# Table of Contents

- **New to Sisense** ................................................................................................................. 12
- **Minimum Requirements and Supported Platforms** .................................................. 16
- **Sisense Basic Concepts and Terminology** ................................................................. 28
- **Sisense User Roles** ........................................................................................................... 46
- **Sisense Architecture** ............................................................................................................. 51
- **Downloading and Installing Sisense** ................................................................. 58
- **Setting Up Your Production Environment** .............................................................. 67
- **Customizing the Installation** ............................................................................................. 72
- **Performing a Silent Installation** .................................................................................... 75
- **Installing Sisense Offline** ............................................................................................. 78
- **Backing Up Sisense** ......................................................................................................... 84
- **Researching Your Upgrade** ............................................................................................. 93
- **Backing Up Sisense** ......................................................................................................... 95
- **Performing an Upgrade** ................................................................................................. 103
- **Testing Your Upgrade** ..................................................................................................... 111
- **Licensing Terminology** ................................................................................................. 114
- **Launching Sisense: The Sisense Web Application** .................................................... 117
- **Tutorial: Getting Started with Sisense** ....................................................................... 119
- **Release Notes** .................................................................................................................. 144
- **Manage Data** .................................................................................................................... 145
  - **Data Models** ..................................................................................................................... 145
- **Navigating the Data Page** ............................................................................................... 149
- **ElastiCubes** ..................................................................................................................... 151
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ElastiCubes – Technical Overview</td>
<td>153</td>
</tr>
<tr>
<td>Creating ElastiCubes</td>
<td>160</td>
</tr>
<tr>
<td>Navigating the Model Editor</td>
<td>161</td>
</tr>
<tr>
<td>Introduction to Data Sources</td>
<td>163</td>
</tr>
<tr>
<td>Changing Connectivity Settings for Data Sources</td>
<td>171</td>
</tr>
<tr>
<td>Working with Data</td>
<td>174</td>
</tr>
<tr>
<td>Finding Tables and Columns</td>
<td>175</td>
</tr>
<tr>
<td>Creating and Removing a Relationship between Tables</td>
<td>177</td>
</tr>
<tr>
<td>Previewing Data from a Table</td>
<td>179</td>
</tr>
<tr>
<td>Managing Tables and Columns</td>
<td>186</td>
</tr>
<tr>
<td>Creating Data Dictionaries</td>
<td>192</td>
</tr>
<tr>
<td>Importing and Exporting ElastiCube Data</td>
<td>196</td>
</tr>
<tr>
<td>Handling Relationship Cycles</td>
<td>198</td>
</tr>
<tr>
<td>Many-to-Many Relationships</td>
<td>205</td>
</tr>
<tr>
<td>Chasm and Fan Traps</td>
<td>219</td>
</tr>
<tr>
<td>Handling Ragged and Unbalanced Data Hierarchies</td>
<td>223</td>
</tr>
<tr>
<td>Overview</td>
<td>231</td>
</tr>
<tr>
<td>Working with Build Logs</td>
<td>233</td>
</tr>
<tr>
<td>Transforming and Enriching Data</td>
<td>235</td>
</tr>
<tr>
<td>Augmented Text Deduplication</td>
<td>236</td>
</tr>
<tr>
<td>Adding a Custom Column</td>
<td>247</td>
</tr>
<tr>
<td>Editing Field Formulas</td>
<td>251</td>
</tr>
<tr>
<td>Adding a Custom Table</td>
<td>253</td>
</tr>
<tr>
<td>Editing SQL Statements</td>
<td>254</td>
</tr>
<tr>
<td>Previewing Results</td>
<td>256</td>
</tr>
<tr>
<td>SQL Reference</td>
<td>258</td>
</tr>
<tr>
<td>ElastiCube for Advanced Business Scenarios</td>
<td>264</td>
</tr>
<tr>
<td>Integrating Data</td>
<td>268</td>
</tr>
</tbody>
</table>
Column Chart ................................................................. 419
Indicator ................................................................. 424
Line Chart ................................................................. 428
Pie Chart ................................................................. 434
Pivot ........................................................................... 439
  Limitations .................................................................. 446
Polar Chart ................................................................. 448
Scatter Chart ............................................................... 451
Scatter Map .................................................................. 454
Sunburst Widget ........................................................... 460
Table ............................................................................ 463
Adding Text Widgets ....................................................... 467
Treemap ....................................................................... 469
Widget Designer ............................................................ 473
Selecting Colors in Widgets ............................................. 479
Renaming Fields in Widgets ............................................. 485
Formatting Numbers in Widgets ........................................ 485
Changing a Widget’s Visualization ...................................... 490
Combining Two Types of Visualizations in a Widget .......... 491
Creating a Continuous Chart with Missing Date Values .... 493
Adding Drill Hierarchies to Widgets ................................. 496
Managing Widgets ......................................................... 499
Field Suggestions ........................................................ 504
Introduction to Filters ..................................................... 506
Interacting with Filters as a Viewer ................................. 507
Interacting with Filters as a Designer ............................... 510
Creating Dashboard Filters ............................................. 511
Editing and Deleting a Dashboard Filter .............................. 530
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Managing Sisense Users</td>
<td>821</td>
</tr>
<tr>
<td>Managing Dashboard Users</td>
<td>822</td>
</tr>
<tr>
<td>Managing User Groups</td>
<td>829</td>
</tr>
<tr>
<td>User Sessions</td>
<td>837</td>
</tr>
<tr>
<td>Customizing User Roles</td>
<td>842</td>
</tr>
<tr>
<td>Integrating Active Directory</td>
<td>856</td>
</tr>
<tr>
<td>Account Lockout Thresholds</td>
<td>867</td>
</tr>
<tr>
<td>Introduction to SSO</td>
<td>872</td>
</tr>
<tr>
<td>SSO via JWT</td>
<td>873</td>
</tr>
<tr>
<td>SSO via SAML 2.0</td>
<td>883</td>
</tr>
<tr>
<td>Managing Your Users Dashboards</td>
<td>893</td>
</tr>
<tr>
<td>Data Management</td>
<td>896</td>
</tr>
<tr>
<td>Adding and Removing ElastiCube Servers</td>
<td>897</td>
</tr>
<tr>
<td>Sharing ElastiCube Models</td>
<td>901</td>
</tr>
<tr>
<td>Managing Drill Hierarchies</td>
<td>904</td>
</tr>
<tr>
<td>System Settings</td>
<td>907</td>
</tr>
<tr>
<td>Changing the Sisense Web Application’s Language</td>
<td>908</td>
</tr>
<tr>
<td>Internationalization and Localization</td>
<td>914</td>
</tr>
<tr>
<td>Translating Sisense Metadata</td>
<td>915</td>
</tr>
<tr>
<td>System Configuration</td>
<td>925</td>
</tr>
<tr>
<td>License Utilization</td>
<td>927</td>
</tr>
<tr>
<td>Configuration Manager</td>
<td>929</td>
</tr>
<tr>
<td>Sisense Migration</td>
<td>934</td>
</tr>
<tr>
<td>Migrating ElastiCubes</td>
<td>936</td>
</tr>
<tr>
<td>Editing .ecube Files</td>
<td>940</td>
</tr>
<tr>
<td>Building ElastiCubes with the PSM</td>
<td>940</td>
</tr>
<tr>
<td>Editing the .ecube File</td>
<td>942</td>
</tr>
<tr>
<td>Migrating Dashboards in Sisense</td>
<td>946</td>
</tr>
</tbody>
</table>
New to Sisense

This section describes how Sisense works, how to set Sisense up, and provides a tutorial for building your first dashboard from scratch.
Learn how Sisense Works

Sisense's end-to-end, single stack solution simplifies complex data in 3 short steps:

- **Model**: Blend massive amounts of data from a variety of sources into the Sisense ElastiCube.

- **Analyze**: Once your data is inside the ElastiCube, interact with it in a visual way to get deeper understanding, or zoom out to understand the holistic data landscape.

- **Impact**: Create dashboards that make it easy to get insights and take action.

The topics below describe in more detail how Sisense works.

[Sisense Basic Concepts and Terminology](#)
[Sisense Architecture Technical Overview](#)
Set Up Sisense

Before you start building ElastiCubes and designing dashboards, you need to set Sisense up, the topics below explain how:

- Downloading and Installing Sisense
- Setting Up Your Production Environment
- Customizing the Installation
- Performing a Silent Installation
- Migrating Sisense
- Upgrading Sisense
- Launching Sisense
Give Sisense a Try

Now that you've set Sisense up, it's time to play. See the tutorial below to learn how to import data and build your first dashboard.

Tutorial: Getting Started

See Minimum Requirements and Supported Platforms to see what you need to support Sisense.
Minimum Requirements and Supported Platforms

This page includes prerequisites and supported platforms that are required when working with Sisense.

Below you can find links to the section relevant for you:

- Supported Web Browsers
- Supported Operating Systems
- Hardware Requirements
- Sisense Recommendations
- Considerations
Supported Web Browsers

The Sisense Web Application runs in the following HTML5 supported browsers:
- Internet Explorer 11
- Google Chrome
- Firefox
- Safari version 7 and higher

**Note:** When embedding iFrames, Sisense supports Safari 10 and higher.

Microsoft Edge is not currently supported.

The Sisense Web Application also works in mobile phone and tablet browsers that support HTML5. Click [here](#) to learn more about mobile compatibility.
Supported Operating Systems

For Windows

The ElastiCube Server and ElastiCube Manager can be installed on the following 64-bit operating systems:
- Windows 7, Windows 10
- Windows Server 2008 R2 through Windows Server 2019

If you are installing Sisense in Windows Server 2019, see Disabling Windows Defender Real-Time Protection.

Note: While Sisense supports different versions of Windows, it is highly recommended that your production environments use Windows Server 2008 R2 and later.

For Linux

Sisense Linux Cloud- is certified to run on the following operating systems:
- Ubuntu 18.04 LTS
- RHEL 7.x
- Centos 7
- Amazon Linux 2.0
Hardware Requirements

Sisense scales up to billions of records with typical query response times in under a few seconds.

This section suggests hardware requirements for various performance capacities of the ElastiCube Server when connecting to data sources with Live connections or importing data into an ElastiCube. Actual capacity requirements are provided after consultation with Sisense. Extreme scenarios may require additional resources.

Note the following:

- For Sisense Viewers, only an HTML5 compliant web browser is required.
- For cloud deployments, regardless if you are using AWS/Azure/Rackspace/etc., you must choose a machine that meets the recommended hardware configuration described in the table below.
System Requirements

**Live Models**

When connecting to a Sisense Live data source, Sisense recommends a minimum of 16GB memory with 8 cores. For use cases with 100s of users concurrently connecting to your data source, contact your CSM for more information about Sisense’s minimum requirements.

You can create up to 400 Live models on one Sisense instance.

**ElastiCube Models**

The table below describes the minimum requirements for your production Sisense Server responsible for managing ElastiCubes. It is important to note that it is possible to exceed the large-scale machine deployment size given in the table below with machines that have more memory and cores.

<table>
<thead>
<tr>
<th>Deployment Size</th>
<th>Small Scale</th>
<th>Medium Scale</th>
<th>Large Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>8 Logical Cores</td>
<td>16 Logical Cores</td>
<td>32 Logical Cores</td>
</tr>
<tr>
<td>System Memory</td>
<td>16GB</td>
<td>64GB - 128GB</td>
<td>256GB - 512GB</td>
</tr>
<tr>
<td>Sisense Installation Directory</td>
<td>~20GB</td>
<td>~20GB</td>
<td>~20GB</td>
</tr>
<tr>
<td>ElastiCube Data Directory</td>
<td>~50GB</td>
<td>~200GB</td>
<td>~800GB</td>
</tr>
<tr>
<td>Application Logs Directory</td>
<td>3G</td>
<td>5G</td>
<td>5G</td>
</tr>
</tbody>
</table>
Sisense Recommendations

Sisense recommends that your Sisense server meets or exceeds the recommendations listed above. The actual requirements of your Sisense server may vary depending on the number of concurrent users, builds running in parallel, number of ElastiCubes hosted on a server, the complexity of the ElastiCubes, and additional factors specific to your server, for example, non-Sisense applications running on the same server. When deploying Sisense on a single server, it is recommended to plan in advance for possible additions (for example free slots for RAM or hard disk) on physical machines or virtual scale up (for example adding computing resources) on virtual machines. It is also highly recommended to use a dedicated machine (physical or virtual) and refrain from using the Sisense server for other services.

For optimal performance, Sisense highly recommends that your servers have a processor that supports AVX (Advanced Vector Extensions), which is leveraged by Sisense for improved query performance and user concurrency.

The following table lists sizing recommendations for a Sisense installation deployed on a single Windows server, based on average use cases and hardware demands. The specific recommendations depend on the specific query load patterns and use cases.

If you find that you are running in the upper limits of the large-scale recommendations, consider splitting the server functionality into multiple nodes or a Linux architecture deployment.

<table>
<thead>
<tr>
<th></th>
<th>Small-Scale</th>
<th>Mid-Scale</th>
<th>Large-Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users</td>
<td>50</td>
<td>200</td>
<td>500</td>
</tr>
<tr>
<td>Active (Concurrent) Users</td>
<td>10</td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Small-Scale</td>
<td>Mid-Scale</td>
<td>Large-Scale</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>ElastiCubes</td>
<td>5</td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td>Number of tables per ElastiCube</td>
<td>20</td>
<td>50</td>
<td>70</td>
</tr>
<tr>
<td>Parallel ElasticCubes Build Processes</td>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Live Models</td>
<td>100</td>
<td>400</td>
<td>400</td>
</tr>
<tr>
<td>Max Number of rows per Table</td>
<td>20M</td>
<td>300M</td>
<td>1B</td>
</tr>
<tr>
<td>Number of Custom Tables in an ElastiCube</td>
<td>5</td>
<td>10</td>
<td>25</td>
</tr>
<tr>
<td>Number of Custom Columns in an ElastiCube</td>
<td>10</td>
<td>20</td>
<td>50</td>
</tr>
<tr>
<td>Number of Different Sources (Connectors) per ElastiCube</td>
<td>3</td>
<td>5</td>
<td>10</td>
</tr>
</tbody>
</table>
Expected Usage for a Single Machine

Sisense customers vary in terms of production deployments that represent different needs. The following is typical usage for a single Windows machine, which can serve as a good indication of the typical single machine load. Such a machine will be mid-scale or large-scale, depending on the number of concurrent users and the load of the build process. If your expected usage is approaching the maximum values, consider splitting the server functionality into multiple nodes or a Linux architecture deployment.

Note that maximum values represent the maximum load tested in different scenarios, however, there is no hard limit that will prevent a Sisense system on a single machine from crossing these values*(1).

<table>
<thead>
<tr>
<th>Category</th>
<th>Typical Deployment</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of ElastiCubes</td>
<td>5-10</td>
<td>40</td>
</tr>
<tr>
<td>Number of Tables per ElastiCube</td>
<td>5-15</td>
<td>150</td>
</tr>
<tr>
<td>Data Size (Total ElastiCubes)</td>
<td>20-30 GB</td>
<td>1000 GB</td>
</tr>
<tr>
<td>Concurrent Users</td>
<td>10</td>
<td>100</td>
</tr>
</tbody>
</table>

*(1) Appropriate license may be required to reach a certain load. For example, the number of maximum rows in a table can be limited by the purchased Sisense license regardless of the system capabilities.
Considerations

The following are some considerations to take into account when deploying Sisense:

**CPU Considerations**

Each Sisense widget produces at least one query to the ElastiCube or Live source when viewed. As a dashboard is a collection of widgets, viewing a dashboard produces multiple queries, which in turn raises CPU usage. Concurrent queries affect CPU utilization and are independently spread between different CPUs. While this concurrency is a core activity of Sisense, it is important to consider the correlation between the number of concurrent users/active dashboards and increasing CPU usage. It is especially important to track CPU usage when increasing the number of concurrent users or active dashboards. Lack of CPU resources will cause dashboards to load slowly. When CPU usage consistently increases above 80%, it is recommended to add CPU resources.

In addition, custom tables and custom columns will increase the CPU load at the end of a build process according to the complexity of the functions used by those custom tables and columns. In this case, adding more CPU resources will help to decrease the total build time.

Building an ElastiCube can consume a lot of CPU resources. However, CPU consumption is not linear while performing parallel builds, and it is possible to run several builds at the same time. A single Windows server supports up to four parallel builds. Sisense is typically capable of running four parallel builds while consuming only half of the CPU resources compared to the same builds run separately. It is possible to run build processes and view dashboards at the same time, however, as both are CPU consumers, Sisense recommends that you schedule builds to run at times of low dashboard viewing, like during the night.

**RAM Considerations**
Building ElastiCubes and viewing dashboards both require memory. While it is possible to view dashboards at the same time as running build processes, the RAM resources required for those activities will be aggregated. When using a single-server Sisense deployment, it is important to take both into account and plan RAM usage accordingly. To prevent out-of-memory build failures, it is advisable to run the build process at low-usage schedules or separate the build and the query nodes to different machines.

The size of your ElastiCube model, whether it is the row-count per table, the number of tables, or a combination of the two, has a major impact on the memory consumption of a Sisense server during a build and query. Memory consumption often peaks when building a custom table. Columns with unique data types, like strings, which are indexed during the build process, are cached in memory, for example, large string columns with many unique values may increase memory consumption.

There is a correlation between the data size on the disk and the required RAM. Sisense InChip® technology will load large portions of the ElastiCube to the memory hence the number of active cubes and the cubes total size is a very important sizing factor for the RAM size.

**Disk Space Considerations**

The Sisense installation file is typically around 1.5GB in size. Once Sisense is installed, Sisense uses about 20GB of space. In general, when determining how much disk space to reserve for Sisense, keep in mind the size of the data to be imported into your ElastiCube models. You should keep additional space to support your ElastiCube models, as these are duplicated during the build process. The duplicate copy is not removed at the end of the build process but serves as the basis of the next build.
An additional important consideration is the custom columns and tables, which are materialized on the disk during the build process. Hence it is important to add the required disk space for these data to the total estimated data size. In addition to the imported data size, the ElastiCube also holds metadata like indexes. Actual cube size on the disk will be extended due to this internal metadata, hence it is good practice to plan for 20% overhead on top of the imported data and custom tables.

Typically the ElastiCube data is stored compressed. However, it is good practice to plan for the required disk size without taking the compression into account, since in different scenarios one may choose to disable the data compression for build and query performance improvements.

If you need to store your ElastiCube models in another location other than the default C drive, see Change the Location of ElastiCube Data Folders.

For staging or demo environments, consider using high-speed hard disks (e.g. 7200RPM+) for the home directory and backups, while for production environments, use Solid State Disks (SSD) utilizing NVMe to ensure high I/O performance.

**Page File Considerations**

Windows OS page size settings have an effect on performance and should be considered. The default recommendation is to set the page size to 150% of the RAM size. When possible, and disk space is not an issue we recommend to set the page size to 300% of the RAM size. By extending the page size one can use storage resources to avoid out-of-memory errors but need to consider that when paging is used the system will respond slower. In a large system with 128GB RAM or more, it is not always feasible to set a huge amount of page file on the server as it requires very large disk space. For the server's with a large amount of RAM, you might want to limit the page file size equal to 128 GB at least.
In addition, it is recommended to split page files on two different drives (Preferably on two different Physical/Virtual Disks) on the server for better disk I/O performance.

**Data Sources Considerations**

Connectors affect memory consumption linearly. The usage of each connector has a predefined (and configurable) memory allocation, hence increasing the number of different connectors which are used during the build process increases the memory consumption.
Sisense Basic Concepts and Terminology
Sisense Architecture

The Sisense system is comprised of the following components.
ElastiCube Manager

Sisense is a visual environment in which you create ElastiCubes. Sisense is available online or locally on the desktop where Sisense is installed. Sisense enables you to structure and import multiple data sources, create relationships between data, and perform ETL (Extract, Transform, and Load) processes to prepare data for analysis and visualization. For more information, see ElastiCube Manager.
ElastiCube Server

The ElastiCube Server is installed locally on your computer and enables access to ElastiCubes. Both the Sisense Web Application and Sisense query the ElastiCube Server and receive results.
Sisense Server Console

The Sisense Server Console provides administration functions for managing ElastiCubes on the ElastiCube Server.
Sisense Web Application

The Sisense Web application is an interactive web application that provides the user interface in which users can design, share, view and explore dashboards. The Sisense Web application runs in popular web browsers and enables access to the Sisense server. Dashboards can also be viewed on mobile devices.
Sisense Application Server

The Sisense Application Server is installed locally on your computer and hosts the Sisense Web Application.
Data Models

Data models are abstract entities that organize your data and determine how your tables relate to one another. Sisense has two types of models, Live models and ElastiCube models. Which model you use depends on how you connect to your data source. If you are connecting to a live data source, you will use a Live model. If you are importing your data into Sisense, you will use an ElastiCube model.

Data models enable you to mash up terabytes of data from a variety of sources, for example:

- Traditional relational databases, such as SQL Server, MySQL or Oracle
- File-based data sources, such as spreadsheets (Excel) and CSV files
- High Performance databases such as Snowflake, Redshift, Big Query
- Online web services, such as Salesforce.com, Google AdWords, Google Analytics, Zendesk and more

Live and ElastiCubes models are created and managed from the Data page. Both are described in more detail below.
Live Models

Live models are a type of data model used to manage the schema over your Live data source. Live connections are useful for getting near real-time updates, with changes in your data reflected in your dashboard. For more information, see Sisense Live.
ElastiCube Models

ElastiCube are Sisense’s proprietary super-fast data stores. ElastiCubes are specifically designed to withstand the extensive querying typically required by business intelligence applications. ElastiCubes are based on Sisense's proprietary In-Chip technology, which leverages the Sisense engine. Read more about the technology behind ElastiCube in Sisense's technology section.
ElastiCube Sets

Sisense ElastiCube Sets are collections of ElastiCubes with identical schemas that support high availability by allowing you to query running ElastiCubes within the ElastiCube Set while other ElastiCubes are building.

For more information see ElastiCube Sets.
Widgets

Each widget is a dynamic visualization of data. You pick the type of data to appear in a widget and you pick the type of visualization (chart type). A few examples of widgets are displayed below:

Column Chart Widget
Indicator Widget

GROWTH OF VISITORS

13.2%

# of Visitors  220K
# Pivot Widget

<table>
<thead>
<tr>
<th>Category</th>
<th>Female</th>
<th></th>
<th>Male</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Revenue</td>
<td>Total Quantity</td>
<td>Total Revenue</td>
<td>Total Quantity</td>
</tr>
<tr>
<td>Apple Mac Desktops</td>
<td>1,151</td>
<td>4</td>
<td>26,071</td>
<td>47</td>
</tr>
<tr>
<td>Apple Mac Laptops</td>
<td>5,310</td>
<td>18</td>
<td>50,717</td>
<td>110</td>
</tr>
<tr>
<td>Calculators</td>
<td>600</td>
<td>15</td>
<td>3,147</td>
<td>38</td>
</tr>
<tr>
<td>Camcorders</td>
<td>2,790</td>
<td>23</td>
<td>26,890</td>
<td>124</td>
</tr>
<tr>
<td>Camera Flashes</td>
<td>1,168</td>
<td>12</td>
<td>18,140</td>
<td>115</td>
</tr>
<tr>
<td>Car Amplifiers</td>
<td>734</td>
<td>6</td>
<td>4,619</td>
<td>32</td>
</tr>
<tr>
<td>Car Speakers and Subwoofers</td>
<td>612</td>
<td>9</td>
<td>6,968</td>
<td>90</td>
</tr>
<tr>
<td>Cell Phones</td>
<td>94,323</td>
<td>638</td>
<td>605,011</td>
<td>2,122</td>
</tr>
<tr>
<td>Digital Cameras</td>
<td>34,045</td>
<td>153</td>
<td>216,707</td>
<td>669</td>
</tr>
<tr>
<td>DVD Players</td>
<td>1,488</td>
<td>23</td>
<td>5,309</td>
<td>85</td>
</tr>
</tbody>
</table>
Scatter Chart Widget

To learn more about the different widget types, click [here](#).
Dashboards

A dashboard is a collection of one or more widgets that visualize the data that you select and design.

You define which widgets appear in the dashboard, their design, how they are organized and the filtering of the data that appears.

You can access your own dashboards and dashboards others shared with you. Sisense allows you to create as many dashboards as you need. No limitations!

When you create a dashboard, you are the dashboard’s owner. Owners can share dashboards that they have created with other Sisense users.

Sisense provides a variety of built-in automatic dashboard sharing features. You can easily share a dashboard with others in the Sisense environment or have a dashboard automatically delivered to your chosen recipients by email (on a scheduled basis or upon each data update).
Fields

Widgets are composed of fields that represent the data in your data sources. You create widgets by simply selecting from the fields displayed in a Data Browser, which appears in various places across the product. Each field represents a column of data in the ElastiCube.

When designing widgets, fields can be categorized into three groups:

**Numeric Fields**: Numbers (quantified data), such as salaries, sales, scores, number of clicks and so on. This is data that you may want to aggregate or calculate. For example, the sum of sales or the average of costs.

**Date Fields**: Dates can describe both date and time values. Dates can be used to organize your data into hierarchies according to year, quarter, and month.
or into buckets of time such as by hour or 15 minute intervals. For more information, see Working with Time.

**Descriptive Fields**: Items used to label and categorize, such as Products, Locations, Categories and so on.

Generally, widgets combine both these types of fields.

For example, to show the sales of a product over time, Sales is the numeric data and both Products and Time are descriptive data.

Descriptive data can be considered as numeric when it is simply a count of the number of items. For example, Activities can be considered numeric when it is a count of the number of Activities. A number can be considered as descriptive when it is an actual entity. For example, an identification ID or phone number.

For information regarding Sisense internals and a more advanced explanation of the Sisense architecture, click Sisense Architecture Internals.
Sisense User Roles

Sisense licenses three categories of roles:

Administrator

Admin: Administrators can access the Admin page of the Sisense Web Application from where they can manage all system configuration, data sources, ElastiCubes, users, user groups and more. Sisense supports multiple Administrators.

Note: Sys.Admin is a special type of Administrator who installs Sisense on your system. There is only one Sys.Admin per account.

Data Admin: Data Administrators can manage all data sources in the system. They have access to the Admin page of the Sisense Web Application where they can only manage ElastiCubes and live connections. Data Administrators can also add and manage servers. This is useful for migrating Sisense across environments. Data Administrators do not have access to User Management and System Configuration.

Designer

Data Designer: Data Designers can create and edit ElastiCubes they created or were shared with them. Data Designers can access the Admin page of the Sisense Web Application where they can manage the ElastiCubes and live Connections they have access to in the Data Sources page. In addition, Data Designers can see servers, but cannot add new servers. Data Designers do not have access to User Management and System Configuration.

Designer: Designers can create, design, edit and share dashboards. Sisense Designers determine whether the user with who they share a dashboard has editing rights (is a Designer) or only viewing rights (Viewer).

Viewer

Viewer: Viewers can view, explore, drill down, make selections and filter the dashboards that Designers share with them. Viewers only need a standard web browser; no plugins or downloads. They cannot create new dashboards or edit existing ones.

Note: This roles can be customized through the Sisense REST API. For more information, see Customizing User Roles.
<table>
<thead>
<tr>
<th></th>
<th>Administrator Roles</th>
<th>Designer Roles</th>
<th>Viewer Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Admin</td>
<td>Data Admin</td>
<td>Data Designer</td>
</tr>
<tr>
<td><strong>System Management</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access Admin Page</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Edit System Configuration</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Configure SSO</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>User Management</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create/Delete Users</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create/Delete Groups</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add Active Directory Users</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ElastiCube Management</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add/Delete Servers</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>View Servers</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Add Live Connections</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Create/Delete ElastiCubes</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Edit ElastiCubes</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Share ElastiCubes</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Edit Servers</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Create ElastiCube Sets</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Role</td>
<td>Administrator Roles</td>
<td>Designer Roles</td>
<td>Viewer Roles</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td></td>
<td>Admin</td>
<td>Data Admin</td>
<td>Data Designer</td>
</tr>
<tr>
<td>Delete ElastiCube Sets</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Manage Hierarchies</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Configure Data Access Rights</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Stop/Restart Servers</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Stop/Restart ElastiCubes</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Set Up Data Security</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Access Data Page</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Dashboard Design</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create/Edit/Delete Dashboards</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Copy Dashboard to Another Server</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Restore Dashboards</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Change Color Palettes</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create/Edit/Delete Widgets</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Rename Widgets</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Duplicate Widgets</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Change Widget Type</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Role</td>
<td>Administrator Roles</td>
<td>Designer Roles</td>
<td>Viewer Roles</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td></td>
<td>Admin</td>
<td>Data Admin</td>
<td>Data Designer</td>
</tr>
<tr>
<td>Drill into Widget</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Reorder Widgets</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Create/Delete Dashboard Filters</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Toggle Dashboard Filters On/Off</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Create/Delete Widget Filters</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Toggle Widget Filters On/Off</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Sharing**

<table>
<thead>
<tr>
<th>Role</th>
<th>Administrator Roles</th>
<th>Designer Roles</th>
<th>Viewer Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export Dashboards to PDF, Img, CSV</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Share Dashboards</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Migrate Dashboards</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Pulse**

<table>
<thead>
<tr>
<th>Role</th>
<th>Administrator Roles</th>
<th>Designer Roles</th>
<th>Viewer Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Pulse</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Create Data Alert</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Create Build Alert</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Delete Alerts *</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Administrator Roles</td>
<td>Designer Roles</td>
<td>Viewer Roles</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td></td>
<td>Admin</td>
<td>Data Admin</td>
<td>Data Designer</td>
</tr>
<tr>
<td>Edit Scripts</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

* The user must be the owner of the dashboard, ElastiCube, or alert.
Sisense Architecture

This topic provides an advanced description of Sisense’s architecture for IT managers and Administrators responsible for supporting Sisense in their organization.

For a high-level overview of the Sisense architecture, click "Sisense Architecture" on page 29.

When working with Sisense, there are two possible types of deployments. The first type is a Single Server Deployment in which Sisense is installed on a single server. Single Server Deployments can be leveraged to support single integrations or as proof of concept integrations for larger deployments.

The second type is a Distributed Deployment where Sisense is installed on multiple machines, but only part of the components are active on each one. This type of configuration is used to support heavier traffic loads and "Multi-Node Deployments" on page 1056.

The Sisense full-stack solution is provided in a single installation process. After installing Sisense on a server, you will have access to all of the Sisense functionality.

The diagram below describes the main components of a Sisense installation and how Sisense supports a full-stack solution:
In this diagram, the yellow components specify Sisense client applications or APIs. The blue components specify components of the Sisense server.

The Windows service and application names that you can see in the Windows Task Manager under the Services or Processes tabs appear in a dark blue font. When Sisense is installed on a single server, the dotted line indicates the boundaries of the server firewall. The lines between the components indicate communication between them, and its direction. When the Sisense Server listens on a specific port for inbound traffic, the relevant ports must be open in the firewall for the communication to succeed.

The diagram above describes the ports used by the services, for more information about communication ports, see Communication Ports for Single Server Deployments.
Sisense Architecture

Sisense supports a full stack solution from data, such as transaction details, user information, and machine-generated data, to dashboards accessible by viewers. The Sisense Server is responsible for managing data. Before you can begin to manage data however, Sisense communicates with remote servers for importing data sources, licensing information, tracking and similar activities.

The Sisense Server manages data and supports the Sisense Web Server, which hosts the Sisense Web Application. In a Single deployment, the Sisense Server resides locally on your machine. In a Clustered deployment, the Sisense Server can reside on one of your machines or across many machines depending on your needs. For example, if you want to support a high availability environment, you may install the ElastiCube Server on one machine, the Application Server on another, and the Multi-Node Orchestration server on another machine.
Sisense Server

The Sisense Server is comprised of the following components:

**Web Server**: The Sisense Web Server is installed locally on your computer and hosts the Sisense Web application. It provides access to Sisense dashboards and Sisense Admin screens for user management. Each dashboard created is maintained within the Web Server catalog and application database, which contains metadata regarding users, groups, dashboards, data models, and system settings. The Web Server is comprised of many additional services required by the Sisense front-end, including a Proxy service for routing application requests, query services that transform JAQL to SQL, plug-in management, and more.

There are three main client applications that communicate with the Application Server:

- **Sisense Web App**: This Sisense Web Application is the client application for dashboard creation and access, and system administration.
- **Sisense Mobile App**: This Sisense Mobile Application is an Android and Apple-device compatible application for viewing dashboards across devices.
- **REST API**: The Sisense REST API provides programmatic access to the Sisense Application Server functionality including managing ElastiCubes, user and group management, white labeling, user roles, reporting, and security.

**ElastiCube Server**: The ElastiCube Server supports the management of ElastiCubes and the data they contain. The Sisense Server also supports the Application Server, which queries the ElastiCube Server where the ElastiCubes are hosted. Within the ElastiCube Server are the following main components:

- **ElastiCube Catalog**: The catalog maintains a list of ElastiCubes contained within the ElastiCube server.
- **Data Connectors**: The Data Connectors service is responsible for managing both native and 3rd party connectors.

**Client Applications**:

- **Sisense ElastiCube Manager**: A Windows client application for managing data sources. The ElastiCube Manager provides the GUI for creating ElastiCubes, including importing data sources, preparing them, and building the ElastiCubes.
**Sisense Shell (PSM.exe):** A command line interface for managing ElastiCubes, such as building an ElastiCube, attaching and detaching ElastiCubes from a server, updating data sources, etc. The Sisense Shell can also be controlled programmatically.

**Sisense Server Console:** A Windows application that controls Sisense servers, including attaching ElastiCube folders to Sisense servers and starting and stopping ElastiCubes.

**Licensing:** This service checks and validates licensing information.

**Multi-Node Orchestration:** The Sisense Orchestrator Service is an automated service that can be configured to synchronize builds across the ElastiCube Set.

