DATE or DATE_TIME data type.

4. In the CALENDAR TYPE column for the column(s) you chose, double-click the existing calendar name, and then select your custom calendar.

5. Click **Save Changes**.

   Now, date-related searches in the selected worksheet, table or view use your custom calendar.

Configure internal authentication

Summary: Many organizations don't use LDAP or active directory for user authentication. In other scenarios, some users are not in LDAP and are created only in ThoughtSpot. In such cases, you can choose to authenticate users against ThoughtSpot internal authentication.

Many organizations don't use LDAP or active directory for user authentication. In other scenarios, some users are not in LDAP and are created only in ThoughtSpot. In such cases, you can choose to authenticate users against ThoughtSpot internal authentication.

> **Note:** Before the user logs into ThoughtSpot, the user must exist in ThoughtSpot. This is independent of the authentication mechanism.

If you have been using ThoughtSpot with users you created manually, and you now want to transition to LDAP, please contact ThoughtSpot Support [See page 0]. They can assist you in migrating existing users to their LDAP equivalents. <!–

Configure internal authentication

To authenticate users using ThoughtSpot internal who are not available in the LDAP or AD system:

1. Log into ThoughtSpot from a browser.
2. Click the **Admin** menu on the top navigation bar.

   This opens the ThoughtSpot Management Console.

3. Click **Settings** menu on the top navigation bar.

4. In the Settings panel, click **Authentication: Internal** and then **Configure** option.

5. Choose **Enable** under Authentication: Internal option. If you want to disable the internal authentication, configure AD or LDAP first, and then select **Disable** and save the configuration.

6. Click **Save** to configure the internal authentication. –>

Configure SSL

Summary: Secure socket layers (SSL) provide authentication and data security when sending data to and from ThoughtSpot.

You can use SSL to enable both HTTP and LDAP data security.

About SSL

Companies usually secure applications that access data. To use SSL with ThoughtSpot, you must use your company's own SSL certificate. The certificate is issued for each domain or service. If you plan to use SSL for both HTTP(S) and LDAP(S), you must have two separate certificates.

If you do not have an SSL certificate, there are options:

- Check with your IT department if they have an SSL certificate you can use.
- Obtain the certificate from an issuing authority.
- Disable SSL and loose the security it provides. Use the following command:

```
tscli ssl off
```

ThoughtSpot works with a wide variety of SSL types, from a wide variety of vendors.

Required ports

To use SSL, the following ports must be open:

- 443
- 80

Configure SSL for web traffic

Note: Do not use a passphrase when creating certificates.

To verify if you're prompted to specify a passphrase, invoke the command `openssl rsa -check`

`-in pk.key`. If the answer is 'yes', remove the passphrase to use the key.

To add SSL and enable HTTPS in ThoughtSpot, obtain the SSL certificate chain [See page 38] and the private key [See page 38].

SSL certificate chain

The SSL certificate chain must be in `.PEM` format. This is an `X.509v3` file that contains ASCII (Base64) armored data, packed between `BEGIN` and `END` directives. It can be a bundle of certificates.

Private key

The private key must be in compatible `.PEM` format. It cannot be password or passphrase protected.

Configure SSL using tscli

Follow these instructions to install the SSL certificate using tscli:

1. Use the instructions from the certifying authority where you obtained the certificate.

   This is usually sent to you by email, or available for download.

2. Copy the certificate and key files to ThoughtSpot:

   ```
   $ scp <key> <certificate> admin@<IP_address>:<certificate-path>
   ```

3. Log into the Linux shell using SSH.

4. Change to the directory where you copied the files:

   ```
   $ cd <certificate-path>
   ```

5. To install the certificate, issue the `tscli` command:

```
$ tscli ssl add-cert <key> <certificate>
```

6. To test that the certificate is correctly installed, log into the ThoughtSpot application [See page 0].

   You should see that the application's URL begins with `https://`.

Set the recommended TLS version

There are a couple of security vulnerabilities due to SSL certificates supporting older versions of TLS (Transport Layer Security). This procedure shows you how to set the recommended TLS version to avoid these vulnerabilities.

The PCI (Payment Card Industry) Data Security Standard and the FIPS 140-2 Standard require a minimum of TLS v1.1 and recommends TLS v1.2.

ThoughtSpot supports SSL v3, TLS v1.0, and TLS v1.1 for backwards compatibility. However, the recommended version is TLS v1.2. Therefore, to set the recommended TLS version:

1. Enable your web browser to support TLS v1.2. This can be done in your browser's advanced settings.
2. Log in to the Linux shell using SSH..
3. Issue the following command:

   ```
   tscli ssl set-min-version 1.2
   ```

   This will block all usage of older versions.

Configuration string for load balancers

When enabling SSL support on a load balancer's server-side SSL client profile, use the following list of ciphers to ensure compatibility between the load balancer and ThoughtSpot.

```
EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
```

The following ciphers are currently supported:

```
|   TLSv1.2:
|     ciphers:
|       TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 - strong
|       TLS_DHE_RSA_WITH_AES_256_CBC_SHA - strong
|       TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 - strong
|       TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 - strong
|       TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 - strong
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA - strong
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 - strong
|       TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 - strong
|     compressors:
|       NULL
|_  least strength: strong
```

You can retrieve these from the ThoughtSpot web server (not against the load balancer) by running the following command on any ThoughtSpot node: `nmap --script ssl-enum-ciphers -p 443 <ThoughtSpot_node_IP_address>` You must ensure that your load balancer supports these ciphers.

Configure SAML

Summary: You can use the Security Assertion Markup Language (SAML) to authenticate users.

ThoughtSpot enables you to use the Security Assertion Markup Language (SAML) to authenticate user. You can set up SAML using the management console or through the shell on ThoughtSpot using a `tscli`-based configurator. It is configured to work using service provided by an Identity Provider (IDP).

Configuration prerequisites

Before you configure SAML, collect the following information:

- ❏ ThoughtSpot service address [See page 41]
- ❏ Service port [See page 41]
- ❏ Unique service name [See page 0]
- ❏ Skew time in seconds [See page 42]
- ❏ IDP Metadata XML File [See page 42]
- ❏ Automatically add SAML users to Thoughtspot [See page 42]
- ❏ Also use ThoughtSpot internal authentication [See page 42]

ThoughtSpot service address

DNS name of the load balancer *front-end* for multi-node ThoughtSpot clusters, or of ThoughtSpot *server* For single-node ThoughtSpot cluster.

Service port

Service port for ThoughtSpot instance, typically TCP/443.

Unique service name

The unique key ThoughtSpot uses to identify IDP service. Set by the ThoughtSpot Support Team.

The key has the following format: `urn:thoughtspot:callosum:saml`.

Skew time in seconds

Allowed skew time for authentication, or the duration after authentication response is rejected and sent back from the IDP.

Usually set to `3600` seconds.

IDP Metadata XML File

This file is provided by the IDP. The absolute path to the `idp-meta.xml` file is needed for one-time configuration.

Automatically add SAML users to Thoughtspot: (yes/no)

If you choose 'yes', then new users will be automatically created in ThoughtSpot. If 'no', then SAML users will be added in ThoughtSpot upon first successful SSO login.

Also use ThoughtSpot internal authentication: (y/n)

If 'y', then ThoughtSpot local/internal users (including local administrative users) will still be authenticated outside the scope of SSO.

Configure SAML using tscli

> ℹ **Note:** The configuration persists across updates to newer releases of ThoughtSpot.

To set up SAML on ThoughtSpot for user authentication, follow these steps:

1. Log into the Linux shell using SSH.

2. Run the `saml configure` command to launch the interactive SAML configuration:

```
tscli saml configure
```

3. Complete the configurator prompts with the information you collected in Configuration prerequisites [See page 41].

4. When the configuration completes, open a browser and navigate to the ThoughtSpot login page. It should show the SSO option.

Enable Active Directory based access

Summary: ThoughtSpot supports enabling Active Directory (AD) based access individually on each node where the commands are run.

Enable Active Directory based access on a ThoughtSpot node

ThoughtSpot supports enabling Active Directory (AD) based access individually on each node where the commands are run. There is no provision to enable AD access for the whole cluster with a single command. To enable AD access on a cluster, you need to run these commands on each individual node and on any additional nodes added to the cluster.

The command to enable system AD user access is:

```
tscli sssd enable --user <USER> --domain <DOMAIN>
```

You will then be prompted for password credentials.

> **Note:** The user must have permission to join a computer or VM to the domain.

Set sudoers AD Group on a local node

Just like enabling AD based access on a node, setting `sudo` AD group applies only on the node where the command is run, and is not set for the whole cluster.

The command to allow `sudo` permissions for AD group:

```
tscli sssd set-sudo-group <ACTIVE_DIRECTORY_GROUP_NAME>
```

Clear sudoers AD Group on a local node

Clearing `sudo` AD group only applies on the node where command is run, and is not set for the whole cluster.

The command to clear `sudo` permissions for the AD group:

```
tscli sssd clear-sudo-group <ACTIVE_DIRECTORY_GROUP_NAME>
```

Disable AD based access on a local node

Currently ThoughtSpot supports disabling AD based access individually on each node where the commands are run. There is no provision to disable AD access for the whole cluster with a single command. To disable AD access on a cluster, run these commands on each individual node and any additional nodes added to the cluster.

Command to disable system AD user access is:

```
tscli sssd disable
```

> **Note:** Running this command will also remove the AD group from sudoers list.

Related Information

- sssd [See page 0] in the `tscli` command reference

About LDAP integration

Summary: You authenticate users against an LDAP server.

Some companies use LDAP (Lightweight Directory Access Protocol) to manage user authentication. Using LDAP provides security and makes user management more centralized. You can choose to authenticate users against an LDAP server, against ThoughtSpot, or both.

ThoughtSpot supports both anonymous and non-anonymous LDAP integration. Non-anonymous LDAP binding is more rigorous than anonymous authentication, but it should help you track what your users are querying and keep a log trace for auditing purposes.

If you have been using ThoughtSpot with users you created manually, and you now want to transition to LDAP, please contact ThoughtSpot Support. They can assist you in migrating existing users to their LDAP equivalents.

ThoughtSpot supports LDAP with Active Directory [See page 47].

Configure authentication through Active Directory

Summary: ThoughtSpot enables you to set up integration with LDAP using Active Directory. After successful setup, you can authenticate users against a secure LDAP server.

Configuration prerequisites

Before you configure ThoughtSpot for Active Directory, collect the following information:

- ❏ URL [See page 47]
- ❏ Domain name [See page 47]
- ❏ Search base [See page 48]
- ❏ SSL [See page 48]
- ❏ Automatically add LDAP or AD users in ThoughtSpot? [See page 48]
- ❏ Also use ThoughtSpot internal authentication? [See page 48]

URL

Required to connect to Active Directory.

For example, `ldap://ad.yourdomain.local:389` or `ldap://ad.yourdomain.local:636`

Domain name

Default domain under which users who want to be authenticated against Active Directory reside. When a user logs in with a username, the default domain is added to the username before sending it to the LDAP server. If users reside in multiple sub-domains, you can still designate one of them as the default. Authentication against multiple domains is not supported.

Users who don't belong to the default domain must explicitly qualify their username when they log in.

For example: `username@ad.yourdomain.local`

Search base

LDAP search base. The scope of searching user information, like *email* and *Display name*, within AD.

SSL

If you want to use SSL, you must obtain the SSL certificate from an issuing authority.

If AD servers are behind a load balancer, you must procure the SSL certificate to identify ThoughtSpot to the load balancer. The communication after the load balancer is non-secure. ThoughtSpot does not support a scenario where multiple AD servers provide their own SSL certificates.

Automatically add LDAP or AD users in ThoughtSpot? (yes/no)

If you choose 'yes', new users are automatically created within ThoughtSpot when successfully authenticated against AD. ThoughtSpot doesn't cache passwords for AD-authenticated users.

If you choose 'no', users have to be manually created with a dummy password as a placeholder in ThoughtSpot before they can log in. The username you specify when creating the LDAP-authenticated user manually in ThoughtSpot has to be domain qualified, for example: `username1@ad.yourdomain.local`.

In order to log in to ThoughtSpot, the user has to exist in ThoughtSpot independent of whether that user is authenticated against AD or against ThoughtSpot's internal authentication.

Also use ThoughtSpot internal authentication? (yes/no)

If you choose 'yes', ThoughtSpot will first attempt to authenticate the user against AD. If that attempt fails, it will then attempt to authenticate the user as an internal/local ThoughtSpot user. If either of these succeed, then the user is successfully logged in. This is useful in scenarios where some users are not in AD and are created only in ThoughtSpot.

Configure LDAP using tscli

You do not have to create a user called `tsadmin` on your LDAP server. Internal authentication can be used for `tsadmin`. To configure AD based authentication, follow these steps:

1. Log in to the Linux shell using SSH.
2. Run the command to configure AD authentication:

```
$ tscli ldap configure
```

3. Answer the prompts using the information you collected under **Before you begin** section. For example:

```
 Choose the LDAP protocol:
 [1] Active Directory
 Option number: 1
 Configuring Active Directory
 URL to connect to Active Directory. (Example: ldap://a
d.yourdomain.local:389): ldaps://ad.yourdomain.local:636
 Default domain (Example: ldap.thoughtspot.com): yourdom
ain.local
 Use SSL (LDAPS) (y/n): n
 LDAP search base (Example: cn=Users): cn=Users,ou=orgun
it,dc=youdomain,dc=local
 Automatically add LDAP users in ThoughtSpot (y/n): y
 Also use ThoughtSpot internal authentication (y/n): y
```

4. If you are using SSL, add the SSL certificate for AD [See page 50].
5. If you want to remove the AD configuration, issue the following command:

```
$ tscli ldap purge-configuration
```

Add the SSL certificate for LDAP

Summary: Install the certificate to support LDAPS

When you set up LDAP, you specified whether or not to use SSL for LDAP (LDAPS). If using SSL, you must install the LDAP SSL certificate. Before you can add the SSL certificate, you must Configure LDAP for Active Directory [See page 47].

You must have the SSL certificate before you start. For more information on obtaining an SSL certificate, see Configure SSL (secure socket layers) [See page 37].

To add the SSL certificate for LDAP:

1. Follow the instructions from your certifying authority to obtain the certificate. This is usually sent by email or available for download.
2. Copy the certificate to ThoughtSpot:

```
$ scp <certificate> admin@<IP_address>:<path>
```

3. Log in to the Linux shell using SSH.
4. Change directories to where you copied the certificate:

```
$ cd <path>
```

5. Run the command to configure SSL for LDAP, designating an alias for this certificate using the `<name>` parameter:

```
$ tscli ldap add-cert <name> <certificate>
```

Test the LDAP configuration

Summary: This procedure allows you to test the LDAP connection you created.

After configuring LDAP, you can test to make sure it is working by issuing a command.

1. Log in to the Linux shell using SSH.
2. Issue the LDAP testing command, supplying the information for the LDAP server you configured, as in this example:

```
 $ ldapsearch -x -h 192.168.2.61 -p 389 -D "testuser@lda
p.thoughtspot.com" -W -b "dc=ldap,dc=thoughtspot,dc=co
m" cn
```

3. Supply the LDAP password when prompted.
4. If the connection works, you can see a confirmation message.

Sync users and groups from LDAP

Summary: Use this procedure to synchronize your ThoughtSpot system with an LDAP server.

Before synchronizing users and groups, you need this information:

- IP address and port of the server where your ThoughtSpot instance is running. This hostport is needed in the following format `http(s)://<host>:<port>` or `http(s)://<domain>`.
- Administrator login username and password for your ThoughtSpot instance.
- URL of the LDAP server, or hostport.

  For example, `ldap://192.168.2.48:389`

- Login username and password for the LDAP system.

  An example username would be `moo_100@ldap.thoughtspot.com`

- Distinguished Name (DN) for the base to start searching for users in the LDAP system.

  For example, `DC=ldap,DC=thoughtspot,DC=com`

- Location of the Python synchronization script, in case you want to modify it or create your own: `/usr/local/scaligent/release/callosum/utilities/ldap_sync_python_api/syncUsersAndGroups.py`

There are two ways for you to fetch users and groups from LDAP and populate them into your ThoughtSpot system:

- Run the synchronization script in interactive mode, which will walk you through the process (shown here).
- Create your own Python script by using the ThoughtSpot Python APIs. If you need details on the Python APIs, contact ThoughtSpot Support. If you choose this method, you can run the script periodically using a cron job.

To run the LDAP sync script in interactive mode:

1. Log in to the Linux shell using SSH.

2. Run the command to start the script:

```
python syncUsersAndGroups.py interactive
```

3. Answer the prompts using the information you collected above. For example:

```
 Complete URL of TS server in format "http(s)://<hos
t>:<port>": http://10.77.145.24:8088
 Disable SSL authentication to TS server (y/n): y
 Login username for ThoughtSpot system: admin
 Login password for ThoughtSpot system: 12345
 Complete URL of server where LDAP server is running in
format ldap(s)://<host>:<port>: ldap://192.168.2.48:389
 Login username for LDAP system: moo_100@ldap.thoughtspo
t.com
 Login password for LDAP system: 12345
 Syncs user and groups between LDAP and TS systems (y/
n): y
 Delete entries in ThoughtSpot system that are not curre
ntly in LDAP tree being synced (y/n): n
 Distinguished name for the base to start searching grou
ps in LDAP System: DC=ldap,DC=thoughtspot,DC=com
 Scope to limit the search to (choice number)
 0:base Searching only the entry at the base DN
 1:one Searching all entries on level under the base DN
- but not including the base DN
 2:tree Searching of all entries at all levels under an
d including the specified base DN: 2
```

```
 Filter string to apply the search to: (|(CN=TestGroupAl
pha)(CN=TestGroupBeta))
```

Answering this prompt is optional. If left blank, the default value of `'(CN=*)'` will be used.

```
 Apply sync recursively, i.e. Iterates through group mem
bers and creates member groups, users and relationships
in a recursive way. (y/n): n
```

This prompt is asking if you would like to include group members even if they do not belong to the current sub tree that is being synced.

4. Alternatively, to input your own shorthand script commands:

   Issue the Python script commands, supplying all this information, following this format example:

```
 python syncUsersAndGroups.py script \
 --ts_hostport <ts_hostport> \
 --disable_ssl \
 --ts_uname <ts_username> \
 --ts_pass <ts_password> \
 --ldap_hostport '<ldap_hostport>' \
 --ldap_uname '<ldap_username>' \
 --ldap_pass '<ldap_password>' \
 --sync \
 --purge \
 --basedn 'DC=ldap,DC=thoughtspot,DC=com' \
 --filter_str '(|(CN=TestGroupAlpha)(CN=TestGroupBet
a))' \
 --include_nontree_members
```

   The bottom half of the preceding command targets sub trees under the DC called TestGroupAlpha and TestGroupBeta, and iterates through them recursively to create/sync users, groups, and their relationships in the ThoughtSpot system. It also deletes any other entities created in the ThoughtSpot system from this LDAP system that are not currently being synced.

`syncUsersAndGroups.py` command-line switches

The following table provides a description of each command-line switch available for the `syncUsersAndGroups` python script.

Switch	Description
`--ts_hostport <ts_host-port>`	ThoughtSpot cluster host port. Default port is 8088.

`--disable_ssl`	Controls the communication between the sync script and the ThoughtSpot cluster. It disables SSL communications between the script and the cluster ONLY, and prevents the need to provide SSL certs during the script execution in order to create users and groups.
`--ts_uname`	ThoughtSpot cluster username. The `admin` user is usually used.
`--ts_pass`	ThoughtSpot cluster password.
`--ldap_hostport`	AD/LDAP server port that is queried. Default is 389.
`--ldap_uname`	Username for the LDAP/AD server.
`--ldap_pass <ldap_pass-word>`	Password for the LDAP/AD server.
`--sync`	Syncs users and groups which match the `basedn` and `filter_str` queries to your ThoughtSpot cluster.
`--purge`	Purges any users that exist in ThoughtSpot, but not in AD.
`--basedn`	Place in the directory that will be searched for users.
`--filter_str`	Further filters results from your base DN.
`--include_nontree_mem-bers`	Includes group members from LDAP/AD even if they do not belong to the current subtree that is being synced.

Configure NAS file system

Summary: Some operations, like backup, restore, and data loading, require either reading or writing very large files. You can mount a network attached storage (NAS)) file system to support these operations. Your NAS storage can be in the drive format you choose.

About NAS mount

ThoughtSpot enables you to mount a NAS file system for storing or accessing large files. The file system will be mounted at the same location on each node in the cluster automatically. When any node is restarted, the file system will be mounted again automatically, if it can be found.

When supplying a directory for writing or reading a backup, you can specify the a new mount point within `/export` as the directory to use. Likewise, you can stage data there for loading. It is best to have 2 separate NAS volumes, individually dedicated to data loads and backups.

Backups are written by the Linux user `admin`. If that user does not have permission to write to the NAS file system, you can write the backups to a disk (for example `/export/sdc1`, `/export/sdd1`, `/export/sde1`, or `/export/sdf1`) and then set up a cron job that executes as root user and copies the backup to the NAS device every night, then deletes it from the directory.

Do not send the periodic backups or stage files on `/export/sdb1` since it is a name node. It is used internally by Hadoop Distributed File System (HDFS) and if this drive fills up, it can cause serious problems. Do not allow backups or data files to accumulate on ThoughtSpot. If disk space becomes limited, the system will not function normally.

Mount NAS using tscli

To mount a NAS file system using the tscli, follow these steps:

1. Log in to the Linux shell using SSH.
2. Mount the directory to the file system by issuing the appropriate command:
    - Example for an NFS (Network File System) directory:

```
tscli nas mount-nfs --server storageservername.fi
le.yourdomain.net
   --path_on_server <path>  /tsdev-backup --moun
t_point /export/BACKUPS/
   --options vers=<version>,sec=<security schem
e>,<OPTIONS>
```

> **Note:** Other command-line options are available to forward to the command (default: `noexec`).

- Example for a CIFS (Common Internet File System) directory:

```
tscli nas mount-cifs --server storageservername.f
ile.yourdomain.net
  --path_on_server /tsdev-backup --mount_point /e
xport/BACKUPS/
  --username 'avtprdweutspotdev' --uid 1001 --gi
d 1001 --options 'vers=3.0'
```

> **Note:** Other command-line options are available to forward to the `mount.cifs` command (default: `noexec`).

3. Use the mounted file system by referring to its mount point.

4. When you are finished with it, you can optionally unmount the NAS file system:

```
tscli nas unmount --dir <directory>
```

Set up monitoring

Summary: Setting up monitoring is a one time operation.

To configure monitoring of your cluster, set up the frequency of heartbeat and monitoring reports and an email address to receive them.

1. Log in to the Linux shell using SSH.
2. Issue the `tscli` command to set up monitoring:

   ```
   tscli monitoring set-config
      --email <email>
      --heartbeat_interval <heartbeat_interval>
      --report_interval <report_interval>
   ```

   The parameters are:

   - `--email <email>` is a comma separated list (no spaces) of email addresses where the cluster will send monitoring information.
   - `--heartbeat_interval <heartbeat_interval>` is the heartbeat email generation interval in seconds. Must be greater than 0.
   - `--report_interval <report_interval>` sets the cluster report email generation interval in seconds. Must be greater than 0.

3. To view your settings and verify that they have been applied, issue:

   ```
   tscli monitoring show-config
   ```

   You should see information like:

```
Monitoring Configuration:

  Alert Email: dev-alerts@thoughtspot.com

  Heartbeat Interval: 900 sec

  Report Interval: 21600 sec
```

4. After the heartbeat interval has passed, check your email to verify that emails are being delivered.
5. If you don't receive any emails, verify that email is working [See page 28].

Configure support services

Summary: There are several configurations you can set up in your installation to ensure your company's support from ThoughtSpot works smoothly.

Set up a reverse tunnel for support

You can set up a reverse tunnel to allow ThoughtSpot Support to get access to your ThoughtSpot instance, to perform support-related activities. This setup is scalable, more secure, and a much simpler alternative to using a virtual meeting room.

Granting remote support access can streamline troubleshooting activities, because it enables your support agent to work directly on your computer form a secure setting. The remote tunnel enables SSH and HTTP access to your ThoughtSpot instance by ThoughtSpot Support. This access can be granted and revoked easily, so you can enable it for a troubleshooting session, and then disable it again. Before doing this procedure, make sure your company's security policies allow reverse tunneling.

> **Note:** Before you set up a reverse tunnel, open port `22` in your firewall outgoing rules to whitelist `tunnelrelay.thoughtspot.com`.

Using remote support with tscli

To enable remote support, follow these steps:

1. Contact ThoughtSpot [See page 0] and open a support ticket for making the appropriate reverse tunnel settings on our end. Provide the cluster name of the cluster for which you want to enable remote support.
2. After the ticket is completed, continue with the remaining steps in this procedure to make the settings on your side.
3. Log into the Linux shell using SSH.
4. Issue the command to configure the destination for the remote tunnel.

You only need to do this one time, when you are enabling the tunnel for the very first time. After that, this setting persists when you start and stop the remote tunnel.

```
$ tscli support set-remote --addr tunnelrelay.thoughtsp
ot.com --user ubuntu
```

5. Test that the setting is configured:

```
$ tscli support show-remote
```

6. Enable the remote tunnel:

```
$ tscli support start-remote
```

7. Contact ThoughtSpot [See page 0] and test the setup with your ThoughtSpot Support contact.
8. After your remote session with ThoughtSpot Support is over, turn the remote tunnel off until you need to use it again:

```
$ tscli support stop-remote
```

You can repeat the steps to start and stop the remote tunnel as needed for future support operations.

9. Ensure that the remote tunnel is disabled:

```
$ tscli support show-remote
```

Configure a secure file server

ThoughtSpot Support uses a secure file server to distribute new releases and to access your logs and troubleshooting files. You must also use a secure server connection to enable the optional performance statistics collection.

Before uploading files to the secure file server, obtain your user name and password for logging into the secure file server. You can get these from ThoughtSpot Support.

Configuring the connection to the file server is a one-time operation. You do not have to reconfigure the connection unless your password changes. Note that you can do a one time override of the user and password you used to configure the connection. This is done by passing a different user and password on the command line when uploading or downloading a file.

To configure the connection to the secure file server, follow these steps:

1. Log in to the Linux shell using SSH.
2. Issue the command to configure the file server:

```
 $ tscli fileserver configure --user <user_name> [--pass
word <password>]
```

If you do not supply the `--password` parameter, you are prompted to enter it.

Call home with cluster usage data

“Call home” data is metadata and usage data from your ThoughtSpot cluster. This data allows ThoughtSpot’s Support team to troubleshoot your cluster. They use the data to see basic usage information over time for your ThoughtSpot instance. ThoughtSpot’s “call home” functionality intermittently sends a call home bundle of statistics to a ThoughtSpot S3 server through HTTPS. The data is encrypted at rest on the server.

By default, call home is enabled on your cluster. You can disable call home in by doing the following:

1. Log into the ThoughtSpot server as `admin` user.
2. Use the `tscli` command to disable.

```
$ tscli callhome disable
```

Designate a support contact

A support contact person can answer your questions about data and search at your company. If the person can't answer your system and software-related questions, that person should submit the questions to ThoughtSpot Support. The designated support contact should have an available email and phone number.

To designate the custom support contact, follow these steps:

1. Log into the Linux shell using SSH.
2. Issue the `tscli` command to set the email address:

```
$ tscli support set-admin-email <email_address>
```

3. Issue the `tscli` command to set the phone number:

```
$ tscli support set-admin-phone <phone_number>
```

4. If you need to reset both of these to the default (ThoughtSpot Support), issue these commands:

```
$ tscli support rm-admin-email
$ tscli support rm-admin-phone
```

Manage the feedback contact

Users in ThoughtSpot may be asked for feedback for new or BETA features in the system. By default, feedback goes directly to ThoughtSpot support.

Alternatively, and especially in cases of using ThoughtSpot in embedded mode, you can send feedback to someone in your company.

Your designated feedback contact should have an available email. To designate the custom feedback contact, follow these steps:

1. Log into the Linux shell using SSH.
2. To set the feedback email address, issue this command:

```
$ tscli support set-feedback-email <email_address>
```

3. Verify the email address is set:

```
$ tscli support show-feedback-email
```

To reset the email to the default (ThoughtSpot support), issue this command:

```
$ tscli support rm-feedback-email
```

You can also choose not to send feedback on your system. Issue this command:

```
$ tscli support set-feedback-email ' '
```

How users find your company's support contact

After you set the custom support contact information, your users can see it in the following parts of ThoughtSpot:

- In the Help Center, when a user selects **Contact Support**.

- In error messages, when a user selects **What Happened?**

Network policies

Summary: Lists the required ports, protocols and policies for an installation.

For regular operations and debugging, there are some ports you you must keep open to network traffic from end users. Another list of ports must be open for intracluster traffic, and for inbound and outbound access to the cluster. ICMP v4 is used for checking the health of the cluster.

Required ports for operations and debugging

The following ports must be open for requests from your user population. There are two main categories: operations and debugging.

Port	Protocol	Service Name	Direction	Source	Destination	Description
22	SSH	SSH	bidirectional	Administrators IP addresses	All nodes	Secure shell access. Also used for scp (secure copy).
80	HTTP	HTTP	bidirectional	All users IP addresses	All nodes	Hypertext Transfer Protocol for website traffic.
443	HTTPS	HTTPS	bidirectional	All users IP addresses	All nodes	Secure HTTP.
12345	TCP	Simba	bidirectional	Administrators IP addresses	All nodes	Port used by ODBC and JDBC drivers when connecting to ThoughtSpot.
2201	HTTP	Orion master HTTP	bidirectional	Administrator IP addresses	All nodes	Port used to debug the cluster manager.
2101	HTTP	Oreo HTTP	bidirectional	Administrator IP addresses	All nodes	Port used to debug the node daemon.
4001	HTTP	Falcon worker HTTP	bidirectional	Administrator IP addresses	All nodes	Port used to debug the data cache.

Port	Protocol	Service Name	Direction	Source	Destination	Description
4251	HTTP	Sage master HTTP	bidirectional	Administrator IP addresses	All nodes	Port used to debug the search engine.

Required ports for cluster communication

This reference lists the potential ports to open when setting up your security group.

Required ports for intracluster operation

Static ports are used for communication between services within the cluster. ThoughtSpot recommends that you open all ports within a cluster. This not required, but it will ensure that cluster communication works properly if additional ports are used in a future software release.

If your organization does not allow you to open all ports, make sure you open the required intracluster ports listed in the following table. In addition, a number of ports are dynamically assigned to services, which change between runs. The dynamic ports come from the range of ports that are dynamically allocated by Linux (20K+).

Port	Protocol	Service Name	Direction	Source	Dest.	Description
80	TCP	nginx	inbound	All nodes	All nodes	Primary app HTTP port (nginx)
443	TCP	Secure nginx	inbound	All nodes	All nodes	Primary app HTTPS port (nginx)
2100	RPC	Oreo RPC port	bidirectional	All nodes	All nodes	Node daemon RPC
2101	HTTP	Oreo HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Node daemon HTTP

Port	Protocol	Service Name	Direction	Source	Dest.	Description
2181	RPC	Zookeeper servers listen on this port for client connections	bidirectional	All nodes	All nodes	Zookeeper servers listen on this port for client connections
3181	RPC	Zookeeper servers listen on this port for client connections	bidirectional	All nodes	All nodes	Zookeeper servers listen on this port for client connections
4181	RPC	Zookeeper servers listen on this port for client connections	bidirectional	All nodes	All nodes	Zookeeper servers listen on this port for client connections
2200	RPC	Orion master RPC port	bidirectional	All nodes	All nodes	Internal communication with the cluster manager
2201	HTTP	Orion master HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Port used to debug the cluster manager
2205	TCP	Cluster update service TCP port	bidirectional	All nodes	All nodes	Internal communication with the cluster manager
2210	RPC	Cluster stats service RPC port	bidirectional	All nodes	All nodes	Internal communication with the stats collector
2211	HTTP	Cluster stats service HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Port used to debug the stats collector
2230	RPC	Callosum stats collector RPC port	bidirectional	All nodes	All nodes	Internal communication with the BI stats collector

Port	Protocol	Service Name	Direction	Source	Dest.	Description
2231	HTTP	Callosum stats collector HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Port used to debug the BI stats collector
2240	RPC	Alert manager	bidirectional	All nodes	All nodes	Port where alerting service receives alert events
2241	RPC	Alert manager	bidirectional	All nodes	All nodes	Port where alerting service receives alert events
2888	RPC	Ports used by Zookeeper servers for communication between themselves	bidirectional	All nodes	All nodes	Ports used by Zookeeper servers for communication between themselves
3181	RPC	Ports used by Zookeeper servers for communication between themselves	bidirectional	All nodes	All nodes	Ports used by Zookeeper servers for communication between themselves
3888	RPC	Ports used by Zookeeper servers for communication between themselves	bidirectional	All nodes	All nodes	Ports used by Zookeeper servers for communication between themselves
4000	RPC	Falcon worker RPC port	bidirectional	All nodes	All nodes	Port used by data cache for communication between themselves
4001	HTTP	Falcon worker HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Port used to debug the data cache
4002	HTTP	Falcon worker HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Port used to debug the data cache

Port	Protocol	Service Name	Direction	Source	Dest.	Description
4003	RPC	Falcon worker RPC port	bidirectional	All nodes	All nodes	Port used by data cache for communication between themselves
4004	RPC	Falcon worker RPC port	bidirectional	All nodes	All nodes	Port used by data cache for communication between themselves
4021	RPC	Sage metadata service port (exported by Tomcat), Callosum services like meta-data services, medata-dependency service, scheduling service, session-less service, spotiq service	bidirectional	All nodes	All nodes	Port where search service contacts metadata service for metadata
4181	RPC	Ports used by Zookeeper servers for communication between themselves	bidirectional	All nodes	All nodes	Ports used by Zookeeper servers for communication between themselves
4201	HTTP	Sage auto complete server HTTP interface port	bidirectional	Admin IP addresses and all nodes	All nodes	Port used to debug the search service
4231	HTTP	Sage index server HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Port used to debug the search service
4232	RPC	Sage index server metadata subscriber port	bidirectional	All nodes	All nodes	Port used for search service internal communication
4233	RPC	Sage index server RPC port	bidirectional	All nodes	All nodes	Port used for search service internal communication

Port	Protocol	Service Name	Direction	Source	Dest.	Description
4241	HTTP	Sage auto complete server HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Port used to debug the search service
4242	RPC	Sage auto complete server RPC port	bidirectional	All nodes	All nodes	Port used for search service internal com-munication
4243	RPC	Sage auto complete server metadata sub-scriber port	bidirectional	All nodes	All nodes	Port used for search internal communication
4244	RPC	Sage auto complete server metadata sub-scriber port	bidirectional	All nodes	All nodes	Port used for search internal communication
4245	RPC	Sage auto complete server metadata sub-scriber port	bidirectional	All nodes	All nodes	Port used for search internal communication
4243	RPC	Sage auto complete server metadata sub-scriber port	bidirectional	All nodes	All nodes	Port used for search internal communication
4251	RPC	Sage master RPC port	bidirectional	All nodes	All nodes	Port used for search service internal com-munication
4405	RPC	Diamond (graphite) port	bidirectional	All nodes	All nodes	Port used for communication with monitoring service
4406	RPC	Diamond (graphite) port	bidirectional	All nodes	All nodes	Port used for communication with monitoring service
4500	RPC	Trace vault service RPC port	bidirectional	All nodes	All nodes	Trace collec-tion for ThoughtSpot services
4501	HTTP	Trace vault service HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Debug trace collection

Port	Protocol	Service Name	Direction	Source	Dest.	Description
4851	RPC	Graphite manager RPC port	bidirectional	All nodes	All nodes	Communication with graphite manager
4852	HTTP	Graphite manager HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Debug graphite manager
4853	RPC	Elastic search stack (ELK) manager RPC port	bidirectional	All nodes	All nodes	Communication with log search service
4853	HTTP	Elastic search stack (ELK) manager HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Debug log search service
9200	RPC	Elastic search (ELK)	bidirectional	All nodes	All nodes	Communication with log search service
5021	RPC	Callosum services like meta-data services, medata-dependency service, scheduling service, session-less service, spotiq service	bidirectional	All nodes	All nodes	Port where search service contacts meta-data service for metadata
5432	Postgres	Postgres database server port	bidirectional	All nodes	All nodes	Communication with Postgres database
6021	RPC	Callosum services like meta-data services, medata-dependency service, scheduling service, session-less service, spotiq service	bidirectional	All nodes	All nodes	Port where search service contacts meta-data service for metadata
7021	RPC	Callosum services like meta-data services, medata-dependency service, scheduling service, session-less service, spotiq service	bidirectional	All nodes	All nodes	Port where search service contacts meta-data service for metadata

Port	Protocol	Service Name	Direction	Source	Dest.	Description
8020	RPC	HDFS namenode server RPC port	bidirectional	All nodes	All nodes	Distributed file system (DFS) communication with clients
8021	RPC	Callosum services like meta-data services, medata-dependency service, scheduling service, session-less service, spotiq service	bidirectional	All nodes	All nodes	Port where search service contacts metadata service for metadata
8080	HTTP	Tomcat	bidirectional	All nodes	All nodes	BI engine communication with clients
8081	HTTP	Callosum/Tomcat status	bidirectional	All nodes	All nodes	BI engine communication with clients
8787	HTTP	Periscope (UI) service HTTP port	bidirectional	All nodes	All nodes	Administration UI back end
8888	HTTP	HTTP proxy server (tinyproxy)	bidirectional	All nodes	All nodes	Reverse SSH tunnel
11211	Memcached	Memcached server port	bidirectional	All nodes	All nodes	BI engine cache
12345	ODBC	Simba server port	bidirectional	All nodes	All nodes	Port used for ETL (extract, transform, load)
8480	HTTP	HDFS journalnode server HTTP port	bidirectional	All nodes	All nodes	Debug DFS metadata
8485	HTTP	HDFS journalnode server HTTP port	bidirectional	All nodes	All nodes	Debug DFS metadata
50070	HTTP	HDFS namenode server HTTP port	bidirectional	All nodes	All nodes	Debug DFS metadata
50090	HTTP	HDFS secondary namenode server HTTP port	bidirectional	All nodes	All nodes	Debug DFS metadata
50075	HTTP	HDFS datanode server HTTP port	bidirectional	All nodes	All nodes	Debug DFS data

Port	Protocol	Service Name	Direction	Source	Dest.	Description
50010	HTTP	HDFS datanode server HTTP port	bidirectional	All nodes	All nodes	Debug DFS data
50020	HTTP	HDFS datanode server HTTP port	bidirectional	All nodes	All nodes	Debug DFS data
7000	TCP	Cassandra KV store database	bidirectional	All nodes	All nodes	Debug DFS data
7001	TCP	Cassandra	bidirectional	All nodes	All nodes	Debug DFS data
9042	HTTP	Munshi server impression service, Cassandra	bidirectional	All nodes	All nodes	Debug DFS data
9160	TCP	Cassandra	bidirectional	All nodes	All nodes	Debug DFS data
4010	HTTP	Falcon moderator	bidirectional	All nodes	All nodes	Debug DFS data
4011	HTTP	Falcon moderator	bidirectional	All nodes	All nodes	Debug DFS data
20123 - 32768	TCP (dynamic)	Dynamic port in this range used for various services and anciliary services like atlas, caffeine, callhome, callosum, falcon, monitoring, munshi server, nlp, object_search, postgres, sage UBR, spotiq snapshot, timely	All nodes	Services		
5270	TCP	Cluster monitoring service (ELK)	bidirectional	All nodes	All nodes	Services
5271	TCP	Cluster monitoring service (ELK)	bidirectional	All nodes	All nodes	Services
5601	TCP	Kibana UI (ELK)	bidirectional	All nodes	All nodes	Services
6311	TCP	R service	bidirectional	All nodes	All nodes	Services
8008	TCP	Video recorder	bidirectional	All nodes	All nodes	Services

Port	Protocol	Service Name	Direction	Source	Dest.	Description
9090	TCP	Timely	bidirectional	All nodes	All nodes	Services
	ICMPv4	Used for health check of cluster nodes	bidirectional	All nodes	All nodes	Services

Required ports for inbound and outbound cluster access

ThoughtSpot uses static ports for inbound and outbound access to the cluster.

Port	Protocol	Service Name	Direction	Source	Dest.	Description
22	SCP	SSH	bidirectional	ThoughtSpot Support	All nodes	Secure shell access.
80	HTTP	HTTP	bidirectional	ThoughtSpot Support	All nodes	Hypertext Transfer Protocol for website traffic.
443	HTTPS	HTTPS	bidirectional	ThoughtSpot Support	All nodes	Secure HTTP.
12345	TCP	Simba	bidirectional	ThoughtSpot Support	All nodes	Port used by ODBC and JDBC drivers when connecting to ThoughtSpot.
2049	TCP	NFS: In case one needs to mount NFS share on TS node.	bidirectional	ThoughtSpot Support	All nodes	Port used by NFS.
123	UDP	NTP service	bidirectional	ThoughtSpot Support	All nodes	Port used by NTP service.

Port	Protocol	Service Name	Direction	Source	Destination	Description
443	TCP	HTTPS	outbound	All nodes	208.83.110.20	For transferring files to thoughtspot.egnyte.com.

Port	Protocol	Service Name	Direction	Source	Destination	Description
443	TCP	HTTPS	outbound	All nodes	For transferring product usage data to mixpanel cloud.	outbound
443	TCP	HTTPS	outbound	All nodes	je8b47jfif.execute-api.us-east-2.amazon-aws.com s3.us-west-1.amazon-aws.com s3-us-west-1.amazon-aws.com s3.dualstack.us-west-1.amazonaws.com	For transferring monitoring data to InfluxCloud. (Given address will resolve to point to AWS instances).
25 or 587	SMTP	SMTP or Secure SMTP	outbound	All nodes and SMTP relay (provided by customer)	All nodes	Allow outbound access for the IP address of whichever email relay server is in use. This is for sending alerts to ThoughtSpot Support.
389 or 636	TCP	LDAP or LDAPS	outbound	All nodes and LDAP server (provided by customer)	All nodes	Allow outbound access for the IP address of the LDAP server in use.

Required ports for IPMI (Intelligent Platform Management Interface)

ThoughtSpot uses static ports for out-of-band IPMI communications between the cluster and ThoughtSpot support.

Port	Protocol	Service Name	Direction	Source	Dest.	Description
80	HTTP	HTTP	bidirectional	ThoughtSpot Support	All nodes	Hypertext Transfer Protocol for website traffic.
443	TCP	S-HTTP	bidirectional	ThoughtSpot Support	All nodes	IPMI GUI and for HTML5-based IPMI console access.

Port	Protocol	Service Name	Direction	Source	Dest.	Description
623	UDP	Serial-over-LAN	bidirectional	ThoughtSpot Support	All nodes	IPMI GUI and for HTML5-based IPMI console access.

Related information

EC2 Best Practices [See page 0]

Configure load balancing and proxies

Summary: A load balancer is needed in front of a server group in order to direct traffic to individual servers in a way that maximizes efficiency.

Here are some of the best practices and guidelines for a typical implementation with ThoughtSpot. Your experience may differ depending on your environment and preference.

Load balance across ThoughtSpot nodes

The following shows a network architectural diagram which includes a load balancer for ThoughtSpot nodes.

The load balancer is an appliance in your infrastructure that routes traffic automatically to nodes to provide failover. You can also place a load balancer or proxy in front of the ThoughtSpot appliance if you'd like external network users to access the system.

The best way to load balance across all ThoughtSpot nodes in a cluster is to map one domain name (FQDN) to all the IPs in the cluster in a round robin fashion.

For example, if you want to use a DNS server based load balancing, then you can define multiple "A" resource records (RR) for the same name.

Below is an example of how you could set that up

```
thoughtspot.customer.com IN A 69.9.64.11
thoughtspot.customer.com IN A 69.9.64.12
thoughtspot.customer.com IN A 69.9.64.13
thoughtspot.customer.com IN A 69.9.64.14
```

The example indicates that IP addresses for the domain thoughtspot.customer.com are 69.9.64.11, 69.9.64.12, 69.9.64.13, and 69.9.64.14.

Session Affinity

Session Affinity refers to directing requests to the same application server for the time it takes to complete a task.

In order for session affinity to work on ThoughtSpot, HTTPS (an SSL certificate) has to be installed on the load balancer level. If it is installed outside of the load balancer, session affinity may not occur and the ThoughtSpot system will fail.

Web proxies

You can access ThoughtSpot through any standard web proxy server. Web proxies are fairly universal regardless of the application they are proxying. However, ThoughtSpot doesn’t use any new protocols, like SPDY or HTTP/2, which may have a dependency on the proxy. Instead, ThoughtSpot is commonly placed behind a web HTTP/HTTPS proxy.

Additionally, the proxy can round robin across multiple nodes in the ThoughtSpot backend. You can essentially use the web proxy as a load balancer. Therefore, your session will carry over if the proxy round robins between the ThoughtSpot backends as long as the URL doesn’t change.

Customize ThoughtSpot Help

Summary: You customize ThoughtSpot Help to be specific to your data, examples, and documentation.

You can customize the Help for your ThoughtSpot application to tailor it to your organization. Configuring these Help settings sets system-wide defaults for all your users.

When your ThoughtSpot users click the Help icon, they see a list of links.

As an administrator, you can add your own links to this list. This allows you to include documentation specific to your company, such as information about the data available in ThoughtSpot, where to get support internally, or company-specific training.

You can also edit existing links, change icons, or remove items from the help listing altogether.

Customize look and feel

Summary: You can brand the ThoughtSpot application for your company.

You can customize the look and feel of the ThoughtSpot application for your company. Configuring these settings sets system-wide defaults for all your users.

Style customization is enabled by default. You can configure your cluster to disable this functionality. Contact support@thoughtspot.com [See page 0] for information about disabling this feature.

Where to customize styles

A user with administrative rights can view and access the customization on the **Admin** page.

Use the **Style Customization** page to access the configuration settings.

General guidelines for customization

Your changes take effect either immediately or with browser refresh. You can revert your changes by using the **Reset** button which displays when your cursor moves to the right of any setting.

The following table lists the style customizations you can configure.

Setting	Description
Application Logo (Default) & Favicon	Sets a default application and favicon logo. This should be 140 pixels square.
Application Logo (Wide)	This logo should be 440 x 100 pixels.
Chart Visualization Fonts	Set a font for chart labels. You can specify any Web Open Font Format (`WOFF`) file.
Table Visualization Fonts	Set a font for table labels. You can specify any Web Open Font Format (`WOFF`) file.
Embedded Application Background	Set the background for an embedded ThoughtSpot instance. This is only used if you are embedding ThoughtSpot in another application.
Chart Color Palettes	Set the default palette for all charts. To set a value, however over a color value and enter a HEX value or select one from the chart. Individual users can still customize their own chart colors. They can use the **Reset colors** option on a chart to clear their changes.
Footer text	Define a footer to appear with the ThoughtSpot application.
Page title	Sets the title for the browser tab.

How to specify fonts

You can set your systems default fonts by specifying either or both of the **Chart Visualization Fonts** and **Table Visualization Fonts** values. To set this values, your font must be defined in a Web Open Font Format (`WOFF`) file.

Changes to chart and table defaults apply only to charts and features created *after* you configure a value. If you change table fonts, older tables retain their previous fonts. Some settings can be overridden on a per-object level. Users cannot configure their own defaults.

To set a new font:

1. Make sure you have a WOFF file available for your font.

2. Select the label you want to change.

3. Click the **+** button.

   The system displays the **Custom Font** dialog.

4. Click the **Custom Font** field.

   The system displays the file finder.

5. Add the `WOFF` file you want.

   The file appears in the font dialog

6. Click **Save** to change the font.

How to specify the behavior or clickable links in data

There is a system-wide setting which determines what happens when a user clicks on a link within the data. When your data includes URLs, they display as clickable links in ThoughtSpot tables. By default, clicking on a link opens the URL in a separate tab. But there is a system-wide setting that can be changed to open the links within the context in which they appear.

Changing this setting opens the links:

Link type	Opens in
Link in search result table in ThoughtSpot	Same browser tab as ThoughtSpot application
Link in table embedded in an iFrame	Same iFrame that contains the table
Link in full ThoughtSpot application embedded in an iFrame	Same iFrame that contains the application

Slack integration

Working with ThoughtSpot, you can configure your installation to work with Slack. Users can use the integration, called **Spot**, to make queries or view charts. Any users with administrative rights can apply a `spot` sticker to specific objects. Then, the object is available through Slack.

The first time a user messages Spot, it returns a login link for ThoughtSpot. After logging in, the user's Slack and ThoughtSpot accounts are connected. Actions a user makes from Slack are tied to the user's permissions and authorization.

Spot workflow for administration

Here are the high level steps:

1. Work with support@thoughtspot.com [See page 0] to install the Spot Slack bot on your cluster.
2. Log into ThoughtSpot.
3. Label answers, pinboards, and other objects with the spot sticker.
4. Start Spot Bot.
5. Register Spot bot with your company's Slack instance.
6. Register your Spot Slack account to ThoughtSpot.

Related Information

Relevant `tscli` commands are here [See page 0], but these will not work until Spot is enabled by ThoughtSpot Support. Support will work with you to install Spot, and then provide the rest of the workflow to you, including `tscli` command usage.

Load and manage data

The fastest and easiest way to load a new table is by importing it using the Web browser. This is best for one time data loads of small tables which do not have complex relationships to other tables. This method is limited to tables that are under 50 MB (megabytes) in size.

Using ThoughtSpot Loader, you can script recurring loads and work with multi-table schemas.

If your data already exists in another database with the schema you want to use in ThoughtSpot, you can pull the schema and data in using the ODBC or JDBC driver.

These are the methods you can use to load data, along with the benefits of each method:

Method	Description	Benefits
Connect to external databases [See page 504]	Use the ThoughtSpot Embrace to read directly from the external databases.	Easy way to set up and enable the connection between ThoughtSpot and external databases. Users can send live query to the external databases, without having to replicate data again in ThoughtSpot for analysis.
Load data from the ThoughtSpot UI [See page 90]	Use the ThoughtSpot Web interface to upload an Excel or CSV (comma separated values) file from your local machine.	Easy way to do a one-time data load of a small file (under 50MB).End users can upload their own data and explore it quickly.
Import with the ThoughtSpot Loader (tsload) [See page 144]	Use TQL and tsload to load data directly into the back end database that ThoughtSpot uses.	Best way to load large amounts of data or a schema with multiple tables.Can be scripted and used for recurring data loads, such as monthly sales results or daily logs. Can be integrated with an ETL solution for automation.
Use the ODBC/JDBC driver to connect to ThoughtSpot	Use the ODBC or JDBC client with your ETL tool. For information, see the ThoughtSpot Data Integration Guide.	Make use of an established ETL process and tool(s).Connect to ThoughtSpot using third party tools like SSIS. You don't need to define a schema to accept the data load.

If you're uploading data through the Web interface, you can use a native Excel file. If you want to use a CSV (comma separated values) or delimited file, or you are loading using ThoughtSpot Loader, you must create CSV files with the data to be loaded [See page 91] first.

> **Note:** End users will almost always work with worksheets and data they upload.

Related Information

- Load CSV files with the UI [See page 90]
- Append data through the UI [See page 0]
- Schema planning concepts [See page 102]
- Overview of schema building [See page 121]
- Import CSV files with tsload [See page 144]
- How to view a data schema [See page 96]

Configure casing

Summary: You can set the type of case sensitivity you would like to see reflected in the ThoughtSpot display.

Before you load your data, you should consider the type of casing you would like your data to reflect. The case sensitivity for source data strings is preserved in the display. So, the visual display of results is identical to the input case that is loaded.

> ℹ **Note:** The casing will remain lowercase in other parts of the application, such as when you ask a question or filter.

It is important to note that string casings aren't applied globally, but by column. So datasets will have different string casings as long as they're in different columns. Tables that are already compacted will keep their lowercase format. In these cases, to get the specific string case that you want, you would have to truncate related tables and reload them.

To take advantage of case configuration, you need to have ThoughtSpot Support enable it on your cluster for you. In addition, title casing should be disabled for string casing to properly work.

Load CSV files with the UI

Summary: The simplest way to load data is to upload a CSV or Excel file from the ThoughtSpot Web interface.

Loading data through the Web browser is recommended for smaller tables (under 50MB) with simple relationships between them. This method is recommended for small, one time data loads. Using this method, the data schema is created for you automatically.

Any user who belongs to a group that has the privilege **Has administration privileges** or **Can upload user data** can upload their own data from the browser.

Your data should be in a CSV (comma separated values) before you load it. A CSV file is a text file made up of data fields separated by a delimiter and optionally enclosed with an enclosing character. If your data contains multiple tables, you can have a separate CSV for each table.

Formatting the CSV

Your ETL (extract, transform, load) process will typically generate CSV files. You can also create a CSV file from a Microsoft Excel spreadsheet by opening the spreadsheet in Excel, choosing **Save As** and selecting CSV.

A CSV file contains a delimiter that marks the separation between fields in the data. The delimiter is usually comma, but it can be any character. The file also contains fields optionally enclosed with double quotes. Use these guidelines when creating the CSV file:

- If the CSV contains column headers, they must match the column names in the database exactly.
- Often a `|` (pipe) or tab is used as the delimiter, because it may be less likely to occur within the data values.
- When a field contains a double quote, it must be escaped with the character specified in the escape character argument in `tsload`.
- When a field contains the delimiter, the field must be enclosed in double quotes.

ThoughtSpot supports a wide range of date and timestamp formats [See page 666] in the CSV file. Blank values in user uploaded CSV files are interpreted as NULL values. These include the values (case insensitive):

- `NULL`
- `\N`
- `NA`
- `N/A`
- [space]

If you are appending data to an existing schema or table, columns in the CSV file must be in the same order as defined in the target table.

If you are loading a fact table that joins to dimension tables, you must load the fact table first, and then the dimension tables. The joining key must be a single column of unique values in the dimension table. `NULL` values in the fact table cannot be joined.

Create a CSV file

The first step in loading data is to obtain or create one or more CSV files that contain the data to be loaded into ThoughtSpot. CSV is a common format for transferring data between databases. ThoughtSpot requires this format.

Most applications such as Microsoft Excel or Google Sheets can output CSV formatted files. If your source is an Excel spreadsheet or Google Sheet:

1. Save, export, or download the file in CSV format. The exact procedure you use will depend on the source application.
2. Review the file's format before uploading it to ThoughtSpot.

   Your source data may be in another database. If this is the case, your company's ETL (extract, transform, load) process will typically generate CSV files. If your source is another database:

3. Connect to the source database.
4. Extract each table you wish to import into ThoughtSpot as a CSV file.

The column delimiter should be a `,` (comma), `|` (pipe), or tab.

For general information about CSV files and the rules for creating them, see the Comma-separated_values [See page 0] on Wikipedia.

Load the CSV File

Any user who belongs to a group that has the privilege **Has administration privileges** or **Can upload user data** can upload their own data from the browser. To load the CSV or Excel file into ThoughtSpot:

1. Log into ThoughtSpot from a browser.
2. Click **Data**, on the top navigation bar.
3. Click the ellipses icon ••• , in the upper right corner, and select **Upload Data**.

4. Upload the CSV or Excel file by doing one of these options:
   a. Click **Browse your files** and select the file.
   b. Drag and drop the file into the drop area.
5. Answer the question **Are the column names already defined in the file header?**
6. Answer the question **Are the fields separated by?** Click **Next**.

7. Click the column header names to change them to more useful names, if you'd like. Click **Next**.

8. Review the automatically generated data types for each column, and make any changes you want.

   There are four data types: Text, Integer, Decimal, and Date.

9. Click **Import**.

   When an upload is complete, the system reports the results and offers you some further actions.

- Click **Link to Existing Data** if you want to link the data you uploaded to the data in another table or worksheet.
- Click **Search** if you want to begin a new search.
- Click **Auto analyze** if you want to use the SpotIQ feature to find insights in your new data.

Append to an existing table

You can append data to your existing system tables through the ThoughtSpot application, even if the tables were initially loaded using `tsload`. The CSV file must have the same structure as the table it is being loaded into, including number and type of columns, in the same order as the target table.

To append data into ThoughtSpot:

1. Log in to ThoughtSpot from a browser.
2. Click **Data** on the top navigation bar.

3. Click the name of the table you would like to append data to.
4. Click the **Load Data** button.

5. Upload the CSV or Excel file by doing one of these options:
   - Click **Browse your files** and select the file.
   - Drag and drop the file into the drop area.
6. Answer the question **Are the column names already defined in the file header?**.
7. For the question **Do you want to append to the existing data or overwrite it?**, select **Append**.
8. Answer the question **Are the fields separated by?**, and click **Next**.
9. Click **Upload**.
10. Click **Link to existing data** if you want to link the data you uploaded to the data in another

table or worksheet. Or click **Ask a question** if you want to begin a new search.

How to view a data schema

Summary: Use the schema viewer to see tables and worksheets and their relationships.

ThoughtSpot has a ##Schema Viewer** though which you can examine the database schema. It is interactive and configurable, so you can see the level of detail that is relevant to your work.

You must have **Admin** privileges to use the **Schema Viewer**.

Accessing the Schema Viewer for all objects

1. Click the **Data** tab in the top menu.

2. Click the ellipses icon, , and select **View Schema**.

1. When the schema appears, you can see that you can control the visible part of the schema.

Legend	Action
1.	See either *All* (Default), or *Yours*.
2.	See either *All types* (Default), *Worksheets*, *Tables*, or *Views*.
3.	Select artifacts tagged with stickers.

1. The list of tables, worksheets, and imported data on the left changes as you select the various filters. The schema view focus changes in tandem.

2. To center the view panel on a specific table, worksheet, or view, click that object.

   You can also drag the objects around in the viewer to position them better.

Accessing Schema Viewer for a single worksheet, table, or view

You can now see the schema for each object for tables, worksheets, or views.

1. Click the **Data** tab in the top menu.

2. Select from the list of the possible objects:

Legend	Action
1.	See either *All* (Default), or *Yours*.
2.	See either *All types* (Default), *Worksheets*, *Tables*, or *Views*.
3.	Select artifacts tagged with stickers.

3. From the list of objects, select one. Here, we selected the worksheet *West Regional Sales*.

4. At the top of the worksheet, click the **Schema** tab.

5. The join information and the schema for the worksheet appear.

Why use the Schema Viewer

You can use the Schema Viewer to discover the following information:

- What is the relationship between two tables?
- What tables make up this worksheet, and how are they joined?

The schema viewer shows joins between tables, join directionality, and join type.

How the Schema Viewer shows joins

You can use the Schema Viewer to review your schema and ensure that it was modeled using best practices. For example, joins appear in different colors to distinquish their type:

- Red is used for generic relationships
- Green is used for primary key/foreign key joins

When viewing a worksheet, you can also see what joins connect the tables: the inner, left outer, right outer, or full outer joins

A good rule to follow is “Keep it Green”. This means that you can get better results from PK/FK joins rather than from using generic relationships. You should only use generic relationships when the tables being joined have a many-to-many rather than a PK/FK structure. If you find tables that have been joined

using a generic relationship, but could have used a PK/FK join, you should drop the relationship and create a PK/FK join instead. To do this, you need to use the ALTER TABLE…DROP RELATIONSHIP statement in TQL. Then use ALTER TABLE…ADD FOREIGN KEY to create the PK/FK join.

Worksheet view

Worksheets are often based on more than one table. The worksheet schema will show schemas for the tables behind the worksheet, as well as the joins between tables *that were created as a part of the worksheet*.

Click a worksheet, to see it in the Schema Viewer. If the schema view is not showing the schema behind the worksheet, double click the tab on the top right of the worksheet object.

The worksheet view shows the following information:

- All tables in the worksheet, and the relationships between these tables.
- Source columns for all columns of a worksheet.
- Keys and definitions for each relationship, as well as join paths and types.
- Columns that are derived from formulas.
- Correct join paths for newly created chasm trap worksheets. Chasm trap worksheets created prior to ThoughtSpot version 4.4 do not show the correct join paths.

Related Information

- Worksheet joins [See page 343]
- Modify joins within a worksheet [See page 347]
- Change the schema using TQL [See page 134]
- Constraints [See page 108]

Schema planning concepts

Summary: Considerations in creating a schema for the ThoughtSpot Loader.

Before you can load data with ThoughtSpot Loader, you must create a schema to receive it, using the SQL command line interface (TQL).

The TQL syntax is similar to the SQL used in other relational databases, but with some important differences. Use DDL (data definition language) to create the schema into which you can load the data. We recommend combining all DDL statements into a single script for creating the schema.

Before writing your TQL script, you need to understand some basic ThoughtSpot concepts.

About databases and schemas

ThoughtSpot organizes objects in a hierarchical namespace. Databases contain schemas, which contain tables.

ThoughtSpot can contain one or more databases, and each database can have multiple schemas. If you do not specify a schema, the default schema (`falcon_default_schema`) is used automatically. This makes it easier to add tables to the database without the need to explicitly create a schema.

If you do create an additional schema, you must refer to its objects using the syntax `<schemaname>.<objectname>`. If you do not qualify the schema when referencing its objects, the default schema (`falcon_default_schema`) will always be assumed.

By default, ThoughtSpot creates an internal database to host tables corresponding to data that is imported by users from a Web browser.

Review the structure of your data

The schema you create to hold the data needs to be a good fit for your data. First, familiarize yourself with the tables you want to load, and understand their structure. Make note of this information for each table:

- The column names and data types
- Type of table (fact or dimension)
- Primary key column(s)
- The size of the table on disk
- Any other tables it can be joined with (foreign keys)

Here's what you must take into account in your TQL for creating each table, based on these properties:

Table type	Table size	To be joined with	Schema recommendations
Fact	Any	Small dimension table(s)	Sharded. Foreign key references the primary key in the dimension table.
Fact	Any	Large dimension table(s)	Sharded on the same distribution key as the dimension table it will be joined with. Foreign key references the primary key in the dimension table.
Fact	Any	Another fact table	Sharded on the same distribution key as the fact table it will join with. Many-to-many relationship defines how the tables will be joined.
Dimension	under 50MB	Fact table(s)	Replicated (not sharded). Has a primary key.

Table type	Table size	To be joined with	Schema recommendations
Dimension	over 50MB	Fact table(s)	Distributed dimension table, sharded on the same distribution key as the fact table it will be joined with. Primary key must be the same as the distribution key.

Where to go next

- **Data types [See page 105]**

  ThoughtSpot supports the common data types. Compare these with the data types you want to load, and do any necessary conversion ahead of loading the data.

- **Constraints [See page 108]**

  Constraints include primary keys, foreign keys, and relationships. Relationships allow you to create a generic relationship for use when you want to join tables that don't have a primary key/foreign key relationship.

- **Sharding [See page 112]**

  For the best performance, you should split (or shard) very large tables across nodes. If you have a large dimension table, you might choose to co-shard it with the fact table it will be joined with.

- **Chasm traps [See page 118]**

  In a complex schema, you may have a fact table with no relationship to another fact table, except that each contains a foreign key to a shared dimension table. This is known as a chasm trap, and ThoughtSpot can handle it!

Data types

Summary: ThoughtSpot supports the common data types.

Before you import data, compare the data types you want to load with these supported data types. Then, convert your data before loading it. Typically, you would export the data, transform it to meet these type rules, and then load the data. This is known as an extract-transform-load process.

Supported data types

The tables you create to receive the data must have the same number of columns and data types as the data you will be loading. Choose a data type for each column from the list of supported data types:

Data	Supported data types	Details
Character	`VARCHAR(*n*)`	Specify the maximum number of characters, as in `VARCHAR(255)`. The size limit is 1GB for `VARCHAR` values.
Floating point	`DOUBLE` or `FLOAT`	`DOUBLE` is recommended. `DOUBLE` has a range of `1.7E +/- 308` (15 digits).
Boolean	`BOOL`	Can be `true` or `false`.
Integer	`INT` or `BIGINT`	`INT` holds 32 bits. `BIGINT` holds 64 bits. `INT` has a range of `-2,147,483,648` to `2,147,483,647`. `BIGINT` range is `-9,223,372,036,854,775,808` to `9,223,372,036,854,775,807`.
Date or time	`DATE`, `DATETIME`, `TIMESTAMP`, `TIME`	`DATETIME`, `TIMESTAMP`, and `TIME` are stored at the granularity of seconds. `TIMESTAMP` is identical to `DATETIME`, but is included for syntax compatibility.

Warning: There is a 1GB limitation on the number of characters for `VARCHAR`. If you have any `VARCHAR` data that exceeds this limit, the entire load will fail.

Geographical data types

For geographical data types, use VARCHAR. For latitude and longitude, you can use either `VARCHAR` or `DOUBLE`. After loading the data, designate it as a geographical data type when you Edit the system-wide data model [See page 277]. Wherever abbreviations or codes are used, they are the same as what the USPS (United States Postal Service) recognizes.

These data types can be designated as geographical data, which enables them to be visualized using the Geo chart types:

- Countries, for example:
    - United States
        - `long name` : United States
        - `name_sort` : United States of America
        - `abbreviation` : U.S.A.
        - `adm0_a3` : USA
        - `adm0_a3_is` : USA
        - `adm0_a3_us` : USA
        - `admin` : United States of America
        - `brk_a3` : USA
        - `brk_name` : United States
        - `formal_en` : United States of America
        - `iso_a2` : US
        - `iso_a3` : USA
        - `iso_n3` : 840

- `COUNTY` for counties in the United States, for example:
    - santa clara county
    - pike county, ohio
    - pike county, OH

- `STATE_PROVINCE` for states in the United States, for example:
    - `name` : California
    - `US Postal Service abbreviation` : CA

- `LATITUDE` which must be used with `LONGITUDE`, for example:
    - 37.421023
    - 1.282911

- `LONGITUDE` which must be used with `LATITUDE`
    - 122.142103
    - 103.848865

- `ZIP_CODE` for zip codes in the United States
    - `po_name`: MT MEADOWS AREA
    - `ZIP`: “00012”
    - `zip2`: 12

- Other Sub-nation Regions which are administrative regions found in countries other than the United States, for example:
    - bremen
    - normandy
    - west midlands

> ⚠ **Important:** You cannot upload your own custom boundaries.

Constraints

> **Summary:** Constraints allow you to build relationships and join tables.

Constraints include primary keys, foreign keys, and relationships. Relationships allow you to create a generic relationship for use when you want to join tables that don't have a primary key/foreign key relationship.

Primary keys

When a primary key is selected for a table, it impacts data loading behavior. When a new row is added:

- If another row already exists with same primary key, it is updated with the values in the new row.
- If a row with the same primary key does not exist already, the new row is inserted into the table.

This behavior is referred to as "upsert" because it does an `INSERT` or an `UPDATE`, depending on whether a row with the same primary key already exists.

Note that ThoughtSpot does not check for primary key violations across different shards of the table. Therefore, you need to shard the table on the primary key columns if you require this "upsert" behavior.

Foreign key relationships

Foreign key relationships tell ThoughtSpot how two tables can be joined. These relationships are only used for joining the tables, and not for referential integrity constraint checking.

The directionality of primary key - foreign key relationships is important. The foreign key relationship is defined on the fact table and references the primary key(s) in the dimension table. So you can think of the fact table as the source and the dimension table as the target. In the schema viewer, notice that the arrow that represents a PK/FK join points to the dimension table.

If you use primary and foreign keys, when users search the data from the search bar, tables are automatically joined. For example, assume there are two tables:

- revenue, which is a fact table
- region, which is a dimension table

There is a foreign key on the fact table on `regionid` which points to the id in the region dimension table. When a user types in “revenue by region”, the two tables will be joined automatically.

Foreign keys have to match the primary key of the target table they refer to. So if there are multiple columns that make up the primary key in the target table, the foreign key must include all of them, and in the same order.

Generic relationships (many-to-many)

You may have a schema where there is a fact table that you want to join with another fact table. If there isn’t a primary key/foreign key relationship between the tables, you can use many-to-many to enable this. You can do this by using the RELATIONSHIP syntax to add a link between them, that works similarly to the WHERE clause in a SQL join clause.

> **Note:** Using generic relationships is not a best practice. In cases where you have two fact tables you want to join, it is better to find a way to create a bridge table between them, so you have a chasm trap. Look at your two fact tables to see if they share some common data that you could use to create a dimension table between them. For example a date or product dimension could be use to join an inventory fact table with a sales fact table. This is best done in your ETL process, before bringing the data into ThoughtSpot.

> **Note:** A many-to-many implementation does not protect from over counting in some searches. If you plan to use it, make sure your searches don’t include aggregation or count searches that will count one value multiple times, because it satisfies the join condition for multiple rows.

This is a special kind of relationship, that applies to specific data models and use cases. For example, suppose you have a table that shows wholesale purchases of fruits, and another table that shows retail fruit sales made, but no inventory information. In this case, it would be of some use to see the wholesale purchases that led to sales, but you don’t have the data to track a single apple from wholesale purchase through to sale to a customer.

In a many-to-many relationship, the value(s) in a table can be used to join to a second table, using an equality condition (required) and one or more range conditions (optional). These conditions act like the WHERE clause in a SQL JOIN clause. They are applied using AND logic, such that all conditions must be met for a row to be included.

To use a many-to-many relationship, you need to follow a few rules:

- There must be one equality condition defined between the two tables.
- Each table must be sharded on the same key that will be used for the equality condition.
- There can optionally be one or more range conditions defined.

This example shows the TQL statements that create the two fact tables and the relationship between them.

```
TQL> CREATE TABLE "wholesale_buys" (

  "order_number" VARCHAR(255),

  "date_ordered" DATE,

  "expiration_date" DATE,

  "supplier" VARCHAR(255),

  "fruit" VARCHAR(255),

  "quantity" VARCHAR(255),

  "unit_price" DOUBLE

  )  PARTITION BY HASH (96) KEY ("fruit");


TQL> CREATE TABLE "retail_sales" (

  "date_sold" DATE,

  "location" VARCHAR(255),

  "vendor" VARCHAR(255),

  "fruit" VARCHAR(255),

  "quantity" VARCHAR(255),

  "sell_price" DOUBLE

  )  PARTITION BY HASH (96) KEY ("fruit");


TQL> ALTER TABLE "wholesale_buys" ADD RELATIONSHIP WITH "retai
l_sales" AS "wholesale_buys"."fruit" = "retail_sales"."fruit" a
nd ("wholesale_buys"."date_ordered" < "retail_sales"."date_sol
d" and "retail_sales"."date_sold" < "wholesale_buys"."expiratio
n_date");
```

Sharding

Summary: Sharding partitions very large tables into smaller, faster, more easily managed parts called data shards.

ThoughtSpot tables can be replicated or sharded. Replicated tables exist in their entirety, the complete data set, on each node. Sharded tables consist of a single data set divided into multiple tables or shards. The shards have identical schemas but different sets of data.

When to use sharding

By default, ThoughtSpot tables are replicated, you must explicitly shard tables. Sharding your tables impacts the total amount of memory used by the table as well as its performance.

For example, you might shard a large table of sales data. So, you could divide a single sales table into shards each of which contains only the data falling within a single year. These shards are then distributed across several nodes. Requests for sales data are dispersed both by the year and the location of the shard in the node cluster. No single table or node is overloaded, and so the performance of a query and the system load are both improved.