**Remote Support Analysis:** This service is responsible for controlling logging, monitoring and automatic software updates.

**Message Bus:** The Message bus is a module responsible for transporting events across Sisense components.
Sisense Remote Servers

Sisense remote servers are used for license management, monitoring and software updates.
Data Sources

Data sources may be files or databases located on servers on a local network, or remote locations and web services such as SalesForce and Google AdWords. Sisense supports a variety of data sources through native connectors, customer REST connectors, and ODBC drivers. These connectors are used to import data into the ElastiCube Server.
Downloading and Installing Sisense

This topic describes how to download Sisense and the installation process for getting Sisense up and running.
Downloading Sisense

If you want to try Sisense for the first time, sign up for the free trial at Sisense.com.
Otherwise, you can download the latest version of Sisense here.
See Sisense Release Versions if you need to download a previous version of Sisense.
Installing Sisense

The entire Sisense solution can be installed from one file, either locally or in a central place in your organization within a couple of minutes.

All Sisense components are installed with a default configuration, as follows:
- Sisense Web Application
- ElastiCube Manager
- Sisense Server Console
- ElastiCube Server

**Note:** To install Sisense, you must have administrator privileges on the machine where you are installing Sisense and your environment must meet the Minimum Requirements and Supported Platforms. In addition, Internet Explorer's enhanced security option should be disabled.

**Note:** If you are installing Sisense in Windows Server 2019, see Disabling Windows Defender Real-Time Protection.

**To install Sisense:**

Open the downloaded executable file to run the installation.

**Note:** If your Windows Firewall is active, click Allow access to continue installing Sisense.

If you are not logged into Sisense, enter your username and password to confirm licensing and begin the installation process. If you are logged in...
already, the activation screen is not displayed.
After entering your login details, select **Get Everything** to run the default installation:

Thank you for choosing **Sisense**

To install Sisense, simply choose which installation you would like below:

- **Get Everything**
  - Everything included, recommended settings, no questions asked.
- **Custom Installation**

Welcome | Download & Install | Checking Components | Finish
If you are upgrading your version, click **Upgrade**.
Installing Sisense behind a Firewall

The standard installation requires internet access to download the necessary components. For machines behind a firewall, or without internet access, you can download and install the full self-contained installation file.

For detailed instructions, see Installing Sisense Offline.
Recommended Antivirus Exclusions for Sisense

Some folders, processes, and services may have to be excluded from anti-virus scanning when you use anti-virus software. If these are not excluded, you may see unexpected behavior such as problems accessing or installing Sisense. See Sisense's Recommended Exclusions for the complete list of services, processes, and folders that need to be excluded.
Related Topics:

If you need to customize your installation, see Customizing the Installation. You can customize your installation during the first installation or after an installation, but not during an upgrade.

If you want to install Sisense on a remote machine, see Performing a Silent Installation.

If you are migrating your installation to a different machine, see Backing Up and Migrating a Sisense Installation.

If you have any problems installing Sisense, see the Installation And Upgrade FAQ.
Setting Up Your Production Environment

This topic describes how to set up a production environment for hosting your Sisense implementation.

Sisense recommends installing Sisense in a staging environment, where you can check all of your ElastiCubes and dashboards, as well as test different configurations and customization options. When everything is working as expected, deploy Sisense in your production environment. To learn more about migrating your installation, see Migrating Sisense.

Note: Port 8081, or 443 for secure connections, should be reserved for the Sisense application only. No other applications should run on these ports on your server.
Providing Remote Access to the Sisense Web Environment

To maximize the Sisense experience, you will want to make your dashboards accessible to external users outside the organization’s network. This requires the following configurations:

- Open TCP port (default is 8081) to access the Sisense Web Application internally.
- Ensure that your Sisense machine has a public IP address or domain name associated with a public IP for external access.
- Open port 443 if you are using SSL. Sisense recommends using SSL for secure connections or a VPN solution. For more information about implementing SSL, see Setting Up SSL.

External users who were granted access can view dashboards by entering the machine’s IP and the port number, for example: xx.xxx.xx.xx:8081 or my.company.com:8081

Click here to learn more and see troubleshooting tips.
Providing Remote Access to Sisense

In some cases, you may want to provide external access to Sisense for designing and modeling your ElastiCubes. In such cases it is necessary to consult your Sisense Success Manager to make sure your licensing terms allow this option. In addition, make sure that ports 811, 812 on your Sisense machine are open. **Note:** To avoid exposing your data, do not open ports 811/812 without the proper security measures.
Sisense Email Server

Sisense provides you with an email server for sending reports and notifications to your users from Sisense, to be used in POC and testing environments. When you migrate your deployment to a production environment, you should configure your own custom email server, so you can manage your emails according to your company's policies.

For more information, see Setting Up a Custom Email Server.
Base URLs

Base URLs are the consistent part of a web address for a site or web application. In Sisense, the default address is localhost:8081. Sisense allows you to specify a Sisense hostname as a subfolder of a domain, for example, baseurl.sisense.com/reporting. For more information, see Configuring Base URLs.
Customizing the Installation

This topic describes how you can customize the settings for your web server application during the installation process or after Sisense is installed. For example, you can configure your server to use the Long index edition of Sisense if your ElastiCubes are expected to contain a table with over 300 million rows.

**Note:** You cannot customize the settings during an upgrade process. If you upgraded Sisense and want to customize settings, refer to the post-installation option described below.

**To access the customization options during the installation process:**
While installing Sisense, click **Customize Installation**, and define the settings as described below.
To access the customization options after you have already installed Sisense:

Open the Control Panel, and go to Programs > Programs and Features. Right-click Sisense, and click Change. Click Continue > Change Features, and define the settings as described below.
Settings

ElastiCube Server
Select the Long Index edition if your ElastiCubes are expected to contain a table with over 300 million rows.
It is not recommended to use this configuration setting for ElastiCubes in which the largest table does not include at least 300 million rows.
Performing a Silent Installation

This topic describes how to install Sisense silently.

You can install Sisense using the command-line interpreter, for example, when you need to run an installation on a remote machine.

You can also customize the installation using variables, for example, install Sisense without the sample dashboards.

To perform a silent installation:

Download the latest sisense version.

Open the command line interpreter and type in the installation commands as follows:

Full installation (without previous user activation)

\<Location of downloaded executable file\>\SisenseLatestFull.exe -q -username="<>" -password="<>"

Full installation (for user already activated by Sisense)

\<Location of downloaded executable file\>\SisenseLatestFull.exe -q

Full offline installation (requires a license key from Sisense)

**Note**: To get an offline license, go to [My Account](#) and click GET KEY, or contact your Sisense Success Manager.

After attaining your key, copy and paste it after \-offlinelicense= (replacing \<offline_license_key\> in the example below).

\<Location of downloaded executable file\>\SisenseLatestFull.exe -q -offlinelicense=\<offline_license_key\>

Installation without Sample ElastiCubes and dashboards

\<Location of downloaded executable file\>\SisenseLatestFull.exe -nosamples -q

Choosing Sisense Website name and port

\<Location of downloaded executable file\>\SisenseLatestFull.exe -port=\<port\> -web=\<website\>
file>\SisenseLatestFull.exe -q -
webname=<website_name> -webport=<website_port>
Silent Customized Installations

There are two options available for customizing silent installations:

**bigdata**: Add if you want to install the Long Index version of Sisense. Long Index determines how Sisense transforms your data when loading it into an ElastiCube. Sisense builds your ElastiCube using pointers that represent your data. These pointers are used to find the actual values so no information is lost. To reduce memory consumption and improve performance, Sisense builds ElastiCubes with 32-bit pointers reducing the size of the pointers. You may experience “Out of memory” errors when working with big data such as: 300 million rows of numeric fields, 2GB of string fields. For example, let’s assume that your data contains strings with a length of 1024 characters and 4,194,304 unique rows such that 1024*4,194,304 > 4GB in content after indexing, you should install the Long Index version of Sisense. By installing the Long Index edition, Sisense will use 64-bit pointers. While this is fine for big data, you should use the default installation of Sisense when working with data smaller than the figures mentioned above for faster performance. edition if your ElastiCubes are expected to contain a table with over 300 million rows.

It is not recommended to use this configuration setting for ElastiCubes in which the largest table does not include at least 300 million rows.

**webport**: Add to customize what port Sisense runs on.

For example:

```
SisenseLatestFull.exe -q -bigdata -webport=80
```

After Sisense is installed, the only option to change the port is through the **Port** field in the Configuration Manager.

**To uninstall Sisense using the command-line interpreter:**

In the command-line interpreter, enter:

```
<Location of downloaded executable file>\SisenseLatestFull.exe /q /uninstall
```
Installing Sisense Offline

The standard installation of Sisense requires internet access to download additional necessary components. To support machines behind firewalls, or without internet access, you need to download and install the full Sisense installation file. After you have installed Sisense, you need to provide an offline activation key.

The steps below describe how to install Sisense offline and activate it.
Prerequisites

Before installing Sisense offline, make sure you have the following installed:

Microsoft .NET 4.6.1 Framework

IIS – Internet information Services (Microsoft’s Web server), see the following instructions per your operating system

Windows 7 or 8
Windows Server 2008
Windows Server 2012
Installing Sisense

**Note:** If you are installing Sisense in Windows Server 2019, see [Disabling Windows Defender Real-Time Protection](#).

To install Sisense offline:
- Download the [latest full version of Sisense](#).
- Open the installation file and click **Run**. The Welcome to Sisense window is displayed.

In the Welcome to Sisense window, click **Activate**.

---

By clicking the button above you agree to our [Terms and Conditions](#).
In the Sisense Activation window, click **Behind a Firewall**. The product ID is displayed.

Copy the product ID.

In the My Account page, from the left menu, select **GET KEY**. Paste your product ID from the Sisense installation into the field and click **GET KEY**. Your Off License key is displayed.

**GET OFFLINE LICENSE KEY**

SiSense Prism occasionally validates its license status via an Internet connection. For situations where an Internet connection is unavailable (or blocked by a firewall), an offline license key can be generated from this page instead.

```plaintext
nMaUr4KLZGTXLwTwleaqYD0uqiFUOZ3Mpxt2DTFcxd2eF4EWafo7fDT85hI2
Nx47we6lk3XR94RGBXs5V9k7FgkNs9TPITjQqPqkhET5jAmh+QJR9WCUC52
1Hcz eooDC
Sl8e 75Lmg
JjMd 19gfaI0
/oqt QEqIS
cwbk yYOG
UoTl xlcBoV
Wzyl ydejB
maE 5zgY
Ve9 Sbdyp
515t yfmiq3
FDh 7WmB
oJym jz2AB
sE5+ Ln+Bk
uub6 iarYhr
XrCV idwrM
CaXL@gso9hmp b3sm2n5o  XCd4i32 p6o4jL 4i29UXQh8a+hJ7
zOWN69hvr01DTbQwx7vgho+riK+1ju+P68N38s0Q6hYsdVTLy1kWFTDbjR4/
1o99znfPYwxBzEAgwZ+elwZ2mxcHmuMjlTXfhAEljXymegYZhC8b3JN1jBd/7L
```
Copy the key and paste it into the key area in the Sisense Activation window.

Click **Install**. Sisense is activated and you can begin to install Sisense.
Back Up Sisense

The backup process requires backing up your ElastiCubes and the Sisense Web Application.

In addition, if you have any custom plugins located in the directory `C:\Program Files\Sisense\PrismWeb\plugins`, this folder should be backed up as well.

The procedures below describe how to back up your ElastiCubes, the Sisense Web Application, and Sisense in high availability environments.
Backing Up Your ElastiCubes

Your ElastiCube Server contains information about your ElastiCubes, including the schema and the data that was imported when the ElastiCube was built. This information is saved in a collection of folders stored in the ElastiCube Data folder. The files and folders in the ElastiCube Data must be saved and their structured maintained to ensure your implementation will work as expected on the migrated machine.

**To back your ElastiCubes:**

In Windows Services, stop **Sisense.ECMS**.

Open the directory:

```
%ProgramData%\Sisense\PrismServer\ElastiCubeData
```

**Note:** This is the default path. To check which path is being used, open the Sisense Server Console and click on the Server Preferences icon to verify.
the correct path.

Copy the data to your backup location.
Backing Up the Sisense Web Application

The Sisense Web Application uses information stored in an instance on the Sisense application database where the Sisense Web Application is installed. This instance stores data about users, dashboards, widgets, and other information. This information must be stored to ensure your Sisense Web Application runs properly.

To back up the Sisense Web Application:

In Windows Services, stop the Sisense.Repository and Sisense.Discovery services.

Open the directory:

```
%ProgramData%\Sisense\PrismWeb\Repository\DB
```

Copy the data to your backup location maintaining the same file structure.

Open the directory:

```
%ProgramData%\Sisense\Infra\Discovery
```

Copy the data to your backup location maintaining the same file structure.

(Optional) Back up your custom plugins by copying the directory:

```
C:\Program Files\Sisense\app\plugins
```

For information on how to automate the backup process, see Automatically Backup Sisense Web Data.
Backing Up Sisense in a High Availability Environment

If you have implemented a high availability environment for Sisense and want to back up your installation, the files described below should be backed up in addition to the files described above in Backing Up Your ElastiCubes and Backing Up Your Sisense Web Application:

**Build Node(s):**
- C:\Program Files\Sisense\Sisense.Orchestration\config\config.json

**Web Node(s):**
- C:\Program Files\Sisense\Infra\MongoDB\keyfile. See Creating a KeyFile for more information.
- C:\Program Files\Sisense\Infra\MongoDB\mongodbconfig.conf
- C:\Program Files\Sisense\Infra\Discovery\conf\zoo.cfg
Upgrade Guide

Sisense releases several major versions a year, and a few minor versions in between. Each version includes new features and enhancements as well as other improvements.

Sisense allows you to decide when you want to upgrade to the latest version. There are no automatic updates for Sisense. If you decide to upgrade, follow the steps below to ensure a smooth upgrade. These steps include best practices, instructions for upgrading in single server versus Multi-Node environments, and suggestions for avoiding potential problems that can occur.

**Important**: Which version you are upgrading from matters. If you are upgrading from a version earlier than Sisense V7.2, see the [Sisense V7.2 Upgrade guide](#). If you are upgrading from Sisense V7.2 and later, there are no special requirements.

The following are a list of steps you should review or perform when upgrading Sisense.

- [Researching Your Upgrade](#)
- [Backing Up Sisense](#)
- [Performing an Upgrade](#)
- [Testing Your Upgrade](#)
- [Upgrading Your Production Environment](#)
Next Steps

Researching Your Upgrade
Upgrade Guide

Sisense releases several major versions a year, and a few minor versions in between. Each version includes new features and enhancements as well as other improvements.

Sisense allows you to decide when you want to upgrade to the latest version. There are no automatic updates for Sisense. If you decide to upgrade, follow the steps below to ensure a smooth upgrade. These steps include best practices, instructions for upgrading in single server versus Multi-Node environments, and suggestions for avoiding potential problems that can occur.

**Important:** Which version you are upgrading from matters. If you are upgrading from a version earlier than Sisense V7.2, see the [Sisense V7.2 Upgrade guide](#). If you are upgrading from Sisense V7.2 and later, there are no special requirements.

The following are a list of steps you should review or perform when upgrading Sisense.

- [Researching Your Upgrade](#)
- [Backing Up Sisense](#)
- [Performing an Upgrade](#)
- [Testing Your Upgrade](#)
- [Upgrading Your Production Environment](#)
Next Steps

Researching Your Upgrade
Researching Your Upgrade

Before you upgrade, you should learn about the new features and enhancements in the version you are upgrading to, and check if there are any breaking changes. See the Release Notes for more information. Also, check Sisense’s Minimum Requirements to make sure your environment meets Sisense’s recommendations, and that all the necessary communication ports are open.

If you are upgrading from a version earlier than Sisense V7.2, review the Sisense V7.2 Upgrade guide first for some important changes. **Best Practice**: Sisense recommends that you never install a beta version of Sisense in your production environment. You should also never restore your production environment using a backup of a beta version.

Sisense recommends that you review the following:
- Release Notes
- Minimum Requirements
- Communication Ports

If you are upgrading from a version of Sisense earlier than Sisense V7.2, see the Sisense V7.2 Upgrade Guide. If you are upgrading from version Sisense V7.2 or later, there are no special requirements.
Next Steps

Backing Up Sisense
Backing Up Sisense

The backup process for Sisense requires backing up your ElastiCubes and the Sisense Web Application. In addition, if you have customized your environment, you may want to back up your customizations as well.

This topic describes how to back up your ElastiCubes, the Sisense Web Application, and any customizations you may have performed.
Backing Up Your ElastiCubes

Your ElastiCube Server contains information about your ElastiCubes, including the schema and the data that was imported when the ElastiCube was built. This information is saved in a collection of folders stored in the ElastiCube Data folder. The files and folders in the ElastiCube Data must be saved and their structured maintained to ensure your implementation will work as expected on the migrated machine.

**To back your ElastiCubes:**
In Windows Services, stop **Sisense.ECMS**.

Open the directory:

```
%ProgramData%\Sisense\PrismServer\ElastiCubeData
```

**Note:** This is the default path. To check which path is being used, open the Sisense Server Console and click on the Server Preferences icon to verify.
the correct path.

Copy the data to your backup location.
Back up the Sisense Web Application

The Sisense Web Application uses information stored in an instance on the Sisense application database where the Sisense Web Application is installed. This instance stores data about users, dashboards, widgets, and other information. This information must be stored to ensure your Sisense Web Application runs properly.

You can automate the back up process for the Sisense Web Application. For more information, see Automatically Backup Sisense Web Data.

**To back up the Sisense Web Application:**

In Windows Services, stop the Sisense.Repository and Sisense.Discovery services.

Open the directory:

```
%ProgramData%\Sisense\PrismWeb\Repository\DB
```

Copy the data to your backup location maintaining the same file structure.

Open the directory:

```
%ProgramData%\Sisense\Infra\Discovery
```

Copy the data to your backup location maintaining the same file structure.

(Optional) Back up your custom plugins by copying the directory:

```
C:\Program Files\Sisense\PrismWeb\plugins
```
Backing Up Sisense in a High Availability Environment

If you have implemented a high availability environment for Sisense and want to back up your installation, the files described below should be backed up in addition to the files described above in Backing Up Your ElastiCubes and Backing Up Your Sisense Web Application:

**Build Node(s):**
- C:\Program Files\Sisense\Sisense.Orchestration\config\config.json

**Web Node(s):**
- C:\Program Files\Sisense\Infra\MongoDB\keyfile. See Creating a KeyFile for more information.
- C:\Program Files\Sisense\Infra\MongoDB\mongodbconfig.conf
- C:\Program Files\Sisense\Infra\Discovery\conf\zoo.cfg
Save Custom Configurations

While Sisense preserves your configuration between upgrades, it is a best practice to backup configuration files for any custom configurations you may have in your environment.

The list below includes possible custom configurations that you should back up if you’ve implemented:

**Note:** Save copies of these files in a safe location.

**SSL:** If you have configured secure connections to Sisense, you should back up your SSL certificates. See [Setting Up SSL](#) for more information.

**SSO:** If your users sign in through SSO, you should back up your SSO configuration. See [Single Sign On](#) for more information.

**Plug-ins/Add-ons:** If you have implemented any custom plugins or add-ons, you should back these up. See [Backing Up the Sisense Web Application](#) Step 6 above for more information.

**Metadata Translations:** If you have translated Sisense metadata, you should back up your translations. See [Translating Sisense Metadata](#) for more information.

**Rebranded Email Templates:** If you have customized Sisense’s automated emails, you should back up your templates. See [Rebranding Sisense Automated Emails](#) for more information.

**Message Broker:** If you have implemented high availability for the message broker, you should back up the configuration. See [Replicating the Message Broker](#) for more information.

**FIPS:** If your Sisense implementation is FIPS-compliant, you should back up the MongoDB configuration file. See [FIPS](#) for more information.

**Virtual Machines**

If your Sisense environment is hosted on a virtual machine, you should save an image of your environment and save it in a safe location. If you create an image, you do not need to back up the other custom configuration files described above.
Save Sisense Assets

In addition to your ElastiCubes, dashboards, and any custom configuration files you have backed up, you should also collect information about your environment so you can verify that your system is fully functional after upgrading Sisense. It is recommended that you note the number of dashboards, ElastiCubes, users, and groups you have. Having this information will help you later on when you verify the success of your upgrade.

You can see the complete list of your Users, Groups, and Data Sources in the Admin page.

For a list of dashboards, see the Analytics page in List view.
Next Steps

Performing Your Upgrade
Performing an Upgrade

After backing up your Sisense files and configuration, the next step is to upgrade Sisense. If you are hosting a single Sisense server, then you can upgrade Sisense using the Sisense Installer. If you are hosting multiple Sisense servers, you can use the Multi-Node Deployment tool to silently upgrade each of your servers remotely.

**Best Practice:** Upgrade a staging environment so you can know how the upgrade will affect your users and your configuration before upgrading your production environment.
Upgrading a Single Server Environment

To install the latest version:

Download the latest version of Sisense. For more information, see [Downloading and Installing Sisense](#).

Run the installation file as a Windows Administrator. Sisense automatically detects if a version is already installed, and displays the upgrade wizard.

Click **Continue**.
Click **Upgrade**.
Upgrading a Multi-Node Deployment

To upgrade a Multi-Node deployment, you can use the Multi-Node deployment tool. This tool supports Sisense V7.2 and later. The Multi-Node deployment tool remotely installs or upgrades Sisense on your server to the version you select in the tool.

Before proceeding with the installation, ensure that the relevant communication ports are available on each of the servers. See Communication Ports for Multi-Node Deployments for more information.

To upgrade a Multi-Node deployment:
Download the Deployment Wizard. See Installing the Sisense Multi-Node Deployment Wizard if you have not already downloaded and installed it.
After installing the Deployment Wizard, the screen below should be displayed. If you have closed the Deployment Wizard, in your Sisense High Availability Configuration directory, run SisenseHAWizard.exe.
Enter your Sisense login credentials and click **Sign In**.

Select the version of Sisense you are upgrading to.

Click **Download Agent Installation** to install the Sisense Agent and click **Next**.

In the Deployment Wizard, enter your Windows server credentials. These credentials will be used to connect to each of the servers in your environment, so the user name and password should be the same for each.
server and the user must have Admin privileges.

In the **Server Configuration** area, enter the address of your server in the **Add Server** field.

In **Role**, select the role of the server. There are two possible roles, **Application Node** and **Build Node**. If your server is going to support Sisense’s web services or act as a query node, select **Application Node**. If your server is going to be the build node where ElastiCubes are built and then distributed to other nodes, select **Build Node**.

If you selected **Application Node** as your server’s role, you need to define what components it will support. There are three options you can select:

- **Web**: Supports the web services of Sisense
- **Query**: Responsible for running queries and hosting ElastiCubes
- **Web and Query**: The server will support the web application and act as a query node

Click **Add Server** and repeat Steps 6-8 for each server you wish to add.

After each of your servers has been configured, click **Next**.

(Optional) If you have a load balancer, enter its IP address in the **Load Balancer** field. Your load balancer routes requests from your application nodes to your active query nodes to maximize performance and capacity utilization. This ensures communication between your load balancer,
In the **MongoDB Configuration** area, create a replica set for supporting your application database. This ensures that each application database has the same metadata about ElastiCubes, dashboards, and filters. For more information, see [Creating Replica Sets](#).

Click **Next**. Sisense will automatically complete the setup of your Multi-Node environment. Close the Deployment Wizard once the setup is complete. After you have created your environment, the next step is to define how builds are distributed across all your servers. See [Distributing ElastiCube Builds to Query Nodes](#).
Next Steps

Testing Your Upgrade
Testing Your Upgrade

After you have upgraded your staging environment or your production environment, Sisense recommends you test the following to verify that your upgrade was a success:

If you have implemented SSL, access the Sisense server from an external network, using SSL, and ensure the dashboards load as expected.

If you have implemented SSO, log in to Sisense using SSO, and make sure that a user can see all of their dashboards.

If you use any plugins or add-ons, load dashboards using plugins, and make sure it’s loaded correctly.

If you have rebranded Sisense, check the following:
  Rebranded emails are sent and are displayed as expected
  Homepage and logo are displayed as expected
  Dashboards that are embedded into your site and application are working as expected

If you have implemented active directory, log in to Sisense using an account from active directory and make sure your user can log in and see all of their dashboards.

Create a new ElastiCube and import a data source (Excel or CSV file, or any other).

If you have dashboards or widgets using custom scripts, load the dashboard or the widget to make sure they’re loaded correctly.

Run a build of an existing ElastiCube successfully.

Create a new dashboard, and add a Table or Pivot widget. Ensure the widget loads data.

If you collected information about your environment, including how many ElastiCubes, dashboards, users, and groups you have, verify that the number of assets in the upgraded deployment is correct.

Export a widget to Excel. Ensure the file is downloaded and has the relevant data.

Export a dashboard to PDF. Ensure the PDF is created and opened successfully.

Send a PDF report of a dashboard through Sisense and ensure it’s received.
If you have a multi-node deployment, *in addition* to the list above, test the following as well:

- Shut down one of the query nodes and verify that dashboards return the correct result.
- Verify that ElastiCube distribution works and that the **Last Build** time in the Data Source section of the **Admin** page changes.
Next Steps

Upgrading Your Production Environment
Licensing Terminology

The following is a list of some of the terms defining license scope or product options that will be reflected on the Sales Order defining the license granted in the Software.

“Administrator(s)” means the Authorized User(s) who are Personnel of Licensee who have the authorization and access for server, user and data management.

“Advanced IT Package” means a set of features that are bundled in a license option for Authorized Servers, that is licensed hereunder if specified on the applicable Sales Order and that includes:

SSO is defined at this link: /documentation/configuring-single-sign-sso/

API is defined at the following links:

REST API – https://documentation.sisense.com/sisense-rest-api/

SQL API – https://documentation.sisense.com/sql-api/

SDK means the Software Development Kit, a component of the Software, which allows the development or configuration of the Software.

“Authorized Users” means individual users granted access to use the Software on a named basis. Each Authorized User shall receive a personal login and password which shall be maintained securely by Licensee from unauthorized use. The number of Authorized Users licensed hereunder is specified on the Sales Order(s) executed hereunder.

“Authorized Servers” means the number of Licensee’s servers on which the Software may be installed as specified in the applicable Sales Order. For purposes of this Agreement, each Authorized Server shall be licensed for a specific number of Rows or Cores as specified on the applicable Sales Order.
“Backup/Development Server” means a Software instance that is to be used for the sole purpose of development, backup, staging and other non-production uses and may not be used in any way for production use.

“Central Monitoring Service” means a set of features permitting monitoring by Licensee of certain technical aspects of the operation of the Software. The features are bundled in a license option for Authorized Servers, that is licensed hereunder if specified on the applicable Sales Order. The features currently consist of:

- **Sisense Monitor**: a service that collects data about query, build and Authorized Server performance and displays this information in a dashboard to assist Licensee in monitoring and troubleshooting potential issues.

**Note**: Licensing of Authorized Servers with the Advanced IT Package is a prerequisite for licensing the Central Monitoring Service. The Advanced IT Package includes a license to permit use of the Central Monitoring Service on up to two (2) Authorized Servers. Use of the Central Monitoring Service on additional Authorized Servers must be purchased and reflected on the applicable Sales Order.

“Core” shall be a collection of one or more processor threads and a set of shared execution resources. A processor thread is the architectural state within a processor that tracks execution of a software program thread/task. Hyperthreading and other current and future technologies that materially expand the processing capacity of a Core shall not be permitted to increase the licensed processing capacity of the Core-based licenses granted under this Agreement unless otherwise agreed by the parties.

“Dashboards” means the graphic representation of data extracted from the data source(s) ingested into the Software as designed and created by Designers and/or Administrators using the Software in accordance with this Agreement.

“Designer(s)” means the Authorized User(s) who are Personnel of Licensee that are authorized to create, edit and share data models and Dashboards.
“Natural Language Package” means a set of natural language generation features available for license as an add-on to the Authorized Server licenses as described in the Documentation.

“Personnel” shall mean employees and contractors of Licensee where such personnel's access to the Software is controlled by Licensee.

“Row” shall be a single record in a table of a Sisense data model. For licensing purposes, only Rows in tables that were added to the data model from external sources are counted (i.e., Base table). If a base table is used by several data models, it is counted only once.

“Viewer(s)” means the Authorized User(s) who are Personnel of Licensee that are authorized to view and filter the Dashboards that Designers share with them via a standard web browser. An Unlimited Viewer license allows unlimited Viewers to be authorized to access the specified Authorized Server. Please note that while there is no legal limit to the number of Viewers, the technical capacity of the server may limit the number of Viewers that may actually access the Authorized Server.
Launching Sisensethe Sisense Web Application

This topic describes how to launch the Sisense Web Application and ElastiCube Manager.

To launch Sisensethe Sisense Web Application:

Use one of the following options to launch Sisense:

- If you have Sisense installed on your machine, select Sisense from the Start menu.
- From a standard browser, go to the URL of the Sisense environment provided to you by your System Administrator.

To launch Sisense:

Use one of the following options:

- **From Windows**: Open the Windows Start menu and select Sisense ElastiCube Manager. This opens the desktop version of Sisense.
- **From the Sisense Web Application**: Click Data in the top menu. This opens the Data page where a list of your ElastiCubes is displayed.

To log into Sisense:

- If you installed Sisense, you can use the same credentials to log into Sisense. If an account has been created for you, you should receive a password activation email. Click on the link to activate your account.

After logging into the Sisense Web Application, the Analytics page is displayed.
From the **Analytics** page, you can see dashboards that have been shared with you or some sample dashboards provided by Sisense.
Tutorial: Getting Started with Sisense
Welcome!

In this basic tutorial, you will learn how to connect to data, and how to build your first dashboard.

The demo data in this tutorial is based on sample E-commerce data, and a common market analysis scenario will be used.

If you have already connected to data sources, you can skip this step and go directly to Step 2 – Your First Dashboard.

*Step 1 Connect to Data*
Introduction

To work with data in Sisense you need ElastiCubes. An ElastiCube model is Sisense’s proprietary analytical database, which enables you to connect multiple data sources and run complex queries in split seconds.

You can connect to databases (like SQL Server, MySql, Oracle, etc.), files (text, csv, Microsoft Access, etc.) and online web services (Google AdWords/Analytics, Salesforce, Zendesk, etc.).

In this tutorial you will create your first ElastiCube model using some sample ECommerce data.

Please download the following two sample files:

- [GettingStarted ECommerce.csv](#) – A CSV file with a few hundred thousand entries.
- [GettingStarted Brands.xlsx](#) – An Excel file with data on brands sold in our ECommerce sample.
1. Open the Data Page

Look for **Sisense** in your Windows Start Menu and open it.
From the menu bar, select **Data > + ElastiCube**. This creates an empty ElastiCube model. You can also choose to create a Live model where you connect directly to your data source, but for this tutorial, let’s stick with an ElastiCube.

Now, give your ElastiCube model the name “Tutorial”:

![Add new ElastiCube](image)

You just created an empty ElastiCube. The next step is to add some data to it.
2. Add Data Sources

In Sisense, you can import data into the ElastiCube, or connect directly to your data source with a Live connection. With our CSV and Excel files, we will import these into the ElastiCube.

**Table #1 – CSV File**

Import the base table, which is in the form of a CSV file. This table contains info about what kind of products have been purchased, and how much they cost.

Click **Add Data** and select CSV from the list:
You will be prompted to choose how you want to upload your CSV file. Select **File Upload**.

Drag the GettingStarted.CSV file to the Upload area or click browse and navigate to the file.
After you upload the CSV file, select it from the Uploads list and click **Next**:

Select the GettingStarted Ecommerce file from the Select Table list. You can preview the data inside it by clicking 📊. This displays some more settings.
you can use to customize your data, but for now, click **Done**.

Congratulations, you have connected to your first data source. You can now see the CSV file in your schema. The name of the table in the ElastiCube model defaults to the file name in this case. You can double-click the title to rename it and remove the CSV extension. Your ElastiCube model should look like this:
Table #2 – Excel File

Information about the brands that were sold in the main table exists in another table, this time in an Excel file.

Click **Add Data** and select **Microsoft Excel File** from the list.

You will be prompted to select an Excel file. Find and select “GettingStarted Brands.xlsx”, which you downloaded in Step 1 (you can also download the file [here](#)).

In the new window leave the default settings and click **Add**.

Congratulations, you have now added your second table, an Excel file.

To rename the table, click the table once and select and rename the table to “Brands”.

Your ElastiCube model should look like this:
3. Connect Different Data Sources

So far you have added two tables from two different data sources. Now let’s see how to connect them.

In the ElastiCube model, creating and deleting relationships between tables is as simple as drag and drop.

Any two fields of the same type (numeric/text/date) can be connected. In this sample, both tables have a “Brand ID” column. Let’s connect them:

Drag the Brands table onto the Ecommerce table. This opens the Relationships pane.

Select Brand ID from both tables and click Done. You should now see the two tables are joined through the column Brand ID.
4. Build the ElastiCube

Now that you have defined your ElastiCube model, you are ready to build it. This will pull the data from the data sources into the ElastiCube model.

Click **Build** in the top menu.

The build will start. You will see the progress in Build log. Wait for the build to finish:
Congratulations! You have successfully built your first ElastiCube model.
5. Dashboard Time!

Click **Analytics** in the top menu to open the Sisense Web Application and create a new dashboard.

### Step 2 Your First Dashboard

Welcome to Step 2 of the Sisense tutorial. In Step 1 – Connect to Data of this tutorial, you connected to data and built your first ElastiCube model. Now, you are ready to jump right into creating your first dashboard.

Although it is recommended to complete Step 1 in advance, you can jump right into Step 2 if you wish.

If you're arriving directly from Step 1, you should already be in the process of creating a new dashboard after clicking Analytics in the top menu of Sisense:

To Create Your Dashboard
Click Here
Introduction

Sisense is designed to let business users easily create powerful and meaningful dashboards, without being a developer or data scientist.

To get a feel for the process, Sisense has put together this tutorial which shows you how to do a quick market analysis to identify what products are meeting customers’ needs, and in which markets. You will use appropriate visualizations to create a dashboard that provides answers to four questions:

- **Sales Trend**: Is annual revenue increasing?
- **Optimal Product Mix**: What product lines should we grow?
- **Segmentation**: Which customer segments should we develop?
- **Market Size and Growth**: Where are our best markets located?

To make things interesting, your data set is based on sales and customer data from an E-commerce site.
1 – Create a New Dashboard

Click + to create a new dashboard on the Analytics page.

In the Data Set field, select the ElastiCube model to which you want to connect. If you’ve completed Step 1 – Connect to Data, select ‘Tutorial’ in the Data Set field, otherwise choose ‘Sample ECommerce’.

A default title name matching the ElastiCube model name will be given to the dashboard. Feel free to change it.

Click Create.

Now that you have a dashboard, you can start adding widgets to it.
2 – Create Your First Widget: Sales Trends

Create your first widget to get a general idea of your sales performance.

Click Select Data on the left side of the screen:

Welcome to your new dashboard!
To create a new widget, first select the data you would like to visualize.

A list of available fields will be displayed grouped according to the table they belong to. This window is called ‘Data Browser’, and lets you easily find the fields available in your ElastiCube model. Select the ‘Revenue’ field:
Now you immediately get a ready made widget that shows you all revenue to date. We call this kind of widget an “Indicator”.

But to understand revenue over time, add a time field to the mix:

Click + Add More Data:

In the data browser, hover over the Date field and click More... Then select Quarters:

Click Create to add the line chart widget to the dashboard.

The revenue trend is also positive showing annual growth, so now go ahead and explore what’s driving this growth. It would be interesting to uncover profitable product categories.
3 – Create a Scatter Chart to See Cost and Revenue

Now try to understand the interaction of cost and revenue to discover the most profitable product lines.

On the top of the dashboard click + Widget.
Click Select Data and select the ‘Category’ field.
Click Add More Data and select the ‘Revenue’ field.
Click Add More Data again and this time hover over the ‘Cost’ field and click More... > Average.

Now select the scatter chart icon 🗼 from the widget buttons on the left, and click Create.

It’s simple to see that product categories on the right generate the most revenue and those products in the lower left have the lowest cost and revenue. Hover over each scatter point to see the category behind it, to discover that Cell Phones and PDAs generate most of the revenue. You now understand which product lines to sell, but to get a complete view of your market, you need to identify optimal customer segments to target in each market – start by creating segments based on age.
4 – Creating Pie Charts to Visualize Customer Segmentation

On the top of the dashboard, click **New Widget**. Click **Select Data** and select the ‘Age Range’ field. Click **Add Data** and select the ‘Revenue’ field.

Now select the pie chart icon 🍎 from the widget buttons on the left, and click **Create**.

You can clearly see that customers between the ages of 65+ are the most valuable in terms of the revenue they generate. This insight can be used to focus advertising budgets and develop unique campaigns to grow these customer segments. Now, you can turn your attention to identifying which markets are growing at the quickest rate.
Creating a Map to See Market Growth

Before you focus on growth, you should focus on a specific year first, by adding a filter.

The Filter panel on the right of the dashboards lets you add dynamic filters to the dashboard.

Click Filter Your Dashboard on the right side of the dashboard.
In the Data Browser, click on the Date field.
Click the multi/single selection button to make it a single selection filter.
Leave only 2013 selected, and click OK.

Your dashboard is now filtered to show data only for the year 2013.

Now create a map:

Click at the top of the dashboard.
Click Select Data and select the Country field.
Click Add More Data again and select the Revenue field.
Click on the scatter map button type on the left.
Now you see a map, where the biggest circles represent the countries with
the highest revenue. But if you want to concentrate on the markets with the biggest growth in revenue, it's easy:
Hover and click to open the menu of the Total Revenue field and select **Quick Functions > % Change Over Time > Growth.**

Click **Create.**
Now the biggest circles on the map represent countries with highest growth. You can see for example that the United States is very hot right now. You can use this information to focus resources as well as further analyze these markets to understand revenue drivers.
6 – Rearrange the Widgets in the Dashboard

You can rearrange the dashboard by dragging and dropping, and resizing the widgets. Dragging one widget on top of another splits the area in the dashboard and allocates space for both widgets.

See this animation to get a feeling of how to rearrange widgets:

Go ahead and rearrange your dashboard however you see fit.
You can now easily see all four visualizations showing sales trends, product mix, customer segments and fast growing markets. You can explore and drill into the details to get further insight.
7 – Filtering and Drill Downs

In our pie chart we can clearly see that our strongest age range is 65+. Let’s explore this age range a little more:

Click the 65+ slice in the pie chart.

As a result, a new filter is added to the filter panel on the right, and the entire dashboard is now filtered to focus on this age range.

Taking a look at the updated scatter chart, you can see that Monitors are by far the most revenue generating category for this age range.

Right-click the 65+ slice in the pie chart and select **Drill Into**... from the menu.

A Data Browser will pop up, letting you select any field to drill into. Click on the **Brand** field.

You can now see which brands contribute the most to the revenue of your selected age range.

You can make direct selections by dragging the mouse over the points that interest you in the scatter and line charts as well.
8 – So What Did We See Here?

The dashboard you created enables you to interactively analyze three market priorities – where to sell, what to sell and who to sell to. More importantly it makes it simple to see the interaction of product categories, client segments and revenue, to optimize marketing and sales activities and track performance targets.

This was just a quick glimpse of what you can do. Now it’s time to start exploring on your own.
Release Notes

Sisense Windows Click to open the Sisense on Windows Release Notes

Sisense Cloud Native Linux Click to open the Sisense Cloud Native Linux Release Notes
Manage Data

This section describes data models in Sisense, how to connect to data sources, and how to avoid common mistakes when preparing your schemas.

Data Models

In Sisense, there are two types of data models, ElastiCube models and live models. ElastiCubes are Sisense’s proprietary, high-performance analytical database specifically designed to withstand extensive querying typically required for your business intelligence application. With ElastiCube models, you import data from a variety of data sources into a Sisense ElastiCube, which becomes the database that supports your dashboard. These models then need to be updated as the data in your sources changes.

Unlike ElastiCube models, live models run queries directly against the data source. Sisense performs no additional caching on the data. This provides you with near real-time data updates in your dashboard, though live connections rely on the source database for all queries. This means the queries are only as fast the data source.

In Sisense, Designers create data and live models to represent how data is brought into Sisense and how should it be used when supporting your dashboards. Which model you choose is determined by what you want to do with your dashboard. ElastiCube models tend to be faster when you are working with complex dashboards with large data sets while live models are preferred when you need near real-time updates. ElastiCube models and live models can be used together in a single dashboard to support both use-cases.
From the **Data** page, you can see a list of all the data models that you have created or that were shared with you.

**Related Topics**

[Navigating the Data Page](#)
ElastiCubes

With ElastiCubes, you connect to your data sources and import your data. Once your data is in Sisense, you can then design your ElastiCube model in the Model Editor.

After you have designed your schema, it's time to build the ElastiCube. Building an ElastiCube takes all your data from all your sources, and imports it into the ElastiCube.

The topics below describe ElastiCubes and how to model them.

Related Topics
- ElastiCubes
- Creating ElastiCubes
- Navigating the Model Editor
- Introduction to Data Sources
- Working with Data
- Transforming and Enriching Data
- Functions Overview
- Building ElastiCubes
- Build Settings and Data Accumulation Behavior
- Elasticubes for Advanced Business Scenarios
Live Models

To create dashboards built on live connections to a data source, you create live models. These data models include connection and credential details to the data sources. After you have created your live model, you publish it. Publishing the live model adds it to your list of data models from which you can select when working with dashboards.

The topics below describe live models, and how to create and publish them.

**Related Topics**
- Sisense Live Connect
- Creating Live Models
- Adding Live Connections
- Working with Live Models
- Publishing Live Models
- Working with Live Widgets
Navigating the Data Page

From the **Data** page of ElastiCube Manager, you can manage your ElastiCubes and view their current states.

The **Data** page contains a list of all the ElastiCubes and live models you have created or that have been shared with you.

To locate ElastiCubes or live models easily, on the right side of the page, you can view ElastiCubes and live models you recently worked on, that were created by you, or that were shared with you. In addition, on the left side, you can locate an ElastiCube through the **Search** field.

Each tile contains information about the status of the ElastiCube and live model, its owner, and who its shared with it. You can open your ElastiCube by clicking on its tile or you can manage your ElastiCubes by clicking and selecting one of the following options:

- **Stop**: Click to stop the ElastiCube. ElastiCubes that have been stopped cannot be queried. This is useful for troubleshooting ElastiCubes and issues with your dashboard.
- **Duplicate**: Click to create a copy of a live or ElastiCube model. See [Duplicating Data Models](#) for more information.
- **Restart**: Click to restart an ElastiCube that you have stopped.
**Dependencies:** Click to see what entities are using the ElastiCube or live model. Entities include data security rules, drill hierarchies, dashboards, widgets (from other dashboards), and Pulse alerts.

**Schedule Build:** Click to schedule builds to automatically synchronize with the underlying data source to ensure all data is up to date. See [Scheduling Builds](#) for more information.

**Delete:** Click to delete an ElastiCube or live model. If the ElastiCube or live model does not have any entities dependent (dependencies) on it, then you will be asked to confirm that you want to delete the ElastiCube. Otherwise, you will be prompted to remap your dependencies to another data source. See [Deleting ElastiCubes](#) for more information.

**Cancel all Queries:** Click to cancel all current queries to the ElastiCube.

**Go to Admin:** Opens the Data Source tab in the Admin page where you can manage your ElastiCube or live model.

Some features and functionality supported in the desktop version of Sisense are not supported Sisense currently. These features will be added in the near future. In the meantime, you can find workarounds for missing features at [Sisense V7.0 Backwards Compatibility](#).
ElastiCubes
Introduction

The ElastiCube is Sisense’s unique, high-performance analytics database with super-fast data stores that are specifically designed to withstand extensive querying typically required by business intelligence applications. ElastiCubes allow you to bring in data from multiple sources, and then merge, manipulate and query the data as if it was one consolidated data set. ElastiCubes perform so well, that in most cases the creation of dedicated OLAP cubes and/or optimized data marts are completely unnecessary – even when dealing with hundreds of millions of rows of raw data.

One of the biggest advantages of ElastiCubes is the ability to easily mash up multiple data sources. It is made up of fields where each value in one field has a corresponding value in another field. The data for an ElastiCube can come from one source, multiple sources or even from multiple physical locations. Once the data is inside the ElastiCube, it is all the same and every field coming from every table can be analyzed in the context of any other – quickly.
Benefits

ElastiCube technology make queries over hundreds of millions of rows of raw data return in seconds, with moderate hardware requirements including standard desktop-class computers with commodity hardware. More importantly, ElastiCubes can do this without having to pre-aggregate and pre-calculate the data ahead of time and store it on the hard-drive, thus radically reducing required import/processing time and storage space.

ElastiCubes are most useful when one or more of the following is true:
- Large amounts of data need to be analyzed
- Data for analysis originates from multiple disparate sources

ElastiCubes – Technical Overview

Relational databases (RDBMS) like SQL Server, Oracle, MySQL and even Access all store tabular data row-by-row. This structure is best for transactional/operational systems that require large numbers of concurrent insertions. With indexes, it can also provide realistic query response times for row-based queries that do not frequently require aggregations or joining of many tables.

Data analysis often requires aggregation of data as well as merging of data located in multiple disparate tables. When dealing with these types of queries, relational databases reach their limits pretty quickly. The only way to extend these limits is by putting in stronger hardware and pre-aggregating data to reduce the amounts of calculations that occur in real time.
The ElastiCube Columnar Database

ElastiCube data is held in a Columnar Database Management System (CDBMS) that stores data field-by-field. Each field is individually stored in a memory-mapped file, the same mechanism the Windows operating system pagefile uses for memory dumping and loading.

When a query is executed over an ElastiCube, only fields referenced in the query need to be loaded into memory. This leaves enough space for actually processing the query entirely in memory without any read/write to the hard-drive – the prime reason for poor performance of queries. Once a field is no longer used, it is removed from memory and its consumed space is freed.

This approach has several advantages:

**Query Response Time**

Queries over data sets containing millions of rows of data return in seconds even under modest hardware configurations such as desktop computers.

**Materialization Time**

ElastiCubes do not require pre-aggregations and/or creation of indexes to assure fast query response, therefore the actual creation of an ElastiCube takes a fraction of the time of a data mart or an OLAP cube.

**Storage Space**

Pre-aggregations and the creation of indexes are not needed to assure fast query response, making an ElastiCube’s size significantly smaller than a datamart or an OLAP cube.

**High Compression**

This columnar storage strategy makes the data much more suitable for high levels of compression, without loss of detail or accuracy. This means less hardware is needed; less disk space and less RAM than for an equivalent-sized, traditional Business Intelligence DB.
64-bit Support
Written and designed to natively support 64-bit processing, it vastly increases the amount of memory the system can address at any given time. 64-bit architecture means you can work with virtually unlimited amounts of data.

True Multi-User, Multi-Application Architecture
ElastiCubes are not tightly coupled with the application layer of the system. This frees up a single ElastiCube to handle multiple applications and users. Not having to reproduce your data model for every application saves significant time developing and maintaining your dashboards and reports.
Just-In-Time, In-Memory Processing

**Smart Cache and Instruction Recycling**

CPU cycles and RAM space are the two most precious resources in any computer, and ElastiCube is designed to use both as efficiently and speedily as possible. Using our sophisticated caching algorithm, the data is only loaded into memory when it’s needed. As part of this algorithm, compute- and time-intense calculations are also intelligently cached to further reduce I/O calls.

**Cache-aware Algorithm**

Additional sophisticated algorithms further increase Sisense’s performance. Once data is loaded into memory, the main performance bottleneck becomes CPU cache misses that naturally come with random access. The ElastiCube is specifically designed to minimize these errors by employing a unique cache-aware algorithm, further increasing Sisense’s performance by an additional order of magnitude.

**Compressed Calculations**

Every DB compresses data to save disk space and RAM. ElastiCube is designed to work directly on this compressed data, so that the need for decompression is virtually eliminated, further increasing ElastiCube’s performance.
Designed with Standard Hardware in Mind

Just about every new computer on the market—even portables like iPhones and iPads—are built with very powerful multi-core processors, putting several CPUs into one. ElastiCube was built specifically to take advantage of these powerful CPUs, further increasing Sisense’s performance on standard hardware, enabling you to run multiple applications and support multiple users.

**Highly RAM-efficient**

The thing we know for sure about DBs is that they grow. Fast. So no matter how much fancy footwork is done with completely in-memory DBs, eventually you run out of RAM space and need to upgrade—at least your RAM (best case) or your entire hardware platform (worst, very expensive, case). At Sisense we know this, so we spent years designing the ElastiCube to be able to handle terabytes—billions of rows—of data efficiently and quickly, even on standard PC hardware.
Unified Analytics Engine

Sisense can execute queries against a wide variety of data sources as if they were all of the same type, essentially making the individual characteristics of each physical data source unimportant. Our Unified Analytics Engine is what makes this possible.

When Sisense imports data, the Unified Analytics Engine creates a metadata layer, or abstraction layer, which is then used to formulate queries across any number of tables from any number of data sources in any number of formats. It even supports the combined querying of resident and external (live) database sources without first loading data into the database!

These capabilities provide the user with unparalleled flexibility and speed in creating, executing and sharing highly complex reports, dashboards, and analytic applications, with any number and variety of data sources.
Compliant with Industry Standards

**Supports SQL-92 Standard**
Even with all this advanced technology, we knew that none of it would be any good if our users couldn’t access their existing data. So, we built in an SQL layer to the system, which allows users to integrate Sisense to external applications without needing to learn new scripting languages.

**Seamless Integration with Existing Data Sources**
Got an ODBC/OleDB compliant DB today? Great, we built in the ability to access those, too. ElastiCube will seamlessly connect to those data sources so, again, there is no need to learn a new language or write special code to connect to your existing data. With ElastiCube there’s no need to start over, you just get faster, easier, and more scalable, with minimal need for IT.
Creating ElastiCubes

Before you can analyze dashboards, you need to add data to Sisense and then model it. The first step then is to create an ElastiCube. Once you create your ElastiCube, the Model Editor displays an empty canvas where you can begin to import and model your data.

For a short video overview of creating an ElastiCube, see the tutorial below.

To create an ElastiCube:
Open Sisense and click Data in the top menu. Your ElastiCubes and live models are displayed.

Click + ElastiCube. The Add new ElastiCube dialog box is displayed.
Name the new ElastiCube and click Save.
The Model Editor is displayed, where you can start creating your data model.
Navigating the Model Editor

The first time you create an ElastiCube or a live data model, it's empty until you start adding some data. Once you connect to your data source and select what data is going to be imported or queried, it's represented in Sisense as circular nodes. Each node is a table. The color of a node is determined by its data source, for example, data imported from a CSV file is one color while data imported via SQL is another. The colors of the data source are described in the legend on the bottom right of the editor.

The size of the table reflects the number of relations and the number of columns the table has. So, a table with several relations and a lot of columns will be larger than a table with a single relation and fewer columns. For example:

The ‘Conditions time of stay’ table is a custom table with a single relation and the ‘Admissions’ table contains data from a CSV file and has four relations.

The Model Editor contains a variety of information and functionality that lets you prepare your data model as needed. A sample screen is shown below.
**Navigation Pane:** This area contains a list of your tables and their columns. On the left side of the table name is an icon that indicates what type of data is included, Date, Numeric or Text. On the right side is a join icon when the column is connected to another table. See [Finding Tables and Fields](#) for more information.

**Schema:** The schema contains your data model that represents all the data to be added to the ElastiCube and the relationships between the table. Some of your tables might appear with icon attached them. These indicate the status of your build and table. For example, indicates that the table has changed since the last ElastiCube build while indicates that a custom expression is currently in draft mode. See [Working with Data](#) for more information.

**Schema Menu:** This menu lists the following buttons:
- **Data:** Click to add a new table to your ElastiCube.
- **Custom:** Click to add a custom table to your ElastiCube.
- **Relationships:** Click to open a preview window where you can join two tables. See [Creating Relationships](#) for more information.
- **Build:** Click **Build** to initiate an ElastiCube build.
- **Undo/Redo:** Click these buttons to undo or redo any recent changes to your schema.

**Legend:** The legend lists a description of all your data sources in the ElastiCube and the color that represents them in your schema.
Introduction to Data Sources

Connecting to your data is the first step in analyzing your business information. Sisense has native connectors to many SQL databases, NoSQL sources, and popular web applications.
For a short video overview of connecting to data, see below.

There are two ways you can connect to your data sources. You can either import the data into the ElastiCube or you can connect directly to the data source, which is known as a live connection.
Which connection type you choose depends on your use case and the data source itself. Importing your data into the ElastiCube takes advantage of Sisense high performance database. However, if your data changes frequently and your dashboard must reflect this, than a live connection is probably the best choice. Keep in mind however that live connections work only as well as the data source. The tables below provide a list of supported data sources and links to the documentation for ElastiCube and live data sources.
## Live Connectors

<table>
<thead>
<tr>
<th>Connector</th>
<th>Download Link</th>
<th>More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google BigQuery</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>MemSQL</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>MySQL</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>Oracle</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>Redshift</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>Snowflake</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>SQL Server</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
</tbody>
</table>
## ElastiCube Connectors

<table>
<thead>
<tr>
<th>Connector</th>
<th>Download Link</th>
<th>More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>Bing Search*</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>Box *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>CSV</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>Custom REST</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>Dropbox *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>DynamoDB *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>Elasticsearch *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>Exact Online *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>Excel</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>FTP *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>Generic JDBC Driver</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>Generic ODBC Driver</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>Generic OLE DB Driver</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>Gmail *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>Google AdWords</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>Google Analytics</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>Google BigQuery</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>Google Sheets</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>Heroku Postgres</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>Connector</td>
<td>Download Link</td>
<td>More Information</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>---------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Hive *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>HubSpot *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>Intacct *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>JSON *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>MailChimp *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>MemSQL</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>Microsoft Active Directory *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>Microsoft Dynamics CRM *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>Microsoft Dynamics Nav *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>MongoDB</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>MySQL</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>Oracle</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>Quickbooks Desktop *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>Quickbooks Online *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>Redshift</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>Salesforce</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>ServiceNow *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>SharePoint Excel Services *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>Sisense ElastiCube</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>Snowflake</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>SparkSQL *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>Connector</td>
<td>Download Link</td>
<td>More Information</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Splunk *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>Stripe *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>SQL Server</td>
<td>Pre-Installed</td>
<td>Documentation</td>
</tr>
<tr>
<td>Xero *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>XML *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>YouTube *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>YouTube Analytics *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
<tr>
<td>Other Connectors *</td>
<td>Download</td>
<td>Documentation</td>
</tr>
</tbody>
</table>

* Certified connectors are provided by our certificated partner. Sisense carefully evaluates and certifies each external connector to work with Sisense. This helps us ensure that our external connectors are always up-to-date and fully supported.

**Notes for the certified connectors:**
- The drivers are certified for Sisense v7.2 and above.
- Click the above links to download a 30-days free-trail of the driver. Contact Sisense for the full license version.
- Sisense v7.4 and above: Click the above links to download a ready-to-use driver.
- Sisense prior to v7.4: Click the above links to download a 30-days free-trail of the driver. Contact Sisense for the full license version.
Technology Partner Connectors

Technology partners provide the service of connecting to data sources via the data pipeline, managing API changes and limits, and delivering comprehensive support as part of their service. Import your data into a data warehouse (Redshift, Google BigQuery, Snowflake, SQL Server, MySQL, PostgreSQL, and more) to access your data with either ElastiCube or live data models. Contact your Account Executive or Customer Success Manager to discuss the full functionality our technology partners provide, as well as to start your free trial. For more information, click [here](#).

<table>
<thead>
<tr>
<th>Connector</th>
<th>Connector</th>
<th>Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adobe Analytics</td>
<td>Apple Search Ads</td>
<td>AppsFlyer</td>
</tr>
<tr>
<td>DoubleClick</td>
<td>Facebook</td>
<td>Facebook Ads</td>
</tr>
<tr>
<td>Freshdesk</td>
<td>Jira</td>
<td>LinkedIn Ads</td>
</tr>
<tr>
<td>Magento</td>
<td>Marketo</td>
<td>Netsuite</td>
</tr>
<tr>
<td>Pardot</td>
<td>Shopify</td>
<td>Twitter Ads</td>
</tr>
<tr>
<td>Zendesk</td>
<td>And more...</td>
<td></td>
</tr>
</tbody>
</table>
Vendor JDBC Connectors

With the Sisense's generic JDBC driver you can leverage native JDBC connectors provided by the data source vendors to connect to additional data sources.

<table>
<thead>
<tr>
<th>Connector</th>
<th>More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Athena</td>
<td>Download</td>
</tr>
<tr>
<td>Azure Data Explorer</td>
<td>Download</td>
</tr>
<tr>
<td>Dremio</td>
<td>Download</td>
</tr>
<tr>
<td>PrestoDB</td>
<td>Download</td>
</tr>
<tr>
<td>Teradata</td>
<td>Download</td>
</tr>
<tr>
<td>Vertica</td>
<td>Download</td>
</tr>
</tbody>
</table>
Connecting to More Data Sources and Frameworks

You can connect to many additional data sources by leveraging Sisense’s generic frameworks such as the [generic ODBC driver](https://example.com/generic_odbc), [JDBC driver](https://example.com/jdbc), or [Custom REST Framework](https://example.com/rest). Sisense supports these frameworks with frequent upgrades and new functionality. However, full connectors to specific data sources created through these frameworks are not supported.
Changing Connectivity Settings for Data Sources

This topic describes how to change the connectivity settings for an existing data source.

For a short video overview of changing connectivity settings, see below.

After you have modeled your data in the ElastiCube, a data source may have changed location or you may want to change a table without importing a new table into your schema from scratch. If this happens, you can update your connection settings to the data source.

When updating connectivity settings, you can choose one of the following:

- **Change Provider**: This enables you to change the source of the data being accessed, for example changing a provider from SQL Server to MySQL. After you select the new provider, you need to choose the relevant database and tables from the new provider. This is useful if you want to replace a table in your schema with a new data source, but don’t want to change your schema.

- **Change Connection**: This enables you to use the same provider, but change your credentials to the data source or the database previously selected.

- **Change Table**: This enables you to use the same provider and credentials, but change the database previously selected.

If a table has changed, but the source and its settings are the same, for example a new column was added, and you want to update your schema without importing the data into the schema all over again, click **Refresh** in the table’s menu.

**Note**: The data source name, columns, and column types must remain the same for existing widgets in the dashboard to be able to reference the data correctly.

The table below describes which sources support changing connectivity settings in the web-based ElastiCube Manager:
<table>
<thead>
<tr>
<th>Data Source</th>
<th>Connection Settings</th>
<th>Change Database</th>
<th>Table Settings</th>
<th>Sync Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excel</td>
<td>N/A</td>
<td>N/A</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>CSV</td>
<td>N/A</td>
<td>N/A</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Google Sheets</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MySQL</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SQL Server</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Oracle</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Redshift</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**To change connectivity settings:**

In your schema, select the Data Source view in the Navigation Pane.

Data  »  Sample ECommerce

For the relevant data source, select ➤ **Connection Settings** and one of the following options:
Change Provider: Selecting this option opens the first step of the Connection Wizard where you select the data source. In the Connection Wizard, click to select a data source connection type, for example, change MySQL to an Oracle database. When done, enter the connection credentials and select the relevant tables.

Change Connection: Selecting this option opens the second step of the Connection Wizard where you define the connection settings of the data source. In the Connection Wizard, click to edit the login details to the data source. When done, select the relevant tables.

Change Table: Selecting this option opens the third step of the Connection Wizard where you select the relevant database from a list of databases in your data source. In the Connection Wizard, select the relevant database. After you have selected the relevant database in the Connection Wizard, click Done. The connection settings are updated.
Working with Data

Once your data is imported into Sisense, Designers can organize the data in a variety of ways. One of the most common ways is to create a relationship between tables. After you create a relationship, the tables behave as a single table, which can make analyzing your data more easy and accurate.

The following topics describe how you can create relationships with your data and how to avoid some of the pitfalls of working with complex data:

- Navigating Sisense
- Finding Tables and Columns
- Creating and Removing Relationships
- Previewing Data in a Table
- Managing Tables and Columns
- Handling Relationship Cycles
- Many-to-Many Relationships
- Chasm and Fan Traps
Finding Tables and Columns

Through the ElastiCube Search tool, you can locate field tables and fields. The Search tool is displayed in the Navigation Pane of the ElastiCube Manager.

When you perform a search, all the relevant tables and columns are returned as you begin to type.

To organize the results displayed in the pane, toggle between the Data Source icon 🏛️ and the Tables icon ⌁.

Each result has an icon on the left side of it. These icons indicate what is being returned.

The meaning of each icon is as follows:

- 🏛️ Data source
- ⌁ Table
- 📜 Text data
- 🔍 Numeric data
- 📄 Date data
To find a table or field in your ElastiCube:

On the left-side of the schema, enter the table or field in the **Search** field. As you begin typing the letters or name of the table/field you are searching, the results are displayed below and the relevant table or field is highlighted in the schema.
Creating and Removing a Relationship between Tables

Relationships specify the connection between tables and enable you to pull data together in meaningful ways. For example, order information is more useful when you know which customer placed each order.

However, you don’t need to store both the customer and order information in the same table. Customer and order data can be stored in two related tables with a relationship specified between the two tables to view each order and its corresponding customer information.

In practical terms relationships allow you to query and combine data from multiple tables in your dashboard.

For a quick overview of joining data, see the video below.

To manually create a relationship between tables:
- Drag a table with the relevant field onto another table with the relevant field.
  The Relationship preview window is displayed.
- Click the columns to be joined from each table. Ensure both fields that you are using to create the relationship are of the same type, for example decimal fields.
  See the video below for a quick example.

To check or change the field type, in the left pane, click the field type of the field you want to change and select the new type from the list of possible
types displayed below it.

To delete an existing relationship:
Click on any part of the existing relationship. The Delete icon is displayed. Click to delete the relationship.
Previewing Data from a Table

The Preview table lets you see all the values for all the rows and columns in a table. This includes any custom columns, tables, and ETL processes that have been applied to your table.

The Preview table makes it easy to verify that the data in your table is accurate and complete.

In addition to displaying a preview of your data, from the Preview table, you can view which columns are connected to other tables and add custom columns.

**Note:** To view underlying table data, you will first need to build your ElastiCube.

**To review the underlying table data:**

Click the relevant table and select the magnifying glass icon to display the Preview table.

The Preview table contains a variety of information and functionality that relate to your table. A sample preview is displayed below.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Column Filter</td>
<td>In the left pane, you can search for columns in your table. As you begin to type, any relevant results are displayed in the left pane. This does not affect the results displayed in the preview.</td>
</tr>
<tr>
<td>2 Connected Table</td>
<td>This icon indicates that the column is connected to another table. Click the icon to disconnect the relationship or to view more information about the connection. See Creating and Removing a Relationship between Tables for more information.</td>
</tr>
<tr>
<td>3 Table Details</td>
<td>This information describes the number of columns and rows included in the table.</td>
</tr>
<tr>
<td>4 Menu Options</td>
<td>This is a list of tasks you can perform on a column. Hover over the column to display the menu and click it to display the options.</td>
</tr>
<tr>
<td>5 Filter Column</td>
<td>Click to apply filters to the data in the column. You can filter numeric data and text data. See Filtering Columns for more information.</td>
</tr>
<tr>
<td>6 Sort Column</td>
<td>Click to sort a column. Hover over the column name to display the sort icon. You can sort the column by date, numerically, or alphabetically depending on the type of column.</td>
</tr>
<tr>
<td>7 Add Custom Column</td>
<td>Click to add a custom column to the table. See Adding a New Custom Column for more information.</td>
</tr>
<tr>
<td>8 Sample Data</td>
<td>Click to see a sample of your data. See Sampling Data for more information.</td>
</tr>
</tbody>
</table>
Filtering Columns

You can filter data displayed in your Preview table’s columns by defining conditions for numeric data and text data. To limit what data is displayed in the preview, you can add multiple conditions that you define. For example, a Text filter can limit what strings are included in the preview.

Filtering the data in the Preview table does not affect the data itself, just what you see in the preview.

Filtering your preview is useful if you need to check specific data to verify its accuracy of your table’s completeness.

**To filter entire rows or columns of data from your preview:**

In the Preview table, hover over the header in the relevant column to display the Filter icon.

Click the Filter icon to define the filter’s conditions.

| Text     | Numeric |
Define your filters. Click **Add Condition** to add more conditions to the filter or click **Clear** to erase all the conditions. After you have defined your conditions, click **Apply** to update the preview.
Sampling Data

If you are working with a large dataset, you may want to restrict the amount of data displayed in your Preview table. Sisense provides three options for displaying a sample of your data:

**Top:** The first rows that are contained at the top of your data.

**Bottom:** The last rows that are contained at the bottom of your data.

**Sample:** A random selection of rows contained in your table. Sisense selects a sample of your data randomly by rows. If you open the preview again, the same rows will be displayed until your next ElastiCube build.

With each sampling option, you can determine how many rows are display according to the value of the **Number of Rows** field. You can modify this value to display as few as 10 rows or include as many as you like up to the full amount of rows included in the table.

**To view a sample of your data:**
In Sisense, open a preview of your table.

In the top-right corner of the Preview table, click the **Sampling** Icon.
Select the relevant sampling option for your data and enter the amount of rows to be included in the sample in the **Number of Rows** field.

Click **Done**. Your Preview table is updated.
Managing Tables and Columns

After you have imported your data into Sisense, it's time to shape it into a data model that can support your dashboard and provide accurate information to your Viewers.

You can modify your data on two levels: the Table level and the Column level. Any changes you make to your schema are reflected in the ElastiCube only and do not affect the data in your data source.

For a short video overview of managing the tables and columns in your data model, see below:
Table Level

You can manage tables imported into the ElastiCube through the Navigation Pane or through the table itself in the schema.

The table below describes the menu options available from the Navigation Pane and the corresponding icons available in the table’s menu. Either method allows you to modify and manage your tables.