To optimize ThoughtSpot performance, you should *shard* very large fact tables whenever possible. If you have a large dimension table, you might choose to shard it along with the fact table it is joined with. Sharding both the fact and dimension table is known as *co-sharding*.

Table sizes and sharding recommendations

Number of rows per shard	5-10 million
Maximum	10 million rows per shard
Maximum number of shards	~ 80% of CPU cores

Example

Number of rows in table	1.1 billion
CPUS in cluster	256
HASH (128)	~50% of total CPUs
	8.6 million rows per shard

How to shard

Sharding is a type partitioning and is sometimes called *Horizontal partitioning*. The term sharding is particular to situations where data is distributed not only among tables but across nodes in a system. To create a sharded table add the add `PARTITION BY HASH ( )` clause to your `CREATE TABLE` statement.

```
TQL> CREATE TABLE ...
...PARTITION BY HASH (96) KEY ("customer_id");
```

The `HASH` parameter determines the number of shards and the `KEY` parameter the sharding key. The recommended number of shards depends upon the number of nodes in your cluster:

Number of Nodes	Number of Shards
1	32
2	64
3	96
4-12	128
13-24	256
25-36	384

Number of Nodes	Number of Shards
37-48	512
49-60	640
61-72	768

If you omit the `PARTITION BY HASH` statement or if the `HASH` parameter is 1 (one), the table is unsharded. This also means the table physically exists in its entirety on each node.

If you want to use the primary key for sharding, specify that the table is to be partitioned by `HASH` on the primary key, as in this example:

```
TQL> CREATE TABLE "supplier" (
  "s_suppkey" BIGINT,
  "s_name" VARCHAR(255),
  "s_address" VARCHAR(255),
  "s_city" VARCHAR(255),
  "s_phone" VARCHAR(255),
  CONSTRAINT PRIMARY KEY ("s_suppkey")
  )  PARTITION BY HASH (96) KEY ("s_suppkey");
```

The system does not use primary keys as sharding keys by default. If you specify the `PARTITION BY HASH` statement with a `HASH` greater than 1 (one) *but omit the* `KEY` *parameter* ThoughtSpot shards the table randomly. This is not recommended; avoid this by always ensuring you specify the `KEY` parameter with a HASH greater than 1 (one).

How to choose a shard key

When you shard a large table, you select a *shard key* from the table. This key exists in every shard. You can use any data type that is valid for use as the primary key as the shard key. Choosing a shard key plays an important role in the number of shards and the size of any single shard.

A shard key should contain a value that has a good distribution (roughly the number of rows with each value in that column). This value is typically part of the primary key, but it can include other columns. For example:

```
CREATE TABLE "sales_fact"
  ("saleid" int,
  "locationid" int,
  "vendorid" int,
  "quantity" int,
  "sale_amount" double,
  "fruitid" int,
  CONSTRAINT
PRIMARY KEY("saleid,vendorid"))
PARTITION BY HASH(96)
KEY ("saleid");
```

Notice the shard key contains the `saleid` value that is also part of the primary key. When creating a shard key use these guidelines.

- Include one or more values of the table's primary key in the shard key.

  This prevents scenarios where the data with the same primary key ends up in different shards and nodes because the shard key changed.

- If you expect to join two tables that are both sharded, make sure both tables use the same shard key.

  This guideline ensures better join performance. So, for example, if you have two tables and the primary keys are:

  `PRIMARY KEY("saleid,vendorid")` on A `PRIMARY KEY("saleid,custerid")` on B

  You should use `saleid` for your shared key when you shard both table A and B.

- Choose a shard key so that the data is distributed well across the keys.

  For example, suppose the table you want to shard has a primary key made up of `saleid`, `custid` ,and `locationid`. If you have 10K sales but 400 locations, and 2000 customers, you would not want to use the `locationid` in your shard key if 5k sales were concentrated in just 2 locations. The result would be data in fewer shards and degrade your performance. Instead, your shard key may be `custid` ,and `locationid`.

- Choose a shard key that results in a wide variety of keys.

For example, suppose the table you want to shard has a primary key made up of `saleid`, `productid`,and `locationid`. Suppose the table has 10K sales, 40 locations, and 200 products. If the sales are evenly distributed across locations you would not want to use the `locationid` in your shard key. Instead, `saleid` and `productid` would be the better choice as it results in a wider variety of keys.

As mentioned in the previous section, it is possible to simply use the primary key as a shard key. It isn't a good idea to use shard keys outside of the primary key. The reason is that it, with a non-primary shard key, it is possible to get two versions of a record if the shard key for a record changes, but the primary key doesn't. A second version reults because, in the absence of a unique shard key, the system create a secondary record rather than doing a SQL MERGE (`upsert`).

Sharded dimension tables

In a typical schema, you'd have a sharded fact table with foreign keys to small dimension tables. These small dimension tables are replicated in their entirety and distributed on every node. This works best where dimension tables under 50MB in size.

If you have a large dimension table, replicating it and distributing it can impact the performance of your ThoughtSpot System. In this case, you want to shard the dimension tables and distribute it the same way as the fact table it joins to.

When sharding both a fact and its dimension table (known as co-sharding) keep in mind the guidance for creating a shard key. Only shard dimension tables if the dimension table is large (over 50MB) and the join between the fact and dimension tables use the same columns. Specifically, the tables must:

- be related by a primary key and foreign key
- be sharded on the same primary key/foreign key
- have the same number of regions (or shards)

If these requirements are met, ThoughtSpot automatically co-shards the tables for you. Co-sharded tables are always joined on the sharding key. Data skew can develop if a very large proportion of the rows have the same sharding key.

This example shows the `CREATE TABLE` statements that meet the criteria for sharding both a fact table and its dimension table:

```
TQL> CREATE TABLE products_dim (
  "id" int,
  "prod_name" varchar(30),
  "prod_desc" varchar(100),
  PRIMARY KEY ("id")
)
PARTITION BY HASH (96) KEY ("id")
;

TQL> CREATE TABLE retail_fact (
  "trans_id" int,
  "product_id" int,
  "amount" double,
  FOREIGN KEY ("product_id") REFERENCES products_dim ("id")
)
PARTITION BY HASH (96) KEY ("product_id")
;
```

If a dimension table is joined to multiple fact tables, all of the fact tables must be sharded in the same way as the dimension table. Self-joins are not supported.

Joining two sharded fact tables

You can also join two sharded fact tables with different shard keys, this is known as *non co-sharded* tables. It may take a while to join two tables sharded on different keys since a lot of data redistribution is required. Therefore, ThoughtSpot recommends that you use a common shard key for two fact tables.

You are not limited by the column connection or relationship type.

Chasm traps

Summary: A chasm trap occurs when two many-to-one joins converge on a single table.

In a complex schema, you may have a fact table with no relationship to another fact table, except that each contains a foreign key to a shared dimension table. This is known as a chasm trap, and ThoughtSpot can handle it!

Understand how chasm traps occur

A fact table, just as it sounds, stores facts about your business. If you are selling apples, the sales fact table has facts about these apples.

SaleID	AppleTypeID	StoreID	Units Sold
4	55	2	12
8	34	33	3
10	09	09	1

Dimension tables describe the attributes that are interesting to analyze. For example, the apple table might look like this.

AppleTypeID	Color	Name	Use
55	Red	Red Delicious	Snack
34	Green	Granny Smith	Cooking
09	Yellow	Golden	Snack

As you can imagine, in a business you might have several fact tables that access dimension tables. So, an apple business may record waste as well as sales.

TimeID	AppleTypeID	StoreID	Units Wasted
4	55	2	2
8	34	33	43
10	09	09	11

Both the sales and waste tables are facts that reference the apple dimension table.

A chasm trap in a data schema can introduce problems of over counting if you join the two fact tables through their shared dimension table. This diagram shows a typical complex schema with several tables that are related over a chasm trap:

Examples of use cases where a chasm trap could occur when attribution analysis compare campaign data with purchase data, where all they have in common is that both contain a customer identifier that is a foreign key to a customer dimension table. Chasm traps also occur, for example, in cost of sales analysis when wholesale orders data is only related to the retail sales data through a shared products dimension table.

In many databases, joining tables across a chasm trap creates a *Cartesian product* or *cross join*. That is each row from the first fact table is joined to each row from the second table. A Cartesian product causes over counting when computing counts and aggregates. ThoughtSpot protects you from this kind of over counting.

There are still just a few things to look out for when using a schema that contains chasm traps:

- The tables should be joined to the dimension table by an equi-join (a primary key/foreign key relationship). They cannot be joined using a range of values.
- Review the column setting called Attribution Dimension [See page 309]. You may need to change this setting if some of the columns in the shared dimension table should not be used for attribution when combining fact tables.
- Tables that will be joined across a chasm trap do not have to be co-sharded. They will be joined appropriately automatically in the most efficient way.

Chasm trap limitations

Join information in **What am I Looking At?** does not appear for searches on a worksheet containing a chasm trap or on base tables that are related over a chasm trap.

Overview of schema building

Summary: Before you can load data into ThoughtSpot, you must build a database schema to receive it.

You can build a schema by writing a SQL script that creates the objects in your schema. ThoughtSpot provides the ThoughtSpot SQL Command Line (TQL) for creating, viewing, and managing a schema using SQL. Your script can use any statements that are supported in ThoughtSpot SQL Command Line (TQL). The TQL syntax is similar to the SQL used in other relational databases, but with some important differences.

You can use DDL (data definition language) to create the schema into which you can load the data. We recommend placing all your DDL statements into a single script for creating the schema.

Upload the script through the browser

You can upload an your SQL script directly through the browser in the ThoughtSpot application. You can edit the script or add to it right within the browser, too. The steps to build a schema through the browser are:

1. Write a SQL script to create the schema [See page 125]
2. Import a schema (use the SQL editor) [See page 132]

Use TQL on the SQL command line

You can choose to run your SQL script within the Linux shell instead. You can run TQL in interactive command line mode, or you can write a script and use TQL to run it. The SQL syntax in ThoughtSpot is called TQL for ThoughtSpot SQL. The ThoughtSpot SQL Command Line (TQL) runs in an interactive mode. To invoke TQL Log in to the Linux shell using SSH and type `tql`. At the prompt, type `h` or `help` to see a list of supported commands.

The steps to build a schema using TQL include:

1. Connect to the database with the ThoughtSpot SQL Command Line (TQL) [See page 123].
2. Write a SQL script to create the schema [See page 125].

3. Type your SQL commands on the command line, terminating each command with a semicolon (;).

   Commands can span multiple lines. ThoughtSpot supports a limited number of SQL commands, plus some custom SQL extensions. For example, you can specify the number of shards and the distribution key as part of the CREATE TABLE syntax. A full list of supported SQL in TQL is available in the TQL reference [See page 605].

Where to go next

- **Connect with TCL and create a schema [See page 123]**

  Having examined the structure of the data to be loaded and become familiar with the ThoughtSpot SQL Command Line (TQL), you are now ready to create the schema.
- **Write a SQL script to create the schema [See page 125]**

  Using a SQL script to create your schema is a recommended best practice. This makes it easier to adjust the schema definitions and recreate the schema quickly, if needed.
- **Schema creation examples [See page 127]**

  These examples demonstrate the steps involved in creating a schema using the ThoughtSpot SQL Command Line (TQL). After the schema is created, you can load data into it with ThoughtSpot Loader.
- **Upload and run a SQL script) [See page 132]**

  You can run a SQL script to create your database schema through the browser, without having to log in to the shell on the ThoughtSpot instance. You can edit the script and run it directly in the browser to create the schema.

Connect with TQL and create a schema

To perform administrative tasks directly in the database, you will use the ThoughtSpot SQL Command Line (TQL). TQL supports many, but not all, common SQL commands.

Connect with TQL

Before connecting with TQL, you need:

- Access to your ThoughtSpot instance Linux shell from a client machine.
- The administrator OS login.

To connect to TQL:

1. Log in to the Linux shell using SSH.
2. Invoke TQL:

```
$ tql

TQL>
```

3. Enter your SQL command, followed by a semicolon (`;`).

Enter a SQL script

Having examined the structure of the data to be loaded and become familiar with the ThoughtSpot SQL Command Line (TQL), you are now ready to create the schema.

This method is a good way to get familiar with TQL and how to create database objects, but when creating a schema in a production system, you will most likely Write a SQL script to create the schema [See page 125].

To create the schema directly in TQL:

1. Connect to the database with the ThoughtSpot SQL Command Line (TQL) [See page 123].
2. If the database you will be using does not exist, create it now:

```
TQL> CREATE DATABASE my_database;
```

3. Connect to the database:

```
TQL> USE my_database;
```

4. If you wish to use a schema other than the default one, create it now:

```
TQL> CREATE SCHEMA my_schema;
```

5. Issue a `CREATE TABLE` command for each table you will create, using the information in Plan the schema [See page 102].

> ☑ **Tip:** Foreign key declaration within a `CREATE TABLE` will show the table created even if there are problems with the foreign key. Therefore, it is good practice to also issue a separate `ALTER TABLE ADD CONSTRAINT FOREIGN KEY` command.

How to write a SQL script

Summary: Using a SQL script to create your schema is a recommended best practice. This makes it easier to adjust the schema definitions and recreate the schema quickly, if needed.

The schema creation script is a text file that contains all the SQL commands to create your schema. Comments should be enclosed in the comment tags /* and */.

Enclose all object names (schema, table, and column) in double quotes and any column values in single quotes in your scripts. Object names that are also reserved words in SQL, or that contain special characters (any character other than alphanumeric or _), must be surrounded by double quotes. If you see the error message “Error parsing SQL. Check SQL input.”, you should check for object names without double quotes in your script.

If you are working in a schema other than the default schema, object names must be fully qualified, as in `"<schema_name>"."<object_name>"`.

If your schema includes constraints to define relationships between tables (foreign key, or the `RELATIONSHIP` syntax), it is recommended that your script first creates all the tables, and then at the end, creates the relationships between them using the `ADD CONSTRAINT` syntax. This makes it easier to troubleshoot the script and make changes.

If TQL is run using the flag `--allow_unsafe`, your statements will always execute without this warning. Note that when running TQL from a script, you must decide what behavior you want if the script contains changes that affect dependent objects. If you want the script to run even if objects with dependencies are affected, run it using this flag, for example:

```
cat safest_script_ever.sql | tql --allow_unsafe
```

1. Open a new file in a text editor.
2. Type in the command to create the database, if it does not already exist:

```
CREATE database <db_name>;
```

3. Type in the command to specify the database to use:

```
USE database <db_name>;
```

4. Type in the command to create the schema, if you don't want to use the default schema:
5. Type in each of the CREATE TABLE statements, with its column definitions, primary key constraints, and sharding specification (if any).
6. At the end of your script, optionally type in the `ALTER TABLE` statements to add foreign keys to use in joining the tables.
7. Save the file.
8. Run the script using one of these methods:
   - Import a schema (use the SQL editor) [See page 132].
   - Log in to the shell [See page 10], copy your script to your ThoughtSpot instance using scp, and pipe it to TQL:

```
$ cat create-schema.sql | tql
```

Schema creation examples

Summary: Simple examples that illustrate how to use the TQL and the ThoughtSpot Loader.

These examples demonstrate the steps involved in creating a schema using the ThoughtSpot SQL Command Line (TQL). After the schema is created, you can load data into it with ThoughtSpot Loader.

Simple schema creation example

The example creates a database (`tpch`) with two tables (`customer` , `transaction`). The example does not create a schema explicitly. So it will use the default schema (falcon_default_schema).

In this example:

- The table `customer` has a primary key called customer_id. The table `customer_transactions` has a primary key called transaction_id.
- The `customer` table is unsharded.
- The `customer_transactions` table is sharded into 96 shards using the `transaction_id` column.
- Both tables have referential integrity on `customer_id` .

```
$tql

TQL> CREATE DATABASE tpch;

TQL> USE tpch;

TQL> CREATE TABLE customer (
      name  VARCHAR(100),
      address VARCHAR(255),
      zipcode INT,
      customer_id INT,
      CONSTRAINT PRIMARY KEY (customer_id)
      );

TQL> CREATE TABLE customer_transactions (
     transaction_id INT,
     customer_id INT,
     amount DOUBLE,
     transaction_date DATETIME,
     CONSTRAINT PRIMARY KEY (transaction_id),
     CONSTRAINT FOREIGN KEY (customer_id) REFERENCES
     customer(customer_id)
     )  PARTITION BY HASH (96) KEY (transaction_id);
```

More complex schema creation example

The example uses a custom schema called sample_schema to hold the tables. Because of this, every table reference has to be schema qualified.

```
$ tql

TQL> CREATE DATABASE "sample_db";

TQL> USE "sample_db";

TQL> CREATE SCHEMA "sample_schema";

TQL> CREATE TABLE "sample_schema"."customer" (

  "c_custkey" BIGINT,

  "c_name" VARCHAR(255),

  "c_address" VARCHAR(255),

  "c_city" VARCHAR(255),

  "c_nation" VARCHAR(255),

  "c_region" VARCHAR(255),

  "c_phone" VARCHAR(255),

  CONSTRAINT PRIMARY KEY ("c_custkey")

);

TQL> CREATE TABLE "sample_schema"."supplier" (

  "s_suppkey" BIGINT,

  "s_name" VARCHAR(255),

  "s_address" VARCHAR(255),

  "s_city" VARCHAR(255),

  "s_nation" VARCHAR(255),

  "s_region" VARCHAR(255),

  "s_phone" VARCHAR(255),
```

```
  CONSTRAINT PRIMARY KEY ("s_suppkey")

);

TQL> CREATE TABLE "sample_schema"."lineorder" (

  "lo_orderkey" BIGINT,

  "lo_linenumber" BIGINT,

  "lo_custkey" BIGINT,

  "lo_partkey" BIGINT,

  "lo_suppkey" BIGINT,

  "lo_orderdate" DATE,

  "lo_orderpriority" VARCHAR(255),

  "lo_shippriority" VARCHAR(255),

  "lo_quantify" BIGINT,

  "lo_extendprice" BIGINT,

  "lo_ordtotalprice" BIGINT,

  "lo_discount" BIGINT,

  "lo_commitdate" DATE,

  CONSTRAINT PRIMARY KEY ("lo_orderkey","lo_linenumber"),

  CONSTRAINT FOREIGN KEY ("lo_custkey") REFERENCES "sample_sche
ma"."customer" ("c_custkey"),

  CONSTRAINT FOREIGN KEY ("lo_suppkey") REFERENCES "sample_sche
ma"."supplier" ("s_suppkey")

  )  PARTITION BY HASH (96) KEY (lo_orderkey);
```

Upload and run a SQL script

Summary: Importing a schema through the Web browser makes it possible to run your SQL script without needing to have a Linux login.

You can run a SQL script to create your database schema through the browser, without having to log in to the shell on the ThoughtSpot instance. You can edit the script and run it directly in the browser to create the schema. You can use this capability in any of these ways:

- Create the SQL script ahead of time [See page 125], and use the browser to run it.
- Use the editor to type your SQL directly into the browser.
- Use the browser SQL interface as an interactive SQL editor, for example to test an existing script or make changes to an existing schema.

1. Log into ThoughtSpot from a browser.
2. Click **Data**, on the top navigation bar.
3. Click the ellipses icon ••• , and select **Upload schema**.

4. Drag and drop your SQL file into the browser, or choose **Browse Your Files** to locate it.
5. You’re now in the SQL editor. Use it to view your script and make any changes.

6. When ready, run your script by clicking the **Execute** button.
7. If there are any errors, correct them and run the script again.

How to change a schema

Summary: After you've created a schema and loaded data, you may find yourself wishing you'd set things up a little differently. You can make changes to the schema, such as changing the primary key, relationships to other tables, and sharding.

Making changes to a schema after data has been loaded and users have created worksheets or pinboards on the tables requires care, so that you don't lose the relationship between the objects created in ThoughtSpot and the underlying tables. If you follow the procedures here, your tables will retain their relationships to the objects created on top of them.

> ☑ **Tip:** Always take a snapshot of your database before making any schema changes. This snapshot allows you to revert back to the prior state if you make an error, or something doesn't work as you expected after the schema change.

Change the primary key for a table

Use this procedure to change the primary key for a table. But use it with caution, particularly if you are changing to a primary key for which values are not unique.

You can change the primary key of a table without having to `TRUNCATE` it first and reload the data. However, changing the primary key could result in data deletion. This is because of the upsert behavior which is applied when multiple rows have the same primary key. This is very important to understand ahead of time, if you are considering changing to a primary key for which values are not unique.

To change the primary key, first remove any existing primary key, and then define a new one (if any). You do not have to truncate the tables to do this operation beginning in version 3.2. Any dependent objects (pinboards or worksheets) will remain intact.

To change the primary key of a table:

1. Create a manual snapshot [See page 256].
2. Connect to the database with the ThoughtSpot SQL Command Line (TQL) [See page 123].
3. Drop the existing primary key (if any), by issuing a command like this example:

```
TQL> ALTER TABLE "cart"
     DROP CONSTRAINT
     PRIMARY KEY;
```

Dropping a primary key can impact existing worksheets, answers, and pinboards. The system warns you if dropping a primary key impacts other objects. To continue, use the `--allow_unsafe` flag.

4. Add a new primary key, if desired:

```
TQL> ALTER TABLE "cart"
     ADD CONSTRAINT
     PRIMARY KEY ("owner_id");
```

5. Test that any dependent objects (pinboards, worksheets, etc.) are still working correctly.
6. Delete the snapshot you created earlier using the command:

```
tscli snapshot delete <name>
```

Change a relationship between tables

Use this procedure to remove a relationship between tables or define a new one. This operation works for both kinds of relationships: foreign key or generic relationship.

To change a relationship between two tables, first remove any existing relationship, and then define the new relationship (if any). You do not have to truncate the tables to do this operation. Any dependent objects (pinboards or worksheets) will remain intact.

To change the relationship between tables:

1. Create a manual snapshot [See page 256].
2. Connect to the database with the ThoughtSpot SQL Command Line (TQL) [See page 123].
3. Issue the command to drop the existing relationship

Before dropping a relationship TQL checks for and then warns of any dependent objects. To continue with the drop any way, use the `--allow_unsafe` flag. The following examples illustrate several different types of drop operations.

Drop a foreign key by name, if it was given a name when it was defined:

```
TQL> ALTER TABLE
     "sales_fact"
     DROP CONSTRAINT
     "FK_PO_number";
```

Drop a relationship by name, if it was given a name when it was defined:

```
TQL> ALTER TABLE "fruit_dim"
     DROP RELATIONSHIP "REL_dates";
```

Drop the foreign key relationship explicitly, if it doesn't have a name, by referencing the two tables that are joined. This drops all foreign keys between the two tables:

```
TQL> ALTER TABLE "shipments"
     DROP CONSTRAINT
     FOREIGN KEY "orders";
```

Drop all generic relationships between two tables:

```
TQL> ALTER TABLE "wholesale_buys"
     DROP RELATIONSHIP
     WITH "retail_sales";
```

4. Define a new relationship, if you want to, using `ALTER TABLE...ADD CONSTRAINT...`
5. Test that any dependent objects (pinboards, worksheets, etc.) are still working correctly.
6. Delete the snapshot you created earlier using the command:

```
tscli snapshot delete <name>
```

Change sharding on a table

You can change the sharding on a table or remove it altogether (creating a replicated table) using this procedure. This procedure preserves the data within the table.

This procedure reshards a table. This is also called redistributing or repartitioning. You can use this method to reshard a table without losing its data or metadata. This means that worksheets and pinboards built on top of the table will continue to work.

You can use these steps to do any of these operations:

- shard a table that was previously replicated.
- change a replicated table to a sharded table.
- change the number of shards to use for a sharded table.

To change the sharding on a table:

1. Create a manual snapshot [See page 256].
2. Connect to the database with the ThoughtSpot SQL Command Line (TQL) [See page 123].
3. Issue the command to change the sharding using this syntax:

```
TQL> ALTER TABLE <table>
     [SET DIMENSION | SET FACT
     [PARTITION BY HASH
     [(<shards>)]
     [KEY(<column>)]]]
```

For example:

- To make a sharded table into a dimension table (replicated on every node), use:

```
ALTER TABLE "products"
   SET DIMENSION;
```

- To make a dimension table into a sharded (fact) table or change the number of shards, use:

```
ALTER TABLE "sales"
   SET FACT PARTITION BY HASH (96)
   KEY ("productID");
```

4. Test that any dependent objects (pinboards, worksheets, etc.) are still working correctly.
5. Delete the snapshot you created earlier using the command:

```
tscli snapshot delete <name>
```

Convert column data type

Summary: You can convert the data in a column from one data type to another by issuing a TQL command.

There are some details you should be aware of when doing a data type conversion.

Data type conversion behavior

When converting from one data type to another, any values that can not be converted will be set to NULL. If errors occur during data type conversion, the operation is aborted. However, you may choose to force the conversion despite the errors. You can start TQL in allow_unsafe mode to continue with the data conversion, at your own risk, of course! To start TQL in unsafe mode, issue this command:

```
tql --allow_unsafe
```

Multiple columns of a single table can be converted using a single TQL command. The behavior is transactional. So for example, you would issue a command like this example:

```
ALTER TABLE products
   MODIFY COLUMN product_id int,
   MODIFY COLUMN supplier VARCHAR(4);
```

Also note that changing data type has implications on the primary key and sharding enforcement. For example, changing the data type of a column that is part of the sharding key would lead to a redistribution of data. Then imagine that the sharding key column contained the text values `00100`, `0100`, and `100`, which all map to same integer value. If this type of a column is changed from a `VARCHAR` to an `INT`, then it would be subject to the upsert behavior on primary keys. So, in this example, only one of the three rows would be preserved.

Be aware that data type conversion will preserve the data in the underlying database table, but there is no guarantee that any objects built on top of it (worksheets or pinboards) will be preserved. This is because you might make a data type change that makes a chart built on top of the table invalid (for example a growth chart would be invalidated if the date column it depends on were changed to a varchar column).

Supported data type conversions

In general, the data type conversions that make logical sense are supported. But there are a few nuances you should be aware of:

- When you convert from `INT` to `BOOL`, zero is converted to false, and all non-zero values are converted to true.
- When you convert from `BOOL` to `INT`, true gets converted to 1, and false gets converted to 0.
- When you convert from `DOUBLE` to `INT`, the value gets rounded.
- When you convert from `INT` to `DOUBLE`, the value gets rounded.
- When you convert from `DATETIME` to `DATE`, the date part of value is preserved and the time part is dropped.
- When you convert from `DATE` to `DATETIME`, the time gets added as `00:00:00`. The date part of the value is preserved.
- When you convert from `DATETIME` to `TIME`, the time part of the value is preserved.
- Conversion from TIME to `DATETIME` is not supported.

Date and time conversions

Some data type conversion require a format string. These include:

- conversion from `DATE` / `TIME` / `DATETIME`
- conversion to `DATE` / `TIME` / `DATETIME`

For these types of conversions, you can use a special syntax using parsinghint and the date format specifications supported in the strptime library function [See page 0].

For the example, first create a table with a timestamp stored as a `VARCHAR`:

```
CREATE TABLE fruit_sales
   (time_of_sale VARCHAR(32));

INSERT INTO fruit_sales
   VALUES ('2015-12-29 13:52:39');
```

Now, convert the column from a `VARCHAR` to `DATETIME`, using the format `%Y-%m-%d %H:%M:%S`:

```
ALTER TABLE fruit_sales
   MODIFY COLUMN time_of_sale DATETIME
   [parsinghint="%Y-%m-%d %H:%M:%S"]
```

Finally, convert the column back to VARCHAR:

```
ALTER TABLE fruit_sales
   MODIFY COLUMN time_of_sale VARCHAR(32);
```

String to boolean conversions

String to boolean conversions have format strings, too. You can use `parsinghint` as you do for date and time conversions. You can choose among these approaches:

OPTION 1: Specify string values for both true and false. Any non-matching values get converted to null. In this example, “100” gets converted to true, and “0” gets converted to false. “-1” gets converted to null.

```
ALTER TABLE db
   MODIFY COLUMN s bool [parsinghint="100_0"];
```

OPTION 2: Specify a string value for true. Any non-matching value gets converted to false. In this example, “100” gets converted to true, “-1” and “0” get converted to false.

```
ALTER TABLE db
   MODIFY COLUMN s bool [parsinghint="100_"];
```

Option 3: Specify a string value for false. Any non-matching value get converted to true. In this example, “-1” and “100” get converted to true, and “0” gets converted to false.

```
ALTER TABLE db
   MODIFY COLUMN s bool [parsinghint="_0"];
```

String to boolean conversions

When converting from a string to a boolean, you must specify a string for true and false. By default, a string to boolean conversion generates `true` for `true`, `false` for `false`.

```
ALTER TABLE db
   MODIFY COLUMN b varchar(32);
```

But you may override the default strings that get generated by using `parsinghint`, as in this example:

```
ALTER TABLE db
   MODIFY COLUMN b varchar(32) [parsinghint="tr_fa"];
```

Change the Data Type of a Column

When you issue the TQL command to convert a column from one data type to another, the conversion is handled automatically. However, you must ensure that any visualizations built on top of the table display correctly.

You should always take a snapshot of your database before making any schema changes. This will allow you to revert back to the prior state if you make an error, or something doesn’t work as you expected after the schema change.

When changing a data type in an existing table, be aware that answers and pinboards created on top of that table (or worksheets that include it) may change. This is because charts and aggregations depend upon the data type. So for example changing from `INTEGER` to `VARCHAR` could break charts that used the numeric data type `INTEGER` to calculate an average or a total. Because of this, use caution, and check all dependent objects before and after changing the data type, to ensure that they display as intended.

To change the data type of a column:

1. Connect to the database with the ThoughtSpot SQL Command Line (TQL) [See page 123].
2. Issue the command to change the data type using this syntax:

```
TQL> ALTER TABLE <table>
     MODIFY COLUMN <column> <new_data_type>;
```

For example:

```
ALTER TABLE fact100
   MODIFY COLUMN product_id int;
```

Import CSV files with tsload

Summary: The tsload command is a common way to import data from a CSV file.

Use ThoughtSpot Loader (`tsload`) to load data from a CSV text file into an existing table in ThoughtSpot. ThoughtSpot Loader (`tsload`) is a common way to import data. When using `tsload` , you can load larger datasets and make the loading process repeatable through scripting. The `tsload` command accepts flags that enable you to specify column and row separators, date or timestamp formats, null value representations, and similar parameters. Many of these options have defaults that you can override.

Before importing data, you need to Build the schema [See page 121].

To use ThoughtSpot Loader, type the command `tsload` followed by the appropriate flags. You can see the list of the flags it accepts in the ThoughtSpot Loader flag reference [See page 618] or by issuing `tsload –help` .

tsload supports both full and incremental data loads. For incremental loads, an upsert (insert or update) is performed. If an incoming row has the same primary key as an existing row, it updates the existing row with the new values.

You can integrate tsload into your ETL environment for more automated data loads. Most ETL tools provide the ability to write target data into files and support scripted post-transformation actions that can include loading data into ThoughtSpot. This procedure describes manually loading data, but the tsload commands could be saved as a script:

1. Log in to the Linux shell using SSH.
2. Change to the directory where your CSV files are staged.
3. Use the following syntax to invoke `tsload` , specifying the appropriate flags and your data source file:

```
$ tsload --target_database=my_database
       --target_table=my_table --alsologtostderr
       --empty_target --source_file=my_file.csv --v 1
       --field_separator="separator_char"
```

This example imports the CSV file `ssbm_customer.csv` into the table CUSTOMER:

```
$ tsload --target_database=SAMPLE_DB
       --target_table=CUSTOMER --alsologtostderr
       --empty_target --source_file=ssbm_customer.csv
       --v 1 --field_separator "|"
```

4. After the processing begins, you can see messages that indicate the progress, and then two summary messages after the load is complete.

```
 Started processing data row
 Source has 32 data rows, ignored row count 0
 Waiting for rows to commit...(please wait)

 Source summary
 --------------
 Data source:                     ssbm_customer.csv
 Source data format:              csv
 Header row?:                     no
 Tokenizer Options:               escape_char: "" fiel
d_separator: "|" enclosing_char: "\""
 Date format:                     %Y%m%d
 Date time format:                %Y%m%d %H:%M:%S
 Flexible mode?:                  no

 Load summary
 ------------
 Target table:                    CUSTOMER
 Should empty target?:            yes
 Status:                          Successful
 Rows total:                      32
 Rows successfully loaded:        30
 Rows failed to load:             0
 Rows duplicate/omitted:          2
 % of Rows successfully loaded:   93.75 %
 Load Rate (MB/s):                0.00 MB/s
 Load Rate (Rows/s):              3.53 Rows/s
 Start time (Wallclock):          Tue Jan 29 09:09:07
 End time (Wallclock):            Tue Jan 29 09:09:08
 Total load time = 1.13 seconds = 0.02 minutes = 0.00 ho
urs
 Data size = 50 bytes = 0.06 KB = 0.00 MB
```

5. In the load summary, be sure to check the **Rows duplicate/omitted** number. This indicates the number of rows (if any) that were omitted from loading because they did not satisfy the table constraints. A common cause of this would be a duplicate primary key. If any rows were omitted, review your CSV file, make the required adjustments, and then load it again.
6. After correctly loading your file, repeat this process to load data from any additional CSV files.

Loading data from an AWS S3 bucket

If you have data in .csv format stored in an AWS bucket, you can load it directly to ThoughtSpot.

(5.3.1 and later) Assigning S3 read-only role to your EC2 instance

If your cluster is running 5.3.1 or later, you can assign an S3 read-only role to your ThoughtSpot EC2 instance(s) so the instance(s) can access the S3 bucket from which you want to load the data. This eliminates the need to enter the AWS S3 credentials when loading your data. For details, see: Using an IAM Role to Grant Permissions to Applications Running on Amazon EC2 Instances [See page 0] in Amazon's AWS documentation.

> **Note:** If you are using S3 for persistent storage, and assigned the *ec2rolewithfulls3access* IAM role to your instance, you do not need to complete this step.

To load data from an AWS S3 bucket, do the following:

1. Log in to the Linux shell of your AWS instance using SSH.

2. Use the following syntax to invoke `tsload`, specifying the appropriate flags and your data source file:

   ```
   $ tsload --source_file "/aws/default/<my_file_in_aws>"
         --target_database "<my_database_in_ThoughtSpot>"
   --target_table "<my_table_in_the_database_in_ThoughtSpo
   t>"
   ```

   This example imports the CSV file `teams.csv` into the table `teams` in the database `temp`:

   ```
   $ tsload --source_file "/aws/default/teams.csv"
         --target_database "temp" --target_table "teams"
   ```

3. After running the `tsload` command, you are prompted to enter additional AWS S3 information:

   - AWS S3 bucket name

- AWS S3 region
- AWS S3 credentials (accesskey;secret_key)*
- AWS S3 root (prefix for S3 object search path)

Optionally, these four pieces of information can be inserted at the beginning of the command (in step 2), using the following flags:

- `--aws_s3_bucket_name "<bucket name>"`
- `--aws_s3_region_name "<region name>"`
- `--aws_s3_credentials "<credentials>"` *
- `--aws_s3_root "<search path>"`

> ℹ **Note: *AWS S3 credentials is not used in the 5.3.1 release, if an S3 read-only role is assigned to your instance.**

4. After the processing begins, progress messages appear, and then source and load summary messages after the load is complete.

Loading data from a GCP GCS bucket

If you have data in .csv format stored in a GCS bucket, you can load it directly to ThoughtSpot.

Assigning GCS read-only role to your GCP instance

You can assign a read-only role to your ThoughtSpot GCP instance(s) so the instance(s) can access the GCS bucket from which you want to load the data. This eliminates the need to enter the GCP GCS credentials when loading your data. For details, refer to the storage setting detailed in the GCP Create an instance [See page 469] section.

> ℹ **Note:** If you are using GCS for persistent storage, you already assigned the *Set access for each API* scope to your instance, and specified *Full* storage access, so you do not need to complete this step.

Create the database and table

1. Log in to the Linux shell of your GCP instance using SSH.
2. Invoke TQL:

```
$ tql

TQL>
```

3. Create the database:

```
TQL> CREATE DATABASE temp;
```

4. Connect to the database:

```
TQL> USE temp;
```

5. Create the table

```
TQL> create table teams (id int, name VARCHAR(255));
TQL> exit;
```

To load data from an GCP GCS bucket, do the following:

- Use the following syntax to invoke `tsload`, specifying the appropriate flags and your data source file:

```
 $ tsload --source_file /gs/default/team.csv
         --target_database temp
         --target_table teams
         --gs_bucket_name "my_gcs_bucket"
         --has_header_row 2>/dev/null

 $ Header row read successfully
   Source has 2 data rows, has header row, ignored row c
ount 0
   Waiting for rows to commit...(please wait)
   Source summary
   --------------
   Data source:                /gs/default/team.csv
   Source data format          csv
   Header row?                 yes
   Tokenizer Options:          escape_char: "" field_se
parator: "," enclosing_char: "\"" null_value: "(null)" t
railing_field_separator: false
   Date format:                %Y%m%d
   Date time format:           %Y%m%d %H:%M:%S
   Flexible mode?              no
   Load summary
   ------------
   Target table:               teams
   Should empty target?        no
   Status:                     Successful
   Rows total:                 2
   Rows successfully loaded:   2
   Rows failed to load:        0
   % of Rows successfully loaded: 100.00 %
   Load Rate (MB/s):            0.00 MB/s
   Load Rate (Rows/s):          1.13 Rows/s
   Start time (Wallclock):      Wed Oct 30 23:30:11
   End time (Wallclock):        Wed Oct 30 23:30:13
   Total load time = 1.78 seconds = 0.03 minutes = 0.00
hours
   Data size = 19 bytes = 0.02 KB = 0.00 MB
```

Verify the table contents

1. Invoke TQL:

```
$ tql

TQL>
```

2. Connect to the database:

```
TQL> USE temp;
```

3. Show the table data:

```
TQL> select * from teams;
```

Table contents are displayed:

```
id|name
-------
1|sameer
2|sandeep
```

Use a script to load data

Summary: Learn how to load one or files in bulk with a script.

If you need to load data from multiple CSV files, create a script to automate the process. You can also use a similar script to automate recurring data feeds.

Understand how to create a script

The data loading script is a text file that contains all the calls to `tsload` for loading the data from your CSV files.

The example script shown here uses the `cat` command to read the data file, and pipes it to `tsload`. When creating and testing your script, you may wish to replace each `cat` with `cat -10`, to load only the first ten lines of each file. This allows you to quickly run a test of your script. When the test succeeds for all the data files, you can then remove each `-10`, so the complete files will load when you run the script again.

1. Log in to the Linux shell using SSH.
2. Navigate to the directory that contains your CSV files and open a new file in a text editor.
3. Type in the commands to load the data.

   This example shows commands to load three files:

```
 cat Players.csv | tsload
    --target_database baseball --target_table "players"
    --empty_target --field_separator ","
    --max_ignored_rows 10 --bad_records_file bad_record
s.txt
    --has_header_row --alsologtostderr --null_value ""

 cat AllstarFull.csv | tsload
    --target_database baseball --target_table "allstarfu
ll"
    --empty_target --field_separator ","
    --max_ignored_rows 10 --bad_records_file bad_record
s.txt
    --has_header_row --alsologtostderr --null_value ""

 cat Appearances.csv | tsload
    --target_database baseball --target_table "appearanc
es"
    --empty_target --field_separator ","
    --max_ignored_rows 10 --bad_records_file bad_record
s.txt
    --has_header_row --alsologtostderr --null_value ""
```

4. Save the file.
5. Run the script:

```
$ ./load_baseball_data.sh
```

Loading data efficiently

If you have a very large data file that takes a long time to load, you can reduce the load time by splitting it up into multiple files and loading them in parallel using multiple invocations of `tsload`. If the size of any of your data files is greater than 50 million rows, running `tsload` in parallel can reduce the load time significantly.

- Split up your large data file into multiple smaller files.
- Stage the data files in a location accessible to the node on which you run the script. Usually, you can use an NAS mounted file system [See page 56].

- Create a script to load the files in the following example.
- Run the script to load the files. You will make your script multi-threaded by invoking multiple loader threads (between 1 and 5 are recommended).

To optimize the load time even further, determine what the bottleneck is and adjust your process accordingly. If the disk I/O for reading the data files is the bottleneck, you can stage the data files on separate NAS mounted file systems and reference them accordingly in your script. If the CPU on the machine you're using to run the load script is the bottleneck, you can split the load script into the same number of parts as you have nodes in your ThoughtSpot instance, place one script on each node, and run them in parallel. Make sure the other nodes are able to access the data files where they are staged. Running the load script on separate nodes will put the data on all the nodes, just as when you run the script on a single node. Running the script on all the nodes at the same time just lets you take advantage of CPU power of each node for hashing data files.

For example, suppose you have 30 days of data in 30 files, one for the data collected on each day. Each day's data file contains 10 million rows, for a total of 300 million rows of data. You want to load the whole month of data. For this example we'll have 5 loader processes, each one handling 6 days of data. Here is a sample script you could use the load the data files in parallel:

```
/* Script load_script.sh, loads 30 days of data in parallel */

#!/bin/bash

pidlist=""

cat day1.csv day2.csv day3.csv day4.csv day5.csv day6.csv | tsl
oad
--target_database sales --target_table SALES_FACT --max_ignore
d_rows 10
--bad_records_file ./SALES_FACT.bad --date_format %Y-%m-%d
--date_time_format "%Y-%m-%d %H:%M:%S" --source_data_format del
imited
--field_separator "|" --null_value "" --enclosing_character
"\""
--boolean_representation 1_0 &

pidlist="$pidlist $!" &

cat day7.csv day8.csv day9.csv day10.csv day11.csv day12.csv |
tsload
--target_database sales --target_table SALES_FACT --max_ignore
d_rows 10
--bad_records_file ./SALES_FACT.bad --date_format %Y-%m-%d
--date_time_format "%Y-%m-%d %H:%M:%S" --source_data_format del
imited
--field_separator "|" --null_value "" --enclosing_character
"\""
--boolean_representation 1_0 &

pidlist="$pidlist $!" &

cat day13.csv day14.csv day15.csv day16.csv day17.csv day18.cs
v | tsload
--target_database sales --target_table SALES_FACT --max_ignore
d_rows 10
--bad_records_file ./SALES_FACT.bad --date_format %Y-%m-%d
--date_time_format "%Y-%m-%d %H:%M:%S" --source_data_format del
imited
--field_separator "|" --null_value "" --enclosing_character
"\""
--boolean_representation 1_0 &

pidlist="$pidlist $!" &
```

```
cat day19.csv day20.csv day21.csv day22.csv day23.csv day24.cs
v | tsload
--target_database sales --target_table SALES_FACT --max_ignore
d_rows 10
--bad_records_file ./SALES_FACT.bad --date_format %Y-%m-%d
--date_time_format "%Y-%m-%d %H:%M:%S" --source_data_format del
imited
--field_separator "|" --null_value "" --enclosing_character
"\""
--boolean_representation 1_0 &

pidlist="$pidlist $!" &

cat day25.csv day26.csv day27.csv day28.csv day29.csv day30.cs
v | tsload
--target_database sales --target_table SALES_FACT --max_ignore
d_rows 10
--bad_records_file ./SALES_FACT.bad --date_format %Y-%m-%d
--date_time_format "%Y-%m-%d %H:%M:%S" --source_data_format del
imited
--field_separator "|" --null_value "" --enclosing_character
"\""
--boolean_representation 1_0 &

pidlist="$pidlist $!" &

wait $pidlist
```

Call your script using a command like:

```
nohup bash ./load_script.sh > master_log.txt &

tail -f master_log.txt
```

Constructing your script in this way will execute all the commands in the background, and output to the file `master_log.txt`. You can see a running status as the commands in the script execute. After the script completes, you can check the log file for detailed information, such as the number of rows that loaded successfully.

Delete a data source

Summary: How to prepare for and delete a data source using the ThoughtSpot application.

There are two separate ways to delete a data source, through the browser or through TQL [See page 161] describes the dependency checking that occurs when deleting or changing a table using TQL. When you want to delete a data source, you first need to handle any dependent objects that have been built on top of it. You can easily see these dependencies, and choose how to handle them before deleting the data source.

Check data source dependencies

You can see all of the dependencies for any data source (worksheet or table) on the **Data** page.

To view dependent objects for a data source:

1. Click **Data** on the top navigation bar.
2. Click the name of the data source whose dependencies you want to view.
3. Click **Dependents**.

   You will see a list of the names of the dependent objects (worksheets and pinboards), and the columns they use from that data source. You can use this information to determine the impact of changing the structure of the data source or to see how widely it is used.

4. Click a dependent object to modify or delete it.

   If you want to remove the dependency by modifying the dependent object, you must remove all search terms or columns that refer back to the data source you are trying to delete.

5. When all dependencies have been removed, you will be able to go back and delete the data source.

Delete a data source

You can delete data sources from the browser, as long as they were not created by an administrator through `tsload`.

You can delete data sources from the browser if they were created in the browser. These types of data sources include:

- Data imported from the browser.
- Worksheets.

ThoughtSpot checks for dependencies whenever you try to delete a table or worksheet.

1. Click **Data** on the top navigation bar.

2. Check the box next to the name of the data source you want to delete.

3. Click the delete icon.

4. If you attempt to delete a data source with dependent objects, the operation will be blocked.

   You will see a list of dependent objects with links.

5. Click a dependent object to modify or delete it.

   If you want to remove the dependency by modifying the dependent object, you must remove all search terms or columns that refer back to the data source you are trying to delete.

6. When all dependencies have been removed, you will be able to go back and delete the data source.

Delete or change a table in TQL

Summary: You can delete a data source in the web browser or using ThoughtSpot SQL Command Line (TQL).

When you enter a TQL statement, the system warns you of possible dependency consequences with a prompt asking if you'd like to proceed. This should make you feel safe issuing TQL commands, even commands like dropping a table.

If TQL is run using the flag `--allow_unsafe`, your statements will always execute without this warning. Note that when running TQL from a script, you will need to decide what behavior you want if the script contains changes that affect dependent objects. If you want the script to run even if objects with dependencies are affected, run it using this flag, for example:

```
cat safest_script_ever.sql | tql --allow_unsafe
```

If you do not run the script using the flag, it will fail if any of its commands might cause problems with dependent objects.

TQL actions with possible dependency consequences include:

- Change, add, or remove a primary key.
    - When changing or adding a primary key, if the key in question is not unique in the data it may cause deletion of rows, because of upserts occurring when duplicate primary keys are found.
    - When changing or removing a primary key, incoming foreign key relationships will be broken.
- Change a column datatype.
- Add a relationship or foreign key.
- Drop a relationship or foreign key constraint.
- Change or add a sharding key.
- Drop a table, schema, or database.

When issuing one of the previous commands, you will see a warning message similar to this:

```
TQL> ALTER TABLE table1
     DROP CONSTRAINT PRIMARY KEY;

WARNING: This operation will break the Foreign Key relationshi
p “products”
with table “sales”, which will break 34 user-visible visualizat
ions and
2 Worksheets. We recommend taking a snapshot before performing
this operation.
Do you wish to proceed? (yes/no).
```

Understand groups and privileges

Summary: Creating groups and assigning users to them makes privilege management easier.

Before people can log in and use ThoughtSpot, you need to create a username, a password, and a membership in one or more groups for them.

This page describes manual creation of users, groups, and privileges, but you can also manage users through LDAP [See page 46] or SAML. For information on setting up SAML authentication, see the *ThoughtSpot Application Integration Guide*.

Privileges and groups

Privileges determine what kinds of actions users are allowed to do. You assign privileges to groups. Then, you create users and assign them to groups. This is how you grant users access to different capabilities in ThoughtSpot.

Each group includes a set of privileges for its users. The privileges a group has determine the actions that its members are allowed to do. If a user belongs to more than one group, they will have the highest level of the privileges from all the groups they belong to. Plan your groups so that you can use them to assign a common set of privileges to multiple users. Good planning will pay off in ease of administration and a better search experience.

There is a default group called **All**, which includes every user in ThoughtSpot. When you create a new user, they will be added to the **All** group automatically. You cannot delete the **All** group or remove members from it.

You can also have a hierarchy of groups. That is, groups can belong to (that is, be children of) other groups. When using group hierarchies, permissions are inherited from the parent group. So if you're a member of a sub-group, you would automatically have the privileges of the parent group.

List of privileges

Here are the different privileges, and the capabilities they enable:

Privilege	Description
Can administer ThoughtSpot	Can manage Users and Groups and has view and edit access to all data. Users with this privilege can also download a saved answer.
Can upload user data	Can upload their own data from the application's **Data** page using **Actions > Upload data**.
Can download data	Can download data from search results and pinboards.
Can share with all users	Can see the names of and share with users outside of the groups the user belongs to. Members of groups with this privilege can also share with groups marked as **NOT SHAREABLE**.
Can manage data	Can create worksheets and views. Note that to edit a worksheet or a view created by another user, you must have the **Edit** permission on that object, and it must be shared with you.
Can use experimental features	Can access trial and experimental features that ThoughtSpot makes available to early adopters.
Can invoke Custom R Analysis	Can access R scripts to further explore search answers. Includes options to invoke R scripts on visualizations, create and share custom scripts, and share the results of R analysis as answers and pinboards.
Can schedule pinboards	Can create pinboard schedules and edit their own scheduled jobs.
Has SpotIQ privilege	Can use the SpotIQ feature. If this privilege is not enabled for the user, they can still see "Did you know" SpotIQ insights on the ThoughtSpot home page.
Can administer and bypass RLS	Users in groups with this privilege (directly or through group inheritance): • Are exempt from row-level security (RLS) rules. • Can add/edit/delete existing RLS rules. • Can check or uncheck Bypass RLS on a worksheet. Your installation configuration may enable or disable the availability of this privilege. By default, it is enabled. Administrators or groups with the privilege **Can administer ThoughtSpot** can grant this privilege.

Typically, the **ALL** group has a common set of privileges applies such as the **Can upload user data** and/or **Can download data** privileges.

Privileges are additive, meaning that if a user belongs to more than one group, they will have the highest level of privileges from among the groups they are a member of. They are also inherited from the parent, so that a sub-group gets all the same privileges of its parent, all the way up the group hierarchy.

If you add the privilege **Has administration privileges** to a group, note that all users in that group will be able to see all the data in ThoughtSpot. Administrators can see all data sources, and Row level security [See page 221] does not apply to them.

Permissions to see and edit tables, worksheets, and pinboards are set when you share them with users and groups, as described in the topic Data security [See page 203].

The following table shows the intersection of user privilege and ability:

	Create/Edit WS	Create View	Modify Col. Props.[1]	Upload Data	Download Data	Share within Group	Share with all users	Manage RLS rules	CrUD Relationships	Read Relationships	See Hidden Cols	Join with Upload Data	Schema Viewer	Use Scheduler	Use Auto-Analyze
Can administer ThoughtSpot	Y	Y	Y	Y	Y	Y	Y	Y	Y[2]	Y	Y	Y	Y	Y	Y
Can upload user data	N	N	N	Y	N	Y	N	N	Y[3]	Y[4]	N	N	N	N	N
Can download data	N	N	N	N	Y	Y	N	N	N	Y[4]	N	N	N	N	N
Can manage data	Y	Y	Y	Y	N	Y	N	N	Y[4]	Y[4]	Y[5]	Y	N	N	N
Can share with all users	N	N	N	N	N	Y	Y	N	N	Y[4]	N	N	N	N	N
Has SpotIQ privilege	N	N	N	N	N	N	N	N	N	Y[4]	N	N	N	N	Y
Can Administer and By-pass RLS	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N
None	N	N	N	N	N	Y	N	N	N	Y[4]	N	N	N	N	N

	Create/Edit WS	Create View	Modify Col. Props. [1]	Upload Data	Download Data	Share within Group	Share with all users	Manage RLS rules	CrUD Relationships	Read Relationships	See Hidden Cols	Join with Upload Data	Schema Viewer	Use Scheduler	Use Auto-Analyze

Table notes:

1. Applies to non-owners only.
2. Any tables.
3. Author of at least one table in relationship.
4. Only when read permission for columns used in the relationship.
5. With edit permission.

Related information

- Add a group and set security privileges [See page 169]
- Add a user [See page 183]

Create, edit, or delete a group

Summary: ThoughtSpot has intuitive and powerful user group management for assigning privileges, user selection, multi-tier subgroups, default Pinboard assignment, and emailing.

Before adding users, create the groups to which they belong. Each group includes a set of privileges for its users.

Create a group

To create a group and add privileges for the group, follow these steps:

1. Log into ThoughtSpot from a browser.
2. Click the **Admin** icon, on the top navigation bar.

3. In the **Admin** panel, click **User Management**.

4. In the **User Management** section, click **Groups**.

5. Click the **+ Add Group** button on the upper right-hand side of the list of groups.

6. In the **Add a new group** modal, enter the details for the new group:

Field	Description
Group name	Enter a unique name for the group.
Display name	Name of the group as it appears in ThoughtSpot.
Sharing visibility	Indicate whether objects can be shared with this group. When set to **SHAREABLE**, this group is an option in the **Share** dialog.
Description	Optionally, enter a description.
Privileges	Check the privileges [See page 163] you want to grant to the group. If you add the privilege **Has administration privileges** to a group, all users in that group can see all the data in ThoughtSpot. Administrators can always see all data sources, and Row level security [See page 221] does not apply to them.

7. You can also add *Groups [See page 171]* (these would be the subgroups of the group you are editing), *Users [See page 172]*. Also, see *Default Pinboards [See page 0]*.

8. Click **Add** to create the group.

Default Pinboards

You cannot add default Pinboards to a new user group. You must create it first, and then edit it to add default Pinboards. See **Edit a group > Default Pinboards [See page 175]**.

Groups

Follow these steps to assign subgroups to the group:

1. Click the **Groups** tab.
2. Select the groups you want to add in the list by clicking the box next to the group name.
3. You can also use **Search** to find groups by name.

Users

Follow these steps to assign users to the group:

1. Click the **Users** tab.
2. Select the users you want to add in the list by clicking the box next to the user name.
3. You can also use **Search** to find users by name.

Edit a group

After adding a group, you can edit its settings to add or revoke privileges. The new settings apply to all group members.

To edit an existing group, follow these steps:

1. Log into ThoughtSpot from a browser.

2. Click the **Admin** icon, on the top navigation bar.

3. In the **Admin** panel, click **User Management**.

4. In the **User Management** section, click **Groups**.

5. Find the group you want to edit in the list, and click its name.

   If you don't immediately see the name of the group, try searching for it.

6. You can change the *Group name [See page 170]*, *Display name [See page 170]*, *Sharing visibility [See page 0]*, *Description [See page 170]*, and the selected *Privileges [See page 170]*.

   Here, we added the *Can manage data* privilege.

You can also make changes to the *Default Pinboards [See page 175]*, *Groups [See page 176]* (these would be the subgroups of the group you are editing), *Users [See page 177]*, or *Email [See page 178]*.

7. Make your changes, and click **Update**.

Default Pinboards

To assgin default pinboards to groups, follow these steps:

1. Create a group, or choose an existing group. Note its name.

2. In the **Pinboards** interface, find the correct Pinboards, and share them with this group. See Share a Pinboard [See page 213].

3. Open the Group for editing. See Edit a group [See page 173].

4. Click the **Default Pinboards** tab.

5. From the list of shared Pinboards, select 1-3 default Pinboards in the list by clicking the box next to the Pinboard name.

6. You can also use **Search** to find Pinboards by name.

7. Click **Update** to save changes.

Groups

When editing a group, keep in mind that only subgroups or possible subgroups appear in the list of groups. The **No Groups in Group** only indicates there are no children in this group's hierarchy. Do not underestimate the importance of the parent(s) of the group, because each group inherits the privileges of each of its parent groups.

Follow these steps to change subgroups of the group:

1. Click the **Groups** tab.

2. Select the groups you want to add in the list by clicking the box next to the group name.

3. You can also use **Search** to find groups by name.

4. Deselect the groups you want to remove from the list by clearing the box next to the group name.

5. Click **Update** to save changes.

Users

Follow these steps to change the users of the group:

1. Click the **Users** tab.
2. Select the users you want to add in the list by clicking the box next to the user name.
3. You can also use **Search** to find users by name.
4. Deselect the users you want to remove from the list by clearing the box next to the user name.
5. Click **Update** to save changes.

Email

You can configure groups so that users receive a *welcome email* that introduces them to ThoughtSpot, and initiates the onboarding process.

Follow these steps to configure group-wide emails:

1. Click the **Email** tab.
2. Under **Resend welcome email**, select either either *All users* or *New users*.
3. Enter optional text for the email. Here, we added “Welcome!”
4. To send the email immediately, click **Send**.
5. To test the email, click “Test welcome email”
6. Click **Update** to save changes.

Deleting groups

To delete existing groups, follow these steps:

1. Log into ThoughtSpot from a browser.

2. Click the **Admin** icon, on the top navigation bar.

3. In the **Admin** panel, click **User Management**.

4. In the **User Management** section, click **Groups**.

5. Select the groups you plan to delete by clicking the box next to the group name.

   If you don't immediately see the name of the group, try searching for it.

6. Click **Delete**.

List group members

When browsing through users or subgroups, you can often see only a limited list. To check for other users, search for the name of a specific user or subgroup.

Add multiple users to a group

To add multiple users to a group, you must be on the **Users** interface. Follow these steps:

1. Log into ThoughtSpot from a browser.

2. Click the **Admin** icon, on the top navigation bar.

3. In the **Admin** panel, click **User Management**.

4. In the **User Management** section, click **Users**.

5. Select the names of users you plan to add to groups by clicking the box next to the user name.

   If you don't immediately see the user name, try searching for it.

6. Click the **Add Users to Groups** button on the top of the list of users.

7. In the **Add Users to Groups** interface, select the groups by clicking the box next to the group name.

8. Click **Add**.

Create, edit, or delete a user

Summary: For each unique person who accesses ThoughtSpot, you must create a user account. When you create a user manually in ThoughtSpot, you continue to manage that user in ThoughtSpot.

You can create users directly in ThoughtSpot, or import users and user groups through LDAP or similar protocols.

For users who have access through LDAP, the LDAP installation manages all user information.

Create a user in ThoughtSpot

This procedure demonstrates how to create a user manually. When you create a user, you can specify the *username [See page 185]*, *display name [See page 185]*, *sharing visibility [See page 185]*, the *password [See page 185]*, *email [See page 185]*, whether they get a *welcome email [See page 185]* and its *text [See page 0]*, and assign *group [See page 186]* memberships. The user inherits privileges and permissions directly from the group assignments.

Note that all users automatically belong to the group **All**.

To create a new user and assign that user to groups, follow these steps:

1. Log into ThoughtSpot from a browser.
2. Click the **Admin** icon, on the top navigation bar.

3. In the **Admin** panel, click **User Management**.

4. In the **User Management** section, click **Users**.

5. Click the **+ Add User** button on the upper right-hand side of the list of users.

6. In the **Add a new user** interface, enter the details for the new user:

Field	Required?	Description

Username	Yes	A login name for the user. Usernames must be unique and lowercase. If you are using Active Directory to authenticate users, and your LDAP configuration requires users to be created manually (they are not created automatically in ThoughtSpot upon authentication), the username you specify has to be domain-qualified (`user–name@ldap.thoughtspot.com`), and you must enter a dummy password.
Display name	Yes	A unique name for the user (usually their first and last name).
Sharing visibility	Yes	Indicate whether objects can be shared with this user. When set to **SHAREABLE**, this user is an option in the **Share** dialog.
Change password	Yes	A password.
Confirm password	Yes	Enter the password again.
Email	Yes	The user's email address. ThoughtSpot uses this for notification when another user shares something with them, for onboarding, for the **Ask an Export** feature, and others. Note that during cluster configuration, the domain is specified. ThoughtSpot does not accept emails outside this domain.
Send a welcome email	No	When checked, this option ensures that the new user receives a welcome email.
Email message text	No	Enter text of the optional welcome email here.

Groups	Recommended	Select the groups for the user. Note that if you add the user to a group that has the privilege **Has administration privileges**, they can see all the data in ThoughtSpot. When you create a new user, the groups they belong to define the following attributes for the user: • **Privileges**: the actions they can perform, defined when you Add a group and set security privileges [See page 169]. • **Permissions**: the data they can access and view, defined when you consider Data security [See page 203]. Administrators can see all data sources, and Row level security [See page 221] does not apply to them.

7. Click **Add** to create the user.

Note that this process of identifying the user's needs contributes to a robust onboarding process. See Onboarding users [See page 0].

Edit an existing user

As an administrator, you can edit a user account, and change the user's groups. You can also help users by resetting their password, and evaluating their onboarding experience to ensure they receive the best possible introduction to relevant information in ThoughtSpot.

To edit an existing user, follow these steps:

1. Log into ThoughtSpot from a browser.

2. Click the **Admin** icon, on the top navigation bar.

3. In the **Admin** panel, click **User Management**.

4. In the **User Management** section, click **Users**.

5. Click the username in the list to open the **Edit User** interface.

   If you don't immediately see the username you plan to edit, try searching for it.

6. In the **Edit User** interface, edit the basic user information.

   You can change the *username* *[See page 185]*, *display name* *[See page 185]*, *sharing visibility* *[See page 185]*, *passwords* *[See page 185]*, and *user's email* *[See page 185]*.

You can also Preview onboarding [See page 188], and make changes to the *Groups [See page 189]* assigned to the user.

7. Click **Update**.

Preview onboarding

You can click **Preview onboarding** to evaluate this user's first experience with ThoughtSpot. After previewing the user's default data source and Pinboards, you may choose to change the **Group [See page 0]** assignments.

Groups

Follow these steps to change the user's groups:

1. Click the **Groups** tab.
2. Select the groups you want to add in the list by clicking the box next to the group name.
3. You can also use **Search** to find groups by name.
4. Deselect the groups you want to remove from the list by clearing the box next to the group name.
5. Click **Update** to save changes.

Delete users

To delete users, follow these steps:

1. Log into ThoughtSpot from a browser.
2. Click the **Admin** icon, on the top navigation bar.

3. In the **Admin** panel, click **User Management**.

4. In the **User Management** section, click **Users**.

5. Select the users you plan to delete by clicking the box next to the username.

   If you don't immediately see the username you plan to delete, try searching for it.

6. Click **Delete**.

Allow users to sign up

You can allow people in your organization to sign up for ThoughtSpot by clicking a button on the sign-in page.

You do this by providing them with the ThoughtSpot sign-up URL. When they go to this URL, they see the sign-up button.

When a person clicks the sign-up button, they go to a sign-up page that you've already set up outside of ThoughtSpot. This can be any page you want to use for registering new users.

Create the ThoughtSpot sign-up URL

Follow this syntax for creating the URL:

```
https://<your-thoughtspot-URL>/?signUpEnabled=true&signUpButtonLink=https://<your-sign-up-page-URL>
```

Example:

```
https://thoughtspot.mycompany.com/?signUpEnabled=true&signUpButtonLink=https://signup.mycompany.com/thoughtspot
```

Customize the sign-up button text

By default, the button text is ‘Sign up’, but you have the option to change it. To do that, you add the ’signUpButtonText’ parameter to the URL and include the custom text you want. If the text contains spaces, you must replace each space with a percent sign and the number 20 (%20). For example, if you wanted the button text to be ‘Request Account’, you would use ‘Request%20Account’.

Follow this syntax for creating the URL with custom button text:

```
https://<your-thoughtspot-URL>/?signUpEnabled=true&signUpButtonText=<your-sign-up-button-text>&signUpButtonLink=https://<your-sign-up-page-URL>
```

Example:

```
https://thoughtspot.mycompany.com/?signUpEnabled=true&signUpButtonText=Request%20Account&signUpButtonLink=https
thoughtspot
```

Display the sign-up button by default

If you want to display the sign-up button on the ThoughtSpot sign-in page without using the special sign-up URL, contact ThoughtSpot support.

Overview of security features

There are several aspects of security, including access and permissions, data security and privacy, and security from an IT perspective.

- System Security [See page 195] refers to audit logs and security policies.
- Data Security [See page 203] refers to which users can see which data in the ThoughtSpot application, and includes:
    - Users and Groups [See page 163]
    - Privileges [See page 163]
    - Table and columns sharing [See page 209]
    - Row level security [See page 221]
    - Worksheet sharing [See page 211]
    - Pinboard sharing [See page 213]
- Network Security refers to ports for external traffic and traffic within the cluster. Some ports must remain open for handling network requests from outside the ThoughtSpot instance. To see a list of network ports that must remain open to outside traffic, and for inter-cluster communication, review the information in Network ports [See page 66].

System security tools and processes

Summary: System security refers to audit logs and security policies.

ThoughtSpot includes a number of management tools, monitoring applications, and automated processes to support system security. System security includes managing access and privileges, audit logs, security policies, and Linux OS installed package updates.

Audit logs

There are several ways you can view audit log information in ThoughtSpot. You can see recent events in the Control Center or view more detailed audit logs using tscli. Administrators can view audit logs of configuration changes users have made to ThoughtSpot in these ways:

- Monitor events from the Control Center [See page 404].
- Generate audit log reports through the `tscli` command.

You can access an audit log of cluster events through tscli. You can also access information on cluster updates, configurations, data loading and metadata events.

Use the `tscli event list` command to return an audit list of events from the cluster. The syntax is:

```
tscli event list
   [--include <all|config|notification>]
   [--since <hours,minutes,days>
   | --from <yyyymmdd-HH:MM>
   --to <yyyymmdd-HH:MM>]
   [--detail]
   [--summary_contains
   <'string1'| 'string2' ...>]
   [--detail_contains
   <'string1'| 'string2' ...>]
   [--attributes
   <key1='value1'|
   key2='value2' ...>]
```

Optional parameters are:

Parameter	Description
`--include`	Specifies the type of events to include, and can be `all` , `config` , or `notification` .
`--detail`	Returns the events in a detail format rather than a tabular summary, which is the default.
`--summary_contains <'string1' | 'string2' ...>`	Specifies a string to check for in the event summary. Enclose strings in single quotes, and separate multiple strings with &pipe;. Events that match all specified strings will be returned.
`--detail_contains <'string1'| 'string2' ...>`	Specifies a string to check for in the detail. Enclose strings in single quotes, and separate multiple strings with `|` (pipe symbol). Events that match all specified strings will be returned.
`--attributes <key1='value1' &pipe; key2='value2' ...>`	Specifies attributes to match as key=value pairs. Separate multiple attributes with `|` (pipe symbol). Events that match all specified key/value pairs will be returned. Put single quotes around the value(s).

And a time window made up of either:

- `--since <hours,minutes,days>` is a time in the past for where the event audit begins, ending at the present time. Specify a human readable duration string, e.g. 4h (4 hours), 30m (30 minutes), 1d (1 day).

Or both:

- `--from <yyyymmdd-HH:MM>` is a timestamp for where to begin the event audit. It must be of the form: yyyymmdd-HH:MM.
- `--to <yyyymmdd-HH:MM>` is a timestamp for where to end the event audit. It must be of the form: yyyymmdd-HH:MM.

To get audit logs:

1. Log in to the Linux shell using SSH.
2. Issue the `tscli event list` command, with the desired parameters, for example:

```
$ tscli event list
   --include config
   --since 24 hours
```

Security policies

Security policies are the principles and processes ThoughtSpot uses in development to ensure a product that conforms to security standards. Security policies ensure a secure product with each release. When a release is in development, each build is tested using Qualys Network Security and Vulnerability Management Suite. Issues and vulnerabilities are fixed proactively, based on the results.

The ThoughtSpot Engineering and ThoughtSpot Support teams are notified of Common Vulnerabilities and Exposures (CVEs), so they can patch OS packages proactively as well. You can view installed packages along with their version numbers at any time, in order to see if you require an update to ThoughtSpot.

Whenever a CVE is identified, and an OS package needs to be updated, the next patch release will include the patch or update. You can view installed Linux packages at any time, along with the version numbers of the installed packages.

Third-party security software for security, governance, and monitoring of ThoughtSpot

You can install supported third-party security and monitoring software [See page 198] on a ThoughtSpot cluster.

About third-party security and monitoring software

Summary: You can install third-party software for security, governance, and monitoring of ThoughtSpot.

In addition to the ThoughtSpot monitoring and security features, some companies require specific additional third-party software to comply with their internal IT policies. This allows them to support all of their systems with a common set of security and management tools.

For example, you may wish to accomplish some security and monitoring tasks with your own third-party software. These tasks include things like pushing alerts, events, forensics, audit trails, insights, etc. from ThoughtSpot to your own local monitoring systems.

Supported third-party software

ThoughtSpot supports installation of the following third-party software on the ThoughtSpot instance:

- Qualys Qualys is a widely used technical vulnerabilities and security compliance scanning tool. For more information about Qualys, see the Qualys documentation [See page 0].
- SNMP (Simple Network Management Protocol) SNMP is an industry standard protocol used for monitoring network traffic and alert events.
- Splunk You can install Splunk rsyslog and use it to forward ThoughtSpot logs to Splunk. For more information about Splunk, see the Splunk documentation [See page 0].

Install third-party software

For details on how to install third-party software, see: Installing third-party security and monitoring software [See page 201]

What is not supported

When installing and configuring third-party software on a ThoughtSpot cluster, follow the following guidelines to avoid interfering with cluster operations:

- Avoid making any direct changes to any files outside of the /home directory.
- Do not remove existing SSH keys or authorized keys from /home/admin/.ssh
- Excessive resource usage, e.g. CPU, disk, memory, processes, etc.
- Killing any system or ThoughtSpot services, or causing node reboots.

Do not change any system wide configuration which may affect ThoughtSpot, such as:

- Network, e.g. IP addresses, DNS resolution
- Storage, e.g. removing existing mount points, removing drives
- Security, e.g. selinux

Qualys

Qualys is supported for scanning of ThoughtSpot clusters for security vulnerabilities.

SNMP Traps

ThoughtSpot has a built-in alerting service that can also be used to send SNMP traps. Many third-party monitoring systems share the common standard of using SNMP traps, and you can take advantage of those capabilities with ThoughtSpot.

ThoughtSpot supports SNMP for read only. So for example, you can read the IP address of the cluster, but not change it using SNMP.

See the Alert code reference [See page 701] for details.

Splunk rsyslog

ThoughtSpot monitoring and alerting logs are written to standard locations in the file system. This allows you to use rsyslog to collect them and send them to Splunk.

Here are some links to help you learn where various logs are written in ThoughtSpot:

- Monitoring logs [See page 379]
- Audit logs [See page 195]
- Alert code reference [See page 701]

Installing third-party security and monitoring software

Summary: You can install third-party software for security, governance, and monitoring of ThoughtSpot.

This procedure shows how to install supported third-party security and monitoring software [See page 198] on a ThoughtSpot cluster:

To install third-party software

1. Log in to the Linux shell using SSH.

2. Issue the `tscli ansible checkout` command, specifying a temporary directory, for example:

```
$ tscli ansible checkout
Checking out playbooks successfully in /tmp/111895937.
```

3. Switch to the temporary directory that was created.

```
$ cd /tmp/111895937
```

4. In the temporary directory, save or edit the playbooks and modules.
   If you want to create a global ordering between playbooks, name them in alphabetical order, e.g. 10.first.yml, 20.second.yml, etc. You can also specify the order line by line in order.txt within the same directory.

5. Commit your changes. This command will validate the playbook first, and then apply it.

   Use the –local flag if you want to commit the change only to local storage on the local node. Otherwise, push it will go to centralized storage, and your changes will apply to all nodes in the cluster.

- To apply your changes globally to all nodes in the cluster, issue the command:

```
$ tscli ansible commit
```

- To apply your changes on the local node only, issue the command:

```
$ tscli ansible commit --local
```

This commits your changes. If there is a problem with the configuration. you will see an error message in standard output.

Data security

Summary: Data security refers to which users can see which data in the ThoughtSpot application.

Sharing and security privileges govern what data a user can access and what they can do with the data. Admins can use privileges to regulate access to information and provide a personalized user experience.

Users, groups, and privileges

Data security applies to users and groups. Users can be managed manually [See page 163] or through LDAP [See page 46]. Each user can have membership in one or more groups. Admins can make security settings that determine what users are allowed to do in ThoughtSpot. These settings are applied at the group level.

The following table shows the intersection of user privilege and ability:

	Create/Edit WS	Create View	Modify Col. Props.[1]	Upload Data	Download Data	Share within Group	Share with all users	Manage RLS rules	CrUD Relationships	Read Relationships	See Hidden Cols	Join with Upload Data	Schema Viewer	Use Scheduler	Use Auto-Analyze
Can administer ThoughtSpot	Y	Y	Y	Y	Y	Y	Y	Y	Y[2]	Y	Y	Y	Y	Y	Y
Can upload user data	N	N	N	Y	N	Y	N	N	Y[3]	Y[4]	N	N	N	N	N
Can download data	N	N	N	N	Y	Y	N	N	N	Y[4]	N	N	N	N	N
Can manage data	Y	Y	Y	Y	N	Y	N	N	Y[4]	Y[4]	Y[5]	Y	N	N	N
Can share with all users	N	N	N	N	N	Y	Y	N	N	Y[4]	N	N	N	N	N
Has SpotIQ privilege	N	N	N	N	N	N	N	N	N	Y[4]	N	N	N	N	Y
Can Administer and By-pass RLS	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N
None	N	N	N	N	N	Y	N	N	N	Y[4]	N	N	N	N	N

	Create/Edit WS	Create View	Modify Col. Props. [1]	Upload Data	Download Data	Share within Group	Share with all users	Manage RLS rules	CrUD Relationships	Read Relationships	See Hidden Cols	Join with Upload Data	Schema Viewer	Use Scheduler	Use Auto-Analyze

Table notes:

1. Applies to non-owners only.
2. Any tables.
3. Author of at least one table in relationship.
4. Only when read permission for columns used in the relationship.
5. With edit permission.

Security model for sharing objects

You can share with groups and with individual users. Sharing of tables can be defined at the table, column, or row level. This provides flexibility in modeling your data security policy. Security and sharing settings apply to several different types of objects, each of which has its own security default settings and rules.

Object type	Description	Default security model
Tables	The source data tables that have been loaded using ThoughtSpot Loader.	Administrator users have access to source tables. They can share a table with other users or groups. See [Share tables and columns](share-source-tables.html#)
Columns	The columns in the source data tables that have been loaded using ThoughtSpot Loader.	Administrator users have access to columns in the source tables. They can share selected columns with other users or groups. See [Share tables and columns](share-source-tables.html#)
Rows	The rows in the source data tables that have been loaded using ThoughtSpot Loader.	All rows in the source tables are shared with all users by default.
Imported data	Data that was imported using a Web browser.	Only the user who imported the data (and any user with administrator privileges) has access to it by default. They can share a table (or selected columns) with other users or groups. See [Share tables and columns](share-source-tables.html#)
Worksheets	A worksheet created using a Web browser.	Only the creator of the worksheet (and any user with administrator privileges) has access to it by default. They can share a worksheet with other users or groups. See [Share worksheets](share-worksheets.html)
Pinboards	A pinboard of saved search results.	Anyone who can view a pinboard can share it. See [Share a pinboard](share-pinboards.html)

Understanding SHAREABLE

When you share an object, only the users and groups that have **SHAREABLE** set for the **Sharing visibility** option appear on the dialog.

Only users in the **Administrators** group or users with **Admin** privileges can share with groups marked as **NOT SHAREABLE**. Members of a group with **Can share with all users** authorization can also share with groups marked as **NOT SHAREABLE**.

Users in groups marked **NOT SHAREABLE** cannot share objects among themselves. In multi-tenant scenarios, admins can create groups that bring together portions of two non-share groups so that they can share. For example, the members of group C can share even if they belong to other groups that cannot.

Row level security

ThoughtSpot includes robust row level security, which allows you to filter all objects users see based on conditions you set at the level of row values in base data tables.

You may find it useful to create groups for RLS. To prevent these groups from appearing in the **Share** dialog, create a **NOT SHAREABLE** group with a single user and an RLS group with another single user (1-to-1).

Related information

- Revoke access (unshare) [See page 218]
- Row level security [See page 221]

Share tables and columns

Summary: You can share an entire table, or only some of its columns.

By default, when data is loaded using the ThoughtSpot Loader, ODBC, or JDBC, it is only visible to administrators. Data imported from a Web browser is visible to administrators and the user who uploaded it. Administrators and owners can share **Can View** or **Can Edit** privileges on tables with other users, who can further share them with others.

Permissive or strict sharing

Use caution when sharing tables, because any objects created from them will have dependencies on the tables and their underlying structure. Objects created from tables can include worksheets, answers, and pinboards. This means that if a user wants to drop or modify a table, any object that depends upon it must be edited or removed first, to remove the dependency.

For this reason, it is a best practice to only grant the **Edit** permission on tables to a small number of users. If you want to prevent shares from also revealing the columns regardless of where it appears (worksheets, answers, and pinboards), you can ask ThoughtSpot Customer Support to enable a stricter behavior.

How to share

Share a table or imported data by following these steps:

1. Click **Data** in the top navigation bar.

2. Click **Tables**.

3. Select one or more tables to share, and click the **Share** icon.

4. Select **Entire Table** or **Specific Columns**.

5. If you selected **Specific Columns**, select the column to share.

6. Click **+** and select the users and groups with whom you want to share.

7. Configure the level of access by selecting from the dropdown list. You can select:
   - **Can View** to provide read-only access. This enables viewing the table data and defining worksheets on the table.
   - **Can Edit** to allow modification. This enables renaming, modifying, or deleting the entire table and adding or removing its columns.

8. Click **Add and Save**.

9. Click **Done**.

Share worksheets

Summary: You can share worksheets with users or with groups. Sharing a worksheet allows users to select it as a data source and search it.

When you share a worksheet, all of its columns are shared. Sharing a worksheet does not share the underlying tables. If you want to share the underlying tables, see Share tables and columns [See page 209]. A worksheet can be shared by the owner of the worksheet, or by an administrator. Users can start searching a worksheet as soon as the worksheet is shared with them.

To share a pinboard:

1. Click **Data** on the top navigation bar and choose **Worksheets**.

2. Select one or more worksheets to share, and click the **Share** icon.

3. Click **+ Add users or groups** and select users or groups that you want to share with.

4. Configure the level of access by selecting from the dropdown list. You can select:
    - **Can View** to provide read-only access. Enables viewing the worksheet and searching on it.
    - **Can Edit** to allow modification. Enables renaming, modifying filters, or deleting the worksheet and adding or removing its columns. To add columns to a worksheet a user needs access to the underlying table.

5. Click **Add and Save**.
6. Click **Done**.

Share a pinboard

> **Summary:** Whenever you view a pinboard you have the option of sharing it with others.

When you share a pinboard what you are really sharing is a live link to the pinboard, when you click **Share with…**. So whenever someone else views it, they will see the most recently saved version with the most recent data. You do not have to be an administrator or the owner to share saved pinboards. Any user can share them, based on the access levels the user has.

To share a pinboard:

1. Configure it to look as it must appear when shared.
2. From within a pinboard, click the ellipses icon , and select **Share**.

Alternatively, select the pinboard you want to share from the list of pinboards and click **Share**. (The profile picture or avatar for the owner of each pinboard is shown in the list.)

3. Click the plus (+) at the bottom of the Share dialog, and select users or groups with whom you want to share.

4. Configure the level of access by selecting from the dropdown next to each user or group. Available options are based on your own access level. For example, if you have only **View** access, you will not have an option to share as **Edit**. You can select:

    - **Can View** to provide read-only access. If the person doesn't have access to the underlying data, they can only view a shared pinboard. If they change anything on the pinboard, their changes are not saved. In order to persist the changes, the user would need to make a copy of the modified pinboard.

- **Can Edit** to allow modification. Enables renaming or deleting the shared pinboard. If a person with edit privileges modifies a shared pinboard, their changes will be saved to it.

5. Click **Add** to save your changes, then click **Done**.

Security for SpotIQ functions

SpotIQ is a feature in ThoughtSpot that automatically generates insights into system data. The feature works on all the data in your system, search queries, saved answers, and pinboards. Because SpotIQ uses the same data security model as other data in the system, there is no need to build a new schema or security model to support it.

SpotIQ automatically generates insights into data when a user requests them. These requests can be run immediately or users can schedule regular analysis. SpotIQ requests for insights can put additional load on your system depending on the amount of data being analyzed and how frequently users make use of it.

Users are required to have the **Has Spot IQ privilege** to use this feature. Users without this privilege cannot view insights unless the users that generated the insights add them to a pinboard and share the pinboard with others.

You may want to restrict access to a subset of your users or even to a subset of your managers. To restrict access to this feature:

1. Create a group called **SpotIQUsers**.
2. Grant this group **Has Spot IQ privilege**.
3. Enable users to this group.

Revoke access (unshare)

You may need to revoke access to an object (table, worksheet, or pinboard) that you have previously shared. Unsharing an object is very similar to sharing it.

To unshare one or more objects:

1. Go to the area where the object(s) you want to unshare is located. From the top menu bar:
    - If the object is a table or worksheet, click **Data**.
    - If the object is a pinboard, click **Pinboards**.
    - If the object is an answer, click **Answers**.
2. Find the object(s) in the list, and check the corresponding box(es).
3. Click the **Share** icon.

4. Click the **X** next to the users and groups that you want to remove from sharing.

5. Click **Save**, and then click **Done**.

Share
Who has access to pinboard "ThoughtSPORT Overview"
V
victoria
victoria
Can View
ThoughtSPORT
ThoughtSPORT
Can View
basic
basic
Can View
N
nathan
Nathan
Can Edit
vicky
vicky
Can View
+
DONE

About row level security (RLS)

Summary: Using row level security, you can restrict data that appears in search results and pinboards by group.

Row level security (RLS) allows you to restrict a group's access to table row data. You do this by creating a *rule* that associates a filter with a group. When a group member searches, views an answer, or otherwise works with data, ThoughtSpot evaluates the rules and prevents the display of the restricted data. Users see only the data they are permitted to see.

How does RLS impact user interactions?

The security rules apply to objects shared with users individually or through groups they are a member of. The rules restrict the visible data when users:

- view a table
- view a worksheet derived from the table
- search for data in the worksheet or table
- view answers from restricted data &endash; either that they've created or that were shared with them
- interact with pinboards from restricted data &endash; either that they've created or that were shared with them

Search suggestions also fall under row-level security. If a user would not have access to the row data, then values from the row do not appear in **Search** suggestions.

Why use RLS?

RLS allows you to set up flexible rules that are self-maintaining. An RLS configuration can handle thousands of groups. There are several reasons you might want to use row level security:

Reason	Example

Hide sensitive data from groups who should not see it.	In a report with customer details, hide potential customers (those who have not yet completed their purchase) from everyone except the sales group.
Filter tables to reduce their size, so that only the relevant data is visible.	Reduce the number of rows that appear in a very large table of baseball players, so that players who are no longer active are not shown except to historians.
Enable creation of a single pinboard or visualization, which can display different data depending on the group who is accessing it.	Create one sales pinboard that shows only the sales in the region of the person who views it. This effectively creates a personalized pinboard, depending on the viewer's region.

Related information

- To continue learning about RLS, see How rule-based RLS works [See page 223].

- **Search** suggestions relies on compile indices to present suggestions to users from your data.

  See Manage suggestion indexing [See page 295] to learn how to configure suggestions.

How rule-based RLS works

Summary: Use rule-based RLS to restrict a group's access to data. Users see only accessible row data.

Row level security works at the group level and is configured on tables. A table's RLS rules also apply to any objects with data from that table. So, searches, answers, worksheets, and pinboards that rely on a table's data fall under RLS rules.

Worksheet queries and RLS

You cannot set RLS rules on worksheets, only on tables. However, administrators can disable RLS on worksheets that are derived from tables with RLS rules. After RLS rules are disabled, users with access to the worksheet can see all its data.

By default, worksheet queries only take into account RLS rules on tables whose columns appear in the query. Other related tables that may underly the worksheet are ignored. This means that not all RLS rules on underlying tables are applied when a user queries a worksheet.

You can configure a stricter application of RLS rules to take into account RLS rules from all the tables underlying the worksheet. This is recommended if you have key dimension tables that worksheets rely on but that are not necessarily regularly accessed through query. To do this, contact ThoughtSpot Customer Support.

Privileges that allow users to set, or be exempt from, RLS

Users in the **Administrators** group or with the **Has administration privilege** have full access to everything in the system. As a result:

- Row level security does not apply to them.
- They can create, edit, and delete RLS rules.
- They can also disable RLS rules on individual worksheets.

If your installation has enabled the **Can Administer and Bypass RLS** privilege, administrators can also grant **Can Administer and Bypass RLS** to groups. Members of groups with **Can Administer and Bypass RLS**:

- Are exempt from row-level security (RLS) rules.
- Can add/edit/delete existing RLS rules.
- Can check or uncheck Bypass RLS on a worksheet.

This behavior is true regardless of whether the privilege is from a direct group membership or indirect (through a group hierarchy).

Examples of RLS rules

An RLS rule evaluates against two system variables:

Function	Description	Examples
`ts_groups`	Returns a list of all the groups the current logged in user belongs to. For any row, if the expression evaluates to true for any of the groups, the user can see that row.	`ts_groups = 'east'`
`ts_username`	Returns the user with the matching neame.	`ts_username != 'mark'`

ThoughtSpot filters a table's rows by evaluating a rule against the authenticated user.

A rule is an expression that returns a boolean, `TRUE` or `FALSE`. If the rule evaluates to `TRUE`, a user can see that row. If the rule evaluates to `FALSE` for the user, then the user cannot view the data and instead they see the message `No data to display`.

Rule expression can be implicit or explicit. And rules may or may not contain logic. A simple implicit RLS rule has the format:

`COLUMN_FILTER` **= ts_groups**

An example of an explicit rule that contains logic would be:

```
if ( COLUMN_FILTER ) then true else false
```

Rules can also reference tables other than the table you are securing.

Consider a simple RLS rule example. Your company has `vendor-purchase` table such as:

DATE	VENDOR	AMOUNT
12/11/39..	zendesk	116.00
12/11/39..	getquik com ca	289.70
12/11/39..	ikea	113.91
12/11/39..	costco	274.43
12/11/39..	waiters wheels pa	66.52
12/11/39..	waiters whee	76.49
12/11/39..	chipotle	175.33

You want to give your vendors the ability to see trends in company purchases. You give vendor personnel access to ThoughtSpot *and* add them to self-titled vendor groups. So, all users from the Starbucks vendor are in the `Starbucks` group and all users from `round table` are in the `Round Table` group. Then, you set a **Row security** on the `vendor-purchase` table as follows:

```
VENDOR = ts_groups
```

Only users in `Starbucks` group see `starbucks` data and so forth. Rules ignore case inconsistencies and spaces are evaluated so `round table` in the data matches the `Round table` group but not a group named `RoundTable` .

Rules can be simple or they can incorporate logic such as `if/then` rules. For example, vendors should see their own data but your accounts payable group needs to see all the vendor data:

```
VENDOR = ts_groups or 'Accounts Payable' = ts_groups
```

This rule continues to work as you add data from new vendor or team members to `Accounts Payable` . In this way, a well-written rule is *self maintaining*, meaning you don't have to revisit the rule as your system changes.

You can also create rules that reference tables other than the table you are securing. For example, if you have a `sales` table and `store` dimension table, you can use attributes from the `store` table to secure the `sales` table.

Multiple rules and multiple group membership

You can define multiple rules on table. In this case, ThoughtSpot treats the rules as additive. That is, they are applied using an `OR` operator. If any of the rules evaluate to `true` for a user on a row, that row's data is visible.

If a user is a member of multiple groups, the user can see all the rows that are visible to all of their groups. The most permissive policy is used.

Members of groups with **Can Administer and Bypass RLS** are exempt from row-level security (RLS) rules. This is true regardless of whether the group membership is direct or indirect (through a group hierarchy).

Best practices for using Rule-Based Row Level Security

Use these best practices for Rule-Based Row Level Security:

- Use **Share** as the first level of data access.

  Non-administrative users and groups have no way to access any data without first having it shared with them. So, only share what you need.

  When you share, share worksheets. This is a general best practice. Worksheets simplify the data environment for end users; they only need to choose among a few sources, rather than many tables. Also, one worksheet can also combine data from several tables.

- Set row level security wherever you want to keep data secure.

  It is always a possible that a particular search only includes data from a single table, and a user will see something they shouldn't. So, protect your data by setting row level security wherever you want to keep data secure.

- Explicitly grant access for users that should see all rows.

  As soon as you define a rule on a table for one group, you prevent access by all others outside of that group hierarchy. Subsequent rules should specifically add groups that need access.

- Keep in mind that multiple rules on a table are additive with `or`.

  If you are concerned with security, start with very limited access. Then, expand the access as needed.

- Keep rules simple.

  Complex rules can impact the system performance. So, err on the side of simple rules rather than complex rules with a lot of logic.

Related information

- To learn the procedure you follow for setting a rule, Set RLS rules [See page 228]

- For a list of operators and functions you can use to build RLS rules see Row level security rules reference [See page 670].

- For information on bypassing rules on a worksheet, see Change inclusion, join, or RLS for a worksheet [See page 341].

Set row level security rules

Summary: Explains the process for setting RLS rules.

When rule-based row level security (RLS) is set, it prevents users from seeing data they shouldn't in tables and the objects derived from them. You must have administrative rights on ThoughtSpot to set RLS rules.

Before you create a rule, make sure you have read How rule-based RLS works [See page 223].

Create a rule on a table

You can set RLS rules *only* on tables. To set up rule-based row level security, do the following:

1. Click **Data**, and double-click a table.
2. Click **Row security**.
3. Click **+ Add row security**.

The system displays the Rule Builder.

You define row level security by creating an expression that gets evaluated for every row and group combination. This powerful feature can be used with up to thousands of groups.

4. Open the Rule Builder.
5. Give your rule a name.
6. Enter an expression for your rule.

The rule gets evaluated against an authenticated user for every row and group combination. If the rule evaluates to true, the user can't see that row's data. Use the variable **ts_groups** to refer to the group name.

You can see a list of available operators by clicking on **Rule Assistant**.

As you type, ThoughtSpot suggests formula syntax, variables, and column names. If you can't remember the exact column name or variable you want to use, the suggestions can help.

When your expression is valid, a green indicator appears at the bottom of the Rule Builder.

7. Click **Save**.

The rule you created is listed in the rules. You can edit the rule or add more rules by clicking **+ Add**.

Test your rule with restricted and unrestricted users

To test your rule, log in as users in different groups. Search within the table for data both that you test user can and can't access. Make sure your test users are seeing the appropriate rows.

Related information

- Administrators can bypass the RLS rules set on a the table at the worksheet level. See how to "Change inclusion, join, or RLS for a worksheet [See page 341] in this documentation for more information.

- For a list of operators and functions you can use to build RLS rules see Row level security rules reference [See page 670].

ThoughtSpot Lifecycle

Summary: This topic covers security processes for the entire lifecycle of a ThoughtSpot deployment from development, release, installation, upgrades, to software patching.

Overview

A ThoughtSpot deployment consists of the following high level software systems:

- Operating System (OS) and software packages installed on the OS
- Third-party software
- ThoughtSpot application services (binaries and configuration)

Operating System

All ThoughtSpot physical appliances, virtual machines (VMs) and public cloud images come pre-installed with CentOS 7. The CentOS [See page 0] distribution of Linux is owned by RedHat [See page 0] and closely tracks versions of RedHat Enterprise Linux (RHEL).

ThoughtSpot uses the minimal install of CentOS 7 with the addition of a few software packages (e.g. Python) needed for ThoughtSpot operations. The most notable change to the installation is to the Linux kernel, which is sourced from the current long term stable kernel version instead of the default included in CentOS 7 (kernel-lt package). To list all the installed packages, see Checking Package Versions [See page 234].

Third-Party Software (Middleware)

Third party software used includes Java, Boost C++ libraries, Google protocol buffers, etc. These are software components necessary for operation of the ThoughtSpot application. ThoughtSpot only uses software licensed for distribution.

Development and Release Process

ThoughtSpot releases its software as a tarball containing all the ThoughtSpot application (binaries and configuration), third-party software, and an operating system image. Installation or update using this release tarball on appliances, VMs, or cloud instances updates each of these components.

Operating System

Building the operating system image including software packages is a multi-step process:

1. Begin with the set of packages in the base OS image and our added packages.
2. Configure all installation to only use official public RedHat repositories.
3. For each package, install the current stable version including any security patches.
4. Bring up the image on all supported platforms for stability and performance testing along with the ThoughtSpot application stack. Success criteria: no OS impact on stability or performance.
5. Scan the Operating System and ThoughtSpot application stack using Qualys scans with additional modules enabled: Vulnerability Management, Web App Scanning.
6. Review all vulnerabilities found. Success criteria is zero severity 4+ vulnerabilities.
7. Assuming all above testing and exit criteria are met, the OS image is considered qualified.

Third-Party Software

Third-party software is periodically sourced from the upstream distribution of each software component. Unlike OS and ThoughtSpot application, this changes less frequently and on an as needed basis, when any new security vulnerability or stability issue is discovered in the library. The list of all third-party software as well as licensing details are here.

ThoughtSpot Application

ThoughtSpot follows industry standard best practices for writing robust software. Every code change is reviewed by at least one engineer. Our engineering team consists of senior engineers from Enterprise software and web companies.

ThoughtSpot uses a small number of proven programming languages powering some of the largest enterprises in the world. ThoughtSpot tracks stability, performance, and reliability of our software and services aggressively. The ThoughtSpot platform is trusted by dozens of global F2000 organizations.

Protection of Source Code

Source code is private and not shared publicly, e.g. all distribution to customers is in binary or minified format to discourage reverse engineering.

Automated Tools

We use automated tools and infrastructure like Jenkins, Kubernetes, AWS, partnering with the teams behind these systems so as to adopt best practices. For example, all our automation runs through Jenkins, which is managed by CloudBees (the company behind Jenkins) using an enterprise license with regular security patching, and so on. We upgrade our automation tools regularly.

Independent Testing

Independent testing is done outside of the product team by pre sales and post sales before promoting to production. Some areas are tested by third party testers.

Security Hardening

Starting 4.5.1.5, we have also taken specific steps to incorporate most of CIS standard recommendations towards hardening.

Installation and Upgrade Process

ThoughtSpot is installed or updated from a release tarball which contains the ThoughtSpot application (binaries and configuration), third-party software, and Operating System image.

Operating System Image Installation

Installing ThoughtSpot on any node (VM, cloud instance, appliance) automatically updates the operating system and required packages on the node. No Internet or repository access is required for this, the update is applied directly from the release tarball.

Specifically, all nodes running ThoughtSpot are required to have two root partitions on their boot drive of which one of them is booted from at any given time. During installation or update, the Operating System image contained in the release tarball is copied into the second currently-unused root partition and the system switches to it through a reboot.

Checking OS Package Versions

The following command run from any ThoughtSpot node will indicate versions of all installed packages:

```
rpm -qa
```

Upgrades

ThoughtSpot patches the Operating System at the time of upgrades. The exact same process used during installation is also applied during upgrades. The previous OS image on a node gets replaced by the new image carried in the release tarball.

Only some releases may patch the Operating System, not all. Typically, all major and minor releases (e.g. 4.4, 4.5, 4.5.1, 5.0) upgrade OS patches, whereas only some patch releases (e.g. 4.4.1.4) contain OS patches.

Distributed Clusters and Failure Handling

On distributed clusters, individual nodes receive the OS image from the release tarball individually. Initially, the new image is deployed on a single node only. When that node is deemed healthy following the update and a rich set of tests, the image is made available to remaining nodes in the cluster.

If a node fails to patch, then ThoughtSpot support will modify the upgrade workflow to either retry the patching or skip and exclude the node.

Third-Party Software

Installation or upgrade of ThoughtSpot deployments automatically upgrades all third-party software to the version included in the release tarball.

Security Scanning and Patching Process

The ThoughtSpot Security team continuously scans security bulletins for new vulnerabilities discovered in included OS packages (e.g., Linux Kernel, libc) and third party software (e.g., Java). Additionally, weekly scans are done for all release branches using Qualys with the following additional modules enabled: Vulnerability Management, Web App Scanning. The security scans discover vulnerabilities at all layers: OS, third-party software, as well as ThoughtSpot application binaries and configuration. Additionally, ThoughtSpot periodically scans all source code for third-party software as well as ThoughtSpot's proprietary code base for vulnerabilities or unsafe usage using SourceClear.

After a critical new vulnerability is found (severity 4 or 5), ThoughtSpot includes the corresponding patch in the next patch release for all supported release branches. Consult ThoughtSpot documentation or support to find out if you are on an active or supported release branch.

After a new patch release with a critical security vulnerability is available, customers are encouraged to upgrade their deployment quickly.

Latency

We recommend customers to wait for the next regular release for receiving security patches. However, should a critical vulnerability be discovered in the interim, ThoughtSpot can push out a new patch release containing the required patches, if available upstream.

ThoughtSpot targets a three week or less cadence for generating patch releases for all supported release branches. Timeline for the new release and patching depends on availability of the patch upstream (e.g., not all vulnerabilities in Linux are immediately fixed) and qualification (ThoughtSpot qualifies each build on each supported cloud and on-prem platform). If a fix is unavailable upstream at the moment, customers and ThoughtSpot support can work together to identify potential workarounds.

Storage Security

Encryption at Rest

- On-prem: Not supported yet
- Cloud: Supported on AWS [See page 0], GCP [See page 0], Azure [See page 0]

Secure Erase

Current erase guide [See page 0]

Encryption of data in transit within a cluster

Overview

ThoughtSpot supports encryption of data in transit within a cluster (traffic flowing between multiple nodes in a cluster). Encryption in transit within the cluster is primarily needed for cloud deployments of ThoughtSpot. This is accomplished using IPSec.

IPSec operates in two modes: tunnel mode and transport mode. ThoughtSpot recommends using **transport mode** to set up IPSec for encrypting in-transit data.

Note: While IPSec provides additional security, it also reduces network bandwidth between nodes.

ThoughtSpot supports IPSec encryption using strongSwan (an open-source IPSec-based VPN solution for Linux and other UNIX based operating systems).

Summary:

1. Use IPSec in Transport mode for host-to-host IPSec communication.
2. Use the strongSwan package.
3. Use AES-GCM for ESP protocol (since it provides authenticated encryption and provides better network bandwidth).

Note: IPSec is supported in ThoughtSpot software versions starting from 4.5.1.4

Deployment

1. **Enabling IPSec**: Run following command on any ThoughtSpot node after cluster has been configured and right built has been deployed: `# tscli ipsec enable`.
2. **Disabling IPSec**: Run following command on any node of the cluster: `# tscli ipsec disable`.
3. **Checking status of IPSec**: Run following command on any node of the cluster `# tscli ipsec status`.
4. IPSec configuration and settings are NOT persistent across cluster backup/restore and will have to be re-enabled.

5. **Adding a node in an IPSec enabled cluster**: IPSec settings are automatically configured across nodes as a nodes get added to a ThoughtSpot cluster that has IPSec enabled.
6. **Removing a node in IPSec enabled cluster**: IPSec settings are not impacted when nodes get removed from a ThoughtSpot cluster that has IPSec enabled. If any failure occurs during node removal, IPSec would need to re-enabled by using the following command: `tscli ipsec enable`.
7. **Manually enable/disable IPSec on a single node(if any IPSec related failure occurs on that node)**:

   Manually enable and start strongSwan:

   `sudo systemctl enable strongswan`

   `sudo systemctl start strongswan`

   Manually stop and disable strongSwan:

   `sudo systemctl stop strongswan`

   `sudo systemctl disable strongswan`

   Note: You can check `/etc/strongswan/ipsec.conf` and `/etc/strongswan/ipsec.secrets` on each node to verify that they are consistent across all nodes.

Firewall configuration

The following ports must be open between nodes to allow IPSec encryption:

- UDP port 500 (for IKE)
- UDP port 4500 (for IPSec over IDP)
- IP Protocol 50 (ESP)

System administration

System administration includes applying upgrades, backing up and restoring the cluster, snapshotting, and adding or removing nodes.

Administration tools

Use these tools to perform administrative actions:

- tscli [See page 621]: an administrative command line interface.
- tsload [See page 618]: a command for loading data directly into the database.
- TQL [See page 605]: a command line SQL interface to interact with databases.

Send logs when reporting problems

You can generate a log bundle which you can then send to ThoughtSpot Support or you can send logs direct to your administrator.

Generate log bundle

Before you can send a log bundle to ThoughtSpot Support, you must Connect to the ThoughtSpot Support file server [See page 60]. This is a one-time setup operation.

To generate a log bundle:

1. Log in to the Linux shell using SSH.
2. Issue the command to generate the log bundle:

```
tscli callhome generate-bundle
   --d <directory> --since <num_of_daysd>
```

   Note: Don't forget to include `d` after your specified number of days. For example, `30d`.

3. Change directories to the directory where you wrote the log bundle.
4. Issue the command to send the log bundle to ThoughtSpot Support:

```
tscli fileserver upload
   --file_name <file>
   --server_dir_path <path>
```

Send a log to the administrator

Alternately, you can easily send log files directly to your administrator with a single click. When ThoughtSpot encounters a problem, a red bar displays in the browser with an error message. You can use the **Report Problem** option to complete this task.

Click **Report Problem** in the bottom right corner of the error message.

The logs will be sent to your administrator as an email attachment from your email account. Your administrator then has the option to followup with ThoughtSpot, if necessary.

Set up recording for Replay Search

Summary: You can use the recording to create training for your users on how to search your own data.

Recording a search replay requires administrator privileges and a Firefox browser. You must override some of your browser security settings in order to use the ThoughtSpot application to make the recording. This is a one time setup operation. If you do not wish to do this, you can replay the search and record it using QuickTime, Camtasia, or another screen cam recording tool.

To record a search replay using ThoughtSpot:

1. While viewing a chart or table in ThoughtSpot, click the **Replay Search** icon.

2. Click the **Record Replay** button.

   If you do not see the button, you must log in as a user with administrator privileges.

A message will display, showing a URL and a domain or an IP address.

3. Make note of both of these items.

4. Open a new browser tab and go to the URL shown in the message (for example, “about:config”).

   Depending on which browser and version you are using, you may need to access the browser configurations through a menu or by typing in a different URL. Check your own browser help section for information on how to access the browser configuration settings, if necessary.

You may see a message warning that you are about to override the browser settings.

5. If you trust yourself, click “I’ll be careful, I promise!”.

6. Find the setting for **media.getusermedia.screensharing.allowed_domains**, and add the domain used by ThoughtSpot.

   This domain will be the same one you made note of from the **Cannot record screen** message.

media.getusermedia.agc_enabled	default	boolean	false
media.getusermedia.audiocapture.enabled	default	boolean	false
media.getusermedia.browser.enabled	default	boolean	true
media.getusermedia.noise	default	integer	1
media.getusermedia.noise_enabled	default	boolean	true
media.getusermedia.playout_delay	default	integer	10
media.getusermedia.screensharing.allow_on_old_platforms	default	boolean	false
media.getusermedia.screensharing.allowed_domains	default	string	webex.com,*.webex.com,ciscospark.com,*
media.getusermedia.screensharing.enabled	default	boolean	true
media.gmp-gmpopenh264.abi	**user set**	**string**	**x86_64-gcc3-u-i386-x86_64**
media.gmp-gmpopenh264.lastUpdate	**user set**	**integer**	**1454453226**

7. If you see a message asking if you’d like to share your screen with the IP address or domain name of ThoughtSpot, select **Entire screen**.

8. When the search replay has been recorded, you can see a confirmation. Select **Download**.

9. Save the recording on your computer by selecting **Save File** and clicking **OK**.

Opening ThoughtSpot tutorial Untitled.mp4
You have chosen to open:
ThoughtSpot tutorial Untitled.mp4
which is: MP4 file (5.5 MB)
from: https://192.168.2.247
What should Firefox do with this file?
Open with
QuickTime Player (default)
Save File
Do this automatically for files like this from now on.
Cancel
OK

Upgrade a cluster

ThoughtSpot is installed or updated from a release tarball which contains the ThoughtSpot application (binaries and configuration), third-party software, and Operating System image. Third party softwares are licensed software components necessary for operation of the ThoughtSpot application. These include Java, Boost C++ libraries, Google protocol buffers, and so on.

ThoughtSpot patches the Operating System at the time of upgrades. The exact same process used during installation is also applied during upgrades. The previous OS image on a node gets replaced by the new image carried in the release tarball.

> **Note:** ThoughtSpot Support will contact you to schedule an update when a minor or major upgrade becomes available.

Understand the backup strategies

Summary: Consider the strategies for backing up your ThoughtSpot cluster.

Snapshots

A snapshot is a point-in-time image of your running cluster. Snapshots are both taken on and restored to a cluster while it is running. Each cluster has a periodic snapshot configuration enabled by default. This configuration instructs the system to periodically take snapshots. Creation of a snapshot takes about 20 seconds. After creation, a snapshot persists on disk in the cluster's HDFS.

You can also create a snapshot manually. You should create a snapshot before making any changes to the environment, loading a large amount of new data, or changing the structure of a table. A snapshot may only be restored to the same cluster on which it was taken. The cluster software release version must match the snapshot release version.

If you need to move data between clusters or restore to a cluster that was updated to a new release, contact ThoughtSpot Support.

Backups

A backup is a procedure that stores a snapshot outside of a ThoughtSpot cluster. Backups are stored in a directory on a local or network file system. You can store all of the data associated with a snapshot or a portion of that data or only metadata. There is no default configuration enabled for backing up a cluster. You can configure on yourself or you can take backups manually. Backing up periodically protects your company from losing data and/or user work.

You can use a backup to restore a cluster to a prior state, a differently configured appliance, or move the backup from an appliance to a virtual cluster or vice versa.

Offline backup cluster

The most robust strategy for backup and recovery requires having a backup cluster offline that is kept in sync with the production cluster. Then, if the production cluster fails, the backup cluster can be drafted to take its place with minimal loss of work and disruption to operations.

Details on this architecture and instructions on setting it up are available in the ThoughtSpot Disaster Recovery Guide, which you can request from ThoughtSpot.

Choosing a strategy

Depending on your situation and your goals, you can choose to use either a snapshot or a backup. This table should help you decide:

	Snapshot	Backup
Purpose	Restore to a cluster to particular point in time.	• Restore a cluster to a prior state. • Move a cluster to a different appliance. • Move a cluster to VM appliance. • Removing a node. • Restoring to a cluster that runs a different release from the one where the backup was taken.
Storage	In the cluster's HDFS	Outside the cluster on either local or NAS disk.
Advantages	• Can be taken on, or restored to, a running cluster • Fastest create and restore	• Very stable. • Can be used to recover from data loss or corruption, even if the cluster is destroyed. • Can be typed as *full*, *lightweight*, or *dataless*.

Limitations	• Include all data, state, and metadata created between snapshot create and restore. • Lost if the HDFS name node fails, if you lose multiple disks, or if the entire cluster is destroyed • Can be restored only to the cluster were they are taken	• Require deleting the existing cluster first. • You are responsible for validating your backup configuration as viable for restoring a cluster. • Best practice recommends you to maintain multiple backups. • Typically, very large in memory size. </ul> </td> </tr> </table> You should never restore from a snapshot or backup yourself. Contact ThoughtSpot Support for help.

Understand backup/snapshot schedules

Summary: Explains how schedules and gives examples.

You can schedule periodic snapshots and backups. For snapshots, ThoughtSpot comes configured with a strongly recommended periodic snapshot policy. For backups, there is no such policy but you may want to create one or several of your own configurations. This section helps to understand existing schedules and how to configure new schedules.

Configuration format

ThoughtSpot uses a protocol buffer [See page 0] configuration file to hold snapshot and backup policies. There are slight differences between the configuration of snapshots and backups. You can read more about these later. However, the file format defines a `schedule` structure which is the same for both snapshots and backups. The following example shows the `schedule` format:

```
schedule {
    period {
        number: integer
        unit: MINUTE | HOUR | DAY
        }
    retention_policy {
        bucket {
            time {
                number: integer
                unit: MINUTE | HOUR | DAY
             }
             capacity: integer
        }
    }
    offset_minutes_from_sunday_midnight: integer
}
```

The `schedule` has the following components:

`period`	Specifies the frequency in the chosen `unit`. You can specify the `unit` as `MINUTE`, `HOUR`, or `DAY`.
`retention_policy`	Specifies retention intervals. Retention is on a first-in-first-out (FIFO) basis. So, the oldest result is always discarded. You can specify the `unit` as `MINUTE`, `HOUR`, or `DAY`. You can specify multiple retention buckets and they can have different retention policies.
`offset_minutes_from_sunday_midnight`	Determines the minute within the hour you'd like execution to start. Setting this to zero is equivalent to midnight.

Work through an example schedule

In this section, you work through an example schedule. This is a working example that is the actual default snapshot schedule set on every ThoughtSpot instance.

```
schedule {
    period {
        number: 1
        unit: HOUR
    }
    retention_policy {
     bucket {
         time {
            number: 1
            unit: HOUR
         }
         capacity: 3
     }
     bucket {
        time {
            number: 4
            unit: HOUR
        }
        capacity: 2
        }
    }
    offset_minutes_from_sunday_midnight: 0
}
```

Under this policy, a snapshot is taken every hour starting at midnight. You can see that by combining the `period` of 1 hour with the midnight offset of 0.

```
schedule {
    period {
        number: 1
        unit: HOUR
    }
    ...
    }
    offset_minutes_from_sunday_midnight: 0
}
```

Using this frequency, a total of 24 snapshots are taken in a day.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	…	24

If you were to specify a `number` of 2, the frequency changes. The first execution would start at midnight but subsequent executions would happen every 2 hours as shown here:

2	4	6	8	10	12	14	…	24

You use the `retention_policy` to control how many snapshots are kept. In this example, the first bucket retains a snapshot every three hours.

```
retention_policy {
    bucket {
        time {
            number: 1
            unit: HOUR
        }
        capacity: 3
    }
    ...
 }
```

At the beginning of the fourth hour, the system discards the snapshot from the first hour. This is in accordance with FIFO behavior. Therefore, this retention bucket contains snapshots from hours 2, 3, and 4.

1D	2R	3R	4R	5	6	7	8	9	10	11	12	13	14	…	24

The second bucket retains the snapshot taken at four hour intervals.

```
 retention_policy {
    ...
    bucket {
        time {
            number: 4
            unit: HOUR
        }
        capacity: 2
    }
}
```

It retains two of these four-hour-interval snapshots at any one time. By hour 9 during the day, you have the snapshots from hour 4 and hour 8 in this second bucket.

1	2	3	4R	5	6	7	8R	9	10	11	12	13	14	…	24

Consider what you will have in the first bucket in hour 9? The first bucket has the snapshots from hour 9, 8, and 7.

At the end of the day, in the first bucket, you can have 22, 23, and 24th snapshot. While in the second bucket, you will have the 20th hour and the 24th hour snapshots.

1	…	12	13	14	15	16	17	18	19	20R	21	22R	23R	24R

What if you changed the `period` frequency to every 2 hours? What would you have retained in your buckets at hour 24?

1	…	12	14	16	18R	20R	22R	24R

As you can see, when defining a policy it can be helpful to graphically represent the frequency you configure. Then, determine which time blocks are important to retain before determining your retention bucket.

Work with snapshots

Summary: A snapshot is a point-in-time image of your running cluster. You can use a snapshot to restore the cluster to a specific point in time. In this section, we describe how to work with the default snapshot configuration that is enabled on every cluster, and how make manual snapshots.

To work with snapshots, use the `tscli` command line interface.

> **Warning:** Backups rely on the snapshot system, so you must never disable the periodic snapshot system. If you disable periodic snapshots and enable periodic backups, the backups either fail or use an outdated snapshot.

Create a manual snapshot

You must create a snapshot before making any changes to the environment, loading a large amount of data, or changing the structure of a table. ThoughtSpot supports up to 20 manual snapshots. You must clear them to create new snapshots. To delete a snapshot, contact ThoughtSpot Support [See page 0].

ThughtSpot generates a snapshot in approximately 20 seconds, depending on the size of the cluster. To restore from a snapshot, contact ThoughtSpot Support [See page 0].

> **Note:** During an upgrade, all snapshots from the previous version of ThoughtSpot become manual snapshots.

Create a snapshot using tscli

To create a snapshot using the `tscli`, follow these steps:

1. Log into the Linux shell using SSH.
2. Create a snapshot, specifying its name and the reason for creating it.

Note that snapshot names must not exceed 44 characters.

```
$ tscli snapshot create <name> <reason> <ttl>
```

3. Check that the snapshot was created successfully by checking the directory listing:

```
$ tscli snapshot ls
```

Configure periodic snapshots with snapshot policy

Each ThoughtSpot cluster automatically makes periodic snapshots based on its default snapshot policy. The default snapshot policy is enabled for every cluster. You can display the current policy for periodic snapshots. You can change this policy.

View snapshot policy

To view your current periodic snapshot policy, follow these steps:

1. Log into the Linux shell using SSH.
2. Enter `tscli snapshot-policy show` command to view the policy.

In the following example, the policy starts at midnight on Sunday. It retains the snapshots from the previous three 4-hour intervals, and two snapshots from two of the previous 4-hour intervals. Here, ThoughtSpot retains 7 periodic snapshots overall. For detailed information about understanding the schedule, see Understand backup/snapshot schedules [See page 251].

```
$ tscli snapshot-policy show
schedule {
  period {
    number: 1
    unit: HOUR
  }
  retention_policy {
    bucket {
      time {
        number: 1
        unit: HOUR
      }
      capacity: 3
    }
    bucket {
      time {
        number: 4
        unit: HOUR
      }
      capacity: 4
    }
    bucket {
      time {
        number: 1
        unit: DAY
      }
      capacity: 4
    }
    bucket {
      time {
        number: 1
        unit: WEEK
      }
      capacity: 2
    }
  }
  offset_minutes_from_sunday_midnight: 0
}
enabled: true
```

Update the snapshot policy

Do not change the default policy, unless it is recommended by ThoughtSpot Support [See page 0].

> **Note:** Your policy must hold no more than 20 snapshots at any time. Exceeding this number impacts cluster performance.

To update your current periodic snapshot policy, follow these steps:

1. Log in to the Linux shell using SSH.
2. Enter `tscli snapshot-policy update`.
3. In the editor, make changes to the current policy, and save them.

You cannot delete the snapshot policy. However, you can either disable or enable it.

Disable the snapshot policy

To disable your current periodic snapshot policy, follow these steps:

1. Log into the Linux shell using SSH.
2. Enter `tscli snapshot-policy disable`.

Enable the snapshot policy

To enable a specific periodic snapshot policy, follow these steps:

1. Log into the Linux shell using SSH.
2. Enter `tscli snapshot-policy enable`.

Understand backup modes

A backup is a procedure that stores a snapshot outside of a ThoughtSpot cluster. You can use a backup to restore a cluster to a prior state, a differently configured appliance, or move it to from an appliance to a virtual cluster or vice versa. Other advanced administrative operations also use backups.

You can create a manual backup or configure an automated, periodic backup A backup stores snapshot outside of a ThoughtSpot cluster. For manual backups, the system creates a backup using the named snapshot you specify. For periodic backups, the system uses the most recent snapshot to create the backup.

> ❗ **Warning:** You should never disable the periodic snapshot system as backups rely on it. For example, if you have disabled the periodic snapshots system and periodic backups are enabled, then the periodic backup may use a very outdated snapshot or it may fail all together.

Backups are usually stored on a NAS (network attached storage) file system [See page 56] but you can store them on a local disk as well. When creating a backup, ThoughtSpot copies a release tarball and several supporting files to a disk you specify. Storing these supporting files takes about 10 GB of extra space beyond the backup itself. The final backup image is smaller because these extra files are removed after the backup completes successfully. So, make sure you have enough disk space both to *take* a backup and store the result. Use the `tscli storage df` command to identify the amount of space available.

You can create a backup using one of three modes, full, lightweight or dataless.

Full backups

Full backups are entire backups of the cluster with all data, whether loaded from the web interface or from `tsload`. This is the best mode for restoring a cluster and all your data. After a `FULL` backup is created, you can move them between clusters, even if the cluster configuration is different. Full backups can be as large as 20 GB in addition to the 5 GB of additional files. Some installations can exceed these limits, this is why it is important to test your backup configuration.

Before creating a manual backup or configuring automated backups, make sure there is enough disk space on the target disk. Consider an example, where you want to store three backups. If the backup itself takes 18GB, you need about 18 + 5 = 23 GB of free disk space. Don't forget that the backup size can grow over time, so you should occasionally check to ensure you are not in danger of running out of disk space to store backups.

Lightweight backups

Lightweight backups contain everything that makes up a cluster so they contain the following:

- Cluster configuration (SSH, LDAP, etc.)
- In-memory data cache
- All data that is stored unencrypted in HDFS
- Data uploaded by users
- Metadata for the data store
- Users, groups and permissions
- Objects created by users (pinboards, worksheets, and formulas) with their shares and permissions.
- Data model and row-level security rules.

Data loaded through ThoughtSpot Loader (`tsload`), ODBC/JDBC drivers, and Data Connect is excluded. The expectation is that data loaded by `tsload` is from external sources and so can be re-loaded after the cluster is restored. An exception is if these mechanisms were used to load data into tables that were first created through CSV import (that is, a user first loaded the tables using the GUI). In this case, the data, like the tables they were loaded into, are saved.

Dataless backups

A dataless backup saves a backup of the schema (metadata), with no data. Dataless backups allow you to send a copy of your cluster metadata to ThoughtSpot Support for troubleshooting without compromising data security and privacy. The size of a dataless backup is usually within 10's of megabytes provided you do not have customized binaries.

When restoring from a dataless backup, you must supply the correct release tarball, since this type of backup does not include the software release.

Create a manual backup

Summary: Learn how to manually create a backup.

ThoughtSpot enables you to manually create a backup to restore a cluster to a prior state or a differently configured appliance. You can also move the backup from an appliance to a virtual cluster or vice versa. To restore a cluster from a backup, contact ThoughtSpot Support [See page 0].

You create a manual backup from an existing snapshot. So, you must identify an existing snapshot to use or take a new snapshot first. The time required to take a backup depends on the data size. Taking a backup does not take long, and happens in the background while the cluster is running.

Using tscli

To manually create a backup using ThoughtSpot's command line interface, tscli:

1. Log in to the Linux shell using SSH.
2. Create a manual snapshot or find a snapshot you want to use. To find a snapshot you want to back up, use the following command:

```
$ tscli snapshot ls


-------------------------------------------------------
-------------------

Name          : pre330
Reason        : pre3.3.0
Hdfs snapshot : pre330
Start         : Wed May 4 18:07:32 2016
End           : Wed May 4 18:08:23 2016
Size(Full)    : 13.24 GB
Size(LW)      : 4.96 GB
Size(Dataless): 39.76 MB

-------------------------------------------------------
-------------------

...
```

3. Make sure you have enough room on the target disk.

   In addition to the size of the snapshot, you must have 10 to 12 GB of disk space. This is because the process requires space for temporary files. You can use the `df` command to check disk size.

```
$ df -h
```

4. Create the backup, designating the type of backup [See page 260], the snapshot name, and a directory:

   Choose the mode of backup [See page 260] you want to create, either full, lightweight, or dataless. The destination directory is created for you; do not specify an existing directory. The BASE value is the name of the backup.

```
$ tscli backup create [-h]
    [--mode {full|light|dataless}]
    [--type {full}]
    [--base snapshot_name>]
    [--storage_type {local|nas}][--remote]
     <name> <directory>
```

5. Check that the backup was created by listing all backups:

```
$ tscli backup ls
```

Configure periodic backups

You can configure ThoughtSpot to backup automatically at specified times. The policy allows you to control the type, frequency, retention periods (first-in-first-out), and output location for a periodic backup.

A periodic backup uses the same steps as creating a backup manually. However, you do not have to specify a snapshot name, the system uses the most recent backup. You can backup to a local file system or mount a NAS (network attached storage) file system [See page 56] to hold the backup. A NAS is recommended. Make sure you have adequate space to store the number of backups you want to archive.

Default policy format

The format for a policy includes the following:

```
name: "name_for_backup"
param {
    mode: FULL | DATALESS | LIGHTWEIGHT
    type: STANDALONE
}
schedule {
    period {
      number: integer
      unit: MINUTE | HOUR | DAY
    }
    retention_policy {
          time {
          number: integer
          unit:  MINUTE | HOUR | DAY
        }
        capacity: integer
      }
  }
  offset_minutes_from_sunday_midnight: integer
}
directory: "NAME"
storage_type: NAS | LOCAL
```

Before you begin

Before creating a policy, make sure you have read Understand backup/snapshot schedules [See page 251] for information on configuring a `schedule` element. In addition, you must specify:

Element	Description
`mode`	The backup mode. `FULL` backups are necessary for restoring a cluster. See Work with backups [See page 260] for details on each backup mode.
`type`	Currently, only `STANDALONE` is supported.
`directory`	The location on the disk to place the backup.
`storage_type`	The type of storage you are using. `NAS` storage is recommended for `FULL` backups.

Create a backup policy

Backups cannot start when another backup is still running. So, choose a reasonable frequency for the mode in you policy. For example, a `FULL` backup takes longer than a `DATALESS` backup. Consider the load on the system when configuring. Do not backup up when the system would experience a heavy load. For example, you may want to take `FULL` backups late in the evening or on weekends.

The retention system deletes the oldest stored backup and the corresponding snapshot on a first-in-first-out basis (FIFO). This means that if you set a bucket retention of 1 the system stores a single backup at any one time. The system deletes the older backup after the new full backup is successful. <!–

Configure using Management Console

> ℹ **Note:** The Management Console is now available in **beta** for customers with ThoughtSpot 5.3 or later. Please contact ThoughtSpot Support, if you want to try it.

To configure periodic backups using the admin UI:

1. Log into ThoughtSpot from a browser.

2. Click the **Admin** menu on the top navigation bar.

This opens the ThoughtSpot Management Console.

3. Click **Settings** menu on the top navigation bar.

4. In the Settings panel, click **Backup Policy** and then **Configure** option.

5. Update the backup policy details:

Field	Description
Backup Policy Name	Specify the name of the backup policy.
Location	Specify the backup location.
NAS Path	If you choose NAS, select the NAS path or configure a new NAS mount.

Mode	Select the backup mode. Allowed type are `full` , `light weight` or `dataless` .
Directory Name	Specify the location on the disk to place the backup.
Period	Specify the frequency in the chosen unit. Allowed unit types are Minutes, Hours, or Days.
Retention Policy	Specify the retention intervals in the chosen unit. Allowed unit types are Minutes, Hours, or Days. Retention is on a first-in-first-out (FIFO) basis. So, the oldest backup is always deleted after the new full backup is successful.
Capacity	Specify the retention capacity. **Note:** You can add multiple retention buckets with different retention policies. Click **Add Retention Policy** to specify more policies.

6. Click **Save** to update the backup policy.

–>

Configure using tscli

To configure periodic backups using the tscli:

1. Log in to the Linux shell using SSH.
2. Find a directory with enough disk space to support the `retention_policy number` you configure.

   You can use `df -h` to see free disk space and `tscli snapshot ls` to view existing snapshots and their size on disk.

3. Use the `tscli backup-policy create` command.

   The command opens a `vi` editor for you to configure the backup policy.

4. Write and save the file to store your configuration.

   By default, newly created policies are automatically enabled. To disable a policy, use the `tscli backup-policy disable` command.

5. Verify the policy using the `tscli backup periodic-config <name>` command.

Doing more with backup

The following table lists some additional backup commands you can use.

To	Command
List present backup policies.	`tscli backup-policy ls`
Show a backup policy.	`tscli backup-policy show <name>`
Check the status of a policy.	`tscli backup-policy status <name>`
Change an existing policy.	`tscli backup-policy update <name>`
Disable or enable an existing policy.	`tscli backup-policy disable` or `enable`
Delete a policy	`tscli backup-policy delete <name>`

Finally, you can time a `crontab` job with your periodic backup configuration to move a backup to longer term storage. Simply create a `crontab` job that moves the backup to a location outside of the `directory` defined in the periodic schedule.

About restore operations

When restoring to a running cluster where the ThoughtSpot software is not updated, we recommend that you use a snapshot. But in the case where you updated the cluster to a new release, if the configuration changed significantly, or when restoring to a different cluster, you must restore from a backup.

Restoring from backup require that you first delete the old cluster. Changes to a cluster that require restoring from a backup instead of a snapshot include:

- Removal of a node.
- Restoring to a different cluster from the one where the snapshot/backup was taken.
- Restoring to a cluster running a different release from the one where the snapshot/backup was taken.

You should never restore from a snapshot or backup yourself. To perform a restore from a snapshot or backup, contact ThoughtSpot Support.

About data modeling

Summary: Modeling, tagging, and adding links between your data sources can make the data even easier to search.

Data modeling allows you to define metadata and other aspects of your data. For example, you can give data columns search friendly names or predefine how they can be explored and aggregated. Metadata include such information as **Column Names**, **Column Visibility**, **Column** and **Data Definition**, **Column Rank** and so forth.

When you load data, ThoughtSpot has defaults for data modeling metadata. After loading data, you can start searching your data without doing any data modeling, creating relationships, or tagging. However, since you know your data best, you can customize the modeling settings. Putting some thought into these will make the data even easier and more intuitive to search for your end users.

User interfaces for modeling data

Data modeling is a very lightweight process compared to what you may have experienced in other tools. You can configure the model for an individual data table or you can view and configure all the system data using a modeling file. Editing the data model file requires that you have administrative privileges.

The model file contains a row for each column in your data set. It isn't unusual to have tens of thousand of rows in this file. This means that editing this file is equivalent to editing all the tables simultaneously. When you add new data to your system, this file expands to accommodate the new data columns you have added.

Both of these methods, have the same effect, they improve search. Moreover, while they have different effects of scale, they use the same mechanisms to accomplish these effects.

Modeling topics

The following topics explain how to model your data:

- **Change a table's data model** **[See page 275]**
  Explains how to make modeling settings for a table you've just loaded, or to make a quick change to existing settings.

- **Edit the system-wide data model** **[See page 277]**

  Explains how to define a default data model to use for data system-wide.

- **Data modeling settings** **[See page 282]**

  Explains the possible data model settings and their accepted values. These are the same for a table or the system.

- **Link tables using relationships** **[See page 312]**

  Linked tables can be searched together or combined into a worksheet for easy searching. Tables that have no relationship between their columns can not be combined in a single search.

- **About stickers** **[See page 318]**

  You can create stickers to make it easier for people to find data sources and pinboards.

Change a table's data model

Summary: You can adjust the data model for a newly loaded table.

To make modeling settings for a data source you've just loaded, or to make a quick change to existing settings, use the ThoughtSpot web interface. You can adjust the **Columns** settings from the data management listing.

You can change all the same data model settings here as in the model file. This method is easier and faster, unless you need to make many settings in bulk. In that case, using the model file [See page 277] is recommended.

About data sources

You can change the data modeling settings for base **Tables**, **Worksheets**, and **Views** **[See page 0]**. Worksheets will inherit the data modeling settings from the tables upon which they are based. However, if you make further changes to a base table *after* you've created worksheets on it, the new data model changes will not propagate up. You must make any new data model changes directly to the worksheets (if you want them).

Change the data model for a data source

1. Click **Data** on the top navigation bar.

2. Click a data source you own or can edit.

This brings up the **Columns** screen, where you can make your modeling settings.

3. Modify one or more column settings.

   Descriptions of the possible settings are listed in Data modeling settings [See page 282].

4. Save your changes.

5. To check your changes, use the **SEARCH** page to search for across the changed data.

Related information

- Data modeling settings [See page 282]
- Edit the system-wide data model [See page 277]
- Understand data sources [See page 0]

Edit the system-wide data model

Summary: Edit the modeling file to edit your data settings.

When you load data, ThoughtSpot uses defaults for data modeling metadata. You change these defaults using the data modeling file if you have access to the **ADMIN > Data Management** page. Editing this file allows you to view and edit all the system data columns. When you (or your users) add new data to your system, this file changes as it expands to accommodate new data columns.

> **Tip:** If you just want to change a subset of your data, use the Change a table's data model [See page 275] instead.

Overview of the modeling process

The data formats you use in your system are controlled by the modeling file, an Excel file. To make these changes you download the model file, change the model, and upload your changes back into the system.

In each row of the modeling file, all the data properties corresponding to a column from your data are listed. You can modify many of these properties by typing in the new value. Remember these important guidelines when editing the model file:

- Do not modify any value in a column which contains **DoNotModify** in the field under the column heading.
- Make sure to keep the file in the same format as it had when you downloaded it.

The model file contains a row for each column in your data set. It isn't unusual to have tens of thousand of rows in this file. You can change all or a subset of rows. You can edit the file to leave the heading rows and only those rows you want to change. This can make the file more convenient to work with.

The model file must be saved as UTF-8 encoded. If your model file includes multi-byte characters, make sure you save it in the correct format or you won't be able to upload it after making your changes.

Download the model file

Before you can make changes to the model file, you need to download it. Then, you edit it using Microsoft Excel, vi/vim, or a similar text editing tool.

To obtain the model file:

1. Log in to ThoughtSpot from a browser as an Administrator user.
2. Click the **ADMIN** tab in the top navigation bar.

3. Click **Data Management**, then click **Business Data Model**.
4. Click **Download**.

Edit the file and change the settings

You can make changes to the settings using this procedure. To see a list of the changes you can make, see Data modeling settings [See page 282]. You can edit any of the values in the model file, except for those where the words **DoNotModify** appear under the column header. To make changes in the model file:

1. Open the model file you downloaded (`model.xls`) in Excel, vi/vim, or a text editor.

   If you are using Excel, you may see a warning message.

   Click `YES` to proceed.

2. Find the column you want to modify.

   Descriptions of the meanings of the columns are listed in Data modeling settings [See page 282].

3. Select the value you want to change.
4. Type in the new value.
5. After making all your changes, save the model file.

   If you are using Excel, you will see a message. Click `YES` to save the file.

The model file must be saved as UTF-8 encoded. If your model file includes multi-byte characters, edit the file using vi or vim to ensure the file is saved in the correct format. Otherwise, you won't be able to upload it after making your edits.

Upload the edited file

After you have made changes to the modeling file, you must upload it back to ThoughtSpot before the changes will take effect. To upload the model file:

1. Log in to ThoughtSpot from a browser as an Administrator user.
2. Click **ADMIN**, on the top navigation bar.

3. Click **Data Management**, then click **Business Data Model**.
4. Click **Browse your files** to upload the model.xls file, or drag and drop it in the zone.

If you receive an error message upon uploading the file, check that it does not include any multi-byte characters (for example, Japanese or other multi-byte language characters). If it does, you must download the file again and make your edits using vi or vim.

If you choose to remove all the rows you have not changed from the model file before uploading it. If you upload a model file that includes only the changed rows, you won't lose any of the pre-existing model file settings. This is a good option if your model file is causing an error on upload, but you aren't sure where in the model file the problem is.

As soon as the file is uploaded, ThoughtSpot performs any necessary re-indexing for you automatically. Your new settings will be reflected within a few minutes.

Related information

- Data modeling settings [See page 282]
- Change a table's data model [See page 275]

Overview of data modeling settings

You can change these settings in two ways, both of which change the model. If you want to make a few small changes, you should make them in the ThoughtSpot application [See page 275]. If you want to make many changes you should edit the modeling file [See page 277]. Whether you are changing data modeling settings using the modeling file or the Web interface, the settings and their accepted values are the same.

Modeling settings

The following index includes mutable data modeling settings that you can apply to columns, both tables and worksheets.

Setting	Description
Column Name [See page 0]	Sets the name of the column to be used in searches.
Description [See page 285]	Adds a text description of what the column contains.
Data Type [See page 105]	**Read only.** Shows the column's data type.
Column Type [See page 286]	Sets the type of column, either `ATTRIBUTE` or `MEASURE`.
Additive [See page 287]	Controls the type of aggregations that will be available for a column.
Aggregation [See page 287]	Sets the default aggregation type for a column.
Hidden [See page 291]	Sets the visibility of a column.
Synonyms [See page 291]	Adds synonyms that can be used in the search bar to refer to a column.
SpotIQ Preference [See page 293]	Excludes specified columns from SpotIQ analyses. By Default, all columns are included in SpotIQ.
Index Type [See page 295]	Sets the type of index that will be created for a column.

Setting	Description
Geo Config [See page 301]	Enables a column to be used in GeoMap visualizations.
Index Priority [See page 295]	Changes the priority of a column in search suggestions.
Format Pattern [See page 303]	Specifies the format to use for numeric values or dates that show in the column.
Currency Type [See page 307]	Specifies the format of currencies in a column.
Attribution Dimension [See page 309]	Only applies to tables that join over a Chasm Trap [See page 118]. Designates whether the tables depend on this column for attribution.
Calendar Type [See page 29]	Specifies what type of calendar a date type column uses. It can be Gregorian calendar (default), a fiscal calendar, or any custom calendar.
Entity Category [See page 369]	Specifies how to categorize the data in the column: person, place, time, and so on. Important for configuring SearchIQ (**Beta**).
SearchIQ Enabled	Indicates if the column is enabled for SearchIQ. Note that SearchIQ is in **Beta**.

Data modeling best practices

As a best practice, make any data modeling settings in the table when you will be creating multiple worksheets that use that table. This way, you won't have to make the same settings in each worksheet. The settings will be inherited when you create worksheets that uses columns from the table.

If you have settings that only apply in the context of a particular worksheet, make those settings in the worksheet rather than in the underlying table(s).

Note that if you make your settings at the table level, and then create a worksheet that uses columns from the table, the settings are inherited from the table at the point in time that the worksheet is created. If you then go back and change the settings at the table level, your changes will not be reflected in the worksheet.

If you want the worksheet to have the changes you made at the table level, you must drop those columns from the worksheet and re-add them. Then save the worksheet. At this point, the new settings will be used in the worksheet. Note that any saved answers or pinboards based on the worksheet may display differently because of your changes. For example, if you've changed the GeoMap setting from "None" to "Country", you will now see a map where before you might have seen a table.

Related information

- Model the data for searching [See page 0]
- Add a geographical data setting for a column [See page 301]

Set column name, description, and type

Summary: Modeling includes setting basic information for a data column such as its name, description, and type.

Basic information for a data column is its **NAME**, **DESCRIPTION**, and **TYPE**. All of these can influence how a user experiences your data. For example, the **DESCRIPTION** appears as a “tip” when a user hovers over a column. So it is the means for helping users understand where the data comes from.

Change the column name

Column Name (UI)/**ColumnName** (model file) is the name that displayed to users for that column in ThoughtSpot. The column name is what users type to add that column to their search. Change the text that is shown for the column names in ThoughtSpot to make the names more meaningful to users.

The model file contains a row for each column in your data set. It isn’t unusual to have tens of thousand of rows in this file. You can change all or a subset of rows. You can edit the file to leave the heading rows and only those rows you want to change. This can make the file more convenient to work with.

The default is the name you gave the column when you defined the table in the database or imported the CSV file from the browser.

1. Find the column name you want to change.
2. Type in the new column name.
3. Save your changes.

Change column description

Description (UI)/**ColumnDescription** (model file) an optional description for the corresponding column. You can provide a description for a specific column, to provide additional information for users about the data it contains. When a user hovers over the column, a tooltip will show this description.

To create a column description:

1. Find the column description you want to change.
2. Enter a new description.

3. Repeat for all columns where you want to add a description.
4. Save your changes.

Change column type

Column Type (UI)/**ColumnType** (model file) describes the kind of data a column stores. This is set automatically upon defining the table, but in some cases, you may want to change the type. There are two types of columns:

- `ATTRIBUTE` contains a property, like name, address, or id number.
- `MEASURE` contains a numeric value that can be compared in a meaningful way using math, such as a count or measurement.

When a new table is created, the default column type is set according to the **Data Type** (UI)/**DataType** (model file) defined for each column. By default, columns with the numeric data types (`FLOAT` , `DOUBLE` , `INT` , or `BIGINT`) are assigned the type `MEASURE` . Columns with `VARCHAR` , `BOOL` , or date/time data types are assigned the type `ATTRIBUTE` .

Usually the default setting for column type works fine. But occasionally, you must change a `MEASURE` to an `ATTRIBUTE` . Examples of numeric values for which mathematical operations are not meaningful include:

- ID numbers
- Key values that are primarily used for joining tables
- Product number or SKU
- Sports team member jersey number
- Year, when separate from a date (e.g. 1999, 2000)

To change the column type:

1. Find the column type you want to change.
2. Change it to either `MEASURE` or `ATTRIBUTE` .
3. Save your changes.

Related information

- Model the data for searching [See page 0]
- Hide column or define a column synonym [See page 291]

Set ADDITIVE or AGGREGATION

> **Summary:** You can allow aggregate on MEASURE columns and some ATTRIBUTE columns.

Aggregation is grouping many units or parts into a new value. In data, aggregation gathers multiple input values and calculates an summary value from them. You then use this aggregated value to do an analysis.

Every summary value results from a data aggregation function. An example aggregation function would be average or minimum. You can control how aggregation works in your data.

Making an ATTRIBUTE column ADDITIVE

Your data may contain a column with a numeric data type that you have defined as an `ATTRIBUTE` rather than a `MEASURE`. For example, you may have `ATTRIBUTE` column with an `INTEGER` data type that represents age. Typically, these columns have an **ADDITIVE** setting of `NO`. Within a search result that contains data from this column, the options for each column on the left side of the screen includes:

- **UNIQUE COUNT OF**
- **TOTAL COUNT OF**

To display extended aggregate view options, you must set **ADDITIVE** to `YES` on these `ATTRIBUTE` columns. This option is only possible on columns that have a numeric data type (`FLOAT`, `DOUBLE` or `INTEGER`) or a date data type (`DATE`, `DATETIME`, `TIMESTAMP`, or `TIME`). After you make this change, these additional view options area-charts offered:

- **TOTAL OF**

- **AVG OF**
- **STD DEVIATION OF**
- **VARIANCE OF**
- **TOTAL COUNT OF**
- **UNIQUE COUNT OF**
- **MIN OF**
- **MAX OF**

To change this setting:

1. Find the column whose **ADDITIVE** setting you want to change
2. Select the **ADDITIVE** toggle.
3. Change the value to one of these:
    - `YES` or `NO`, if using the Web interface.
    - `TRUE` or `FALSE`, if using the model file.
4. Save your changes.

Change Aggregation

Both `MEASURE` columns and `ATTRIBUTE` columns support **AGGREGATION** operations. To aggregate a column without having to enter the aggregation type explicitly in your searches every time, you can set a default **Aggregation** for that column. Setting this default can make combining data more intuitive and faster.

`ATTRIBUTE` columns have **AGGREGATION**(UI)/**AggregationType** (model file) values with default aggregate type of **NONE**. You can change **AGGREGATION** to one of the supported aggregation types. To extend the available aggregation actions, set **ADDITIVE** on these columns to `YES` (`TRUE`).

Aggregate type	Description
NONE	Does no aggregation. This is the default for `ATTRIBUTE` type columns.
SUM	Adds the values together and returns the total. This is the default for `MEASURE` type columns.
AVERAGE	Calculates the average of all the values.
MIN	Calculates the minimum value.
MAX	Calculates the maximum value.
STD_DEVIATION	Calculates the standard deviation of all the values.
VARIANCE	Calculates the variance of all the values.
COUNT	Calculates the total number of values.
COUNT_DISTINCT	Calculates the total number of distinct values.

Keep in mind that not all `MEASURE` data should be aggregated. Consider a table containing data about athletes on a sports team. The data contains some numerical values, including points scored, salaries, and jersey numbers for each of the players. Because jersey number is an INTEGER, it would become a column of type `MEASURE` (not `ATTRIBUTE`). So it will aggregate, by default. But you may want to make its aggregation type **NONE** instead. This ensures that search results that include jersey number will not attempt to compare or aggregate those values in a way that is not meaningful.

To set this value.

1. Find the column whose default aggregation type you want to change
2. Select its **Aggregation**. If using the modeling file, use the **AggregationType** setting.
3. Select the new default aggregation type.
4. Save your changes.

Related information

Model the data for searching [See page 0]

Hide a column or define a synonym

Summary: Hide a column from users or make it easier to find by assigning a synonym.

You can hide columns from users in ThoughtSpot without dropping them from the database. It is common to hide a column when its data contains identifier columns that are used to join tables, but which do not have any meaning to users.

Alternatively, rather than hiding a column, you can make it easier to find by creating synonyms for it. This is helpful, for example, when different departments refer to the data using different terminology.

Hide a column

As the number of columns in the dataset increases, the search experience requires more effort. Users have to navigate through larger numbers of columns to choose the correct one. There might also be some columns in the dataset that you don't want to expose to the users.

Change the **HIDDEN** (UI)/**Hide** (model file) setting to hide a column. By default, all columns in a data source appear in ThoughtSpot. To hide these columns, set the **HIDDEN** setting to `YES`.

1. Find the **HIDDEN** (UI)/**Hide** (model file) setting for a column.
2. Set its value to `YES`.
3. Save your changes.

Create synonyms for a column

When users search a data source, they might try typing different words to try to retrieve a particular column. This could be due to different groups in your organization using different terms for the same data. Or maybe your users just intuitively use different words when searching for that item. Using synonyms allows them to access the data even if the term they choose isn't the same as the actual column name.

You can set column synonyms for columns in tables, user imported data, and worksheets. The returned table or chart uses the *actual column name*, but the search bar reflects the term the user typed in (the synonym).

To create a synonym for a column:

1. Find the column for which you want to add synonyms.
2. Select its **Synonyms**.
3. Enter a comma-separated list of the synonyms.

   If a synonym is more than one word, enclose it in double quotes. If you are using the Web interface, you would type:

   ```
   profit,"gross profit"
   ```

   If you are using the model file, the list of synonyms must be enclosed in square brackets:

   ```
   [profit, "gross profit"]
   ```

4. Save your changes.

Related information

Model the data for searching [See page 0]

Set columns to exclude from SpotIQ analyses

Summary: You can specify columns to exclude from SpotIQ analyses.