<table>
<thead>
<tr>
<th>Navigation Pane</th>
<th>Table Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preview &amp; Edit</td>
<td>⚫️🔍</td>
<td>Displays a preview of your table’s data. See Previewing your Data for more information.</td>
</tr>
<tr>
<td>Relationships</td>
<td>⬤</td>
<td>Displays related tables together side by side. See Creating Relationships for more information.</td>
</tr>
<tr>
<td>Rename</td>
<td>⚫️🔍</td>
<td>Allows you to rename the table.</td>
</tr>
<tr>
<td>Navigation Pane</td>
<td>Table Icon</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Duplicate</td>
<td>![Duplicate Icon]</td>
<td>Allows you to duplicate the table.</td>
</tr>
<tr>
<td>Refresh Schema</td>
<td>![Refresh Schema Icon]</td>
<td>Allows you to update a table's columns if there was a change on the data source without having to add the table again.</td>
</tr>
<tr>
<td>Tags &amp; Description</td>
<td>![Tags &amp; Description Icon]</td>
<td>Allows you to tag a table with metadata that you can use to organize your data and search for it later. See <a href="#">Tagging Your Data</a> for more information.</td>
</tr>
<tr>
<td>Hide</td>
<td>![Hide Icon]</td>
<td>Allows you to hide a table. Hidden tables are not seen by Dashboard Viewers, but might be used for custom calculations.</td>
</tr>
<tr>
<td>Add Custom Column</td>
<td>![Add Custom Column Icon]</td>
<td>Allows you to add a custom column to the ElastiCube. See <a href="#">Adding Custom Columns</a> for more information.</td>
</tr>
<tr>
<td>Build Behavior</td>
<td>![Build Behavior Icon]</td>
<td>Allows you to define the build behavior for a specific table. See <a href="#">Build Settings and Data Accumulation Behavior</a> for more information.</td>
</tr>
<tr>
<td>Delete</td>
<td>![Delete Icon]</td>
<td>Allows you to delete a table.</td>
</tr>
<tr>
<td>Navigation Pane</td>
<td>Table Icon</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>--</td>
<td>&gt;</td>
<td>By default, tables are pinned when you import your data. You can click this icon to unpin your tables so your tables are organized by Sisense around other pinned tables.</td>
</tr>
</tbody>
</table>

To view your table’s meta, including its name, path, provider, location, and the number of columns it contains, hover over the table in the schema or in the Navigation Pane.
Column Level

You can manage tables imported into the ElastiCube through the Navigation Pane. Columns that are part of a table are displayed below the table. On the left side of the column is an icon that indicates what type of data is in the column, Numeric, Date, or Text. You can click this icon to convert the column into another data type.

On the right side of the column’s name is an icon that indicates if the column is conducted to another column through a relationship. You can hover over the icon to view more details or disconnect the relationship.

For each column, you can hover over its name to display a menu from where you view the following options for managing your columns:

<table>
<thead>
<tr>
<th>Menu Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td>Displays related columns together side by side. See Creating Relationships for more information.</td>
</tr>
<tr>
<td>Edit Relationship</td>
<td>Allows you to edit a column’s relationship. This option is only available when the column is connected to another table.</td>
</tr>
<tr>
<td>Rename</td>
<td>Allows you to rename the column.</td>
</tr>
<tr>
<td>Menu Option</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Duplicate</td>
<td>Allows you to duplicates the column in the table. This is useful if you need the same column with some modifications.</td>
</tr>
<tr>
<td>Hide</td>
<td>Allows you to hide a column. Click this option if you want to import the data associated with this column, but do not want this column available to dashboard designers and viewers.</td>
</tr>
<tr>
<td>Indexed</td>
<td>Allows you to index the column data for quicker processing in subsequent builds.</td>
</tr>
<tr>
<td>Drop</td>
<td>Allows you to remove column from the table. After dropping a column, you can view all your dropped columns by previewing the table. In Preview mode, you can restore the column by opening the column's menu and clicking Restore.</td>
</tr>
<tr>
<td>Change Data Type</td>
<td>Allows you to change the data type of your column in the ElastiCube. This does not change the column type in the original data source.</td>
</tr>
<tr>
<td>Description</td>
<td>Allows you to tag a column with metadata that you can use to organize your data and search for it later. See Tagging Your Data for more information.</td>
</tr>
<tr>
<td>Is Accumulated by</td>
<td>Allows you to enable accumulative behavior for date and integer fields in tables imported by the JDBC connector. See Enabling Accumulative Build for JDBC Connector for more information.</td>
</tr>
</tbody>
</table>
Creating Data Dictionaries

To provide you with greater flexibility and improved ease-of-use when modeling your data, Sisense allows you to tag your data and add descriptions. Tags and descriptions allow you to group tables and columns by defining metadata and providing the basis of a data dictionary without affecting your actual data. For example, you can tag several tables with a unique word or description, and then locate that group of columns through the Search field to see all the tagged tables across your schema regardless of which tables the columns belong to. While both are searchable, the difference between a tag and a description is that tags are associated with tables and descriptions are free text fields associated with columns or tables.

After you have tagged your data, you can easily locate your tagged columns and tables in the Search field by entering #, which displays a list of all your tags that you can select to filter your data.

To tag or add a description to your data:

For Tables:
  Open your data model.
In the Navigation Pane, click the relevant table’s menu and select **Tags & Description**.
In the **Tags** field, enter a tag and click + to add the tag. You can repeat this step to add multiple tags.

<table>
<thead>
<tr>
<th>Tags &amp; Description</th>
<th>×</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a Tag and Press Enter</td>
<td>+</td>
</tr>
</tbody>
</table>

Select a color for the tag. The color of a tag is to help you organize related tags.
Click **Done** to save your changes.

**For Columns:**
Open your data model.
In the Navigation Pane, click the relevant column’s menu and select **Description**.

In the **Description** field, enter a description for your column.

Click ✔️ to save your changes.
Importing and Exporting ElastiCube Data

**Note:** This feature is currently available in the desktop version of Sisense. This feature will be migrated to the web-based ElastiCube Manager in the near future. For more information, see [Sisense V7.0 Backwards Compatibility](#).

ElastiCube data can be transferred to different machines. This is useful when you are deploying on a new machine or need to transfer the data to a different server. The transferred data is stored in a compressed ecdata file. The file contains all of the information contained in an ElastiCubeData folder (including the source data), and is used by the Sisense Server Console to build a new ElastiCube.

**To export data:**
- Open the Sisense Server Console.
- Click on the relevant ElastiCube.
- Click **Stop** to stop the specific ElastiCube (not the entire server).
- Click **Export**. Enter a name for the file you are saving, and click **Save**.

![Sisense Server Console]

**To import data:**
- Open the Sisense Server Console.
- Click on the import ecdata file icon 📁.
- Select the location of the ecdata file and click **Open**.
A window will appear showing the import progress. Once complete, the new ElastiCubes will be added to the list of existing ElastiCubes. Make sure the ElastiCube is running before accessing it.

**Note:** You can also transfer data by copying the folder where Sisense stores the data locally, instead of exporting and importing the compressed file.
Handling Relationship Cycles

**Note:** The image on this page were taken in the desktop version of Sisense, however, the same principles described on this page also apply to the ElastiCube Manager.

Good relationships between tables are the key to pulling in data in logical ways. ElastiCubes are usually created from more than one table. To perform calculations on fields that reside in separate tables, you must define a relationship between the tables so the calculation mechanism knows how to navigate from one field to another.

When a single possible path exists between two fields, there is no issue. For example, performing a query involving the Customer Name and Amount field on the ElastiCube schema below can yield only one result, since there is only one possible path leading from the Customer Name field to the Amount field.

![ElastiCube schema diagram](image-url)
Relationship Cycles

In some cases, more than one path exists leading from one field to another. Sometimes this is due to poor database design, and other times it is just a realistic necessity due to the way the data is structured. In such cases, there is no absolute way to determine the required path to take to calculate results. For example, in the ElastiCube schema below, the Customer Name and Amount fields now have multiple paths between them:

- Customer —> Sales History (over the Customer ID field)
- Customer —> Commercial (over the Customer ID field) —> Sales History (over the Business ID field)
- Customer —> Private (over the Customer ID field) —> Sales History (over the Customer ID field)
Automatically Dealing with Relationship Cycles

Theoretically, any one of these possible paths could be the correct one in terms of the results you are trying to calculate. When the calculation engine encounters numerous possible paths, it picks the shortest path, which will have less impact performance-wise on query processing. Below are some general guidelines.

The calculation engine will always prefer paths that have no many-to-many relationships. If all possible paths contain many-to-many relationships, the path with the least number of this type of relationship is preferred.

Similarly, the engine will prefer one-to-one relationships over one-to-many relationships.

If at the end of the first selection process more than one possible path still exists, the engine will prefer the path containing the least number of tables.

Finally, if numerous paths are still possible, one is picked at random.
Breaking Relationship Cycles in Sisense

To ensure the calculation engine takes the path you want it to take, the cycle must be broken. There are a few techniques to do this depending on the required results.
Removing Relationships

One option is to remove one of the relationships in the cycle. For example, removing the Customer ID relationship from the Sales History table will eliminate the cycle. This approach is only applicable when the removed relationship is not used by other queries.
Duplicating Table Elements

Duplicating table elements is another method to break relationship cycles. In the example below, the Sales History table was duplicated and renamed Private Customers Sales History. In addition, the Customer ID relationship was removed from the original Sales History table. This results in the creation of two separate Amount fields, one for commercial customers residing in the original Sales History table, and one for private customers residing in the new Private Customers Sales History. As the Customer ID relationship was removed from the original table, whichever field you use will determine the correct calculation path.
Breaking Relationship Cycles in the Sisense Web Application

You can also use the Sisense web app to invoke a specific path to be used during query execution (as opposed to the solutions mentioned above that are defined on the schema level, in Sisense).

Each widget in Sisense has an associated data layout. If you have more than one path connecting corresponding fields between two tables, you can enforce a specific path by filtering the widget, and selecting just the field that you want to use. For example if the Commerce table is connected to Brands via the Brand and Brand ID fields, then add a filter to the widget and select Brand ID to use that path in the query.
Many-to-Many Relationships

**Note**: The image on this page were taken in the desktop version of Sisense, however, the same principles described on this page also apply to Sisense Online. In databases, a Many-to-Many relationship exists when the value in each field used to create a relationship between tables is included multiple times in each table.

The problem with Many-to-Many relationships is that it can cause duplications in the returned datasets, which can result in incorrect results and might consume excessive computing resources. This section provides solutions and workarounds to common scenarios with many-to-many relationships.

**Example**: A hotel may have a table with reservation data and a table with payment data. Both tables include the name of the guest. A guest can have multiple reservations under their name as well as multiple payments recorded on their name. If a relationship exists between the reservation and payment tables based on the guest’s name, a many-to-many relationship is created, as the guest’s name appears multiple times in each table.

**Summary of Relationship Types**

**One-to-One Relationship**: In this scenario both sides of the relationship have unique values for every row.

**One-to-Many Relationship**: In this scenario one side of the relationship will contain unique values for every row, but the other side of the relationship will contain duplicate values for any or all of the corresponding values in the first table.

**Many-to-Many Relationship**: In this scenario, both sides of the relationship will hold duplicated values, causing excessive calculations for every query run against it.

There are several methods to resolve and bypass a many-to-many relationship. The solution depends on the business model and the logic of the business...
questions at hand. The following solutions differ by business logic and the schema at hand; each solution can be applied to each schema respectively.

The following sections cover:
- Testing your schema to see if it includes many-to-many relationships
- Understanding which scenario best fits your current schema
- According to your schema logic, applying the respective solution

To check if a relationship is Many-to-Many, you need to check the cardinality of the relationship, and determine the number of unique and duplicate values on each side of the relationship.
Testing if a Relationship is Many-to-Many

Many-to-Many relationships occur when two tables are joined on a field containing duplicate values on both tables. For example the same guest may have multiple reservations and multiple payments at a hotel, thus joining on the guest between the reservation and payment table would result in a M2M relationship. When testing, if you get the same value for both the unique and duplicate values, then there is no duplication, and this will either be a One-to-Many or a One-to-One relationship. If the number of duplicate values is larger than the number of unique values, then this side of the relationship has duplicated values, and you will need to investigate the other side of the relationship. If the other side of the relationship yields unique values, this is a one-to-many relationship. If not, you have a many-to-many relationship.

Use the following SQL statement to test for potential M2M relationships:

In Sisense, open the relevant ecube file.
Click Add Data> Custom SQL Expression.
Enter and adjust the SQL statement below.

```
SELECT [Do I have duplications?] FROM (SELECT distinct_count(t1.col1)<count(t1.col1) AS [Do I have duplications?] FROM [Table1] t1 UNION all SELECT distinct_count(t2.col2)<count(t2.col2) FROM [Table2] t2) AS temp GROUP BY [Do I have duplications?]
```
In the top right of the expression editor window, click Parse SQL Expression. If the expression parses successfully, click Preview result table.

If the returned result is 'True' in both lines, a many-to-many relationship exists, and will need to be considered in the ElastiCube design.
Many-to-Many Relationship Prior to Resolution

If the two values are equal, all guest IDs appear only once, making all values unique. Even if the other side of the relationship has duplicate values for guest ID, this is still a One-To-Many relationship, where the unique values are on the reservations side, and the duplicate values are on the Payments side. If there are more than two tables connected to this relationship, that is, if there are more than two tables merged on the same field, a few more options exist. The solution for the single many-to-many relationship will be a sub-problem of this scenario. In this case, you’ll need to run the test on every table to see the uniqueness or duplication of the merged fields.

Two Tables with One Relationship

This section describes two possible workarounds when you have a schema that includes two tables with one relationship:
- Two Separate One-to-Many Relationships
- Creating an Aggregated Table

Two Separate One-to-Many Relationships

The direct solution for such a problem is to break this relationship into two separate one-to-many relationships.
You can use the following diagram to understand the logic behind this testing:

1 - Create a custom SQL expression in the Elasticube. In the expression of this table select all the individual values for the identifier column from both sides. The expression should look like this:
SELECT * FROM
(SELECT DISTINCT r.GuestID, r.GuestName
FROM [Reservations] r
UNION
SELECT DISTINCT p.GuestID, p.GuestName
FROM [Payments] p) AS G

This query will take all Guest ID values from both tables, and using the UNION statement, will bring in only the unique values from both tables, making this a complete list of all distinct Guest ID values.

2 – Merge the Guest ID field from the new ‘linking’ table to the other two Guest ID fields from the other two tables, thus creating two One-To-Many relationships. You can now use this Guest ID field as the rows or axes elements of a widget, pulling in the unique values from the new Guest Dimension, with measures from the two other tables. See image above.

Creating an Aggregated Table

In situations where you have more than one fact table (a Fact table is a primary table containing the measures or fields used for calculations in the dashboard) in the Elasticube, there are several situations when an aggregated table can resolve a many-to-many relationship.
Two fact tables

Assuming you want to segment your data according to a few different dimensions, creating relationships directly between these fields can and will create many-to-many relationships in one of two ways, according to the schema:

Both tables don’t hold unique values, and all values from one table are held in the second table. In this scenario either a linked dimension (as described in the first solution – Two Separate One-to-Many Relationships) or an aggregated table can be created which will hold all the unique values and the desired calculations for one of the tables. To create an aggregate table, create a custom SQL expression and aggregate values from the table that includes all values; its own, and the subset present in the other table with the following expression:
SELECT i.OrderDateKey, i.ProductKey, 
sum(i.DiscountAmount), sum(i.SalesAmount),
avg(i.UnitPriceDiscountPct)
FROM [FactInternetSales] i
GROUP BY i.OrderDateKey, i.ProductKey

This custom SQL expression will select the distinct OrderDateKeys and their corresponding ProductKeys from the FactInternetSales, grouped by these fields, together with single value aggregations for the different fields, in this case, Discount Amount, Sales Amount and the average unit Price discount. After merging the OrderDateKey and Product Key to the two other tables, you will be able to pull the values from this new table into the rows or axes panel of a widget in the Sisense Web Application with measures and additional aggregations from the two other tables.

**Note:** The non-aggregated table needs to be a subset in terms of the primary fields from the aggregated table.

Both tables don't include unique values, and there are different values for several fields in both the tables. Resolving this scenario incorporates both solutions mentioned above. In this scenario, create an aggregated table and a dimension table (both described above). The final resolution should look like this:
Two Fact tables with a date dimension table and an aggregative Products table

More than Two Tables with More than One Relationship

This section provides two possible workarounds when you have a schema that includes more than two tables with more than one relationship:

Options include:
- Using the Lookup function
- Concatenating two tables into one

Using the Lookup Function

In most scenarios, you will aggregate values according to a given ID from the unique side of the relationship to the duplicate side. However in specific cases it'll be vice versa.
For example in the following scenario, in which we have three tables, and between them two one-to-many relationships, this can potentially create a many-to-many relationship, if you query the two leaf tables. This means that the query result table will have multiple rows which won’t be distinguishable one from the other.

**Two consecutive M-to-M relationships**

Using the Lookup Function, you can import values from a remote table by matching values in a different column. This will create a new column in the table where you want to perform an aggregation of a given field(s), with the matching value of the identifying field from the other table. Taking the example of tables T1, T2 and T3, we’d like to run a query that will display aggregations from the duplicate IDs from T1, with a measure from T3. If you run the query as is, you will get multiple values for the query’s result set, and we will not be able to run this aggregation. To resolve this, use the Lookup function to import the values from T3 into T2 and then re-run the query only on tables T1 and T2. Using the lookup function, available in the ‘Miscellaneous Functions’ in the custom SQL editor, you can import the values of ‘M3’ from the ‘T3’ table into the ‘T2’ table. Create a new custom column, and use the Lookup function to import the values of attribute. In this case, the Lookup function should look like this:

```
Lookup([T3],[T3].[M3], [T2].id2,[T3].id2)
```

Running this statement in table T2 will import the matching values of M3 from T3 according to the matching results in ID2 between the two tables.

```
LOOKUP(remote_table,remote_result_column,current_match_column, remote_match_column)
```
Matches the current value with another value from a remote table. The result will be the value in remote_result_column for which the corresponding remote_match_column equals the current_match_column.

Two consecutive M-to-O relationships after Lookup fix

Concatenating Two Tables into One

Assuming you have two separate tables with duplicate ID values in each, and each including different columns for each ID, you can create a new table including all values for every ID, and pull the aggregations from this new table. Note that the two original tables; Table_1, Table_2 have different columns.

Concatenating tables

Using the following SQL statement, you can import the data from both tables, with the IDs and the columns respectively:

```
SELECT s.id AS id, s.m1, s.m2, ToInt( NULL ) m3 , ToInt( NULL ) m4
FROM [Table 1] s
UNION
```
SELECT t.id, ToInt( NULL ), ToInt( NULL ), t.m3, t.m4
FROM [Table 2] t
This will create a table with five columns:
Id
M1 (from table_1)
M2 (from table_1)
M3 (from table_2)
M4 (from table_2)

The values missing from each table respectively will be NULL’s which will result in the following table.
### Concatenated table – result set

#### Reservations
- **GuestID**
- **GuestName**
- attribute 1
- attribute 2
- attribute 3

#### Payments
- **GuestID**
- **GuestName**
- attribute 2
- attribute 3
- attribute 4

```sql
SELECT s.id AS id, s.m1, s.m2, ToInt( NULL ) m3 , ToInt( NULL ) m4
FROM [Sheet1] s
UNION
SELECT t.id, ToInt( NULL ), ToInt NULL , t.m3, t.m4
FROM [Sheet2] t
```

<table>
<thead>
<tr>
<th>id</th>
<th>m1</th>
<th>m2</th>
<th>m3</th>
<th>m4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>98</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>23</td>
<td>76</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>34</td>
<td>67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>32</td>
<td>87</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>56</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>43</td>
<td>76</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>23</td>
<td>76</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>43</td>
<td>45</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>654</td>
<td>65</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>46</td>
<td>34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>87</td>
<td>87</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>242</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

12 Rows
Chasm and Fan Traps

Note: The image on this page were taken in the desktop version of Sisense, however, the same principles described on this page also apply to the web-based ElastiCube Manager. Chasm and fan traps should be avoided when building your ElastiCube schemas.
Chasm Traps

A chasm trap occurs when two “Many-to-One” joins converge on a single table, and the query includes measures from both leaf tables. As a result multiple rows are returned from the tables when processing the query.

If you were to calculate both measures (Qty and Value) simultaneously, like in the following example, the values for Customers will be multiplied due to the inner join between the leaf tables, and the results may be incorrect:

```
SELECT   [Customers].CustomerName,
         sum ([Orders].OrderValue) AS Value,
         sum ([Sales].QuantitySold) AS Qty
FROM      [Customers]
JOIN      [Sales] ON [Customers].CustomerID=[Sales].CustomerID
GROUP BY  [Customers].CustomerName
```

<table>
<thead>
<tr>
<th>CustomerName</th>
<th>Value</th>
<th>Qty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jason</td>
<td>1800</td>
<td>180</td>
</tr>
<tr>
<td>Daniel</td>
<td>600</td>
<td>60</td>
</tr>
</tbody>
</table>
**Fan Traps**

A fan trap occurs when two “many-to-one” joins follow one another in master-detail form (OrderDetails), and the query includes a measure from both the leaf table (OrderDetails) and its immediate master (Orders).

If you try to aggregate both measures simultaneously (using the query below), you will probably get incorrect results:

```sql
SELECT [Customers].CustomerName, 
       sum ([Orders].OrderValue) AS Value, 
       sum ([OrderDetails].OrderQuantity) AS Qty 
join 
[OrderDetails] ON [Orders].OrderId=[OrderDetails].OrderId 
GROUP BY
```

<table>
<thead>
<tr>
<th>CustomerName</th>
<th>Value</th>
<th>Qty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jason</td>
<td>700</td>
<td>90</td>
</tr>
<tr>
<td>Daniel</td>
<td>400</td>
<td>60</td>
</tr>
<tr>
<td>Mike</td>
<td>400</td>
<td>60</td>
</tr>
</tbody>
</table>

The “Qty” measure, corresponding to the leaf measure table (OrderDetails) is calculated correctly, but the “Value” measure, corresponding to the measure held in its master (Orders), is not. This is because we get the “Value” of every OrderID, which may inflate the expected results.
The web application translation module separates the calculations, and unicons the results by generating a query for each of the measures’ paths (path in terms of tables to go by). Then the web application translation module will group all the measures with the same tables’ paths into one query and union it with all the other “same path measures” with different paths.

The described “Chasm Trap” can be prevented like this:

```sql
SELECT [Customers].CustomerName, sum ([Orders].OrderValue) AS Value, 0 AS Qty 
GROUP BY [Customers].CustomerName 
union 
SELECT [Customers].CustomerName, 0 AS Value, sum ([Sales].QuantitySold) AS Qty 
FROM [Customers] JOIN [Sales] ON [Customers].CustomerID=[Sales].CustomerID 
GROUP BY [Customers].CustomerName
```

<table>
<thead>
<tr>
<th>Output</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>CustomerName</td>
<td>Value</td>
</tr>
<tr>
<td>Jason</td>
<td>600</td>
</tr>
<tr>
<td>Daniel</td>
<td>300</td>
</tr>
<tr>
<td>Mike</td>
<td>300</td>
</tr>
<tr>
<td>Jason</td>
<td>0</td>
</tr>
<tr>
<td>Daniel</td>
<td>0</td>
</tr>
</tbody>
</table>

And the “Fan Trap” will be prevented this way:

```sql
SELECT [Customers].CustomerName, 0 AS Value, sum ([OrderDetails].OrderQuantity) AS Qty 
JOIN [OrderDetails] ON [Orders].OrderId=[OrderDetails].OrderId 
GROUP BY [Customers].CustomerName 
union 
SELECT [Customers].CustomerName, sum ([Orders].OrderValue) AS Value, 0 AS Qty 
FROM [Orders] 
JOIN [Customers] ON [Orders].CustomerID=[Customers].CustomerID 
GROUP BY [Customers].CustomerName
```

<table>
<thead>
<tr>
<th>Output</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>CustomerName</td>
<td>Value</td>
</tr>
<tr>
<td>Jason</td>
<td>0</td>
</tr>
<tr>
<td>Daniel</td>
<td>0</td>
</tr>
<tr>
<td>Mike</td>
<td>0</td>
</tr>
<tr>
<td>Jason</td>
<td>600</td>
</tr>
<tr>
<td>Daniel</td>
<td>300</td>
</tr>
<tr>
<td>Mike</td>
<td>300</td>
</tr>
</tbody>
</table>
Handling Ragged and Unbalanced Data Hierarchies

BI tools are designed to ingest data organized in standard hierarchies: hierarchies that have a field on each level of the hierarchy. Thus, complex data hierarchies, such as ragged and unbalanced hierarchies, can be difficult to model and query. If your data is organized in such a hierarchy, you will have to resolve it, meaning to get it into a form that BI tools can ingest, before you can use it in a dashboard.

Depending on the structure of the data, Sisense offers several solutions for resolving complex hierarchies. These solutions range from simple table modelling using a dashboard hierarchy to transforming the data with SQL. This page focuses on several specific types of complex hierarchies. Use the approach delineated below to resolve complex hierarchies.
Types of Data Hierarchies

A *balanced hierarchy* has an equal number of levels in each branch and is easiest to handle. The example shown below is a natural hierarchy of years > quarters > months.

![Balanced Hierarchy](image)

An *unbalanced hierarchy* has at least one branch which does not reach down to the lowest level. The below example of a company org chart is typical of an unbalanced hierarchy. Some divisions have more levels than others.

![Unbalanced Hierarchy](image)

A *ragged hierarchy* is characterized by having entities whose parents 'skip a level'. The below screenshot exemplifies this. The city Washington DC and the Greek cities of Athens and Thessaloniki roll directly to the Country instead of to a State.
Hierarchies can also be both *ragged and unbalanced*, adding an additional level of complexity.
The Scenario

In the below scenario, data hierarchy is represented by the following diagram. Note that it is both unbalanced (some branches do not reach down to the lowest level) and ragged (some branches skip a level).

We start off with a hierarchy defined recursively, as shown below:
The Solution

To address this scenario, in Sisense you need to create placeholder values where a level is skipped in a ragged hierarchy. To start with, you create a new table to identify skipped entities. A new table 'Hierarchy1' has been created for this purpose using the SQL below. For simplicity, numbers are hard-coded to correspond to the levels, and only single levels are skipped. This SQL can be easily modified for other scenarios using the same core logic.

Next you need to consolidate all the branching entities that roll to the same placeholder parent to the same entity.
You then need to union the placeholder values back into the main hierarchy table, and join and replace the appropriate parent IDs. This can be done in either order, but the screenshot below shows the union, then the join.

The non-ragged hierarchy now logically looks as below. It is ready now to resolve using a standard flattening approach. The Entity Name field is used in place of IDs for simplicity.
The SQL to do so looks like the following.

```
SELECT ha.[Entity Name] AS VP, hb.[Entity Name] AS Team,
     hc.[Entity Name] AS Director, hd.[Entity Name] AS [Team Lead], he.[Entity Name] AS Rep
FROM [Hierarchy3] ha
LEFT JOIN [Hierarchy3] hb ON ha.[ID] = hb.[Parent]
LEFT JOIN [Hierarchy3] hc ON hb.[ID] = hc.[Parent]
LEFT JOIN [Hierarchy3] hd ON hc.[ID] = hd.[Parent]
LEFT JOIN [Hierarchy3] he ON hd.[ID] = he.[Parent]
WHERE ha.[Level] = 'VP'
```

Note the N/As in the table. These account for the unbalanced branches where the placeholder values account for the ragged hierarchy. Now it is a simple matter of mapping the new resolved hierarchy table back into the data model. If the facts all roll to the lowest level of the hierarchy, the above logic and resulting table will work as-is. If the facts need to be attributed across different levels of the hierarchy, you would split the structure into different tables. The resolved parent IDs can be used to link
across the hierarchy (see example below).

Roll-ups should now work seamlessly when using these new dimensions, and this should be easily consumed on a dashboard.
Overview

When you build a data model, Sisense displays an intuitive build log that describes each step of the build.

Build logs are useful for investigating problems that occurred during your build. The log displays each table being built as its being built. You can click the table’s name in the log to zoom in on the table in your schema. If a table fails to build, you can click its name in the log and then open the table within your data model to investigate potential causes for the failure.

**Note:** Your build log is available after your first build in Sisense V7.3 and later. Once your build is complete or if the build failed, you can export a detailed log file by clicking in the build log.
The exported log is a JSON file that provides details about each table. For information about understanding the log file, see Working with Build Logs. In addition, you can create a new dashboard from the build log or open any existing dashboards related to the data model.

**Build Process** 6 / 6 Tasks Completed  |  00:00:45

---

- **Build Succeeded**  
  October 09, 2018 09:11

  + Create Dashboard

- 00:00:08  
  Finalizing build  
  Finalization Completed

- 00:00:02  
  **Country** 199 rows  
  Successfully ended

- 00:00:14  
  **Commerce** 613,002 rows  
  Successfully ended
Working with Build Logs

If you need to troubleshoot an issue with a build, you can download the log from the build log.

The build log is a JSON file that contains build objects. The build objects contain key:value pairs that describe the build process for each table.

As the reasons for build failures vary and may be quite complex, there isn’t a single solution for resolving build errors. If you are not able to solve the issue yourself, save your logs and provide them to Sisense Support for additional assistance.

The table below describes some of the keys that might be included in your log and their possible values.
<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timestamp</td>
<td>The date and time the table was built.</td>
</tr>
<tr>
<td>verbosity</td>
<td>The type of information provided, such as Info, Warning, or Error. Info indicates that the build was performed. Warning indicates that something unexpected occurred, but the build was able to proceed. Error indicates that an error occurred and the build could not be continued.</td>
</tr>
<tr>
<td>type</td>
<td>The type of action being preformed by Sisense. There are a variety of actions that may be performed, some are relevant to the table, others to Sisense itself.</td>
</tr>
<tr>
<td>title</td>
<td>The title of the action being performed.</td>
</tr>
<tr>
<td>serverId</td>
<td>Identifier for the server where the build was performed.</td>
</tr>
<tr>
<td>serverName</td>
<td>The name of the server where the build was performed.</td>
</tr>
<tr>
<td>cubeId</td>
<td>The name of the ElastiCube.</td>
</tr>
<tr>
<td>buildSeq</td>
<td>The place in the order of build operations.</td>
</tr>
<tr>
<td>typeValue</td>
<td>This is a generic object that contains information about the build process. The keys in this object vary depending on the build stage.</td>
</tr>
<tr>
<td>dbfarmpath</td>
<td>The location of the ElastiCube on the server.</td>
</tr>
</tbody>
</table>
Transforming and Enriching Data

After you have imported data from your data sources, Data Designers can transform and enrich data in the ElastiCubes. Data Designers can prepare easy-to-use data sets for Dashboard Designers. Sisense provides several features that make it easy to prepare, blend, and analyze data that can be fed to the Sisense Web Application and consumed by Viewers.

The following topics describe how you can transform and enrich your data in Sisense:

Related Topics
- Adding a Custom Column
- Editing a Custom Field
- Adding a Custom Table
- Editing a Custom Table
- Previewing Results
- SQL Reference
- Function Reference
Augmented Text Deduplication

As your data comes from a variety of sources, many times you need to aggregate the same dimension that is represented differently in each source. In some sources, an attribute can be spelled one way, and in another source, a different spelling. In some cases, attributes may not have been spelled correctly when they were added. One way to deal with this is with custom SQL CASE statements, for example:

```sql
case [Name]
when 'John Smith' then 'J. Smith'
end
```

```sql
case [Country]
when 'Afrika' then 'Africa'
end
```

However, this can be very time consuming and lead to mistakes. Leveraging machine learning, Sisense can analyze your data and group similar dimensions automatically for you.

Sisense analyzes and groups your data from your ElastiCube on the column level. This means that Sisense looks at each column and identifies similar strings, and groups them together into a single attribute.

<table>
<thead>
<tr>
<th>Actual</th>
<th>Desired</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country</td>
<td>Sales</td>
</tr>
<tr>
<td>USA</td>
<td>100</td>
</tr>
<tr>
<td>Franse</td>
<td>100</td>
</tr>
<tr>
<td>United States</td>
<td>200</td>
</tr>
<tr>
<td>US</td>
<td>300</td>
</tr>
<tr>
<td>America</td>
<td>100</td>
</tr>
<tr>
<td>France</td>
<td>50</td>
</tr>
<tr>
<td>FR</td>
<td>200</td>
</tr>
<tr>
<td>Country</td>
<td>Sales</td>
</tr>
<tr>
<td>United States</td>
<td>700</td>
</tr>
<tr>
<td>France</td>
<td>350</td>
</tr>
</tbody>
</table>
All the group's attributes are then added to your data model as a custom table that you can use in your dashboard.
How Does it Work?

After you have imported your data and built your ElastiCube, a list of columns are displayed. For each column with a string data type, you can have Sisense analyze that column and group by similar attributes. The first time Sisense analyzes the column, you choose how Sisense analyzes it and how inclusive the results should be.

Sisense has three methods for analyzing data:

**Similarity:** Sisense applies a soft TFIDF (term frequency–inverse document frequency) algorithm where similar strings are grouped together by rank. Terms that appear too often, such as company and inc are scored lower as they are less unique.

**Spelling:** Sisense groups strings similar in spelling.

**Phonetic Similarity:** Sisense groups strings terms that sound similar, for example, strings that use ‘f’ or ‘ph’.

**Recommendation:** Sisense recommends you test each method, and find the results that work best for you.

Next, you select how inclusive the results are. The default is less inclusive, which means it’s more precise. The more inclusive the results, the less precision there
is. If your results are too inclusive, you may need to spend time editing the members in your group.

Less Inclusive
More Inclusive
Most Inclusive

After analyzing the data according to your settings, Sisense creates a set of groups of similar attributes.

On the left side, are the **Group Members**. This is a list of members that have been grouped together according to the algorithm. On the right side is the **Group Name**, or how they should be named. If you think of this as a CASE statement, the members on the left are the conditions, when 'John Smith', and the **Group Name** on the right is the result, then 'J. Smith'. The group name determines how all the members should be called.

Sisense provides these recommendations based on the algorithm you selected, however, you can move members to different groups. For more information, see [Modifying Groups](#).

Once you are satisfied with your groups, click **Done**, and Sisense adds a custom table to your data model. Keep in mind that the new table isn’t included until your next successful build.
Creating Custom Columns with Grouped Data

If you have already built your ElastiCube, you can create custom columns from data grouped together by Sisense.

**To create custom columns:**
Create or open an ElastiCube model. The model must be built before your data can be grouped. If you have not run a build, build the model.
In Table view, click the table that contains the column whose data should be grouped.
For the relevant column, open the column’s menu and click **Group Similar (Dedupe)**.

Click **Analyze Data** and Sisense will analyze the data and display a list of groups. The method Sisense uses by default is to group similar strings and make it less inclusive.