SpotIQ [See page 0] is a ThoughtSpot feature that provides users with insights about their data by automatically surfacing interesting characteristics (trends, correlations, outliers, and so on).

If you have access to tables, worksheets, and views for data modeling purposes, you can specify columns to exclude from SpotIQ analyses. By default, all columns are *included* in SpotIQ analyses.

Exclude columns from SpotIQ analyses

To specify columns to exclude from SpotIQ analyses:

1. Click **Data** in the top menu, and choose **Tables**, **Worksheets**, or **Views [See page 0]**.
2. Click the name of your data source.
3. On the **Columns** tab, find the COLUMN NAMES you want to exclude from SpotIQ analyses, and scroll to the right to find `SPOTIQ PREFERENCE`.
4. Use the drop-down menu to set the `SPOTIQ PREFERENCE` to **EXCLUDE** for each column you want to exclude.
5. Click **SAVE CHANGES** in the upper right.

Include columns in SpotIQ analyses

By default, all columns are included in SpotIQ analyses.

If you have previously set some columns to EXCLUDE and you want to re-set these to be included, do the following.

1. Click **Data** in the top menu, and choose **Tables**, **Worksheets**, or **Views [See page 0]**.

2. Click the name of your data source.

3. On the **Columns** tab, find the COLUMN NAMES you want to set back to include in SpotIQ analyses, and scroll to the right to find `SPOTIQ PREFERENCE`.

4. Use the drop-down menu to set the `SPOTIQ PREFERENCE` to **DEFAULT** for each column you want to include.

5. Click **SAVE CHANGES** in the upper right.

Related information

- SpotIQ tutorial [See page 0]

- Overview of data modeling settings [See page 282]

Manage suggestion indexing

Summary: ThoughtSpot dynamically indexes Search bar suggestions for column names and values.

When a user searches in the **Search** bar, ThoughtSpot supplies the user with suggestions for column names and their column values. The **COLUMN NAME** and any **SYNONYMS** appear in **Search** suggestions. A column's **INDEX TYPE** controls whether and how ThoughtSpot suggests column values.

Additionally, ThoughtSpot uses a column's **INDEX PRIORITY** value to determine where to rank a column's name and values in the search suggestions. These values impact the dynamically calculated *usage based ranking (UBR)*,

Example of Search suggestion behavior

The following example illustrates how searching for `promotion_last_name t` causes the system to suggest several ways of completing the `t` in the search:

The system is suggesting the synonym `type` (callout A) for a column in the `Promotion` table. It is also suggesting a value of `theil` (callout B) for the `Promotion_Last Name` column. If you look in the **Data > Tables** page, you can see that there is a `type` synonym for the `Promotion_Type` column which is using default indexing.

Managing search suggestions through **INDEX TYPE** and **INDEX PRIORITY** is important. Properly configured suggestions can decrease “noise” in the suggestion list. Increasing the visibility of important columns is helpful for new or intermittent ThoughtSpot users.

Understand the default indexing behavior

ThoughtSpot has a system default **INDEX TYPE** behavior for search suggestions. This system default is configured on your cluster and applies to all worksheets and tables. You can override this default behavior on a per-column basis.

The system behavior when the **INDEX TYPE** is **DEFAULT** is as follows:

- With two exceptions, the system indexes all columns using their **COLUMN NAME** value. The exceptions are columns with **COLUMN TYPE** of `MEASURE` and columns with **DATA TYPE** of `DATE`.
- Columns that contain data values with large amount of free-form strings, that is, a length is greater than 50 words, are indexed as `PREFIX_ONLY` by default.

> ❗ **Warning:** If a column has a very large free text values, ThoughtSpot recommends you keep `DEFAULT` or set `DONT_INDEX`. Other settings indexing on these values may generate confusing suggestions.

- Short strings (like a `firstname` column) are indexed using `PREFIX_AND_SUBSTRING` by default, which indexes both prefix and substrings.

- If a column is using has a *cardinality* – the number of unique column values – greater than 10 million, it is not indexed.

If a column's **INDEX TYPE** is *not* **DEFAULT** and the column's cardinality is greater than 30 million, ThoughtSpot does not index the column.

High cardinality and performance

A column's cardinality can impact indexing. If you have a column with a very high cardinality and a very high number of rows, indexing these values can impact your ThoughtSpot performance. ThoughtSpot Support recommends you turn off indexing of primary key columns on extremely large tables (> 10 million rows) in your cluster.

High cardinality is relative to other considerations. In some cases, columns with fewer than 10 million rows but with columns containing long strings can cause performance problems with memory. If you have concerns or questions, your ThoughtSpot Customer Success Engineer can help you determine appropriate cardinality thresholds for your ThoughtSpot installation.

Configure your own cluster defaults

If you need to, you can work with ThoughtSpot Support or your Customer Success Engineer to configure new cluster defaults.

Override the system default on a column

You can change a column's **INDEX TYPE** in the **Data > Tables > Columns** page or in the **Index** value in the modeling file.

The values you can set for **INDEX TYPE** are:

Index type	Description

`DEFAULT`	The default behavior applies to all `ATTRIBUTE` columns that are not `DATE` types. `PREFIX_AND_SUBSTRING` for short values and `PREFIX_ONLY` for long values and free-form text.
`DONT_INDEX`	Prevents indexing on the column values. The column doesn't appear in search suggestions.
`PREFIX_AND_SUBSTRING`	Allows full indexing such that prefix and sub-string search both work for the column values.
`PREFIX_AND_WORD_SUBSTRING`	Allows indexing such that only prefix search works for each word of a multi-word string, for the column values.
`PREFIX_ONLY`	Allows indexing such that only prefix search works for the column values.

Consider a column in which there are four values `ThoughtSpot`, `Thought`, `Spot` and `Thought Spot`. If you search for `sp`, depending on the setting for indexing, the column value search result suggestions will vary:

Index field value	Search bar suggestions
`DEFAULT`	`ThoughtSpot`, `Spot` and `Thought Spot`
`DONT_INDEX`	No suggestions.
`PREFIX_AND_SUBSTRING`	`ThoughtSpot`, `Spot` and `Thought Spot`
`PREFIX_ONLY`	`Spot`
`PREFIX_AND_WORD_SUBSTRING`	`Spot` and `Thought Spot`

To change a value in the application UI:

1. Open a worksheet or table from the **Data** page.
2. Find the column whose index type you want to modify.
3. Set its **INDEX TYPE**.
4. Save your changes.

If you are using the model file, locate the **Index** cell, and enter the **INDEX TYPE** you want to use.

Change a column's suggestion priority

A column's **INDEX PRIORITY** determines the order or rank in which it and its values appear in the search dropdown.

By default, the **INDEX PRIORITY** value is set to `1` for all columns. You can push a column up in the order (increase the rank) by increasing its **INDEX PRIORITY** value. A higher value (like `2`) will cause the corresponding column and its values to appear higher up in the search dropdown than columns with lower value (like `1`).

You should only use numbers between 1-10 in the **INDEX PRIORITY** field.Use a value between `8–10` for important columns to improve their search ranking. Use `1–3` for low priority columns.

To change a value in the application UI:

1. Open a worksheet or table from the **Data** page.
2. Find the column whose index type you want to modify.
3. Change the **INDEX PRIORITY** to a number between 1 and 10.
4. Save your changes.

If you are using the model file, locate the **Index** cell, and enter the priority you want to use.

Related information

- Model the data for searching [See page 0]
- Usage based rankings (UBR) [See page 0].

Add a geographical data setting

Certain attribute columns that contain location data can be used to create GeoMaps. ThoughtSpot supports Latitude, Longitude, Zip Code, US States, US Counties, Countries, and select international sub-nation regions.

You can designate a column as `Geo` by editing the **GEO CONFIG** column in the table **Columns** page. You cannot edit the geo configuration column information in the `model.xls` file.

Guidelines for geographic columns

Columns that can be designated as `Geo` columns need to contain text (`VARCHAR`) data unless they contain latitude/longitude data. Latitude and longitude columns can contain numeric data (`DOUBLE`) or text.

If you are using a column with the data type `DOUBLE` for latitude and longitude, you will also need to change the following settings for those columns:

- set **Column Type** to `ATTRIBUTE`
- set **Additive** to `NO`
- set **Aggregation Type** to `NONE`

For information these settings, see Set ADDITIVE or AGGREGATION [See page 287].

How to edit geographic columns

1. Find the **GEO CONFIG** for the column that contains the geographical data.
2. Select the column to display the **Specify Geographic Configuration** dialog.

3. Change the value to the appropriate **GEO CONFIG**, depending on the kind of geo data the column contains.

   If your data includes latitude and/or longitude columns that are stored as a numeric data type (`DOUBLE`), make these changes for those columns:

   a. Change the **Type** to `ATTRIBUTE` .

   b. Change **ADDITIVE** to `NO` / `FALSE` .

4. Save your changes.

Supported geo maps

For a complete list of supported geo maps in ThoughtSpot, refer to: Geo Map Reference [See page 0]

Related information

Model the data for searching [See page 0]

Set number, date, and currency formats

Summary: Explains how to set key formats for column values.

You can set number, date, and currency display formats. These formats define how these value types display in tables and charts.

Number formats

You can set a format for how numbers are displayed in tables and charts. For example, you can display numbers with a different number of digits after the decimal point, based on the data modeling setting **Format Pattern**. You can use any of the supported number formats for delimiters and number of digits to show using Java Decimal Notation [See page 0]. Currency symbols are not supported.

The system has default values which are:

`#,###`	For integer data types `INT` and `BIGINT`. As you can see, these can only contain numbers, alpha characters are not permitted.
`#,###.00`	for decimal data types `DOUBLE` and `FLOAT`.

These are some examples of formats you can use:

Stored Value	Format Pattern	Display Value
12345.6789	`#,##0.##`	12,345.68
12345.6789	`#,##0.###`	12,345.679
12345.6789	`#,##0.00000`	12,345.68
12345.6789	`#,##0`	12,345
12345.6789	`#,##0.00`	12,345.68
12345	`#,##0.##`	12,345
12345	`#,##0.00`	12,345.00

You can change the date format used to display a column's values for a single table [See page 275] or, by editing the data model, for the entire ThoughtSpot instance [See page 277]. Editing the data model file requires that you have administrative privileges.

1. Decide if the change is for a table or the entire instance.
2. Find the **Format Pattern** for the column.

   This is either a column in a single table or a column in the data modeling file.
3. In the column, enter the format you want to use.
4. Save your changes.

   If you are using a data-modeling file you must upload the new file to your installation.

Profile-based number formatting

Number formatting is set by default based on your ThoughtSpot profile's **Preferred locale** setting. You can set this value to accommodate your geographic locations.

For example, if you are using ThoughtSpot in the US, the number formatting should look like this: `xxx,xxx.xx`. And in Europe, it should look like this: `xxx.xxx,xx`.

Date formats

Format Pattern (UI)/ **Format Pattern** (model file) formats for how dates are displayed in tables and charts. For example, you can display dates in a standard European or US format based on the data modeling setting **Format Pattern**. These are some examples of formats you can use:

Format mask	Description
`YYYY` or `yyyy`	four digit year such as `2017`
`YY` or `yy`	last two digits of year such as `17`
`M`	month with no leading zero `1 - 12`
`MM`	Two digit month `01 - 12`
`MMM`	Three letter month such as `Jan`
`D`	Day of year without a leading zero `0 - 365`
`DD`	Day of year with up to one leading zero `01 - 365`
`DDD`	Day of year with up to two leading zeroes `001 - 365`
`d`	Day of month with no leading zero `1 - 31`
`dd`	Two digit day of month `01 - 31`
`HH`	Two digit `24` hour representation of hour `00 - 23`
`hh`	Two digit `12` hour representation of hour `01 - 12`
`H`	24 hour representation of hour with no leading zero `0 - 23`
`h`	12 hour representation of hour with no leading zero `1 - 12`
`mm`	Minutes `00 - 59`
`m`	Minutes with no leading zero `0 - 59`
`ss`	Seconds `00 - 59`
`s`	Seconds with no leading zero `0 - 59`

Format mask	Description
a	AM/PM indicator

Valid delimiters include most non-alphabet characters. This includes but is not limited to:

- `\` (forward slash)
- `/` (backward slash)
- `|` (pipe symbol)
- `:` (colon)
- `-` (dash)
- `_` (underscore)
- `=` (equal sign)

Examples of valid format masks you can produce for display are as follows:

- `MM/dd/yyyy`
- `MMM`
- `DD/MM/yyyy`
- `MM/dd/yyyy HH:mm`
- `DD/MM/yyyy HH:mm`

To change the date format used to display a column's values for a single table [See page 275] or, by editing the data model, for the entire ThoughtSpot instance [See page 277].

1. Decide if the change is for a table or the entire instance.

2. Find the **Format Pattern** for the column.

   This is either a column in a single table or a column in the data modeling file.

3. In the column, enter the format you want to use.

4. Save your changes.

   If you are using a data-modeling file you must upload the new file to your installation.

Set currency type

You can set a format for how currencies display in tables and charts when using the ThoughtSpot Data API or embedding. For example, you can display currencies in a standard European Euro or US Dollar format, based on the data modeling setting **Currency Type**.

You can change the currency format used to display a column's values for a single table [See page 275]. When you specify the currency type of your data in the **Columns** settings, your currency data will only display the correct format and currency code in the embedded use case. Currency specific symbols are available in the non-embedded use case as well, but they are not localized.

All users are treated as if they are in `en-US` locale unless they are in embed mode and their browser configuration tells ThoughtSpot that they are in some other locale. For example, `100 Polish Zloty` appears as `100zł` to a user in Poland, but without localization enabled, it appears as `PLN 100`.

This subtle difference can be seen when you use the REST API. See the ThoughtSpot Application Integration Guide for more information on the API.

1. Find the **Currency Type** for the column whose display format you want to change.

2. Click it to open the **Specify Currency Type** menu.

3. Select one of the following ways you would like to change the format.

Option	Description
Infer From Browser	Your currency data will be modeled upon the locale of your browser setting.
From a column	Your currency data will be modeled upon the existing currency information in the selected column This option is disabled if there is no `VARCHAR` column to choose from.
Specify ISO Code	Your currency data will be modeled upon your selection from the available currency code choices.

4. Click **Ok** to save your changes.

Related information

Model the data for searching [See page 0]

Change the Attribution Dimension

> **Summary:** The **Attribution Dimension** setting applies only to tables that are related through a chasm trap. If your schema does not include these, you can ignore this setting.

The **Attribution Dimension** setting only applies to tables that join over a Chasm Trap [See page 118]. By default, the attribution dimension setting will be set to `YES`, but you can override that by setting the column's attribution dimension property to `NO`, as described here.

Understand chasm traps and attribute dimension

In the classic chasm trap, two fact tables are related through a shared dimension table. When the two fact tables are joined, the shared column(s) in the dimension table are used to attribute rows in one fact table to match with rows in the other fact table.

Usually, all goes well using this method. But sometimes an incorrect or illogical attribution can result. This can happen because the column chosen is not meaningful for performing this attribution. If you are seeing unexpected results in searches that include tables across a chasm trap, this setting is for you.

Below is an example of a column that is not an attribution dimension. Suppose you have two fact tables, Wholesale Purchases and Retail Sales, that share a common dimension Date.

In this example, the date column in the Date dimension should not be used for attribution, since unrelated rows in both of the fact tables could share the same row in the Date table. Why? Because if Sally bought oranges wholesale on April 25, 2005 and made a retail sale of apples on the same day, there is no logical relationship between those two events. Combining the two events using the date they share will not create any meaningful information.

If matching rows in two fact tables over a chasm trap depends on the values in a column contained in a dimension table, that column is known as an attribution dimension.

In this example, the Product ID column in the Products dimension table is an attribution dimension. For rows where the Product ID in the Wholesale Purchases and in the Retail Sales tables is a match, those rows are logically related in a meaningful way. They can be combined in charts and reports to produce a logical, expected outcome.

How to set attribute dimension

You cannot configure this setting in the model file. You can only configure it on a table-by-table basis. To designate a column as not being an attribution dimension (not producing any meaningful attribution across a chasm trap):

1. Find the column that is not an attribution dimension.
2. Select its **Attribution Dimension**.
3. Set the value to `NO`. If you're using the modeling file, set it to **FALSE**.
4. Save your changes.

Related information

Model the data for searching [See page 0]

Link tables using relationships

You can link tables by creating relationships between their columns. Linked tables can be searched together or combined into a worksheet for easy searching. Tables that have no relationship between their columns can not be combined in a single search.

There are two ways to create relationships between tables:

1. Create a constraint using TQL. [See page 108]
2. Create a relationship through the web interface. [See page 313]

The two methods create the same kind of relationship both from an end user perspective and an administrative perspective. When creating a relationship between two tables, the columns that form the link must be the exact same data type. For example, a column of type `INT32` to another `INT32` column.

Both types of relationships exist within the database. You can also generate a script in TQL that contains all relationships, both the ones create in the web interface and the ones created in TQL.

Relationships created through either method can be managed either in TQL or by going to the **Relationships** page when viewing data in the **Date Modeling** section in the ThoughtSpot application. You can view, modify, or delete relationships in either place.

You may create relationships using a mixture of TQL and the web interface, but the relationships you create cannot form a circular relationship, or “cycle”. If you attempt to create a relationship that would complete a cycle, you will see a message stating that the relationship could not be added because it conflicts with another existing relationship.

Join a table or view to another data source

Summary: Learn how to define joins between a table or view and another table, view, or worksheet

Joining a table or view to another table, view, or worksheet creates a relationship that allows them to be searched together. Choose a column to join on that both tables contain (e.g. employee ID or product key). This process creates a generic join [See page 108] between the table or view and the other table, view, or worksheet on the column you specify.

If you want to create a primary key/foreign key relationship, you need to use TQL [See page 108] rather than the web interface.

You must have either the **Can administrator ThoughtSpot** or the **Can manage data** privilege [See page 163] to create a join relationship. If you’re not an administrator, you also need edit permissions on the table, view, or worksheet.

When creating a join between the columns in two data sources, the linked columns must have the same data type, with the same meaning. That is, they must represent the same data. Normally, you can make this kind of link from a fact table column to a column in a dimension table that uniquely identifies a logical entity in your data such as Employee ID for a person, Product ID for a product, or Date Key for a specific date in a date lookup table.

To create a relationship through the Web interface:

1. To find your table or view, click **Data** in the top menu, and choose **Tables** or **Views**.
2. Click the name of your table or view.
3. Click **Schema**. You will see the list showing existing joins.
4. Click the **+ Add Join** button on the upper right side of the screen.

5. Use the **Map source to destination** dialog to choose the destination table, view, or worksheet you want to join to.

6. Choose the columns you want to join on from the table or view (source) and destination table, view, or worksheet. Click **Next**.

7. Give your join a name and description and click **ADD JOIN**.

8. Repeat these steps until all the joins you want to make have been created.

After creating the join, you may change its name and description by clicking the edit icon. If you want to change the data source or column being joined, you must delete the join and create a new one.

Related Information

- Constraints [See page 108]

Delete a relationship

Summary: You can delete relationship (link) between tables through the application or TQL.

You must have either the **Can administrator ThoughtSpot** or the **Can manage data** privilege [See page 163] to delete a relationship. If you're not an administrator, you also need edit permissions on the table, view, or worksheet.

If you created a relationship (join or link) between tables using the Web interface, you can also delete it from the Web interface. But if the relationship was created using TQL, you must also use TQL to delete it.

To delete a relationship using TQL [See page 134], use an `ALTER TABLE...DROP CONSTRAINT...` or `ALTER TABLE...DROP RELATIONSHIP...` statement.

To delete a relationship from the Web interface:

1. Click **Data** on the top navigation bar.
2. Click the name of the data source you from which you want to remove the relationship.
3. Click **Schema**. You will see the list showing existing joins.

   If this is a worksheet, you must click **Joins within worksheets** and choose **Joins between worksheets**.

4. Find the relationship you want to delete, and click the **Delete icon**.

5. Repeat these steps until all the joins you want to remove have been deleted.

Related Information

- Constraints [See page 108]

About stickers

Summary: Stickers enable you to create categories for classification of objects, including pinboards, answers, data sources, and worksheets.

You can create stickers to make it easier for people to find data sources and pinboards. Stickers are global in scope. This means that everyone can see the stickers and use them to tag objects. They can also filter lists of objects by sticker. Stickers are often used to designate subject areas, such as sales, HR, and finance, but you can use them any way you like.

Keep in mind these permissions when working with stickers:

- Only administrators can create stickers.
- Anyone can apply a sticker.
- Anyone can filter by a sticker.

Create a sticker

Only administrator users can create stickers. Anyone can apply the stickers you create, or use them as filters when selecting from a list of sources or pinboards.

To create a sticker:

1. Navigate to the **Manage Data** or **Pinboards** screen using the icons in the top navigation bar.
2. Choose the currently selected sticker, scroll to the bottom of the list, and click **+ Add**.

3. Type the name for the new sticker.

4. You can change the name of a sticker by clicking the edit icon next to its name.

5. You can change the color of a sticker by clicking the color circle next to its name.

Apply a sticker

Only administrators create stickers, but anyone with edit privileges can tag an object with a sticker.

To tag an object with a sticker:

1. From the top menu, choose Answers, Pinboards, or Data.
2. Find the item(s) you want to tag in the list, and check the box next to its name.
3. Click the apply sticker icon and choose one from the list. You can apply as many stickers as you like to an object.

Filter by stickers

Whenever you are selecting objects from a list, you can filter by sticker to find what you're looking for. Anyone can use stickers to filter lists of pinboards or data sources. You can also filter by sticker when selecting data sources.

To filter by sticker:

1. From the top menu, choose **Answers**, **Pinboards**, or **Data**.

2. Click **Select sticker**, and select the name of the sticker you want to filter by.

Create and use worksheets

Summary: Worksheets are flat tables created by joining columns from a set of one or more tables or imported datasets.

After modeling the data, create worksheets to make searching easier. For example, a sales executive might need to search for information about retail sales. The required data could be contained in several tables (sales, customers, products, stores, etc.), with foreign key relationships between them. An administrator who is familiar with the data model can create a retail sales worksheet, that combines all of the related fact and dimension tables into a single, easy-to-use view, and share it with the sales executive. This provides access to the data without requiring an understanding of how it is structured.

Guidelines for worksheets

Users are often unfamiliar with tables and how they are related to one another. A worksheet groups multiple related tables together in a logical way. You might use a worksheet for these reasons:

- To pre-join multiple tables together.
- To give a user or group access to only part of the underlying data.
- To include a derived column using a formula.
- To rename columns to make the data easier to search.
- To build in a specific filter or aggregation.
- To give users a filtered set of data to search.

Typically, you create one worksheet for each set of fact and dimension tables. For example, you may have a sales fact table and an inventory fact table. Each of these fact tables shares common dimensions like date, region, and store. In this scenario, you would create two worksheets: sales and inventory. The following diagram depicts the workflow for creating the sales worksheet.

The process for creating a worksheet is:

1. Decide which tables to use for the worksheet.
2. Create a new worksheet.
3. Add sources (tables) to the worksheet.
4. Choose the worksheet join rule [See page 339].
5. Select the columns to include.
6. Optionally modify the join types [See page 347] within the worksheet.
7. Optionally create formulas [See page 331].
8. Optionally create worksheet filters [See page 334].
9. Save the worksheet.

10. Share the worksheet with groups or users [See page 211].

Create a worksheet

Create a worksheet to make the data easy for users to search. This process includes adding a new worksheet, after which you will choose the data sources to include in it.

To create a new worksheet:

1. Click **Data**, on the top navigation bar.

2. Click the ellipses icon , and select **Create worksheet**.

Add sources and columns to a worksheet

After creating a worksheet, you need to add the sources that contain the data. Sources is another name for tables. The sources you choose are typically related to one another by foreign keys.

To add the sources to the worksheet:

1. Click the **+** icon.

2. Check the box next to each of the sources you want to include in the worksheet.

   Note that the list of sources only shows the data sources on which you have view or edit privileges.

3. If you want to see what the data inside the sources looks like, click **Explore all data**.

4. Choose the worksheet join rule [See page 339].

5. If you want to disable Row Level Security [See page 223], for this worksheet, check the checkbox to disable it.

6. Click **CLOSE** to save your changes.

7. Expand the table names under **Columns** and select the columns to add to the worksheet, by doing any of the following:

    a. To add all of the columns from a table, click the table name and click **+ Add Columns**.

    b. To add a single column, double-click its name.

    c. To add multiple columns, Ctl+click each column you want to add and click **+ Add Columns**.

    Note that after you add a column, non-related tables (those without a primary/foreign key relationship) become hidden. If you are working with two tables that should be related, but are not, you can add a relationship between them [See page 312].

8. (Optional) modify the join types [See page 347] within the worksheet.

9. (Optional) create formulas [See page 331].

10. (Optional) create worksheet filters [See page 334].

11. Click the ellipses icon ••• , and select **Save**.

12. In the Save Worksheet window, enter a name and description for your worksheet and click **SAVE**.

13. (Optional) Click each column name and enter a more user-friendly name for searching. You can tab through the list of columns to rename them quickly.

14. (Optional) If you want to add a prefix to the name of several columns, select them, click the **Add prefix** button, and type in the prefix.

15. Click the the ellipses icon ••• , and select **Save**.

16. Share your worksheet [See page 211], if you want other people to be able to use it.

Where to go next

- **How the worksheet join rule works [See page 339]**

  Use the worksheet join rule to specify when to apply joins when a search is done on a worksheet. You can either apply joins progressively, as each search term is added

(recommended), or apply all joins to every search.

Edit or rename worksheet

Summary: A worksheet can be edited by anyone with the proper permissions.

If you created a worksheet, or you have edit permissions on it, you can make changes such as adding sources and columns, adding or editing formulas, changing relationships, and changing column names. You can also rename a worksheet or change its description.

Edit a worksheet

To edit a worksheet:

1. Click **Data** on the top navigation bar.
2. Click the name of the worksheet you want to edit.
3. Click the **Edit** button in the upper-right side of the screen.
4. Make your changes to the worksheet.
5. Click the the ellipses icon , and select **Save**.

Rename a worksheet or table

You can change a worksheet or table name from the ThoughtSpot application.

To change the name of a worksheet or table:

1. Click **Data**, on the top navigation bar.
2. Find the worksheet or table you want to rename and click its name.
3. Click the current name, and enter a new name.

Related information

- Change the join rule for a worksheet [See page 341]
- Add joins between a worksheet and other data [See page 343]
- Modify table joins within a worksheet [See page 347]

Create a formula in a worksheet

Summary: You can define formulas and use them to create derived columns in worksheets.

You create formulas by combining standard functions and operators, column names, and constant values.

Anyone who can create a worksheet can add a formula to it. Formulas are not reusable; the formula you create is associated only with the worksheet it belongs to. A complete list of available formulas and examples of each is available in the Formula function reference [See page 683].

You can create a formula in a worksheet by using the Formula Builder. When you do this, the result of the formula gets added to the worksheet as a column. Use these steps to create a formula:

1. Create a new worksheet, or edit an existing one.
2. Click the + button next to **Formulas**.

d

3. Type your formula in the Formula Builder.

Note: Formulas elements are color coded by type and can include the formula operators and functions (blue), the names of columns (purple), and/or constants (black).

4. If you want to change what your formula returns, use the **Advanced settings**.

   Depending on your formula, you may be able to change:

   - Data type
   - ATTRIBUTE or MEASURE
   - Aggregation type

5. You can see a list of formula operators with examples by clicking on **Formula Assistant.**

6. Name the formula by clicking on its title and typing the new name. Click **Save**.

Add a filter to a worksheet

Summary: You can add a filters to a worksheet to limit the data users can access from the worksheet.

Beginning in ThoughtSpot version 5.0, you can add filters to a worksheet to limit the data it contains. This is useful when you have underlying tables that store more data than is necessary for the types of analyses the worksheet is intended for. You can also use worksheet filters to provide data security, when you want different groups of users to be able to see different data without relying on row level security.

To add a filter to a worksheet:

1. Click **Data** in the top menu bar, find your worksheet, and click its name.

2. Click the **Edit Worksheet** button.

3. Click **Filters** on the left menu and click **+**.

4. Choose the column you want to filter on.

5. Select the values to include in your answer.

6. If you want to exclude values, click **Exclude** and choose values to exclude.

7. Click **ADD FILTER**.

If there are too many values, you can use the filter search bar to find the ones you want.

How the worksheet join rule works

Use the worksheet join rule to specify when to apply joins when a search is done on a worksheet. You can either apply joins progressively, as each search term is added (recommended), or apply all joins to every search.

Understand progressive joins

Often, a worksheet includes several dimension tables and a fact table. With progressive joins, if your search only includes terms from the fact table, you can see all of the rows that satisfy your search. But as you add terms from dimension tables, the total number of rows shown may be reduced, as the joins to each dimension table are applied. It works like this:

- If you choose **Apply joins progressively (recommended for most cases)**, joins are only applied for tables whose columns are included in the search.
- If you choose **Apply all joins**, all possible joins are applied, regardless of which tables are included in the search.

When using **Apply joins progressively**, the number of rows in a search using the worksheet depends on which tables are part of the search. The worksheet acts like a materialized view. This means that it contains the results of a defined query in the form of a table. So if a particular dimension table is left out of the search, its joins are not applied.

Rule-Based Row Level Security (RLS) with worksheets

With Rule-Based RLS, you need to protect every table that contains any sensitive data. To do this, you can grant access by creating explicit row level security rules on each of the underlying tables which contain data that row level security should apply to.

When creating the row level security rules for a table that’s part of a worksheet, you aren’t limited to referencing only the columns in that table. You can specify columns from other tables in the worksheet as well, as long as the tables are joined to the table you’re creating the rule on. Then, when creating a worksheet on top of them, the behavior is consistent regardless of the worksheet join rule you choose. Users will never be able to see data they should not, regardless of what their search contains.

Imagine you have a worksheet that contains a `Sales` fact table, and `Customer` and `Product` dimensions that are joined on `Customer SSN` and `Product Code` columns. In order to secure the `Sales` table, you can use `Customer Name` from the `Customer` column to create a row level security rule.

How joins are applied with chasm traps

When working with worksheets and row level security, you need to understand how joins are applied. This is particularly important with chasm trap schemas. For chasm trap schemas, if row level security is only set on one of the tables, people could see data they should not see if the scope of their search does not include that table. (this protects them from having people see the wrong things if they have chasm trap).

For chasm trap *worksheets*, progressive and non-progressive joins do not apply. There is an entirely different methodology for how worksheet joins on a chasm trap schema work with row level security. So you can safely ignore that setting.

Change the join rule or RLS setting for a worksheet

Summary: As long as you have permissions to edit a worksheet, you can always go into it and set a different join or RLS rule.

If you find that the charts and tables built on a worksheet contain a large number of null values (which display as `{blank}` in the web browser), you can fix this by changing the internal joins [See page 347] for the worksheet.

If you have the **Can administer ThoughtSpot** privilege or the **Can manage data** plus edit privilege on a worksheet, you can edit the worksheet and change its RLS or other key settings.

Change/configure a worksheet

Before working through this procedure, make sure you are familiar with how the following affect data:

- internal worksheet joins [See page 347]
- worksheet join rule [See page 339]
- row level security (RLS) [See page 228]

To configure these values for a worksheet:

1. Click **Data** on the top navigation bar and then on **Worksheets**.
2. Click the name of the worksheet you want to edit from the list.
3. Click the **Edit Worksheet** button in the upper right hand side of the screen.
4. Click the **+ icon** next to **Sources**.
5. Scroll to the bottom of the page.
6. Configure the worksheet join rule and RLS setting as needed.

7. Click **CLOSE**.

8. Click the the ellipses icon , and select **Save**.

Join a worksheet to another data source

Summary: Learn how to define joins between a worksheet and a table or view

Joining a worksheet to a table or view creates a relationship that allows them to be searched together. Choose a column to join on that both data sources contain (e.g. employee ID or product key). This process creates a generic join [See page 108] between the worksheet and the table or view on the column you specify.

Note that creating a foreign key relationship [See page 108] is preferred over a generic relationship in most cases, except for when you need to do a range join. Foreign key relationships perform better and protect users from overcounting upon aggregation.

You must have either the **Can administer ThoughtSpot** privilege or the **Can manage data** privilege to create a join relationship. If you're not an administrator, you also need edit permissions on the table, view, or worksheet.

When creating a join between the columns in two data sources, the columns being linked must have the same data type, with the same meaning. That is, they must represent the same data.

To create a relationship through the Web interface:

1. To find your worksheet, click **Data** in the top menu, and choose **Worksheets**.
2. Click the name of your worksheet.
3. Click **Schema**. You will see the list showing existing joins within the worksheet.
4. To view the joins between the worksheet and other data sources, click **Joins within worksheets** and choose **Joins between worksheets**.

5. Click the **+ Add Join** button on the upper right side of the screen.

6. Use the **Map source to destination** dialog to choose the destination table or view you want to join to.

7. Choose the columns you want to join on from the worksheet (source) and destination data source. Click **Next**.

8. Give your join a name and description and click **ADD JOIN**.

9. Repeat these steps until all the joins you want to make have been created.

After creating the join, you may change its name and description by clicking the edit icon. If you want to change the data source or column being joined, you must delete the join and create a new one.

Related Information

- Constraints [See page 108]

Modify joins within a worksheet

Summary: Learn how to change the join type between the tables within a worksheet

When you create a worksheet, you select a join rule [See page 339]. The join rule works together with the joins defined within the worksheet determine how the tables that make up the worksheet are joined, and how those joins behave when searching on the worksheet.

Starting with ThoughtSpot version 5.0, you are not limited to just one join rule for the entire worksheet. You can define different types of joins for each join between tables in a worksheet. Be default, each of these individual table joins uses an inner join. But you can override this at the individual join level.

You must have either the **Can administer ThoughtSpot** privilege or the **Can manage data** privilege to modify joins within worksheets.

Modify a join

To modify the join types within a worksheet, follow these steps:

1. Click the **Data** tab in the top menu.

2. Select **Table**, and then select **Worksheets**.

3. Click the name of the worksheet. Here, we selected the worksheet *West Regional Sales*.

4. At the top of the worksheet, click the **Schema** tab.

5. The join information and the schema for the worksheet appear.
    - The schema representation includes the join type.
    - You can see the list of *Joins within the worksheet*, which include all joins between the underlying tables, both created using TQL [See page 108] and created in the browser [See page 313].
    - The fact table is always on the left side of the join, and it appears on the left side.

Note the available actions for the join view.

Legend	Action
1.	See either *Join within this Worksheet (Default)*, or *Joins from this Worksheet*.

2.	Click the pencil icon to edit the join.
3.	Click the up arrow icon to collapse the join detail.

6. Select the join you plan to modify, and click the **Edit** (pencil) icon next to it.

7. In the **Edit Join** window modal, make the necessary changes:

- Under **Join Type**, select one of *Inner Join*, *Left Outer Join*, *Right Outer Join*, or *Full Outer Join*.
- Under **Join Cardinality**, select either *Many to One*, or *One to One*.

Click **Save**.

Related Information

- Join rule [See page 339]
- Create joins using TQL [See page 108]
- Create join relationships in the browser [See page 313]

Delete a worksheet or table

Summary: When you try to delete a worksheet or atable, you can see a message listing any dependent objects that must be removed first.

ThoughtSpot checks for dependencies whenever you try to remove a table or worksheet. A list of dependent objects appears, and you can click them to delete them or remove the dependency. Then, you can remove the table or worksheet.

To delete a worksheet or table:

1. Click **Data**, on the top navigation bar.

2. Find the worksheet or table you want to remove in the list, and check the box next to its name.

3. Click the **Delete** icon.

If you are attempting to delete a data source with dependent objects, the operation will be blocked. You will see a warning, with a list of dependent objects with links.

4. Click the link for an object to modify or delete it.

When all its dependencies are removed, you will be able to delete the data source.

5. You can also click the name of a worksheet or table and then click **Dependents**, to see a list of dependent objects with links.

   The **Dependents** list shows the names of the dependent objects (worksheets and pinboards), and the columns they use from that source. You can use this information to determine the impact of changing the structure of the data source or to see how widely used it is. Click a dependent object to modify or delete it.

WORKSHEET
Sporting Goods Retail Worksheet
ThoughtSPORT worksheet
Edit Worksheet
Columns
Schema
Data
Dependents
Row Security
SpotIQ Insights
COLUMN NAME
DEPENDENT NAME
TYPE
Age Group
Total Sales by Depar..
View
Date
Top 100 Products M..
View
Product Name
Top 100 Products M..
View
Department
Total Sales by Depar..
View
Sales
Total Sales by Depar..
View
Customer City
Customer Location
Answer
Customer Name
Customer Location
Answer
(showing rows 1-0 of 27)
THOUGHTSPOT

Migrate or restore Worksheets

Summary: You can export an entire ThoughtSpot worksheet in a flat-file format. After optional modification, you can migrate it to a different cluster, or restore it to the same cluster.

In ThoughtSpot, you can download Worksheets to a flat file in `yaml` format, modify the file, and subsequently upload this file either to the same cluster, or to a different cluster. This mechanism supports several scenarios that you may encounter:

- **Migrating from a development environment to a production environment** by downloading the file from the development cluster and uploading the same file into the production cluster
- **Implementing metadata changes outside ThoughtSpot UI**, such as replacing the underlying tables for the entire table, or replacing a single column from one table with a column in another table
- **Making bulk changes**, such as mass renaming of objects defined by the worksheets, and managing duplicates

Prerequisites

You must have **Edit** permissions for the worksheet.

Export Worksheet

To export a worksheet, follow these steps:

1. Click **Data** on the top navigation bar.
2. Click the name of the worksheet you want to edit.
3. Click the ellipses (more options) menu in the upper-right side of the screen.
4. From the menu, select **Export Worksheet**.

   Here, we are downloading the *Sales* worksheet.

5. The worksheet downloads into your default directory.

   You can access the downloaded `*.yaml` file, named for your worksheet either through the file system manager, or by clicking on the name of the downloaded file in the left bottom corner of your Internet browser.

Working with the Worksheet yaml file

You can change the yaml file that represents the Worksheet by editing its content. See YAML Worksheet specification [See page 357] for the full syntax of the YAML file.

Update a Worksheet

To update an existing worksheet, follow these steps:

1. Click **Data** on the top navigation bar.

2. Click the name of the worksheet you want to edit.

3. Click the ellipses (more options) menu in the upper-right side of the screen.

4. From the menu, select **Update Worksheet from file**.

   Here, we are uploading the edited *Sales* worksheet.

5. In the upload interface, click **Browse your files**.

6. In your file system, find and select the YAML file.

7. In the **Update worksheet** interface, click **Upload**.

8. If you constructed the Worksheet file correctly, the **Upload Worksheet** interface displays an *Upload successfull* message.

9. To examine the updated Worksheet, click **Go to Worksheet**.

Related Information

- Worksheet YAML specification [See page 357]

Worksheet YAML specification

Summary: ThoughtSpot worksheet specification may be exported as a YAML file, modified, and imported into the same or different cluster.

To work with Scriptable Worksheets in ThoughtSpot, you can download Worksheets to a flat file in `yaml` format, modify it, and subsequently upload this file either to the same cluster, or to a different cluster.

Syntax of the Worksheet YAML file

The YAML file for Scriptable Worksheets has a specific syntax.

See the Parameters [See page 360] section for details about the keywords used in this example.

```
worksheet [See page 0]:
  name [See page 0]: <worksheet_name>
  description [See page 0]: |-
    This is a multi-line description of the worksheet
    Description line 2
  tables [See page 0]:
  - name [See page 0]: <table_name_1>
  - [alias [See page 0]] : <table_alias>
  - [fqn [See page 0]] : <GUID_of_table_name>
  - name [See page 0]: <table_name_2>
  - name [See page 0]: <table_name_3>
  joins:
  - name [See page 0]: <join_name_1>
    source [See page 0]: <source_table_name>
    destination [See page 0]: <destination_table_name>
    type [See page 0]: [RIGHT_OUTER | LEFT_OUTER | INNER | OUTE
R]
    is_one_to_one [See page 0]: [ false | true ]
  - ...
  table_paths [See page 0]:
  - id [See page 0]: <table_path_name_1>
    table [See page 0]: <table_name_1>
    join_path [See page 0]:
    - join [See page 0]join:
      - <join_name_1>
  - id [See page 0]: <table_path_name_2>
    table [See page 0]: <table_name_2>
    join_path [See page 0]:
    - {}
  - id [See page 0]: <table_path_name_3>
    table [See page 0]: <table_name_3>
    join_path [See page 0]:
    - join [See page 0]:
      - <join_name_1>
    - join [See page 0]:
      - <join_name_2>
      - <join_name_3>
    - join [See page 0]:
      - <join_name_4>
      - <join_name_5>
      - <join_name_6>
  formulas [See page 0]:
  - name [See page 0]: <formula_name_1>
    expr [See page 0]: <formula_definition_1>
    [id]: <unique_identifier>
```

```
  - name [See page 0]: <formula_name_2>
    expr [See page 0]: <formula_definition_2>
  - name [See page 0]: <formula_name_3>
    expr [See page 0]: <formula_definition_3>
  worksheet_columns [See page 0]:
  - name [See page 0]: <column_name_1>
    description [See page 0]: <column_description>
    formula_id [See page 0]: <formula_name_1>
    properties [See page 0]:
      column_type [See page 0]: [ MEASURE | ATTRIBUTE ]
      aggregation [See page 0]: [ SUM | COUNT | AVERAGE | MAX
| MIN |
                      COUNT_DISTINCT | NONE | STD_DEVIATION | VA
RIANCE]
      index_type [See page 0]: [ DONT_INDEX | DEFAULT | PREFI
X_ONLY |
                     PREFIX_AND_SUBSTRING | PREFIX_AND_WORD_SUBS
TRING ]
      is_hidden [See page 0]: [ true | false ]
             index_priority [See page 0] : <index_priority>
      synonyms [See page 0] :
             <synonym_1>
             <synonym_2>
      is_attribution_dimension [See page 0] : [true | false]
      is_additive [See page 0] : [ true | false ]
      calendar [See page 0] : [ default | calendar_name ]
      format_pattern [See page 0] : <format_pattern_string>
      currency_type [See page 0] :
        is_browser : true
          OR
        column : <column_name>
          OR
        iso_code : <valid_ISO_code>
    geo_config [See page 0] :
        latitude : true
          OR
        longitude : true
          OR
        country : true
          OR
        region_name:
        - country : <name_supported_country>
        - region_name : <region_name_in_UI>
    name [See page 0]: <column_name_2>
    description [See page 0]: <column_description>
```

```
    formula_id [See page 0]: <formula_name_2>
    ...
  properties [See page 0]:
    is_bypass_rls [See page 0]: [ true | false ]
    join_progressive [See page 0]: [ true | false ]
```

Parameters of Worksheet YAML file

aggregation

The default aggregation of the worksheet column
Aggregation options depend on the data type.
Possible values: `SUM`, `COUNT`, `AVERAGE`, `MAX`, `MIN`, `COUNT_DISTINCT`, `NONE`, `STD_DEVIATION`, and `VARIANCE`
Default: `SUM`

alias

An alternate name for the table

calendar

Specifies the calendar used by a date column
Can be the Gregorian calendar (`default`), a fiscal calendar, or any custom calendar.
See Set up a custom calendar [See page 29]

column_type

The type of data the column represents
Possible values: `MEASURE` or `ATTRIBUTE`
Default: `MEASURE`

currency_type

The source of currency type
One of:

- `is_browser : true` infer the currency data from the locale of your browser
- `column : <column_name>` extracts the currency information from a specified column
- `iso_code : <valid_ISO_code>` applies currency based on the ISO code; see ISO 4217 Currency Codes [See page 0]

See Set currency type [See page 307]

description

The text that describes an object: a `worksheet`, a `worksheet_column`, and so on.

destination

Name of destination table or view of the join

expr

The definition of the formula

format_pattern

The format pattern string that controls the display of a number, date, or currency column
See Set number, date, and currency formats [See page 303]

formula_id

The `id` of the formula that defines the worksheet column

formulas

The list of formulas in the worksheet
Each formula is identified by `name`, the `expr` (expression), and an optional `id` attribute.

fqn

A GUID for the table name

geo_config

Specifies the geographic information of a column
One of:

- `latitue : true` for columns that specify the latitude
- `longitude : true` for columns that specify the longitude
- `country : true` for columns that specify the country
- `region_name` for specifying a region in a country
  Uses two paired parameters:
  - `country: <country_name>`
  - `region_name: <region_name_in_UI>`, which can be State, Postal Code, District, and so on.

See Add a geographical data setting [See page 301]

id

Specifies the id of an object, such as `table_paths`, `formula`.

index_priority

A value (1-10) that determines where to rank a column's name and values in the search suggestions
ThoughtSpot prioritizes columns with higher values.
See Change a column's suggestion priority [See page 299].

index_type

The indexing option of the worksheet column
Possible values: `DONT_INDEX`, `DEFAULT` (see Understand the default indexing behavior [See page 296]), `PREFIX_ONLY`, `PREFIX_AND_SUBSTRING`, and `PREFIX_AND_WORD_SUBSTRING`
Default: `DEFAULT`
See Index Type Values [See page 297]

is_additive

Controls extended aggregate options for attribute columns
For attribute columns that have a numeric data type (`FLOAT`, `DOUBLE`, or `INTEGER`) or a date data type (`DATE`, `DATETIME`, `TIMESTAMP`, or `TIME`)
Possible values: `true` or `false`
Default: `true`
See Making an ATTRIBUTE column ADDITIVE [See page 287]

is_attribution_dimension

Controls is the column is an attribution dimension

Used in managing chasm traps.
Possible values: `true` by default, `false` to designate a column as not producing meaningful attributions across a chasm trap
Default: `true`
See Change the attribution dimension [See page 309]

is_bypass_rls

Specifies if the worksheet supports bypass of Row-level security (RLS)
Possible values: `true` or `false`
Default: `false`
See Privileges that allow users to set, or be exempt from, RLS [See page 223]

is_hidden

The visibility of the column
Possible values: `true` to hide the column, `false` not to hide the column
Default: `false`
See Hide a column [See page 291]

is_one_to_one

Specifies the cardinality of the join
Possible values: `true`, `false`
Default: `false`

join

Specific join, used in defining higher-level objects, such as table paths
Defined as `name` within `joins` definition

join_path

Specification of a composite join as a list of distinct `join` attributes
These `join` attributes list relevant joins, previously defined in the `joins`, by name.
Default: `{}`

join_progressive

Specifies when to apply joins on a worksheet
Possible values: `true` when joins are applied only for tables whose columns are included in the search, and `false` for all possible joins
Default: `true`
See How the worksheet join rule works [See page 339]

joins

List of joins between tables and views, used by the worksheet
Each join is identified by `name`, and the additional attributes of `source`, `destination`, `type`, and `is_one_to_one`.

name

The name of an object. Applies to `worksheet`, `table`, `join`, `formula`, and so on.

properties

The list of properties of the worksheet column
Each column can have the following properties, depending on its definition: `column_type`, `aggregation`, `index_type`, `is_hidden`, `index_priority`, `synonyms`, `is_attribution_dimension`, `is_additive`, `calendar`, `format_pattern`, `currency_type`, and `geo_config`.

source

Name of source table or view of the join

synonyms

Alternate names for the column, used in search
See Create synonyms for a column [See page 291]

table

Specific table, used in defining higher-level objects, such as table paths
Defined as `name` within `tables` definition

table_paths

The list of table paths
Each table path is identified by the `id`, and additional attributes of `table` and `join_path`.

tables

List of tables used by the worksheet
Each table is identified by `name`.

type

Join type
Possible values: `LEFT_OUTER` for left outer join, `RIGHT_OUTER` for right outer join, `INNER` for inner join, `OUTER` for full outer join
Default: `RIGHT_OUTER`

worksheet

Top-level container for all object definitions within the worksheet

worksheet_columns

The list of columns in the worksheet
Each worksheet is identified by `name`, `description`, `formula_id`, and `properties`.

Limitations of working with Worksheet YAML files

There are certain limitations to the changes you can apply be editing a Worksheet through YAML.

- Formulas and columns can either have a new name, or a new expression. You cannot change both, unless migrating or updating the worksheet two times.
- It is not possible to reverse the join direction in the YAML script.
- It is not possible to include Worksheet filters in the YAML script.

Related Information

- Migrate or restore Worksheets [See page 353]

Enable SearchIQ

> **Summary:** Enable SearchIQ to provide natural language search for some or all of your users.

Note: SearchIQ is in Beta

- SearchIQ is only available in English.
- The feature is off by default; to enable it, contact ThoughtSpot Support [See page 0].
- You must have the **Can use experimental features** permission.

SearchIQ is a different search experience that understands more natural, speech-like search language than the original ThoughtSpot search or **Search+** [See page 0]. For example, you can search for `What was my top selling product last month?`, instead of having to type `top 1 product by sales last month`. You can also speak your search using the voice-to-text capability of your operating system.

Users can easily teach SearchIQ their language [See page 0] as they use it. Over time, SearchIQ gets smarter about how to interpret terms that don't come directly from keywords, column names, or data values. In order to give it time to learn, the best practice recommendation is to enable SearchIQ for a small group of users initially. Over time you can roll it out to larger groups as SearchIQ learns what your users search for the most.

To enable SearchIQ on a ThoughtSpot instance:

1. Call ThoughtSpot Support and ask them to turn on SearchIQ for your ThoughtSpot instance.

2. Grant the **Can use experimental features** privilege to any group that should have access to the SearchIQ search experience.

Note: Because the SearchIQ search experience is very different from the original ThoughtSpot search experience, you should give your new users some orientation on it, so they know what to expect.

Related information

- About SearchIQ [See page 0]
- Use SearchIQ [See page 0]
- Teach SearchIQ your language [See page 0]

Optimize SearchIQ

Summary: For SearchIQ to work well, you must enable, optimize, and enable it on the data source.

SearchIQ is in Beta.

> **Note:** SearchIQ is turned off by default, you can have ThoughtSpot Support enable it for you.
>
> SearchIQ is only available in English.

SearchIQ is a search experience that understands a more natural, speech-like search language. Contrast it with the original ThoughtSpot search [See page 0]. For example, you can search for `What was my top selling product last month?`, instead of typing `top 1 product by sales last month`. You can also speak your search using the voice-to-text capability of your operating system.

Data source owners must enable SearchIQ for their data sources, and then optimize the data source so users get better results. These are the primary steps for enabling and optimizing SearchIQ:

1. Enable columns for SearchIQ [See page 0]
2. Add Experts for SearchIQ [See page 0]
3. Train SearchIQ [See page 0]

Prerequisites

- SearchIQ is turned off by default; ask your administrator to enable it.
- You must be a member of a group that has the **Can use experimental features** permission.

Enable Columns

See Enable Columns for SearchIQ [See page 0].

Add Experts

See Add Experts for SearchIQ [See page 0].

Train SearchIQ

See Train SearchIQ [See page 0].

Mark the source as optimized for SearchIQ

After you finish training SearchIQ on your data source, click the **Data** tab, navigate to the data source, select its **SearchIQ** tab, scroll to the bottom left corner, and finally turn on the setting for **SearchIQ Optimized**.

How does SearchIQ work for end Users

After you finish training SearchIQ on the Dataset, we recommend that you share with the users a list of the questions you used for training SearchIQ, the names of columns and their descriptions, common synonyms, and some guidance on how to ask a question.

Remember that users who experience SearchIQ see only the columns that SearchIQ knows.

Related information

- About SearchIQ [See page 0]
- Use SearchIQ [See page 0]
- Teach SearchIQ your language [See page 0]

Set entity categories for SearchIQ

Summary: Entity categories help describe the column content, and assist SearchIQ in interpreting natural language queries.

Note: SearchIQ is in Beta

- SearchIQ is only available in English.
- The feature is off by default; to enable it, contact ThoughtSpot Support [See page 0].
- You must have the **Can use experimental features** permission.

Entity categories specify how to categorize the data in the column, by entity type. By default, `ENTITY TYPE` is not set. Entity categories support SearchIQ [See page 0] so that when you type a natural question, ThoughtSpot better knows how to interpret it. For example, if you ask “who are the top performers?” ThoughtSpot will first choose columns set with `PERSON` from which to return answers. If you ask “when was the movie Jurassic Park released?”, columns set to `TIME` will be used to answer the “when” part of the question, and so forth.

If you have access to tables and worksheets for data modeling purposes, you can specify entity categories for their columns. This lets you designate that column as representing a person, place, or thing. So then when a user asks “Which”, “Who”, or “Where”, an appropriate response can more easily be found in the data.

If you’re not sure how to change the data modeling settings, see the Overview of data modeling settings [See page 282].

About Entity Categories

These are the available Entity Categories:

Category	Description
PERSON	Contains data that represents a person, relevant to questions about “who?”
PLACE	Contains data that represents a location, relevant to questions about “where?”

Category	Description
TIME	Contains data that represents a date or time, relevant to questions about “when?”
PRODUCT	Contains data that represents a product
ZIP_CODE	Contains zip code data, relevant to questions like “where?” or “what zip code?”
LAT_LONG	Contains data that represents geographical positioning, relevant to questions like “where?”
COMPANY_ORG	Contains data that represents a company or organization
NUM_TYPES	Contains numerical data

Set Entity Categories

To specify entity categories:

1. Click **Data** in the top menu, and choose **Tables** or **Worksheets**.
2. Click the name of your table or worksheet.
3. On the **Columns** tab, find the COLUMN NAMES for which you want to specify entity categories, and scroll to the right to find `ENTITY CATEGORY`.
4. Use the drop-down menu to set the `ENTITY CATEGORY` to the type you want.
5. Click **SAVE CHANGES** in the upper right.

Related information

- SearchIQ [See page 0]
- Overview of data modeling settings [See page 282]

Edit the SearchIQ mappings

Summary: In SearchIQ, you can change the language mappings established earlier, during initial configuration.

Note: SearchIQ is in Beta

- SearchIQ is only available in English.
- The feature is off by default; to enable it, contact ThoughtSpot Support [See page 0].
- You must have the **Can use experimental features** permission.

When a user types a term in the SearchIQ search bar, SearchIQ interprets what the user means through language mappings. Users can teach their language to SearchIQ using Teach SearchIQ your language [See page 0].

As an administrator, you can see and change these language mappings by editing the mappings file. You can access the mappings file if you are in the Administrator group, because then you can have access to the **ADMIN > Data Management** page. You can download the mappings file there. This allows you to view and edit all the language mappings users made, in bulk.

Overview of the mapping process

The language mappings in SearchIQ are controlled by a CSV (comma separated values) file called the mapping file. The mapping file is saved with an .xls file extension, so you can open and edit it using Excel. To make these changes you will download the mapping file, change the mappings, and upload your changes back into the system.

In each row of the mapping file, you will see a natural language search term, that is mapped to a search keyword, or something that occurs in the data (like a column name or value). You can use the mapping file to remove mappings that have been made in error. Remember these important guidelines when editing the Mappings file:

- Do not delete rows from the file. Instead, type “True” in the **Delete Y/N** column for any mapping you want to delete.
- Make sure to keep the file in the same format as it had when you downloaded it.

The model file contains a row for each natural language mapping a user has created using the Teach feature of SearchIQ. It isn't unusual to have tens of thousand of rows in this file.

Download the mapping file

Before you can make changes to the mapping file, you need to download it. Then, you edit it using Microsoft Excel, vi/vim, or a similar text editing tool.

To obtain the model file:

1. Log in to ThoughtSpot from a browser as an Administrator user.
2. Click **ADMIN** in the top navigation bar.
3. Click **Data Management**, then click **Language Mappings**.
4. Click **Download**.

Edit the file and change the settings

You can delete any of the mappings in the file. To delete a mapping:

1. Open the model file you downloaded (`mappings.xls`) in Excel, vi/vim, or a text editor.

   If you are using Excel, you may see a warning message similar to this.

   Click `YES` to proceed.

2. Find the row for which you want to delete the mapping.
3. Find the **Delete Y/N** column.
4. Type **True** in the corresponding cell.
5. After making all your changes, save the model file.

   If you are using Excel, you will see a message similar to this one. Click `YES` to save the file.

The mapping file must be saved as UTF-8 encoded. If your mapping file includes multi-byte characters, edit the file using vi or vim to ensure the file is saved in the correct format. Otherwise, you won't be able to upload it after making your edits.

Upload the edited file

After you have made changes to the mapping file, you must upload it back to ThoughtSpot before the changes will take effect. To upload the mapping file:

1. Log in to ThoughtSpot from a browser as an Administrator user.
2. Click **ADMIN**, on the top navigation bar.
3. Click **Data Management**, then click **Language Mappings**.
4. Click **Browse your files** to upload the mappings.xls file, or drag and drop it in the zone.

   If you receive an error message upon uploading the file, check that it does not include any multi-byte characters (for example, Japanese or other multi-byte language characters). If it does, you must save it as UTF-8 encoded. You may need to download the file again and make your edits using vi or vim.

As soon as the file is uploaded, your new settings will be reflected within a few minutes.

Related information

- Teach SearchIQ your language [See page 0]

Job management (scheduled pinboards)

Summary: All jobs on your cluster will appear on the Jobs Management page. You can also view jobs for individual pinboards under the pinboard Actions dropdown.

The **Jobs Management** page found on the **Admin** section in the ThoughtSpot web application allows you to create and manage jobs, namely scheduled pinboards. Scheduled pinboards should help with preparing for recurrent meetings, when reviewing the same pinboard is necessary. They should also be useful when you have metrics you want to monitor at a consistent interval, like daily or monthly sales targets.

You can get pinboards emailed to you on a regular basis and do analysis offline. This introduces an additional format for you to consume and share pinboards with others, including those who don't have a ThoughtSpot account.

Contact ThoughtSpot Support if scheduled pinboards is not enabled on your cluster, or you can run the command `tscli scheduled-pinboards` to enable it yourself.

Scheduled pinboard creators

Administrators and users with can schedule pinboard privilege can schedule and manage pinboard jobs. These scheduled pinboard creators must have at least edit-only and view-only rights to the pinboard they want to share.

> ❗ **Warning:** It is recommended that admins carefully choose who to give can schedule pinboard privilege to, since there is a possible security hole where a user with limited access can get a pinboard email with all access data.

Row level security

The scheduled pinboards respect row level security rules. This means if the recipients are users in ThoughtSpot, then they can only see data based on their own access to the pinboard. If the user does not have at least view-only access to the pinboard, then they will not see anything in the email. However, if the recipients are from outside of the cluster, then they will have access to the dataset of the pinboard based on the sender's permissions.

Scheduled pinboard formats

The pinboard visualizations are attached to the scheduled email as CSV or PDF files. Saved configurations such as pinboard filters are applied to the attachments. Refer to the table to see how the pinboard data is represented in each file format.

CSV	PDF
The CSV file gets data only for table visualizations.	The PDF file gets data for all visualizations.
The email has n CSV attachments, where there are n table visualizations in the pinboard.	The email has only one attachment file, which includes every visualization on its own page.
Table visualizations have all data rows that they're supposed to have.	Table visualizations include only the first 100 rows.
In the case of a corrupted pinboard: no email is sent. An error message indicating failure to export data is visible on the Admin Jobs Management page.	In the case of a corrupted pinboard: the PDF attachment has empty/error screenshots.
In the case of a corrupted visualization: an email with the visualizations whose data can be exported is sent. An error message indicating visualization export error is visible on the Jobs Management page.	In the case of a corrupted visualization: the PDF attachment has empty/error slots for the corrupted visualizations.

The size of each email is limited to 25 MB, which matches most email services size limitations.

And the total number of recipients for a scheduled pinboard job cannot exceed the default of 1000.

Related information

For information on creating a pinboard, see the Schedule a pinboard job [See page 0].

Scheduled pinboards management

> **Summary:** You can manage all scheduled pinboards on the Jobs Management page under Admin.

Users who are not admins, but have can schedule pinboard privilege, can only view pinboard schedules they've created. You can select specific jobs and choose to pause, resume, edit, or delete them. You can have up to 50 scheduled jobs on your cluster at time. Contact ThoughtSpot Support if you'd like to increase this limit.

Bulk actions

Select the scheduled pinboards and use the **Delete**, **Resume**, and **Pause** buttons to perform these bulk actions. Deleting a pinboard will also delete any schedules linked to it.

Job statuses

Clicking on the row of a job will open a detailed view of every generated update of that job. You can see the start and end times of the job, as well as the status. Clicking on a job will show more information about the status updates.

Pinboard links

Click the scheduled pinboard name link to jump to a Edit schedule page, where you can edit the schedule configurations.

You can also click the pinboard link provided in the scheduled pinboard emails to jump to the pinboard in ThoughtSpot. In order to have the link direct you to the correct URL, you must first configure front end host and port access. Contact ThoughtSpot Support to configure these settings.

Introduction to monitoring

System monitoring tools in ThoughtSpot include an **Admin > System Health** page and system logs. Additionally, ThoughtSpot provides several worksheets and out-of-the-box system monitoring pinboards. From these worksheets, you can create your own custom visualizations and pinboards. This page introduces these features and directs you towards more detailed information.

System Health center

The ThoughtSpot application includes a **System Health** center, for easy monitoring of usage, alerts, events and general cluster health. You view the System Health Center by choosing the **Admin** icon and then selecting **System Health**.

Only users with administrative privileges can view the **System Health** center. However, administrative users can present to others the information that displays in the **System Health** center.

Administrators can also create their own, custom boards that reflect system data in ways that are meaningful to specific departments or groups. For more information, see the following documentation:

- Health Overview board [See page 381]
- Data board [See page 394]
- Cluster Manager board [See page 397]
- Alerts and Events board [See page 397]
- System worksheets [See page 402]
- System pinboards [See page 404]

Much of the data presented by these boards is also available through `tscli` commands [See page 621].

Log files

Many of the administration commands output logging information to log files. The logs get written into the fixed directory `/export/logs`, with a sub-directory for each subsystem. The individual log directories are the following:

- `/export/logs/orion`
- `/export/logs/oreo`
- `/export/logs/hadoop`
- `/export/logs/zookeeper`

You can also view [additional topics that also touch on log files [See page 0] throughout the documentation.

System monitoring notifications

You can configure ThoughtSpot to send emails to addresses you specify with monitoring reports and a cluster heartbeat. Follow these steps to Set up monitoring [See page 58].

Overview board

The **Overview** pinboard summarizes essential information about your cluster and its users. Choose **Admin > System Health > Overview** to see this pinboard.

Understand system boards and pinboards

The **Overview** page includes system panels and standard ThoughtSpot ***answers [See page 381]***. The system charts are generated in real time and rely on internal system data. The answers rely on underlying system worksheets which are available to ThoughtSpot administrators. The information in these worksheets is updated hourly from internal tables that collect monitoring statistics.

Each answer has a menu. You can present or copy the links to the system charts. The answers have a subset of the ThoughtSpot answer menu. You can use the menu to do additional actions such as download the answer or present information about your ThoughtSpot cluster. While you can interact with and change the search, you cannot save changes to the underlying query.

You can also interact with the answers, drilling into them to explore the detail as with any other pinboard answer.

To find out how a particular answer was created, do the following:

1. Select **Edit** from the panel menu.

   This displays a **Search** bar.

2. Investigate the components of the search as you would normally.

Cluster Summary

This system panel contains basic information about your cluster. The **NUMBER OF NODES** is the number of installed nodes. This doesn't reflect the active nodes which may be more or less.

This summary includes the **LAST SNAPSHOT TIME** it reflects whether regular snapshots of your cluster are collected. This value should update regularly in real time. If you do not see it change or empty, you should check your cluster snapshot policy using the `tscli` command:

```
$ tscli snapshot-policy show
schedule {
  period {
    number: 1
    unit: HOUR
  }
  retention_policy {
    bucket {
      time {
        number: 1
        unit: HOUR
      }
      capacity: 3
    }
    bucket {
      time {
        number: 4
        unit: HOUR
      }
      capacity: 2
    }
  }
  offset_minutes_from_sunday_midnight: 0
}
enabled: false
```

You can see this policy is disabled, which is a problem. Production clusters should enable the default snapshot policy. When you show or enable the snapshot policy, you can see your `tscli` command reflected in the **Configuration Events** panel on this same page.

Display the features used in a cluster configuration

1. Log into the ThoughtSpot cluster as the `admin` user.
2. Use the `tscli feature` subcommand to display your current configuration.

```
 $ tscli feature get-all-config
 +--------------------------------+----------
+---------------+
 |              NAME              |  STATUS  | CONFIGUR
ATION |
 +--------------------------------+----------
+---------------+
 | Firewall                       | Disabled
|              |
 | Saml                           | Disabled
|              |
 | Ldap                           | Disabled
|              |
 | CustomBranding                 | Disabled
|              |
 | CustomBrandingFontCustomization | Disabled
|              |
 | DataConnect                    | Disabled
|              |
 | RLS                            | Enabled
|              |
 | Callhome                       | Enabled
|              |
 | SSHTunnel                      | Enabled
|              |
 | Fileserver                     | Disabled
|              |
 +--------------------------------+----------
+---------------+
```

Relational Data Cache

This section reports real-time information about tables in your cluster. Worksheet data is not included.

Value	Description
TABLES LOADED	Number of currently loaded tables.
TABLES BEING UPDATED	Number of table loads in-progress.
NEW TABLES BEING LOADED	Number of tables being loaded for the first time.

Value	Description
ROWS	Number of rows combined across all tables in ThoughtSpot.

Relational Search Engine

Value	Description
TABLES SEARCHABLE	Tables that are indexed and can be searched.
TABLES BEING INDEXED	Total of in-progress table indexing.
NEW TABLES BEING INDEXED	Total of first-time, in-progress table indexing.
TOKENS SEARCHABLE	Number of ***tokens [See page 381]*** of all table (combined) indexed in ThoughtSpot.

Critical Alerts

Displays critical and warning alerts. This includes when an alert was generated and from which service and machine. Administrators can get a custom report by issuing a `tscli alert list` on the appliance:

```
tscli alert list --since 4w
```

The critical alerts you can encounter in this display are the following:

- `TASK_FLAPPING`

  Msg: `Task {{.Service}}.{{.Task}} terminated {{._actual_num_occurrences}} times in last {{._earliest_duration_str}}`

  This alert is raised when a task is crashing repeatedly. The service is evaluted across the whole cluster. So, if a service crashes 5 times in a day across all nodes in the cluster, this alert is generated.

- `OREO_TERMINATED`

  Msg: `Oreo terminated on machine {{.Machine}}`

  This alert is raised when the Oreo daemon on a machine terminates due to an error. This typically happens due to an error accessing Zookeeper, HDFS, or a hardware issue.

- `HDFS_DISK_SPACE`

  Msg: `HDFS has less than {{.Perc}}% space free`

  Raised when a HDFS cluster is low on total available disk space.

- `ZK_INACCESSIBLE`

  Msg: `Zookeeper is not accessible`

  Raised when Zookeeper is inaccessible.

- `PERIODIC_BACKUP_FLAPPING`

  Msg: `Periodic backup failed {{._actual_num_occurrences}} times in last {{._earliest_duration_str}}`

  This alert is raised when a periodic backup failed repeatedly.

- `PERIODIC_SNAPSHOT_FLAPPING`

  Msg: `Periodic snapshot failed {{._actual_num_occurrences}} times in last {{._earliest_duration_str}}`

  This alert is raised when periodic snapshot failed repeatedly.

- `APPLICATION_INVALID_STATE_EXTERNAL`

  Msg: `{{.Service}}.{{.Task}} on {{.Machine}} at location {{.Location}}`

  Raised when Application raises invalid state alert.

The possible alert types are `CRITICAL`, `WARNING`, `ERROR`, and `INFO`. For a full reference, see the Alert code reference [See page 701].

Space Utilization

The **Space Utilization** chart is one of the available charts for you to use when checking the cluster overview. This line chart displays the total capacity and estimated used capacity over time.

The x-axis is by time and the y-axis measures the size in GB. You can zoom in and see daily or hourly utilization data. So, in the **Space Utilization** chart above, the green line shows the amount of capacity in use in the system, while the red line shows the total capacity. An increase in the red line at the end of a time period indicates the addition of extra hardware, resulting in increased capacity.

The query for this answer is the following:

```
day of timestamp
total capacity (gb)
total used space (gb)
daily
last 90 days last 25 hours
total capacity (gb) > 0
total used space (gb) > 0
```

The chart relies on the `TS: Internal Table Wise Capacity WS` worksheet. It tracks total used space, which consists of raw uncompressed data, including replication.

Monthly Active Users

This chart shows the number of active users in the system over the last four months, and during the current month. An active user is defined as a user who has logged in at least one time in the defined time interval, in this case months.

The query for this answer is the following:

```
monthly
last 4 months this month
active users
user != {null}
```

This query relies on the `TS: BI server` worksheet.

Monthly Ad-hoc Searches

Number of ad-hoc searches (queries) issued per month. An ad-hoc query is defined as any search or change to a search that builds a new answer (result). An ad-hoc search can also be generated through SpotIQ or another UI/API interaction.

ThoughtSpot considers all of the following as ad-hoc searches (queries):

- User edits tokens (boxed terms) in the search bar.
- User opens an existing saved answer and makes changes to tokens in the search bar.

- User opens an existing saved pinboard, edits a visualization, and makes change to the search tokens.
- Searches initiated by an API call for data with runtime filters

It is not considered a search (query) in this context if a user opens an existing saved aggregated worksheet and makes changes to its underlying query.

The query for this answer is the following:

```
ad-hoc search
user action = 'answer_pinboard_context' 'answer_saved' 'answe
r_unsaved'
monthly
last 4 months this month
```

This answer relies on data from the `TS: BI Server` worksheet.

Monthly Pinboard Views

Number of times a saved pinboard is viewed by a user. These scenarios are considered pinboard views:

- User opens an existing saved pinboard.
- User opens an embedded pinboard from a URL.
- Pinboard data is accessed using the an API.

These scenarios are not considered pinboard views:

- A user opens SpotIQ tab pinboards.
- A user opens admin tab pinboards.
- The system loads a pinboard on the homepage.
- The system loads the ‘learn how to use ThoughtSpot’ pinboard.

The query underlying this answer is:

```
pinboard views
user action = 'pinboard_embed_view' 'pinboard_tspublic_no_runti
me_filter' 'pinboard_tspublic_runtime_filter' 'pinboard_view'
monthly
last 4 months this month
```

The query uses the `TS: BI Server` data source.

Top Users Last Month

This answer shows the top ThoughtSpot users ranked by number of actions the users performed in the last 30 days. The possible user actions include:

`answer_unsaved`	User makes a change to tokens in the search bar.
`answer_saved`	User opens an existing saved answer and makes changes to tokens in the search bar.
`answer_pinboard_context`	User opens an existing saved pinboard, edits a context viz and makes a change to tokens in the search bar.
`answer_aggregated_worksheet`	User opens an existing saved aggregated worksheet and makes changes to tokens in the search bar.
`answer_upgrade`	Requests made for the sole purpose of upgrade.
`pinboard_view`	User opens an existing saved pinboard.
`pinboard_filter`	User adds, removes or applies values to a pinboard filter.
`pinboard_ad_hoc`	User drills down in a pinboard viz.
`data_chart_config`	Request for new data being generated following a chart config change.
`data_show_underlying_row`	Request to show underlying data for a data row(s).
`data_export`	Request to export data.