**OR**

Click **Change Grouping Method** to modify how Sisense groups your data. After you have selected the relevant method, click **Analyze Data**.
A list of groups and their members are displayed. Modify the group as required and click Done. A new custom table is added to your data model.
Modifying Groups

Before creating your custom columns, you can modify Sisense’s recommended groups manually. The Group Similar window contains a variety of options for modifying your groups.

Enter the name of a group or its members to filter the list of members displayed.
Click to sort the list of groups by size or alphabetically.
Select which members are to be grouped when adding to your custom columns. By default, all groups are selected.
Click the Group Name to rename it, replace it with another group, or merge the group with another group. When merging with another group, the members of the first group are added to the group you select. When the column is created, all members receive the group name of the group they are merged to.
Click a member to move it to another group or create a new group from that member.
Editing Grouped Custom Columns

After you have created your custom columns, you can edit your groups. Any changes you make to the groups affect the data displayed in your dashboards after your next build.

From your ElastiCube model, you can edit the groups by selecting the relevant table and click **Group Similar (Edit)**.

**Note:** If you edit the table by clicking **Edit & Preview**, any changes you make are overwritten the next time group your data.

After you click **Group Similar (Edit)**, the Group Similar window is displayed with a list of your groups. You can now modify the groups as necessary. For more
information, see Editing Grouped Custom Columns. As you make your changes, new values are highlighted in yellow and you can sort
Limitations

Sisense supports up to 1000 unique values based on the largest groups. Custom columns can be migrated with your ElastiCubes, but if you want to make any changes after migrating them, you must regroup your data from scratch. Sisense supports grouping data on tables up to 10 million rows.
Adding a Custom Column

Sisense makes it easy for you to add new columns to existing tables. This can be useful if you need to combine data from different existing columns, and when you need to cleanse and prepare data. The new columns also provide Designers with additional fields to use as is, or as a basis for even more advanced calculations in their widgets. You can use SQL to customize the values contained within the custom columns.

For a short video overview of adding a custom column to your data model, see below:

**To add a new column:**
In the Navigation Pane, select :> Add Custom Column.

OR
In the ElastiCube model, select the table and click ⋮ and Add Custom Column.
The new column will be added to the table, and an SQL Editor is displayed where you define your custom SQL expression.

In the **New Custom Column** field, enter the name of your custom column and click ✅.

In the SQL Editor, enter your custom expression.

Click ✅ to add your column to the table.
Custom Field Use Cases

Here are some typical scenarios for creating custom fields.

**Row Level Calculations**
Calculate revenue from your sales data.

\[
\text{Gross Revenue} = (\text{Unit Price} \times \text{QuantitySold})
\]
\[
\text{Net Revenue} = (\text{Unit Price} \times \text{QuantitySold}) - (\text{Unit Cost} \times \text{QuantityPurchased})
\]

**Time Difference Calculations**
Work out the arrival time based on due and actual arrival dates.

\[
\text{daydiff (DueDate,ReceivedAt)}
\]

**Relative Difference Calculations**
Figure out the relative difference between your price and competitor price.

\[
\frac{(\text{CompetitorPrice} - \text{MyPrice})}{\left(\frac{\text{CompetitorPrice} + \text{MyPrice}}{2}\right)}
\]

**Buckets**
Create data buckets according to your required business break points.

\[
\text{CASE}
\begin{align*}
\text{WHEN } & \text{[PriceVsCompetitor%]} < -0.05 \text{ THEN '}-5\%' \\
\text{WHEN } & \text{[PriceVsCompetitor%]} \geq -0.05 \text{ AND} \\
& \text{[PriceVsAvgCompetitor%]} \leq 0.05 \text{ THEN '}-5\%/+5\%' \\
\text{WHEN } & \text{[PriceVsCompetitor%]} > 0.05 \text{ THEN '+5%'}
\end{align*}
\text{end}
\]

**Attributes**
Create attributes to translate data talk to more coherent categories.

\[
\text{CASE}
\begin{align*}
\text{WHEN } \text{Region} = 1 \text{ THEN 'USA'} \\
\text{WHEN } \text{Region} = 2 \text{ THEN 'EUR'} \\
\text{WHEN } \text{Region} = 3 \text{ THEN 'ASIA'}
\end{align*}
\]
Date Conversions
Convert dates into a numeric representation.
\[
\text{getyear(Date)} \times 10000 + \text{getmonth(Date)} \times 100 + \text{getday(Date)} \text{ AS DateNum}
\]
Convert text into dates.
\[
\text{createdate( toint('20'+RIGHT([Timesheet_Date],2)),}
\text{toint(LEFT(Timesheet_Date,2)),}
\text{toint(StrBetween(Timesheet_Date,'/','/'))}
\]

Date Period Comparisons Over Time
Create month to date and year to date calculations.
Click here to read our support article.

Surrogate Keys
\[
\text{EmployeeID+toString([DateNum])+toString(CustomerID)+toString([Project_ID])}
\]

Importing from Another Table
Use the lookup function to import a column from a different table. See "Adding a Custom Table" on page 253 to read more.
Editing Field Formulas

Sisense makes it easy to edit custom tables. You can use SQL to customize the fields contained within a custom table.

When you edit a custom table, any changes you make are saved when you click **Save** unless the expression did not successfully parse. In this case, you are prompted to save a draft of the invalid expression.

If you save the invalid expression, Sisense continues to use the last valid expression that was saved when building an ElastiCube, but the next time you open a preview of the table, the invalid expression will be displayed. This allows you to continue building working ElastiCubes while letting you save incomplete expressions for later.

If you wish to display the last valid expression, you can click the **Refresh** button in the menu bar and then press **Save**.

**To edit a custom table’s SQL expression:**
Select the custom table you want to edit and click 📊.

The custom table area is displayed
Enter SQL statements to access tables and fields that exist in the schema. See related topics for more details on SQL and Function References.

To view results based on the SQL statement, click 📊 Preview.

**Note:** At least one build including the relevant base table must have been completed in order to preview results.
Click **Save** to save your changes.
Adding a Custom Table

With Sisense, you can easily create custom tables that can combine existing data. This is useful for preparing and cleansing data. Follow the steps below to learn how to add custom tables using SQL.

For a short video overview of adding a custom table to your data model using SQL, see below:

To create a custom table:

In Sisense, open the schema of the ElastiCube you want to add to a custom table to.

In the menu bar of the schema, click + Custom. The New Custom Table area is displayed.

Enter the name of the custom table, and click ✓. Enter SQL statements to access tables and fields that exist in Sisense schema.

See also "SQL Reference" on page 258 and "Functions Overview" on page 324.

To view results based on the SQL statement, click Preview.

Note: You can save an incomplete or invalid SQL expression, however when you build the ElastiCube, only the last valid expression will be included. If you have not built an ElastiCube with a valid expression, then the invalid or incomplete expression will not be saved. Click Save to save your changes.

Tip

Press Ctrl + Space to complete an SQL statement.

To make it easier to access table fields, give the table name an alias and reference the alias to bring up the related fields. For example SELECT A. FROM A.

You can add comments in expressions by using the ‘–’ for a single line or /.../ for blocks.
Editing SQL Statements

Sisense makes it easy to edit custom tables. You can use SQL to customize the columns contained within a custom table.

When you edit a custom table, any changes you make are saved when you click Save unless the expression did not successfully parse. In this case, you are prompted to save a draft of the invalid expression.

If you save the invalid expression, Sisense continues to use the last valid expression that was saved when building an ElastiCube, but the next time you open a preview of the table, the invalid expression will be displayed. This allows you to continue building working ElastiCubes while letting you save incomplete expressions for later.

If you wish to display the last valid expression, you can click the Refresh button in the menu bar and then press Save.

To edit a custom table’s SQL:
Select the custom table you want to edit and click 🌐.

The custom table area is displayed.
Enter SQL statements to access tables and fields that exist in the schema. See related topics for more details on SQL References and Function References.

To view results based on the SQL statement, click 🔄 Preview. (Note: At least one build including the relevant base table must have been completed in order to preview results.)
Click Save to save your changes.

Tips
Press Ctrl + Space to complete an SQL statement.
To make it easier to access table fields, give the table name an alias, and reference the alias to bring up the related fields. For example Select A. From table as A.
You can add comments in expressions by using the ‘–’ for a single line or /.../ for blocks.
A good reference on compatible SQL commands is available on Microsoft MSDN. Note: Sisense does not support boolean data types.
Previewing Results

You can preview results as follows:

**Note:** If the results are not displayed for a custom query, this may indicate you have a many-to-many relationship in your data model. Review your model for any problems.

**To view results before connecting to a table:**

In the schema of your ElastiCube, click **Add Data**. Select the relevant connector. Enter your login details and connect to the source data. All tables and views associated with the database will appear in a new window. To preview data contained in a particular table, click **Preview Table**.
To view results in a custom table:
In the schema of your ElastiCube, click the custom table that you want to preview and click 🔄.

Note: At least one build including the relevant base table must be completed in order to preview results.

To view results in a custom field:
In the left pane of your schema, select the custom field and click Edit.
You can use SQL to create new tables and fields in Sisense. Below is an example of an SQL query with some SQL main clauses:

<table>
<thead>
<tr>
<th>SQL Structure</th>
<th>SQL Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT</td>
<td>SELECT CustomerLastName,</td>
</tr>
<tr>
<td>FieldName(s),Function()</td>
<td>CustomerName, ContactEmail,</td>
</tr>
<tr>
<td>, *</td>
<td>Count(OrderID) AS Counter</td>
</tr>
<tr>
<td>FROM TableName 1</td>
<td>FROM Customers c</td>
</tr>
<tr>
<td>INNER JOIN LEFT</td>
<td>INNER JOIN Order o</td>
</tr>
<tr>
<td>JOIN RIGHT JOIN FULL</td>
<td>ON c.CustomerID = o.CustomerID</td>
</tr>
<tr>
<td>JOIN TableName 2</td>
<td>WHERE o.OrderId BETWEEN 10 AND 100 AND</td>
</tr>
<tr>
<td>ON TableName 1.</td>
<td>(c.CustomerName IN</td>
</tr>
<tr>
<td>JoinField = TableName 2. JoinField</td>
<td>('John','Mary','David') OR</td>
</tr>
<tr>
<td>WHERE FieldName</td>
<td>c.CustomerLastName LIKE</td>
</tr>
<tr>
<td>Condition</td>
<td>'Harrison%'</td>
</tr>
<tr>
<td>AND/OR FieldName Condition</td>
<td>GROUP BY c.CustomerLastName,</td>
</tr>
<tr>
<td>Condition</td>
<td>c.CustomerName, c.ContactEmail</td>
</tr>
<tr>
<td>GROUP BY FieldName(s)</td>
<td>HAVING Count(o.OrderID) &gt; 3</td>
</tr>
<tr>
<td>HAVING FieldName(s)</td>
<td>ORDER BY c.CustomerLastName,</td>
</tr>
<tr>
<td>Condition</td>
<td>c.CustomerName</td>
</tr>
<tr>
<td>ORDER BY FieldName(s)</td>
<td></td>
</tr>
</tbody>
</table>
Basic SQL Syntax Guide

### SELECT STATEMENT

| SELECT | SELECT column_name(s) 
FROM table_name |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT *</td>
<td>SELECT * FROM table_name</td>
</tr>
</tbody>
</table>
| SELECT DISTINCT | SELECT DISTINCT column_name(s) 
FROM table_name |

Note: Nested SELECT statements will not work with an AND clause in Sisense.

### FUNCTIONS AFTER SELECT CLAUSE

<table>
<thead>
<tr>
<th>FUNCTIONS</th>
<th>AVG() – Returns the average value</th>
<th>COUNT() – Returns the number of rows</th>
<th>MAX() – Returns the largest value</th>
<th>MIN() – Returns the smallest value</th>
<th>SUM() – Returns the sum</th>
</tr>
</thead>
</table>

### TABLE JOIN FUNCTIONS

| INNER JOIN | SELECT column_name(s) 
FROM table_name1 T1 
INNER JOIN table_name2 T2 
ON T1.column_name=T2.column_name |
|------------|--------------------------------|
| LEFT JOIN | SELECT column_name(s) 
FROM table_name1 T1 
LEFT JOIN table_name2 T2 
ON T1.column_name=T2.column_name |
| RIGHT JOIN | SELECT column_name(s) |
FROM table_name1 T1
RIGHT JOIN table_name2 T2
ON T1.column_name = T2.column_name

FULL JOIN
SELECT column_name(s)
FROM table_name1 T1
FULL JOIN table_name2 T2
ON T1.column_name = T2.column_name

AS (alias)
SELECT column_name AS column_alias
FROM table_nameor
SELECT column_name
FROM table_name AS table_alias

WHERE
SELECT column_name(s)
FROM table_name
WHERE column_name operator value

FUNCTIONS AFTER THE WHERE CLAUSE

<table>
<thead>
<tr>
<th>AND / OR</th>
<th>SELECT column_name(s) FROM table_name WHERE condition AND</th>
<th>OR condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BETWEEN</td>
<td>SELECT column_name(s) FROM table_name WHERE column_name BETWEEN value1 AND value2</td>
<td></td>
</tr>
<tr>
<td>IN</td>
<td>SELECT column_name(s) FROM table_name WHERE column_name IN</td>
<td></td>
</tr>
</tbody>
</table>
### IN

- `(value1, value2, ...)`

### LIKE

- `SELECT column_name(s)`
- `FROM table_name`
- `WHERE column_name LIKE pattern`

---

### ORDERING AND GROUPING AFTER WHERE CONDITIONS

#### GROUP BY

- `SELECT column_name, aggregate_function(column_name)`
- `FROM table_name`
- `WHERE column_name operator value`
- `GROUP BY column_name`

#### ORDER BY

- `SELECT column_name(s)`
- `FROM table_name`
- `ORDER BY column_name [ASC | DESC]`

#### HAVING

- `SELECT column_name, aggregate_function(column_name)`
- `FROM table_name`
- `WHERE column_name operator value`
- `GROUP BY column_name`
- `HAVING aggregate_function(column_name) operator value`

---

### CLAUSES TO COMBINE TABLES

#### UNION

- `SELECT column_name(s) FROM table_name1`
- `UNION`
- `SELECT column_name(s) FROM table_name2`
<table>
<thead>
<tr>
<th>UNION ALL</th>
<th>SELECT column_name(s) FROM table_name1</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNION ALL</td>
<td>SELECT column_name(s) FROM table_name2</td>
</tr>
</tbody>
</table>
See Also

Basic SQL Guide
MSDN SQL Reference
ElastiCube for Advanced Business Scenarios

This section provides examples of more advanced scenarios, and the recommended methods for implementing the required business logic. The examples are categorized into the following data manipulation methods: Integrating, Formatting and Enhancing.
Integrating Data

Integrate and merge data from different sources into a single ElastiCube structure by identifying common keys between the different tables. Proper planning is important for merging the data; on the one hand, you need to avoid creating unnecessary relationships, and on the other hand, make sure you don’t have any many to many relationships. Examples include:

Creating a Common Date Selection: Create a common date field from multiple date sets (from multiple data sources), and still keep the ability to use each original date field individually.

Financial Reporting: Bring in an additional data source to help analyze data from transaction systems. For example, Financial GL data will include all transactions, but may not have all the income statement or balance sheet reporting definitions.

Looking Up Values: Look up a value from one table and bring it into another table. For example, knowing how much a marketing campaign costs versus the sales opportunity amount is an important KPI to measure.
Reformatting Data

Reformat field data to free space, and make fields more readable and usable. For example, convert a date field to numeric, or reduce the precision of real numbers.

You can reformat fields within the ElastiCube using a custom SQL expression. **Numeric Representation of Date Fields**: Create a date table that is represented by a numeric representation instead of a date field to improve the query performance, as well as provide more flexibility, including the ability to filter a date range.
**Enhancing Data**

*Enhance data* by adding attributes/records that did not exist in the original data source.

- **Calculating Derived Facts**: Derived facts are additional facts that we calculate while importing or delivering the data.
- **Calendar vs. Fiscal Year**: Align a fiscal calendar with a Gregorian calendar.
- **Time Zone Conversion**: Use a source table to convert dates and times from different time zones into a uniform data set.
- **Currency Conversion**: Convert one currency into another using custom fields and a currency exchange rate table.
- **Current vs. Previous Period for Specific Date Range**: Compare data such as sales between a current period and a past period.
- **Calculating the Number of Open Orders per Day**: Check open sales orders where the order has been placed, but has not yet been delivered.
- **Slowly Changing Dimensions**: Transactional data does not usually change, however the data that describes the associated dimensions may change. See how to manage dimensions that may be updated with new values within the data warehouse at different points in time.
Integrating Data

**Note:** The images on this page were taken in the desktop version of Sisense, however, the same principles described on this page also apply to the web-based version of Sisense.

The following examples explain how to integrate and merge data from different sources into a single ElastiCube structure. This requires properly planning how to merge the data; to avoid creating unnecessary relationships, while avoiding many to many relationships. Examples in this section:

- Creating a Common Date Selection
- Financial Reporting
- Looking Up Values
Creating a Common Date Selection
Business Case

When pulling together data from multiple sources, you will have a number of different dates. Marketing has a Campaign Date, Sales has an Opportunity Date and Finance has a GL Date.
Modeling Challenge

This type of data leaves us with three sets of dates. Modeling it properly will allow you to select from one common date field while still leaving you the option to choose from one of the three date fields individually.
Solution

Create a custom table that retrieves a unique list of the dates used between the three tables.

Use the following syntax:

```
Select [GL Date] AS [Common Date]
FROM [GL Entries]
Union
Select [Marketing Campaign Date] AS [Common Date]
FROM [Marketing Campaigns]
Union
Select [Opportunity Date] AS [Common Date]
FROM [Sales Opportunities]
```

Link the four tables together:
Results

This gives us the common date field to use. This allows, for example, to select a month that will narrow down the selections across all three tables.
Financial Reporting
Business Case

Transactional systems are meant for handling transactions and not for reporting and analysis. For example, Financial GL data will include all the transactions but may not include all income statement or balance sheet reporting definitions.
Modeling Challenge

This data usually resides in other data sources or tables.
Solution

Here is what our sample GL entries look like. It is not very useful to analysis and reporting.

Another data source is needed to help define how the data will be analyzed and reported.
Start by bringing in another Data Source that contains details about the accounts and how they are categorized:
The two tables are linked on the account number. As there is one record in the GL Categories for multiple rows in the GL Entries, this is known as a One-to-Many Relationship.

The end results show that we can easily view the data in an organized fashion for analysis and reporting.

<table>
<thead>
<tr>
<th>AcctGrouping2</th>
<th>AcctGrouping1</th>
<th>AcctNo</th>
<th>Sum Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>COGS</td>
<td>COGS</td>
<td>1009</td>
<td>35,434</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1010</td>
<td>231,231</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2001</td>
<td>44,634</td>
</tr>
<tr>
<td>Salaries</td>
<td></td>
<td>3001</td>
<td>3,543,432</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3002</td>
<td>242,423</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3003</td>
<td>523,543</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3004</td>
<td>91,343</td>
</tr>
<tr>
<td>Expenses</td>
<td>Operation</td>
<td>3009</td>
<td>8,546,214</td>
</tr>
<tr>
<td></td>
<td>Expenses</td>
<td>3010</td>
<td>8,214,443</td>
</tr>
<tr>
<td>Travel &amp; Entertainment</td>
<td></td>
<td>3005</td>
<td>123,134</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3006</td>
<td>54,354</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3007</td>
<td>234,234</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3008</td>
<td>4,374,573</td>
</tr>
<tr>
<td>Revenue</td>
<td>License</td>
<td>1001</td>
<td>799,231</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1002</td>
<td>2,264,080</td>
</tr>
<tr>
<td>Maintenance</td>
<td></td>
<td>1005</td>
<td>996,276</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1006</td>
<td>1,286,252</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1007</td>
<td>321,347</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1008</td>
<td>6,534,321</td>
</tr>
<tr>
<td>Services</td>
<td></td>
<td>1003</td>
<td>3,319,673</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1004</td>
<td>2,335,265</td>
</tr>
<tr>
<td>Sum Amount</td>
<td></td>
<td></td>
<td>44,115,437</td>
</tr>
</tbody>
</table>
Looking Up Values
Business Case

Sometimes it is necessary to look up a value from one table and bring it into another table. For example, knowing how much a Marketing Campaign costs versus the Sales Opportunity amount.
Modeling Challenge

These two amounts typically reside in different systems. We need to look up the value from the Marketing system and bring it into the main Sales table.
Solution

In the Sales Opportunities table, create a custom field that will allow to look up a value from the Marketing Campaigns table based on the Marketing Campaign ID. You can do this using the Lookup function. For details, see here.
Reformatting Data

By reformatting a field, you can create a more readable, and more usable format for analysis, as well as less space consuming in some cases. For example, convert a date field to a numeric field. You can reformat fields within the ElastiCube using a custom SQL expression. An example:
 Numeric Representation of Date Fields
Business Case

Create a “Date” table using a numeric representation of a Date field instead of the Date field itself to improve query performances. In addition, a numeric representation of a Date can provide the ability of filtering the data by time range.
Solution

To convert a Date ("4/21/2012 12:36:56 AM") to an integer (20120421), use the following syntax:

\[10000 \times \text{getyear(Date)} + 100 \times \text{getmonth(Date)} + \text{getday(Date)}\]

To get the hours and minutes too, use a bigint numeric representation. So, "4/21/2012 12:36:56 AM" will become: 201204210036, using the following syntax:

\[\text{tobigint}(100000000 \times \text{getyear(DateTime)} + \text{getmonth(DateTime)} \times 1000000 + \text{getday(DateTime)} \times 10000 + 100 \times \text{gethour(DateTime)} + \text{getminute(DateTime)})\]
Enhancing Data

**Note:** The images on this page were taken in the desktop version of Sisense, however, the same principles described on this page also apply to the online version of Sisense.

The following examples explain how to add attributes and/or records that did not exist in the data source. Examples include:

- Calculating Derived Facts
- Calendar vs. Fiscal Year
- Time Zone Conversion
- Currency Conversion
- Current vs. Previous Period for Specific Date Range
- Calculating the Number of Open Orders per Day
- Slowly Changing Dimensions
Calculating Derived Facts
Business Case

Derived Facts are additional facts that we calculate while importing or delivering the data. For example:

\[
\begin{align*}
\text{Amount} &= \text{Qty} \times \text{UnitPrice} \\
\text{Profit} &= \text{Amount} - \text{Cost} \\
\text{Inventory Ratio} &= \frac{\text{Sum(Qty Sold)}}{\text{Sum(Qty Ordered)}} \\
\text{AVG Price} &= \frac{\text{Sum(Qty UnitPrice)}}{\text{Sum(Qty)}}
\end{align*}
\]
Modeling Challenge

You must decide whether to calculate the derived facts “on demand”, meaning in the web application, or in advance in the ElastiCube. Take into consideration that calculating ‘On Demand’ Derived Facts in the web application can enable more dynamic filtering, while calculating them in the ElastiCube stage will save query time when retrieving the data, and enforce calculation consistency, especially with non-trivial facts. This is due to the fact that the dashboard designer/end users will receive consistent results for measures, instead of having to create the complex measures individually, by their own understanding.
Solution

In the following schema you can create a derived fact to calculate the inventory ratio per product.

Create a custom table using an SQL Expression that joins the “Order Details” table with the “Products” table and returns the division result of “Quantity” and “UnitOnOrder”, with the following Syntax:

```sql
SELECT
    [Products].ProductID,
    tofloat(sum(UnitsOnOrder))/tofloat(sum(Quantity)) AS InventoryRatio
FROM [Products] JOIN [Order Details]
ON [Products].ProductID=[Order Details].ProductID
GROUP BY [Products].ProductID
HAVING
tofloat(sum(UnitsOnOrder))/tofloat(sum(Quantity))>0
```

The result table will give the desired results:
Connect the custom table to the rest of the tables:

<table>
<thead>
<tr>
<th>ProductID</th>
<th>InventoryRatio</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>1.614731</td>
</tr>
<tr>
<td>31</td>
<td>2.555476</td>
</tr>
<tr>
<td>49</td>
<td>2.423077</td>
</tr>
<tr>
<td>74</td>
<td>0.8754209</td>
</tr>
<tr>
<td>2</td>
<td>1.66509</td>
</tr>
<tr>
<td>32</td>
<td>2.020202</td>
</tr>
<tr>
<td>21</td>
<td>1.535433</td>
</tr>
<tr>
<td>37</td>
<td>2.4</td>
</tr>
<tr>
<td>70</td>
<td>0.4773562</td>
</tr>
<tr>
<td>56</td>
<td>0.3958828</td>
</tr>
<tr>
<td>43</td>
<td>0.4827586</td>
</tr>
<tr>
<td>68</td>
<td>0.4255319</td>
</tr>
<tr>
<td>3</td>
<td>2.560976</td>
</tr>
<tr>
<td>64</td>
<td>3.243243</td>
</tr>
<tr>
<td>66</td>
<td>3.34728</td>
</tr>
<tr>
<td>45</td>
<td>1.929134</td>
</tr>
<tr>
<td>48</td>
<td>3.043478</td>
</tr>
</tbody>
</table>

**Note:** You can also add the “InventoryRatio” measure to the “Products” table using the Lookup() function by “ProductID”.

---

**Sisense**
Calendar vs. Fiscal Year
Business Case

A large number of companies use a fiscal calendar that does not comply with the Gregorian 12-month calendar.
Modeling Challenge

This requires modeling the data properly so that the data can be reported or analyzed via the normal calendar or via the revised fiscal calendar.
Solution

In this example, let’s assume that the Fiscal Calendar starts on September 1st. So if we are in the calendar year of 2013, then the fiscal year of 2014 starts September 1st. To accomplish this, we create a custom field that takes the date field and adds four months to it.

When you create a pivot table in the web application, you will see that the new year (2014) starts in September using the Fiscal field.
<table>
<thead>
<tr>
<th>Days in Date</th>
<th>Calendar Year</th>
<th>Fiscal Year</th>
<th>Total Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>08/29/2013</td>
<td>2013</td>
<td>2013</td>
<td>35,756</td>
</tr>
<tr>
<td>08/30/2013</td>
<td>2013</td>
<td>2013</td>
<td>3,687</td>
</tr>
<tr>
<td>08/31/2013</td>
<td>2013</td>
<td>2013</td>
<td>35,617</td>
</tr>
<tr>
<td>09/01/2013</td>
<td>2013</td>
<td>2014</td>
<td>1,108</td>
</tr>
<tr>
<td>09/02/2013</td>
<td>2013</td>
<td>2014</td>
<td>91,387</td>
</tr>
<tr>
<td>09/03/2013</td>
<td>2013</td>
<td>2014</td>
<td>11,330</td>
</tr>
<tr>
<td>09/04/2013</td>
<td>2013</td>
<td>2014</td>
<td>69,273</td>
</tr>
</tbody>
</table>
Time Zone Conversion
Business Case

In many cases, we need to generate reports based on data from different time zones.
Modeling Challenge

When working with different time zones, the challenge is to store all of the business transactions in an absolute time reference that does not change with the seasons, locations (for instance – GMT), or daylight saving. Therefore, the absolute transition time is a combination of location and date.
Solution

The aim is to add an “absolute time” field to every business transaction, based on its location and time.

Step 1 – Create a Reference Source Table
Create a source table (database table / Excel / CSV) that contains the countries and cities that exist in the database, a numeric representation of timestamp range to determine if the transaction belongs to daylight savings time or not (see this [web site](#)), and the UTC to allow the conversion to GMT.

For example:

<table>
<thead>
<tr>
<th>Country</th>
<th>City</th>
<th>DST_From</th>
<th>DST_To</th>
<th>UTC</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>Seattle</td>
<td>20120311.2</td>
<td>20121103.1</td>
<td>-7</td>
</tr>
<tr>
<td>USA</td>
<td>Seattle</td>
<td>20121103.1</td>
<td>20130310.2</td>
<td>-8</td>
</tr>
<tr>
<td>USA</td>
<td>Seattle</td>
<td>20130310.2</td>
<td>20131027.1</td>
<td>-7</td>
</tr>
<tr>
<td>USA</td>
<td>Seattle</td>
<td>20131027.1</td>
<td>20140309.2</td>
<td>-8</td>
</tr>
<tr>
<td>UK</td>
<td>London</td>
<td>20120325.1</td>
<td>20121028.2</td>
<td>0</td>
</tr>
<tr>
<td>UK</td>
<td>London</td>
<td>20121028.2</td>
<td>20130330.1</td>
<td>1</td>
</tr>
<tr>
<td>UK</td>
<td>London</td>
<td>20130330.1</td>
<td>20131027.2</td>
<td>0</td>
</tr>
<tr>
<td>UK</td>
<td>London</td>
<td>20131027.2</td>
<td>20140330.1</td>
<td>1</td>
</tr>
</tbody>
</table>

Step 2 – Add a Numeric Representation of the OrderDate
To associate the Order Date with its UTC, create a custom field of type “Decimal” with a numeric representation of the Date timestamp, using this SQL statement:

```
getyear(OrderDate)*10000+getmonth(OrderDate)*100+getday(OrderDate)+ToDouble(gethour(OrderDate))/100
```

The result table should look like this:
Step 3 – Join between the Two Tables

The third step includes creating a custom SQL expression that joins between the two tables and creating the “Absolute Time” custom field within it (“GMTDate”). This is to create a synchronization between all the transactions. The custom field will be created using the “add hours” function with the matching UTC value. See the following script:

```
SELECT
[Orders].CustomerID,
[Orders].EmployeeID,
[Orders].Freight,
[Orders].OrderDate,
[Orders].OrderID,
[Orders].ShipAddress,
```
[Orders].ShipCity,
[Orders].ShipCountry,
AddHours(([Orders].OrderDate),[GMT Conversion.csv].UTC)
AS GMTDate
FROM [Orders]
JOIN
[GMT Conversion.csv]
ON
[Orders].ShipCity=[GMT Conversion.csv].City AND
[Orders].DateNum>=[GMT Conversion.csv].DST_From AND
[Orders].DateNum<[GMT Conversion.csv].DST_To

The result table will look like this:

<table>
<thead>
<tr>
<th>CustomerID</th>
<th>EmployeeID</th>
<th>Freight</th>
<th>OrderDate</th>
<th>OrderID</th>
<th>ShipAddress</th>
<th>ShipCity</th>
<th>ShipCountry</th>
<th>GMTDate</th>
</tr>
</thead>
<tbody>
<tr>
<td>WHITC</td>
<td>5</td>
<td>4.56</td>
<td>7/31/2012 12:00:00 AM</td>
<td>10269</td>
<td>1029 - 12th Ave. S</td>
<td>Seattle</td>
<td>USA</td>
<td>7/30/2012 4:00:00 PM</td>
</tr>
<tr>
<td>WHITC</td>
<td>5</td>
<td>4.56</td>
<td>7/31/2012 12:00:00 AM</td>
<td>10269</td>
<td>1029 - 12th Ave. S</td>
<td>Seattle</td>
<td>USA</td>
<td>7/30/2012 5:00:00 PM</td>
</tr>
<tr>
<td>WHITC</td>
<td>5</td>
<td>4.56</td>
<td>7/31/2012 12:00:00 AM</td>
<td>10269</td>
<td>1029 - 12th Ave. S</td>
<td>Seattle</td>
<td>USA</td>
<td>7/30/2012 4:00:00 AM</td>
</tr>
<tr>
<td>WHITC</td>
<td>5</td>
<td>4.56</td>
<td>7/31/2012 12:00:00 AM</td>
<td>10269</td>
<td>1029 - 12th Ave. S</td>
<td>Seattle</td>
<td>USA</td>
<td>7/30/2012 5:00:00 PM</td>
</tr>
<tr>
<td>BSBBEV</td>
<td>7</td>
<td>22.77</td>
<td>8/26/2012 12:00:00 AM</td>
<td>10289</td>
<td>Fauntleroy Circus</td>
<td>London</td>
<td>UK</td>
<td>8/26/2012 1:00:00 AM</td>
</tr>
<tr>
<td>BSBBEV</td>
<td>7</td>
<td>22.77</td>
<td>8/26/2012 12:00:00 AM</td>
<td>10289</td>
<td>Fauntleroy Circus</td>
<td>London</td>
<td>UK</td>
<td>8/26/2012 12:00:00 AM</td>
</tr>
<tr>
<td>BSBBEV</td>
<td>7</td>
<td>22.77</td>
<td>8/26/2012 12:00:00 AM</td>
<td>10289</td>
<td>Fauntleroy Circus</td>
<td>London</td>
<td>UK</td>
<td>8/26/2012 1:00:00 AM</td>
</tr>
<tr>
<td>BSBBEV</td>
<td>7</td>
<td>22.77</td>
<td>8/26/2012 12:00:00 AM</td>
<td>10289</td>
<td>Fauntleroy Circus</td>
<td>London</td>
<td>UK</td>
<td>8/26/2012 12:00:00 AM</td>
</tr>
<tr>
<td>WHITC</td>
<td>4</td>
<td>23.29</td>
<td>11/1/2012 12:00:00 AM</td>
<td>10344</td>
<td>1029 - 12th Ave. S</td>
<td>Seattle</td>
<td>USA</td>
<td>10/31/2012 4:00:00 PM</td>
</tr>
<tr>
<td>WHITC</td>
<td>4</td>
<td>23.29</td>
<td>11/1/2012 12:00:00 AM</td>
<td>10344</td>
<td>1029 - 12th Ave. S</td>
<td>Seattle</td>
<td>USA</td>
<td>10/31/2012 5:00:00 PM</td>
</tr>
<tr>
<td>WHITC</td>
<td>4</td>
<td>23.29</td>
<td>11/1/2012 12:00:00 AM</td>
<td>10344</td>
<td>1029 - 12th Ave. S</td>
<td>Seattle</td>
<td>USA</td>
<td>10/31/2012 4:00:00 PM</td>
</tr>
<tr>
<td>SEVES</td>
<td>5</td>
<td>288.43</td>
<td>11/21/2012 12:00:00 AM</td>
<td>10359</td>
<td>90 Wadhurst Rd.</td>
<td>London</td>
<td>UK</td>
<td>11/21/2012 1:00:00 AM</td>
</tr>
<tr>
<td>SEVES</td>
<td>5</td>
<td>288.43</td>
<td>11/21/2012 12:00:00 AM</td>
<td>10359</td>
<td>90 Wadhurst Rd.</td>
<td>London</td>
<td>UK</td>
<td>11/21/2012 12:00:00 AM</td>
</tr>
<tr>
<td>SEVES</td>
<td>5</td>
<td>288.43</td>
<td>11/21/2012 12:00:00 AM</td>
<td>10359</td>
<td>90 Wadhurst Rd.</td>
<td>London</td>
<td>UK</td>
<td>11/21/2012 1:00:00 AM</td>
</tr>
<tr>
<td>EASTC</td>
<td>1</td>
<td>71.97</td>
<td>11/26/2012 12:00:00 AM</td>
<td>10364</td>
<td>35 King George</td>
<td>London</td>
<td>UK</td>
<td>11/26/2012 1:00:00 AM</td>
</tr>
<tr>
<td>EASTC</td>
<td>1</td>
<td>71.97</td>
<td>11/26/2012 12:00:00 AM</td>
<td>10364</td>
<td>35 King George</td>
<td>London</td>
<td>UK</td>
<td>11/26/2012 12:00:00 AM</td>
</tr>
<tr>
<td>EASTC</td>
<td>1</td>
<td>71.97</td>
<td>11/26/2012 12:00:00 AM</td>
<td>10364</td>
<td>35 King George</td>
<td>London</td>
<td>UK</td>
<td>11/26/2012 1:00:00 AM</td>
</tr>
<tr>
<td>EASTC</td>
<td>1</td>
<td>71.97</td>
<td>11/26/2012 12:00:00 AM</td>
<td>10364</td>
<td>35 King George</td>
<td>London</td>
<td>UK</td>
<td>11/26/2012 1:00:00 AM</td>
</tr>
<tr>
<td>SEVES</td>
<td>1</td>
<td>22.21</td>
<td>12/9/2012 12:00:00 AM</td>
<td>10377</td>
<td>90 Wadhurst Rd.</td>
<td>London</td>
<td>UK</td>
<td>12/9/2012 1:00:00 AM</td>
</tr>
<tr>
<td>SEVES</td>
<td>1</td>
<td>22.21</td>
<td>12/9/2012 12:00:00 AM</td>
<td>10377</td>
<td>90 Wadhurst Rd.</td>
<td>London</td>
<td>UK</td>
<td>12/9/2012 12:00:00 AM</td>
</tr>
<tr>
<td>SEVES</td>
<td>1</td>
<td>22.21</td>
<td>12/9/2012 12:00:00 AM</td>
<td>10377</td>
<td>90 Wadhurst Rd.</td>
<td>London</td>
<td>UK</td>
<td>12/9/2012 1:00:00 AM</td>
</tr>
<tr>
<td>SEVES</td>
<td>1</td>
<td>22.21</td>
<td>12/9/2012 12:00:00 AM</td>
<td>10377</td>
<td>90 Wadhurst Rd.</td>
<td>London</td>
<td>UK</td>
<td>12/9/2012 12:00:00 AM</td>
</tr>
<tr>
<td>SEVES</td>
<td>2</td>
<td>34.86</td>
<td>12/19/2012 12:00:00 AM</td>
<td>10388</td>
<td>90 Wadhurst Rd.</td>
<td>London</td>
<td>UK</td>
<td>12/19/2012 1:00:00 AM</td>
</tr>
<tr>
<td>SEVES</td>
<td>2</td>
<td>34.86</td>
<td>12/19/2012 12:00:00 AM</td>
<td>10388</td>
<td>90 Wadhurst Rd.</td>
<td>London</td>
<td>UK</td>
<td>12/19/2012 12:00:00 AM</td>
</tr>
<tr>
<td>SEVES</td>
<td>2</td>
<td>34.86</td>
<td>12/19/2012 12:00:00 AM</td>
<td>10388</td>
<td>90 Wadhurst Rd.</td>
<td>London</td>
<td>UK</td>
<td>12/19/2012 1:00:00 AM</td>
</tr>
</tbody>
</table>
Step 4 – Make Schema Adjustments

For the next step, do the following:
- Replace the current Orders table with the new one,
- Refer to the new “Absolute Time” custom field (“GMTDate”) as the leading date field
- Make the reference tables (“Orders” and “GMT Conversion.csv”) invisible.
Currency Conversion
Business Case

Most data for entities is recorded in their local reporting currency (ie $ for United States, £ for UK). Here we want to convert all the amounts to USD.
Modeling Challenge

This requires determining the Currency Rate of the region and then multiplying the value in local currency by the associated Exchange Rate by Month.
Solution

Create two custom fields in the GL Entries. The first will look up the Currency code of the region. This field will be used along with a month field to link to the Exchange Rates table.