`pinboard_tspublic_runtime_filter`	Request to TSPublic/pinboarddata with runtime filters.
`answer_aggregated_worksheet_save`	User updates aggregated worksheet.
`answer_add_new_filter`	User adds a filter using the UI.
`data_show_underlying_viz`	Request to show underlying data for a data row(s).
`answer_view`	User opens an existing, saved answer.
`answer_viz_context_view`	User opens an existing saved pinboard, edits a context viz.
`pinboard_insight_view`	User opens SpotIQ tab pinboards.
`pinboard_admin_view`	User opens admin tab pinboards.
`pinboard_embed_view`	User opens embed pinboard from a URL.
`pinboard_homepage_view`	On loading of homepage pinboard.
`pinboard_learn_view`	On loading learn pinboard.
`pinboard_tspublic_no_runtime_filter`	Request to TSPublic/pinboard data without runtime filters.

The query underlying this answer is:

```
top 10
ranked by user actions
user action != 'invalid'
user != {null}
user
last 30 days today
```

The query uses the `TS: BI Server` data source.

Top Pinboards Last Month

This answer shows the top ThoughtSpot users ranked by number of pinboard actions the user performed in the last 30 days. The possible user actions include:

- User opens an existing saved pinboard (‘pinboard_view’).

- User opens an embedded pinboard from a URL (‘pinboard_embed_view’).
- Pinboard data is accessed using an API (‘pinboard_tspublic_no_runtime_filter’ or ‘pinboard_tspublic_runtime_filter’).

The query underlying this answer is:

```
top 10
ranked by pinboard views
user action = 'pinboard_embed_view' 'pinboard_tspublic_no_runti
me_filter' 'pinboard_tspublic_runtime_filter' 'pinboard_view'
pinboard != {null}
pinboard
last 30 days today
```

The query uses the `TS: BI Server` data source.

Configuration Events

This system answer displays recent events that changed the configuration of the system. This panel displays configuration events related to:

Cluster Configuration	Reports configuration actions from the `tscli` and `tql` commands.
Metadata Management	Events related to metadata such as column names, column visibility, column and data definition, column rank and so forth.
User Management	Events related to creating, updating, or adding new users and groups.

For a more detailed list, including the user that issued a command, you can use the `tscli event list` command. Administrators can `ssh` into the cluster and specify a time period or even a type of command to include.

```
[admin@testsystem ~]$ tscli event list --since 3d
+-------------------------------+-----------------------
+----------------------------+
|             DATE              |       USER
|           SUMMARY          |
+-------------------------------+-----------------------
+----------------------------+
| 2018-03-06 11:57:10 -0800 PST |eadmin@thoughtspot.int | User
Management: User      |
|                               |                       | "gues
t_1" updated          |
| 2018-03-06 11:48:10 -0800 PST |admin                  | tscl
i node ls              |
| 2018-03-06 11:17:04 -0800 PST |eadmin@thoughtspot.int | Metad
ata Management:       |
|                               |                       | Metad
ata object "Number of |
|                               |                       | Oppor
tunity                |
|                               |                       | AE an
d Stage" of type      |
...
```

About deprecated boards

There are a number of deprecated boards on this page. They are there to support older installations that relied on them. New installations, should not use or rely on deprecated boards. Older installations that have used these boards in some way, should use the new boards and remove any dependencies.

Related information

`tscli logs` command [See page 0]

Data board

The **Data** page shows all the stored tables with details on the last update time, time taken for auto-indexing, number of rows, and so forth.

Table Information

DATABASE	USER SCHEMA	NAME
FalconUserDataDataBase	FalconUserDataSch..	USERDATA-a6c0991e-462d
FalconUserDataDataBase	FalconUserDataSch..	USERDATA-96a40275-7427
FalconUserDataDataBase	FalconUserDataSch..	USERDATA-bcda2191-cd6c
FalconUserDataDataBase	FalconUserDataSch..	USERDATA-9dc1bfc7-2d27-
FalconUserDataDataBase	FalconUserDataSch..	USERDATA-29546f4d-84ac-
thoughtspot_analytics	falcon_default_sche..	candidates
FalconUserDataDataBase	FalconUserDataSch..	USERDATA-4343525d-261b
FalconUserDataDataBase	FalconUserDataSch..	USERDATA-0694fd57-fadf-
FalconUserDataDataBase	FalconUserDataSch..	USERDATA-51435761-0aac
FalconUserDataDataBase	FalconUserDataSch..	USERDATA-0fb9daec-5230-
dw	falcon_default_sche..	fact_lead_transitions
FalconUserDataDataBase	FalconUserDataSch..	USERDATA-12b3cf23-0de5-

(showing rows 1 - 14 of 913.)

You can click a column name to sort by table name. This table information is based on an underlying system data, you can present it or copy the link, but you cannot access the underlying query.

Database Status

The **Database Status** column can have the following possible values:

Status	Meaning
`READY`	The data has been loaded.
`IN PROGRESS`	The data is still being loaded.
`STALE`	The data is not up to date.
`ERROR`	The table is invalid. Call Customer Support.

Replication Status

The **Replicated** column indicates if the table has been replicated or sharded. This can be used in conjunction with the **Total Shards** column to see how your data is distributed.

If the table is replicated on a multi node system, the **Used Capacity (MB)** column will indicate the total space used on all nodes. a 10MB table replicated on a 4 node system will show 40MB used capacity for example.

Search Status

The **Search Status** column can have the following possible values:

Status	Meaning
`READY`	The data is up to date and searchable.
`NOT READY`	The data is not ready to be searched.
`DELETING INDEX`	The table has already been deleted, but the index still exists due to the latency between the database and search engine.
`INDEXING DISABLED`	Either too many tokens exist in a column for it to be indexed, or indexing has been disabled manually.

Status	Meaning
`CREATING INDEX`	The index is being created.
`UPDATING INDEX`	A change has been made to indexing or the data, and the index is being updated to reflect it.

Cluster Manager board

The **Cluster Manager** section show detailed information about a cluster including latency over time, snapshot status, installed release, node functions, and logs.

Only the **Average Latency Last 2 Weeks (sec)** panel contains a standard ThoughtSpot visualization. The other visualizations rely on internal queries that are not accessible. You can, however, **Present** or **Copy link** on them.

Cluster Details

The fields on this visualization has the following meaning:

Field	Description
Cluster Name	The name of the cluster defined at installation time.
Cluster ID	The ID of the cluster set at installation time.
Last Update Time	Last time the cluster was updated.
Release	Version of the current release.
Zookeeper Servers	IP addresses of the Zookeeper servers.

HDFS Name Nodes	Control nodes for Hadoop Distributed File System (HDFS).

Cluster Logs

The fields on this visualization has the following meaning:

Field	Description
Time	A timestamp indicating when an action occurred.
Type	Type of action.
Release	Identifies the full release number.

You can also use the `tscli logs` command to review log data from your cluster.

List of Snapshots

This visualization shows the snapshots *and the backups* taken on the cluster. The fields on this visualization has the following meaning:

Field	Description
Time	A timestamp indicating when a snapshot or backup happened occurred.
Name	Name of the snapshot file. These files are stored in the `/usr/local/scaligent/backup` directory on your cluster.
Reason	Identifies the reason the snapshot/backup was created. You should see several period snapshots if your cluster is configured properly. You may also see evidence here of manual backups. For example, you should be sure your cluster is backed up before major events such as upgrades. Email support@thoughtspot.com [See page 0] if you don't see evidence your cluster is periodically creating snapshots.
Size	Size of the backup in gigabytes.

Average Latency Last 2 Weeks (sec)

This visualization relies on the `TS: BI Server` worksheet to display the average database latency over the last 15 days. The database latency measures how long it takes for a search to return data from ThoughtSpot - this does not include the time taken to send the answer back to the client, it measures internal processing time. You can use the visualization menu to drill down to its underlying query:

```
average datacache (sec)
average total (sec)
daily
last 15 days
for database latency (us) > 0
```

Related information

`tscli logs` command [See page 0]

Alerts and Events board

The **Alerts and Events** section shows notifications, alerts, and an audit trail of cluster configuration changes.

Alerts

The fields on this answer have the following meaning:

Field	Description
Time	When the alert was sent.
Type	The ID of the event.
Message	The text of the alert message.

For a full reference of possible alerts, see the Alert code reference [See page 701].

Configuration Events

This system answer displays recent events that changed the configuration of the system. This list can contain the same types of information available on the **Admin System Health > Overview** page. This answer displays the **Time**, the **User** that performed the action, and a **Summary** of the action.

Notification events

This answer displays notifications of data loads. The display the **Time**, the **User** that performed the action, and a **Summary** of the action. Notifications are kept for 90 days before being discarded.

Related information

Alert code reference [See page 701]

System worksheets

Most of the monitoring information in **System Health** are sourced from system worksheet which administrators can view, but not modify. The underlying tables are protected system tables that cannot be accessed directly. However, administrators can create new, custom monitoring reports from the worksheets.

List the system worksheets

To list the system worksheets:

1. Go to the **Data** tab.
2. Choose **All** and **Worksheets**.
3. Enter `TS:` in the search field.

Summary of the worksheets

Worksheet	Description
`TS: BI Server`	Contains data related to the systems associated with underlying the ThoughtSpot BI server. This includes database latency, browser clients, size of responses, and more.
`TS: Database`	Contains information related to the database cache and queries run on the database. For example, you could use this worksheet to see data on the query errors returned by the database.

`TS: Internal Capacity WS`	Describes cluster memory capacity by node over time.
`TS: Internal Table Wise Capacity WS`	Describes memory capacity by node, table name, shard count, and CSV replication over time.
`TS: Metrics`	Contains metrics by cluster and host over time.
`TS: Search`	Contains data related to the number of searches (queries) run in the system. This contains information such as uptime, host, and timestamps.
`TS: Service Resources`	Contains data related to cluster nodes including page faults, memory usage, memory failures, and more.
`TS: Table Info`	Describes the named tables by timestamp, row count, row shards, and row size.
`TS: Table Row Counts`	Describes the named tables by timestamp and row count.
`TS: Table Row Counts and Shards`	Describes the named tables by timestamp, row count, row shards, and row size.
`TS: Table Row Size`	Describes the named tables by timestamp and row size.
`TS: Table Shards`	Describes the named tables by timestamp and shard count.

System pinboards

There are several system monitoring pinboards in ThoughtSpot that provide answers for system status and resource usage questions. The information in these pinboards are updated hourly from internal data sources that collect monitoring statistics.

Only users with administrative privilege can view the monitoring pinboards. They are based on worksheets, which administrators can view, but not modify. However, you can create new monitoring pinboards from the worksheets.

List system pinboards

To list these system pinboards:

1. Go to the **Pinboards** tab.
2. Choose **All**.

3. Enter `TS:` in the search field.

Summary of system pinboards

Pinboard	Description
`Learn how to use ThoughtSpot`	Contains search replays created in the system.
`TS Stats: Alert Detail`	Combines alerts, notification events, and configuration events boards.
`TS Stats: Cluster Detail`	Contains cluster details, logs, snapshots, and latency data.
`TS Stats: System Information and Usage`	Replicates the **Admin > System Health > Overview** page.
`TS Stats: Latency Visualizations`	Latency on servers and ***impression [See page 404]*** counts. This data comes from the `TS: BI Server` worksheet.
`TS Stats: Table Status`	Replicates the **Data** board.

About deprecated boards

The deprecated boards (`TS status: Usage - Deprecated` and `TS StaTS: Queries -- Deprecated`) are there to support older installations. New installations, should not use or rely on deprecated boards. Older installations that have used these boards in some way, should use the new boards and remove any dependencies.

About troubleshooting

The information here provides very basic troubleshooting. For more detailed troubleshooting, Contact ThoughtSpot [See page 0].

- **Get your configuration and logs [See page 407]**

  For troubleshooting on specific incidents or cluster problems, getting a log bundle can help.
- **Upload logs to ThoughtSpot Support [See page 411]**
- **Network connectivity issues [See page 413]**

  If network connectivity to and from ThoughtSpot is not working, try using these steps to find and correct the issue.
- **Check the timezone [See page 414]**

  ThoughtSpot comes configured with the timezone where it is to be installed.
- **Browser untrusted connection error [See page 415]**

  If you are not using a SSL certificate for authentication, users will see an untrusted connection error in their browser when accessing ThoughtSpot. The error looks slightly different depending upon the Web browser being used.
- **Characters not displaying correctly [See page 416]**

  Your CSV files are more likely to load smoothly if they are encoded with UTF-8. If you're having problems with some characters rendering incorrectly, you can convert the files to UTF-8 encoding before loading the data.
- **Clear the browser cache [See page 417]**

  You might occasionally see unexpected behavior that is due to the Web browser caching information from ThoughtSpot. In this case, clearing the browser cache and reloading the page should resolve the problem.
- **Cannot open a saved answer that contains a formula [See page 420]**
- **Data loading too slowly [See page 423]**

  Some tables may take an unusually long time to load due to a high data version issue. This issue normally arises when ThoughtSpot completes an upgrade or the system is recovering after a crash.
- **Search results contain too many blanks [See page 424]**

  If you find that your search results contain too many blanks when your data source is a worksheet, there is a simple adjustment you can make to fix this.

Get your configuration and logs

For troubleshooting on specific incidents or cluster problems, two things are important. Understanding your current configuration and getting a log bundle.

Check your configuration

1. Log into the ThoughtSpot cluster as the `admin` user.
2. Use the `tscli feature` subcommand to display your current configuration.

```
 $ tscli feature get-all-config
 +---------------------------------+----------
+---------------+
 |              NAME               |  STATUS  | CONFIGUR
ATION |
 +---------------------------------+----------
+---------------+
 | Firewall                        | Disabled
|               |
 | Saml                            | Disabled
|               |
 | Ldap                            | Disabled
|               |
 | CustomBranding                  | Disabled
|               |
 | CustomBrandingFontCustomization | Disabled
|               |
 | DataConnect                     | Disabled
|               |
 | RLS                             | Enabled
|               |
 | Callhome                        | Enabled
|               |
 | SSHTunnel                       | Enabled
|               |
 | Fileserver                      | Disabled
|               |
 +---------------------------------+----------
+---------------+
```

How to get logs

There are two ways to get logs:

- When ThoughtSpot encounters a problem, a red bar displays in the browser with an error message. You can click **What Happened?** in the error message for more details. To download related logs, click **Download Trace**. Send the logs as an email attachment to the support contact that is provided. Clicking **Report Problem** will also send the logs as an email attachment to your administrator.

- You can generate a log bundle using the `tscli` command `tscli logs collect` if you are comfortable with Linux. The command lets you specify which logs to collect and from what time periods.

  Usage for this command is:

```
tscli logs collect
  --include <selector | glob>
  [--exclude <selector | glob>]
  [--since <hours,minutes,days>
  | --from <yyyymmdd-HH:MM>
  --to <yyyymmdd-HH:MM>]
  [--out <path>]
  [--maxsize <size_in_MB_or_GB>]
  [--sizeonly]
```

  The full list of all selectors is:

- `all` collects all of the logs listed from the system and the ThoughtSpot application.
- `system` collects all system logs, e.g. syslog, upstart, mail logs, etc.
- `ts` collects all logs from the ThoughtSpot application. This includes falcon, sage, orion core (cluster management), etc.
- `orion` collects all orion logs including cluster management, hdfs, zookeeper, etc.

Detailed syntax and options are listed in the tscli command reference [See page 621].

Examples

Here are some examples of usage for the command `tscli logs collect`:

To collect all logs from the past day to the default location (`/tmp/logs.tar.gz`):

```
$ tscli logs collect --include all --since 1d
```

In this example, `all` is a selector for all the available logs.

In most cases, you can probably use the selector `ts` to only capture logs for the ThoughtSpot application:

```
$ tscli logs collect --include ts --since 2d
```

For debugging cluster management issues, use a command like this one, which collects logs for system and orion from the past 2 hours. The output is written to /tmp/debug.tar.gz as specified using `--out`:

```
$ tscli logs collect --include system,orion --since 2h --out /t
mp/debug.tar.gz
```

This command collects logs from a specific time window:

```
$ tscli logs collect --include system,orion --from 20150520-1
2:00 --to 20150522-12:30
```

Advanced usage alert! You can also use `--include` and `--exclude` to specify filesystem paths as a glob pattern. This works like the Linux find(1) command. Pass all the entries in `--include` starting with `/` to find(1), and all entries in `--exclude` which are not selectors to find(1) using the `-not -path` flag.

```
$ tscli logs collect --include system,orion --exclude *hadoo
p*,*zookeeper* --since 2h
```

The above command collects all system and all orion logs, but excludes hadoop (hdfs) and zookeeper logs. See Upload logs to ThoughtSpot Support [See page 411] about using a secure file sever to collect log files or other files needed for troubleshooting. You can easily send log files to this file serve directly from the ThoughtSpot instance.

Upload logs to ThoughtSpot Support

ThoughtSpot Support uses a secure file sever to collect log files or other files needed for troubleshooting. You can easily send log files to this file serve directly from the ThoughtSpot instance.

Metrics collection

ThoughtSpot collects the diagnostic information from your system on an ongoing basis: there is no time needed to collect diagnostic information after a problem is reported. These metrics allow for:

- Our support team can begin working to remediate any issue with you immediately.
- Metrics provides direct visibility to the ThoughtSpot team on your system's limits. Therefore, our Support team can proactively identify critical threshold issues and work to prevent failures. Metrics also help reduce SLA times as the team can debug much faster.
- ThoughtSpot can tune search algorithms by studying search history and schema.
- ThoughtSpot analyzes expensive and complex query patterns to look for performance optimizations.

Finally, the metrics pipeline allows ThoughtSpot to identify application-use patterns that contribute to performance bottlenecks with specific browsers and help your team prevent or alleviate them.

Other log uploads

Before you can upload a file to the secure file server:

1. Configure the connection to the file server [See page 60].
2. Obtain the directory path on the file server.

The server directory path for uploading a file is formatted like this example: `/Shared/support/<customer_name>`. If you do not know the customer name, contact ThoughtSpot Support [See page 0].

You can upload files directly to the file server using this procedure:

1. Log in to the Linux shell using SSH.
2. Navigate to the directory where the file to be uploaded is located.
3. Issue the command to upload the file, specifying the file name and directory path:

```
 $ tscli fileserver upload --file_name <file> --server_d
ir_path <path>
```

When your upload succeeds, you will see a confirmation message.

Network connectivity issues

If network connectivity to and from ThoughtSpot is not working, try using these steps to find and correct the issue.

To troubleshoot network connectivity for ThoughtSpot:

1. Make sure that the network cables are connected correctly.
2. Check that the network cable is connecting the nodes to the network switch.
3. Try replacing the cable with a cable from a known working system to rule out a bad cable or switch connectivity issues.
4. Make sure the eth0 interface is connected to the network by issuing: `ethtool eth0` The port that's currently connected will have "link detected" in the last line of the output.
5. If the networking settings have been reconfigured, reboot each of the nodes.

Check the timezone

ThoughtSpot comes configured with the timezone where it is to be installed. Data is imported based on the timezone of the node from which `tsload` or `tql` is run. To see the timezone your ThoughtSpot node is running under, log into the server as the `thoughtspot` user and run the `date` command:

```
[thoughtspot@ts.server etc]$ date
Tue Feb 20 09:07:04 PST 2018
```

To see the current timezone used by the ThoughtSpot application, choose **Admin > Cluster Manager** and review the **Cluster Details**:

Cluster Details

Property	Content
Cluster Name	local
Cluster ID	local
Release	dev
Last Update Time	Feb 16, 2018, 4:14:31 AM PST
Zookeeper Servers	172.18.248.8:2181
HDFS Name Nodes	172.18.248.8:8020

The timezones should match.

Sometimes the timezone that is listed by `date` is not the active timezone for the ThoughtSpot application and the application needs to be reset. If you need to change the timezone, contact ThoughtSpot Support [See page 0] and they will change the timezone for you. You may need to change the timezone if you move the server between data centers.

Browser untrusted connection error

If you are not using a SSL certificate for authentication, users will see an untrusted connection error in their browser when accessing ThoughtSpot. The error looks slightly different depending upon the Web browser being used.

ThoughtSpot uses secure HTTP (the HTTPS protocol) for communication between the browser and ThoughtSpot. By default there is no SSL certificate for authentication. This must be added by the site administrator. If the site administrator has not added the certificate, the browser warns the user.

Browser	Warning
Google Chrome	The site's security certificate is not trusted!
Mozilla Firefox	This Connection is Untrusted

If you see the warning message, choose one of the following options:

- Ask the site administrator to install the certificate.
- Ask the site administrator to turn off SSL using this command in the shell on the ThoughtSpot instance:

```
$ tscli ssl off
```

- You can choose to ignore the message, and access ThoughtSpot without SSL.

Characters not displaying correctly

Your CSV files are more likely to load smoothly if they are encoded with UTF-8. If you're having problems with some characters rendering incorrectly, you can convert the files to UTF-8 encoding before loading the data.

You might see unexpected characters in your data, especially characters whose ASCII values are at the high and low end of possible values. Some examples of characters that can appear incorrectly are: æ, ñ, ä, í, ö.

If this happens, your data will look like this:

berguvsv[?]gen
vinb[?]ltet

Instead of displaying correctly like this:

berguvsvägen
vinbæltet

To encode your data as UTF-8:

1. On Windows, open your CSV file in Notepad. Save the file as CSV with the Unicode option.
2. On Linux or MacOS, issue a command like:

```
$ iconv -f -t UTF-8 <in_file>.csv > <out_file>.csv
```

3. Reload the data.
4. Attempt to import it again.

Clear the browser cache

You might occasionally see unexpected behavior that is due to the Web browser caching information from ThoughtSpot. In this case, clearing the browser cache and reloading the page should resolve the problem. You can usually resolve these situations by clearing the browser cache:

- During a ThoughtSpot session, the browser suddenly displays a white screen and reloading does not fix the problem. This is due to a self-signed SSL certificate that has timed out during the session.
- When accessing the Help Center, you see a login screen. This is due to a problem during automatic authentication in the Help Center, after which the bad login gets cached by the browser.

To resolve any of these situations, clear the browser cache:

1. Clear the browser cache.

   This works a little differently on individual browser versions and operating systems. For example, when using Chrome, to get to the browser cache settings, navigate to:

   ```
   chrome://settings/clearBrowserData
   ```

2. Click **Clear browsing data**.

   This is the name of the button on Chrome. The name may vary slightly on other browsers.

3. Reload the page.

   For example, on Chrome you would click the **Reload** icon:

Now the problem should be fixed, and the page will appear as expected.

Cannot open a saved answer that contains a formula

When working with formulas, keep in mind the data types they return. You may occasionally see unexpected results, or even be unable to open a saved answer, due to problems with data types and formulas.

In this scenario, “data type” refers the data type as defined in the column definition when creating the schema (INT, TIMESTAMP, VARCHAR, etc.).

When you define a formula, both the data type it returns is set automatically. This can lead to problems, if you build another formula that uses the output of the first formula as input. This can be hard to understand, so an example will be helpful.

Suppose you have created a worksheet that contains a formula called “weekday” defined as:

```
day_of_week(date)
```

The output of that formula is the day of the week (Monday, Tuesday, etc.) returned as a text string (VARCHAR, ATTRIBUTE).

Then suppose you create an answer using the worksheet as a source. And in the answer, you create another formula on top of the formula column in the worksheet. This formula is supposed to return the day of the week that is two days after the given day of the week:

```
weekday + 2
```

In this case, you have effectively created a formula on top of another formula. This works fine, so long as the data types in the worksheet formula can work in the answer formula. If not, you may not be able to save the answer, or open it after it has been saved. Here, the second formula you created does not work, because it is invalid. It is trying to subtract a number from a text string.

If you encounter this issue, you must open the worksheet and edit its formula so that it returns the type expected by the formula that was built on top of it. In this case, a numeric data type.

You must change the underlying worksheet column to use day_number_of_week instead of day_of_week. This is because day_number_of_week returns a numeric data type.

Here are the steps to resolve an issue like this:

1. Open the underlying worksheet that contains the formula whose output data type you need to change.
2. Click the formula name to edit the formula.

3. In the Formula Builder, modify the formula, so that it returns the expected data type.

   There are data type conversion formulas available to make this easier. To view them and their syntax, open the **Formula Assistant**, and expand the section called **Conversion**.

4. Make your changes, and saving the formula by clicking **Save**.
5. Save the worksheet by clicking **Save**.
6. Now you will be able to open the answer that was created on top of the worksheet.

Data loading too slowly

Some tables may take an unusually long time to load due to a high data version issue. This issue normally arises when ThoughtSpot completes an upgrade or the system is recovering after a crash.

The data version is the number of loads that have been historically applied to a table. Every completed load increments the version number of the table by one. ThoughtSpot would need to process each version of the table during restoration, which could increase the time it takes to build the table.

There are a few steps you can take to check for a high data version issue and fix it. To improve data loading speed, run the following command to find the number of tables that are building and their names

```
tscli cluster status --mode table
```

You may notice that a few small tables are taking up a lot of time to be built. This could simply be due to the deceivingly large size of the table. There is also the chance it could be due to a high data version issue. To determine if this is a high data version issue, check the size of the table by running the following command:

```
echo 'show statistics for server;' | tql
```

If there is a large number of rows in the table, go on to shard the table. If the table has a small number of rows, then the slow loading speed is caused by a high data version issue, and you do not have to shard the table. Use the compact table functionality to trim the table down to its actual size:

```
tql> compact table <table name>;
```

Search results contain too many blanks

If you find that your search results contain too many blanks when your data source is a worksheet, there is a simple adjustment you can make to fix this.

If you find that the charts and tables built on a worksheet contain a large number of null values (which display as {blank} in the web browser), you can fix this by changing the inclusion rule [See page 0] for the worksheet.

An inclusion rule that specifies **Exclude Empty Rows (Inner Join)** will reduce the number of null values in the result. Imagine a worksheet that includes data about a retail grocery store. There are rows in the worksheet from these source tables:

Table Name	Description
sales	Fact table with sales made per product per store.
products	Dimension table with information about every product.
stores	Dimension table with information about every store.

When creating the worksheet, suppose you choose **Include Empty Rows (Left Outer Join)** for the inclusion rule and **Progressive Joins** for the join rule. In this case, if you type “product name” in your search, you can see a list of all the products that exist. Suppose you then add “store name” to your search. You will see a lot of null ({blank}) values in the result. This happens because the columns “store name” and “product name” are joined through the fact table, “sales”. So for every product that has never been sold in a particular store, you can see {blank} in the “store name” column. This may be what you want to see, in which case, you can leave the worksheet as is, and choose **Exclude** for the {blank} values in your table or chart, whenever you don’t want to see them.

However, in may cases, including all the {blank} values could confuse end users. So if you encounter this problem, you can edit the worksheet, and change the inclusion rule [See page 341] to **Exclude Empty Rows (Inner Join)**. Now when searching for “store name” and “product name” on the worksheet, users will not be overwhelmed by null values. They’ll only see the rows where a particular product has been sold in a particular store.

Overview

You can install a ThoughtSpot cluster on a hardware appliance, cloud service, or VMware appliance. Your ThoughtSpot installation cannot mix node types. For example, you can have either hardware or VMware nodes, but not both. You can, however, have a cloud cluster for development and use an appliance for production.

This guide instructs you how to prepare each of the following:

- Hardware appliance [See page 426]
- Amazon Web Services (AWS) EC2 [See page 444]
- Microsoft Azure [See page 454]
- Google Cloud Platform (GCP) [See page 466]
- VMware [See page 479]

After you configure your nodes, you can contact ThoughtSpot Support [See page 500] by phone, mail, email, or by filing a support ticket.

Hardware appliance overview

Summary: What is in the box.

The ThoughtSpot appliance hardware will be installed in a rack in your data center. This section describes the typical physical configuration.

Hardware provided by ThoughtSpot

When your ThoughtSpot appliance arrives, the following items will be included:

Item Name	UOM	Qty
Round Hole to Sq Hole Adapter Kit (For Slide Rail Management)	Each	1
Power Cord, C13 to C14, 6 feet	Each	2
Power Cord, C13 to NEMA 5-15, 6 feet[1] **This power cord is not included with the Haswell platform.**	Each	2
Document, Rack Rail Installation, TS-2000	Each	1
TS-2000 Quick Start Guide	Each	1
Bezel Assembly, TS-2000	Each	1
Slide Rail Kit	Each	1
Appliance (containing 1-4 nodes, depending on ordered configuration)	Each	1
SFP+ Connector per ordered node (data connection)	Each	1
5m Fiber cable per ordered node (data connection)	Each	1
5m Network cable per ordered node (management connection)	Each	1

1: The supply voltage, 120 VAC, available when using a NEMA-15 power cord is an insufficient input to achieve the full power output required by the Haswell power supply. Only the C13 to C14 power cord should be used with the Haswell platform.

Additional hardware requirements

You must supply the following items, as they will not be included with your ThoughtSpot appliance:

- Data center with proper cooling
- 2U of rack space per appliance (post depth 26.5” - 36.4”)
- AC power **Attention:** Refer to Hardware details [See page 429] for power input requirements.
- 10GbE infrastructure (switch) - 1x port required / node
- 100MbE infrastructure (switch) - 1x port required /node
- Network cable Cat 5e/6 (node management)[1]
- 10G connection: SFP+ for switch side[2]

1. One 5m CAT 5e/6 network cable, per node, is provided with the appliance for management port connection. Customer supplied cable can be used if preferred.

2. One SFP+ connector is provided, per node, for the node side data connection. One 5m fiber cable is also provided. The customer must provide switch side SFP+ that is compatible with their switch. Customer supplied DAC cables or fiber cables can be used if preferred.

Typical physical deployment

These diagrams show a physical configuration with three blocks of four nodes each. Your appliance can have 1-4 nodes, depending on the ordered configuration.

Server Rack (42U) Back (Customer Supplied)
10Gb or 1Gb
10GbE Switch
100 MbE Switch
ThoughtSpot Analytic Search Appliances 2RU. (Each with 1-4 nodes. Shown with 4 for illustration purposes.)
Block 1
Block 2
Block n
D
C
B
A
LAN
Data / Node Fabric
Node Management (IPMI)
AC Power (see About the hardware)
10GbE (SPF+) Data Connection 1x/node
100 MbE Management port connection 1x/node

Client
Data/Node Fabric (10 GbE Switch)
Node Management (100 MbE Switch)
1x 10GbE (SFP+) Data per node
1x 100MbE Mgmt per node
D
C
B
A

Supported hardware

Summary: Required and provided installation hardware.

This section lists all required hardware that is needed to successfully install your ThoughtSpot appliance in your data center. Some hardware will be provided with your appliance, while the rest must be provided on-site.

The ThoughtSpot instance hardware is configured for fast data searching and reliability. This overview details the hardware specification and installation. The system is made up of compute nodes, which form a cluster. The 2U system includes up to 4 nodes and can hold up to 1TB of data. This can be scaled out.

Network connection

Before you can access ThoughtSpot, you need a network connection.

Refer to Network Ports [See page 66] in the Administrator's Guide to see which ports must remain open to outside traffic for handling certain network requests and for inter-cluster communication. The Administrator's Guide [See page 8] also provides information on network security and how to test your network connectivity between nodes.

Here are some more details on ports and node communication:

- Port redundancy (bonding) is not supported. Only one 10G port is active per node.
- Nodes communicate with each other through the 10G connection (data ports).
- All nodes should be on the same VLAN – ideally connected to the same top of rack switch.
- IPMI ports are used for management functions of the nodes.

Appliance hardware platforms

You can deploy the ThoughtSpot Analytical Search engine on Haswell appliance hardware platforms, with the following specifications:

Details	Haswell
Dimensions	2 RU chassis (17.25 x 3.47 x 28.5 in.)
# of nodes	Populated with 1 to 4 nodes
Node specifications	Each node is independent and consists of a server board (removable from rear), 1x 200GB SSD, 3x 2TB HDD
Max power consumption	2000 W
Required power input	200-240 / 11.8 - 9.8A / 50-60Hz

Note: ThoughtSpot deployments are no longer offered on Ivy Bridge platforms.

Chassis views

These diagrams show the front and rear chassis views. The marked features are present on all four nodes on the rear of the chassis even though they are only pointed out on one node in the diagrams.

The chassis appear fully populated (4-nodes). Your appliance may be populated with 1-4 nodes, depending on the ordered configuration. If less than 4-nodes were ordered, the empty slot will be filled with a filler panel.

Control Panel - Node B
Control Panel - Node D
Control Panel - Node A
SATA Drives (24)
Control Panel - Node C
IPMI LAN Port
Ethernet Ports
R
L
USB Ports
VGA Port

Control Panel - Node B
Control Panel - Node D
Control Panel - Node A
SATA Drives (24)
Control Panel - Node C
IPMI LAN Port
Zero Slot
Ethernet Ports
R
L
USB Ports
VGA Port

Cable networking

This section reviews the types of cables needed for 10GbE networking and how to plug them in. There are three types of cables to consider for 10GbE networking:

- Fiber
- Direct Attach Copper (DAC)
- Category 6a (not supported by ThoughtSpot)

Option 1 - Fiber cables

Fiber can be run long distances to the switch.

These cables require gigabit interface converters (GBICs), SFP+ form factor.

Remember: ThoughtSpot does not supply cables or GBICs

The GBIC must be plugged into a data port on the back of the appliance before plugging in the fiber cables.

The fiber cables must then be plugged into the GBIC.

Option 2 - DAC/Twinax cables

Copper can only be run short distances to the switch. An SFP+ is attached to the cable.

Here is how you would plug in a DAC cable.

Non-option - Category 6a cables (not supported by ThoughtSpot)

There are no adapters for these cables. The 10GbE NIC (Network Interface Card) used on the ThoughtSpot appliance is not compatible with this type of cable/connection.

Setup and start the appliance

This section explains how to install and start the appliance.

Before you begin

The ThoughtSpot appliance comes pre-installed with all the required software. Network settings on the appliance are required prior to using the appliance. Reference ThoughtSpot's site survey for the information specific to the customer's network environment that is required to configure the appliance.

- If ThoughtSpot's site survey form was completed and returned to ThoughtSpot prior to the appliance being shipped, the appliance may be pre-configured for your network environment and ready to install and connect to your network.
- If the network configuration was not pre-set, then this step must be done as part of the installation process.
- If assistance is needed to determine the configuration status of the appliance, please contact ThoughtSpot Support.

Installation procedure

To install and start the appliance and connect to your network:

1. Refer to the Rack Install Guide to install the appliance securely in your data center.
2. Plug in the power cord, but do not turn the appliance on yet. See the figure of the **Location of the power and UID buttons on the control panel** for the power button location.
3. Connect the IPMI dedicated LAN port to a dedicated LAN for system management.
4. Connect the data port(s) on the back of the appliance to your 10GbE network switch. Only the one 10GbE port shown in the following **Back of Node** figures is active. Only one 10GbE port connection is needed.

5. Turn on the appliance by pressing and releasing the power button for each node and allow time for the nodes to boot up completely.

Each node has its own power and UID buttons. Turning the system off using the power button removes the main power, but keeps standby power supplied to the system. Therefore, you must unplug the AC power cord from any external power source before servicing. The power button for each node has a built-in LED which will turn green when the power is on.

There is also a UID button, which is used to turn on or off the blue light function of the LED. After the blue light is activated, the unit can be easily located in very large racks and server banks. A blue LED is also illuminated on the corresponding node, visible from the rear of the chassis.

6. After the appliance has been turned on, verify that both LEDs (IPMI and data NICs) on each network card are lit.
7. Connect a keyboard and monitor to each node in turn. You should see a login prompt on the screen. If you don't see one or the screen isn't responsive, press the key combination control, alt, and F2 on your keyboard, which should allow you to attempt to log in.
8. Log in as username admin, using the default password.
9. Run the following commands and capture the output at every stage:

```
 sudo ipmitool lan print 1
 sudo ipmitool lan set 1 ipsrc static
 sudo ipmitool lan set 1 defgw ipaddr <IPMI_GATEWAY_ADD
R>
 sudo ipmitool lan set 1 netmask <IPMI_VLAN_SUBNET_MASK>
 sudo ipmitool lan set 1 ipaddr <IPMI_NIC_IP_ADDR>
 sudo ipmitool lan print 1
```

10. Run and capture the output of the following commands as well:

```
ifconfig eth0
ifconfig eth1
ifconfig eth2
ifconfig eth3
sudo ethtool eth0
sudo ethtool eth1
sudo ethtool eth2
sudo ethtool eth3
```

11. Share the output of all commands with the ThoughtSpot team who will then determine the next steps.

Cloud overview

ThoughtSpot can currently be deployed in the following cloud provider environments:

- Amazon Web Services (AWS) EC2 [See page 444]
- Microsoft Azure [See page 454]
- Google Cloud Platform (GCP) [See page 466]

The ThoughtSpot cloud deployment consists of cloud compute (VM) instances and an underlying persistent storage layer. The number of instances required for a cloud deployment is based on the size of the data that needs to be analyzed in ThoughtSpot. The instances act as a distributed cluster of nodes to serve query responses.

	AWS	**Azure**	**GCP**
Compute	Virtual machines deployed in your AWS VPC	Virtual machines in your Azure VNET	Virtual machines in your GCP VPC
Persistent storage	Deployment options: 1. Elastic Block Storage 2. S3 + Elastic Block Storage	Premium SSD Managed Disks	Zonal SSD persistent disk

To determine the number of instances and the persistent storage requirements to provision your cluster, please refer to the available instance types for your cloud service provider in the next section.

ThoughtSpot cloud instance types

Refer to the following guidelines for how to set up ThoughtSpot on each cloud service:

- AWS instance types [See page 0]
- Azure instance types [See page 454]

- GCP instance types [See page 466]

Reducing your cloud infrastructure costs

ThoughtSpot recommends following these guidelines to help reduce the cost of your cloud deployment.

Use small and medium instance types when applicable

For ThoughtSpot customers who are deploying their instance with lower data sizes (<=100 GB), ThoughtSpot supports “small” (20 GB data) and “medium” (100 GB data) instance types, as provided at the links above, to help reduce the costs of cloud infrastructure. These are instances with lower CPU/RAM sizes (16/32 vCPU and 128 GB/256 RAM). Advanced lean configuration is required before any data can be loaded onto these instances.

Please contact ThoughtSpot support for assistance with this configuration.

Shut down and restart your cluster

If you do not need your ThoughtSpot cluster to be up and running 24/7, you can shut down your cluster and restart it during normal usage hours to save on the infrastructure costs of running ThoughtSpot instances in cloud provider environments.

To shut down and restart your cluster, do the following in the tscli:

1. Ensure there are no issues with the cluster by running: `$ tscli cluster check`

   The above command should return no failure messages.

2. Stop the cluster by running: `$ tscli cluster stop`

   Wait until you see the message: “Done stopping cluster”

3. Go to your cloud provider’s console and shut down all of the ThoughtSpot VMs in your cluster.

4. When you are ready to use ThoughtSpot again, start up your node VMs.

5. Restart your cluster by running: `$ tscli cluster start`

You should see the message: ”Started pre-existing cluster”

Depending on the size of your cluster, you may need to wait several minutes before the system is up and running. Make sure you budget for this startup time to ensure that the system is fully operational before you expect people to use it.

6. Ensure that your cluster is ready for use by running: `$ tscli cluster status`

   The following messages are displayed to indicate your cluster is up and running:

   `...` `Cluster: RUNNING`

   `Database: READY`

   `Search Engine: READY`

Automating your cloud deployment

You can automate your deployment, using the free tools in the ThoughtSpot Cloud Deployment GitHub repository [See page 0].

For more information about automating your cloud deployment, read Deploying ThoughtSpot in the Cloud Using Terraform and Ansible [See page 0].

AWS configuration options

Summary: Your instances require specific configurations of memory, CPU, storage, and networking capacity.

ThoughtSpot can be deployed in your AWS environment by deploying compute (VM) instances in your Amazon VPC as well as an underlying persistent storage infrastructure. Currently two configuration modes are supported by ThoughtSpot:

- Mode 1: Compute VMs + EBS-only persistent storage
- Mode 2: Compute VMs + EBS and S3 persistent storage

The cost of infrastructure for deploying ThoughtSpot is cheaper when using S3. However, there are differences in where data is loaded, as well as in the backup and restore procedure. For assistance in choosing the best mode for your organization, contact your ThoughtSpot representative. For more information on purchasing ThoughtSpot on AWS, see: ThoughtSpot Pricing [See page 0].

All AWS VMs in a ThoughtSpot cluster must be in the same availability zone (and therefore, also in the same region). ThoughtSpot does not support deploying VMs in the same cluster across availability zones. For more information, see Regions and Availability Zones [See page 0] in Amazon's AWS documentation.

ThoughtSpot AWS instance types

The following sections contain the supported and recommended instance types for a ThoughtSpot AWS deployment. When setting up your cluster in AWS, use the information here to select an instance type, configure the number of instances required for the storage you need, and add data volumes to your cluster.

For example: If you were deploying a total cluster data size of 1 TB using the standard r5.16xlarge instance type, you would need 4 instances (VMs), because the per-VM user data capacity of that instance type is 250 GB. If you were deploying EBS-only data volumes, you would need 2x1 TB data volumes per VM.

VMs with EBS-only persistent storage

Per VM user data capacity	Instance type	CPU/RAM	Recommended per-VM EBS volume
20 GB	r4.4xlarge, r5.4xlarge	16/122, 16/128	2X 400 GB
100 GB	r4.8xlarge, r5.8xlarge	32/244, 32/256	2X 400 GB
192 GB	m5.24xlarge	96/384	2X 1 TB
250 GB	r4.16xlarge, r5.16xlarge	64/488, 64/512	2x 1 TB
384 GB	r5.24xlarge	96/768	2X 1.5 TB

VMs with EBS and S3 persistent storage

Per VM user data capacity	Instance type	CPU/RAM	Recommended per-VM EBS volume
20 GB	r4.4xlarge, r5.4xlarge	16/122, 16/128	1x 500 GB
100 GB	r4.8xlarge, r5.8xlarge	32/244, 32/256	1x 500 GB
192 GB	m5.24xlarge	96/384	1x 500 GB
250 GB	r4.16xlarge, r5.16xlarge	64/488, 64/512	1x 500 GB
384 GB	r5.24xlarge	96/768	1x 500 GB

Note: The S3 bucket size is approximately equal to the size of the user data.

Related information

- EC2 instance types [See page 0]
- EC2 pricing [See page 0]
- EBS pricing [See page 0]
- Placement groups [See page 0]

Set up ThoughtSpot in AWS

Summary: After you determine your configuration options, you must set up your virtual machines (VMs) on AWS using a ThoughtSpot Amazon Machine Image (AMI).

Overview of ThoughtSpot setup in AWS

The high-level process for setting up ThoughtSpot in AWS involves these steps:

1. Gain access to ThoughtSpot AMIs.
2. Choose a VM instance configuration recommended by ThoughtSpot.
3. Set up your Amazon S3 bucket (optional).
4. Set up your ThoughtSpot cluster in AWS.
5. Contact ThoughtSpot to finish setting up your cluster.
6. Open the required network ports for communication for the nodes in your cluster and end users.

About the ThoughtSpot AMI

An Amazon Machine image (AMI) is a preconfigured template that provides the information required to launch an instance. You must specify an AMI when you launch an instance in AWS.

To make deployment easy, the ThoughtSpot AMI includes a custom ThoughtSpot image, with the following components:

- A template for the root volume for the instance, such as an operating system, an appliance server, and applications.
- Launch permissions that control which AWS accounts can use the AMI to launch instances.
- A block device mapping that specifies the volumes to attach to the instance when it launches.

The ThoughtSpot AMI has specific applications on a CentOS base image. The AMI includes the EBS volumes necessary to install ThoughtSpot in AWS. When you launch an EC2 instance from this image, it automatically sizes and provisions the EBS volumes. The base AMI includes 200 GB (xvda), 2X400 GB (xvdb), and SSD (gp2). It contains the maximum number of disks to handle a fully loaded VM.

Prerequisites

To install and launch ThoughtSpot, you must have the following:

- Familiarity with Linux administration, and a general understanding of cloud deployment models.
- The necessary AWS Identity and Access Management (IAM) users and roles assigned to you to access and deploy the various AWS resources and services as defined in the Required AWS components section that follows.

  For more information about IAM, see: What Is IAM? [See page 0] in Amazon's AWS documentation.

Required AWS components

- An AWS VPC. For details, see VPC and Subnets [See page 0] in Amazon's AWS documentation.
- A ThoughtSpot AMI. For details, see the next section.
- AWS security groups. For required open ports, see network policies [See page 488].
- AWS VM instances. For instance type recommendations, see ThoughtSpot AWS instance types [See page 0].
- EBS volumes.
- (Optional) If deploying with S3 persistent storage, one S3 bucket dedicated to each ThoughtSpot cluster.

Guidelines for setting up your EC2 instances

- Sign in to your AWS account [See page 0].
- Copy the following ThoughtSpot public AMI which has been made available in N. California region to your AWS region:

  AMI Name: thoughtspot-image-20190718-dda1cc60a58-prod

  AMI ID: ami-0b23846e4761375f1

  Region: N. California

  > ℹ **Note:** The AMI is backward-compatible with ThoughtSpot releases 5.1.x - 5.2.x.

- Choose the appropriate EC2 instance type: See ThoughtSpot AWS instance types [See page

0] for supported instance types.

- Networking requirements: 10 GbE network bandwidth is needed between the VMs. This is the default for the VM type recommended by ThoughtSpot.
- Security: The VMs that are part of a cluster need to be accessible by each other, which means they need to be on the same Amazon Virtual Private Cloud (VPC) and subnetwork. Additional external access may be required to bring data in/out of the VMs to your network.
- Number of EC2 instances needed: Based on the datasets, this number varies. Please check ThoughtSpot AWS instance types [See page 0] for recommended nodes for a given data size.
- Staging larger datasets (> 50 GB per VM), may require provisioning additional attached EBS volumes that are SSD (gp2).

Setting up your Amazon S3 bucket

If you are going to deploy your cluster using the S3-storage option, you must set up that bucket before you set up your cluster. Contact ThoughtSpot Support [See page 0] to find out if your specific cluster size can benefit from the S3 storage option.

To set up an Amazon S3 bucket in AWS, do the following:

1. In AWS, navigate to the S3 service dashboard by clicking **Services**, then **S3**.
2. Make sure the selected region in the top-right corner of the dashboard is the same region in which you plan to set up your cluster.
3. Click **Create bucket**.
4. In the **Name and region** page, enter a name for your bucket, select the region where to set up the cluster, and click **Next**.
5. On the **Properties** page, click **Next**.
6. On the Configure options page, make sure **Block *all* public access** is selected, and click **Next**.
7. On the Set permissions page, click **Create bucket**.

Encrypting your data at rest

ThoughtSpot makes use of EBS for the data volumes to store persistent data (in the EBS deployment model) and the boot volume (in the EBS and S3 deployment models). ThoughtSpot recommends that you encrypt your data volumes prior to setting up your ThoughtSpot cluster. If you are using the S3 persistent storage model, you can encrypt the S3 buckets using SSE-S3. ThoughtSpot does not currently support AWS KMS encryption for AWS S3.

For more information on encryption supported with AWS:

- For EBS, see Amazon EBS Encryption [See page 0] in Amazon's AWS documentation.
- For S3, see Amazon S3 Default Encryption for S3 Buckets [See page 0] in Amazon's AWS documentation.

Setting up your ThoughtSpot cluster

To set up a ThoughtSpot cluster in AWS, do the following:

1. In AWS, navigate to the EC2 service dashboard by clicking **Services**, then **EC2**.

2. Make sure your selected region is correct in the top-right corner of the dashboard. If not, select a different region you would like to launch your instance in. Let ThoughtSpot support know if you change your region.

3. Start the process of launching a VM by clicking **Launch Instance**.

4. Click the **My AMIs** tab, find the ThoughtSpot AMI from the list, and click **Select**.

5. On the Choose an Instance Type page, select a ThoughtSpot-supported instance type. (See ThoughtSpot AWS instance types [See page 0].)

6. Click **Next: Configure Instance Details**.

7. Configure the instances by choosing the number of EC2 instances you need. The instances must be on the same VPC and subnetwork. ThoughtSpot sets up the instances to be in the same ThoughtSpot cluster.

   S3 storage setting: If you are going to use the S3 storage option, you must go to the **IAM role** menu and select **ec2rolewithfulls3access**. This setting gives your instance access to all S3 buckets in your account's region.

   To restrict the access to a specific bucket, you must create a new IAM role that provides access to the specific bucket, and select it instead. For details on that, click **Create new IAM role**.

8. Click **Next: Add Storage**. Add the required storage based on the storage requirements of the instance type you have selected, and the amount of data you are deploying. For specific storage requirements, refer to ThoughtSpot AWS instance types [See page 0].

9. When you are done modifying the storage size, click **Next: Add Tags**.

10. Set a name for tagging your instances and click **Next: Configure Security Group**.

11. Select an existing security group to attach new security groups to so that it meets the security requirements for ThoughtSpot.

    ☑ **Tip: Security setting for ThoughtSpot**

    - The VMs need intragroup security, i.e. every VM in a cluster must be accessible from one another. For easier configuration, ThoughtSpot recommends that you enable full access between VMs in a cluster.
    - Additionally, more ports must be opened on the VM to provide data staging capabilities to your network. Check Network policies [See page 0] to determine the minimum required ports that must be opened for your ThoughtSpot appliance.

12. Click **Review and Launch**. After you have reviewed your instance launch details, click **Launch**.

13. Choose a key pair. A key pair consists of a public and private key used to encrypt and decrypt login information. If you don't have a key pair, you must create one, otherwise you won't be able to SSH into the AWS instance later on.

14. Click **Launch Instances**. Wait a few minutes for it to fully start up. After it starts, it appears on the EC2 console.

Prepare the VMs (ThoughtSpot Systems Reliability Team)

⚠ **Important:** This procedure is typically done by a ThoughtSpot Systems Reliability Engineer (SRE). Please consult with your ThoughtSpot Customer Service or Support Engineer on these steps.

Before we can install a ThoughtSpot cluster, an administrator must log into each VM through SSH as user "admin", and complete the following preparation steps:

1. Run `sudo /usr/local/scaligent/bin/prepare_disks.sh` on every machine.

2. Configure each VM based on the site-survey.

When complete, your storage is mounted and ready for use with your cluster.

Launch the cluster

Upload the TS tarball to one of the VMs, and proceed with the normal cluster creation process, using tscli cluster create [See page 0].

If you are going to use S3 as your persistent storage, you must enable it when running this command, using the **enable_cloud_storage** flag. Example: `tscli cluster create 6.0-167.tar.gz --enable_cloud_storage=s3a`

When the setup is complete, you can load data into ThoughtSpot for search analytics.

Open the required network ports

To determine which network ports to open for a functional ThoughtSpot cluster, see Network policies [See page 488].

Related information

EC2 Best Practices [See page 0]

Loading data from an AWS S3 bucket [See page 146]

Azure configuration options

ThoughtSpot can be deployed in your Azure environment by deploying compute (VM) instances in your VNET as well as an underlying persistent storage infrastructure. Currently we support Premium SSD Managed Disks for persistent storage. For more information, see Managed Disks pricing [See page 0] in Microsoft's Azure documentation.

All Azure VMs (nodes) in a ThoughtSpot cluster must be in the same availability zone (and, therefore, also in the same region). ThoughtSpot does not support deploying VMs(nodes) of the same cluster across availability zones. For more information, see What are Availability Zones in Azure? [See page 0] in Microsoft's Azure documentation.

ThoughtSpot Azure instance types

Per VM user data capacity	Instance type	CPU/ RAM	Recommended per-VM Premium SSD Managed Disk volume
200 GB	E64sv3	64/432	2x1 TB
100 GB	E32sv3	32/256	2X 400 GB
20 GB	E16sv3	16/128	2X 400 GB
120 GB	D64v3	64/256	2X 1 TB

Set up ThoughtSpot in Azure

Summary: After you determine your configuration options, you must set up your virtual machines using a ThoughtSpot image for Azure.

About the ThoughtSpot image

To provision ThoughtSpot in the Azure portal, you'll need to access the ThoughtSpot Virtual Machine in the Azure Marketplace.

The ThoughtSpot Virtual Machine comes provisioned with the custom ThoughtSpot image to make hosting simple. A virtual machine is a preconfigured template that provides the information required to launch an instance of ThoughtSpot. It includes the following:

- A template for the root volume for the instance (for example, an operating system, an appliance server, and applications).

The ThoughtSpot Virtual Machine has the ThoughtSpot software installed and configured, on an CentOS base image. Check with your ThoughtSpot contact to learn about the latest version of the ThoughtSpot Virtual Machine.

Due to security restrictions, the ThoughtSpot Virtual Machine does not have default passwords for the administrator users. When you are ready to obtain the password, contact ThoughtSpot Support.

Set up ThoughtSpot in Azure

Follow these steps to provision and set up the VMs and launch ThoughtSpot.

Prerequisites

Complete these steps before launching your ThoughtSpot Virtual Machine:

1. Obtain an Azure login account.
2. Set up usage payment details with Microsoft Azure.
3. Set up a Resource Group.

Create an instance

To get started, you need to log into the Azure portal, create a resource group, get the ThoughtSpot Virtual Machine [See page 0] on the Azure Marketplace [See page 0], create a resource based on the VM, and complete initial setup. You can either start at the Marketplace or from within the resource group you just created, as described here.

1. Log in to the Azure portal.

   In a browser, go to http://azure.microsoft.com [See page 0], and log in to your Azure account.

2. Create a Resource Group.

3. Next, create a resource based on the ThoughtSpot Virtual Machine.

   a. Click **Create a resource**, search the Marketplace for the ThoughtSpot Virtual Machine, and select it.

b. On the ThoughtSpot Virtual Machine page, click **Create**.

Configure basic settings

1. Provide a name and password for your new virtual machine.

2. Choose a disk type.

   > ☑ **Tip:** the new SSD disk types are currently available for only particular regions, so if you choose this disk type, make sure it's supported on the region you chose for your VM.

   See Standard SSD Disks for Virtual Machine workloads [See page 0] for more on SSD disks.

3. Provide a Resource Group, by clicking `existing` and selecting one.

4. Select a location.

5. Click **OK** to save the Basics, which should look similar to the following example.

Choose a machine size

For **Choose a size**, select `E64S_V3 standard`.

Configure network settings, storage, and other options

Prerequisite: Get the details needed for setting up the Virtual Network, Subnet, and Network Security Group from your Azure support team.

1. For storage, select **Yes** to **use managed disks**.

2. Under **Network**, select **Virtual network**, then **Subnet**, then **Public IP addresses**, and set those names, addresses, and ranges approriately for your network.

3. Open the necessary Inbound and Outbound ports to ensure that the ThoughtSpot processes do not get blocked.

   The minimum ports needed are:

Port	Protocol	Service
22	SSH	Secure Shell access
80	HTTP	Web access
443	HTTPS	Secure Web access
12345	TCP	ODBC and JDBC drivers access
2201	HTTP	Cluster Debugging
2101	HTTP	Node daemon Debugging
4001	HTTP	Data Cache Debugging

   ℹ **Note:** ThoughtSpot requires that nodes purchased from Azure must be reachable to each other so that they can communicate and form a distributed environment. ThoughtSpot only requires that those ports be accessible between nodes within a cluster. Use your discretion about whether to restrict public access or not for all nodes/ all ports

4. Leave other configurations such as `auto shutdown` and `monitoring` on their default settings.

5. Click **OK**.

   Azure will do the final validation check.

Review the Summary

Verify that the validation check succeeded and that summary of information shown is correct. If you find errors, reconfigure as needed.

When you are satisfied with the virtual machine setup, click **Create**.

Prepare for starting up ThoughtSpot

Prerequisite: To log in to the VM, you need the private key that is available in the image. You can obtain this from your ThoughtSpot contact.

1. Obtain the VM's public and private IP addresses.
    - To see the public IP, click the VM name link. This will show the public IP of the VM.
    - To see the private IP click Networking (under SETTINGS on the left side of the screen).

2. Connect to the VM through SSH, using the private key provided for the admin user.
    - You must file a support ticket to obtain this private key; it is necessary for the first login.
    - This key is different from the credentials, or the private keys supplied in earlier steps, which do not work in this context.

3. Update the password for both the `admin` and the `thoughtspot` users.

```
$ sudo passwd admin Changing password for user admin
$ sudo passwd thoughtspot Changing password for user tho
ughtspot
```

4. Update the file `/etc/hosts` with all the node IP addresses for the other VMs that will be part of the ThoughtSpot cluster.

Add Storage Disks

1. Go back to the VM and click it.
2. Add 2 SSD disks of 1TB each.
3. Click **Add data disk** and choose **Create disk from the menu**.
4. Create one mode data disk (demo-disk2) and save them both.
5. Click **Save** to add the disks to the VM.
6. Verify that the disks were added by issuing this command in the shell on the VM:

```
$ lsblk
```

Which returns results like:

```
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
fd0       2:0    1    4K  0 disk
sda       8:0    0  200G  0 disk
├─sda1    8:1    0    1G  0 part /mntboot
├─sda2    8:2    0   20G  0 part /
├─sda3    8:3    0   20G  0 part /update
└─sda4    8:4    0  159G  0 part /export
sdb       8:16   0    1T  0 disk
└─sb1     8:17   0    1T  0 part /mnt/resource
sdc       8:32   0    1T  0 disk
sdd       8:48   0    1T  0 disk
sr0      11:0    1  628K  0 rom
```

7. Unmount the temporary disk by issuing:

```
$ sudo umount /mnt/resource
```

8. Prepare the disks /dev/sdc and /dev/sdd for ThoughtSpot by issuing the command:

```
$ sudo /usr/local/scaligent/bin/prepare_disks.sh /dev/sd
c /dev/sdd
```

> ❗ **Warning:** Do not use the disk /dev/sdb. This is reserved for ThoughtSpot use.

9. Check the disks status by issuing:

```
$ df -h
```

10. Repeat the steps in this section for each node in your cluster.

Make network support settings

> ☑ **Tip:** All changes in this section must be re-applied each time after a cluster is created or updated. If these changes are not present, a reboot of the VMs will not have network access. So when updating these files, keep a backup to copy after any subsequent cluster creation or update.

1. Update hostnames for all the nodes by issuing:

```
$ sudo hostnamectl set-hostname <HOSTNAME>
```

   If you are using a static name, you can issue:

```
sudo hostnamectl set-hostname <HOSTNAME> --static
```

2. Update `/etc/hosts` with the IP and hostname:

```
$ sudo vi /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE=eth0 ONBOOT=yes BOOTPROTO=dhcp HWADDR=<Add eth0 M
AC> TYPE=Ethernet USERCTL=no PEERDNS=yes IPV6INIT=no
```

3. Do not reboot any of the nodes, until these changes are made to each node:

   a. Open the grub file /update/etc/default/grub in an editor:

```
   $ sudo vi /update/etc/default/grub
```

   b. Change the line:

```
   GRUB_CMDLINE_LINUX="console=tty0 console=ttyS1,115200
n8"
```

to:

```
   GRUB_CMDLINE_LINUX="console=tty0 console=ttyS1,115200
n8 net.ifnames=0"
```

c. Save your changes.

4. Issue these commands:

```
$ sudo cp /update/etc/default/grub /etc/default/
$ rm /usr/local/scaligent/bin/setup-net-devices.sh
```

5. Reboot the nodes.

GCP configuration options

ThoughtSpot can be deployed in your GCP environment by deploying compute (VM) instances in your VPC as well as an underlying persistent storage infrastructure. Currently two configuration modes are supported by ThoughtSpot:

- Mode 1: Compute VMs + SSD Persistent Disk storage-only
- Mode 2: Compute VMs + SSD Persistent Disk and Google Cloud Storage (GCS).

For more information about Persistant Storage, see Zonal Persistent SSD disks [See page 0] in Google's Cloud documentation.

For more information about Google Cloud Storage, see Cloud Storage Buckets [See page 0] in Google's Cloud documentation.

All GCP VMs (nodes) in a ThoughtSpot cluster must be in the same zone (and, therefore, also in the same region). ThoughtSpot does not support deploying VMs (nodes) of the same cluster across zones. For more information, see Regions and Zones [See page 0] in Google's Cloud documentation.

ThoughtSpot GCP instance types

VMs with Persistent Disk-only storage

Per VM user data capacity	Instance type	CPU/RAM	Recommended per-VM Zonal Persistent SSD Disk volume
208 GB	n1-highmem-64	64/416	2x 1 TB
312 GB	n1-highmem-96	96/624	2x 1.5 TB
100 GB	n1-highmem-32	32/208	2X 400 GB
20 GB	n1-highmem-16	16/122	2X 400 GB
180 GB	n1-standard-96	96/330	2X 1 TB

VMs with Persistent Disk and Google Cloud storage

Per VM user data capacity	Instance type	CPU/RAM	Recommended per-VM Zonal Persistent SSD Disk volume
208 GB	n1-highmem-64	64/416	1X 500 GB
312 GB	n1-highmem-96	96/624	1X 500 GB
100 GB	n1-highmem-32	32/208	1X 500 GB
20 GB	n1-highmem-16	16/122	1X 500 GB
180 GB	n1-standard-96	96/330	1X 500 GB

Set up ThoughtSpot in GCP

After you've determined your configuration options, set up your virtual machines (VMs). The ThoughtSpot base image for booting the VMs and some other aspects of system setup will be shared with you on GCP by ThoughtSpot.

About the ThoughtSpot and Google Cloud Platform

ThoughtSpot uses a custom image to populate VMs on GCP. The base image is a Centos derived image, which will be available to you in your Google Compute Engine project for Boot disk options under Custom Images.

Ask your ThoughtSpot liaison for access to this image. We need the Google account/email ID of the individual who will be signed into your organization's GCP console. We will share ThoughtSpot's GCP project with them so they can use the contained boot disk image for creating ThoughtSpot VMs.

Overview

Before you can create a ThoughtSpot cluster, you must provision VMs. You use the Google Compute Engine (GCP) platform for creating and running VMs [See page 0].

The following topics walk you through this process.

Prerequisites

1. Ensure that **Network Service Tier** is set to **Premium** for all VMs to be used in your ThoughtSpot cluster.
2. A ThoughtSpot cluster requires 10 Gb/s bandwidth (or better) between any two nodes. This must be established before creating a new cluster.

Setting up your Google Cloud Storage (GCS) bucket

If you are going to deploy your cluster using the GCS-storage option, you must set up that bucket before you set up your cluster. Contact ThoughtSpot Support [See page 0] to find out if your specific cluster size will benefit from the GCS storage option.

1. Sign in to the Google Cloud Console [See page 0].
2. Go to the Storage dashboard.
3. Click **CREATE BUCKET**.
4. Enter a name for your bucket, and click **CONTINUE**.
5. For location type, select **Region** and use the Location drop-down menu to select the region where you are going to set up your instance, and click **CONTINUE**.
6. For default storage class, make sure **Standard** is selected, and click **CONTINUE**.
7. For access control model, make sure **Set permissions uniformly at bucket-level** is selected, and click **CONTINUE**.
8. For advanced settings, leave Encryption set to **Google-managed key**, do not set a retention policy, and click **CREATE**.

When you create your instance, make sure you set Storage to **Read Write** access.

Create an instance

1. Sign in to the Google Cloud Console [See page 0].
2. Go to the Compute Engine dashboard, and select the associated ThoughtSpot project.

3. Select **VM instances** on the left panel and click **CREATE INSTANCE**.

4. Provide a name for the instance, choose a region, choose number of CPUs (e.g., 8 vCPUs for a cluster), and click **Customize** to further configure CPUs and memory.

5. For **Machine type** set the following configuration:

Setting	Value
Cores	`64 vCPU`
Memory	`416 GB`
Extend memory	Enabled (checkmark)
CPU platform	`Automatic` (or select one of the preferred CPU platforms, `Intel Skylake` or `Intel Broadwell`, if available)

6. Configure the Boot disk.

a. Scroll down to the find the **Boot disk** section and click **Change**.

b. Click **Custom Images** on the tabs at the top, select a ThoughtSpot base image and configure the boot disk as follows:

Setting	Value
Image	`ThoughtSpot`
Boot disk type	`Standard persistent disk`
Size (GB)	`250`

> ℹ **Note:** ThoughtSpot updates these base images with patches and enhancements. If more than one image is available, the latest one is always at the top of the list. Both will work, but we recommend using the latest image because it typically contains the latest security and maintenance patches.

c. Click **Select** to save the boot disk configuration.

7. Back on the main configuration page, click to expand the advanced configuration options (**Management, security, disks, networking, sole tenancy**).

8. Attach two 1 TB SSD drives. These drives will be used for the data storage.

   a. Click the **Disks** tab, and click **Add new disk**.

b. Configure the following settings for each disk.

Setting	Value
Type	`SSD persistent disk`
Source type	`Blank disk`
Size (GB)	`1024`

vmb-ts-data-disk (Blank, 1024 GB)
Name (Optional)
vmb-ts-data-disk
Type
SSD persistent disk
Source type
Image
Blank disk
Mode
Read/write
Read only
Deletion rule
When deleting instance
Keep disk
Delete disk
Size (GB)
1024
Estimated performance
Operation type
Read
Write
Sustained random IOPS limit
30,720.00
30,000.00
Sustained throughput limit (MB/s)
491.52
400.00
Encryption
Data is encrypted automatically. Select an encryption key management solution.
Google-managed key
No configuration required
Customer-managed key
Manage via Google Cloud Key Management Service
Customer-supplied key
Manage outside of Google Cloud
This new disk will be added once you create the new instance
Done
Cancel

9. (For use with GCS only) In the Identity and API access section, make sure Service account is set to **Compute Engine default service account**, and under Access scopes, select **Set access for each API**.

10. (For use with GCS only) Scroll down to the Storage setting, and set it to one of the following options:
    - To use Google Cloud Storage (GCS) as persistent storage for your instance, select **Read Write**.
    - To only use GCS to load data into ThoughtSpot, select **Read Only**.

11. Customize the network settings as needed, preferably use your default VPC settings.

12. Repeat these steps to create the necessary number of such VMs.

Prepare the VMs (ThoughtSpot Systems Reliability Team)

> ⚠ **Important:** This procedure is typically done by a ThoughtSpot Systems Reliability Engineer (SRE). Please consult with your ThoughtSpot Customer Service or Support Engineer on these steps.

Before we can install a ThoughtSpot cluster, an administrator must log into each VM through SSH as user “admin”, and complete the following preparation steps:

1. Run `sudo /usr/local/scaligent/bin/prepare_disks.sh` on every machine.
2. Configure each VM based on the site-survey.

Launch the cluster

Upload the TS tarball to one of the VMs and proceed with the normal cluster creation process, using tscli cluster create [See page 0].

If you are going to use GCS as your persistent storage, you must enable it when running this command, using the **enable_cloud_storage** flag. Example: `tscli cluster create 6.0-167.tar.gz --enable_cloud_storage=gcs`

Open the required network ports

To determine which network ports to open for a functional ThoughtSpot cluster, see Network policies [See page 488].

Related information

Connecting to Google Cloud Storage buckets [See page 0]

Loading data from a GCP GCS bucket [See page 148]

VMware configuration overview

Congratulations on purchasing the ThoughtSpot instance. This section is an overview of the ThoughtSpot AI-Driven analytics platform hosted on the VMware vSphere Hypervisor (ESXi) 6.5 environment.

About ThoughtSpot on VMware

The VMware virtualization platform provides highly scalable and efficient memory and CPU resources management that can be used by ThoughtSpot instances. Additionally, the VMware virtualization environment is an easy transition between development and production environments. The following diagram shows the components of a VMware and ThoughtSpot architecture:

Note: This is a generic representation; Only CentOS-based virtual machines are supported with ThoughtSpot.

Your database capacity will determine the number of ThoughtSpot instances and the instance network/storage requirements. In addition, you can scale your ThoughtSpot VMs as your dataset size grows.

Supported configurations

ThoughtSpot Engineering has performed extensive testing of the ThoughtSpot platform on VMware for the best performance, load balancing, scalability, and reliability. Based on this testing, ThoughtSpot recommends the following *minimum specifications* for an individual VMware ESXi host machine:

Per VM user data capacity	CPU/RAM	Data disk
20 GB	16/128 GB	800 GB
100 GB	32/256 GB	800 GB
256 GB	72/512 GB	6 TB
Note: All cores must be hyperthreaded. 200GB SSD boot disk required for all configurations.		

Locally attached storage provides the best performance.

SAN can be used, but must comply with the following requirements:

- 136 MBps minimum random read bandwidth
- 240 random IOPS (~4ms seek latency)

NAS/NFS is not supported since its latency is so high that it tends to be unreliable.

All virtualization hosts should have VMware vSphere Hypervisor (ESXi) 6.5 installed.

ThoughtSpot provides a VMware template (OVF) together with a VMDK (Virtual Machine Disk) file for configuring a VM. VMDK is a file format that describes containers for virtual hard disk drives to be used in virtual machines like VMware Workstation or VirtualBox. OVF is a platform-independent, efficient, extensible, and open packaging distribution format for virtual machines.

The ThoughtSpot VM configuration uses thin provisioning and sets the recommended reserved memory, among other important specifications. You can obtain these files from your ThoughtSpot Customer Success Engineer.

Questions or comments?

We hope your experience with ThoughtSpot is excellent. Please let us know how it goes, and what we can do to make it better. You can contact ThoughtSpot [See page 500] by filing a support ticket, email or phone.

Set up ThoughtSpot in VMware

Summary: Learn how to install a ThoughtSpot cluster in a VMware environment.

This page explains how to install a ThoughtSpot cluster in a VMware VSphere Hypervisor (ESXi) 6.5 environment. For each hardware node, you must:

- Complete the prerequisites
- Use the ThoughtSpot Open Virtualization Format (OVF) file to create a virtual machine (VM)
- Add hard disks to the VM

Prerequisites

This installation process assumes you have already acquired your host machines. You can install on a cluster with any number of nodes. A one node cluster is suitable for a sandbox environment but is insufficient for a production environment. You need at least three nodes for high availability (HA), but there is no limit on the number of nodes.

1. Make sure you have installed the Hypervisor on each of your nodes.

   The VM template, by default, captures a 72-core configuration. If your physical host has more than 72 cores, you may want to edit VM to have (`n–2`) cores (for a physical host with n cores) to fully take advantage of computing power of the physical host. Extra cores help performance.

   You should aim to allocate 490 GB or more RAM.

2. Create datastores for all solid-state drive (SSD) and hard drive devices.

Use the OVF to Create a VM

1. **Download [See page 0]** the `ThoughtSpot OVF` to a location on an accessible disk.

2. Log into the ESXi web portal.

3. Select **Virtual Machines > Create/Register VM**.

   The system displays the dialog for selecting an OVF template.

4. Choose the OVF template and click **Next**.

   The system prompts you to select a storage.

5. Choose the SSD as the destination and click **Next**.

   The system displays the **Deployment Options** dialog.

6. Enter the options and click **Next**.

Setting	Value
Network mappings	Select the correct network for your installation.
Disk provisioning	Choose Thin.
Power on automatically	Check this box.