The first field in the GL Entries is created using the lookup function to retrieve values from the Currency Codes table.

```
Lookup([Currency Codes],[Currency Code],Region,Region)
```

Then create a second Custom Field for the Month of the GL Date.

```
GetMonth([GL Date])
```

Next, link the fields together (note that both Month fields were set to Integer and the Currency Codes table to Invisible).
The Local Amount multiplied by the Exchange Rate gives the Converted USD Amount.

### LOCAL and USD

<table>
<thead>
<tr>
<th>Region</th>
<th>GL Month</th>
<th>Local Amount</th>
<th>Exchange Rate</th>
<th>Converted USD Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>IL</td>
<td>1</td>
<td>1,303,490</td>
<td>.25</td>
<td>325,873</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>833,853</td>
<td>.27</td>
<td>225,140</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1,915,644</td>
<td>.28</td>
<td>536,380</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1,303,288</td>
<td>.27</td>
<td>345,371</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1,232,185</td>
<td>.29</td>
<td>357,334</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>1,743,100</td>
<td>.28</td>
<td>479,353</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>1,387,776</td>
<td>.28</td>
<td>388,577</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>17,116</td>
<td>.29</td>
<td>4,964</td>
</tr>
<tr>
<td>US</td>
<td>1</td>
<td>939,352</td>
<td>1.00</td>
<td>939,352</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>752,517</td>
<td>1.00</td>
<td>752,517</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>604,556</td>
<td>1.00</td>
<td>604,556</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>648,043</td>
<td>1.00</td>
<td>648,043</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>769,723</td>
<td>1.00</td>
<td>769,723</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>884,302</td>
<td>1.00</td>
<td>884,302</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>550,515</td>
<td>1.00</td>
<td>550,515</td>
</tr>
</tbody>
</table>
Current vs. Previous Period for Specific Date Range
Business Case

In many cases we would like to compare our business’ performance last week, to the week before, or maybe we would like to see a percentage of sales growth for the current month/quarter compared to the previous month/quarter.
Modeling Challenge

Since we want the compared time range to be as flexible as possible, the solution has to include both layers – ElastiCube and web application.
Solution

Create a custom table in the ElastiCube to summarize the totals/counts per day for the source table:

```sql
SELECT 
    a.Date, 
    sum(a.Revenue) AS value 
FROM [Accord 2011 Client List] AS a 
GROUP BY a.Date 
```

Create a custom table in the ElastiCube with current vs. previous values, by adjusting the script below:

```sql
SELECT 
    curr.Date AS date, 
    curr.value AS current, 
    prev.value AS prev 
FROM [sum] curr 
LEFT JOIN [sum] AS prev 
ON curr.Date = addyears(prev.Date,1) 
UNION 
SELECT 
    addyears(prev.Date,1) AS date, 
    curr.value, 
    prev.value 
FROM [sum] prev 
LEFT JOIN [sum] AS curr 
ON prev.Date= addyears(curr.Date,-1) 
```

In the web application, add a ‘date range picker’ using the days from the custom table. Then add two new numeric indicators. In the first numeric picker add the
'sum of the current value', in the second numeric picker, add the 'sum of the previous value'.
In the date range picker, select the days of interest and you will see the current and previous values.
Calculating the Number of Open Orders per Day
Business Case

An open sales order is where the order has been placed but has not yet been delivered. If for example there is an order for 100 items and against this order only 50 items have been delivered (it is partially delivered). A high level of open orders per day may indicate that something is wrong with orders handling.
Modeling Challenge

We cannot just count the number of orders per day because it will exclude orders that were open on a certain day and are already closed. Therefore, we will need to create a snapshot of the number of open orders per day.
Solution

Import an Excel file with all dates listed in the Orders table into the ElastiCube.

To improve query performance, convert all the date fields into numeric representations (for more information, see Numeric Representation of Date fields).

Create the following custom table:

SELECT
s.Dates,
tm.Created_At,
tm.Closed_At,
tm.TicketId
FROM [All Dates] s LEFT JOIN [Orders] tm
ON s.DateInt >= tm.CreatedAtInt
AND (tm.SolvedAt IS NULL OR s.DateInt <= tm.SolvedAtInt)
Slowly Changing Dimensions
Business Case

Transactional data typically does not change, however the data that describes the associated dimensions may change. This example demonstrates how to manage dimensions that may be updated with new values within the data warehouse at different points in time.

For example, a customer that was living in NYC and moved to LA earlier this year.

<table>
<thead>
<tr>
<th>Date</th>
<th>Customer</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/1/2011</td>
<td>John Smith</td>
<td>NYC</td>
</tr>
<tr>
<td>1/1/2013</td>
<td>John Smith</td>
<td>LA</td>
</tr>
</tbody>
</table>
Modeling Challenge

Following the example above, when the transactions were made last year, the customer was living in NYC. Later this year, the customer moved to LA. If you decide to refer only to the last city and summarize revenue by city, the NYC transaction will be credited to LA only because the customer currently lives there.

If you connect the “Customer” field within the above table to the “Customer” field in the fact table, you will create a “Many to many” relationship because “Customer” is not a unique identifier of this table.
Solution

The solution is to change the level of granularity of the “Customer_City” table and add the “Date” field to the key –
  Concatenation of the Slowly Changing Dimension table’s unique identifier (for example – Customer_ID + Date)
  Creation of the same concatenated field in the transactions table.
  Merge between the 2 keys.
This way, you can associate the [Customer_ID + Date] key of every transaction with the relevant customer city.
Functions Overview

Sisense supports many functions that you can use for data calculations. For a short video overview of the available types of functions, see below:

The following topics describe the functions Sisense supports:
- Date and Time Functions
- Logical Functions
- Mathematical Functions
- String and Text Functions
- Web Functions
Date and Time Functions

You can use date and time functions in custom tables and fields. See Adding a New Custom Table and Defining and Editing Field Formula for further details.

For a short video overview of the date and time functions, see below:

Below are explanations of the available date and time functions:

**ADDYEARS(datetime, number)**
**ADDQUARTERS(datetime, number)**
**ADDMONTHS(datetime, number)**
**ADDDAYS(datetime, number)**
**ADDHOURS(datetime, number)**
**ADDMINUTES(datetime, number)**
**ADDSECONDS(datetime, number)**

Adds a given number of years|quarters|months|days|hours|minutes|seconds to a specified date/time. An example of when to use this is when the fiscal year is not the same as the Gregorian calendar’s beginning and end dates.

**CREATEDATE(year,month,day)**

Creates a timestamp from a given year, month and day. Time is set to midnight.

**CURRENTDATE()**

Returns the current date.

**CURRENTTIME()**

Returns the current time.

**CURRENTTIMESTAMP()**

Returns the current timestamp.

**GETYEAR(datetime)**
GETQUARTER(datetime)
GETMONTH(datetime)
GETDAY(datetime)
GETHOUR(datetime)
GETMINUTE(datetime)
GETSECOND(datetime)

Returns a number that represents the
eyear|quarter|month|day|hour|minute|second in a given date/time.

DAYOFWEEK(datetime)
Returns the day, out of seven days in a week, represented by a given date/time.

DAYOFYEAR(datetime)
Returns the day, out of 365 days in a year, represented by a given date/time.

WEEKOFYEAR(datetime)
Returns the week, out of 52 weeks in a year, represented by a given date/time.

DAYNAME(number)
Returns the name of the day represented by the given number.

YEARDIFF(end,start)
QUARTERDIFF(end,start)
MONTHDIFF(end,start)
DAYDIFF(end,start)
HOURDIFF(end,start)
MINUTEDIFF(end,start)
SECONDDIFF(end,start)

Returns the difference in years|quarters|months|days|hours|minutes|seconds

TODATETIME()
Converts a specified value to a DateTime value. Sisense supports the string
format: YYYY-MM-DD HH:mm:ss:ms and YYYY-MM-DD HH:mm:ss. If not
HH:mm:ss are entered, Sisense assumes that the time is 12:00AM.
Logical Functions

You can use mathematical functions in custom tables and fields. See Adding a New Custom Table and Defining and Editing Field Formula for further details. Below are explanations of available logical functions.

**IFBIGINT(condition, true value, false value)**
Returns the first value if the boolean evaluate is true, else returns the second value.

**IFDATETIME(condition, true value, false value)**
Returns the first value if the boolean evaluate is true, else returns the second value.

**IFDOUBLE(condition, true value, false value)**
Returns the first value if the boolean evaluate is true, else returns the second value.

**IFINT(condition, true value, false value)**
Returns the first value if the boolean evaluate is true, else returns the second value.

**IFSTRING(condition, true value, false value)**
Returns the first value if the boolean evaluate is true, else returns the second value.

**INRANGE(value, start, end)**
Returns true if a given value is between start and end.

**ISNULL(value)**
Returns true if a given value is null.

**LOOKUP(remote_table,remote_result_column,current_match_column,remote_match_column)**
The Lookup function imports a field from one table into another table, by matching two other corresponding fields from both tables. The result will be the value in remote_result_column for which the corresponding remote_match_column equals the current_match_column.

For a short video demonstration of the Lookup function, see below:
Mathematical Functions

You can use mathematical functions in custom tables and fields. See Adding a New Custom Table and Defining and Editing Field Formula for further details.

Below are explanations of available mathematical functions.

**ACOS(number)**
Returns the angle, in radians, whose cosine is the given number. Also referred to as arccosine.

**ASIN(number)**
Returns the angle, in radians, whose sine is a given number. Also referred to as arcsine.

**ATAN(number)**
Returns the angle in radians whose tangent is a given number. Also referred to as arctangent.

**CEILING(number)**
Returns the smallest integer that is greater than, or equal to, the given number.

**COS(number)**
Returns the trigonometric cosine of a given angle, in radians.

**COSH(number)**
Returns the hyperbolic cosine of a given number.

**COT(number)**
Returns the cotangent of a given number.

**DIV(number1, number2)**
Returns the number of times number2 fits completely in number1.

**EXP(number)**
Returns the exponential value of a given number.
FLOOR(number)
Returns the largest integer less than or equal to the given numeric expression.

ISINFINITY(number)
Returns -1 if the number evaluates to negative infinity, 1 if it evaluates to positive infinity, and 0 otherwise.

ISINTEGER(number)
Returns whether a given number is a whole number.

LOG(number)
Returns the natural logarithm of a given number.

LOG10(number)
Returns the base-10 logarithm of the given float expression

MAXVAL(number1,number2)
Returns the highest of two numbers.

MINVAL(number1,number2)
Returns the smallest of two numbers.

MOD(number1,number2)
Returns the remainder of dividing number1 by number2 (modulo).

PI()
Returns the constant value of the mathematical PI.

POWER(number1,number2)
Returns the value of the given number raised to a specified power.

RAND()
Returns a random number between 0 and 32767

RANK()
Returns the rank (i.e. row number) of the current record in a table.

RANKASC(order_by_field)
Returns the row number ordered by given field in ascending manner.
Example: (3, 5, 5, 19) is ranked in ascending order (1, 2, 3, 4).
You can also sort by multiple columns:
RANKASC([partitionA],[partitionB]...[partitionN],[orderByField])

Example: RANKASC([EmployeeID],[CategoryID],[Quantity])

**RankCompetitionAsc(partition fields list, order field)**
Returns the rank of each row within the partition of a result set. The rank of a row is one plus the number of ranks that come before the current row ordered by the rightmost field parameter in ascending order.
Leftmost field parameters (optional) define the partitions in which each ordering starts.
Example: (3, 5, 5, 19) is ranked in ascending order (1, 2, 2, 4).

**RankCompetitionDesc(partition fields list, order field)**
Returns the rank of each row within the partition of a result set. The rank of a row is one plus the number of ranks that come before the current row ordered by the rightmost field parameter in descending order.
Leftmost field parameters (optional) define the partitions in which each ordering starts.
Example: (3, 5, 5, 19) is ranked in descending order (4, 2, 2, 1).

**RankDenseAsc(partition fields list, order field)**
Returns the dense rank of rows within the partition of a result set without any gaps in the ranking. The rank of a row is one plus the number of distinct ranks that come before the current row ordered by the rightmost field parameter in ascending order.
Leftmost field parameters (optional) define the partitions in which each ordering starts.
Example: (3, 5, 5, 19) is ranked in ascending order (1, 2, 2, 3).

**RankDenseDesc(partition fields list, order field)**
Returns the dense rank of rows within the partition of a result set without any gaps in the ranking. The rank of a row is one plus the number of distinct ranks that come before the current row ordered by the rightmost field parameter in descending order.
Leftmost field parameters (optional) define the partitions in which each ordering starts.
Example: (3, 5, 5, 19) is ranked in descending order (3, 2, 2, 1)

**RANKDESC(order_by_field)**
Returns the row number ordered by given field in descending manner.
Example: (3, 5, 5, 19) is ranked in ascending order (4, 3, 2, 1).

You can also sort by multiple columns:
RANKDESC([partitionA],[paritionB]…[partitionN],[orderByField])
Example: RANKDESC([EmployeeID],[CategoryID],[Quantity])

**ROUND(number,precision)**
Returns the given number, rounded to a specified precision.

**SIN(number)**
Returns the trigonometric sine of a given angle, in radians.

**SINH(number)**
Returns the hyperbolic sine of a given number.

**SQRT(number)**
Returns the square root of a given number.

**TAN(number)**
Returns the tangent of a given number.

**TANH(number)**
Returns the hyperbolic tan of a given number.

**TOBIGINT(string)**
Converts a string representing a valid BigInt value to BigInt type.
TODOUBLE(string)
Converts a string representing a valid double value to Double type.

TOINT(string)
Converts a string representing a valid Int value to Int type.
String and Text Functions

You can use mathematical functions in custom tables and fields. See Adding a New Custom Table and Defining and Editing Field Formula for further details.

For a short video overview of the string and text functions, see below:

Below are explanations of the available string and text functions:

**ASCII(char)**
Returns the Int ascii code of a given character.

**CHAR(number)**
Returns a character for a given ascii code.

**CONCAT(string1,string2)**
Concatenates string1 and string2.

**INDEXOF(string1,string2,n)**
Returns the index of the n-occurrence of string2 in string1.

**INSERT(string1,string2,overrides,index)**
Inserts string2 into string1 at a specified index, possibly overriding a specified number characters in string1.

**INSERTSTR(string1,string2,index)**
Inserts string2 into string1 at specified zero-based index.

**ISNULLOREMPTY(string)**
Returns true if the string is null or empty.

**LEFT(string,length)**
Returns a substring of a specified length, starting at the first character.

**LENGTH(string)**
Returns the length of a given string.
LOWER(string)
Returns a lowercase version of a given string.

LTRIM(string)
Removes whitespaces from the start of the string.

REMOVEAT(string,index)
Removes a character at a specified index from a given string.

REMOVERANGE(string,index,count)
Removes a specified number of characters from a given string starting at a specified index.

REPEAT(string,number)
Returns a concatenation of a string repeated a given number of time.

REPLACEALL(string,oldstring,newstring)
Replaces all occurrences of old string with new string.

RIGHT(string,length)
Returns a substring of a specified length, end with the last character.

RTRIM(string)
Removes whitespaces from the end of the string.

SOUNDEX(string)
Returns a four-character (SOUNDEX) code to evaluate the similarity of two objects.

STRBETWEEN(string, startstring, endstring)
Returns the substring contained between startstring and endstring.

STRPARTS(string,delimiter,n)
Returns the string delimited by a specified delimiter. The search ends after the n-occurrence of the delimiter is found.

SUBSTRING(string,index,length)
Returns the substring of a specified length, starting at a specified index.

TOSTRING(object)
Converts a given value to a string.

**TRIM(string)**
Removes whitespaces from both ends of a string.

**UPPER(string)**
Returns an uppercase version of a given string.
Web Functions

You can use web functions in custom tables and fields. See Adding a New Custom Table and Defining and Editing Field Formula for further details. Below are explanations of available web functions.

**Note:** In all circumstances, the URL must include ‘http://’. For example, if the URL is www.sisense.com, the syntax should be GetFile (‘http://www.sisense.com’)

**DomainIs: true/false**
DomainIs (URL, compared domain)

**GetFile: string GetFile (url)**
Extract the filename portion of the path in the given url and returns it.
The filename is everything after the last slash in the path.

**GetHost: string GetHost (url)**
Extract host from the given url, for example ‘www.wikipedia.org’ in ‘http://www.wikipedia.org’

**GetPassword: string GetPassword (url)**
Extract password from the given url.
Specified in the url following the user name, with leading colon (‘:’) and trailing at sign (@), as in ‘http://user:password@host/’

**GetPath: string GetPath (url)**
Extract path from the given url.
Path is the url portion following the host name, for example ‘/wiki/Business_intelligence’ in ‘http://www.wikipedia.org/wiki/Business_Intelligence’
GetPort: string GetPort (url)
Extract port from the given url, parsed as integer, for example 123 in
'http://www.wikipedia.org:123/'

GetQuery: string GetQuery(url)
Extract the query from the given url.
Specified in the url after the '?' and before (optional) the '#' sign, for example
'par=val' in 'http://www.wikipedia.org/?par=val#ref'.

GetQueryParamValue (url,param name): string GetQueryParamValue
(url,param name)
Extract param value from the given url’s query, by the given param name.
Param/Value pairs when supplied, are delimited by an ampersand or a
semicolon.

GetRef: string GetRef (url)
Extract ref from given url.
Specified in the url starting after last appearance of the '#' sign, for example 'ref'
in 'http://www.wikipedia.org/?par=val#ref'

GetScheme: string GetScheme (url)
Extract scheme from given url.
Skips leading spaces and followed colon (':') sign, for example
'http://www.wikipedia.org/

GetUserName: string GetUserName (url)
Extract user name from given url.
Specified in the url before the host name and delimited by an '@' as in
'http://user@host/.'

HasHost: true/false HasHost (url)
Get whether the given url specifies a host.

HasPassword: true/false HasPassword (url)
Get whether the given url specifies a password.
HasPath: true/false HasPath (url)
Get whether the given url specifies a path.

HasPort: true/false HasPort (url)
Get whether the given url specifies a port.

HasQuery: true/false HasQuery (url)
Get whether the given url specifies a query.

HasRef: true/false HasRef (url)
Get whether the given url specifies a ref.

HasScheme: true/false HasScheme (url)
Get whether the given url specifies a scheme.

HasUsername: true/false HasUsername (url)
Get whether the given url specifies a user name.

HostIsIpAddress: true/false HostIsIpAddress (url)
Get whether the given url host name is an ip address.

IsEmptyUrl: true/false IsEmptyUrl (url)
Get whether the given url is empty.

IsFileUrl: true/false IsFileUrl (url)
Get whether the given url is a file url.

IsSecuredUrl: true/false IsSecuredUrl (url)
Get whether the given url scheme indicates a secured connection(https).

IsStandardScheme: true/false IsStandardScheme (url)
Returns true if the scheme for the current url is a known ‘standard’ scheme.
Standard schemes have an authority and a path section (including file:).

IsValidUrl: true/false IsValidUrl (url)
Get whether the given url has a valid form.

PathForRequest: true/false PathForRequest (url)
Returns the path that should be sent to the server.
This is the path, parameter, and query portions of the url.
SchemeIs: true\false SchemeIs (url)

Get whether the given url scheme portion matches the given scheme.
Building ElastiCubes

After defining a schema, you need to build your ElastiCube. Building an ElastiCube imports the data from your data sources that you added to your schema. You must build an ElastiCube at least once before the ElastiCube data can be used in a dashboard.

For a short overview of build settings, see below.

If your data set is very large, a build may take some time. You can cancel the build in the build log that is opened when you run a build.

To build an ElastiCube:
In the Data page, open the relevant ElastiCube and click Build. The Build Options dialog box is displayed.

Select **Replace All**, **By Table**, or **Changes Only**. **Replace All** builds the ElastiCube from scratch with the current data from all the sources in the schema. **By Table** builds the ElastiCube as defined on the table level where all tables not defined are built from scratch. **Changes Only** builds the ElastiCube with data for tables that were added to the schema or modified since the last build. For an in-depth description of these settings and the implications of each setting, see [ElastiCube Build Settings](#).

Toggle the switch below to limit the amount of rows imported for each table.

This is useful for building samples of your data and reduces the overall amount of time it takes to build an ElastiCube.

Click **Build**.

A log is displayed that describes the build process. When the build is complete, the following message appears in the logs: Build successfully ended.
Building ElastiCubes Remotely

**Note:** This functionality is only available in the desktop version of Sisense. This functionality will be available Sisense in the near future.

You can build the ElastiCube locally on your machine or build the ElastiCube remotely from another machine in which Sisense is installed. This is useful for example when your configuration supports Test and Production environments. For more information on setting up production environments, click [here](#).

When connecting remotely, your ElastiCube server must have Sisense installed and a unique Sisense license. While you can have multiple Administrators working on and building an ElastiCube, only one Administrator can work on an ElastiCube at a time to prevent data corruption. In addition, the machine in which the build is performed, must have sufficient resources to be able to handle all of your data. See Sisense’s [minimum requirements](#) for more information.

**To build an ElastiCube remotely:**
- On the ElastiCube server where the ElastiCube is to be built, open ports 811 and 812 for inbound and outbound access.
- In Sisense, open the Sisense Server Console by clicking +.
- In the **Connection Settings** window, enter a name for the server and its IP address.
- Click **Connect**. The list of ElastiCubes for that server are displayed in the Sisense Server Console.
- In Sisense, click **Build**.
- Under Target Server, select the new server you have just added.
- Define your build settings and click **Build**.
ElastiCube Build Settings

When you build your ElastiCube, data is imported from your data sources into the ElastiCube according to settings you define on the model level and on the table level.

This topic describes how data can be imported on each level and the implications of each of the ElastiCube build settings.
Model Level

On the model level, you have three choices for importing data, Replace All, By Table, and Changes Only.

- Replace All
- By Table
- Changes Only

Replace data in all tables with the current data from the source

Sample Data

Build  Cancel
**Replace All**

When you replace all your data on the model level, any existing data imported into your ElastiCube is overwritten and any custom tables or columns are recalculated. Sisense imports an exact copy of your data from the source. If the size of your data is large or the connection speed to the source is slow, it can take a long time to build your ElastiCube.

The diagram below illustrates what happens before and after a Replace All build:

<table>
<thead>
<tr>
<th>Day 1</th>
<th>Data Source</th>
<th>Sisense</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Column 1</strong></td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td><strong>Column 2</strong></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>Column 1</strong></td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td><strong>Column 2</strong></td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Day 2</th>
<th>Data Source</th>
<th>Sisense</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Column 1</strong></td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td><strong>Column 2</strong></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>Column 1</strong></td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td><strong>Column 2</strong></td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td><strong>Column 1</strong></td>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td><strong>Column 2</strong></td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

**Note:** While building the ElastiCube with schema changes or entire ElastiCube builds, you can continue to run queries. In the event that the build fails, Sisense restores the original version of the ElastiCube and attempts the build again. While Sisense restores the original ElastiCube, some downtime may occur while Sisense copies the ElastiCube to your drive. The downtime is the amount of time it takes to copy the ElastiCube locally. If you require high availability for your data, Sisense provides this through ElastiCube Sets. When Sisense restores an ElastiCube a copy of the original ElastiCube is created. You should verify that your server can store multiple copies ElastiCubes until the restoration process is complete.
By Table

When you build By Table, all your data is overwritten except for tables where you have specifically defined their build behavior in the schema.

This option is only displayed when a table’s build behavior has been defined in the schema. When you have customized a table’s build behavior, an icon is displayed that indicates the table’s build behavior, for example, Changes Only or Append. These icons are also described in the legend in the bottom-right corner of the schema.

In the Build Settings dialog box, you can click the table links to see which tables have been configured or are set to be overwritten.

For more information about table build behavior, see Table Level below.
Changes Only

When you build by Changes Only, you build only new tables or tables that have changed since the last build. This is useful when you frequently need to refresh a large data source. For example, if you have a data source that is updated daily, rather than rebuild the entire ElastiCube daily, you can just import the new data added each day. This option can significantly reduce the amount of time it takes to complete a build.

Tables that have changed since the last build include the following:
- New tables that were added
- Tables where columns were added or dropped
- Custom tables where the expression has changed
- Tables with custom columns where the expression has changed
- Custom tables that have SQL expressions which reference other tables that have changed

If your table meets any of the above criteria, then the table’s data is overwritten and rebuilt from scratch.
Table Level

You can define your build’s behavior per table by selecting the table’s menu > Build Behavior, and selecting the relevant option.

Then, when you define what type of build you want to do, select the By Table option.
You can set the build behavior of a table to one of the following options:

- **Replace All**
- **By Table**
- **Changes Only**

Build behavior is selected at the individual table level.

- 7 Tables set to Replace All
- 1 Table set to Changes Only
Replace All (Default)

Replaces all data at the time of the build. This is recommended for dimension tables, for example: store attributes, or dimensions, which describe the objects in a fact table.
Append

Adds all the data from the source table and appends it to the existing data in the ElastiCube.

The diagram below illustrates what happens before and after an Append build:

Day 1

<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
</tr>
</tbody>
</table>

Sisense

<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
</tr>
</tbody>
</table>

Day 2

<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
</tr>
</tbody>
</table>

On Day 2, the data taken from the source was appended onto the existing data in Sisense without ignoring any rows of data.
Changes Only

Does not import any data unless changes have occurred in the table. This is recommended with summary/snapshot fact tables and with data marts (smaller subsets of data, tailored for specific needs).

Tables that have changed since the last build include the following:
- New tables that were added
- Tables where columns were added or dropped
- Custom tables where the expression has changed
- Tables with custom columns where the expression has changed
- Custom tables that have SQL expressions which reference other tables that have changed
Accumulate By

Adds additional rows of data incrementally to an existing table according to the data in a specified integer or date column that acts as an index. You can select a specific integer or date column that will be used to determine whether to accumulate data at the time of the build. The column acts as an index, and if the index value is greater in a subsequent build, then data is accumulated for the table. This is recommended for detailed fact tables, for example: store quantitative information for analysis.

**Note:** Only date and integer columns are supported.

To select the integer or date column within a table as the parameter to accumulate by, click on the column’s menu, and then select **Accumulate By**.

Accumulation behavior for integers and for dates are as follows:

When you select an integer, only source rows with a value greater than the maximum index value in the ElastiCube table will be inserted. Existing data in the ElastiCube table will not be modified or deleted.

The diagram below illustrates what happens before and after an Accumulative build:
On Day 2, the integer value 3 was set as an index so new rows that are less than the integer value are not added in the next build. In this case, D and E of Column 1 were excluded as 2 and 1 of the new rows are less than the index value 3.
Accumulative Build Support

Some data sources (for example, CSV files) do not support accumulating data on the column level, and in such cases data will be duplicated. The table below lists the data providers that do not support accumulative builds and appending data to tables.

Note: Data providers that are not listed in this table support accumulate builds both by appending to table and accumulating by column.

<table>
<thead>
<tr>
<th>Data Source Provider</th>
<th>Sisense Web Application</th>
<th>Desktop ElastiCube Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Append to Table</td>
<td>Accumulate by Column</td>
</tr>
<tr>
<td>CSV file</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Google AdWords</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Google Analytics</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Google Spreadsheets</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MS Excel</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>OLEDB</td>
<td>Not supported</td>
<td>Yes</td>
</tr>
<tr>
<td>Salesforce</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Scheduling Data Loads

After you build an ElastiCube, you can schedule future builds to automatically synchronize with the underlying data source to ensure all data is up to date. There are two options you can choose when scheduling a build: by time or by intervals. Builds scheduled by time occur according to the days and hour you define. You can configure Sisense to build an ElastiCube automatically any or every day of the week at the same hour.

Builds scheduled by intervals occur after the defined interval passes. You can set a single interval for an ElastiCube.

**Note**: Only one option can be selected at a time. The last configuration you set is the configuration that will be used.

If you want to cancel a scheduled build, select **Disable** and click **Save**.

By default, scheduled builds are Full builds (replace data in all tables with the current data from the source), unless any table within the ElastiCube is set to Accumulative or a table is set to be ignored.

Your ElastiCube’s build configuration determines how the scheduled build is to be performed. For more information about how to build ElastiCubes, click [here](#).

**To schedule a build:**
In the **Data** page, open the menu of the relevant ElastiCube and select **Schedule Build**. The Schedule Build window is displayed.

Select the relevant scheduling option:

- **Disable**: Click to disable a scheduled build.
- **Daily**: Select the relevant days and define the hour when the build is to occur. For each selected day, the build will occur at the same hour defined. **Note**: The build will occur according to the timezone of the Sisense server, and not according to the timezone of your machine.
- **Interval**: Select the time in between automatic builds. The first automatic build begins after you save your settings and the interval passes.

Click **Save**.
Changing an ElastiCube

This topic describes how to make changes in your ElastiCube.

To update an ElastiCube:
- Open the Data page and click the relevant ElastiCube. The schema of the ElastiCube is displayed.