7. Review your selection and click **Finish**.

8. Wait for the template to be loaded.

   Depending on your network speed, loading can take several minutes.

9. Make sure that VM is powered off.

Add hard disks to the VM

Use Case	HDFS Disk Requirements
POC	2 x 1 TB on HDD
Production	3 x 2 TB on HDD

For a proof of concept (POC), follow these steps to create two 1TB HDFS disks on HDD storage, as shown here (2 x 1TB).

For production deployments, ThoughtSpot requires you to have three 2TB HDFS disks on HDD (3 x 2TB). For this use case, follow these same steps to create the additional, larger capacity disks.

1. Edit the VM you just created.

2. Select **Add hard disk > New hard disk**.

The new disk appears as a new row under the only existing SSD row.

3. Click the **New Hard disk** to expand the detailed configuration options.

4. For a proof of concept, set the options as follows. (For production deployments, set the size to 2TB.)

Setting	Value
size	1 TB
Location	Use the **Browse** button to select the hard disk store.
Thin provisioned	Check this box.

You should see something similar to the following:

5. Save your changes.
6. Repeat steps 1-5 to create more hard disks.
7. Power on the VM
8. After the VM is online, run the following command to prepare the HDFS disks:

```
sudo /usr/local/scaligent/bin/prepare_disks.sh
```

Next steps

There is no network at this point on your VMs. As a prerequisite:

1. Verify that Network Adapter type is set to VMware vmxnet3 (Recommended).
2. Verify that all ESXi hosts in your VMware farm for ThoughtSpot have been trunked to the VLAN assigned to your ThoughtSpot VMs.
3. Verify that the console of all ThoughtSpot VMs is accessible in VMware vCenter Server.

After you finish, go to the ThoughtSpot Support website [See page 0] and use the support ticket for installation tasks. If necessary, create a new ticket.

Network policies

Summary: Lists the required and optional ports for an installation.

For regular operations and for debugging, there are some ports you must keep open to network traffic from end users. Another, larger list of ports must be kept open for network traffic between the nodes in the cluster.

Required ports for operations and debugging

The following ports must be open for requests from your user population. There are two main categories: operations and debugging.

Port	Protocol	Service Name	Direction	Source	Destination	Description
22	SSH	SSH	bidirectional	Administrators IP addresses	All nodes	Secure shell access. Also used for scp (secure copy).
80	HTTP	HTTP	bidirectional	All users IP addresses	All nodes	Hypertext Transfer Protocol for website traffic.
443	HTTPS	HTTPS	bidirectional	All users IP addresses	All nodes	Secure HTTP.
12345	TCP	Simba	bidirectional	Administrators IP addresses	All nodes	Port used by ODBC and JDBC drivers when connecting to ThoughtSpot.
2201	HTTP	Orion master HTTP	bidirectional	Administrator IP addresses	All nodes	Port used to debug the cluster manager.
2101	HTTP	Oreo HTTP	bidirectional	Administrator IP addresses	All nodes	Port used to debug the node daemon.
4001	HTTP	Falcon worker HTTP	bidirectional	Administrator IP addresses	All nodes	Port used to debug the data cache.

Port	Protocol	Service Name	Direction	Source	Destination	Description
4251	HTTP	Sage master HTTP	bidirectional	Administrator IP addresses	All nodes	Port used to debug the search engine.

Network Ports

This reference lists the potential ports to open when setting up your security group.

Required ports for intracluster operation

Static ports are used for communication between services within the cluster. ThoughtSpot recommends that you open all ports within a cluster. This not required, but it will ensure that cluster communication works properly if additional ports are used in a future software release.

If your organization does not allow you to open all ports, make sure you open the required intracluster ports listed in the following table. In addition, a number of ports are dynamically assigned to services, which change between runs. The dynamic ports come from the range of ports that are dynamically allocated by Linux (20K+).

Port	Protocol	Service Name	Direction	Source	Dest.	Description
80	TCP	nginx	inbound	All nodes	All nodes	Primary app HTTP port (nginx)
443	TCP	Secure nginx	inbound	All nodes	All nodes	Primary app HTTPS port (nginx)
2100	RPC	Oreo RPC port	bidirectional	All nodes	All nodes	Node daemon RPC
2101	HTTP	Oreo HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Node daemon HTTP

Port	Protocol	Service Name	Direction	Source	Dest.	Description
2181	RPC	Zookeeper servers listen on this port for client connections	bidirectional	All nodes	All nodes	Zookeeper servers listen on this port for client connections
3181	RPC	Zookeeper servers listen on this port for client connections	bidirectional	All nodes	All nodes	Zookeeper servers listen on this port for client connections
4181	RPC	Zookeeper servers listen on this port for client connections	bidirectional	All nodes	All nodes	Zookeeper servers listen on this port for client connections
2200	RPC	Orion master RPC port	bidirectional	All nodes	All nodes	Internal communication with the cluster manager
2201	HTTP	Orion master HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Port used to debug the cluster manager
2205	TCP	Cluster update service TCP port	bidirectional	All nodes	All nodes	Internal communication with the cluster manager
2210	RPC	Cluster stats service RPC port	bidirectional	All nodes	All nodes	Internal communication with the stats collector
2211	HTTP	Cluster stats service HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Port used to debug the stats collector
2230	RPC	Callosum stats collector RPC port	bidirectional	All nodes	All nodes	Internal communication with the BI stats collector

Port	Protocol	Service Name	Direction	Source	Dest.	Description
2231	HTTP	Callosum stats collector HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Port used to debug the BI stats collector
2240	RPC	Alert manager	bidirectional	All nodes	All nodes	Port where alerting service receives alert events
2241	RPC	Alert manager	bidirectional	All nodes	All nodes	Port where alerting service receives alert events
2888	RPC	Ports used by Zookeeper servers for communication between themselves	bidirectional	All nodes	All nodes	Ports used by Zookeeper servers for communication between themselves
3181	RPC	Ports used by Zookeeper servers for communication between themselves	bidirectional	All nodes	All nodes	Ports used by Zookeeper servers for communication between themselves
3888	RPC	Ports used by Zookeeper servers for communication between themselves	bidirectional	All nodes	All nodes	Ports used by Zookeeper servers for communication between themselves
4000	RPC	Falcon worker RPC port	bidirectional	All nodes	All nodes	Port used by data cache for communication between themselves
4001	HTTP	Falcon worker HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Port used to debug the data cache
4002	HTTP	Falcon worker HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Port used to debug the data cache

Port	Protocol	Service Name	Direction	Source	Dest.	Description
4003	RPC	Falcon worker RPC port	bidirectional	All nodes	All nodes	Port used by data cache for communication between themselves
4004	RPC	Falcon worker RPC port	bidirectional	All nodes	All nodes	Port used by data cache for communication between themselves
4021	RPC	Sage metadata service port (exported by Tomcat), Callosum services like meta-data services, medata-dependency service, scheduling service, session-less service, spotiq service	bidirectional	All nodes	All nodes	Port where search service contacts metadata service for metadata
4181	RPC	Ports used by Zookeeper servers for communication between themselves	bidirectional	All nodes	All nodes	Ports used by Zookeeper servers for communication between themselves
4201	HTTP	Sage auto complete server HTTP interface port	bidirectional	Admin IP addresses and all nodes	All nodes	Port used to debug the search service
4231	HTTP	Sage index server HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Port used to debug the search service
4232	RPC	Sage index server metadata subscriber port	bidirectional	All nodes	All nodes	Port used for search service internal communication
4233	RPC	Sage index server RPC port	bidirectional	All nodes	All nodes	Port used for search service internal communication

Port	Protocol	Service Name	Direction	Source	Dest.	Description
4241	HTTP	Sage auto complete server HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Port used to debug the search service
4242	RPC	Sage auto complete server RPC port	bidirectional	All nodes	All nodes	Port used for search service internal communication
4243	RPC	Sage auto complete server metadata subscriber port	bidirectional	All nodes	All nodes	Port used for search internal communication
4244	RPC	Sage auto complete server metadata subscriber port	bidirectional	All nodes	All nodes	Port used for search internal communication
4245	RPC	Sage auto complete server metadata subscriber port	bidirectional	All nodes	All nodes	Port used for search internal communication
4243	RPC	Sage auto complete server metadata subscriber port	bidirectional	All nodes	All nodes	Port used for search internal communication
4251	RPC	Sage master RPC port	bidirectional	All nodes	All nodes	Port used for search service internal communication
4405	RPC	Diamond (graphite) port	bidirectional	All nodes	All nodes	Port used for communication with monitoring service
4406	RPC	Diamond (graphite) port	bidirectional	All nodes	All nodes	Port used for communication with monitoring service
4500	RPC	Trace vault service RPC port	bidirectional	All nodes	All nodes	Trace collection for ThoughtSpot services
4501	HTTP	Trace vault service HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Debug trace collection

Port	Protocol	Service Name	Direction	Source	Dest.	Description
4851	RPC	Graphite manager RPC port	bidirectional	All nodes	All nodes	Communication with graphite manager
4852	HTTP	Graphite manager HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Debug graphite manager
4853	RPC	Elastic search stack (ELK) manager RPC port	bidirectional	All nodes	All nodes	Communication with log search service
4853	HTTP	Elastic search stack (ELK) manager HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Debug log search service
9200	RPC	Elastic search (ELK)	bidirectional	All nodes	All nodes	Communication with log search service
5021	RPC	Callosum services like meta-data services, medata-dependency service, scheduling service, session-less service, spotiq service	bidirectional	All nodes	All nodes	Port where search service contacts meta-data service for metadata
5432	Postgres	Postgres database server port	bidirectional	All nodes	All nodes	Communication with Postgres database
6021	RPC	Callosum services like meta-data services, medata-dependency service, scheduling service, session-less service, spotiq service	bidirectional	All nodes	All nodes	Port where search service contacts meta-data service for metadata
7021	RPC	Callosum services like meta-data services, medata-dependency service, scheduling service, session-less service, spotiq service	bidirectional	All nodes	All nodes	Port where search service contacts meta-data service for metadata

Port	Protocol	Service Name	Direction	Source	Dest.	Description
8020	RPC	HDFS namenode server RPC port	bidirectional	All nodes	All nodes	Distributed file system (DFS) communication with clients
8021	RPC	Callosum services like meta-data services, medata-dependency service, scheduling service, session-less service, spotiq service	bidirectional	All nodes	All nodes	Port where search service contacts metadata service for metadata
8080	HTTP	Tomcat	bidirectional	All nodes	All nodes	BI engine communication with clients
8081	HTTP	Callosum/Tomcat status	bidirectional	All nodes	All nodes	BI engine communication with clients
8787	HTTP	Periscope (UI) service HTTP port	bidirectional	All nodes	All nodes	Administration UI back end
8888	HTTP	HTTP proxy server (tinyproxy)	bidirectional	All nodes	All nodes	Reverse SSH tunnel
11211	Memcached	Memcached server port	bidirectional	All nodes	All nodes	BI engine cache
12345	ODBC	Simba server port	bidirectional	All nodes	All nodes	Port used for ETL (extract, transform, load)
8480	HTTP	HDFS journalnode server HTTP port	bidirectional	All nodes	All nodes	Debug DFS metadata
8485	HTTP	HDFS journalnode server HTTP port	bidirectional	All nodes	All nodes	Debug DFS metadata
50070	HTTP	HDFS namenode server HTTP port	bidirectional	All nodes	All nodes	Debug DFS metadata
50090	HTTP	HDFS secondary namenode server HTTP port	bidirectional	All nodes	All nodes	Debug DFS metadata
50075	HTTP	HDFS datanode server HTTP port	bidirectional	All nodes	All nodes	Debug DFS data

Port	Protocol	Service Name	Direction	Source	Dest.	Description
50010	HTTP	HDFS datanode server HTTP port	bidirectional	All nodes	All nodes	Debug DFS data
50020	HTTP	HDFS datanode server HTTP port	bidirectional	All nodes	All nodes	Debug DFS data
7000	TCP	Cassandra KV store database	bidirectional	All nodes	All nodes	Debug DFS data
7001	TCP	Cassandra	bidirectional	All nodes	All nodes	Debug DFS data
9042	HTTP	Munshi server impression service, Cassandra	bidirectional	All nodes	All nodes	Debug DFS data
9160	TCP	Cassandra	bidirectional	All nodes	All nodes	Debug DFS data
4010	HTTP	Falcon moderator	bidirectional	All nodes	All nodes	Debug DFS data
4011	HTTP	Falcon moderator	bidirectional	All nodes	All nodes	Debug DFS data
20123 - 32768	TCP (dynamic)	Dynamic port in this range used for various services and anciliary services like atlas, caffeine, callhome, callosum, falcon, monitoring, munshi server, nlp, object_search, postgres, sage UBR, spotiq snapshot, timely	All nodes	Services		
5270	TCP	Cluster monitoring service (ELK)	bidirectional	All nodes	All nodes	Services
5271	TCP	Cluster monitoring service (ELK)	bidirectional	All nodes	All nodes	Services
5601	TCP	Kibana UI (ELK)	bidirectional	All nodes	All nodes	Services
6311	TCP	R service	bidirectional	All nodes	All nodes	Services
8008	TCP	Video recorder	bidirectional	All nodes	All nodes	Services

Port	Protocol	Service Name	Direction	Source	Dest.	Description
9090	TCP	Timely	bidirectional	All nodes	All nodes	Services
	ICMPv4	Used for health check of cluster nodes	bidirectional	All nodes	All nodes	Services

Required ports for inbound and outbound cluster access

ThoughtSpot uses static ports for inbound and outbound access to the cluster.

Port	Protocol	Service Name	Direction	Source	Dest.	Description
22	SCP	SSH	bidirectional	ThoughtSpot Support	All nodes	Secure shell access.
80	HTTP	HTTP	bidirectional	ThoughtSpot Support	All nodes	Hypertext Transfer Protocol for website traffic.
443	HTTPS	HTTPS	bidirectional	ThoughtSpot Support	All nodes	Secure HTTP.
12345	TCP	Simba	bidirectional	ThoughtSpot Support	All nodes	Port used by ODBC and JDBC drivers when connecting to ThoughtSpot.
2049	TCP	NFS: In case one needs to mount NFS share on TS node.	bidirectional	ThoughtSpot Support	All nodes	Port used by NFS.
123	UDP	NTP service	bidirectional	ThoughtSpot Support	All nodes	Port used by NTP service.

Port	Protocol	Service Name	Direction	Source	Destination	Description
443	TCP	HTTPS	outbound	All nodes	208.83.110.20	For transferring files to thoughtspot.egnyte.com.

Port	Protocol	Service Name	Direction	Source	Destination	Description
443	TCP	HTTPS	outbound	All nodes	For transferring product usage data to mixpanel cloud.	outbound
443	TCP	HTTPS	outbound	All nodes	je8b47jfif.execute-api.us-east-2.amazon-aws.com s3.us-west-1.amazon-aws.com s3-us-west-1.amazon-aws.com s3.dualstack.us-west-1.amazonaws.com	For transferring monitoring data to InfluxCloud. (Given address will resolve to point to AWS instances).
25 or 587	SMTP	SMTP or Secure SMTP	outbound	All nodes and SMTP relay (provided by customer)	All nodes	Allow outbound access for the IP address of whichever email relay server is in use. This is for sending alerts to ThoughtSpot Support.
389 or 636	TCP	LDAP or LDAPS	outbound	All nodes and LDAP server (provided by customer)	All nodes	Allow outbound access for the IP address of the LDAP server in use.

Required ports for IPMI (Intelligent Platform Management Interface)

ThoughtSpot uses static ports for out-of-band IPMI communications between the cluster and ThoughtSpot support.

Port	Protocol	Service Name	Direction	Source	Dest.	Description
80	HTTP	HTTP	bidirectional	ThoughtSpot Support	All nodes	Hypertext Transfer Protocol for website traffic.
443	TCP	S-HTTP	bidirectional	ThoughtSpot Support	All nodes	IPMI GUI and for HTML5-based IPMI console access.

Port	Protocol	Service Name	Direction	Source	Dest.	Description
623	UDP	Serial-over-LAN	bidirectional	ThoughtSpot Support	All nodes	IPMI GUI and for HTML5-based IPMI console access.

Contact support

You can contact ThoughtSpot by phone [See page 500], mail [See page 0], email [See page 500], or by filing a support ticket [See page 500].

File a support ticket

If you encounter a technical issue, file a support ticket using the Support Portal ticket filing system at:

http://support.thoughtspot.com/ [See page 0]

Please provide as much detail as possible about your issue, to help us resolve it quickly.

You need a Support Portal login to file a ticket. Please contact ThoughtSpot to get an account, if necessary.

Address

ThoughtSpot, Inc. 910 Hermosa Ct Sunnyvale, CA 94085

Phone numbers

Phone Number	Description
1-800-508-7008 ext 1	ThoughtSpot Support
1-800-508-7008	Toll-free number for ThoughtSpot headquarters.

Email

Reason for contacting us	Email
Sales inquiries	sales@thoughtspot.com [See page 0]

Reason for contacting us	Email
Customer support and software update inquiries	support@thoughtspot.com [See page 0]
Other inquiries	hello@thoughtspot.com [See page 0]

Introduction to Data Integration

This guide explains how to integrate ThoughtSpot with other data sources for loading data. It also includes information on installing and using the ThoughtSpot clients (ODBC, JDBC, and more).

ThoughtSpot Clients

ThoughtSpot provides certified clients to help you load data easily from your ETL tool or another database. These include ODBC and JDBC drivers.

You can obtain the ThoughtSpot client downloads from the Help Center. Always use the version of the ThoughtSpot clients that corresponds with the version of ThoughtSpot that you are running. When upgrading, make sure to upgrade your clients as well.

> ⚠ **Important:** The ETL tool must add a data transformation step if the source column data type does not exactly match the target's, ThoughtSpot's, column data type. The driver does not do any implicit conversions.

Methods for loading data

There are several ways to load data into ThoughtSpot, depending on your goals and where the data is located. Always consider your requirements for recurring loads when planning how best to bring the data into ThoughtSpot.

Here are the options, with information on where to find the documentation for each method:

Method	Description
ThoughtSpot Loader (tsload)	ThoughtSpot Loader is a command line tool to load CSV files into an existing database schema in ThoughtSpot. This is the fastest way to load extremely large amounts of data, and it can be run in parallel. You can also use this method to script recurring loads. See the ThoughtSpot Administrator Guide for details.

Method	Description
User Data Import	Users can upload a spreadsheet through the web interface with User Data Import. This is useful for giving everyone easy access to loading small amounts of their own data. See the ThoughtSpot Administrator Guide for details.
ODBC [See page 508]	ThoughtSpot provides an ODBC (Open Database Connectivity) driver to enable transferring data from your ETL tool into ThoughtSpot.
JDBC [See page 557]	ThoughtSpot provides a JDBC (Java Database Connectivity) driver to enable transferring data from your ETL tool into ThoughtSpot.
Microsoft SSIS (SQL Server Integration Services) [See page 534]	You can use the ODBC driver to connect to SSIS and import data into ThoughtSpot. Basic instructions are included in this guide.
Connect to Pentaho [See page 565]	You can use the JDBC driver to connect to Pentaho and import data into ThoughtSpot. Basic instructions are included in this guide.

Where to go next

- **Server-side prerequisites for using JDBC/ODBC to import data [See page 507]**

  You must follow setup prerequisites for importing data using JDBC/ODBC.

- **About the ODBC Driver [See page 508]**

  You can use the ThoughtSpot ODBC driver to bring data into ThoughtSpot from your ETL tool or database.

- **About the JDBC Driver [See page 557]**

  Java Database Connectivity (JDBC) is a Java standard API that allows applications to interact with databases in a standard manner. ThoughtSpot has JDBC support through a JDBC driver that we provide.

Embrace overview

Summary: Using Embrace, you can perform live query on external databases.

If your company stores source data externally in data warehouses, you can use ThoughtSpot Embrace to directly query that data and use ThoughtSpot's analysis and visualization features, without moving the data into ThoughtSpot. If you decide later you want to copy your data into ThoughtSpot, you can also do that with Embrace.

Embrace supports the following external databases:

- Snowflake
- Amazon Redshift (*in beta*)

To enable Embrace, contact ThoughtSpot support.

How it works

You create a connection to the external database, choosing the columns from each table that you want to explore in your live query. Primary key and foreign key relationships are imported along with the primary and foreign key tables. If there are any joins in the tables of your connection, they are also imported. After your connection is complete, it becomes a **linked** data source in ThoughtSpot that allows you to query the external database directly. It's easy to apply transformations and filter the data also.

Key benefits

- Set up and deploy ThoughtSpot faster by connecting directly to the external database.
- Eliminate the need to move data into ThoughtSpot for analysis.
- Centralize data management and governance in the external database.
- Save significant time and money by avoiding ETL pipelines.
- Set up and schedule sync of data into ThoughtSpot.
- Connect to multiple external databases.

Embrace modes

Embrace has two operating modes:

- **Linked**: ThoughtSpot queries your data in the external database.
- **Synced**: ThoughtSpot queries a copy of your data stored in ThoughtSpot.

When you create your connection to an external database, by default, it is a **Linked** connection. If you want to copy the external data into ThoughtSpot, you must sync the data. The features available with Linked and Synced tables are slightly different.

Features in Embrace modes

Feature	Linked Tables	Synced Tables
Simple Search	Yes	Yes
Complex searches like Versus, Inline Subquerying, Growth	Yes	Yes
Search Suggestions for column names	Yes	Yes
Search Suggestions for column values	Yes	Yes
Headlines at the bottom that summarize tables	Yes	Yes
All Chart Types & Configurations	Yes	Yes
SpotIQ Instant Insights	No	Yes
SpotIQ pre-computed insights	No	Yes
Table and Column Remapping	Yes	N/A
Custom Calendar	No	Yes
Materialized Views	No	Yes
Indexing of table columns	Yes	Yes

Next steps

- **Add a connection [See page 0]**

Create the connection between ThoughtSpot and tables in an external database.

- **Sync [See page 0]** Set your connection to copy tables from the external database into ThoughtSpot.
- **Modify a connection [See page 0]**

  Edit, remap or delete a connection to tables in an external database.
- **Connectors reference [See page 0]**

  Source cloud data connectors, and their connection credentials, supported by Embrace.

JDBC and ODBC setup prerequisites

Before you can use JDBC or ODBC to import data into ThoughtSpot, you must do the following server-side configuration:

1. Open up the ThoughtSpot firewall to allow incoming requests to Simba server.

   ```
   tscli firewall open-ports --ports 12345
   ```

2. Confirm that the `simba_server` process is up. Output of the command below should contain exactly one line, as shown below.

   ```
   ps -ef | grep simba_server | grep -v grep
   admin    26679 25672  0 Jul13 ?        00:01:49 simba_se
   rver_main --logbufsecs=0
   ```

   For assistance, contact ThoughtSpot Support.

Overview of the ODBC Driver

Summary: Use the ODBC driver to bring data in from your ETL tool or database.

ThoughtSpot comes packaged with an ODBC (Open Database Connectivity) driver, so that you can transfer data between ThoughtSpot and other databases. Basic knowledge of ODBC data source administration is helpful when setting up ODBC.

Supported operating systems for the ODBC driver are:

- Microsoft Windows 32-bit
- Microsoft Windows 64-bit
- Linux 32-bit
- Linux 64-bit

Version compatibility and connection parameters

To ensure compatibility, always use the ODBC driver with the same version number as the ThoughtSpot instance to which you are connecting. You can make a secure ODBC connection to the ThoughtSpot database by configuring a user and password combination with the driver. For detailed information about connection parameters, see the ODBC and JDBC configuration properties [See page 594]

Supported Data Types

The ODBC driver supports these data types:

- `INT`
- `BIGINT`
- `BOOLEAN`
- `DOUBLE`
- `FLOAT`
- `DATE`
- `TIME`
- `TIMESTAMP`
- `DATETIME`

- `CHAR`
- `VARCHAR`

Source and target data compatibility

By default, ThoughtSpot takes a permissive approach to data type compatibility between source and target data in ODBC. In this mode, ThoughtSpot *assumes* that the incoming data matches exactly with the target data types and loads the table as is.

Alternatively, you can explicitly require that ThoughtSpot match the source data types exactly and, if it can't find a match, it returns an error and the data load fails. In this mode, for example, if the target ThoughtSpot data type for a column is INT, the source data type for that column must be INT in order for the data load to succeed.

By toggling ***strict*** and ***permissive*** `true` and `false` options, you can configure settings along a scale of behavior between the permissive, automatic approach and the strictness of the "must match" approach.

		Strictness	
		true	false
Permissiveness	true	Data types are inferred and automatically converted. ThoughtSpot returns an error in cases where the data conversion is not possible. Data load fails in its entirety if any data contains mismatches. You must correct the problem in the source data and try the load again.	Data types are inferred and automatically converted. No error is thrown even if source and target data types don't match. Data load continues even when the source and target data types don't match. This means your data load may contain data types that you do not intend or that are not helpful. You are responsible for checking and validating the data in this case.

false	The source and target data types must match. If any data contains mismatches, ThoughtSpot returns an error to the client a data load fails in its entirety. You must correct the problem in the source data and try the load again. This is the strictest configuration.	No data types are inferred and conversion does not check for matches. This is the most permissive configuration.

Your customer support engineer can assist you in configuring custom ODBC behavior. Regardless of the configuration you choose, you must validate that the results of data loading *as they appear* in ThoughtSpot are what you require.

Data type conversion matrix

The following table describes the conversion matrix between SQL data types and ThoughtSpot data types.

Source SQL Data Types	BOOL	INT32	INT64	DOUBLE	FLOAT	CHAR	DATE	TIME	DATETIME
`SQL_BIT`	Y	Y	Y	Y	Y	Y	–	–	–
`SQL_TINYINT`	Y	Y	Y	Y	Y	Y	–	–	–
`SQL_SMALLINT`	Y	Y	Y	Y	Y	Y	–	–	–
`SQL_INTEGER`	Y	Y	Y	Y	Y	Y	–	–	–
`SQL_BIGINT`	Y	Y	Y	Y	Y	Y	–	–	–
`SQL_CHAR`	Y	Y	Y	Y	Y	Y	Y	Y	Y
`SQL_VARCHAR`	Y	Y	Y	Y	Y	Y	Y	Y	Y
`SQL_LONGVARCHAR`	Y	Y	Y	Y	Y	Y	Y	Y	Y
`SQL_BINARY`	–	–	–	–	–	Y	–	–	–
`SQL_VARBINARY`	–	–	–	–	–	Y	–	–	–
`SQL_LONGVARBINARY`	–	–	–	–	–	Y	–	–	–

Source SQL Data Types	BOOL	INT32	INT64	DOUBLE	FLOAT	CHAR	DATE	TIME	DATETIME
SQL_DOUBLE	Y	Y	Y	Y	Y	Y	–	–	–
SQL_REAL	Y	Y	Y	Y	Y	Y	–	–	–
SQL_FLOAT	Y	Y	Y	Y	Y	Y	–	–	–
SQL_NUMERIC	Y	Y	Y	Y	Y	Y	–	–	–
SQL_GUID	–	–	–	–	–	Y	–	–	–
SQL_INTERVAL_MINUTE_TO_SECOND	–	–	–	–	–	Y	–	–	–
SQL_INTERVAL_HOUR_TO_SECOND	–	–	–	–	–	Y	–	–	–
SQL_INTERVAL_HOUR_TO_MINUTE	–	–	–	–	–	Y	–	–	–
SQL_INTERVAL_DAY_TO_SECOND	–	–	–	–	–	Y	–	–	–
SQL_INTERVAL_DAY_TO_MINUTE	–	–	–	–	–	Y	–	–	–
SQL_INTERVAL_DAY_TO_HOUR	–	–	–	–	–	Y	–	–	–
SQL_INTERVAL_YEAR	–	Y	Y	–	–	Y	–	–	–
SQL_INTERVAL_MONTH	–	Y	Y	–	–	Y	–	–	–
SQL_INTERVAL_DAY	–	Y	Y	–	–	Y	–	–	–
SQL_INTERVAL_HOUR	–	Y	Y	–	–	Y	–	–	–
SQL_INTERVAL_MINUTE	–	Y	Y	–	–	Y	–	–	–
SQL_INTERVAL_SECOND	–	Y	Y	–	–	Y	–	–	–
SQL_TYPE_TIME	–	–	–	–	–	Y	–	Y	Y
SQL_TYPE_DATE	–	–	–	–	–	Y	Y	–	Y
SQL_TYPE_TIMESTAMP	–	–	–	–	–	Y	Y	Y	Y

If a conversion is not possible, an error is returned to the client to indicate conversion failure. The ETL tool must add a data transformation step if the source column data type does not exactly match the target's ThoughtSpot column data type. The driver does not do any implicit conversions.

Install the ODBC driver on Windows

Summary: Use this procedure to obtain the Microsoft Windows ODBC driver and install it.

ThoughtSpot's ODBC connection relies on the SimbaEngine X SDK [See page 0] to connect through ODBC or JDBC to ThoughtSpot's remote data stores. The instructions on this page explain how to configure the Simba ODBC driver on a Windows workstation.

Make sure you have read the overview material in the ODBC driver overview [See page 508]. This workstation is the same machine where you plan to run your ETL activities.

Prerequisites

These instructions include directions to use the `ssh` command. Make sure your Windows workstation is equipped with a tool such as Putty [See page 0] for making `ssh` connections to your ThoughtSpot server.

The ODBC driver for Windows requires Visual C++ Redistributable for Visual Studio 2013. You are prompted to install it during installation of the driver if it isn't already installed.

To check if this Microsoft tool is already installed, search for it on your workstation.

If it isn't installed, make sure you download and install it [See page 0] before continuing.

Check the ThoughtSpot IP and the simba_server status

Before you begin, you need to know the IP address or DNS name of the server you intend to connect your server to.

1. SSH as `admin` or the `thoughtspot` user to your ThoughtSpot node.
2. Verify the node IP(s).

```
$ tscli node ls
172.18.231.17
172.18.231.18
```

3. Make a note of each IP; there may be more than one.
4. Configure the ThoughtSpot firewall to allow connections from your ETL client, by running the following command on any ThoughtSpot node: `tscli firewall open-ports --ports 12345`
5. Exit or close the shell.

Download the driver

On the workstation where you want to connect from, do the following:

1. Navigate to the **Downloads** [See page 0] page.
2. Download the **ODBC Driver for Windows**.
3. Unzip the file you downloaded at a convenient location on your workstation.

4. Take a moment to examine the contents of the new directory.

   There are two different Windows ODBC installers included in the file you downloaded.

   - ThoughtSpotODBC (x86).msi for Windows 32-bit
   - ThoughtSpotODBC (x64).msi for Windows 64-bit

Install the driver and supporting software

The installation process installs the Simba driver and adds the ODBC Administrator software to your workstation. You use this software to configure the driver.

1. Launch the installer for your version of Windows.

2. Click **Next** to continue.

3. Accept the End User License Agreement (EULA), and click **Next**.

4. Specify the destination folder where the driver will be installed.

5. Enter the ThoughtSpot server details, and click **Next**.

- For **Server(s)**, provide a comma separated list of the IP addresses of each node on the ThoughtSpot instance.
- For **Database**, optionally specify the database to use. If you skip this entry, you must provide the database each time you connect using ODBC.

6. Confirm that the install can begin by clicking **Install**.

7. You may see a security warning.

8. Click **Yes** to continue.

A confirmation message appears when the installation is complete.

9. Click **Finish**.

Configure the driver and test your connection

After installation completes, use the ODBC Administrator to configure the ODBC connection on your Windows workstation. For example, you may want to add a default schema or change the server IP address or the default database.

It is recommended to add a default schema. If you don't specify a default schema, you must supply it every time you use the ODBC driver.

At this point, you can test your ODBC connection to ThoughtSpot. It is important to recall that the username/password you use belongs to a ThoughtSpot application user. Typically, this user is a user with data management or administrative privileges on the application.

1. Before trying the ODBC connection, confirm a username/password that can log into the ThoughtSpot applications.
2. Click the **Data** tab, and confirm the user's privileges.
3. Return to your workstation.
4. Locate and open the **ODBC Data Source Administrator (64-bit)** application.

5. Click the **System DSN** tab.

6. Select **ThoughtSpot** and click **Configure…**

7. Click **Options…**

8. Ensure you have the following **Custom Property** values set:

Custom Property	Value
`SCHEMA`	`falcon_default_schema` is the default
`UID`	The username of a user with data management privilege.
`PWD`	The password for the username you specify.

You don't have to use the ThoughtSpot default schema. You can specify your own. We recommend that you define a default schema. Otherwise, you must supply a schema every time you use the ODBC driver. Moreover, without a schema (or if the schema is not present), the ODBC driver returns an error that states that the schema could not be found.

Similarly, adding the `UID` and `PWD` properties are not required. If you don't add them, you are prompted to supply them each time you connect.

When you are done, your options should look similar to the following:

9. When you are done, click **OK** to save your new properties.
10. Click **Test Connection** to test your database connection.

11. Click **Cancel** to close the **DSN Configuration** dialog.
12. Click **OK** to close the **Client Configuration Dialog** the dialog.
13. Click **OK** to close the **ODBC Data Source Administrator (64-bit)** application.

Now, you are ready to begin using the connection you've configured.

Related information

- Enable ODBC logs [See page 580].
- Configure multiple connections on Windows [See page 522].

Configure multiple connections on Windows

Summary: You can add multiple ODBC data sources.

Use this procedure if you want to add an additional data source after creating a single source succeeds [See page 512]. The main reason for needing to set up multiple ThoughtSpot ODBC data sources is that you have a production cluster and a test or development cluster.

1. Locate and open the **ODBC Data Source Administrator (64-bit)** application.

2. Click the **System DSN** tab.
3. Select **Add**.

   The system lists the available drivers.

4. Choose the **ThoughtSpot ODBC Driver** and click **Finish**.

   The system displays the **Client Configuration Dialog** dialog.

5. Enter the your data source configuration.

Configuration Property	Value
Data Source Name	The name you want to call the data source.
Description	A description of the data source.
Server IP	A list of the IP addresses for each node, separated by commas.
Server Port	12345
Idle Timeout	Time in seconds after which an idle ODBC connection times out.
Login Timeout	Time in seconds after which a login request times out.
Query Timeout	Time in seconds after which a query times out.

6. Configure custom properties by clicking **Options**.

   The system displays the **Options** dialog.

7. Add these properties using the **Add** to enter an option and click **OK** after to save an option.

Option	Value
DATABASE	The default database to connect to.
SCHEMA	The default schema to connect to. Use `falcon_default_schema` if you aren't sure.
CONNECTIONTIMEOUT	Optional. Seconds before an idle connection times out.

The key must be defined exactly as it appear here, using all capital letters. You can find other supported properties in ODBC and JDBC configuration properties [See page 594].

8. When you are done, click **OK** to save your new configuration.

9. Click **Test Connection** to test your database connection.

If your test connection fails, enable ODBC logging [See page 580] to troubleshoot.

10. Click **Cancel** to close the **DSN Configuration** dialog.
11. Click **OK** to close the **Client Configuration Dialog** the dialog.
12. Click **OK** to close the **ODBC Data Source Administrator (64-bit)** application

Deploy SSL with ODBC on Windows

You can configure a secure ODBC connection between your ThoughtSpot cluster and a remote Windows Machine. This article explains the SSL resources and ODBC configuration options you need to enable SSL for an ODBC connection.

Prerequisites

Before configuring SSL over the ThoughtSpot ODBC connection, make sure that your system administrator has created and configured your network's Certificate Authority. Additionally, the system administrator should have available both the proper Private Key and Server Certificate.

Configure the ThoughtSpot cluster nodes

> ⚠ **Important:** Portions of this procedure require that you work with your ThoughtSpot Customer Service or Support Engineer.

The SimbaServer Configuration Properties reference [See page 0] includes full details on SSL Configuration Properties [See page 0].

Before you change your ODBC configuration, decide on a path where you will store the Private Key and Server Certificate, for example, you could decide to use `/home/admin/Simba_SSL/` as the path.

Then, do the following on *every ThoughtSpot node* in your cluster.

1. Create the path on the node.
2. Copy the SSL certificate and private key to this path.
3. Edit the node's `/etc/thoughtspot/simba.ini` file (Simba server configuration) with your favorite editor.
4. Add the following lines:

```
 SslCertfile=/home/admin/Simba_SSL/Server-Certificate.pe
m
 SslKeyfile=/home/admin/Simba_SSL/Private-Key.pem
 UseSsl=Required
```

5. Restart the Simba service.

   You must work with your ThoughtSpot Customer Success or Support Engineer to do this.

Deploy the certificate on your windows workstation

Please note that the SSL settings on the server and client are interdependent.

The SimbaClient for ODBC Configuration Properties [See page 0] reference describes how to set parameters on the client to use SSL (scroll down to useSsl section at the end). The Simba documentation also provides a chart showing configuration properties for SSL [See page 0] where you can see how different combinations of SSL settings on client and server will behave. For example:

- Setting both server and client to `UseSsl=Enabled` provides the ability for clients to connect with or without SSL.
- Setting both server and client to `UseSsl=Required` requires that all clients use SSL.

> ℹ **Note:** Note that the SSL and certificate parameters can be set through the pre-defined options on the options dialog, but customers have reported that these are not always reliable. In the following procedure, we recommend using custom properties to define these settings (either preemptively, or as a solution if the ODBC connection over SSL does not work with the pre-defined options). There is no harm in setting both. Example settings are: `UseSSL = Required` and `SslCACertfile = C:\ODBC-SSL\CA.pem`

On the workstation you want to use for your ODBC connection, specify the level of SSL you want to use on the client along with the path to the CA certificate, and then test the connection.

1. Save the CA certificate to a secure location on the workstation disk.

Choose a location where the certificate is unlikely to be deleted by mistake, for example, `C:\ODBC-SSL\CA.pem` is an example of a full path to such a location.

2. Open your ThoughtSpot ODBC connection configuration dialog.
3. Click **Options**.
4. Check the **Require SSL** option and/or add SSL as a custom property.

Options
SSL
Disable SSL
Enable SSL
Require SSL
You can also set SSL on the client here, but this is less dependable
SSL Certificate File:
C:\ODBC-SSL\CA.pem
Custom Property
CONNECTIONTIMEOUT
DATABASE data
MSGFETCHSIZE 10
Add...
Remove
Edit Property
Click to add SSL parameter as a custom property (in addition to or as an alternative to the pre-defined option above)
Cancel

Edit Property
Custom Property
Key:
UseSSL
Value:
Required
Set UseSSL to set this to either Enabled or Required. We suggest Required.
OK
Cancel

5. Enter the location of the CA certificate in the **SSL Certificate File** field and/or add the CA certificate as a custom property. Be sure to provide the full path to the certificate (`{certificate_directory}\{CA_certificate}.pem`).

Options
SSL
Disable SSL
Enable SSL
Require SSL
SSL Certificate File:
Custom Property
CONNECTIONTIMEOUT
DATABASE data
MSGFETCHSIZE 10
PUSHEDPAR
UseSSL
Add...
Remove
Edit Property
Cancel
Click to add the CA certificate as a custom property (in addition or as an alternative to the pre-defined option)

Edit Property
Custom Property
Key: SslCACertfile
Value: C:\ODBC-SSL\CA.pem
OK
Cancel
Specify the full path to your CA certificate as a custom property called SslCACertfile

6. When you are done, click **OK** to save your new properties.

7. Click **Test Connection** to test your database connection.

8. Click **Cancel** to close the configuration dialog.
9. Click **OK** to close the **Client Configuration Dialog** the dialog.
10. Click **OK** to close the **ODBC Data Source Administrator (64-bit)** application.

Set up the ODBC Driver for SSIS

Summary: Use SSIS to set up the ODBC Driver.

Microsoft SSIS (SQL Server Integration Services) is a data integration and workflow applications platform you can use to connect to ThoughtSpot. The platform is a component of the Microsoft SQL Server database software.

You can use a SSIS connection to perform data migration tasks. Its data warehousing tool is useful for data ETL (extraction, transformation, and loading). The SSIS Import/Export Wizard creates packages that transfers data with no transformations. It can move data from a variety of source types to a variety of destination types, including text files and other SQL Server instances.

Use SSIS to set up the ODBC Driver by creating a connection manager. This manager connects an OLE DB Source and the ODBC Destination.

Prerequisites

On Windows 64-bit, you have to install both the 32-bit and 64-bit ThoughtSpot ODBC drivers. In addition, they must be named the same, such as ThoughtSpot. By default they are named ThoughtSpot-32 and ThoughtSpot-64. This is required because the 64-bit SSIS shows a list of 32-bit ODBC drivers when you configure an ODBC target. However, it executes the 64-bit driver. If the drivers aren't named the same, then you can get an error stating the driver doesn't exist.

Set up the driver

To set up the ODBC driver using SSIS:

1. Open your SQL Server visual development tool that is based on Microsoft Visual Studio.
2. Select **OLE DB Source**, and click **New**.

   Where ODBC provides access only to relational databases, OLE DB provides access to data regardless of its format or location.
3. Add the server by name from the machine accessible list.

4. Enter the authentication information: db name, user name, password, and test connection.

   You can add the UID and password by clicking on **Options**.

5. Click **File** and select **New**, then **Project**.
6. Select the **Integration Services** tab under **Installed > Templates > Business Intelligence**.
7. Enter a name in the **Name** field and click **OK**.

8. Select the **SSIS Toolbox** tab on the left hand side of the platform, and drag and drop **Data Flow Task** to the main window.

9. Double click the **Data Flow Task** icon when it appears in the center of the page.

10. Navigate back to the **SSIS Toolbox** tab. You now want to create sources and destinations.

Create sources and destinations

1. Under **Other Sources**, find **OLE DB Source** and drag and drop it to the main window.
2. Double click the **OLE DB Source** icon when it appears in the center of the page to open the OLE DB Source Editor.
3. Select a new OLE DB connection manager by clicking **New**.
4. In the Configure OLE DB Connection Manager window, select your **Data connection** and click **OK**.

If you do not see your data connection, you will have to create a new one in the Connection Manager by clicking **New**.

5. Back in the OLE DB Source Editor, select the **Name of the table or the view**, and click **OK**.

6. Select the table, and see what columns are in it.

   In this example, a single column, `c1`, is selected.

Configure the ODBC Data Source Administrator

The ODBC Data Source Administrator has to be configured to connect to ThoughtSpot and bring the table in.

1. Search for and open your **ODBC Data Sources (32-bit)** program.
2. Click the **System DSN** tab and select **ThoughtSpot_32**.
3. Click **Configure**.

4. In the Client Configuration Dialog, enter the **Server IP** and **Server Port**.

   Enter any node IP that has Simba server running on it. In **Secondary Servers**, you must specify all node IPs, because ThoughtSpot must resolve to the server Simba runs on, and that server can change after an upgrade. Enter one server IP per line. The line return serves as a separator. Comma separated values are not supported.

5. Click **OK** twice to close the Client Configuration Dialog and the ODBC Data Source Administrator.

Create a file to take the feed

Now that you have set up your source, create the empty table in ThoughtSpot to take this feed. SSIS does not allow you to create the table in ThoughtSpot. You have to do this first in TQL. In Pentaho, it will create the table in ThoughtSpot, but not in SSIS.

Create the ODBC Destination. Use the one you created and named in the ODBC Data Source Administrator.

1. In the **SSIS Toolbox** tab, under **Other Destinations**, drag and drop **ODBC Destination** to the main window.
2. Drag the **blue arrow** to connect the OLE DB Source icon to the ODBC Destination icon.
3. Double click the **ODBC Destination** icon.

4. Use ODBC Destination to set the **Batch size** for the connection in the Connection Manager tab. You can set the size to be up to 10,000.

If the load fails, the entire batch will be lost, and you will have to start that load over again.

5. Set the **Transaction Size** to match the total number of rows that are expected to be loaded in the load cycle.

   Your transaction size can be quite large—even spanning a million rows. However, too many small batches can leave the cluster in a rough state. This is because each batch acts as a separate transaction and creates a separate commit. Too many of these will slow down our system since each transaction creates a “data version” in our system. In Pentaho, the transaction size setting is called Commit Size.

6. Set the **Transaction Option** attribute of the Data Flow Task to **Supported**.

7. In the **Mappings** tab, validate the mapping or change it.

   You can have different column names in each database if you map them. Of course, they must be of the same or compatible datatype.

8. Start the import job by clicking the **Start** button.

   You should see an animation indicating that the data is transferring over. When the import is complete, the number of successfully transferred rows is displayed.

You can validate the import using TQL or from the **Data** screen.

Install the ODBC Driver on Linux

Summary: Use this procedure to obtain the Linux ODBC driver and install it.

ThoughtSpot's ODBC connection relies on the SimbaEngine X SDK [See page 0] to connect through ODBC or JDBC to ThoughtSpot's remote data stores. The instructions on this page explain how to configure the Simba ODBC driver on a Linux workstation.

Make sure you have read the overview material in the ODBC driver overview [See page 508]. This workstation is the same machine where you plan to run your ETL activities.

Check the ThoughtSpot IP and the simba_server status

Before you begin, you need to know the IP address or DNS name of the server you intend to connect your server to.

1. SSH as `admin` or the `thoughtspot` user to your ThoughtSpot node.
2. Verify the node IP(s).

```
$ tscli node ls
172.18.231.17
172.18.231.18
```

3. Make a note of each IP; there may be more than one.
4. Configure the ThoughtSpot firewall to allow connections from your ETL client, by running the following command on any ThoughtSpot node: `tscli firewall open-ports --ports 12345`
5. Exit or close the shell.

Install the Simba client

On your workstation, where you want to connect from, do the following to get the ODBC driver:

1. Open a browser on your workstation.

2. Navigate to the **Downloads** [See page 0] page.
3. Click **ODBC Driver for Linux** to download the driver.
4. Open a terminal on your workstation.
5. Change directory to the location where you downloaded the file.
6. Optionally, move the file to a permanent location on your machine.

   When you expand the downloaded file it will create a directory in the location.

7. Unzip the zip file:

   ```
   gunzip ThoughtSpot_linux_odbc_<version>.tar.gz
   ```

8. Extract the contents of the `tar` file.

   ```
   tar -xvf ThoughtSpot_linux_odbc_<version>.tar
   ```

   This extracts a subdirectory called `linux` into the current directory.

9. Take a moment to examine the contents of the new directory.

   The structure contains a Simba client library, supporting libraries and setup files for two different architectures. It also continues error messages for multiple languages.

```
├── Bin
│   ├── Linux_x86
│   │   └── libSimbaClient.so
│   └── Linux_x8664
│       └── libSimbaClient.so
├── ErrorMessages
│   ├── de-DE
│   │   ├── ...
│   ├── en-US
│   │   ├── CSCommonMessages.xml
│   │   ├── ClientMessages.xml
│   │   ├── JNIDSIMessages.xml
│   │   ├── ODBCMessages.xml
│   │   ├── SQLEngineMessages.xml
│   │   └── ServerMessages.xml
│   ├── es-ES
│   ├── fr-FR
│   ├── it-IT
│   └── ja-JP
├── Lib
│   ├── Linux_x86
│   │   ├── libcrypto.so
│   │   ├── libcrypto.so.1.0.0
│   │   ├── ...
│   └── Linux_x8664
│       ├── ...
└── Setup
    ├── .simba.javaquickstart.ini
    ├── .simba.javaultralight.ini
    ├── .simba.quickstart.ini
    ├── .simba.ultralight.ini
    ├── .simbaclient.ini
    ├── .simbaserver.javaquickstart.ini
    ├── .simbaserver.javaultralight.ini
    ├── .simbaserver.quickstart.ini
    ├── .simbaserver.ultralight.ini
    ├── odbc.ini
    └── odbcinst.ini

14 directories, 92 files
```

The `linux/Setup` directory contains the key ODBC configuration files and sample Simba client configurations you can use later in this procedure.

10. You must know your workstation architecture to continue, confirm your workstation's architecture.

    You can use the `arch` or the `uname` command or both.

```
$ arch
  x86_64
$ uname -a
  Linux nebula-docs-production-4vfnv 4.4.108-1.el7.elrepo.x86_64 #1 SMP Mon Dec 25 09:55:39 EST 2017 x86_64 x86_64 x86_64 GNU/Linux
```

    In previous examples, the workstation is a 64 bit workstation. Your workstation may be 32-bit. You can use this architecture information in the procedures that follow.

(Optional) Install unixODBC tools for testing

The procedures on this page rely on the unixODBC tools to test your configuration and connection. If you are experienced with ODBC and want to skip this, you can. Simply substitute your preferred mechanism in the subsequent procedures where references are made to the unixODBC tools.

> ❗ **Warning:** Your ThoughtSpot installation contains a version of the unixODBC tools. These tools are incompatible with CentOS. Do not use these tools if you are performing this procedure on your ThoughtSpot server.

1. Search for the unixODBC tools on your system.

   The `yum` package manager searches for software already installed or available on your system or from the configured repositories. Depending on your workstation configuration, you may need to use the `sudo` command with your workstation.

```
$ yum search unixODBC
...
 * updates: repos-lax.psychz.net
================ N/S matched: unixODB
C==============================
opensips-unixodbc.x86_64 : OpenSIPS unixODBC Storage sup
port
unixODBC-devel.i686 : Development files for programs whi
ch will use the unixODBC library
unixODBC-devel.x86_64 : Development files for programs w
hich will use the unixODBC library
erlang-odbc.x86_64 : A library for unixODBC support in E
rlang
freeradius-unixODBC.x86_64 : Unix ODBC support for freer
adius
unixODBC.i686 : A complete ODBC driver manager for Linux
unixODBC.x86_64 : A complete ODBC driver manager for Lin
ux
```

Make note of the correct package to install for your architecture.

2. Install the appropriate package for your architecture.

In this case the command installs the tools for a 64-bit architecture. A 32-bit package needs the `unixODBC.i686` package.

```
 [admin@nebula-docs-odbc-test-cxmrn ~]$ yum install unix
ODBC.x86_64
 Loaded plugins: fastestmirror, ovl
 Loading mirror speeds from cached hostfile
  * base: mirror.linuxfix.com
  * elrepo: repos.lax-noc.com
  * epel: mirror.hmc.edu
  * extras: centos-distro.cavecreek.net
  * rpmforge: mirror.lstn.net
  * updates: repos-lax.psychz.net
 Resolving Dependencies
 --> Running transaction check
 ---> Package unixODBC.x86_64 0:2.3.1-11.el7 will be ins
talled
 ...
 Complete!
```

3. Verify the files were installed.

```
$ ls /usr/bin/isql
/usr/bin/isql
$ ls /usr/bin/odbcinst
/usr/bin/odbcinst
```

Set up your environment

In this section, you set parameters in your workstation to support your ODBC connection.

1. Copy the library for your architecture from the `Lib` directory on your Linux machine.

Library	Architecture
`/linux/Lib/Linux_x86`	32-bit
`/linux/Lib/Linux_x8664`	64-bit

2. Add the location's path to the `LD_LIBRARY_PATH` environment variable.

   For example if your architecture is 64-bit and you keep the library in your `home` directory:

```
export LD_LIBRARY_PATH=~/linux/Lib/Linux_x8664/
```

3. Use the `echo` command to verify the path was added correctly.

```
echo $LD_LIBRARY_PATH
```

4. Copy the `odbc.ini` file to the `/etc` directory.

```
$ cp ~/linux/Setup/odbc.ini  /etc
```

If you have trouble making the copy, use the `sudo` command to make the move.

5. Copy the `odbcinst.ini` file to the `/etc` directory.

```
$ cp ~/linux/Setup/odbcinst.ini  /etc
```

6. Copy the hidden `.simba.quickstart.ini` file to the `/etc` directory, renaming it in the process to `simbaclient.ini`.

```
$ cp ~/linux/Setup/.simba.quickstart.ini  /etc/simbacli
ent.ini
```

7. Update your environment with the `ODBCSYSINI` and `ODBCINI` variables.

```
$ export ODBCSYSINI=/etc/
$ export ODBCINI=/etc/odbc.ini
```

8. Use the `/usr/bin/odbcinst` command to confirm your settings:

```
$ /usr/bin/odbcinst -j
unixODBC 2.3.1
DRIVERS............: /etc/odbcinst.ini
SYSTEM DATA SOURCES: /etc/odbc.ini
FILE DATA SOURCES..: /etc/ODBCDataSources
USER DATA SOURCES..: /etc/odbc.ini
SQLULEN Size.......: 8
SQLLEN Size........: 8
SQLSETPOSIROW Size.: 8
```

Edit the /etc/simbaclient.ini file

When you are ready, follow this procedure to configure the driver.

1. Edit the `/etc/simbaclient.ini` file with your favorite editor. 2.

2. Change the `ErrorMessagesPath` property to point to the location where you unzipped the client.

```
[Driver]
ErrorMessagesPath=<path_to_error_messages_directory>
```

3. Comment out the `# Generic ODBCInstLib` value.

4. Uncomment the `ODBCInstLib` property.

   When you are done, your file looks like the following:

```
# Generic ODBCInstLib
#   iODBC
#ODBCInstLib=libiodbcinst.so

#   SimbaDM / unixODBC
ODBCInstLib=libodbcinst.so
```

5. Save and close the `/etc/simbaclient.ini` file.

Edit the odbcinst.ini file

The `odbcinst.ini` file is a registry and configuration file for ODBC drivers. Depending on your workstation architecture, you configure the 32-bit or 64-bit driver.

1. Open the file `/etc/odbcinst.ini` in your favorite editor.

2. Comment out the driver that you don't need.

   For example, if you are using 64-bit, comment out 32-bit.

3. Edit the `Driver` line so that it contains the path to the file `libSimbaClient.so`

   Use the path where you copied the library files. For example, for the 64-bit ODBC driver:

```
 [ThoughtSpot(x64)]
 APILevel            = 1
 ConnectFunctions    = YYY
 Description         = ThoughtSpot 64bit ODBC driver
 Driver              = /home/admin/linux/Bin/Linux_x866
4/libSimbaClient.so
 DriverODBCVer       = 03.52
 SQLLevel            = 1
```

4. Make sure the remaining driver is named `ThoughtSpot` without any special characters.

   When you are done, you should see something similar to the following:

```
 #[ThoughtSpot]
 #APILevel            = 1
 #ConnectFunctions    = YYY
 #Description         = ThoughtSpot 32bit ODBC driver
 #Driver              = /usr/local/scaligent/toolchain/l
ocal/simba/odbc/linux/Bin/Linux_x86/libSimbaClient.so
 #DriverODBCVer       = 03.80
 #SQLLevel            = 1

 [ThoughtSpot]
 APILevel            = 1
 ConnectFunctions    = YYY
 Description         = ThoughtSpot 64bit ODBC driver
 Driver              = /home/admin/linux/Bin/Linux_x866
4/libSimbaClient.so
 DriverODBCVer       = 03.80
 SQLLevel            = 1
```

5. Save and close the `/etc/odbcinst.ini` file.

Edit the odbc.ini file

The `odbc.ini` file is a registry and configuration file for ODBC DSNs (Data Source Names). This file relies on the drivers registered in the `/etc/odbcinst.ini` file. Depending on your workstation architecture, you configure the 32-bit or 64-bit driver.

1. Open the file `/etc/odbc.ini` in the editor of your choice.

2. Comment out the configuration that you don't need.

   For example, if you are using 64-bit, comment out 32-bit.

3. Locate the `Description` section for the type of Linux you are using (32-bit or 64-bit).
4. Locate the line that begins with `ServerList`.
5. Replace `127.0.0.1` with a comma separated list of the IP addresses of each node on the ThoughtSpot instance.

   The syntax for the `ServerList` is:

   ```
   ServerList = <node1_IP> 12345, <node2_IP> 12345 [, <node3_IP> 12345, ...]
   ```

   If you need to obtain the IP addresses of the ThoughtSpot cluster nodes, run the command `tscli node ls` from a Linux shell on a ThoughtSpot appliance.

6. Do not edit the port number, leave it as `12345`.

   When you are done, your entry will look similar to the following (this example is for the 64-bit ODBC driver):

   ```
   [ThoughtSpot]
   Description = ThoughtSpot 64-bit ODBC Driver
   Driver = ThoughtSpot
   ServerList =  172.18.231.17 12345
   Locale = en-US
   ErrorMessagesPath = /home/admin/linux/ErrorMessages
   UseSsl = 0
   #SSLCertFile = # Set the SSL certificate file path. The certificate file can be obtained by extracting the SDK tarball
   #LogLevel = 0 # Set log level to enable debug logging
   #LogPath = # Set the debug log files path
   DATABASE = # Set the default database to connect to
   SCHEMA = # Set the default schema to connect to
   ```

7. Save and close the `odbc.ini` file.

Test your ODBC connection

At this point, you can test your ODBC connection to ThoughtSpot. It is important to recall that the username/password you use belongs to a ThoughtSpot application user. Typically, this user is a user with data management or administrative privileges on the application.

1. Before trying the ODBC connection, make sure you can use this username/password to login into the ThoughtSpot application.
2. Confirm the user's privileges by going to the **Data** tab.
3. Go back to your workstation's terminal shell.
4. Use the `/usr/bin/isql` and confirm you can connect.

   Specify the `ThoughtSpot` DSN:

   ```
   /usr/bin/isql -v ThoughtSpot tsadmin adminpwd
   +---------------------------------------+
   | Connected!                            |
   |                                       |
   | sql-statement                         |
   | help [tablename]                      |
   | quit                                  |
   |                                       |
   +---------------------------------------+
   SQL>
   ```

Now, you are ready to begin using the connection you've configured.

Best Practices for Using ODBC

Summary: To successfully use ODBC, following these best practices is recommended.

When developing tools that use the ODBC driver, use these best practices:

- When setting up ODBC for the first time, begin by using the ThoughtSpot `tsload` for the initial data loads. This allows you to do more in-depth troubleshooting on any initial loading issues. After initial loads work properly, switch to ODBC to perform incremental loads.

- You should create the parameterized SQL statement outside of ODBC. Using this method, the SQL statement can be sent to ThoughtSpot in batches by the ODBC driver, so you only have to update the memory itself. ETL tools have this implemented already (end users shouldn't have to actually write the `INSERT` statement). But as a developer, you may be writing code that leverages the ODBC driver, so this tip can help you write your SQL for the best performance with the driver.

- Data can be loaded into a table through multiple parallel connections. You can achieve this by splitting the input data into multiple parts. Then, load those individual parts through multiple parallel connections. You can use parallel loading even while loading to a single table or multiple tables at the same time.

- When doing an incremental data load, note that the same `UPSERT` behavior that occurs in TQL also occurs. This means that if you import a row whose primary key matches an existing row, the existing row will be updated with the new values.

Related information

- Enable ODBC logs [See page 580]
- Introduction to loading and managing data [See page 87]
- Loading and constraints [See page 108]

JDBC Driver Overview

Summary: Use JDBC to interact with databases in a standard manner.

Java Database Connectivity (JDBC) is a Java standard API that allows applications to interact with databases in a standard manner. ThoughtSpot has JDBC support through a JDBC driver that we provide.

Connector type

There are different types of JDBC connectors. Driver types categorize the technology used to connect to the database. The ThoughtSpot JDBC drive is a type 4 connector. It uses Java to implement a networking protocol for communicating with ThoughtSpot.

This driver is Java driver. There is no client installation or configuration.

When to use JDBC

JDBC can be used whenever you want to connect to ThoughtSpot to insert data programmatically from a Java program or application. You should begin by using the ThoughtSpot Loader for initial data loads and then use JDBC for incremental loads. This is because the ThoughtSpot Loader is generally faster than JDBC. Information on using the ThoughtSpot Loader is available in the ThoughtSpot Administrator Guide.

Version Compatibility

To ensure compatibility, always use the JDBC driver with the same version number as the ThoughtSpot instance to which you are connecting.

Performance Considerations

These are some general recommendations for maximizing the performance of JDBC:

- Insert in batches rather than doing single inserts at a time using the

`PreparedStatement::addBatch()` and `PreparedStatement::executeBatch` commands.

- If you need to upload a lot of data, consider running multiple connections with batch inserts in parallel.

> **Note:** The ETL tool must add a data transformation step if the source column data type does not exactly match the target's, ThoughtSpot's, column data type. The driver does not do any implicit conversions.

Use the JDBC Driver

Summary: How to configure the JDBC driver.

ThoughtSpot's ODBC connection relies on the SimbaEngine X SDK [See page 0] to connect through ODBC or JDBC to ThoughtSpot's remote data stores. The instructions on this page explain how to configure the JDBC driver.

The ThoughtSpot JDBC driver is supplied by a `.jar` file you install on a workstation. This workstation is the same machine where you plan to run your ETL activities.

JDBC configuration parameters

Information	Description
Driver name	`com.simba.client.core.jdbc4.SCJDBC4Driver`
Server IP address	The ThoughtSpot appliance URL or IP address.
Simba port	The simba port, which is `12345` by default.
Database name	This is not the machine login username. The ThoughtSpot Database name to connect to.
username	The name of a ThoughtSpot user with administrator permissions.
password	The password of a ThoughtSpot application user. This is not the machine or SSH userpassword.

For more JDBC configuration options, see also:

- JDBC properties reference [See page 594] in this ThoughtSpot documentation
- SimbaClient for JDBC Configuration Properties reference [See page 0]

Check the ThoughtSpot IP and the simba_server status

Before you begin, you need to know the IP address or DNS name of the server you intend to connect your server to.

1. SSH as `admin` or the `thoughtspot` user to your ThoughtSpot node.
2. Verify the node IP(s).

```
$ tscli node ls
172.18.231.17
172.18.231.18
```

3. Make a note of each IP; there may be more than one.
4. Configure the ThoughtSpot firewall to allow connections from your ETL client, by running the following command on any ThoughtSpot node: `tscli firewall open-ports --ports 12345`
5. Exit or close the shell.

Install the driver

The JDBC driver is a `.jar` packaged application. To use the package, you download it, install it

1. Log in to the local machine where you want to install the JDBC driver.
2. Click **Here** [See page 0] to download the JDBC driver.
3. Click **JDBC Driver** to download the file `thoughtspot_jdbc<version>.jar`.
4. Move the driver to the desired directory on your local machine.
5. Add the downloaded JDBC driver to your Java class path on the local machine.

Write your application

Using JDBC with ThoughtSpot is the same as using any other JDBC driver with any other database. You must provide the connection information, create a connection, execute statements, and close the connection.

Specify each of the nodes in the cluster in the connection string, as shown. This enables high availability for JDBC connections. To find out the nodes in the cluster, you can run the command `tscli node ls` from the Linux shell on the ThoughtSpot instance.

The format for the connection is:

```
jdbc:simba://<node1>:12345,<node2>:12345,<node3>:12345[,…];
           LoginTimeout=<seconds>;DATABASE=<db>;SCHEMA=<schema>
```

For example:

```
jdbc:simba://192.168.2.248:12345,192.168.2.249:12345,192.16
8.2.247:12345;
           LoginTimeout=5;DATABASE=test;SCHEMA=falcon_default_s
chema
```

As shown, the `DATABASE` and `SCHEMA` parameters need to be in all caps. For the `simba` JDBC driver to work with Spark, the `DATABASE` and `SCHEMA` must be specified in the URL. They cannot be specified as a name/value pair as a map or property. For example:

```
val tssqldf1 = sparkSession.read.format("jdbc").options(Map("ur
l" ->
"jdbc:simba://10.84.78.181:12345;DATABASE=movieratings;SCHEMA=f
alcon_default_schema", "driver" ->
"com.simba.client.core.jdbc4.SCJDBC4Driver", "dbtable" -> "Movi
es", "user" ->
"tsadmin", "password" -> "admin")).load()
```

This `InsertData.java` example shows how to use ThoughtSpot with JDBC. This is an example of a reference JDBC application:

```
import java.sql.DriverManager;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class InsertData {

  // JDBC class to use.
  private static final String DB_DRIVER = "com.simba.client.cor
e.jdbc4.SCJDBC4Driver";
  // jdbc_example should be an existing database.

  private static final String DB_CONNECTION = "jdbc:simba://19
2.168.2.129:12345;
     192.168.2.249:12345,192.168.2.247:12345;
     LoginTimeout=5;DATABASE=jdbc_example;SCHEMA=falcon_defaul
t_schema";

  private static final String TABLE_NAME = "jdbc_example";
  private static final String DB_USER = "<username>";
  private static final String DB_PASSWORD = "<password>";

  // Assuming everything in local directory use:
  //   javac InsertData.java
  //   java -cp .:thoughtspot_jdbc4.jar InsertData
  public static void main(String[] argv) {

    try {
      insertRecordsIntoTable();
    }
    catch (SQLException e) {
      System.out.println(e.getMessage());
    }
  }

  /**
   * Insert some records using batch updates.
   *  Assumes a table exists:  CREATE TABLE "jdbc_example" ( "t
ext" varchar(10) );
   */
  private static void insertRecordsIntoTable() throws SQLExcept
ion {

    System.out.println("Inserting records.");
    Connection dbConnection = getDBConnection();
```

```
    PreparedStatement preparedStatement = null;
    String insertTableSQL = "INSERT INTO falcon_default_schem
a.jdbc_example (text) VALUES (?)";

    try {
      preparedStatement = dbConnection.prepareStatement(insertT
ableSQL);

      // Create multiple statements and add to a batch update.
      for (int cnt = 1; cnt <= 10; cnt++) {
        preparedStatement.setString(1, "some string " + cnt);
        preparedStatement.addBatch();
        System.out.println("Record " + cnt + " was added to th
e batch!");
      }
      preparedStatement.executeBatch();  // For large numbers o
f records, recommend doing sets of executeBatch commands.
      System.out.println("Records committed");

    }
    catch (SQLException sqle) {
      sqle.printStackTrace();
    }
    finally {

      if (preparedStatement != null) {
        preparedStatement.close();
      }
      if (dbConnection != null) {
        dbConnection.close();
      }
    }
  }

  /** Create a connection to the database. */
  private static Connection getDBConnection() {
    Connection dbConnection = null;
    try {
      Class.forName(DB_DRIVER);
    }
    catch (ClassNotFoundException e) {
      System.out.println(e.getMessage());
    }
    try {
      dbConnection = DriverManager.getConnection(DB_CONNECTIO
```

```
N, DB_USER,DB_PASSWORD);
      return dbConnection;
    }
    catch (SQLException sqle) {
      System.out.println(sqle.getMessage());
    }

    return dbConnection;
  }

}
```

Related Information

- Enable JDBC logs [See page 586]
- Connection configuration [See page 594]
- Supported SQL commands [See page 592]

Set up the JDBC driver for Pentaho

Summary: JDBC to connect to the ThoughtSpot Simba server from Pentaho.

You can use the Pentaho Data Integration (PDI) to create a JDBC connection. The Pentaho Data Integration (PDI) suite is a comprehensive data integration and business analytics platform. You can use it to create a JDBC connection to ThoughtSpot.

PDI consists of a core data integration (ETL) engine and GUI applications that allow you to define data integration jobs and transformations. Through Pentaho, we primarily use the JDBC driver to set up a connection. The process is not as complicated as with SSIS, and is much more lenient.

Community and enterprise editions of PDI are available. Using the community edition is sufficient, though you may use the enterprise edition, which is subscription based, and therefore contains extra features and provides technical support.

Use JDBC to connect to the ThoughtSpot Simba server from Pentaho. The connection will be made between a new ThoughtSpot Table Input and Output objects.

Check the ThoughtSpot IP and the simba_server status

Before you begin, you need to know the IP address or DNS name of the server you intend to connect your server to.

1. SSH as `admin` or the `thoughtspot` user to your ThoughtSpot node.
2. Verify the node IP(s).

```
$ tscli node ls
172.18.231.17
172.18.231.18
```

3. Make a note of each IP; there may be more than one.
4. Configure the ThoughtSpot firewall to allow connections from your ETL client, by running the following command on any ThoughtSpot node: `tscli firewall open-ports --ports`

`12345`

5. Exit or close the shell.

Install the Simba drivers in the Pentaho directories

Before starting the Pentaho Data Integration (PDI) client and creating the connection, ensure that the Simba JDBC client libraries are present in the Pentaho client/server machines. This will ensure that the drivers picked up at runtime.

1. Log in to the local machine where you have alredy installed the Pentaho Data Integration (PDI) client.
2. Click **Here** [See page 0] to download the JDBC driver.
3. Click **JDBC Driver** to download the file `thoughtspot_jdbc<version>.jar`.
4. Copy the `thoughtspot_jdbc<version>.jar` file to the following directories:
   - `<Pentaho_install_dir>/server/data-integration-server/tomcat/webapps/pentaho-di/WED-INF/lib/`
   - `<Pentaho_install_dir>/design-tools/data-integration/lib/`
   - `<Pentaho_install_dir>/server/data-integration-server/tomcat/lib/`
   - `<Pentaho_install_dir>/design-tools/data-integration/plugins/spoon/agile-bi/lib/`

Set up the driver

This section explains how to set up the JDBC driver using Pentaho. These instructions use Spoon, the graphical transformation and job designer associated with the PDI suite. It is also known as the Kettle project.

Create a transformation

Do the following on your ETL workstation with the Pentaho client:

1. Open the PDI client.

```
./spoon.sh &>/dev/null &
```

2. Right click **View > Transformations** tab.

3. Click **New** to create a new transformation.

4. Click **Input** under the **Design** tab to expand it.

5. Drag and drop **CSV File Input** to the **Transformation** window.

   This opens a new CSV file.

6. Double-click the **CSV File Input** icon to open the **CSV Input** dialog .
7. Name the **Step**.
8. Click **Browse** next to the **Filename** field and provide the file you want to read from.

9. Click **OK**.

10. In the CSV Input dialog, click **Get Fields**.
11. Enter the number of lines you would like to sample in the Sample size dialog.

The default setting is 100.

1. Click **OK** when you are ready.

The tool reads the file and suggests the field name and type.

2. Click **Preview** to preview the data.

3. Enter the number of rows to preview in the **Preview size** dialog.

The default setting is 1000. Click **OK** to start the transformation in preview.

4. Examine the preview data, then click **Close**.

You may want to verify that you are able to read the data using the SQL query from ThoughtSpot.

5. Click **OK** in the CSV Input dialog to confirm your CSV input settings.

Define the Output

1. Click **Design > Output**.

2. Drag and drop **Table output** to the **Transformation** window.

3. Double click the **Table output** icon to open the Table output dialog.

4. Enter a **Step name**.

5. Click **New** to create a new connection.

6. Enter or select the following information in the Database Connection dialog:

Field	Description
Connection Name	Any string.
Connection Type	Generic database
Access	Native (JDBC)
Custom Connection URL	`jdbc:simba://SERVER_IP:12345;Database=DATABASE_or_SCHEMA_NAME` </code> The IP is a node in your ThoughtSpot cluster. The name or schema of the database you want to connect to. Use TQL to create a database name if needed. Ensure that there are no leading or trailing spaces.

Custom Driver Class Name	`com.simba.client.core.jdbc4.JDBC4Driver` Ensure that there are no leading or trailing spaces.
User Name	A ThoughtSpot username. If you leave this empty, you are prompted for it at connection time. This user should have **Data Management** privileges on ThoughtSpot.
Password	The password for the **User Name**. If you leave this empty, you are prompted for it at connection time.