Changes are saved as you work and take affect when you build the ElastiCube.
Live Connect

Traditionally, your data was only as fresh as your latest ElastiCube build. With live connections, you can refresh your dashboards manually or set them to refresh automatically every few seconds. Sisense Live Connect queries are run directly against the data source. Sisense performs no additional caching on the data.

The most significant difference between dashboards built on ElastiCubes and dashboards connected to live data sources is how the load is managed. With an ElastiCube, you refresh the data when you build an ElastiCube and query the ElastiCube, not the data sources that define your ElastiCube. With live connections, queries are run against the data source itself. This means that you don't have to wait for an ElastiCube to finish building before you see the results, however, your data source must be able to handle the potential load of multiple refreshes.

To create dashboards built on live connections, you create live models. These data models include connection and credential details to the data sources. After you have created your live model, you publish it. Publishing the live model adds it to your list of data sources from which you can select when working with dashboards.
Limitations

The following features are not supported in live connections:

- Pulse alerts
- Box & Whisker plot
- Certain analytical functions such as Mode and Standard Deviation
- Changing live data sources from dashboards
- Custom SQL tables and columns
Frequently Asked Questions

Q: I cannot connect to my data source?
A: As the connection is live, the data source must be available for Sisense to connect. Also, if your credentials changes, you must update your data set so Sisense can continue to connect and refresh the data.

Q: No data sources are displayed when trying to add one?
A: In Windows Services, verify that the following are running, and if not, restart them:
   - Sisense.CLRConnectorsContainer
   - Sisense.Discovery
   - Sisense.ECMServer
   - IIS

Q: If I use Windows Authentication with my SQL server, what do I enter in the Username and Password fields?
A: Enter the Location and leave the Username and Password fields blank.

Q: After entering a data set name, the Done button is not displayed:
A: Make sure the data set’s name is unique. You cannot create data sets with the same name.
Creating Live Models

Before you can analyze dashboards, you need create a live model.

**To create a live model:**
Open Sisense and click **Data** in the top menu. Your ElastiCubes and live models are displayed.

Click **+ Live**. The Add new Live Connection dialog box is displayed. In **Title**, enter a name for your live model and click **Save**. An empty model is displayed in the Model Editor.
Adding Live Connections

An alternative to importing data into an ElastiCube is to connect directly to the source through a live data connection. With Live Connect, queries can be processed directly in the data source. This allows you to work with only the result set from a query, rather than doing large imports from the source. You may want to leverage this method if you have invested in a high performance database or want to tap into real-time functionality.

For a short video overview of working with live data, see below.

Each live data source has its own requirements. For documentation for the relevant live data sources, see the table below:

<table>
<thead>
<tr>
<th>Connector</th>
<th>More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google BigQuery</td>
<td>Documentation</td>
</tr>
<tr>
<td>MemSQL</td>
<td>Documentation</td>
</tr>
<tr>
<td>MySQL</td>
<td>Documentation</td>
</tr>
<tr>
<td>Oracle</td>
<td>Documentation</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>Documentation</td>
</tr>
<tr>
<td>Redshift</td>
<td>Documentation</td>
</tr>
<tr>
<td>Snowflake</td>
<td>Documentation</td>
</tr>
<tr>
<td>SQL Server</td>
<td>Documentation</td>
</tr>
</tbody>
</table>
Working with Live Models

In the Model Editor, you have a variety of options for working with tables you have connected to from a live data source. You can manage your data on two levels, the Table level and the Column level. Any changes you make to your schema are reflected in the dashboard or widgets of the live model only after you publish the changes.
Table Level

You can manage tables connected to a live model through the Navigation Pane or through the table itself in the schema.

The table below describes the menu options available from the Navigation Pane and the corresponding icons available in the table’s menu. Either method allows you to modify and manage your tables.

<table>
<thead>
<tr>
<th>Navigation Pane</th>
<th>Table Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preview</td>
<td><img src="image" alt="Preview Icon" /></td>
<td>Displays a preview of your table’s data.</td>
</tr>
<tr>
<td>Refresh Schema</td>
<td><img src="image" alt="Refresh Schema Icon" /></td>
<td>Allows you to update a table’s columns if there was a change on the data source without having to add the table again.</td>
</tr>
<tr>
<td>Delete</td>
<td><img src="image" alt="Delete Icon" /></td>
<td>Allows you to delete a table.</td>
</tr>
</tbody>
</table>
To view your table’s metadata, including its name, path, provider, location, and the number of columns it contains, hover over the table in the schema or in the Navigation Pane.
Column Level

Columns that are part of a table are displayed below the table in the Navigation Pane. On the left side of the column is an icon that indicates what type of data is in the column.

On the right side of the column’s name is an icon that indicates if the column is connected to another table through a relationship. You can hover over the icon to view more details or disconnect the relationship.

For each column, you can hover over its name to display a menu from where you view the following options for managing your columns:

<table>
<thead>
<tr>
<th>Menu Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td>Displays related columns together side by side. See Creating Relationships for more information. <strong>For Amazon Redshift and Microsoft SQL Server only.</strong></td>
</tr>
<tr>
<td>Edit Relationship</td>
<td>Allows you to edit a column’s relationship. This option is only available when the column is connected to another table.</td>
</tr>
<tr>
<td>Description</td>
<td>Allows you to tag a column with metadata that you can use to organize your data and search for it later. See Tagging Your Data for more information.</td>
</tr>
</tbody>
</table>
Enabling Relationships between Tables in Live Models

Sisense allows you to enable or disable the creation of relationships between tables in Live data models. By default, creating relationships is enabled for the following databases:

- Amazon Redshift
- Google BigQuery
- MemSQL
- Microsoft SQL Server
- Snowflake

For other databases that support Live models, the Sisense Administrator needs to manually enable relationships between tables.

**Note:** Sisense recommends creating relationships between tables on high-performance databases. If enabling this for other databases, Sisense recommends testing the dashboard performance to ensure that it is not impacted before publishing the dashboard to your users.

For all databases, Sisense Administrators can disable the creation of relationships. In case the Administrator disables this on a data model where relationships were already defined, the users will not be able to create new relationships, but the previously-defined relationships will remain. **Note:** For MemSQL, disabling is currently unavailable.

**To enable/disable relationships between tables in Live models:**

Follow one of the paths for the relevant database:
## Connector and Path

<table>
<thead>
<tr>
<th>Connector</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL, MS-SQL, Oracle, PostgreSQL, Redshift</td>
<td>C:\Program Files\Sisense\DataConnectors\DotNetContainer\Connectors</td>
</tr>
<tr>
<td>Google BigQuery, MemSQL, Snowflake</td>
<td>C:\Program Files\Sisense\DataConnectors\JVMContainer\Connectors</td>
</tr>
</tbody>
</table>

Open the directory for the connector and open the relevant file:

<table>
<thead>
<tr>
<th>Connector</th>
<th>File</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL, MS-SQL, Oracle, PostgreSQL, Redshift</td>
<td>manifest.json file</td>
</tr>
<tr>
<td>Google BigQuery, MemSQL, Snowflake</td>
<td>description.json file</td>
</tr>
</tbody>
</table>

Change the value of the "isLiveRelationsSupported" property to **true**:

![manifest.json](image)

**Note:** Files and settings are overwritten during upgrades.

Save the file.

Restart the following services as described below:

<table>
<thead>
<tr>
<th>Connector</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL, MS-SQL, Oracle, PostgreSQL, Redshift</td>
<td>Sisense.CLRConnectorsContainer service</td>
</tr>
</tbody>
</table>
### Connector | Service
--- | ---
Redshift |  
Google BigQuery, MemSQL, Snowflake | Sisense.JVMConnectorsContainer service

Refresh your browser. When dragging database tables on top of each other in the Data page, the Relationship preview window is displayed. In this window, you can create joins between columns belonging to these tables. For details, click [here](#).

(Optional) To disable relationships, change the value of the "isLiveRelationsSupported" property back to **false** and proceed as above.
Publishing Live Models

After creating and designing a live model, you need to publish it. Publishing a live model adds the model to the list of your data sources from which you can choose when creating a dashboard or changing a data source.

**To publish a live model:**
- In the *Data* page, open the relevant live model.
- In the Model Editor, click 🚀 Publish.
Working with Live Widgets

After you have published your live model, you can begin to create visualizations like you would with any other data source. See Creating Dashboards and Creating Filters for more information.

In Sisense, live data sources have the ⚡ icon next to their name.

While you can create dashboards from a live data model like you would an ElastiCube, you can also add live data models as a data source to a dashboard that already has a data source such as an ElastiCube. When you have multiple data sources working together on a single dashboard, the data source and the widgets built on it operate independently of each other. This allows you to view widgets that visualize time sensitive information in addition to widgets that are refreshed less frequently. For more information about adding multiple data sources, see Changing a Dashboard’s Data Source.

In terms of governance, all existing access rights settings and data security settings can be applied for live connections.
Load Management

The most significant difference between dashboards built on ElastiCubes and dashboards connected to live data sources is how the load is managed. With an ElastiCube, you refresh the data when you build an ElastiCube and query the ElastiCube, not the data sources that define your ElastiCube. With live connections, queries are run against the data source itself. This means that you don’t have to wait for an ElastiCube to finish building before you see the results, however, your data source must be able to handle the potential load of multiple refreshes.

When you refresh the data for a live connection, Sisense uses the connection settings and credentials in the data set to query the live data source. Depending on your Sisense role, there are several options for refreshing the data. Administrators define the refresh rate in seconds and timeout for each data set. After the refresh rate is set by the Administrator, no user can set a refresh rate shorter than the rate set by the Administrator. Administrators can manually refresh the data set from the Data Sources tab in the Admin page.

Keep in mind that each time a refresh takes place, your data source is queried, so your data source needs to handle the refresh rate you define. Designers can reduce the frequency of data refreshes or stop automatic refreshes on a per widget level, but they cannot increase the frequency of refreshes beyond the threshold set by an Administrator.
Refreshing a Data Set

There are several ways in which you can refresh a widget built on a live data model depending on your Sisense user role. Administrators set the default refresh and can refresh the data manually from the Data Sources tab in the Admin page of Sisense.

Designers can refresh a widget from the widget’s information window.

In addition, Designers can reset the Refresh Rate, however, the rate cannot be lower than the default rate set by the Administrator. Like Designers, Viewers can refresh a widget from the widget’s information window, but they cannot modify the refresh rate.
While any users can refresh a widget, Sisense has a minimum 10 second refresh limit. Any refreshes requested within 10 seconds are ignored.
Canceling a Refresh

If you have a refresh that you want to cancel, you can cancel it by clicking **Cancel** in the widget’s information box.

This button is displayed only when a refresh is in progress.

In addition, any time a user leaves a query, for example, to view another page in their browser, Sisense pauses the query.
Deleting ElastiCubes and Live Models

When you no longer want to use an ElastiCube or live data model, you can delete them from the Data page. Deleting an ElastiCube or live model removes it from the server and the ElastiCube’s folder saved locally on your hard drive. After they have been deleted, they cannot be restored.

If your ElastiCube or live model has any dependencies, or entities such as dashboards, that are currently connected to it, some of these entities will also be deleted and you will be prompted to remap to another data source. Entities that are deleted when you delete an ElastiCube include:

- Data security rules
- Build alerts
- Drill hierarchies

Entities that are not deleted, but should be remapped to another data source include:

- Dashboards
- Widgets (from other dashboards)
- Pulse Alerts

Entities that you can remap to another data source are clickable links that when clicked, display a list of all the entities that need to be remapped.
You can choose to remap the entities to another data source manually or select the new data source from the list on the bottom of the Delete ElastiCube dialog box. If you do not remap the entities, the dependencies will not work until you apply a new data source.

**Note:** Any temporary folders that were saved locally during failed builds are not deleted when the ElastiCube is deleted and must be removed manually.

**To delete an ElastiCube or live model:**

In Sisense, click **Data** in the top menu.

In the **Data** page, click the menu of the relevant ElastiCube or live model and select **Delete**.

Click **Yes** to confirm you want to delete the ElastiCube or live model.

OR

If you ElastiCube or live model has any dependencies, remap those entities or select None to remap them later.
Sisense Server Console

**Note:** This feature is currently available in the desktop version of Sisense. This feature will be migrated into Sisense in the near future. For more information, see [Sisense V7.0 Backwards Compatibility](#).

You can use the Sisense Server Console to perform various management tasks on your ElastiCubes, including:

- Scheduling Data Loads
- Importing and exporting data
- Attaching and detaching directories
- Changing the Location of your ElastiCube Data Storage
- Canceling Running Queries

**To open the Sisense Server Console:**

- Right-click the systems tray and double-click the console icon 📣
- or
- Click on the Windows start menu, and select All Programs > Sisense > Sisense Server Console.

ElastiCube data is stored in a default folder, which you can change in the Sisense Server Console.
Changing the Location of the ElastiCube Data Storage

To change the folder:
From the Windows start menu, open the Sisense Server Console.
In the top panel, click on the cog icon to view/change server preferences.
Click **Browse** next to **Default server data folder**.
Select the location of the folder where you want to store ElastiCube data, and click **OK**.
Canceling Running Queries

You can cancel all running queries in your ElastiCube to free up resources. This is useful in cases where your ElastiCube is consuming too many resources, and as a result, causing the Sisense Web Application to be unresponsive.

**To cancel all queries:**
Click on the Windows start menu, and select **All Programs > Sisense > Sisense Server Console**.
Click the relevant ElastiCube to view the menu panel.
Click **Cancel All Queries**.
Dashboards

After importing data into Sisense, the next step is to create dashboards that viewers can interact with.

This section will show you how to get started working with your dashboards.
Creating Dashboards

Designers and Dashboard Designers can create dashboards from the Analytics page where all your existing dashboards are displayed.

When you create a dashboard, your dashboard is initially empty. The next step is to add widgets, or visualizations, to your dashboard to make it meaningful. The topics below provide information about creating dashboards, adding widgets to it, and customizing the data displayed in the dashboard.

- Creating Dashboards
- Adding Widgets to a Dashboard
- Managing Widgets
- Filtering Dashboards
- Formulas
- Working with Time
- Customizing the Dashboard Layout
- Changing the Color Palette
- Changing a Dashboard’s Datasource
Managing and Sharing Dashboards

As a Designer, you can manage dashboards that you have created, or that have been shared with you; these topics explain how.

Organizing Dashboards
Locating Dashboards
Deleting Dashboards
Copying Dashboards
Sharing Dashboards
Interacting with Dashboards

Now that you've created your dashboards, it's time to start interacting with them.

The topics below will help you learn how to get the most out of your dashboards.

- Viewing Dashboards
- Interacting with Filters as a Viewer
- Changing Sisense's Language
- Drilling Down in a Widget
- Making Selections in a Widget
- Exporting Dashboards
Creating Dashboards

There are two ways you can create dashboards: the first is through the Sisense Analytics page, and the second is through the Sisense REST API. The difference between these two methods is that the Analytics page provides an interface where you populate the dashboard with widgets. To create a dashboard through the REST API, you define the dashboard and then populate it with widgets that already exist through their widget IDs.
Creating a Dashboard

Each Dashboard is based on the data selected from one or more ElastiCubes or Live data sources. The instructions below assume that the required ElastiCube(s) and Live data model(s) have already been created. You can also use the sample ElastiCubes that are installed with Sisense if you have not yet created your own. For a short video overview of creating dashboards, see below:

To create a dashboard:
In the Analytics page, click + above the Dashboards list.
OR
Right-click on the folder menu and select New Dashboard.

The following window appears:

Click the name of the Data Set displayed to select the ElastiCube or Live data model you want to work with. ElastiCubes have the 📝 icon next to their name while Live data models have the ⌘ icon next to their name.
Click the name of the Title, and enter a name for the dashboard. This name will appear at the top of the dashboard, and in the Dashboards list.
Click **Create**. You are then automatically guided through the process of creating your first widget in the Widget Wizard, as described in *Adding Widgets to a Dashboard*. 
Creating Dashboards through the REST API

The Sisense REST API enables you to programmatically create dashboards by defining the basic properties of a dashboard such as Title, datasource, and owner. After defining the properties of the dashboard, you define its visualizations by including the widget IDs of pre-existing widgets.

To create a dashboard through the REST API:
In Sisense, click Admin at the top, and then REST API, followed by REST API Reference.

Select Dashboards then POST /dashboards.
In the body of your call, define the values of the following keys:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>String that defines the name of the dashboard as displayed in the Sisense Web Application.</td>
</tr>
<tr>
<td>desc</td>
<td>String that represents a short description of the dashboard.</td>
</tr>
<tr>
<td>Key</td>
<td>Value</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| datasource | Object that defines the data source where the dashboard pulls its data.  
This object contains the following elements:  
**title**: The name of the data source.  
**id**: The address and name of the data source where the dashboard queries its data from.  
**address**: The address of the dashboard.  
**database**: The name of the data source where the data is pulled from by the dashboard.  
**fullname**: The name of the data source. |
| filters | This object defines the dashboard filters that the user add goes to the "filters" object in the .dash file.  
This object contains the following elements:  
**datatype**: The type of data affected by the filter, for example, text.  
**title**: The name of the filter displayed in the Filters pane. |
| layout | Object that defines what widgets are displayed in the dashboard and their layout.  
This object contains the following elements:  
**type**: String that defines how the layout of the dashboard. This value should be "columnar".  
**columns**: Array objects that defines the width of each column in the dashboard and how widgets are displayed in each column.  
**cells**: Object that contains the subcells.  
**subcells**: Object that contains the elements objects where |
<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>elements</td>
<td>widgets are defined.</td>
</tr>
<tr>
<td>defaultFilters</td>
<td>This object defines the default filters of a dashboard. When a user selects &quot;Set as my default filters&quot; from the Filters menu in Sisense, the filters are defined in this object.</td>
</tr>
</tbody>
</table>

Click **POST**. The Dashboard is added to the Dashboards list in the **Analytics** page.
Adding Widgets to a Dashboard

This topic describes how to add widgets to your dashboard through the Widget Wizard.

The Widget Wizard automatically guides you through a very simple process of creating a widget, while offering the best recommendations for displaying the data that you select.

Almost always, simply accepting the wizard’s recommendations will help you create a great widget. However, if you want more control over your widget’s appearance and behavior, click the Advanced Configuration option in the bottom left of the wizard to access a rich variety of additional design options, as described in "Widget Designer" on page 473.

For a short overview of the process of adding a widget, see below:
Using the Widget Wizard

To add a widget to the dashboard:

If this is your first widget, click + Select Data. The Sisense Data Browser opens, and from there you can select one of the fields (columns) from this dashboard’s ElastiCube or Live data source. If you already have widgets in your dashboard, then click the + Widget button.

**Note:** To select data from a different ElastiCube, click from on the right side of the New Widget window, and select an ElastiCube. You can add a title to your widget now or after creating your widget.

In the Data Browser, select a field to add to your widget. The field that you select is automatically displayed in a suggested widget. When only a single numeric field is selected, it is displayed as an Indicator widget. A single descriptive field is displayed as a Pivot widget.
Single Numeric Field

But your widget is not really interesting yet, because it has only one field!
Repeat the step above to add more fields to the widget.
As you add fields, the relevant options are displayed as buttons to the left of your selection.
The fields that you select are listed across the top left of the wizard.
You can click on each visualization button to display the selected fields in that visualization/chart.
For example, the same widget as above could be displayed as a Pie chart:
Tip: Sisense provides its best recommendation for your widget design. However, if you want more control, you can click the Advanced Configuration option on the bottom left of the window to provide a rich variety of additional design options, as described in "Widget Designer" on page 473.
Data Browser

In the Data Browser, you can select and add columns (sometimes called fields) from an ElastiCube or a Live data source to your widget.

An icon appears to the left of each field to indicate its data type:

- Date
- Alphanumeric string
- Numeric

As you type into the Data Browser, the list is dynamically filtered to only show the fields that contain the text you typed.
Simply clicking a field to add it to the widget generally provides great results, however there are more options in the Data Browser.

Hover over a field in the Data Browser and click **More ...** to display additional aggregation (quick functions) and filtering options.

Click the $\text{fx}$ button to define formulas (free-form expressions) that define the field values and filters of a widget. A rich variety of functions are provided for you to use in the formula that you define.
To learn more about these options, see "Formula Editor" on page 556.
Area Chart

An Area Chart is very similar to a Line Chart except that the areas under each line are filled in (colored), and it is possible to display them as stacked. The chart is recommended for displaying absolute or relative (stacked) values over a time period.
Adding Data

In the Widget Designer, click + Select Data to select the field(s) whose values will be placed on the X-Axis.

Typically, the X-Axis of an Area Chart is used to represent time. You must add at least one item to the X-Axis. You can add at most two items. When you add two X-axis items, the chart’s two X-Axis are combined.

In the Values area, click Add+ to select the field whose value determines the height of the area in the chart. Typically, the Y-Axis of an Areas Chart is used to represent numeric data.

You must add at least one field to Values. When you add more than one Value, then each item is represented in the chart by its own color and area.

Break by (optional): Click Add + to select a field by which to break up (group) the data represented in this chart.

Each field added to Break By is represented by its own area in the chart and is automatically added to the chart’s legend.

This is an optional field, and operates under the following conditions:

You can add at most one Break by field.

The Break by option is available only when a single field was added to Values (Y-Axis).

You can add only a single field to Values, if the Break by option was used.
Designing the Area Chart

To fine-tune the appearance of the Area Chart widget, open the widget in Edit mode and on the right pane you can modify your Area Chart with the following options:

**Area Types:** Select how areas are represented in the widget:

**Classic:** Areas overlap, meaning that smaller areas appear on top of larger areas and cover them. The colors indicate the distribution between the values.

**Stacked:** Areas are stacked on top of each other and do not overlap. Hover over the columns to see the percentage distribution among the
values.

Stacked 100: Areas are stacked on top of each other (but do not overlap). The combined area is stretched to the top in order to represent 100%. This option is most commonly used when the relative
distribution of the values is more important than their aggregation.

**Line Type:** Select how the line appears in the widget: **Straight** (Default) or **Smooth**.

**Line Width:** Select either **Thin**, **Bold** (Default) or **Thick**.

**Legend:** Specify whether to show or hide the Legend and its position.

**Value Labels:** Specify whether to show or hide labels showing values in the Area Chart.

**Markers:** Specify whether to show or hide markers over data points.

**X-Axis:**
- **Grid Lines:** Shows (Default) / hides.
- **Labels:** Shows (Default) / hides.
- **Title:** Select the checkbox to display the x-axis title. To edit the title, type in a new title. Click outside the text box to apply the new value.

**Y-Axis**
- **Grid Lines:** Shows (Default) / hides.
Logarithmic: Displays using orders of magnitude.
Labels: Shows (Default) / hides.
Title: Select the checkbox to display the y-axis title. To edit the title, type in a new title. Click outside the text box to apply the new value.
Values on Axis: To change the default minimum, maximum or interval values on the axis, type in the new values, and click outside the text boxes to apply the new values. To restore any of the values to their default states, click on the reset button.
Bar Chart

The Bar Chart is commonly used to compare many items. The Bar Chart typically presents categories or items (descriptive data) displayed along the Y-Axis, with their values displayed on the X-Axis. You can also break up the values by another category or groups.
Adding Data

In the Categories panel, click Add+ to select the field(s) whose values will be placed on the Y-Axis.
Typically, the Y-Axis of a Bar Chart is descriptive data. You must add at least one item to Categories, and a maximum of two items. When you add two fields to Categories, the chart’s Y-Axis are combined.
In Values, select the fields whose values determine the length of each column. Typically, the values of a Bar Chart are used to represent numeric data.
You must add at least one field to Values. When you add more than one value, each item is represented in the chart by its own color and area.
(Optional) Select a field by which to break (group) the data represented in this chart by clicking Add+.
Each group is represented by a different column and is automatically added to the chart’s legend.
Break by is optional, and operates under the following conditions:
You can add at most one Break by field.
The Break by option is available only when a single field was added to Values (Y-AXIS).
You can only add a single field to Values if the Break by option was used.
Designing the Bar Chart

Fine-tune the appearance of the Bar Chart widget. The design options are the same as for Column Chart.

Additional options to customize the design of the Bar Chart are available from the Sisense user community. For example, to sort stacked Bar Chart by category total (rather then by Y-axis values, the default sorting), see here.
Box & Whisker Plot

The Box & Whisker Plot, or Box Plot, widget is a convenient way of visually describing the distribution, variability, and center of a data set along an axis. Box Plots are divided into four quartiles. The middle quartiles are represented by a box that contains 50% of the data and the median value. The upper and lower quartiles contain the maximum and minimum values and the remaining 50% of the data. These quartiles are represented by lines called whiskers. The maximum and minimum values can be adjusted when defining your box plot. By default, these values are within 1.5 times of the IQR (Interquartile Range), however you can set them to the actual maximum and minimum values or to within one standard deviation of the mean of your data.

Box plots divide five descriptive statistics into four equal quartiles. You can view these statistics by hovering over the Box Plot.
For a short video overview of the Box and Whisker plot, see below.
Adding Data

In the **Categories** panel, click **Add +** to select the field whose values will be placed on the X-Axis. The X-Axis of a box plot is used to represent numeric data.

In the **Values** panel, click **Add +** to select the field whose values will be placed on the Y-Axis.
Design the Box Plot

You can fine tune the appearance of the Box Plot widget with the following tools.

**Whisker Values:** Defines how the whiskers are calculated and displayed.
- IQR (Interquartile Range): Default setting. The whisker values are calculated within 1.5 times the IQR (the difference between the 1st and 3rd quartiles).
- Extremums: The whiskers' values include the maximum and minimum values in the calculation.
- Standard Deviation: The whiskers' values are calculated to one standard deviation above and below the mean of the data.

**Box:** Specify whether to display a full or hollow box.

**Value Labels:** Specify whether to show or hide labels indicating the values of each point (circle). Select the orientation of these labels: **Horizontal** (Default), **Diagonal** or **Vertical**.

**Legend:** Specify whether to show or hide the Legend, and select its position.

**Show Outliers:** Specify whether to show or hide labels indicating the values of each point (circle). Select the orientation of these labels: **Horizontal** (Default), **Diagonal** or **Vertical**.

**X-Axis and Y-Axis:**
- **Grid Lines:** Shows (Default) / hides.
- **Labels:** Shows (Default) / hides.
- **Title:** Select the checkbox to display the axis title. To edit the title, type in a new title. Click outside the text box to apply the new value. To change the default minimum, maximum or interval values on the axis, type in the new values, and click outside the text boxes to apply the new values.
- **Values on Axis:** To restore any of the values to their default states, click the reset button.

**Auto Zoom:** When a widget contains more data than can comfortably be displayed in one view, Auto Zoom will resize the chart to include more data, and in some cases adds a zoom bar under the widget. The zoom bar enables you to scroll right and left and to zoom in/out of different parts of the width of the widget.
Calendar Heatmap

The Calendar Heatmap widget visualizes values over days in a calendar-like view, making it easy to identify daily patterns or anomalies.

For a short video overview of calendar heatmaps, see below.
Adding Data

In the **Date** panel, select the date field that you want to be used as the calendar.
The calendar’s range will be between the minimum and maximum date values within the field.

In the **Color** panel, click **Add +** to select the field that determines which days to color.
Designing the Heatmap

Fine-tune the appearance of the Calendar Heatmap widget, using the following tools.

**Months:** Select how many months should be presented at once: 1 (One Month View), 3 (Quarter View), 6 (Half-Year View) or 12 (Year View). When more months exist than currently visible, paging arrows are displayed to move between months.

**Calendar Type:** Select between two types of view modes:
- Classic – Each month is presented in typical calendar format.
- Week View – Months are ordered by weekdays from top to bottom.

**Week Begins On:** Select the first day of the week to be represented in the calendar.
More Configuration Options

Grey Out Weekends – If on, the weekend days will be colored grey in the calendar.
Show Day Number – Toggle whether to show the calendar day numbers inside the widget.
Show Day Name – Toggle whether to show the day name abbreviations on the calendar.
Column Chart

The column chart can be used in different business scenarios, especially for comparing items, and comparing data over time. The chart can include multiple values on both the X and Y-axis, as well as a break down by categories displayed on the Y-axis.
Adding Data

In the Categories panel, click Add + to select the field(s) whose values will be placed on the X-Axis. Typically, the X-Axis of a Column Chart is descriptive data. You must add at least one item to Categories, and at most, two items. When two Category items are added, then the chart’s X-Axes are automatically grouped. To change the order of the categories, drag a category up or down in the Categories list.

One Category Item
In the **Values** area, select the field whose value determines the height of the columns by clicking **Add +**. Typically, the Y-Axis of a Column Chart is used to represent numeric data.

You must add at least one field to **Values**. When more than one Value is added, then each item is represented in the chart by its own color and area.

**Break by**: Select a field by which to break (group) the data represented in this chart by clicking **Add +**.

Each group is represented by a different column and is automatically added to the chart’s legend.

This is an optional field, and operates under the following conditions:

- At most one **Break by** field can be added.
- The **Break by** option is available only when a single field was added to **Values** (Y-Axis).
- Only a single field can be added to **Values**, if the **Break by** option, described below, was used.

Define filters for the widget, as described in [Filtering Dashboards](#).
Designing the Column Chart

Fine-tune the appearance of the COLUMN CHART Widget, using the following tools:

**Column Type**: Select how columns are represented in the Widget:
- **Classic**: Columns are displayed side by side.
- **Stacked**: Columns are stacked on top of each other and do not overlap. The tooltips over the columns show the percentage distribution among the Values.
- **Stacked 100**: Columns are stacked on top of each other (but do not overlap) and the combined column is stretched to represent 100%. This option is most commonly used when the relative distribution of the values is more important than their aggregation.

**Legend**: Specify whether to show or hide the Legend and its position.

![Column Chart Options](chart_options.png)

![Column Chart Examples](column_chart_examples.png)
**Value Labels:** Specify whether to show or hide labels showing values in the Column Chart. You can also select the angle of the labels.

**X-Axis:** Enable or disable the following options:
- **Grid Lines:** Shows (Default) / hides.
- **Labels:** Shows (Default) / hides.
- **Title:** Select the checkbox to display the X-Axis title. To edit the title, type in a new title. Click outside the text box to apply the new value.

**Y-Axis:** Enable or disable the following options:
- **Grid Lines:** Shows (Default) / hides.
- **Logarithmic:** Displays using orders of magnitude.
- **Labels:** Shows (Default) / hides.
- **Title:** Select the checkbox to display the y-axis title. To edit the title, type in a new title. Click outside the text box to apply the new value.

**Values on Axis:** To change the default minimum, maximum or interval values on the axis, type in the new values, and click outside the text boxes to apply the new values. To restore any of the values to their default states, click on the reset button.

**Auto Zoom:** When a widget contains more data than can comfortably be displayed in one view, Auto Zoom will resize the chart to include more data, and in some cases adds a zoom bar under the widget. The zoom bar enables you to scroll right and left and to zoom in/out of different parts of the width of the widget.
Indicator

The Indicator widget provides various options for displaying one or two numeric values as a number, gauge or ticker. It also provides the option to add additional titles and a color-coded indicator icon representing the value, such as a green up arrow or a red down arrow.

For a short video overview of indicators and gauges, see below:
Adding Data

Click **Add +** in the **Value** panel to select one field whose name and value will be shown as the **Main Title** and **Main Value**, as shown below.

Click **Add +** in the **Secondary** panel to select one field whose name and value will be shown as the **Secondary Title** and **Secondary Value**, as shown below.

**Numeric Indicator**

![Numeric Indicator Diagram]

**Gauge Indicator**

![Gauge Indicator Diagram]
For a gauge, select the minimum and maximum values that can be represented by the gauge. These values can either be set as a fixed numeric value that you specify or as another numeric field that you select using the Data Browser (thus making the value dynamic).

In an Indicator widget, you can also define whether an icon appears in the Widget and the icon's color. For example, you can define a red arrow pointing downwards for a negative value. You may refer to Defining Conditional Coloring – Condition Tab for more information.
Designing the Indicator

Fine-tune the appearance of the Indicator widget, using the following tools:

**Indicator Type**: Select whether the Indicator appears in **Numeric** form or as a **Gauge**.

<table>
<thead>
<tr>
<th>Numeric</th>
<th>Gauge</th>
</tr>
</thead>
</table>

**Note**: If you significantly shorten the height of an Indicator widget in the dashboard, it automatically turns into a Ticker type Widget. You may refer to [Ticker Widgets](#) for more information.

**Skin**: Select the skin of the gauge.

**Components**: Select which labels to include in the widget.

**Indicator Sub Type**: (for numeric indicator only): You can choose a another skin option.
Line Chart

The line chart can be used for various business cases, including:
Comparing data over time, for example: to analyze sales revenue for the past year.
Comparing changes over the same period of time for more than one group or category. Example: Analyze expenditures of different business units for the past year.
Adding Data

Add data to the X-axis.
Click **Add +** to select the field(s) that will be represented on the X-Axis. Typically, the x-axis of a Line chart is used to represent time. You must add at least one item to the x-axis, and at most, two items. When you add two x-axis items, the chart’s x-axes are combined and a separate plot (line) is displayed for each unique value of the second selected x-axis.

Add data to the Y-axis.
In the **Values** area, click **Add +** and select the field whose value determines the height of the lines in the chart. Typically, the Y-Axis of a line chart is used to represent numeric data. You must add at least one field. If you add more than one field, then each item is represented in the chart by its own line and color.
Another version of the same chart includes the following differences:
The second value has been changed into a column chart. The second value has been added to the right axis. This is a useful option when the values are using a considerably different scale.
Both options are available by right-clicking on the value.