7. Click **Test** to test your database connection.

8. If you are able to make a successful connection to the ThoughtSpot Simba Server, click **OK**.

9. Click **OK** in the Database Connection dialog to create the new connection.

Import data

1. In the **Table output** dialog, select the connection you just created.
2. Click **Browse** next to the **Target schema** field and select your **Target schema**.
3. Click **OK** when you are done.
4. Connect the **Input CSV** icon to the **Table output** icon by clicking and dragging an arrow.
5. When prompted, choose **Main output of step**.

6. Double click the **Table output** icon to reopen the **Table output** dialog.
7. Enter a **Target table name**.
8. Click **SQL**.

9. In the **Simple SQL editor** dialog, click **Execute**.

   The system processes and then displays the results of the SQL statements.

10. Close all open dialogs.

11. Click the **Play** button at the top of the **Transformation** window to execute the transformation.

12. Click **Launch** in the **Execute a transformation** dialog.

The system prompts you to save it if you have not already.

13. View the **Execution Results**.

test1
100%
Input CSV
TS Output
Execution Results
Execution History
Logging
Step Metrics
Performance Graph
Metrics
Preview data
Stepname Copynr Read Written Input Output Updated Rejected Errors Active Time Speed (r/s) input/output
1 Input CSV 0 0 5 6 0 0 0 0 Finished 0.0s 2,000 -
2 TS Output 0 5 5 0 5 0 0 0 Finished 1.5s 3 -

Troubleshooting Data Integrations

Summary: Learn how to fix connection issues.

This section can help if you're having trouble creating a connection or need to find out more information about what is going on with ODBC or JDBC.

The information contained here is very basic, and mostly about how to enable logs on the client side. If you need more detailed troubleshooting information or help, please contact ThoughtSpot Support.

- **Enable ODBC Logs [See page 580]**

  If you need more information in order to troubleshoot ODBC connections, you can enable logging for ODBC.
- **Enable JDBC Logs [See page 586]**

  To enable logging for JDBC, add the logging parameters to the connect string. Logs are stored on ThoughtSpot.
- **Schema not found error with ODBC [See page 587]**

  When connecting with ODBC, you need to specify both the database and schema to connect to. If no schema is supplied, you will get an error indicating that the schema could not be found.
- **How to improve throughput of the load [See page 589]**

  The transaction/commit size value can improve the throughput of the load when setting up the ODBC Driver.
- **ODBC tracing on Windows [See page 590]**

  Using logs to aid in troubleshooting.

Enable ODBC Logs

Summary: Learn how to troubleshoot ODBC connections.

If you need more information in order to troubleshoot ODBC connections, you can enable logging for ODBC on the workstation you use for connecting to ThoughtSpot. There are two points where you can enable logging:

- the workstation where you run your ETL activities
- the server where the Simba service is running

On both workstation and servers, the verbosity of the log is controlled by the `LogLevel` property. This property can be one of the following:

- `0` or `LOG_OFF` : no logging occurs
- `1` or `LOG_FATAL` : only log fatal errors
- `2` or `LOG_ERROR` : log all errors
- `3` or `LOG_WARNING` : log all errors and warnings
- `4` or `LOG_INFO` : log all errors, warnings, and informational messages
- `5` or `LOG_DEBUG` : log method entry and exit points and parameter values for debugging
- `6` or `LOG_TRACE` : log all method entry points

Larger values include the information from lessor values. For example, if you set `3` or `LOG_WARNING` , you log all warnings *and* all errors.

Enable ODBC logs on a Windows workstation

To enable ODBC logs on Windows:

1. Open the **ODBC Data Source Administrator** and select the **System DSN** tab.
2. Select your ThoughtSpot data source and click **Configure**.

3. In the Client Configuration Dialog, click **Logging**.

4. Choose a **Log Level**, depending on what level of verbosity you want to show in the logs.

5. For **Log Directory:**, type in the fully qualified path where you want the logs to be saved.

6. Click **OK** to save your settings, and **OK** again, to dismiss the ODBC Data Source Administrator.
7. Run the ODBC load.
8. Locate the log file that was generated, and send it to ThoughtSpot Support with a description of the problem.

Enable ODBC logs on a Linux workstation

To enable logging on Linux, follow these instructions:

1. Navigate to the directory where you installed ODBC.
2. Open the `odbc.ini` file in a text editor.

   This file is the registry and configuration file for ODBC.

3. Locate the `LogLevel` and `LogPath` properties.
4. Uncomment the properties.
5. Enter a value for the `LogLevel` the.

   Acceptable values are from 1 to 6 with 6 being the most verbose.

6. Enter the fully qualified path for the `LogPath` values.

   The log will be written here. Your file will look similar to the following: Example for Linux 64-bit:

```
[ThoughtSpot]
Description = ThoughtSpot 64-bit ODBC Driver
Driver = ThoughtSpot
ServerList =  172.18.231.17 12345
Locale = en-US
ErrorMessagesPath = /home/admin/linux/ErrorMessages
UseSsl = 0
#SSLCertFile = # Set the SSL certificate file path. The
certificate file can be obtained by extracting the SDK t
arball
LogLevel = 3 # Set log level to enable debug logging
LogPath = /home/admin/odbc-logs # Set the debug log file
s path
DATABASE = # Set the default database to connect to
SCHEMA = # Set the default schema to connect to
```

7. Save and close the file.
8. To test the configuration, run the ODBC load and review the log files.

Control logs from the Simba server

You may want to collect logs from the Simba service. Do the following to procedure on every ThoughtSpot node running the Simba service.

1. SSH into the ThoughtSpot node.
2. Edit the `/etc/thoughtspot/linux.ini` file.

```
 ...
 [Driver]

 ## Note that this default DriverManagerEncoding of UT
F-32 is for iODBC. unixODBC uses UTF-16 by default.
 ## If unixODBC was compiled with -DSQL_WCHART_CONVERT,
then UTF-32 is the correct value.
 ## Execute 'odbc_config --cflags' to determine if you n
eed UTF-32 or UTF-16 on unixODBC
 DriverManagerEncoding=UTF-32
 DriverLocale=en-US
 ErrorMessagesPath=/usr/home/linux/ErrorMessages/
 LogLevel=0
 LogNamespace=
 LogPath=

 ....
```

3. Uncomment the `LogLevel` setting.

   The `LogLevel` is the level of logging to capture (0-6).

4. Set `LogPath` to a directory to save the logs.

   The `LogPath` is the fully qualified path where ThoughtSpot should write the logs.

5. Work with ThoughtSpot Support to restart the Simba serivce.

   The node IP may change because of the restart. If this happens, repeat the entire procedure.

Enable JDBC Logs

Summary: Configure logging parameter strings.

To enable logging for JDBC, add the logging parameters to the connect string. Logs are stored on ThoughtSpot. Before enabling JDBC logging, you need:

- The level of logging you want to capture.
- The path on the ThoughtSpot server where the logs will be written. Make sure the directory has the correct permissions so that the “admin” Linux user can write logs to it.

To enable JDBC logging:

1. When forming the connect string for JDBC, add these two parameter, separated by “&”:

   For example:

   ```
    jdbc:simba://192.168.2.248:12345;SERVERS=192.168.2.24
   9:12345,
    192.168.2.247:12345;Database=test;Schema=falcon_defaul
   t_schema;**LogLevel=3;LogPath=/usr/local/scaligent/log
   s**
   ```

   The `LogLevel` is the level of logging to capture (0-6). The `LogPath` is the fully qualified path where logs will be written on ThoughtSpot.

2. Run the JDBC code that uses the connection you modified.
3. Check the `LogPath` directory for logs generated by JDBC.

Schema not found error with ODBC

Summary: Correct schema not found errors.

When connecting with ODBC, you need to specify both the `DATABASE` and `SCHEMA` parameters. This is true even if you do not use schema names in ThoughtSpot. If you don't supply a `SCHEMA`, you get an error indicating that the schema could not be found.

The default schema name in ThoughtSpot is `falcon_default_schema`. To set the `SCHEMA` on Windows, adding a custom property with the key `SCHEMA` and the value `falcon_default_schema`.

On Linux, you can edit the properties in the `odbc.ini` file for the driver you are using:

```
[ThoughtSpot]
Description = ThoughtSpot 64-bit ODBC Driver
Driver = ThoughtSpot
ServerList =  172.18.231.17 12345
Locale = en-US
ErrorMessagesPath = /home/admin/linux/ErrorMessages
UseSsl = 0
#SSLCertFile = # Set the SSL certificate file path. The certifi
cate file can be obtained by extracting the SDK tarball
#LogLevel = 0 # Set log level to enable debug logging
#LogPath = # Set the debug log files path
DATABASE = # Set the default database to connect to
SCHEMA = # Set the default schema to connect to
```

Related information

- Configuring ODBC on Windows [See page 512]
- Configuring ODBC on LINUX [See page 545]
- ODBC and JDBC configuration properties [See page 594]

How to improve throughput

> **Summary:** Adjusting the transaction size may correct poor performance and low throughput.

The transaction/commit size value can improve the throughput of the load when setting up the ODBC Driver.

Adjusting the transaction size may correct poor performance and low throughput issues. The transaction size should be set to match the total number of rows that are expected to be loaded in the load cycle. However, increasing this value even higher should help improve throughput of the load.

> ❗ **Warning:** A high transaction size may slow down the ThoughtSpot system.

This is where the transaction size field exists for SSIS. Clicking on the ODBC destination reveals the properties on the right hand side, where the **Transaction Size** can be found.

See Set up the ODBC Driver for SSIS [See page 534] for more details on setting the transaction size.

ODBC tracing on Windows

Summary: Using logs to aid in troubleshooting.

Windows shows ODBC specific tracing in the ODBC Data Source Administrator Tracing tab. You can start tracing there by clicking **Start Tracing Now**. This logs every ODBC call from this system, and prints the input and output for the call.

Although this is lower level information, it can still be helpful in troubleshooting. When you are not sure if it is our driver or the tool causing an issue, doing this trace will help narrow the inquiry.

If you start or stop tracing, make sure you do not have the SSIS client open. Close it, change the trace, and reopen.

Supported SQL commands

Summary: The ThoughtSpot connection drivers support a limited set of SQL commands.

The ODBC and JDBC drivers support a limited set of SQL commands. When developing software that uses a ThoughtSpot ODBC driver, use this reference of supported commands. This reference is intended for developers using other tools (ETL, etc.) to connect to ThoughtSpot through the ODBC or JDBC driver.

> **Note:** ThoughtSpot displays `VARCHAR` fields using lower case, regardless of what the original casing of your loaded data is.

ODBC

These SQL commands are supported for ODBC:

- `CREATE TABLE`

  Creates a table with the specified column definitions and constraints. The table is replicated on each node.

  ```
  CREATE TABLE country_dim (id_number int, country varch
  ar, CONSTRAINT PRIMARY KEY (id_number));
  ```

- `INSERT`

  Creates placeholders in the table to receive the data.

  ```
  INSERT INTO TABLE country_dim (?, ?);
  ```

- `DELETE FROM <table>`

Deletes `ALL` rows from the specified table. Use the `WHERE` clause to specify only certain rows to be deleted. Example: You could remove all data for sales before a certain date to free up space in ThoughtSpot.

```
DELETE FROM country_dim;
```

- `SELECT <cols_or_expression> FROM <table_list> [WHERE <predicates>] [GROUP BY <expressions>] [ORDER BY <expressions>]`

  Fetches the specified set of table data.

```
SELECT id_number, country FROM country_dim WHERE id_number > 200;
```

JDBC

`TRUNCATE` is not supported. Instead, use `DELETE FROM TABLE` which is functionally equivalent to “truncate table” in terms of table compression and so forth.

Connection configuration

> **Summary:** Lists the properties you can set for ODBC or JDBC connections

This section lists the properties you can set for ODBC or JDBC connections.

Setting Properties for ODBC

Not all the parameters Simba accepts are supported by the ThoughtSpot ODBC clients, and ThoughtSpot has added some properties, which are listed separately here. All configuration properties use the type String (text).

You can set these properties on Windows by using the ODBC Administrator [See page 512] client. For Linux, the properties are located in three files, depending on the property type:

Property Type	Location
DSN	`odbc.ini` file
Driver	`odbsinst.ini` file
SimbaSetting Reader	`simbaclient.ini` file

Setting Properties for JDBC

For JDBC, these properties are passed as key value pairs in the connect string. For more information, see Use the JDBC Driver [See page 559].

Properties Reference

The following tables summarize the configuration properties.

Property	Type	Description
`DATABASE`	DSN or Driver	The default database to connect to.
`SCHEMA`	DSN or Driver	The default schema to connect to.
`Description`	DSN	A brief, human-readable description of the DSN. This describes the DSN to users who are deciding which DSN to use.
`Driver`	DSN or Driver	In the driver configuration location, Driver should contain the path to the driver binary. In the DSN configuration location, Driver could contain the path to the driver binary, or it could contain the driver entry in the registry.
`IdleTimeout`	DSN	The time to wait for a response from the server, in seconds. This property is optional, but SimbaClient will wait indefinitely for SimbaServer to respond to a request made to the server unless you specify a timeout period. `IdleTimeout` specifies how many seconds that SimbaClient will wait before aborting the attempt and returning to the application with an error. This timeout corresponds to ODBC's `CONNECTION_TIMEOUT` property and is only used when more specific timeouts, such as `QUERY_TIMEOUT` or `LOGIN_TIMEOUT` aren't applicable.
`Locale`	DSN	The connection locale. If this value is set, it overrides the driver-wide locale. For example, the driver-wide locale could be `en-US`. If the client would prefer `fr-CA`, it can set the connection locale to `fr-CA`. Values are composed of a 2-letter language code (in lower case), and an optional 2-letter country code (in upper case). If the country code is specified, it must be separated from the language code by a hyphen (-).
`LoginTimeout`	DSN	The timeout, in seconds, to wait for a response from the server when attempting to log in. A value of 0 means no timeout. The default value is 60.
`QueryTimeout`	DSN	The timeout, in seconds, to wait for a response from the server during Prepare, Execute, and ExecuteDirect. A value of 0 means no timeout. The default value is 60.
`ServerList`	DSN	A comma separated list of all servers (IP address and port number) to connect to. SimbaClient must be able to find SimbaServer on the network. This property enables server discovery. SimbaClient will try to make a network connection to the servers in the order specified until a connection is made.
`LogLevel`	SimbaSetting Reader	Controls the granularity of the messages and events that are logged. With this keyword, you can control the amount of log output by controlling the kinds of events that are logged. Possible values (case sensitive): • 0 or `LOG_OFF`: no logging occurs • 1 or `LOG_FATAL`: only log fatal errors • 2 or `LOG_ERROR`: log all errors • 3 or `LOG_WARNING`: log all errors and warnings • 4 or `LOG_INFO`: log all errors, warnings, and informational messages • 5 or `LOG_DEBUG`: log method entry and exit points and parameter values for debugging • 6 or `LOG_TRACE`: log all method entry points

Property	Type	Description
`LogPath`	SimbaSetting Reader	Specifies the directory where the log files are created. For example: `LogPath=C:\Simba Technologies\Temp` If this value is not set, the log files are written to the current working directory of the SimbaClient.
`LogFileSize`	SimbaSetting Reader	The size of each log file, in bytes. The default values is 20971520 bytes. When the maximum size of the file is reached, a new file is created.
`LogFileCount`	SimbaSetting Reader	The number of log files to create. When the maximum number of log files has been created, the oldest file will be deleted and a new one created. The default value is 50.
`username`	UID	Part of a user username/password combination. This combination should correspond to a ThoughtSpot application user with permissions appropriate to your ETL requirements. Typically, this user is a user with data management or administrative privileges on the application.
`password`	Password	Part of a user username/password combination. This combination should correspond to a ThoughtSpot application user with permissions appropriate to your ETL requirements. Typically, this user is a user with data management or administrative privileges on the application.

Keyword reference

You can use keywords when asking a question to help define your search. This reference lists the various keywords. You can also see this list of keywords and examples from within the help center.

Keywords in Other Languages

Currently, we offer the following keyword translations.

日本語 [See page 0]	中文 (简体) [See page 0]	Deutsche [See page 0]	Español (latín) [See page 0]	Español (España) [See page 0]
Français (Canada) [See page 0]	Français (France) [See page 0]	Português (Brasil) [See page 0]	Português (Portugal) [See page 0]	Italiano [See page 0]
Dansk [See page 0]	Suomi [See page 0]	Svenska [See page 0]	Norsk [See page 0]	Nederland [See page 0]

Also, see the topic on how to set locale preferences in your user profile [See page 0] to control language, date, and number formats on the ThoughtSpot UI.

General

Keyword	Examples
top	• **top** sales rep by count sales for average revenue > 10000 • sales rep average revenue for each region **top**
bottom	• **bottom** revenue average • **bottom** revenue by state • customer by revenue for each sales rep **bottom**
n	**top 10** sales rep revenue
n	**bottom 25** customer by revenue for each sales rep

Keyword	Examples
sort by	• `revenue by state` **`sort by`** `average revenue` • `revenue by customer` **`sort by`** `region`

Date

Keyword	Examples
after	`order date` **`after`** `10/31/2014`
before	`order date` **`before`** `03/01/2014`
between … and ...	`order date` **`between`** `01/30/2012 and 01/30/2014`
daily year-over-year	`growth of revenue by order date` **`daily year-over-year`**
daily	`shipments by region` **`daily`**
day	`count` **`monday`** `restaurant`
day of week	`revenue by` **`day of week`** `last 6 months`
day of week	`count shipments` **`Monday`**
n days for each month	`sales last` **`2 days for each month`**
n days for each quarter	`revenue last` **`15 days for each quarter`**
n days for each week	`total sold last` **`2 days for each week`**
n days for each year	`revenue last` **`300 days for each year`**
growth of … by … daily	**`growth of`** `sales` **`by`** `order date` **`daily`**
growth of … by … monthly	**`growth of`** `sales` **`by`** `date shipped` **`monthly`** `sales > 24000`
growth of … by … quarterly	**`growth of`** `sales` **`by`** `date shipped` **`quarterly`**
growth of … by … weekly	**`growth of`** `sales` **`by`** `receipt date` **`weekly`** `for pro-ski2000`
growth of … by … yearly	**`growth of`** `sales` **`by`** `date closed` **`yearly`**

Keyword	Examples
growth of … by ...	`growth of sales by order date`
n hours for each day	`sales last 2 hours for each day`
last day by	`customers last day by referrer`
last month by	`customers last month by day`
last *n* days	`visitors last 7 days`
last *n* quarters	`visitors last 2 quarters by month by campaign`
last *n* weeks	`visitors last 10 weeks by day`
last quarter	`customers last quarter sale > 300`
last week	`customers last week by store`
last year	`top 10 customers last year by sale by store for region west`
last *n* years	`visitors last 5 years by revenue for sum revenue > 5000`
month to date	`sales by product month to date sales > 2400`
month year	`commission by sales rep February 2014`
month	`commission January`
month	`revenue by month last year`
monthly year-over-year	`growth of revenue by receipt date monthly year-over-year`
monthly	`commission > 10000 monthly`
n months for each quarter	`cost last 2 months for each quarter`
n months for each year	`last 8 months for each year`
n days ago	`sales 2 days ago`
n months ago	`sales 2 months ago by region`
n months	`visitors last 6 months for homepage visits > 30 by month`
n quarters ago	`sales 4 quarters ago by product name contains deluxe`
n weeks ago	`sales 4 weeks ago by store`

Keyword	Examples
n years ago	`sales 5 years ago by store for region west`
n years	`opportunities next 5 years by revenue`
next day	`shipments next day by order`
next month	`appointments next month by day`
next *n* days	`shipments next 7 days`
next *n* months	`openings next 6 months location`
next *n* quarters	`opportunities next 2 quarters by campaign`
next *n* weeks	`shipments next 10 weeks by day`
next quarter	`opportunities next quarter amount > 30000`
next week	`shipments next week by store`
next year	`opportunities next year by sales rep`
quarter to date	`sales by product quarter to date for top 10 products by sales`
quarterly year-over-year	`growth of revenue by date shipped quarterly year-over-year`
quarterly	`sales quarterly for each product`
n quarters for each year	`last 2 quarters for each year`
today	`sales today by store`
week to date	`sales by order date week to date for pro-ski200`
week	`revenue by week last quarter`
weekly year-over-year	`growth of revenue by date shipped weekly year-over-year`
weekly	`revenue weekly`
n weeks for each month	`sales last 3 weeks for each month`
n weeks for each quarter	`last 2 weeks for each quarter`
n weeks for each year	`last 3 weeks for each year`

Keyword	Examples
year to date	`sales by product` **`year to date`**
year	`revenue by product` **`2014`** `product name contains snowboard`
yearly	`shipments by product` **`yearly`**
yesterday	`sales` **`yesterday`** `for pro-ski200 by store`

Time

Keyword	Examples
detailed	`ship time` **`detailed`**
last minute	`count homepage views` **`last minute`**
last hour	`count unique visits` **`last hour`**
n minutes	`count visitors last` **`30 minutes`**
n hours	`count visitors last` **`12 hours`**
hourly	`visitors by page name` **`hourly`**
n minutes ago	`sum inventory by product` **`10 minutes ago`**
n hours ago	`sum inventory by product by store` **`2 hours ago`**

Text

Keyword	Examples
begins with	`product name` **`begins with`** `'pro'`

Keyword	Examples
contains	`product name contains "alpine" description` **`contains`** `"snow shoe"`
ends with	`product name` **`ends with`** `'deluxe'`
not begins with	`product name` **`not begins`** `with "tom's"`
not contains	`product color not contains 'tan' product color` **`not contains`** `'red'`
not ends with	`product name` **`not ends with`** `"trial"`
similar to	`course name` **`similar to`** `'hand'`
not similar to	`course name` **`not similar to`** `'hand'`

Number

Function	Examples
sum	**`sum`** `revenue`
average	**`average`** `revenue by store`
count	**`count`** `visitors by site`
max	**`max`** `sales by visitor by site`
min	**`min`** `revenue by store by campaign for cost > 5000`
standard deviation	**`standard deviation`** `revenue by product by month for date after 10/31/2010`
unique count	**`unique count`** `visitor by product page last week`
variance	**`variance`** `sale amount by visitor by product for last year`

Comparative

Function	Examples
all	`all`
between... and	`revenue between 0 and 1000`
vs, versus	`revenue east vs west`
>	`sum sale amount by visitor by product for last year sale amount > 2000`
<	`unique count visitor by product by store for sale amount < 20`
>=	`count calls by employee lastname >= m`
<=	`count shipments by city latitude <= 0`
=	`unique count visitor by store purchased products = 3 for last 5 days`
!=	`sum sale amount region != canada region != mexico`

Location

Keyword	Examples
near	`revenue store name county near san francisco`
near... within *n* miles|km|meters	`revenue store name county near alameda within 50 miles`
farther than *n* miles|km|meters from	`average hours worked branch farther than 80 km from scarborough`

Location keywords only work for searches where the data source includes latitude/longitude data.

Period

Keyword	Example
quarter (*date*)	`quarter (purchase date)`
quarter of year (*date*)	`quarter of year (purchase date)`
month of quarter (*date*)	`month of quarter (purchase date)`
week of year (*date*)	`week of year (ship date)`
week of quarter (*date*)	`week of quarter (ship date)`
week of month (*date*)	`week of month (ship date)`
day of year (*date*)	`day of year (ship date)`
day of quarter (*date*)	`day of quarter (ship date)`
day (*date*)	`day (ship date)`
day of month (*date*)	`day of month (order date)`
day of week (*date*)	`day of week (order date)`
hour (*datetime*)	`hour (timestamp)`

In

Keyword	Example
in (*subsearch*)	`in (top 10 store name by sales footwear)`

TQL reference

TQL is the ThoughtSpot language for entering SQL commands. This reference lists TQL commands you can use to do things like creating a schema or verifying a data load.

About using TQL

You can use TQL either through the ThoughtSpot application's web interface [See page 132] or the command line interface [See page 605] in the Linux shell.

Use `--query_results_apply_top_row_count <number>` flag to limit the number of result rows returned by a query. For example:

```
$ tql --query_results_apply_top_row_count 100
```

As a best practice, you should enclose object names (database, schema, table, and column) in double quotes, and column values in single quotes. When referring to objects using fully qualified object names, the syntax is:

```
"database"."schema"."table"
```

To get help on SQL when using TQL, enter `help` on the command line.

You can use TQL to view and modify schemas and data in tables. Remember to add a semicolon after each command. Commands are not case sensitive but are capitalized here for readability.

> **Note:** Worksheets and pinboards in ThoughtSpot are dependent upon the data in the underlying tables. Use caution when modifying tables directly. If you change or remove a schema on which those objects rely, the objects could become invalid.

View schemas and data

Syntax	Description
`SHOW DATABASES`	Lists all available databases. Examples: `SHOW DATABASES;`
`USE <database>`	Switches the context to the specified database. This is required if queries do not use fully qualified names (database.schema.table) for specifying tables. Examples: `USE "fruit_database";`
`SHOW SCHEMAS`	Lists all schemas within the current database. For example: `SHOW SCHEMAS;`
`SHOW TABLES`	Lists all tables within the current database by schema. For example: `SHOW TABLES;`
`SHOW TABLE <table>`	Lists all the columns for a table. For example: `SHOW TABLE "locations";`
`SCRIPT SERVER`	Generates the TQL schema for all tables in all databases on the server. For example: `SCRIPT SERVER;`

Syntax	Description
`SCRIPT DATABASE <database>`	Generates the TQL schema for all tables in a database. For example: `SCRIPT DATABASE "fruit_database";`
`SCRIPT TABLE <table>`	Generates the TQL schema for a table. For example: `SCRIPT TABLE "vendor";`

Syntax	Description

`SELECT <cols_or_expr> FROM <table_list> [WHERE <predicates>] [GROUP BY <expr>] [ORDER BY <expr>]`	Shows specified set of table data.
	
	If you do not specify the TOP number of rows to select, the top 50 rows will be returned by default. The number of rows to return can be set using the TQL command line flag:
	
	`--query_results apply_top_row_count`
	
	You can use the following aggregation functions:
	• sum
	• count
	• count distinct
	• stddev
	• avg
	• variance
	• min
	• max
	
	You can use the following date functions:
	• absyear
	• absmonth
	• absday
	• absquarter
	• date
	• time
	
	For example:
	
	`SELECT TOP 10 "quantity" FROM "sales_fact";`
	
	`SELECT COUNT(*) FROM "vendor";`
	
	`SELECT "vendor", SUM("quantity") FROM "sales_fact" GROUP BY "vendor";`
	
	`SELECT "vendor", SUM("amount") FROM "vendor", "sales_fact" WHERE "sales_fact"."vendorid" = "vendor"."vendorid" AND "amount" > 100 GROUP BY "vendor" ORDER BY "amount" DESC;`
	
	`SELECT "vendor", SUM("quantity") FROM "sales_fact" GROUP BY "vendor" LIMIT 10;`

Schema creation

Syntax	Description
`CREATE DATABASE <database>`	Creates a database. For example: `CREATE DATABASE "fruit_database";`
`CREATE SCHEMA <schema>`	Creates a schema within the current database. For example: `CREATE SCHEMA "fruit_schema";`

`CREATE TABLE <table> (<column_definitions> [<constraints>]) [PARTITION BY HASH (<number>) [KEY ("<column>")]])`	Creates a table with the specified column definitions and constraints.
	
	Use `PARTITION BY HASH` to shard a table across all nodes. If no `KEY` is specified, the table will be randomly sharded.
	
	Do not specify relationship constraints (`FOREIGN KEY` or `RELATIONSHIP`) in the `CREATE TABLE` statement. Instead, define these using `ALTER TABLE` statements at the end of your TQL script, after creating your tables. This method guarantees that tables are created before they are referenced in the constraint definitions. For example:
	
	`CREATE TABLE "vendor" ("vendorid" int, "name" varchar(255));`
	
	`CREATE TABLE "sales_fact" ("saleid" int, "locationid" int, "vendorid" int, "quantity" int, "sale_amount" double, "fruitid" int, CONSTRAINT PRIMARY KEY("saleid")) PARTITION BY HASH(96) KEY ("saleid");`

Schema modification

Syntax	Description
`DROP DATABASE <database>`	Drops a database and all of its schemas and tables. For example: `DROP DATABASE "fruit_database";`
`DROP SCHEMA <schema>`	Drops a schema within the current database, and drops all of the tables in the schema. For example: `DROP SCHEMA "fruit_schema";`
`DROP TABLE <table>`	Drops a table. For example: `DROP TABLE "location";`
`TRUNCATE TABLE <table>`	Removes all data from a table, but preserves its metadata, including all GUIDs, relationships, etc. This can be used to force a new schema for a table without losing the metadata. However, this operation removes all existing data from the table and must be used with caution. You must reload the data following a `TRUNCATE`, or all dependent objects (worksheets and pinboards) in ThoughtSpot will become invalid. For example: `TRUNCATE TABLE "location";`

Syntax	Description

`ALTER TABLE <table> ADD|DROP|RENAME COLUMN <column>`	Alters a table to add, drop, or rename a column.
	
	When you add a column to an existing table, you must provide a default value to use for existing rows. For example:
	
	`ALTER TABLE "cart" ADD COLUMN "nickname" varchar(255) DEFAULT 'no nickname';`
	
	`ALTER TABLE "cart" DROP COLUMN "nickname";`
	
	`ALTER TABLE "cart" RENAME COLUMN "nickname" TO "shortname";`

`ALTER TABLE <table> DROP CONSTRAINT PRIMARY KEY;`	Drops the primary key from a table.
	
	Note that if you then add a new primary key, the same upsert behavior will be applied as with adding any primary key. This can result in data deletion, so make sure you understand how the upsert will affect your data ahead of time. For example:
	
	`ALTER TABLE "sales" DROP CONSTRAINT PRIMARY KEY;`
	
	`ALTER TABLE "sales" ADD CONSTRAINT PRIMARY KEY ("PO_number");`

`ALTER TABLE <table> DROP CONSTRAINT [FOREIGN KEY| RELATIONSHIP] <name>;`	Drops the named foreign key or relationship between two tables. For example:
	
	`ALTER TABLE "sales_fact" DROP CONSTRAINT FOREIGN KEY "FK_PO_number";`
	
	`ALTER TABLE "fruit_dim" DROP RELATIONSHIP "REL_dates";`

Syntax	Description

`ALTER TABLE <table> DROP [CONSTRAINT FOREIGN KEY [<table_name>] | RELATIONSHIP [WITH <table_name>];`	You must use this syntax when dropping relationships between tables created before ThoughtSpot version 3.2. This is because relationships could not be named in older versions.
	Drops the foreign key or relationship between two tables where you cannot reference it by relationship name. If the relationship was created without a name, use:
	• the name of the referenced table, for a foreign key.
	• the name of the related table, for a relationship.
	
	If you drop a foreign key without specifying the referenced table, all foreign keys from the table you are altering will be dropped.
	
	Examples:
	
	`ALTER TABLE "shipments" DROP CONSTRAINT FOREIGN KEY "orders";`
	
	`ALTER TABLE "wholesale_buys" DROP RELATIONSHIP WITH "retail_sales";`
	
	Drops all relationships that have wholesale_buys as a source.
	
	`ALTER TABLE "wholesale_buys" DROP RELATIONSHIP;`
	
	Drops all foreign keys from wholesale_buys.
	
	`ALTER TABLE "wholesale_buys" DROP CONSTRAINT FOREIGN KEY;`

Syntax	Description

`ALTER TABLE <table> [SET DIMENSION | SET FACT [PARTITION BY HASH [(<shards>)] [KEY(<column>)]]]`	Changes the partitioning on a table by doing one of:
	• re-sharding a sharded table
	• changing a replicated table to a sharded table
	• changing a sharded table to a replicated (unsharded) table
	
	By default, ThoughtSpot does not shard dimension tables.
	
	To change the partitioning on a table, or to change a dimension table to a sharded table, use `ALTER TABLE...SET FACT PARTITION BY HASH...;`
	
	To make a sharded table into a dimension table (replicated on every node), use `ALTER TABLE...SET DIMENSION;` command.
	
	Examples of this statement:
	
	`ALTER TABLE "sales_fact" SET FACT PARTITION BY HASH (96) KEY ("PO_number");`
	`ALTER TABLE "fruit_dim" SET DIMENSION;`

`ALTER TABLE <table> MODIFY COLUMN <column> <new_data_type>;`	Changes the data type of a column. This can have implications on sharding and primary key behavior. See About data type conversion [See page 0]. For example: `ALTER TABLE fact100 MODIFY COLUMN product_id int;`

Modify data

Syntax	Description
`INSERT INTO <table> VALUES ...`	Inserts values into a table. Only use this for testing. Do not use `INSERT` on a production system. For example: `INSERT INTO "vendor" VALUES ('helen rose', 'jacob norse', 'eileen ruff', 'manny gates');`

Syntax	Description
`ALTER TABLE <table> SET LOAD PRIORITY <value> <new_data_type>;`	Sets the load priority for a table. Load priority determines the order in which a table is loaded on a cluster restart. You can set any value from `1-100`. The system default for all tables is `50`. For example: `ALTER TABLE 'sales_facts' SET LOAD PRIORITY 1;`

`UPDATE <table> ... SET ... [WHERE ...]`	Updates rows in a table that match optionally provided predicates. Predicates have the form `column = value` connected by the `AND` keyword. Sets the column values to the specified values. For example:
	
	`UPDATE "location" SET "borough" = 'staten island', "city" = 'new york'`
	`WHERE "borough" = 'staten isl' AND city = 'NY';`

`DELETE FROM <table> [WHERE...]`	Deletes rows from a table that match optionally provided predicates. Predicates have the form `column = value` connected by the `AND` keyword. For example.
	
	`DELETE FROM "vendor" WHERE "name" = 'Joey Smith' AND "vendorid" =`
	`'19463';`

Constraints and relationships

Constraints and relationships in ThoughtSpot are used to define the relationships between tables (how they can be joined). However, constraints are not enforced, as they would be in a transactional database. You can define the following constraints when creating a table with `CREATE TABLE`, or add them to an existing table using the `ADD CONSTRAINT` syntax:

Syntax	Description

PRIMARY KEY	Designates a unique, non-null value as the primary key for a table. This can be one column or a combination of columns. If values are not unique, an upsert will be performed if a row includes a primary key that is already present in the data. Some examples: </p>
	
	`CREATE TABLE "schools" ( "schoolID" varchar(15), "schoolName" varchar(255), "schoolCity" varchar(55), "schoolState" varchar(55), "schoolNick" varchar(55), CONSTRAINT PRIMARY KEY ("schoolID") ) ;`
	
	`ALTER TABLE "cart" ADD CONSTRAINT PRIMARY KEY ("cart_id");`
	
	`ALTER TABLE "cart" DROP CONSTRAINT PRIMARY KEY "cart_id";`

FOREIGN KEY	Defines a relationship where the value(s) in the table are used to join to a second table. Uses an equality operator. The foreign key must match the primary key of the table that is referenced in number, column type, and order of columns.
	
	When creating a foreign key, give it a name. You can reference the foreign key name later, if you want to remove it.
	
	Examples of this statement:
	
	`ALTER TABLE "batting" ADD CONSTRAINT "FK_player" FOREIGN KEY ("playerID") REFERENCES "players" ("playerID");`
	
	`ALTER TABLE "batting" ADD CONSTRAINT "FK_lg_team" FOREIGN KEY ("lgID" ,"teamID") REFERENCES "teams" ("lgID" ,"teamID");`
	
	`ALTER TABLE "shipment" ADD CONSTRAINT "FK_PO_vendor" FOREIGN KEY ("po_number", "vendor") REFERENCES "orders" ("po_number", "vendor");`
	
	`ALTER TABLE "shipment" DROP CONSTRAINT "FK_PO_vendor";`

Syntax	Description

RELATIONSHIP	Defines a relationship where the value(s) in the table can be used to join to a second table, using an equality condition (required) and one or more range conditions (optional). These conditions act like a WHERE clause when the two tables are joined. They are applied using AND logic, such that all conditions must be met for a row to be included.
	
	You may add multiple relationships between tables. When creating a relationship, give it a name. You can reference the relationship name later, if you want to remove it.
	
	Examples of this statement:
	
	`ALTER TABLE "wholesale_buys" ADD RELATIONSHIP "REL_fruit" WITH "retail_sales" AS "wholesale_buys"."fruit" = "retail_sales"."fruit" AND ("wholesale_buys"."date_order" < "retail_sales"."date_sold" AND "retail_sales"."date_sold" < "wholesale_buys"."expire_date");`
	
	`ALTER TABLE "wholesale_buys" DROP RELATIONSHIP "REL_fruit";`

Data types

ThoughtSpot supports a simplified list of data types:

Syntax	Description	Examples
Character	• VARCHAR(*n*)	Specify the maximum number of characters, as in VARCHAR(255). The size limit is 1GB for VARCHAR values.
Floating point	• DOUBLE • FLOAT	DOUBLE is recommended.
Boolean	• BOOL	Can be `true` or `false`.

Syntax	Description	Examples
Integer	• INT • BIGINT	INT holds 32 bits. BIGINT holds 64 bits.
Date or time	• DATE • DATETIME • TIMESTAMP • TIME	DATETIME, TIMESTAMP, and TIME are stored at the granularity of seconds . TIMESTAMP is identical to DATETIME, but is included for syntax compatibility.

tsload flag reference

For recurring data loads and for scripting loads, use `tsload` (the ThoughtSpot Loader). This reference section lists all the flags that can be used to modify the behavior of `tsload`.

General tsload flags

Flag	Description	Notes
`--target_database <database>`	Specifies the pre-existing target database into which tsload should load the data.	
`--target_schema <schema>`	Specifies the target schema.	Default is “falcon_default_schema”.
`--target_table <table>`	Specifies the tables that you want to load into the database.	The tables must exist in the database specified by `--target_database`.
`--empty_target`	Specifies that any data in the target table is to be removed before the new data is loaded.	If supplied, any rows that exist in the table specified by `--target_database` and `--target table` will be deleted before this data load. To perform an “upsert” on the existing data, omit this flag or specify `--noempty_target`.
`--max_ignored_rows <number>`	Specifies the maximum number of rows that can be ignored if they fail to load.	If the number of ignored rows exceeds this limit, the load will be aborted.
`--bad_records_file <path_to_file>/<file_name>`	Specifies the file to use for storing rows that failed to load.	Input rows that do not conform to the defined schema in ThoughtSpot will be ignored and inserted into this file.
`--date_format <date_formatmask>`	Specifies the format string for date values.	The default format is `yearmonthday` e.g. “Dec 30th, 2001” and is represented as `20011230`. Use the date format specifications supported in the strptime library function [See page 0].
`--date_time_format <date_formatmask> <time_formatmask>`	Specifies the format string for datetime values.	The default is yearmonthday hour:minute:second e.g. Dec 30th, 2001 1:15:12 and is represented as 20011230 01:15:12. Use the datetime format specifications supported in the strptime library function [See page 0].
`--time_format <time_formatmask>`	Specifies the format string for time values.	The default is hour:minute:second. Use the time format specifications supported in the strptime library function [See page 0].

Flag	Description	Notes
`--v=[0|1|2|3`	Specifies the verbosity of log messages.	Provide a value for verbosity level. By default, verbosity is set to the minimum, which is 0. This value is similar to a volume control. At higher levels your log receives more messages and that log more frequently. This is used for debugging. You should not change this value unless instructed by ThoughtSpot Support.
`--skip_second_fraction`	Skips fractional seconds when loading data.	If supplied, the upserts logic may be affected, especially if the date time being loaded is a primary key, and the data has millisecond granularity. Load the data twice, first time as a string with a primary key, and again with second granularity date time. There is no support to store fractional seconds in the ThoughtSpot system.

File loading tsload flags

The following flags are used when loading data from an input file:

Flag	Description	Notes
`--source_file <path_to_file>/<file_name>`	Specifies the location of the file to be loaded.	
`--source_data_format [csv|delimited]`	Specifies the data file format.	Optional. The default is csv.
`--field_separator "<delimiter>"`	Specifies the field delimiter used in the input file.	
`--trailing_field_separator`	Specifies that the field separator appears after every field, including the last field per row.	Example row with trailing field separator: a,b,c,The default is false.
`--null_value "<null_representation>"`	Specifies how null values are represented in the input file.	These values will be converted to NULL upon loading.
`--date_converted_to_epoch [true|false]`	Specifies whether the “date” or “datetime” values in the input file are represented as epoch values.	

Flag	Description	Notes
`--boolean_representation [true_false | 1_0 | T_F | Y_N]`	Specifies the format in which boolean values are represented in the input file.	The default is T_F. You can also use this flag to specify other values. For example, if your data used Y for true and NULL for false, you could specify: `--boolean_representation Y_NULL`
`--has_header_row`	Indicates that the input file contains a header row.	If supplied, column names in the header row are used to match column names in the target table in ThoughtSpot. If not supplied, the first row of the file is loaded as data, the same as all subsequent rows.
`--escape_character "<character>"`	Specifies the escape character used in the input file.	If no value is specified, the default is “(double quotes).
`--enclosing_character "<character>"`	Specifies the enclosing character used in the input file.	If the enclosing character is double quotes, you need to escape it, as in this example: `--enclosing_character "\""`
`--use_bit_boolean_values = [true | false]`	Specifies how boolean values are represented in the input file.	If supplied, the input CSV file uses a bit for boolean values, i.e. the false value is represented as 0x0 and true as 0x1. If omitted or set to false, boolean values are assumed to be T_F, unless you specify something else using the flag `--boolean_representation [true_false | 1_0 | T_F | Y_N]` .

tscli command reference

Summary: The ThoughtSpot command line interface, or tscli, is an administration interface for the cluster. Use tscli to take snapshots (backups) of data, apply updates, stop and start the services, and view information about the system. This reference defines each subcommand.

The command returns `0` on success, and a non-zero exit codes on failure. The `tscli` command logs errors to the `stderr` directory.

How to use the tscli command

The `tscli` command has the following syntax:

```
tscli [-h]
      [--helpfull]
      [--verbose]
      [--noautoconfig]
      [--autoconfig]
      [--yes]
      [--cluster ]
      [--zoo ]
      {access, alert, ansible, backup, backup-policy, calenda
r, callhome, cassandra,
       cluster, command, dr-mirror, etl, event, feature, filese
rver,
       firewall, hdfs, ipsec, ldap, logs, map-tiles, monitorin
g, nas,
       node, notification, onboarding, patch, rpackage, saml, s
cheduled-pinboards, set, smtp, snapshot,
       snapshot-policy, socialproof spot, sssd, ssl, storage, s
upport,
       tokenauthentication}
```

The `tscli` command has several subcommands, such as `alert`, `backup`, and so on.

Issue subcommands using the following format:

```
tscli [subcommand]
```

Subcommands have their own additional options and actions, such as `tscli backup create` or `tscli backup delete`.

Each subcommand may have several options.

To view help for a subcommand, type `-h` after the subcommand option:

```
tscli [subcommand] -h
```

tscli subcommands

This section lists each subcommand and its syntax.

access

```
tscli access [-h] {list}
```

This subcommand has the following option:

`tscli access list`

Lists objects by last access time, with the following parameters:

`--type TYPE`

Type of object, either answer or pinboard.

`--limit LIMIT`

The number of objects to fetch.

The default is 30.

`--offset OFFSET`

Offset to use to skip objects for batched results.

The default is 0.

`--ascending`

Sorts the answers by access time ascending.

The default is `True`.

alert

```
tscli alert [-h] {count,info,list,off,on,refresh,silence,status,unsilence}
```

This subcommand has the following options:

`tscli alert count`

Lists counts of generated alerts by type.

`tscli alert info`

Lists all alerts. Add `silenced` to list only silenced alerts, `active` to list only active alerts, or `detailed` to get detailed alert information.

`tscli alert list`

Lists the generated alerts, with these parameters:

`--limit LIMIT`

Specifies the number of recent alerts to display.

`--since SINCE`

Lists all alerts raised since a specified time period, in the form of a human readable duration string, such as 4h (4 hours) or 4m (4 minutes).

`tscli alert off`

Disables all alerts from the cluster in the cluster's timezone.

`tscli alert on`

Enables alerts from the cluster.

`tscli alert refresh`

Refreshes alert metadata on the cluster.

`tscli alert silence --name NAME`

Silences the alert with `NAME`. For example, `DISK_ERROR`. Silenced alerts are still recorded in postgres, however emails are not sent out.

`tscli alert status`

Shows the status of cluster alerts.

`tscli alert unsilence --name NAME`

Unsilences the alert with `NAME`. For example, `DISK_ERROR`.

ansible

```
tscli ansible [-h] {checkout,commit} [--local]
```

This subcommand has the following options:

`tscli ansible checkout --host HOST`

Checks out Ansible playbook with the target `HOST` that is running the ts_ansible service.

`tscli ansible commit --host HOST`

Commits Ansible playbooks with the target `HOST` that is running the ts_ansible service.

Use this subcommand to install and configure third-party software on the ThoughtSpot cluster.

For details, see these articles:

- About third party security and monitoring software [See page 198]
- Installing third party security and monitoring software [See page 201]

backup

```
tscli backup [-h] {create,delete,ls,restore}
```

This subcommand has the following options:

`tscli backup create [-h] [--mode {full,light,dataless}] [--type {full,incremental}] [--base BASE]`
`[--storage_type {local,nas}] [--remote] [--no-orion-master]`

Pulls a snapshot and saves it as a backup, with these parameters:

`--mode {full,light,dataless}`

Mode of backups.

The default is `full`.

`--type {full,incremental}`

Type of backup.

Note: `incremental` is not implemented.

The default setting is `full`.

`--base BASE`

Based snapshot name for incremental backup.

Note: Because `incremental` is not implemented, neither is this option.

There is no default setting.

`--storage_type {local,nas}`

Storage type of output directory.

The default setting is `local`.

`--remote`

Takes backup through orion master.

The default setting is `True`.

`--no-orion-master`

Determines whether orion master is available during backup.

The default is `False`.

`tscli backup delete *name*`

Deletes the named backup.

`tscli backup ls`

Lists all backups taken by the system.

`tscli backup restore`

Restores cluster using backup, with the following parameters:

`--release RELEASE`

Restore the cluster on a specific release number.

`--disable_rotate_keys`

Disables cluster rotate key configurations.

The default is `False`.

`--enable_cloud_storage`

Determines whether to enable Cloud Storage setup.

--heterogeneous

Should be set for heterogeneous clusters.

The default is `False`.

backup-policy

```
tscli backup-policy [-h] {create,delete,disable,enable,ls,sho
w,status,update}
```

Manages the backup policy.

This subcommand has the following options:

`tscli backup-policy create`

Prompts an editor for you to edit the parameters of a new periodic backup policy, with the following parameter:
`--config CONFIG`

Specifies the text format of the periodic backup policy config.

`tscli backup-policy delete name`

Deletes the backup policy `name`.

`tscli backup-policy disable name`

Disables the policy `name`.

`tscli backup-policy enable name`

Enables the policy `name`.

`tscli backup-policy ls`

Lists backup policies.

`tscli backup-policy show name`

Shows the backup policy `name`.

`tscli backup-policy status name`

Shows the status of the backup policy `name`.

`tscli backup-policy update name`

Prompts an editor for you to edit the backup policy `name`.

calendar

```
tscli calendar [-h] {create,delete,disable,enable,generate,ge
t,list,update}
```

This subcommand has the following options:

`tscli calendar create`

Creates a new custom calendar, with the following parameters:

`--file_path FILE_PATH`

Path to the CSV file holding custom calendar data.

The default is `None`.

`--name NAME`

Custom calendar name.

The default is `None`.

`--separator SEPARATOR`

The separator used in the CSV file.

The default is `,`.

`--no-header-row`

Flag to indicate that the CSV file has no header row.

The default is `True`.

`--username USERNAME`

The admin username for ThoughtSpot login.

The default is `None`.

`tscli calendar delete`

Deletes a custom calendar table from the system, with the following parameters:

`--name NAME`

Deletes the custom calendar NAME.

The default is `None`.

`--username USERNAME`

The admin username for ThoughtSpot login.

The default is `None`.

`tscli calendar disable`

Disables custom calendar on the cluster.

`tscli calendar enable`

Enables custom calendar on the cluster.

`tscli calendar generate`

Creates a custom calendar table based on given specifications, with the following parameters:

`--name NAME`

A name to create the custom calendar CSV file with.

The default is `None`.

`--start_date START_DATE`

The start date to begin the custom calendar with in the form mm/dd/yyyy.

The default is `None`.

`--end_date END_DATE`

The end date to end the custom calendar with in the form mm/dd/yyyy.

The default is `None`.

`--calendar_type {MONTH_OFFSET,4-4-5,4-5-4,5-4-4}`

The type of custom calendar to create.

The default is `MONTH_OFFSET`.

`--month_offset {January,February,March,April,May,June,July,August,September,October,November,December}`

The month offset to start the year from, if the calendar is the MONTH_OFFSET type.

The default is `January`.

`--start_day_of_week`

The day the week starts on.

The default is `Sunday`.

`--quarter_name_prefix`

The string to prefix a quarter name with.

`--year_name_prefix YEAR_NAME_PREFIX`

The string to prefix a year name with.

`--username USERNAME`

The admin username for ThoughtSpot login.

The default is `None`.

`tscli calendar get`

Procures data of a custom calendar as a CSV file, with the following parameters:

`--name NAME`

Procures data of custom calendar `NAME`

The default is `None`.

`--username USERNAME`

Admin username for ThoughtSpot login.

The default is `None`.

`tscli calendar list`

Procures a list of custom calendars present in the cluster, with the following parameter:

`--username USERNAME`

Admin username for ThoughtSpot login.

The default is `None`.

`tscli calendar update`

Updates a custom calendar table in the system, with the following parameters:

`--file_path FILE_PATH`

Path to the CSV file holding custom calendar data.

The default is `None`.

`--name NAME`

Custom calendar name.

The default is `None`.

`--separator SEPARATOR`

The separator used in the CSV file.

The default is `,`.

`--no-header-row`

Flag to indicate that the CSV file has no header row.

The default is `True`.

`--username USERNAME`

The admin username for ThoughtSpot login.

The default is `None`.

callhome

```
tscli callhome [-h] {disable,enable,generate-bundle}
```

This subcommand has the following options:

`tscli callhome disable`

Turns off the periodic call home feature.

`tscli callhome enable --customer_name CUSTOMER_NAME`

Enables the "call home" feature, which sends usage statistics to ThoughtSpot.

This feature is enabled by default.

The parameter `customer_name` takes the form `Shared/CUSTOMER_NAME/stats`.

The default is `None`.

`tscli callhome generate-bundle [--d D] [--since SINCE]`

Generates the callhome stats tar file, with the following parameters:

`--d D`

Destination folder for the tar file.

There is no default setting.

`--since SINCE`

Grabs `callhome` data from the specified time window in the past.

This should be a human-readable duration string, such as `4h` (4 hours), `30m` (30 minutes), `1d` (1 day).

This option generates a `tar` file of the cluster metrics and writes it to the specified directory, where `SINCE` is how many days back the file must start.

There is no default setting.

cassandra

```
tscli cassandra [-h] {backup,restore}
```

Backs up cassandra.

This subcommand has the following options:

`tscli cassandra backup`

Takes a backup of cassandra, with the following parameters:
`--keyspaces KEYSPACES`

Comma separated list of keyspaces to take a backup of.

The default is `None`.

`backup_dir BACKUP_DIR`

The path to the backup directory to write the backup.

The default is `None`.

`tscli cassandra restore`

Restores cassandra from a backup, with the following parameter:
`--backup_dir BACKUP_DIR`

The path to the backup directory to write the backup.

The default is `None`.

cluster

```
tscli cluster [-h] abort-reinstall-os,abort-update,bucket-  nam
e,check,create,download-release,get-config,list-available-relea
ses,list-downloaded-releases,load,reinstall-os,restore,resume-r
einstall-os,resume-update,set-config,set-min-resource-spec,setu
p-release-host,setup-release-host-key,show-resource-spec,star
t,status,stop,update,update-hadoop}
```

This subcommand has the following options:

`tscli cluster abort-reinstall-os`

Aborts in-progress reinstall.

`tscli cluster abort-update`

Aborts an ongoing cluster update, if safe.

`tscli cluster bucket-name`

Returns the name of the s3 bucket associated with the cluster, if there is one.

`tscli cluster check [--path PATH] [--includes INCLUDES] [--retry RETRY] [--localhost] [--disable-events]`

Checks the status of all nodes in the cluster, with the following parameters:

`--path PATH`

Specifies the working directory of the diagnostic tool.

The default is `/usr/local/scaligent/release`.

`--includes INCLUDES`

Specifies the comma-separated component(s) to be included in the check.

The default is `all`.

`--retry RETRY`

The maximum number of retry times if the node is unreachable.

The default is `10`.

`--localhost`

Runs cluster checks only on localhost.

The default is `False`.

`--disable-events`

Disables raising configuration events.

The default is `False`.

`tscli cluster create *release*`

Creates a new cluster from the release file specified by adding the release number.

Used by ThoughtSpot Support when installing a new cluster. For example, `tscli cluster create 5.3.2.tar.gz`.

This command has the following parameters:

`--disable_rotate_keys`

Disables cluster rotate key configuration.

The default is `False`.

`--enable_cloud_storage {s3a,gcs}`

Determines whether to enable Cloud Storage setup, and which storage format to use.

`heterogeneous`

Should be set for hetergenous clusters.

The default is `False`.

`tscli cluster download-release *release*`

Downloads the specified release to the Hadoop Distributed File Sytem (HDFS) for later upgrading.

`tscli cluster get-config`

Gets current cluster network and time configuration. Prints JSON configuration to `stdout`.

If the system cannot be connected to all interfaces, the command returns an error but continues to function.

This command has the following parameters:

`--local`

Gets the config for the local host only.

The default is `False`.

`--nodes NODES`

A comma separated list of specified nodes to get the config for.

The default is `None`.

`tscli cluster list-available-releases`

Lists the available releases to update to on the cluster.

`tscli cluster list-downloaded-releases`

Lists the releases downloaded to the cluster.

`tscli cluster load *backupdir*`

Loads the state from a specified backup directory onto an existing cluster.

`tscli cluster reinstall-os`

Reinstalls OS on all nodes of the cluster, with the following parameters:

`--secondary SECONDARY`

A secondary drive for reinstall.

The default is `sdd`.

`--stdin`

Command to take JSON configuration from stdin.

`tscli cluster restore --release RELEASE *backupdir*`

Restores a cluster on the specified release number using the backup in the specified directory *backupdir*. If you're restoring from a dataless backup, you must supply the release tarball for the corresponding software release.

This command has the following parameters:

`--disable_rotate_keys`

Disables cluster rotate key configurations.

The default is `False`.

`--enable_cloud_storage {s3a,gcs}`

Determines whether to enable Cloud Storage setup.

`--heterogenous`

Should be set for heterogenous clusters.

The default is `False`.

`tscli cluster resume-reinstall-os`

Resumes in-progress reinstall.

`tscli cluster resume-update`

Resumes in-progress updates, with the following parameter:
`--ignore_if_unhealthy`

Comma separated list of node IPs on which upgrade is not attempted if they are found to be unhealthy. If a node outside of this list is found unhealthy, the upgrade is aborted.

The default is `None`.

`tscli cluster set-config`

Sets cluster network and time configuration. Takes JSON configuration from stdin.

This subcommand has the following parameters:

`--ipv4-only`

Only use ipv4 for node communication. Requires passing ipMap in config unless no-network-change is also specified.

The default is `False`.

`--no-network-change`

This flag ensures that a change made with set-config does not update network settings.

The default is `False`.

`tscli cluster set-min-resource-spec`

Sets the minimum resource configuration of the cluster.

`tscli cluster show-resource-spec`

Prints default or min.

`tscli cluster start`

Starts the cluster.

`tscli cluster status`

Gives the status of the cluster, including release number, date last updated, number of nodes, pending tables time, and services status.

`tscli cluster stop`

Pauses the cluster (but does not stop storage services).

`tscli cluster update`

Update existing cluster.

`tscli cluster update-hadoop`

Updates Hadoop/Zookeeper on the cluster.

command

```
tscli command [-h] {run}
```

Command to run on all nodes.

This subcommand has the following option:

`tscli command run [-h] [--nodes NODES] --dest_dir DEST_DIR [--copyfirst COPYFIRST] [--timeout TIMEOUT] command`

These are the parameters:

`--nodes NODES`

Space-separated IPs of nodes where to run the command.

The default setting is `all`.

`--dest_dir DEST_DIR`

Directory to save the files that contain the output from each node.

This is a mandatory parameter.

`--copyfirst COPYFIRST`

Copy the executable to required nodes first.

The default setting is `False`.

`--timeout TIMEOUT`

Timeout waiting for the command to finish.

The default setting is `60`.

dr-mirror

```
tscli dr-mirror [-h] {start,status,stop}
```

This subcommand has the following options:

`tscli dr-mirror start`

Starts a mirror cluster which will continuously recover from a primary cluster.

`tscli dr-mirror status`

Checks whether the current cluster is running in mirror mode.

`tscli dr-mirror stop`

Stops mirroring on the local cluster.

etl

```
tscli etl [-h] {change-password,disable-lw,download-agent,enabl
e-lw,show-lw}
```

This subcommand has the following options:

`tscli etl change-password --admin_username admin_user --username Informatica_user`

Changes the Informatica Cloud account password used by ThoughtSpot Data Connect.

Required parameters are:

`--admin_username admin_user`

Specifies the Administrator username for ThoughtSpot.

`--username Informatica_user`

Specifies the username for the Informatica Cloud.

`tscli etl disable-lw`

Disables ThoughtSpot Data Connect.

`tscli etl download-agent`

Downloads the ThoughtSpot Data Connect agent to the cluster.

`tscli etl enable-lw [-h] --username USERNAME --thoughtspot_url THOUGHTSPOT_URL --admin_username ADMIN_USERNAME [--groupname GROUPNAME] --org_id ORG_ID [--pin_to PIN_TO] [--proxy_host PROXY_HOST] [--proxy_port PROXY_PORT] [--proxy_username PROXY_USERNAME] [--max_wait MAX_WAIT]`

Contact ThoughtSpot Support for assistance in setting this up.

Required parameters are:

`--username USERNAME`

Username for Informatica Cloud

`--thoughtspot_url THOUGHTSPOT_URL`

URL to reach thoughtspot.

`--admin_username ADMIN_USERNAME`

Admin username for ThoughtSpot

`--groupname GROUPNAME`

`--org_id ORG_ID`

Specifies the Informatica `id` of the company.

`--pin_to PIN_TO`

Specifies the IP address to pin to. If you specify an IP to pin to, that node becomes sticky to the Informatica agent, and will always be used. Defaults to the public IP address of the localhost where this command was run.

`--proxy_host PROXY_HOST`

Proxy server host for network access.

`--proxy_port PROXY_PORT`

Proxy server port.

`--proxy_username PROXY_USERNAME`

Proxy server username.

`--max_wait MAX_WAIT`

Maximum time in seconds to wait for Data Connect agent to start.

`tscli etl show-lw`

Shows the status of ThoughtSpot Data Connect.

It also returns the Informatica username and OrgId.

event

```
tscli event [-h] {list}
```

This subcommand and its options manage event notifications.

This subcommand has the following option:

`tscli event list [-h] [--include INCLUDE] [--since SINCE] [--from FROM] [--to TO] [--`

`limit LIMIT] [--detail] [--summary_contains SUMMARY_CONTAINS] [--detail_contains DETAIL_CONTAINS] [--attributes ATTRIBUTES]`

The `event` subcommand accepts these optional flags:

`--include INCLUDE`

Options are all, config, notification. Default config. (default: config)

`--since SINCE`

Grab events from this time window in the past. Should be a human readable duration string, such as `4h` (4 hours), `30m` (30 minutes), `1d` (1 day).

`--from FROM`

Begin timestamp. Must be of the form: `yyyymmdd-HH:MM`.

`--to TO`

End timestamp. Must be of the form: `yyyymmdd-HH:MM`.

`--limit LIMIT`

Maximum number of events to fetch.

The default setting is `0`.

`--detail`

Print events in detail format. This is not tabular. Default is a tabular summary.

The default setting is `False`.

`--summary_contains SUMMARY_CONTAINS`

Summary of the event will be checked for this string. Multiple strings to check for can be specified by separating them with `|` (pipe). The event is returned if it `matchesALL`. Put single quotes around the param value to prevent undesired glob expansion.

`--detail_contains DETAIL_CONTAINS`

Details of the event will be checked for this string. Multiple strings to check for can be specified by separating them with `|` (pipe). The event is returned if it `matches ALL`. Put single quotes around the param value to prevent undesired glob expansion.

`--attributes ATTRIBUTES`

Specify attributes to match as key=value. Multiple strings to check for can be specified by separating them with `|` (pipe). The event is returned if it `matches ALL`. Put single quotes around the param value to prevent undesired glob expansion.

feature

```
tscli feature [-h] {get-all-config}
```

This subcommand has the following option:

`tscli feature get-all-config`

Gets the configured features in a cluster. The command returns a list of features, such as custom branding, Data Connect, and call home, and informs whether they are enabled or disabled.

fileserver

```
tscli fileserver [-h] {configure,download-release,purge-confi
g,show-config,upload}
```

This subcommand has the following options:

`tscli fileserver configure [-h] --user USER [--password PASSWORD]`

Configures the secure file server username and password for file upload/download, and the call home feature.

You only have to issue this command one time, to set up the connection to the secure file server. Reissue this command if the password changes.

The parameter `PASSWORD` is optional. If a password is not specified, you will be prompted to enter it.

`tscli fileserver download-release [-h] [--user USER] [--password PASSWORD] release`

Downloads the specified release file, including its checksum, and verifies the integrity of release bundle.

You must specify the exact release number, such as `5.1.3`.

Before using this command for the first time, you must set up the file server connection using `tscli fileserver configure`. You can then work with a member of the ThoughtSpot Support team becuase a privileged `user` and a corresponding `password` must be specified to download releases.

`tscli fileserver purge-config`

Removes the file server configuration.

`tscli fileserver show-config`

Shows the file server configuration.

`tscli fileserver upload [-h] [--user USER] [--password PASSWORD] --file_name FILE_NAME --server_dir_path SERVER_DIR_PATH`

Uploads the specified file to the directory specified on the secure file server.

You may optionally specify the `user` and `password` to bypass the credentials specified when configuring the file server connection with `tscli fileserver configure`. Before using this command for the first time, you must set up the file server connection using `tscli fileserver configure`.

This uses the following flags:

`--user USER`

Username of the fileserver.

`--password PASSWORD`

Password of the fileserver. This is required and the command prompts you for it if you do not supply it.

`--file_name FILE_NAME`

Local file to upload.

`--server_dir_path SERVER_DIR_PATH`

Directory path on fileserver. The `SERVER_DIR_PATH` parameter specifies the directory for file upload. It is based on customer name, and takes the form `/Shared/support/customer_name`.

firewall

```
tscli firewall [-h] {close-ports,disable,enable,open-ports,status}
```

This subcommand has the following options:

`tscli firewall close-ports`

Closes specified ports through firewall on all nodes.

Accepts a comma-separated list of ports. Only closes ports that were previously opened using `open-ports`, and ignores ports that were not opened with `open-port`, or closed ports.

Some essential ports are always kept open, such as `ssh`; they are not affected by this command or by `open-ports`.

`tscli firewall disable`

Disable firewall.

`tscli firewall enable`

Enable firewall.

`tscli firewall open-ports --ports ports`

Opens specified ports through a firewall on all nodes.

Accepts a comma-separated list of ports.

Ignores open ports.

Some essential ports are always kept open, such as `ssh`; they are not affected by this command or by `close-ports`.

`tscli firewall status`

Shows whether firewall is currently enabled or disabled.

hdfs

```
tscli hdfs [-h] {leave-safemode}
```

This subcommand has the following option:

`tscli hdfs leave-safemode`

Command to get HDFS `namenodes` out of `safemode`.

ipsec

```
tscli ipsec [-h] {disable,enable,status}
```

This subcommand has the following options:

`tscli ipsec disable`

Disable IPSec

`tscli ipsec enable`

Enable IPSec

`tscli ipsec status`

Show IPSec status on all nodes

ldap

```
tscli ldap [-h] {add-cert,configure,purge-configuration}
```

This subcommand has the following options:

`tscli ldap add-cert name certificate`

Adds an SSL certificate for LDAP. Use only if LDAP has been configured without SSL and you wish to add it. Use `name` to supply an alias for the certificate you are installing.

`tscli ldap configure`

Configures LDAP using an interactive script.

`tscli ldap purge-configuration`

Purges (removes) any existing LDAP configuration.

logs

```
tscli logs [-h] {collect,runcmd}
```

Manages the logging behavior.

This subcommand has the following options:

`tscli logs collect [-h] [--include INCLUDE] [--exclude EXCLUDE] [--since SINCE] [--from FROM] [--to TO] [--out OUT] [--maxsize MAXSIZE] [--sizeonly] [--nodes NODES]`

Extracts logs from the cluster. Does not include any logs that have been deleted due to log rotation.

These parameters have the following values:

`--include INCLUDE`

Specifies a comma separated list of logs to include. Each entry is either a "selector" or a glob for matching files. Selectors must be among: `all`, `orion`, `system`, `ts`. Anything starting with `/` is assumed to be a glob pattern, and it is interpreted through `find(1)`. Other entries are ignored. Put single quotes around the param value to prevent undesired glob expansion (default: `all`)

`--exclude EXCLUDE`

Comma separated list of logs to exclude. Applies to the list selected by `--include`. Params are interpreted just like in `--include`.

`--since SINCE`

Grab logs from this time window in the past. Should be a human-readable duration string, such as `4h` (4 hours), `30m` (30 minutes), `1d` (1 day). (default: None)

`--from FROM`

Timestamp where collection begins; must be of the form `yyyymmdd-HH:MM`.

`--to TO`

Timestamp where collection ends; must be of the form `yyyymmdd-HH:MM`.

`--out OUT`

Tarball path for writing logs from each node.

The default setting is `/tmp/logs.tar.gz`.

`--maxsize MAXSIZE`

Only fetch logs if size is smaller that this value. Can be specified in megabytes or gigabytes, such as `100MB`, `10GB`.

`--sizeonly`

Do not collect logs. Just report the size.

The default setting is `False`.

`--nodes NODES`

Comma separated list of nodes from where to collect logs. Skip this to use all nodes.

`tscli logs runcmd [-h] --cmd CMD [--include INCLUDE] [--exclude EXCLUDE] [--since SINCE] [--from FROM] [--to TO] [--outfile OUTFILE] [--outdir OUTDIR] [--cmd_infmt CMD_INFMT] [--cmd_outfmt CMD_OUTFMT] [--nodes NODES]`

Runs a Unix command on logs in the cluster matching the given constraints. Results are reported as text dumped to standard out, the specified output file, or as tarballs dumped into the specified directory.

Accepts these optional flags:

`--cmd CMD`

Unix-Command to be run on the selected logs. Use single quotes to escape spaces etc. Note the language used to specify CMDSTR has following rules.

- A logfile and its corresponding result file can be referred by keywords `SRCFILE` and `DSTFILE`. For example, `cp SRCFILE DSTFILE`.
- Without any reference to `DSTFILE` in CMDSTR, `DSTFILE` will be appended to CMDSTR for output redirection. For example, `du -sch SRCFILE` gets auto-transtalted to `du -sch SRCFILE > DSTFILE`.
- Without any reference to SRCFILE, content of log is streamed to CMDSTR by pipe. For example, `tail -n100 | grep ERROR` gets auto-transtalted to `cat SRCFILE | tail -n100 | grep ERROR > DSTFILE`.

`--include INCLUDE`

Comma-separated list of logs to include,each entry is either a "selector" or a glob for matching files.

Selectors must be one of `all`, `orion`, `system`, `ts`.

Anything that starts with `/` (forward slash) is assumed to be a glob pattern and interpreted through `find(1)`. Other entries are ignored.

TIP: use single quotes around the parameter value to prevent undesired glob expansion.

The default setting is `all`.

`--exclude EXCLUDE`

Comma separated list of logs to exclude. Applies to the list selected by `--include`. Params are interpreted just like in `--include`.

`--since SINCE`

Grab logs from this time window in the past. Should be a human-readable duration string, such as `4h` (4 hours), `30m` (30 minutes), `1d` (1 day).

`--from FROM`

Timestamp where collection begins; must be of the form `yyyymmdd-HH:MM`.

`--to TO`

Timestamp where collection ends; must be of the form `yyyymmdd-HH:MM`.

`--outfile OUTFILE`

File path for printing all results. By default printed to `stdout`

`--outdir OUTDIR`

Directory path for writing results with original directory structure from each node. Used as an alternative to printing output to `outfile/stdout`.

`--cmd_infmt CMD_INFMT`

Specify if the input file should be compressed or uncompressed before running `CMD`. `C` for compressed, `U` for uncompressed. Don't use this flag if `CMD` works on both.

`--cmd_outfmt CMD_OUTFMT`

Specify if `OUTFILE` generated by `CMD` should be compressed or uncompressed. `C` for compressed, `U` for uncompressed. Don't use this flag if output file is of the same format as the input file.

`--nodes NODES`

Comma separated list of nodes where to run command. Skip this to use all nodes.

map-tiles

```
tscli map-tiles [-h] {disable,enable,status}
```

This subcommand has the following options:

`tscli map-tiles enable [-h] [--online] [--offline] [--tar TAR] [--md5 MD5]`

Enables ThoughtSpot's map tiles, used when constructing geomap charts.

If you don't have internet access, you must download the map tiles tar and md5 files, and append the following to the `tscli` command:

`--online`

Download `maptiles` tar from internet.

The default setting is `True`

`--offline`

Using `maptiles` tar from local disk.

The default setting is `False`

`--tar` *`TAR`*

Specified tar file for map-tiles.

`--md5` *`MD5`*

Specified md5 file for map-tiles.

`tscli map-tiles disable`

Disable map-tiles functionality.

`tscli map-tiles status`

Check whether map-tiles are enabled.

monitoring

```
tscli monitoring [-h] {set-config,show-config}
```

This subcommand has the following options:

`tscli monitoring set-config [-h] [--email EMAIL] [--clear_email] [--heartbeat_interval HEARTBEAT_INTERVAL] [--heartbeat_disable] [--report_interval REPORT_INTERVAL] [--report_disable]`

Sets the monitoring configuration.

The `monitoring` subcommand accepts the following optional flags:

`--email` *`EMAIL`*

Comma separated list (no spaces) of email addresses where the cluster will send monitoring information.

`--clear_email`

Disable emails by clearing email configuration. (default: False)

`--heartbeat_interval HEARTBEAT_INTERVAL`

Heartbeat email generation interval in seconds. Should be greater than 0.

`--heartbeat_disable`

Disable heartbeat email generation. (default: False)

`--report_interval REPORT_INTERVAL`

Cluster report email generation interval in seconds.

Should be greater than `0`.

`--report_disable`

Disable cluster report email generation.

The default setting is `False`.

`tscli monitoring show-config`

Shows the monitoring configuration.

nas

```
tscli nas [-h] {ls,mount-cifs,mount-nfs,unmount}
```

This subcommand has the following options:

`tscli nas ls [-h]`

List mounts managed by NAS mounter service.

`tscli nas mount-cifs [-h] --server SERVER [--path_on_server PATH_ON_SERVER] --mount_point MOUNT_POINT--username USERNAME --password PASSWORD [--uid UID] [--gid GID] [--options OPTIONS]`

Mounts a CIFS device on all nodes.

Accepts the following optional flags:

`--server SERVER`

IP address or DNS name of CIFS service.

For example, `10.20.30.40`.

`--path_on_server PATH_ON_SERVER`

Filesystem path on the CIFS source server to mount NAS.

For example, `/a`.

The default setting is `/` (forward slash).

`--mount_point MOUNT_POINT`

Directory on all cluster nodes where to mount the NFS filesystem on the target.

If this directory does not exist, the command creates it. If this directory already exists, the command uses it for mounting.

For example, `/mnt/external`.

`--username USERNAME`

Username to connect to the CIFS filesystem

`--password PASSWORD`

CIFS password for `--username`

`--uid UID`

The *UID* that owns all files or directories on the mounted filesystem when the server does not provide ownership information.

See `man mount.cifs` for more details.

The default setting is `1001`.

`--gid GID`

The `GID` that owns all files or directories on the mounted filesystem when the server does not provide ownership information.

See `man mount.cifs` for more details.

The default is `1001`.

`--options OPTIONS`

Other command-line options to forward to the `mount.cifs` command.

The default setting is `noexec`.

`tscli nas mount-nfs [-h] --server SERVER [--protocol PROTO --path_on_server PATH_ON_SERVER] --mount_point MOUNT_POINT [--options OPTIONS]`

Mounts a NFS device on all nodes.

Accepts the following optional flags:

`--server SERVER`

IP address or DNS name of NFS service. For example, `10.20.30.40`.

`--path_on_server PATH_ON_SERVER`

Filesystem path on the NFS source server.

For example, `/a/b/c/d`.

The default setting is `/`.

`--mount_point MOUNT_POINT`

Directory on all cluster nodes of the target system.

If this directory does not exist, the command creates it. If this directory already exists, the command uses it for mounting.

For example, `/mnt/external`.

`--options OPTIONS`

Command-line options to mount.

The default setting is `noexec`.

`--protocol PROTO`

One of `nfs` or `nfs4`.

The default is `nfs`.

`tscli nas unmount [-h] --dir DIR`

Unmounts all devices from the specified directory, `DIR`.

This command returns an error if nothing is currently mounted on this directory through `tscli nas mount`.

node

```
tscli node [-h] {check,ls,reinstall-os,resume-reinstall-os,status}
```

This subcommand has the following options:

`tscli node check [-h] [--select {reinstall-preflight}] [--secondary SECONDARY]`

Run checks per node.

Accepts the following flags:

`--select {reinstall-preflight}`

Select the type of node check

The default setting is `reinstall-preflight`.

`--secondary SECONDARY`

Secondary drive for `reinstall-preflight`.

The default setting is `sdd`.

`tscli node ls [-h] [--type {all,healthy,not-healthy}]`

Filter by node state.

The default setting is `all`.

`tscli node reinstall-os [-h] [--secondary SECONDARY] [--cluster]`

Reinstall OS on a node.

Accepts the following flags:

`--secondary SECONDARY`

Secondary drive for reinstall.

The default setting is `sdd`.

`--cluster`

The node part of a cluster.

The default setting is `False`.

`tscli node resume-reinstall-os`

Resume in-progress reinstall

onboarding

```
tscli onboarding [-h] {configure,purge-configuration}
```

Onboarding helps application administrators to bulk update user information. In particular, it configures various in-app email options.

This subcommand has the following options:

`tscli onboarding --help`

Prints help for the onboarding configuration

`tscli onboarding configure`

Configures the onboarding through series of steps.

Asks the user to provide information necessary for onboarding-related functionality, such as the following:

1. Company name
2. Product name
3. Should welcome emails be enabled?
    - Send welcome emails to new users
    - Support email
    - Custom message to include in emails
    - URL of the ThoughtSpot instance
    - URL of the ThoughtSpot documentation

`tscli onboarding purge-configuration`

This command removes all onboarding configuration.

patch

```
tscli patch [-h] {apply,ls,resume-apply,resume-rollback,rollbac
k}
```

This subcommand has the following options:

`tscli patch apply [-h] [release]`

Apply the patch on an existing cluster.

Accepts the following flag:

`release`

The relative path to the patch tar ball.

`tscli patch ls [-h] [--applied] [--rolled_back] [--service SERVICE] [--md5 MD5] [--`

`history]`

Lists the patches currently applied.

Accepts the following flags:

`--applied`

Show only the patches applied since last full release.

The default setting is `False`.

`--rolled_back`

Show only the patches rolled back since last full release.

The default setting is `False`.

`--service SERVICE`

Show patches filtered by service.

The default setting is `None`.

`--md5 MD5`

Shows the details of the patch specified.

The default setting is `None`.

`--history`

Shows the history of all patch apply/rollback release.

The default setting is `False`.

`tscli patch resume-apply [-h]`

Resume patch apply

`tscli patch resume-rollback [-h]`

Resume patch roll-backup

`tscli patch rollback [-h]`

Rollback the patch from an existing cluster

rpackage

```
tscli rpackage [-h] {add,delete,list}
```

Manages R packages available to SpotIQ.

This subcommand has the following options:

`tscli rpackage add [-h] [--repo REPO] [--timeout TIMEOUT] [--dest_dir DEST_DIR] [--nodes NODES] package_name`

Command to add an R package, `package_name`, to the cluster.

Accepts the following flags:

`--repo REPO`

Specify the url of a specific repository to download packages.

`-timeout REPO`

Timeout waiting for the R Package to be installed (default: 60)

`--dest_dir REPO`

Directory where output of this command will be placed

`--nodes NODES`

Space-separated list of IPs for nodes where to run the command.

The default setting is `all`.

`tscli rpackage add [-h] [--timeout TIMEOUT] [--dest_dir DEST_DIR] [--nodes NODES] package_name`

Command to delete an installed R package from the cluster.

Accepts the following flags:

`--timeout REPO`

Timeout waiting before removing the R package.

The default is 60

`--dest_dir REPO`

Directory where to save the output of this command.

`--nodes NODES`

Space-separated list of node IPs where to run the command.

The default setting is `all`.

`tscli rpackage list [-h] [--detailed]`

List all R packages installed on the cluster.

saml

```
tscli saml [-h] {configure,purge-configuration}
```

This subcommand has the following options:

`tscli saml configure [-h]`

Configures SAML.

`tscli saml purge-configuration`

Purges any existing SAML configuration.

To see a list of prerequisites, refer to Configure SAML [See page 41].

scheduled-pinboards

```
tscli scheduled-pinboards [-h] {disable,enable}
```

This subcommand has the following options:

`tscli scheduled-pinboards disable [-h]`

Disable scheduled pinboards for this cluster.

`tscli scheduled-pinboards enable [-h]`

Enables scheduled pinboards, which is disabled in prod clusters by default.

> **Note:** When you enable scheduled pinboards, you should also configure a whitelist of intended email domains. Contact ThoughtSpot Support for help on how to configure a whitelist.

smtp

```
tscli smtp [-h] {remove-mailfromname,remove-mailname,remove-relayhost,remove-saslcredentials,reset-canonical-mapping,set-canonical-mapping,set-mailfromname,set-mailname,set-relayhost,set-saslcredentials,show-canonical-mapping,show-mailfromname,show-mailname,show-relayhost}
```

This subcommand has the following options:

`tscli smtp remove-mailfromname`

Removes current cluster mail from name.

`tscli smtp remove-mailname`

Removes current cluster mail name.

`tscli smtp remove-relayhost`

Removes current cluster relay host.

`tscli smtp remove-saslcredentials`

Clears SASL credentials and disables SMTP AUTH.

`tscli smtp reset-canonical-mapping`

Deletes the current postmap mapping.

`tscli smtp set-canonical-mapping [-h] new_key new_value`

Sets a new Postmap mapping.

`tscli smtp set-mailfromname mailfromname`

Sets the name and an email address from where email alerts are sent for the cluster.

`tscli smtp set-mailname mailname`

Sets the mailname and a domain from where email alerts are sent for the cluster.

`tscli smtp set-relayhost [-h] [--force FORCE] relayhost`

Sets the Relay Host for SMTP (email) sent from the cluster.

Accepts the following flag:

`--force FORCE`

Set even if relay host is not accessible.

The default setting is `False`.

`tscli smtp set-saslcredentials`

Sets SASL credentials and enables SMTP AUTH

`tscli smtp show-canonical-mapping`

Shows the current postmap mapping.

`tscli smtp show-mailfromname`

Shows the mailname, from which email alerts are sent, for the cluster.

`tscli smtp show-mailname`

Shows the mailname, where email alerts are sent, for the cluster.

`tscli smtp show-relayhost`

Shows the for SMTP (email) sent from the cluster.

If there a relay host is not configured, the command returns `NOT FOUND`.

snapshot

```
tscli snapshot [-h] {backup,create,delete,ls,pin,restore,unpin,update-ttl}
```

To learn more about snapshots and backups, see the Understand the backup strategies [See page 248] documentation.

This subcommand has the following options:

`tscli snapshot backup [-h] [--mode {full,light,dataless}] [--type {full,incremental}] [--base *BASE*] [--storage_type {local,nas}] [--remote] *name out*`

Pull snapshot out as a backup.

Accepts the following flags:

`--mode {full,light,dataless}`

Mode of backups. (default: `full`)

`name`

Name of snapshot to pull out as a backup. To list all snapshots, run `tscli snapshot ls`.

`out`

Directory where backup will be written, must not already exist.

`--type {full,incremental}`

Type of backup.

Incremental backup is not implemented.

The default setting is `full`.

`--base BASE`

Based snapshot name for incremental backup.

Incremental backup not implemented yet.

`storage_type {local,nas}`

Storage type of output directory. (default: `local`)

`--remote`

Take backup through Orion master. (default: `True`)

`tscli snapshot create [-h] name reason ttl`

Creates a new snapshot with the specified `name` and `reason`.

This command does not accept `.` (periods). It does accept `-` (dashes or hyphens).

The `ttl` parameter is the number of days after which this snapshot is automatically deleted. A value of `-1` disables automatic deletion.

`tscli snapshot pin [-h] name`

Pins a snapshot so it cannot be deleted or garbage collected.

`tscli snapshot delete [-h] name`

Deletes the named snapshot.

`tscli snapshot ls [-h]`

Lists available snapshots.

`tscli snapshot restore [-h] [--allow_release_change] [--only_service_state] name`

Restores cluster to an existing snapshot.

Accepts the following flags:

`--allow_release_change`

Allow restoration to a snapshot at a different release. (default: `False`)

`--only_service_state`

Restore only service state. (default: `False`)

`tscli snapshot unpin [-h] name`

Unpin a snapshot so it can be deleted or garbage-collected

`tscli snapshot update-ttl [-h] [--disable DISABLE] name ttl`

Updates manual snapshot garbage collection policy.

Accepts the following flags:

`name`

Specifies which snapshot to update.

`ttl`

Extends the manual snapshot

`ttl`

This is the "time-to-live" value.

Use a positive value to increase `ttl`. Use negative values to decrease it.

`--disable DISABLE`

Disable manual snapshot garbage collection.

Setting this value to `True` will override any `ttl` value. (default: False)

snapshot-policy

```
tscli snapshot-policy [-h] {disable,enable,show,update}
```

This subcommand has the following options:

`tscli snapshot-policy disable [-h]`

Disable snapshot policy.

`tscli snapshot-policy enable -h`

Enable specified snapshot policy.

`tscli snapshot-policy show [-h]`

Show snapshot policy.

`tscli snapshot-policy update [-h] [--config CONFIG]`

Update periodic snapshot config. This takes the following parameter

`--config CONFIG`

Text format of periodic backup policy config.

spot

```
tscli spot [-h] {enable}
```

Enables Spot integration.

This subcommand has the following option:

`tscli spot enable [-h] --token TOKEN --thoughtspot_url THOUGHTSPOT_URL [--cache_timeout CACHE_TIMEOUT]`

The `spot` subcommand accepts the following optional flags:

`--token TOKEN`

Slack authorization token for Spot bot. This is required. You receive this token when your Slack administrator adds the Spot application.

`--thoughtspot_url THOUGHTSPOT_URL`

URL for the ThoughtSpot application.

This is required.

`--cache_timeout CACHE_TIMEOUT`

Internal cache timeout.

The default setting is `60000`.

ssl

```
tscli ssl [-h] {add-cert,clear-min-tls-version,off,on,rm-cert,set-min-tls-version,status,tls-status,add-valid-hosts}
```

This subcommand manages the SSL configuration.

To use SSL, the following ports must be open:

- 443

- 80

This subcommand has the following options:

`tscli ssl add-cert [-h] key certificate`

Adds an SSL certificate, key pair.

`tscli ssl clear-min-tls-version [-h]`

Clears any customizations for the minimum TLS version to support.

`tscli ssl off`

Disables SSL. Disabling SSL will stop users from seeing a security warning when accessing ThoughtSpot from a browser if there is no SSL certificate installed.

`tscli ssl on [-h]`

If SSL is enabled and there is no certificate, users will see a security warning when accessing ThoughtSpot from a browser.

`tscli ssl rm-cert`

Removes the existing SSL certificate, if any.

`tscli ssl set-min-tls-version [-h] {1.0,1.1,1.2}`

Sets the minimum supported TLS version. Sets the minimum SSL version to be supported by the ThoughtSpot application. Please ensure that client browsers are enabled for this version or newer.

`tscli ssl status`

Shows whether SSL authentication is enabled or disabled.

`tscli ssl tls-status [-h]`

Prints the status of TLS support.

`tscli ssl add-valid-hosts [-h] VALID_HOSTS`

Enables host validation for the specified host(s). Helps improve security. This feature is for all customers that have or are planning to enable SSL. Multiple hosts must be separated by a comma (,).

Examples:

1. If you want to make sure the valid host is **cluster1.corp.example.com**, you would run the command:
`tscli ssl add-valid-hosts cluster1.corp.example.com`

2. If you want to allow all hosts which have the suffix **corp.example.com**, you would run the command:
`tscli ssl add-valid-hosts *.corp.example.com`. This wild card should be used within the hostname.

3. If you want to allow multiple valid hosts, for example both ***.corp.example.com** and **cluster1**, you would run the command:
`tscli ssl add-valid-hosts *.corp.thoughtspot.com,cluster1`

sssd

```
tscli sssd {enable, disable, set-sudo-group, clear-sudo-group}
```

This subcommand uses system security services daemon (SSSD), and has the following options:

`tscli sssd enable --user USER --domain DOMAIN`

Enables system Active Directory (AD) user access on a single node. You will be prompted for password credentials. The user must have permission to join a computer or VM to the domain.

`tscli sssd disable`

`tscli sssd set-sudo-group ACTIVE_DIRECTORY_GROUP_NAME`

Allows `sudo` permissions for AD group.

`tscli sssd clear-sudo-group ACTIVE_DIRECTORY_GROUP_NAME`

Clears any set AD sudo group.

For more about setting up Active Directory access, see Enable Active Directory based access [See page 44].

storage

```
tscli storage [-h] {gc,df}
```

This subcommand has the following options:

`tscli storage gc [-h] [--log_age LOG_AGE] [--force] [--localhost_only]`

Garbage collect unused storage.

Before issuing this command, you must stop the cluster using `=tscli cluster stop`.

After garbage collection finishes, you can restart the cluster with `tscli cluster start`.

This command frees space in the following directories:

```
/tmp
    /usr/local/scaligent/logs/
    /export/logs/orion
    /export/logs/oreo
    /export/logs/hadoop
    /export/logs/zookeeper
    cores
```

The `storage` subcommand accepts these optional flags:

`--log_age` *`LOG_AGE`*

Deletes logs older than these many hours. Use a non-zero value, becuase zero deletes all temporary files, including the ones that are closed temporarily, while they are passed from one component to the next.

The default setting is `4`.

`--force`

Forces deletion of all logs and temporary files regardless of age. This must only be run on a stopped cluster.

The default setting is `False`.

`--localhost_only`

If used, only the logs on the localhost will be removed. If not specified, the command acts on the entire cluster.

`tscli storage df [--mode disk|hdfs]`

Checks the disk usage on the relevant mounts. Returns output similar to the Linux system command `df -h` *`directory`*.

support

```
tscli support [-h]
   {bundle, restart-remote, rm-admin-email, rm-admin-phone, rm-
feedback-email,
    set-admin-email, set-admin-phone, set-debug-ui-password, se
t-feedback-email,
    set-remote, show-admin-email, show-admin-phone, show-feedba
ck-email,
    show-remote, start-remote, stop-remote}
```

This subcommand has the following options:

`tscli support bundle [-h] [--include INCLUDE] [--exclude EXCLUDE] [--list_selectors] [--since SINCE] [--from FROM] [--to TO] [--out OUT] [--nodes NODES]`

`--include INCLUDE`

Comma-separated list of selectors to include. Each entry is either a "selector" or a glob for matching files. To see the list of valid selectors, run this command with `--list_selectors`. You may also specify `all` to get all selectors and logs, and `basic` to get only the basic selectors.

Selectors can be used for logs collection: `all`, `orion`, `system`, `ts`, or the name of a service.

Anything that starting with `/` (forward slash) is assumed to be a glob pattern, and it is interpreted through `find(1)`. Other entries are ignored.

TIP: Use single quotes around the param value to prevent undesired glob expansion. Use `all` to collect all selectors and all logs.

The default setting is `all_but_logs`.

`--exclude EXCLUDE`

Comma-separated list of selectors to exclude. Applies to the list selected by `--include`. Params are interpreted in the same manner as in `--include`.

Use the special keyword `logs` to exclude logs collection altogether.

There is no default setting.

`--list_selectors`

List the selectors available for `--include` and `--exclude`, and then exit.

The default setting is `False`.

`--since SINCE`

Grabs logs from this time window in the past. Should be a human-readable duration string, such as `4h` (4 hours), `30m` (30 minutes), `1d` (1 day).

There is no default setting.

`--from FROM`

Timestamp when collection begins. Must be of the form: `yyyymmdd-HH:MM`.

There is no default setting.

`--to TO`

Timestamp when collection ends. Must be of the form: `yyyymmdd-HH:MM`.

There is no default setting.

`--out OUT`

Tarball path for dumping the support bundle.

The default setting is `/tmp/support_bundle.tar.gz`.

`--nodes NODES`

Comma separated list of nodes from where to collect logs. Skip this to use all nodes.

There is no default setting.

`tscli support restart-remote`

Restarts remote support.

`tscli support rm-admin-email`

Removes the email address for contacting the customer administrator. Replaces it with the default ThoughtSpot Support email address.

`tscli support rm-feedback-email`

Removes the email address for product feedback. Replaces it with the default ThoughtSpot Support email address

`tscli support rm-admin-phone`

Removes the phone number for contacting the customer administrator. Replaces it with the default ThoughtSpot Support phone number

`tscli support rm-feedback-email`

Removes the email for sending feedback out of the system. To set a blank email address, issue the command `tscli support set-feedback-email`.

`tscli support set-admin-email email`

Sets the email address for contacting the customer administrator. To display a blank email address, issue the command `tscli support set-admin-email`.

`tscli support set-feedback-email email`

Sets the email address for sending feedback. To display a blank email address, issue the command `tscli support set-feedback-email`.

`tscli support set-admin-phone phone_number`

Sets the phone number for contacting the customer administrator. Specify a phone number using any value, such as `+1 800-508-7008 Ext. 1`. To display a blank phone number, issue the command `tscli support set-admin-phone`.

`tscli support set-remote [-h] [--addr ADDR] [--user USER]`

Configures the cluster for remote support through SSH tunneling, where *`ADDR`* is the address of support, such as `tunnel.thoughtspot.com`, and *`USER`* is the support username.

`tscli support show-admin-email`

Shows the email address for customer administrator, if set

`tscli support show-feedback-email`

Shows the email address for product feedback, if set

`tscli support show-admin-phone`

Shows the phone number for customer administrator, if set

`tscli support show-remote`

Shows the status and configuration of remote support

`tscli support start-remote`

Starts remote support

`tscli support stop-remote`

Stops remote support

tokenauthentication

```
tscli cli tokenauthentication [-h] {disable,enable}
```

This subcommand has the following options:

`tscli cli tokenauthentication enable`

Generates a token

`tscli cli tokenauthentication disable`

Purges token login configuration

Date and time formats reference

This is a reference for the date and time contexts and formats you can use with ThoughtSpot. You define data formats in specific contexts and, depending on the context, your choices in data formatting differ. You must understand date and time when you load data in these contexts:

- using data upload from the browser
- through `tsload` command
- through an extract, transform, load (ETL) tool

Data loading formats do not change how data is displayed in tables and charts.

The context where you *can control* date and time formats is data modeling. Data modeling controls how data is displayed in search and their resulting answers.

Data loading formats through tsload

When loading through the `tsload` command you must specify `date` and `timestamp` formats using the format specifications defined in the `strptime` library function [See page 0]. Data is imported based on the timezone of the node from which `tsload` is run.

For `date` data types, the default format is `%Y%m%d`, which translates to `yearmonthday` For example, `Dec 30th, 2001` is represented as `20011230`. For `time` and `datetime` data types, the default is `%Y%m%d %H:%M:%S` which translates to `yearmonthday hour:minute:second`, for example, `Dec 30th, 2001 1:15:12` is represented as `20011230 01:15:12`.

Data modeling formats for browser data upload

These date and time formats are supported in a CSV file when uploading through the browser. You cannot specify the date format; ThoughtSpot will pick the format that fits your data best:

- 1/30/2014
- 2014-01-30
- 2014-1-30
- 30-Jan-2014
- 2014-Jan-30

- 2014-01-30 10:32 AM
- 2014-01-30 14:52
- 2014-01-30 10:32:22
- 2014-01-30 10:32:22 AM
- 2014-01-30 10:32:22.0
- 2014-01-30 10:32:22.0 AM
- 2014-01-30 10:32:22.000
- 2014-01-30 10:32:22.000 AM
- 1/30/2014
- 30-Jan-14
- 01-Mar-02 (assumes 2002)
- 30/1/2014 10:32 AM
- 30/1/2014 14:52
- 30/1/2014 10:32:22
- 30/1/2014 10:32:22 AM
- 30/1/2014 10:32:22.0
- 30/1/2014 10:32:22.0 AM
- 30/1/2014 10:32:22.000
- 30/1/2014 10:32:22.000 AM
- 30-Jan-14 10:32 AM
- 30-Jan-14 14:52
- 30-Jan-14 10:32:22
- 30-Jan-14 10:32:22 AM
- 30-Jan-14 10:32:22.0
- 30-Jan-14 10:32:22.0 AM
- 30-Jan-14 10:32:22.000
- 30-Jan-14 10:32:22.000 AM
- Fri Jan 30 2014 3:26 PM
- Fri Jan 30 2014 13:46
- Fri Jan 30 2014 10:32:22
- Fri Jan 30 2014 10:32:22 AM
- Fri Jan 30 2014 10:32:22.0
- Fri Jan 30 2014 10:32:22.0 AM
- Fri Jan 30 2014 10:32:22.000
- Fri Jan 30 2014 10:32:22.000 AM
- 14:52

- 10:32 AM
- 10:32:22
- 10:32:22 AM
- 10:32:22.0
- 10:32:22.000
- 10:32:22.0 AM
- 10:32:22.000 AM

Data loading formats through an ETL tool

Data loads through ETL uses ODBC or JDBC connections. After you extract the data from the source but before you load it into ThoughtSpot, you must transform any date or timestamp columns into a format that is valid for ThoughtSpot. After the data transformation completes, there is no requirement for explicit data masking. See the data integration guide for more information on loading data through ODBC and JDBC.

Data modeling formats

A user with administrative rights can configure data modeling for data on one or all files. You can set number, date, and currency display formats. These formats define how these value types display in tables and charts. See the Admin Guide for more information about data modeling settings. The following format strings are available for use:

Format mask	Description
`YYYY` or `yyyy`	four digit year such as `2017`
`YY` or `yy`	last two digits of year such as `17`
`M`	month with no leading zero `1 - 12`
`MM`	Two digit month `01 - 12`
`MMM`	Three letter month such as `Jan`
`D`	Day of year without a leading zero `0 - 365`
`DD`	Day of year with up to one leading zero `01 - 365`
`DDD`	Day of year with up to two leading zeroes `001 - 365`

Format mask	Description
`d`	Day of month with no leading zero `1 - 31`
`dd`	Two digit day of month `01 - 31`
`HH`	Two digit `24` hour representation of hour `00 - 23`
`hh`	Two digit `12` hour representation of hour `01 - 12`
`H`	24 hour representation of hour with no leading zero `0 - 23`
`h`	12 hour representation of hour with no leading zero `1 - 12`
`mm`	Minutes `00 - 59`
`m`	Minutes with no leading zero `0 - 59`
`ss`	Seconds `00 - 59`
`s`	Seconds with no leading zero `0 - 59`
`a`	AM/PM indicator

Valid delimiters include most non-alphabet characters. This includes but is not limited to:

- `\` (forward slash)
- `/` (backward slash)
- `|` (pipe symbol)
- `:` (colon)
- `-` (dash)
- `_` (underscore)
- `=` (equal sign)

Examples of valid format masks you can produce for display are as follows:

- `MM/dd/yyyy`
- `MMM`
- `DD/MM/yyyy`
- `MM/dd/yyyy HH:mm`
- `DD/MM/yyyy HH:mm`

Row level security rules reference

ThoughtSpot allows you to create row level security rules using expressions. If an expression evaluates to “true” for a particular row and group combination, that group will be able to see that row. This reference lists the various operators and functions you can use to create rules.

For information on how to use the row level security functions and operators, see About Rule-Based Row Level Security [See page 223]. There is a special variable called ts_groups, which you can use when creating row level security rules. It fetches a list of the groups that the currently logged in user belongs to. For each row, f the expression in the rule evaluates to ‘true’ for any one of these groups, that row will be shown to the user.

You can also see this list of operators and examples from within the Rule Builder by selecting **Rule Assistant**.

Conversion functions

These functions can be used to convert data from one data type to another. Conversion to or from date data types is not supported.

Function	Description	Examples
`to_bool`	Returns the input as a `boolean` data type (`true` or `false`).	`to_bool (0) = false` `to_bool (married)`

Function	Description	Examples
`to_date`	Accepts a date represented as an integer or text string, and a second string parameter that can include `strptime` date formatting elements. Replaces all the valid strptime date formatting elements with their string counterparts and returns the result. Does not accept epoch formatted dates as input.	`to_date (date_sold, '%Y-%m-%d')`
`to_double`	Returns the input as a `double` data type.	`to_double ('3.14') = 3.14` `to_double (revenue * .01)`
`to_integer`	Returns the input as an integer.	`to_integer ('45') + 1 = 46` `to_integer (price + tax - cost)`

`to_string`	Returns the input as a text string. To convert a date data type to a string data type, specify the date format you want to use.	`to_string (45 + 1) = '46'`
		`to_string (revenue - cost)`
		`to_string (date,('%m/%d/%y'))`

Date functions

Function	Description	Examples
`add_days`	Returns the result of adding the specified number of days to the given date.	`add_days (01/30/2015, 5) = 02/04/2015` `add_days (invoiced, 30)`
`add_minutes`	Returns the result of adding the specified number of minutes to input date/datetime/time.	`add_minutes ( 01/30/2015 00:10:20 , 5 ) = 01/30/2015 00:11:20` `add_minutes ( invoiced , 30 )`

Function	Description	Examples
`add_months`	Returns the result of adding the specified number of months to the given date.	`add_months ( 01/30/2015, 5 ) = 06/30/2015` `add_months ( invoiced_date , 5 )`
`add_seconds`	Returns the result of adding the specified number of seconds to the given date.	`add_seconds ( 01/30/2015 00:00:00, 5 ) = 06/30/2015 00:00:05` `add_seconds ( invoiced_date , 5 )`
`add_weeks`	Returns the result of adding the specified number of weeks to the given date.	`add_weeks ( 01/30/2015, 2 ) = 02/13/2015` `add_weeks ( invoiced_date , 2 )`
`add_years`	Returns the result of adding the specified number of years to the given date.	`add_years ( 01/30/2015, 5 ) = 01/30/2020` `add_years ( invoiced_date , 5 )`
`date`	Returns the date portion of a given date.	`date (home visit)`
`day`	Returns the number (1-31) of the day for the given date.	`day (01/15/2014) = 15` `day (date ordered)`
`day_number_of_quarter`	Returns the number of the day in a quarter for a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`day_number_of_quarter (01/30/2015) = 30` `day_number_of_quarter (01/30/2015, 'fiscal') = 91`
`day_number_of_week`	Returns the number (1-7) of the day in a week for a given date with 1 being Monday and 7 being Sunday.	`day_number_of_week(01/15/2014) = 3` `day_number_of_week (shipped)`

Function	Description	Examples

`day_number_of_year`	Returns the number (1-366) of the day in a year from a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`day_number_of_year (01/30/2015) = 30`
		`day_number_of_year ( 01/30/2015, 'fiscal' ) = 275`
		`day_number_of_year (invoiced)`

`day_of_week`	Returns the day of the week for the given date.	`day_of week (01/30/2015) = Friday` `day_of_week (serviced)`
`diff_days`	Subtracts the second date from the first date and returns the result in number of days, rounded down if not exact.	`diff_days (01/15/2014, 01/17/2014) = -2` `diff_days (purchased, shipped)`
`diff_time`	Subtracts the second date from the first date and returns the result in number of seconds.	`diff_time (01/01/2014, 01/01/2014) = -86,400` `diff_time (clicked, submitted)`
`hour_of_day`	Returns the hour of the day for the given date.	`hour_of_day (received)`
`is_weekend`	Returns true if the given date falls on a Saturday or Sunday.	`is_weekend (01/31/2015) = true` `is_weekend (emailed)`
`month`	Returns the month from the given date.	`month (01/15/2014) = January` `month (date ordered)`

Function	Description	Examples

`month_number`	Returns the number (1-12) of the month from a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`month_number (09/20/2014) = 9`
		`month_number ( 09/20/2014, 'fiscal' ) = 5`
		`month_number (purchased)`

`month_number_of_quarter`	Returns the month (1-3) number for the given date in a quarter. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`month_number_of_quarter (02/20/2018) = 2` `month_number_of_quarter (02/20/2018,'fiscal' ) = 1`
`now`	Returns the current timestamp.	`now ()`

`quarter_number`	Returns the number (1-4) of the quarter associated with the given date. Add an optional second parameter to specify 'fiscal' or 'calendar' dates. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`quarter_number ( 04/14/2014) = 2`
		`quarter_number ( 04/14/2014, 'fiscal' ) = 4`
		`quarter_number ( shipped )`

Function	Description	Examples
`start_of_month`	Returns `MMM yyyy` for the first day of the month. Your installation configuration can override this setting so that it returns a different format such as `MM/dd/yyyy` . Speak with your ThoughtSpot administrator for information on doing this.	`start_of_month ( 01/31/2015 ) = Jan FY 2015` `start_of_month (shipped)`

`start_of_quarter`	Returns the date for the first day of the quarter for the given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`start_of_quarter ( 04/01/2014) = Apr 2014`
		`start_of_quarter ( 04/01/2014, 'fiscal') = Feb 2014`
		`start_of_quarter (sold)`

`start_of_week`	Returns the date for the first day of the week for the given date.	`start_of_week ( 06/01/2015 ) = 05/30/2015 Week` `start_of_week (emailed)`

`start_of_year`	Returns the date for the first day of the year for the given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`start_of_year (04/01/2014) returns Jan 2014`
		`start_of_year (04/01/2014, 'fiscal') returns May 2013`
		`start_of_year (joined)`

`time`	Returns the time portion of a given date.	`time (3/1/2002 10:32) = 10:32` `time (call began)`

Function	Description	Examples
`week_number_of_month`	Returns the week number for the given date in a month.	`week_number_of_month(03/23/2017) = 3`
`week_number_of_quarter`	Returns the week number for the given date in a quarter. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`week_number_of_quarter (04/03/2017) = 1` `week_number_of_quarter (04/03/2017, 'fiscal') = 10`
`week_number_of_year`	Returns the week number for the given date in a year. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`week_number_of_year (01/17/2014) = 3` `week_number_of_year ( 01/17/2014, 'fiscal') = 38`

`year`	Returns the year from a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01. Per standard convention, the fiscal year is defined by the year-end date.)	`year (01/15/2014) = 2014`
		`year (12/15/2013, 'fiscal' ) = 2014`
		`year (date ordered)`

Mixed functions

These functions can be used with text and numeric data types.

Function	Description	Examples
`!=`	Returns true if the first value is not equal to the second value.	`3 != 2 = true` `revenue != 1000000`
`<`	Returns true if the first value is less than the second value.	`3 < 2 = false` `revenue < 1000000`
`<=`	Returns true if the first value is less than or equal to the second value.	`1 <= 2 = true` `revenue <= 1000000`
`=`	Returns true if the first value is equal to the second value.	`2 = 2 = true` `revenue = 1000000`
`>`	Returns true if the first value is greater than the second value.	`3 > 2 = true` `revenue > 1000000`
`>=`	Returns true if the first value is greater than or equal to the second value.	`3 >= 2 = true` `revenue >= 1000000`
`greatest`	Returns the larger of the values.	`greatest (20, 10) = 20` `greatest (q1 revenue, q2 revenue)`
`least`	Returns the smaller of the values.	`least (20, 10) = 10` `least (q1 revenue, q2 revenue)`

Number functions

Function	Description	Examples
`*`	Returns the result of multiplying both numbers.	`3 * 2 = 6` `price * taxrate`
`+`	Returns the result of adding both numbers.	`1 + 2 = 3` `price + shipping`

Function	Description	Examples
`-`	Returns the result of subtracting the second number from the first.	`3 - 2 = 1` `revenue - tax`
`/`	Returns the result of dividing the first number by the second.	`6 / 3 = 2` `markup / retail price`
`^`	Returns the first number raised to the power of the second.	`3 ^ 2 = 9` `width ^ 2`
`abs`	Returns the absolute value.	`abs (-10) = 10` `abs (profit)`
`acos`	Returns the inverse cosine in degrees.	`acos (0.5) = 60` `acos (cos-satellite-angle)`
`asin`	Returns the inverse sine (specified in degrees).	`asin (0.5) = 30` `asin (sin-satellite-angle)`
`atan`	Returns the inverse tangent in degrees.	`atan (1) = 45` `atan (tan-satellite-angle)`
`atan2`	Returns the inverse tangent in degrees.	`atan2 (10, 10) = 45` `atan2 (longitude, latitude)`
`cbrt`	Returns the cube root of a number.	`cbrt (27) = 3` `cbrt (volume)`
`ceil`	Returns the smallest following integer.	`ceil (5.9) = 6` `ceil (growth rate)`
`cos`	Returns the cosine of an angle (specified in degrees).	`cos (63) = 0.45` `cos (beam angle)`
`cube`	Returns the cube of a number.	`cube (3) = 27` `cube (length)`
`exp`	Returns Euler's number (~2.718) raised to a power.	`exp (2) = 7.38905609893` `exp (growth)`
`exp2`	Returns 2 raised to a power.	`exp2 (3) = 8` `exp2 (growth)`
`floor`	Returns the largest previous integer.	`floor (5.1) = 5` `floor (growth rate)`
`ln`	Returns the natural logarithm.	`ln (7.38905609893) = 2` `ln (distance)`

Function	Description	Examples
`log10`	Returns the logarithm with base 10.	`log10 (100) = 2` `log10 (volume)`
`log2`	Returns the logarithm with base 2 (binary logarithm).	`log2 (32) = 5` `log2 (volume)`
`mod`	Returns the remainder of first number divided by the second number.	`mod (8, 3) = 2` `mod ( revenue , quantity )`
`pow`	Returns the first number raised to the power of the second number.	`pow (5, 2) = 25` `pow (width, 2)`
`random`	Returns a random number between 0 and 1.	`random ( ) = .457718` `random ( )`

`round`	Returns the first number rounded to the second number (the default is 1).	`round (35.65, 10) = 40`
		`round (battingavg, 100)`
		`round (48.67, .1) = 48.7`

`safe_divide`	Returns the result of dividing the first number by the second. If the second number is 0, returns 0 instead of NaN (not a number).	`safe_divide (12, 0) = 0` `safe_divide (total_cost, units)`
`sign`	Returns +1 if the number is greater than zero, -1 if less than zero, 0 if zero.	`sign (-250) = -1` `sign (growth rate)`
`sin`	Returns the sine of an angle (specified in degrees).	`sin (35) = 0.57` `sin (beam angle)`
`spherical_distance`	Returns the distance in km between two points on Earth.	`spherical_distance (37.465191, -122.153617, 37.421962, -122.142174) = 4,961.96` `spherical_distance (start_latitude, start_longitude, start_latitude, start_longitude)`
`sq`	Returns the square of a numeric value.	`sq (9) = 81` `sq (width)`
`sqrt`	Returns the square root.	`sqrt (9) = 3` `sqrt (area)`
`tan`	Returns the tangent of an angle (specified in degrees).	`tan (35) = 0.7` `tan (beam angle)`

Operators

Operator	Description	Examples

`and`	Returns `true` when both conditions are `true`, otherwise returns `false`.	`(1 = 1) and (3 > 2) = true`
		`lastname = 'smith' and state ='texas'`
		Note: Not available for row-level security (RLS) formulas.

`if…then…else`	Conditional operator	`if (3 > 2) then 'bigger' else 'not bigger'` `if (cost > 500) then 'flag' else 'approve'`
`ifnull`	Returns the first value if it is not `null`, otherwise returns the second value.	`ifnull (cost, 'unknown')`
`isnull`	Returns `true` if the value is `null`.	`isnull (phone)`
`not`	Returns `true` if the condition is `false`, otherwise returns `false`.	`not (3 > 2) = false` `not (state = 'texas')`
`or`	Returns `true` when either condition is `true`, otherwise returns `false`.	`(1 = 5) or (3 > 2) = true` `state = 'california' or state ='oregon'`

Text functions

Function	Description	Examples
`concat`	Returns two or more values as a concatenated text string. Use single quotes around each literal string, not double quotes.	`concat ( 'hay' , 'stack' ) = 'haystack'` `concat (title, ' ', first_name , ' ', last_name)`

Function	Description	Examples
`contains`	Returns true if the first string contains the second string, otherwise returns false.	`contains ('broomstick', 'room') = true` `contains (product, 'trial version')`
`edit_distance`	Accepts two text strings. Returns the edit distance (minimum number of operations required to transform one string into the other) as an integer. Works with strings under 1023 characters.	`edit_distance ('attorney', 'atty') = 4` `edit_distance (color, 'red')`
`edit_distance_with_cap`	Accepts two text strings and an integer to specify the upper limit cap for the edit distance (minimum number of operations required to transform one string into the other). If the edit distance is less than or equal to the specified cap, returns the edit distance. If it is higher than the cap, returns the cap plus 1. Works with strings under 1023 characters.	`edit_distance_with_cap ('pokemon go', 'minecraft pixelmon', 3) = 4` `edit_distance_with_cap (event, 'burning man', 3)`
`similar_to`	Accepts a document text string and a search text string. Returns true if relevance score (0-100) of the search string with respect to the document is greater than or equal to 20. Relevance is based on edit distance, number of words in the query, and length of words in the query which are present in the document.	`similar_to ('hello world', 'hello swirl') = true` `similar_to (current team, drafted by)`
`similarity`	Accepts a document text string and a search text string. Returns the relevance score (0-100) of the search string with respect to the document. Relevance is based on edit distance, number of words in the query, and length of words in the query which are present in the document. If the two strings are an exact match, returns 100.	`similarity ('where is the burning man concert', 'burning man') = 46` `similarity (tweet1, tweet2)`

Function	Description	Examples
`spells_like`	Accepts two text strings. Returns true if they are spelled similarly and false if they are not. Works with strings under 1023 characters.	`spells_like ('thouhgtspot', 'thoughtspot') = true` `spells_like (studio, distributor)`
`strlen`	Returns the length of the text.	`strlen ('smith') = 5` `strlen (lastname)`
`strpos`	Returns the numeric position (starting from 0) of the first occurrence of the second string in the first string, or -1 if not found.	`strpos ('haystack_with_needles', 'needle') = 14` `strpos (complaint, 'lawyer')`
`substr`	Returns the portion of the given string, beginning at the location specified (starting from 0), and of the given length.	`substr ('persnickety', 3, 7) = snicket` `substr (lastname, 0, 5)`

Variables

These variables can be used in your expressions.

Function	Description	Examples
`ts_groups`	Returns a list of all the groups the current logged in user belongs to. For any row, if the expression evaluates to true for any of the groups, the user can see that row.	`ts_groups = 'east'`
`ts_username`	Returns the user with the matching neame.	`ts_username != 'mark'`

Formula function reference

ThoughtSpot allows you to create derived columns in worksheets using formulas. You create these columns by building formulas using the **Formula Assistant**. An individual formula is constructed from *n* combination of operators and functions.

This reference lists the various operators and functions you can use to create formulas.

Operators

Operator	Description	Examples

`and`	Returns `true` when both conditions are `true`, otherwise returns `false`.	`(1 = 1) and (3 > 2) = true`
		`lastname = 'smith' and state ='texas'`
		Note: Not available for row-level security (RLS) formulas.

`if…then…else`	Conditional operator	`if (3 > 2) then 'bigger' else 'not bigger'` `if (cost > 500) then 'flag' else 'approve'`
`ifnull`	Returns the first value if it is not `null`, otherwise returns the second value.	`ifnull (cost, 'unknown')`
`isnull`	Returns `true` if the value is `null`.	`isnull (phone)`
`not`	Returns `true` if the condition is `false`, otherwise returns `false`.	`not (3 > 2) = false` `not (state = 'texas')`
`or`	Returns `true` when either condition is `true`, otherwise returns `false`.	`(1 = 5) or (3 > 2) = true` `state = 'california' or state ='oregon'`

Aggregate functions (group aggregate)

Use the following functions to aggregate data.

Function	Description	Examples
`average`	Returns the average of all the values of a column.	`average (revenue)`
`average_if`	Returns the average of all the columns that meet a given criteria.	`average_if(city = "San Francisco", revenue)`
`count`	Returns the number of rows in the table containing the column.	`count (product)`
`count_if`	Returns the number of rows in the table containing the column.	`count_if(region ='west', region)`
`cumulative_average`	Takes a measure and one or more attributes. Returns the average of the measure, accumulated by the attribute(s) in the order specified.	`cumulative_average (revenue, order date, state)`
`cumulative_max`	Takes a measure and one or more attributes. Returns the maximum of the measure, accumulated by the attribute(s) in the order specified.	`cumulative_max (revenue, state)`
`cumulative_min`	Takes a measure and one or more attributes. Returns the minimum of the measure, accumulated by the attribute(s) in the order specified.	`cumulative_min (revenue, campaign)`
`cumulative_sum`	Takes a measure and one or more attributes. Returns the sum of the measure, accumulated by the attribute(s) in the order specified.	`cumulative_sum (revenue, order date)`

Function	Description	Examples

`group_aggregate`	Takes a measure and optional attributes and filters. Used to aggregate measures with different granularities and filters than the columns used in the search. Commonly used in comparison analysis.
	
	This formula takes the following form:
	`group_aggregate (<aggregation(measure)>, <groupings>, <filters>)`
	
	Define lists using curly brackets, `{ }`. Optional list functions `query_groups` or `query_filters` specify the lists or filters used in the original search. Use `+` (plus) and `-` (minus) to add or exclude specific columns for query groups.
	
	See Flexible aggregation functions [See page 0].
		`group_aggregate (sum (revenue), {ship mode, date}, {} )`
		
		`group_aggregate (sum (revenue), {ship mode , date}, {day_of_week (date) = 'friday'} )`
		
		`group_aggregate (sum (revenue), query_groups(), query_filters() )`
		
		`group_aggregate (sum (revenue), query_groups() + {date}, query_filters() )`

`group_average`	Takes a measure and one or more attributes. Returns the average of the measure grouped by the attribute(s).	`group_average (revenue, customer region, state)`
`group_count`	Takes a measure and one or more attributes. Returns the count of the measure grouped by the attribute(s).	`group_count (revenue, customer region)`
`group_max`	Takes a measure and one or more attributes. Returns the maximum of the measure grouped by the attribute(s).	`group_max (revenue, customer region)`
`group_min`	Takes a measure and one or more attributes. Returns the minimum of the measure grouped by the attribute(s).	`group_min (revenue, customer region)`

Function	Description	Examples
`group_stddev`	Takes a measure and one or more attributes. Returns the standard deviation of the measure grouped by the attribute(s).	`group_stddev (revenue, customer region)`
`group_sum`	Takes a measure and one or more attributes. Returns the sum of the measure grouped by the attribute(s).	`group_sum (revenue, customer region)`
`group_unique_count`	Takes a measure and one or more attributes. Returns the unique count of the measure grouped by the attribute(s).	`group_unique_count (product , supplier)`
`group_variance`	Takes a measure and one or more attributes. Returns the variance of the measure grouped by the attribute(s).	`group_variance (revenue, customer region)`
`max`	Returns the maximum value of a column.	`max (sales)`
`max_if`	Returns the maximum value among columns that meet a criteria.	`max_if( (revenue > 10) , customer region )`
`min`	Returns the minimum value of a column.	`min (revenue)`
`min_if`	Returns the minimum value among columns that meet a criteria.	`min_if( (revenue < 10) , customer region )`

Function	Description	Examples
`moving_average`	Takes a measure, two integers to define the window to aggregate over, and one or more attributes. The window is (current - Num1…Current + Num2) with both end points being included in the window. For example, “1,1” will have a window size of 3. To define a window that begins before Current, specify a negative number for Num2. Returns the average of the measure over the given window. The attributes are the ordering columns used to compute the moving average.	`moving_average (revenue, 2, 1, customer region)`
`moving_max`	Takes a measure, two integers to define the window to aggregate over, and one or more attributes. The window is (current - Num1…Current + Num2) with both end points being included in the window. For example, “1,1” will have a window size of 3. To define a window that begins before Current, specify a negative number for Num2. Returns the maximum of the measure over the given window. The attributes are the ordering columns used to compute the moving maximum.	`moving_max (complaints, 1, 2, store name)`

Function	Description	Examples
`moving_min`	Takes a measure, two integers to define the window to aggregate over, and one or more attributes. The window is (current - Num1…Current + Num2) with both end points being included in the window. For example, “1,1” will have a window size of 3. To define a window that begins before Current, specify a negative number for Num2. Returns the minimum of the measure over the given window. The attributes are the ordering columns used to compute the moving minimum.	`moving_min (defects, 3, 1, product)`
`moving_sum`	Takes a measure, two integers to define the window to aggregate over, and one or more attributes. The window is (current - Num1…Current + Num2) with both end points being included in the window. For example, “1,1” will have a window size of 3. To define a window that begins before Current, specify a negative number for Num2. Returns the sum of the measure over the given window. The attributes are the ordering columns used to compute the moving sum.	`moving_sum (revenue, 1, 1, order date)`
`rank`	Returns the rank for the current row. Identical values receive an identical rank. Takes an aggregate input for the first argument. The second argument specifies the order, `'asc' | 'desc'`.	`rank (sum (revenue) , 'asc' )` `rank (sum (revenue) , ‘desc' )`
`rank_percentile`	Returns the percentile rank for the current row. Identical values are assigned an identical percentile rank. Takes an aggregate input for the first argument. The second argument specifies the order, `'asc' | 'desc'`.	`rank_percentile (sum (revenue) , 'asc' )` `rank_percentile (sum (revenue) , 'desc' )`

Function	Description	Examples
`stddev`	Returns the standard deviation of all values of a column.	`stddev (revenue)`
`stddev_if`	Returns a standard deviation values filtered to meet a specific criteria.	`stddev_if( (revenue > 10) , (revenue/10.0) )`
`sum`	Returns the sum of all the values of a column.	`sum (revenue)`
`sum_if`	Returns sum values filtered by a specific criteria.	`sum_if(region='west', revenue)`
`unique count`	Returns the number of unique values of a column.	`unique count (customer)`
`unique_count_if`	Returns the number of unique values of a column provided it meets a criteria.	`unique_count_if( (revenue > 10) , order date )`
`variance`	Returns the variance of all the values of a column.	`variance (revenue)`
`variance_if`	Returns the variance of all the values of a column provided it meets a criteria..	`variance_if( (revenue > 10) , (revenue/10.0) )`

Conversion functions

Use these functions to convert data from one data type into another data type.

ThoughtSpot does not support *date* data type conversion.

Function	Description	Examples
`to_bool`	Returns the input as a `boolean` data type (`true` or `false`).	`to_bool (0) = false` `to_bool (married)`

Function	Description	Examples
`to_date`	Accepts a date represented as an integer or text string, and a second string parameter that can include `strptime` date formatting elements. Replaces all the valid strptime date formatting elements with their string counterparts and returns the result. Does not accept epoch formatted dates as input.	`to_date (date_sold, '%Y-%m-%d')`
`to_double`	Returns the input as a `double` data type.	`to_double ('3.14') = 3.14` `to_double (revenue * .01)`
`to_integer`	Returns the input as an integer.	`to_integer ('45') + 1 = 46` `to_integer (price + tax - cost)`

`to_string`	Returns the input as a text string. To convert a date data type to a string data type, specify the date format you want to use.	`to_string (45 + 1) = '46'`
		`to_string (revenue - cost)`
		`to_string (date,('%m/%d/%y'))`

Date functions

Function	Description	Examples
`add_days`	Returns the result of adding the specified number of days to the given date.	`add_days (01/30/2015, 5) = 02/04/2015` `add_days (invoiced, 30)`
`add_minutes`	Returns the result of adding the specified number of minutes to input date/datetime/time.	`add_minutes ( 01/30/2015 00:10:20 , 5 ) = 01/30/2015 00:11:20` `add_minutes ( invoiced , 30 )`

Function	Description	Examples
add_months	Returns the result of adding the specified number of months to the given date.	add_months (01/30/2015, 5) = 06/30/2015 add_months (invoiced_date , 5)
add_seconds	Returns the result of adding the specified number of seconds to the given date.	add_seconds (01/30/2015 00:00:00, 5) = 06/30/2015 00:00:05 add_seconds (invoiced_date , 5)
add_weeks	Returns the result of adding the specified number of weeks to the given date.	add_weeks (01/30/2015, 2) = 02/13/2015 add_weeks (invoiced_date , 2)
add_years	Returns the result of adding the specified number of years to the given date.	add_years (01/30/2015, 5) = 01/30/2020 add_years (invoiced_date , 5)
date	Returns the date portion of a given date.	date (home visit)
day	Returns the number (1-31) of the day for the given date.	day (01/15/2014) = 15 day (date ordered)
day_number_of_quarter	Returns the number of the day in a quarter for a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	day_number_of_quarter (01/30/2015) = 30 day_number_of_quarter (01/30/2015, 'fiscal') = 91
day_number_of_week	Returns the number (1-7) of the day in a week for a given date with 1 being Monday and 7 being Sunday.	day_number_of_week(01/15/2014) = 3 day_number_of_week (shipped)

Function	Description	Examples
day_number_of_year	Returns the number (1-366) of the day in a year from a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	day_number_of_year (01/30/2015) = 30 day_number_of_year (01/30/2015, 'fiscal') = 275 day_number_of_year (invoiced)
day_of_week	Returns the day of the week for the given date.	day_of week (01/30/2015) = Friday day_of_week (serviced)
diff_days	Subtracts the second date from the first date and returns the result in number of days, rounded down if not exact.	diff_days (01/15/2014, 01/17/2014) = -2 diff_days (purchased, shipped)
diff_time	Subtracts the second date from the first date and returns the result in number of seconds.	diff_time (01/01/2014, 01/01/2014) = -86,400 diff_time (clicked, submitted)
hour_of_day	Returns the hour of the day for the given date.	hour_of_day (received)
is_weekend	Returns true if the given date falls on a Saturday or Sunday.	is_weekend (01/31/2015) = true is_weekend (emailed)
month	Returns the month from the given date.	month (01/15/2014) = January month (date ordered)

Function	Description	Examples

`month_number`	Returns the number (1-12) of the month from a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`month_number (09/20/2014) = 9`
		`month_number ( 09/20/2014, 'fiscal' ) = 5`
		`month_number (purchased)`

`month_number_of_quarter`	Returns the month (1-3) number for the given date in a quarter. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`month_number_of_quarter (02/20/2018) = 2` `month_number_of_quarter (02/20/2018,'fiscal' ) = 1`
`now`	Returns the current timestamp.	`now ()`

`quarter_number`	Returns the number (1-4) of the quarter associated with the given date. Add an optional second parameter to specify 'fiscal' or 'calendar' dates. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`quarter_number ( 04/14/2014) = 2`
		`quarter_number ( 04/14/2014, 'fiscal' ) = 4`
		`quarter_number ( shipped )`

Function	Description	Examples
`start_of_month`	Returns `MMM yyyy` for the first day of the month. Your installation configuration can override this setting so that it returns a different format such as `MM/dd/yyyy` . Speak with your ThoughtSpot administrator for information on doing this.	`start_of_month ( 01/31/2015 ) = Jan FY 2015` `start_of_month (shipped)`

`start_of_quarter`	Returns the date for the first day of the quarter for the given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`start_of_quarter ( 04/01/2014) = Apr 2014`
		`start_of_quarter ( 04/01/2014, 'fiscal') = Feb 2014`
		`start_of_quarter (sold)`

`start_of_week`	Returns the date for the first day of the week for the given date.	`start_of_week ( 06/01/2015 ) = 05/30/2015 Week` `start_of_week (emailed)`

`start_of_year`	Returns the date for the first day of the year for the given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`start_of_year (04/01/2014) returns Jan 2014`
		`start_of_year (04/01/2014, 'fiscal') returns May 2013`
		`start_of_year (joined)`

`time`	Returns the time portion of a given date.	`time (3/1/2002 10:32) = 10:32` `time (call began)`

Function	Description	Examples
`week_number_of_month`	Returns the week number for the given date in a month.	`week_number_of_month(03/23/2017) = 3`
`week_number_of_quarter`	Returns the week number for the given date in a quarter. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`week_number_of_quarter (04/03/2017) = 1` `week_number_of_quarter (04/03/2017, 'fiscal') = 10`
`week_number_of_year`	Returns the week number for the given date in a year. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`week_number_of_year (01/17/2014) = 3` `week_number_of_year ( 01/17/2014, 'fiscal') = 38`

`year`	Returns the year from a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01. Per standard convention, the fiscal year is defined by the year-end date.)	`year (01/15/2014) = 2014`
		`year (12/15/2013, 'fiscal' ) = 2014`
		`year (date ordered)`

Mixed functions

These functions can be used with text and numeric data types.

Function	Description	Examples
`!=`	Returns true if the first value is not equal to the second value.	`3 != 2 = true` `revenue != 1000000`
`<`	Returns true if the first value is less than the second value.	`3 < 2 = false` `revenue < 1000000`
`<=`	Returns true if the first value is less than or equal to the second value.	`1 <= 2 = true` `revenue <= 1000000`
`=`	Returns true if the first value is equal to the second value.	`2 = 2 = true` `revenue = 1000000`
`>`	Returns true if the first value is greater than the second value.	`3 > 2 = true` `revenue > 1000000`
`>=`	Returns true if the first value is greater than or equal to the second value.	`3 >= 2 = true` `revenue >= 1000000`
`greatest`	Returns the larger of the values.	`greatest (20, 10) = 20` `greatest (q1 revenue, q2 revenue)`
`least`	Returns the smaller of the values.	`least (20, 10) = 10` `least (q1 revenue, q2 revenue)`

Number functions

Function	Description	Examples
`*`	Returns the result of multiplying both numbers.	`3 * 2 = 6` `price * taxrate`
`+`	Returns the result of adding both numbers.	`1 + 2 = 3` `price + shipping`

Function	Description	Examples
`-`	Returns the result of subtracting the second number from the first.	`3 - 2 = 1` `revenue - tax`
`/`	Returns the result of dividing the first number by the second.	`6 / 3 = 2` `markup / retail price`
`^`	Returns the first number raised to the power of the second.	`3 ^ 2 = 9` `width ^ 2`
`abs`	Returns the absolute value.	`abs (-10) = 10` `abs (profit)`
`acos`	Returns the inverse cosine in degrees.	`acos (0.5) = 60` `acos (cos-satellite-angle)`
`asin`	Returns the inverse sine (specified in degrees).	`asin (0.5) = 30` `asin (sin-satellite-angle)`
`atan`	Returns the inverse tangent in degrees.	`atan (1) = 45` `atan (tan-satellite-angle)`
`atan2`	Returns the inverse tangent in degrees.	`atan2 (10, 10) = 45` `atan2 (longitude, latitude)`
`cbrt`	Returns the cube root of a number.	`cbrt (27) = 3` `cbrt (volume)`
`ceil`	Returns the smallest following integer.	`ceil (5.9) = 6` `ceil (growth rate)`
`cos`	Returns the cosine of an angle (specified in degrees).	`cos (63) = 0.45` `cos (beam angle)`
`cube`	Returns the cube of a number.	`cube (3) = 27` `cube (length)`
`exp`	Returns Euler's number (~2.718) raised to a power.	`exp (2) = 7.38905609893` `exp (growth)`
`exp2`	Returns 2 raised to a power.	`exp2 (3) = 8` `exp2 (growth)`
`floor`	Returns the largest previous integer.	`floor (5.1) = 5` `floor (growth rate)`
`ln`	Returns the natural logarithm.	`ln (7.38905609893) = 2` `ln (distance)`

Function	Description	Examples
`log10`	Returns the logarithm with base 10.	`log10 (100) = 2` `log10 (volume)`
`log2`	Returns the logarithm with base 2 (binary logarithm).	`log2 (32) = 5` `log2 (volume)`
`mod`	Returns the remainder of first number divided by the second number.	`mod (8, 3) = 2` `mod ( revenue , quantity )`
`pow`	Returns the first number raised to the power of the second number.	`pow (5, 2) = 25` `pow (width, 2)`
`random`	Returns a random number between 0 and 1.	`random ( ) = .457718` `random ( )`

`round`	Returns the first number rounded to the second number (the default is 1).	`round (35.65, 10) = 40`
		`round (battingavg, 100)`
		`round (48.67, .1) = 48.7`

`safe_divide`	Returns the result of dividing the first number by the second. If the second number is 0, returns 0 instead of NaN (not a number).	`safe_divide (12, 0) = 0` `safe_divide (total_cost, units)`
`sign`	Returns +1 if the number is greater than zero, -1 if less than zero, 0 if zero.	`sign (-250) = -1` `sign (growth rate)`
`sin`	Returns the sine of an angle (specified in degrees).	`sin (35) = 0.57` `sin (beam angle)`
`spherical_distance`	Returns the distance in km between two points on Earth.	`spherical_distance (37.465191, -122.153617, 37.421962, -122.142174) = 4,961.96` `spherical_distance (start_latitude, start_longitude, start_latitude, start_longitude)`
`sq`	Returns the square of a numeric value.	`sq (9) = 81` `sq (width)`
`sqrt`	Returns the square root.	`sqrt (9) = 3` `sqrt (area)`
`tan`	Returns the tangent of an angle (specified in degrees).	`tan (35) = 0.7` `tan (beam angle)`

Text functions

Function	Description	Examples
`concat`	Returns two or more values as a concatenated text string. Use single quotes around each literal string, not double quotes.	`concat ( 'hay' , 'stack' ) = 'haystack'` `concat (title, ' ', first_name , ' ', last_name)`
`contains`	Returns true if the first string contains the second string, otherwise returns false.	`contains ('broomstick', 'room') = true` `contains (product, 'trial version')`
`edit_distance`	Accepts two text strings. Returns the edit distance (minimum number of operations required to transform one string into the other) as an integer. Works with strings under 1023 characters.	`edit_distance ('attorney', 'atty') = 4` `edit_distance (color, 'red')`
`edit_distance_with_cap`	Accepts two text strings and an integer to specify the upper limit cap for the edit distance (minimum number of operations required to transform one string into the other). If the edit distance is less than or equal to the specified cap, returns the edit distance. If it is higher than the cap, returns the cap plus 1. Works with strings under 1023 characters.	`edit_distance_with_cap ('pokemon go', 'minecraft pixelmon', 3) = 4` `edit_distance_with_cap (event, 'burning man', 3)`
`similar_to`	Accepts a document text string and a search text string. Returns true if relevance score (0-100) of the search string with respect to the document is greater than or equal to 20. Relevance is based on edit distance, number of words in the query, and length of words in the query which are present in the document.	`similar_to ('hello world', 'hello swirl') = true` `similar_to (current team, drafted by)`

Function	Description	Examples
`similarity`	Accepts a document text string and a search text string. Returns the relevance score (0-100) of the search string with respect to the document. Relevance is based on edit distance, number of words in the query, and length of words in the query which are present in the document. If the two strings are an exact match, returns 100.	`similarity ('where is the burning man concert', 'burning man') = 46` `similarity (tweet1, tweet2)`
`spells_like`	Accepts two text strings. Returns true if they are spelled similarly and false if they are not. Works with strings under 1023 characters.	`spells_like ('thouhgtspot', 'thoughtspot') = true` `spells_like (studio, distributor)`
`strlen`	Returns the length of the text.	`strlen ('smith') = 5` `strlen (lastname)`
`strpos`	Returns the numeric position (starting from 0) of the first occurrence of the second string in the first string, or -1 if not found.	`strpos ('haystack_with_needles', 'needle') = 14` `strpos (complaint, 'lawyer')`
`substr`	Returns the portion of the given string, beginning at the location specified (starting from 0), and of the given length.	`substr ('persnickety', 3, 7) = snicket` `substr (lastname, 0, 5)`

Alerts code reference

This reference identifies the messages that can appear in the **System Health > Overview > Critical Alerts** and in the **Alerts** dashboard.

Informational alerts

TASK_TERMINATED

Msg: `Task {{.Service}}.{{.Task}} terminated on machine {{.Machine}}`

Type: INFO

This alert is raised when a task terminates.

DISK_ERROR

Msg: `Machine {{.Machine}} has disk errors`

Type: INFO

Raised when a machine has disk errors.

ZK_AVG_LATENCY

Msg: `Average Zookeeper latency is more than {{.Num}} msec`

Type: INFO

Raised when average Zookeeper latency is above a threshold.

ZK_MAX_LATENCY

Msg: `Max Zookeeper latency is more than {{.Num}} msec`

Type: INFO

Raised when max Zookeeper latency is above a threshold.

ZK_MIN_LATENCY

Msg: `Min Zookeeper latency is more than {{.Num}} msec`

Type: INFO

Raised when min Zookeeper latency is above a threshold.

ZK_OUTSTANDING_REQUESTS

Msg: `Number of outstanding Zookeeper requests exceeds {{.Num}}`

Type: INFO

Raised when there are too many outstanding Zookeeper requests.

ZK_NUM_WATCHERS

Msg: `Number of Zookeeper watchers exceeds {{.Num}}`

Type: INFO

Raised when there are too many Zookeeper watchers.

MASTER_ELECTION

Msg: `{{.Machine}} elected as Orion Master`

Type: INFO

Raised when a new Orion Master is elected.

PERIODIC_BACKUP

Msg: `{{.Process}} periodic backup for policy {{.Name}} failed.`

Type: INFO

Raised when periodic backup fails.

PERIODIC_SNAPSHOT

Msg: `{{.Process}} periodic snapshot {{.Name}} failed.`

Type: INFO

Raised when a periodic snapshot fails.

HDFS_CORRUPTION

Msg: `HDFS root directory is in a corrupted state.`

Type: INFO

Raised when HDFS root directory is corrupted.

APPLICATION_INVALID_STATE

Msg: `{{.Service}}.{{.Task}} on {{.Machine}} at location {{.Location}}`

Type: INFO

Raised when Application raises invalid state alert.

UPDATE_START

Msg: `Starting update of ThoughtSpot cluster {{.Cluster}}`

Type: INFO

Raised when update starts.

UPDATE_END

Msg: `Finished update of ThoughtSpot cluster {{.Cluster}} to release {{.Release}}`

Type: INFO

Raised when update completes.

Errors

TIMELY_JOB_RUN_ERROR

Msg: `Job run {{.Message}}`

Type: ERROR

Raised when a job run fails.

TIMELY_ERROR

Msg: `Job manager {{.Message}}`

Type: ERROR

Raised when a job manager runs into an inconsistent state.

Warnings

DISK_SPACE

Msg: `Machine {{.Machine}} has less than {{.Perc}}% disk space free`

Type: WARNING

Raised when a disk is low on available disk space. Valid only in the 3.2 version of ThoughtSpot.

ROOT_DISK_SPACE

Msg: `Machine {{.Machine}} has less than {{.Perc}}% disk space free on root partition`

Type: WARNING

Raised when a machine is low on available disk space on root partition.

BOOT_DISK_SPACE

Msg: `Machine {{.Machine}} has less than {{.Perc}}% disk space free on boot partition`

Type: WARNING

Raised when a machine is low on available disk space on boot partition.

UPDATE_DISK_SPACE

Msg: `Machine {{.Machine}} has less than {{.Perc}}% disk space free on update partition`

Type: WARNING

Raised when a machine is low on available disk space on update partition.

EXPORT_DISK_SPACE

Msg: `Machine {{.Machine}} has less than {{.Perc}}% disk space free on export partition`

Type: WARNING

Raised when a machine is low on available disk space on export partition.

HDFS_NAMENODE_DISK_SPACE

Msg: `Machine {{.Machine}} has less than {{.Perc}}% disk space free on HDFS namenode drive`

Type: WARNING

Raised when a machine is low on available disk space on HDFS namenode drive.

MEMORY

Msg: `Machine {{.Machine}} has less than {{.Perc}}% memory free`

Type: WARNING

Raised when a machine is low on free memory.

OS_USERS

Msg: `Machine {{.Machine}} has more than {{.Num}} logged in users`

Type: WARNING

Raised when a machine has too many users logged in.

OS_PROCS

Msg: `Machine {{.Machine}} has more than {{.Num}} processes`

Type: WARNING

Raised when a machine has more too many processes.

SSH

Msg: `Machine {{.Machine}} doesn't have an active SSH server`

Type: WARNING

Raised when a machine has more than 600 processes.

DISK_ERROR_EXTERNAL

Msg: `Machine {{.Machine}} has disk errors`

Type: WARNING

Raised when more than 2 disk errors happen in a day.

ZK_FD_COUNT

Msg: `Zookeeper has more than {{.Num}} open file descriptors`

Type: WARNING

Raised when there are too many open Zookeeper files.

ZK_EPHEMERAL_COUNT

Msg: `Zookeeper has more than {{.Num}} ephemeral files`

Type: WARNING

Raised when there are too many Zookeeper ephemeral files.

HOST_DOWN

Msg: `{{.Machine}} is down`

Type: WARNING

Raised when a host is down.

TASK_UNREACHABLE

Msg: `{{.ServiceDesc}} on {{.Machine}} is unreachable over HTTP`

Type: WARNING

Raised when a task is unreachable over HTTP.

TASK_NOT_RUNNING

Msg: `{{.ServiceDesc}} is not running`

Type: WARNING

Raised when a service task is not running on any machine in the cluster.

Critical alerts

TASK_FLAPPING

Msg: `Task {{.Service}}.{{.Task}} terminated {{._actual_num_occurrences}} times in last {{._earliest_duration_str}}`

Type: CRITICAL

This alert is raised when a task is crashing repeatedly. The service is evaluted across the whole cluster. So, if a service crashes 5 times in a day across all nodes in the cluster, this alert is generated.

OREO_TERMINATED

Msg: `Oreo terminated on machine {{.Machine}}`

Type: CRITICAL

This alert is raised when the Oreo daemon on a machine terminates due to an error. This typically happens due to an error accessing Zookeeper, HDFS, or a hardware issue.

HDFS_DISK_SPACE

Msg: `HDFS has less than {{.Perc}}% space free`

Type: CRITICAL

Raised when a HDFS cluster is low on total available disk space.

ZK_INACCESSIBLE

Msg: `Zookeeper is not accessible`

Type: CRITICAL

Raised when Zookeeper is inaccessible.

PERIODIC_BACKUP_FLAPPING

Msg: `Periodic backup failed {{._actual_num_occurrences}} times in last {{._earliest_duration_str}}`

Type: CRITICAL

This alert is raised when a periodic backup failed repeatedly.

PERIODIC_SNAPSHOT_FLAPPING

Msg: `Periodic snapshot failed {{._actual_num_occurrences}} times in last {{._earliest_duration_str}}`

Type: CRITICAL

This alert is raised when periodic snapshot failed repeatedly.

APPLICATION_INVALID_STATE_EXTERNAL

Msg: `{{.Service}}.{{.Task}} on {{.Machine}} at location {{.Location}}`

Type: CRITICAL

Raised when Application raises invalid state alert.

User action code reference

This reference identifies the user action codes that can appear in the **System Health** pages and in logs or other reports.

`answer_unsaved`	User makes a change to tokens in the search bar.
`answer_saved`	User opens an existing saved answer and makes changes to tokens in the search bar.
`answer_pinboard_context`	User opens an existing saved pinboard, edits a context viz and makes a change to tokens in the search bar.
`answer_aggregated_worksheet`	User opens an existing saved aggregated worksheet and makes changes to tokens in the search bar.
`answer_upgrade`	Requests made for the sole purpose of upgrade.
`pinboard_view`	User opens an existing saved pinboard.
`pinboard_filter`	User adds, removes or applies values to a pinboard filter.
`pinboard_ad_hoc`	User drills down in a pinboard viz.
`data_chart_config`	Request for new data being generated following a chart config change.
`data_show_underlying_row`	Request to show underlying data for a data row(s).
`data_export`	Request to export data.
`pinboard_tspublic_runtime_filter`	Request to TSPublic/pinboarddata with runtime filters.
`answer_aggregated_worksheet_save`	User updates aggregated worksheet.
`answer_add_new_filter`	User adds a filter using the UI.
`data_show_underlying_viz`	Request to show underlying data for a data row(s).
`answer_view`	User opens an existing, saved answer.

`answer_viz_context_view`	User opens an existing saved pinboard, edits a context viz.
`pinboard_insight_view`	User opens SpotIQ tab pinboards.
`pinboard_admin_view`	User opens admin tab pinboards.
`pinboard_embed_view`	User opens embed pinboard from a URL.
`pinboard_homepage_view`	On loading of homepage pinboard.
`pinboard_learn_view`	On loading learn pinboard.
`pinboard_tspublic_no_runtime_filter`	Request to TSPublic/pinboard data without runtime filters.

Frequently asked questions

Where can I find the version of ThoughtSpot I am using?

Users with administrative privileges can see this displayed on the **Admin > System Health > Overview** page.

I'm not seeing certain columns/values in the drop-down, why?

It could be the index has not built with the latest data or something is causing the column to be dropped.

- Verify the the column is available using the **Data** page.
- View the table columns and check the **INDEX TYPE** value. If it is set to `DONT_INDEX`, change it.
- Check the column's **INDEX PRIORITY** &emdash. make sure it is `1`.

To learn more about modeling data see modeling data [See page 282] in this documentation.