(Optional) Select a field by which to break up the data. Click Add +.
Each item that is a member of the field added to Break by is represented by its own line in the chart, and is automatically added to the chart’s legend.
You can add only one Break by field.
The Break by option is only available when you have a single field in the Y-Axis.
Designing the Line Chart

Fine-tune the appearance of the line chart, using the following tools:

**Line Width**: Select either **Thin, Bold** (default) or **Thick**.
**Line Type**: Select how the line appears in the widget: **Straight** (default) or **Smooth**.
**Legend**: Toggle the legend on or off, and select its position in the chart.
**Value Labels**: Toggle labels on or off, and define how labels are displayed: **Horizontal, Vertical** or **Diagonal**.
**Markers**: Select whether to show or hide markers over data points. Select **Full** or **Hollow**, and **Small** or **Large** markers.

**X-Axis**:
- **Grid Lines**: Show (default) or hide grid lines.
- **Labels**: Show (default) or hide labels.
- **Title**: Select the checkbox to display the x-axis title. To edit the title, type in a new title. Click outside the text box to apply the new value.
- **Values on Axis**: To change the default minimum, maximum or interval values on the axis, type in the new values, and click outside the text boxes to apply the new values. To restore any of the values to their default states, click on the reset button.

**Y-Axis**
- **Grid Lines**: Show (default) or hide grid lines.
- **Logarithmic**: Displays a logarithmic scale instead of a linear scale on the axis.
- **Labels**: Show (default) or hide labels.
- **Title**: Select the checkbox to display the y-axis title. To edit the title, type in a new title. Click outside the text box to apply the new value.
- **Values on Axis**: To change the default minimum, maximum or interval values on the axis, type in the new values, and click outside the text boxes to apply the new values. To restore any of the values to their
default states, click on the reset button. 

Y-Axis

- Grid Lines
- Logarithmic
- Labels
- Title: Total # Visits
- Min: Auto
- Max: Auto
- Interval: Auto
Pie Chart

The Pie chart is used to display proportional data, and/or percentages.
Adding Data

In the Categories panel, click **Add +** to select the data that will be displayed in the Pie Chart. You can add only one field.

Typically, the categories of a Pie Chart contain descriptive data. **Note:** All slices that are smaller or equal to 3% are grouped together into a slice called **Other.** You can hover over this slice to display its breakdown.

In the **Values** area, select the field whose values determine the size of the pie slice by clicking **Add +.** Values are typically used to represent numeric data. **Note:** You can add only one field.
Designing the Pie Chart

Fine-tune the appearance of the Pie Chart widget, using the following tools:

**PIE TYPE:** Select how the pie appears.

<table>
<thead>
<tr>
<th>PIE TYPE</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>classic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>donut</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ring</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Legend:** Specify whether to show or hide the Legend, and select its position. To add a title to the legend, so that users understand clearly which dimension is displayed in the chart, see [here](#).
Labels: Select which labels to display on the pie chart, as indicated below:

Note: Sisense rounds up numbers in Pie charts and this may lead to the percentages totaling to 101%.
Exploring Pie Charts

You can drill down into pie charts just like other widgets. In addition, when your pie chart has so many smaller slices that they are aggregated into a single slice called Others, you can click this slice to display a second pie chart containing a breakdown of the Others slice.

For example, the pie chart below displays a breakdown of the top 5 slices in a pie chart with many smaller slices included in the Others slice.

By clicking the Others slice, you can see a breakdown of all the slices included in the Others slice.

By default, the Others slice of the Pie Chart is set to include all categories below 3%. You can manually change this to another value. The change will be applied to all your Pie Charts. For details, click here.
Pivot

Pivot tables are one of the most useful widgets for visualizing data. They enable you to quickly summarize and analyze large amounts of data.
Adding Data

Select the data to appear in the Pivot.

In the **Rows** panel, click **Add +** to select the field(s) whose values will be placed in the rows of the Pivot table.

<table>
<thead>
<tr>
<th>ROWS</th>
<th></th>
<th>VALUES</th>
<th></th>
<th>COLUMNS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age Range</td>
<td></td>
<td>Total Revenue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0-18</td>
<td></td>
<td>1,527,753</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19-24</td>
<td></td>
<td>3,859,903</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25-34</td>
<td></td>
<td>4,877,854</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35-44</td>
<td></td>
<td>14,764,570</td>
<td></td>
<td></td>
</tr>
<tr>
<td>45-54</td>
<td></td>
<td>6,951,359</td>
<td></td>
<td></td>
</tr>
<tr>
<td>55-64</td>
<td></td>
<td>3,203,916</td>
<td></td>
<td></td>
</tr>
<tr>
<td>65+</td>
<td></td>
<td>4,574,272</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

When more than one Row is selected, the Rows are broken into sub-rows (groups). For example, the following shows **Condition** added as a second Row:

<table>
<thead>
<tr>
<th>ROWS</th>
<th></th>
<th>VALUES</th>
<th></th>
<th>COLUMNS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age Range</td>
<td></td>
<td>Condition</td>
<td>Total Revenue</td>
<td></td>
</tr>
<tr>
<td>0-18</td>
<td>New</td>
<td>645,828</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Refurbished</td>
<td>68,360</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unspecified</td>
<td>255,534</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Used</td>
<td>558,031</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19-24</td>
<td>New</td>
<td>1,100,968</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Refurbished</td>
<td>119,606</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unspecified</td>
<td>159,512</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Used</td>
<td>2,479,817</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In the **Columns** panel, click **Add +** to select the field(s) whose values will be placed in the columns of the Pivot table.

You can drag and drop the fields that you added to **Columns** to the **Rows** area and vice versa.

In the **Values** panel, select a field whose values will appear in the Rows and Columns of the Pivot table.
**Tip:** Right-click the value to add data bars to your pivot table.
Subtotals

You can add subtotals to one or more rows, and define how to calculate subtotals.

**To add subtotals to rows:**
Click on the menu of the Row in the Data Panel, or right-click on the row name (header) in the widget, and select Subtotals.

![Subtotal example](image)

**To define how to calculate the subtotals:**
Click on the menu of the Value in the data panel, or right-click on the value name (header) in the widget, and then click **Subtotal by** and the method.

![Value example](image)

The default option is Auto, which aggregates all the data. The other custom options (Sum, Min, Max, etc.) calculate only the values in the rows above the subtotal.

Look at the following example:
In the top example, using the **Auto** option, the subtotal for average revenue aggregates all the sales and revenue data, thus representing a true subtotal of the average revenue for the Asia region.

In the bottom example, **Average** is used to calculate the subtotal, and therefore calculates the average of all the average revenue values in the rows above the subtotal.
Grand Totals

Grand totals aggregate the rows in the pivot table.

**To add grand totals:**
Click on the menu of the row in the data panel, or right-click on the row name (header) in the widget, and then click *Grand total*.

Grand totals is effected by subtotals if defined. In the following example, the Grand total for Average Sales is 10, representing the average sales for two regions (Asia and Europe). In the example below, Average Sales has a subtotal (see above) set to Sum, and therefore the Grand total is the total of the rows above.

<table>
<thead>
<tr>
<th>Region</th>
<th>Total Sales</th>
<th>[Average Sales]</th>
<th>Total Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia</td>
<td>65</td>
<td>13</td>
<td>1,650</td>
</tr>
<tr>
<td>Europe</td>
<td>35</td>
<td>7</td>
<td>250</td>
</tr>
<tr>
<td>Grand Total</td>
<td>100</td>
<td>10</td>
<td>1,900</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Region</th>
<th>Total Sales</th>
<th>[Average Sales]</th>
<th>Total Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia</td>
<td>65</td>
<td>13</td>
<td>1,650</td>
</tr>
<tr>
<td>Europe</td>
<td>35</td>
<td>7</td>
<td>250</td>
</tr>
<tr>
<td>Grand Total</td>
<td>100</td>
<td>20</td>
<td>1,900</td>
</tr>
</tbody>
</table>
Designing the Pivot Table

Fine-tune the appearance of the PIVOT table, using the following tools.

**Page Size**: Specify how many rows appear in each page. Paging options are provided accordingly.

**Colors**: Select the properties in the table to which you want to add color.

![COLORS]

Select how the rows and columns of the Pivot table are highlighted.
Exception Highlighting

Conditional formatting can be used for exception highlighting in a Pivot table. For example, as shown below:

<table>
<thead>
<tr>
<th>Age Range</th>
<th>Condition</th>
<th>Total Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-18</td>
<td>New</td>
<td>1,209</td>
</tr>
<tr>
<td></td>
<td>Refurbished</td>
<td>292</td>
</tr>
<tr>
<td></td>
<td>Unspecified</td>
<td>360</td>
</tr>
<tr>
<td></td>
<td>Used</td>
<td>1,913</td>
</tr>
<tr>
<td>19-24</td>
<td>New</td>
<td>2,544</td>
</tr>
<tr>
<td></td>
<td>Refurbished</td>
<td>592</td>
</tr>
<tr>
<td></td>
<td>Unspecified</td>
<td>569</td>
</tr>
<tr>
<td></td>
<td>Used</td>
<td>3,790</td>
</tr>
<tr>
<td>25-34</td>
<td>New</td>
<td>5,950</td>
</tr>
<tr>
<td></td>
<td>Refurbished</td>
<td>1,401</td>
</tr>
<tr>
<td></td>
<td>Unspecified</td>
<td>1,407</td>
</tr>
<tr>
<td></td>
<td>Used</td>
<td>8,831</td>
</tr>
</tbody>
</table>

See [Defining Conditional Coloring – Condition](#) for more information.

**Note:** For more styling options, including font style and colors, see [this article](#).

**Limitations**

Sisense supports the exporting of pivot tables of up to 1.5 million cells. Attempting to export a higher number of cells might result in a timeout. The following properties of a pivot table increase the probability of reaching a timeout when dealing with very large tables:

- Sub totals or Grand totals
- Complex formulas in Values
- Data Security rules
- Widget-level filters
To successfully export pivot tables, you are advised to split pivot table with many columns to separate pivot tables.
Polar Chart

Use the polar (radar) chart to compare multiple categories/variables with a spatial perspective in a radial chart.
Adding Data

Click **Add +** to select the field whose values will be represented as columns in the circle. Each Category gets its own data point. Typically, the Categories of a Polar Charts are descriptive data.

In the **Values** area, click **Add+** to select the field whose values determine the distance of the column from the center of the polar circle. This is typically used to represent numeric data.

When you select more than one value, they are stacked on top of each other. Select a field by which to break (group) the data in this chart by clicking **Add +**.

Each field added to Break by is represented by its own column and color in the chart and is automatically added to the chart’s legend. This is an optional field. You can add at most one **Break by** field. The **Break by** option is available only when a single field was added to **Values** (Y-AXIS).
Designing the Polar Chart

Fine-tune the appearance of the Polar Chart widget, using the following tools.

**Polar Type**: Select the type of chart.

**Legend**: Specify whether to show or hide the LEGEND, and select its position.

**Value Labels**: Specify whether to show or hide labels indicating the values. Select the orientation of these labels: **Horizontal** (Default), **Diagonal** or **Vertical**.

**Categories**:
- **Grid Lines**: Shows (Default) / hides.
- **Labels**: Shows (Default) / hides.

**Axis**:
- **Grid Lines**: Shows (Default) / hides.
- **Logarithmic**: Displays using orders of magnitude.
- **Labels**: Shows (Default) / hides.
- **Title**: Select the checkbox to display the axis title. To edit the title, type in a new title. Click outside the text box to apply the new value.

**Values on Axis**: To change the default minimum, maximum or interval values on the axis, type in the new values, and click outside the text boxes to apply the new values. To restore any of the values to their default states, click on the reset button. ☑️
Scatter Chart

The Scatter Chart displays the distribution of two variables on an X-Axis, Y-Axis, and two additional dimensions of data that are shown as colored circles scattered across the chart:

**Point**: A field that for each of its members a scatter point is drawn.

**Note**: The maximum amount of data points is 100,000.

**Size**: An optional field represented by the size of the circles. If omitted, all scatter points are equal in size. If used, the circle size is relative to their value.
Adding Data

In the **X-Axis** panel, click **Add +** to select the field whose values will be placed on the X-Axis.

Typically, the X-Axis of a Scatter Chart is used to represent numeric data. Alternatively, Descriptive data is also supported. You can only add one field.

In the **Y-Axis** panel, click **Add +** to select the field whose values will be placed on the Y-Axis.

Typically, the Y-Axis of a Scatter Chart is used to represent numeric data. Alternatively, descriptive data is also supported. You can only add one field.

In the **Point** panel, click **Add +** to select the point field.

The point area determines the field to be scattered across the chart as circles; one point (circle) appears on the chart for each member of this field. These fields must be descriptive data (not numeric data).

You can only add a point if either the X-Axis or the Y-Axis was defined to represent numeric data.

**Break by/ Color**: Select a field by which to break (group) the fields in the chart. This must be a descriptive field. If you select descriptive data, then the points in the chart are grouped into the members of the field. Each member is represented by a different color.

OR

Select a field by which to color the fields in the chart. This must be a numeric field. If you select numeric data, then the point color is a gradient where the highest value is the darkest color and the lowest value is the lightest color.

In the **Size** panel, select the field that determines the size of the circle. It must be numeric data.
Designing the Scatter Chart

Fine tune the appearance of the SCATTER CHART widget, using the following tools.

- **Legend**: Specify whether to show or hide the Legend, and select its position.
- **Marker Size**: Change the size of the circles. If you are not using a Point value, the slider will change the size of all the points on the map. If you are using the Point value, drag each side of the slider to determine the relative minimum and maximum circle sizes.
- **Value Labels**: Specify whether to show or hide labels indicating the values of each point (circle). Select the orientation of these labels: **Horizontal** (Default), **Diagonal** or **Vertical**.
- **X-Axis** and **Y-Axis**:
  - **Grid Lines**: Shows (Default) / hides.
  - **Logarithmic**: Displays using orders of magnitude.
  - **Labels**: Shows (Default) / hides.
  - **Title**: Select the checkbox to display the axis title. To edit the title, type in a new title. Click outside the text box to apply the new value. To change the default minimum, maximum or interval values on the axis, type in the new values, and click outside the text boxes to apply the new values.
  - **Values on Axis**: To restore any of the values to their default states, click on the reset button. 🔄
Scatter Map

Scatter Maps allow you to visualize geographical data as data points on a map. The map distinguishes data using different colors and sizes for the data points on the map.

For a short video overview of scatter maps, see below:
Adding Data

In the Location panel, you can add any field(s) that contain geographic data, and the map will find the most appropriate point. Supported Location types:
   - Country
   - City
   - State/Province
   - Latitude/Longitude (for example: 37.5601, -122.50131, but not 37.5601° N, 122.50131° W).

For example, if you have a country field, using it by itself will visualize the countries on the map.
If instead you have a city field, using it will visualize the cities on the map automatically.
Improving Geographic Accuracy

**Use multiple geographic fields**
To improve accuracy, it is recommended to use more than one geographic level. For example, if you have both a country and a city field, add both of them to the **Location** panel.

**Define the location type**
Clicking on the geographic field’s menu will present a location type sub-menu, which contains the following options:
- Auto
- Country
- State/Province
- City

By default, **Auto** is always selected. This option always tries to figure out which fits best. For better accuracy, you can specify the appropriate location type.
Using Latitude and Longitude data

To visualize latitude and longitude data, you have to add one field containing latitude data, and another field containing longitude data, in this order.

**Color**

Select a numeric field here to color the scatter points on the map according to their value and color rule.

The value of the field is displayed in the tooltip of the scatter point on the map. Hover over the point to see it.

**Size**

Select a numeric field here to make the size of the scatter points on the map relative to their value in the field’s value range.

**Details**

You can select additional data (numeric or descriptive) that will appear when hovering over the data point in the map.
Designing the Scatter Map

Fine-tune the appearance of the Scatter Map Widget, using the following tools.

**Marker Style**: Choose between various marker styles.

![Marker Style Image]

**Marker Size**: Select the relative size for the markers.

![Marker Size Image]

When the Size has a field in it, the slider has two levers controlling the minimum and maximum size of the points.
Sunburst Widget

The Sunburst widget is similar to a pie chart but is multi-dimensional. Whereas a pie chart combines one field and one numeric value, the Sunburst widget can display multiple rings, one for each field. Each ring in the Sunburst shows a breakdown of its parent ring slice.

For a short video overview of sunburst charts, see below:
Adding Data

In the Categories panel, select the fields whose values will be represented as slices of the ring.
Typically, the Categories of a Sunburst widget contain descriptive data. Multiple fields can be added, so that each field is represented as a ring whose values are broken up by its parent ring slice. You can reorder the fields using drag and drop.

In the Values panel, click Add + to select the field whose values determine the size of the ring. Values are typically used to represent numeric data. You can only add one field.
Designing the Sunburst Widget

Fine-tune the appearance of the Sunburst widget using the following tools.

**Center**: Select which info is presented in the center of the Sunburst widget when hovering over a ring slice:
- **Value**: Show the actual numeric value associated with the selected slice.
- **Contribution**: Show the contribution (percentage) of the selected slice to the entire field.
- **Contribution to parent**: Show the contribution (percentage) of the selected slice to its parent ring slice.

**Tooltip**: Determine whether to show the actual value or contribution in the slice tooltip when hovering over it.

**Legend**: Specify whether to show or hide the Legend, and select its position in the widget.
Table

The Table widget displays a broader view of your data, presenting raw and non-aggregated data in columns, with as much fields and metrics as needed. For a short video overview of the table widget, see below:
Adding Data

Click **Add +** to select the field(s) that you want to add to the table. To rearrange the order of the columns, click and drag a column up or down.
Designing the Table Panel

Fine-tune the appearance of the Table widget, using the following tools:

**Borders**: Select the borders to display in the table.

**Column Width**: You can resize the column widths as follows:
- **Fit to Window Size**: Columns are resized automatically to fit the contents in the window.
- **Fit to Content**: You can manually resize the column widths by dragging the column separator.

**Colors**: Select whether to color (highlight) alternating columns and rows, and the table header.

**Word Wrap**: Select whether to wrap texts in headers and rows.

**Infinite Scroll**: Toggle the Infinite Scroll option On or Off.
On: Include all items in a single page with scrolling. Additional data loads as you scroll down in the page.
Off: Specify how many rows you want in a single page. Above the specified number, you will have paging options.
Infinite Scroll in the Off position.
Adding Text Widgets

You can add Text widgets to create titles and texts that stand out, or when you want to add more descriptive explanations to your dashboard and surrounding visualizations.

In the Text widget’s settings, you can find plenty of options for creating different text styles. Some examples of what you can do include changing the font color, selecting a background color, adding hyperlinks, and defining the text alignment.

To add a Text widget:
Click on the Text Widget icon in the top menu.

A new Text widget appears in your dashboard.
Type in your text. As you begin to type, a tool bar with text and formatting options is displayed.

Click outside of the widget area to save the text and formatting.
Examples

To select a font style:
Click anywhere in the relevant paragraph.
From the Styles list, select one of the five predefined styles.

To add a hyperlink:
Select the word or text fragment to which you want to add a link.
Click the hyperlink icon and enter the URL.
You can also select which text to display instead of the selected (linked) text,
and whether to open the link in the same window, or in a new window.

To change the background color:
Click on the paint bucket icon to open the color palette.
Select a color from the palette, or enter a custom color (Hexadecimal format).
Treemap

The Treemap is a multi-dimensional widget that displays hierarchical data in the form of nested rectangles. This type of chart can be used in different scenarios, for example, instead of a column chart if you have to compare too many categories and sub-categories.

For a short video on treemaps, see below.
Adding Data

In the **Categories** panel, click **Add+** to select the fields whose values will be represented as rectangles in the chart. Typically, the Categories of a Treemap widget contain descriptive data. You can add up to three fields, so that a rectangle hierarchy is created. You can reorder the fields using drag and drop.

In the **Size** panel, select the field whose values determine the size of the rectangles by clicking **Add +**. Values are typically used to represent numeric data. You can add only one field.

In the **Color** panel, click **Add +** to select the field to use as the color basis for your rectangles.

The following coloring scenarios are supported:
- **Color by a numeric value:** If you choose a numeric value, then each rectangle will be colored according to its value in the field’s range of values. You can manipulate the coloring rules in the color formatting menu.
- **Color by field members:** If you choose a field, then all rectangles that are members of that field will be colored the same. For example, if you choose to color by the field that is also used as the first category, then the entire Treemap will be split into groups of colors, like in the image at the top of this page. If you choose to color by the field that is used as the second category, then all rectangles that share the same value will also share the same color, as in the image below:
**Note:** You can only color by a field that is also used in the Categories panel. If you choose a field that is not there, it will be automatically added.
Designing the Treemap

Fine-tune the appearance of the Treemap widget, using the following tools.

**Captions:** Control which info is presented in the treemap. A checkbox will be visible for each field that is used in the Categories panel. Turning a field on/off will determine if the field member name will be visible inside the treemap. Note that this info will always be available in the tooltip when hovering over a rectangle, regardless of the on/off setting.

**Tooltip:** Select whether to show the actual value or contribution in the rectangle tooltip when hovering over it.
Widget Designer

The Widget Designer lets you fine-tune a widget’s appearance and behavior. For a short video overview of this process, see below:
Opening the Widget Designer

To display the Widget Designer:
On the dashboard, click the Pencil (Edit) button that appears in the top-right corner of a Widget.

Alternatively, while creating a new widget, in the Widget Wizard, click the Advanced Configuration option that appears at the bottom-left of each window. The Widget Designer is then displayed showing the widget in the
same state as in the Widget Wizard preview.
Navigating the Widget Designer

The Widget Designer enables you to select the data to be included in a widget in addition to providing a variety of options for customizing the visualization used to show the data.

Selecting the Widget Visualization: Enables you to change the visualization of the widget.

Adding Data to the Widget (Data Panel): Used for selecting the values that you want to appear in the widget and those needed for grouping the data. The options differ depending on the visualization (chart type).

Previewing the Widget: The center of the window displays the current design and content of the widget, which automatically changes each time you select a different option in this window.

Adding a title to the dashboard: Click Set a Title and type in a new title for the widget. You can also add a widget title directly from the dashboard without entering the Widget Designer. Click here to learn more.

Accessing More Options: Displays a menu of additional options for the widget.

Filtering the Widget (Filter Panel): Lets you manage the filters that affect this widget. Learn more in the filtering page.

Designing the Widget (Design Panel): Provides a variety of options for fine-tuning the appearance of the Widget, including labels, legends, line types and more. Some of these options need to be turned on for you to
configure them. The options differ depending on the visualization (chart type).

**Update on every change**: Selected – Data changes are automatically and dynamically updated in the display. Not selected – you have to click **UPDATE** to display your latest changes in the widget. Clearing this checkbox may be useful when you are working with very large datasets where query times might be slower.
Fine-tuning a Widget

To learn more about adding data and fine-tuning the design of a specific widget, click on a widget from the list below.

- Indicator
- Column Chart
- Line Chart
- Area Chart
- Area Map
- Bar Chart
- Pie Chart
- Polar Chart
- Scatter Chart
- Scatter Map
- Pivot
- Sunburst
- Treemap
- Calendar Heatmap
- Table
- Box & Whisker Plot

See also Additional Widget Design Options and Extending Dashboard Functionality with JavaScript.
Selecting Colors in Widgets

Different field values are represented by different colors. The Widget Designer automatically assigns these colors and displays them in the widget legend that is displayed in the Widget Designer and in the dashboard.

The Widget Designer provides the following options for manipulating these colors:

- **Single Color**: For selecting a specific color for all values
- **Range**: For selecting a range of colors
- **Conditional**: For defining colors based on conditional data.
Selecting a Specific Color – Single Color Tab

The following procedure describes how to assign a specific color to a data value that does not change. This color only applies to the widget in which it is defined.

**To select a specific color:**
Click on a color in the Widget Designer Data Panel to display a color picker window.
Select the **Single Color** tab.
Select a color and click **OK**.

**Note:** To select a custom color that does not exist in the palette, enter a hexadecimal value into the # Custom field.
Selecting a Range of Colors – Range Tab

The **Range** tab is available for the following widgets:
- Column Chart and Bar Chart
- Scatter Map and Area Map

The following procedure describes how to assign a range of colors to a field according to its value.

**To select a color range:**
- Click on a color in the Widget Designer Data panel to display a color picker window.
- Select the **Range** tab.
- By default, the range configuration is set to ‘Auto’, which automatically assigns a color range.
- Select **Manual** to manually configure the color range, as follows:
  - **Change Left/Right Colors**: Click the color rectangles at the right and left to change the start and end color of the range.
  - **Customize Range**: By default, the color range is set to match the minimum and maximum values of the data. Instead, you can manually set a minimum value. All values that are below this value will be assigned to the color on the left. Accordingly, selecting a Max value means that all values larger than it will be assigned the color on the right. By default, the ‘Middle’ color is grey. All values larger than this will start gradually getting the right color. All values smaller than this will start gradually getting the left color. Changing the ‘Middle’ value controls which value the color grey is assigned to.
  - **Coloring Method**: Selecting **Min/Max** will display the full color range. Selecting **Min** will display all the colors between the minimum and middle values, and higher than the middle value will be gray. Selecting **Max** will display all values between the middle value and the maximum value, while all other values will be displayed in gray.
  - **Transition**: Select Steps to define a distinct number of colors in the range. Select **Gradient** for a linear color range.
Defining Conditional Coloring – Conditional Tab

Define colors based on the value of the field as defined by a logical expression. In addition, for Indicator widgets, you can also define whether an icon appears in the widget and select the icon’s color. For example, you can define a red arrow pointing downwards for a negative value.

For a short video overview of conditional coloring, see below:

To set conditional coloring:

- Click on a color in the Widget Designer Data panel to display a color picker window.
- Select the Conditional tab. Each row in this window represents a condition to which you can apply a value. If the condition is true, then that value is shown in the color that you select in this row. For example, one row can state that a negative value is displayed in red and another row can state that a positive value is green.
- In each row, define an expression to be applied to the value of this field by selecting an operator and specifying a value, as shown below.
- Select a color for the field.
- If required, add and define additional condition rows by clicking + Add condition.
- In the dashboard, these conditions are evaluated in the order in which they appear in this window – from top to bottom. The first condition that is true, determines the color. Reorder the condition rows as necessary by dragging them up or down.
- Click OK.
<table>
<thead>
<tr>
<th>Single Color</th>
<th>Range</th>
<th>Conditional</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Drag up or down

<table>
<thead>
<tr>
<th>Condition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>≤</td>
<td>fx 100</td>
</tr>
<tr>
<td>≥</td>
<td>fx 300</td>
</tr>
<tr>
<td>≥</td>
<td>fx 200</td>
</tr>
<tr>
<td>≥</td>
<td>fx 101</td>
</tr>
</tbody>
</table>
Renaming Fields in Widgets

The names of fields in widgets are taken from the raw data. To make these names more comprehensible, you can rename the fields in your widget.

To rename a field in a widget:
In the Widget Designer, right-click on a field, for example, in the X-AXIS, Y-AXIS, VALUES, CATEGORIES areas and so on.
In the menu, select Rename.

Note: Renaming a field only affects how it is displayed in the current widget. This field’s name is not affected in the Data Browser or in other widgets in the dashboard.
Formatting Numbers in Widgets

You can change the way numbers are displayed and formatted in widgets. For example, you can display a number as a percentage, and you can define the number of decimal places.

**To format numbers in a widget:**
In the Widget Designer, for example, in the X-Axis, Y-Axis, Values, Categories areas and so on, hover over, and click 123.
Select how the number, currency or percentage appears. The preview area at the bottom of this window shows a preview of how each option will appear in the widget.
Locale Settings

The formats for dates, times and numbers in your dashboards are based on your computer’s operating system or browser’s locale settings (depending on which browser you use).

The following locales are supported:
- United States
- United Kingdom
- Israel
- Canada
- South Africa
- Australia
- Netherlands
- Germany
- Ireland
- Mexico
- France
- China
- Brazil

Notes: You can override the default locale settings for all users, by updating a parameter via the Rest API. To use a locale file that is not included in the above list of countries, you can manually add a locale file in the following location:

C:\Program Files\Sisense\app\client\resources\base\localization.

Click here to access Locale files and view their codes.
Currency Settings

The default currency symbol used in the dashboard is the US dollar ($), however you can change the symbol by either selecting another symbol from the drop-down list, or by typing in a different symbol. The list of currency symbols will include common currency symbols, the currency defined in your browser’s locale settings, and the symbol that you added (if you did so).

After you select a different symbol, additional viewers will see the same symbol that you applied.
Changing a Widget’s Visualization

You can change the visualization of a widget (for example, from a Line Chart to a Pie Chart). When you change the type, all relevant values and definitions are transferred to the new visualization.

To change a widget’s visualization:
Open the Widget Designer, as described in Opening the Widget Designer. Click the Visualization selector, shown below, to display a menu of visualization options.

Select a different visualization, and click Apply.
Combining Two Types of Visualizations in a Widget

In chart widgets (such as Line Charts, Area Charts, Column Charts and Bar Charts), you can represent one or more selected field(s) using a different visualization than the visualization of the original widget.

For example, the following example shows a Line Chart in which the **Total Revenue** is still represented as a Line Chart (the default), but the **Total Quantity** is represented as a Column Chart.

To select a different visualization for a field in the widget:
- In the Widget Designer, in the **Values** area, right-click the field that you want to change.
- Select **Series Type** from the menu. A menu of alternative visualization types is displayed.
- Select the visualization to use for this field only.

**Tip:** If necessary, drag the field up or down in the Widget Designer to have a visualization brought to back or front.
**Note:** The chart keeps the same visualization type as was originally created. The Design panel still displays only the design options of the widget's original visualization type.
Creating a Continuous Chart with Missing Date Values

If your data is missing date values, you can add the missing dates to your chart to create a continuous data flow. In addition, you can select whether to display missing dates as gaps in your chart, or include null values as zeros, to avoid gaps, and maintain a continuous data flow.

For example, if your chart begins June 8th and ends June 20th, but there is no value on June 13th, you can choose to include the missing date on the x-axis. You can also choose whether to continue the graph (create zero values) or cut the graph where there are missing values.

The default chart is missing June 13th.

After selecting the Continuous option, the chart includes the missing date(s).
In addition, you can select to continue the line, by treating null (missing) values as zero values.

To create a continuous chart:
In the X-Axis panel, click on the menu, and select **Continuous**.

To treat null values as zero values, click on the menu of the Values panel, and select **Treat Null as Zero**.
Adding Drill Hierarchies to Widgets

Dashboard Viewers have the option to drill down in a widget, and get an in-depth view of a selected value.

While Viewers can select a drill-down path from the complete list of fields, it’s easier to select a commonly needed drill hierarchy from a short list. This is especially true when there is a lot of data, and the Viewer needs to remember specific fields, and select them each time.

As an Administrator, you can add predefined drill hierarchies to widgets.

To enable drill hierarchies in a widget:

In the Widget Designer, click on the Categories menu, and select Hierarchies...
Select the checkboxes next to the drill hierarchies that you want to make available to dashboard viewers.
Note: Hierarchies with a lock icon will always appear in the widget.
unlock this option, please contact your Sisense Administrator.
Disabling the Drill to Anywhere Option

As an Administrator, you can restrict viewers to predefined hierarchies only, by disabling the option to drill down freely to any field.

The drill-to-anywhere option is enabled by default for all users and widgets. You can disable the drill-to-anywhere option in the Sisense web app per widget (see below), or by user role in the REST API.

**To disable the drill-to-anywhere option:**

In the widget menu, deselect the Enable Drill to Anywhere option.

As a result, the Choose Another… option is removed from the widget’s Drill options.

The left image below displays the widget with the option to drill to anywhere. The image on the right displays the widget with only predefined drill hierarchies.
Managing Widgets

This topic describes how you can edit, copy, and delete your dashboard's widgets.
Editing a Widget

When you first create a new dashboard (as described in Creating a New Dashboard) or a new widget (as described in Adding Widgets to a Dashboard), the Widget Wizard automatically guides you through Sisense Web’s best recommendations for visualizing the data that you select. Then, after a widget is created, you can fine-tune/edit it in the Widget Designer. The Widget Designer provides a variety of options for changing the widget’s appearance and behavior.

See Fine-tuning a Widget for more information.
Copying a Widget

You can create a copy of a widget within the same dashboard or another dashboard in your list. This is useful for saving the original widget in the dashboard before making changes. The new widget is separate from the original so no changes to the new widget affect the original widget.

To copy a Widget within the same dashboard:

In Dashboard view, click the widget menu button in the top-right corner of the widget to be copied and select **Duplicate**.

If you are editing the widget, click the widget menu button in the top-right corner of the widget to be copied and select **Save A Copy**.

To copy a Widget to another dashboard:

Select the Header of the widget to be copied and drag it to the dashboard you want to copy it to.
Note: When you copy a widget to another dashboard whose data source is another ElastiCube, the Dashboard filters of the new dashboard will not affect the widget.
Deleting a Widget

To delete a Widget:
Click the widget menu in the top-right corner of the Widget and select Delete. Confirm the deletion.
Field Suggestions

To quickly and easily create meaningful visualizations in your dashboard, Sisense displays field suggestions to Dashboard Designers when creating new widgets.

After creating your first dashboard, Sisense begins to display field suggestions the next time you create a widget.

When determining what fields to suggest, Sisense looks at fields being used in other widgets on the same server, which fields are typically working together in other widgets, and what fields you have already selected.

Fields that are already being used in the widget are not suggested.

Sisense has exposed the POST /suggestions API endpoint in the Sisense 1.0 REST API that enables you to retrieve a list of suggestions based on Sisense’s algorithm for calculating field suggestions. This is useful for displaying suggested fields in
your own application. For more information, see the Sisense REST API documentation.
Introduction to Filters

The Filters panel on the right of the dashboard provides data filtering options that affect the data displayed in the widgets. You can change these filters to focus on more specific data.

Your Sisense user role determines how you can interact with filters. As a Designer you can create filters that are distributed to Viewers who interact with your dashboards. See Interacting with Filters as a Designer. Viewers can interact with filters and save their changes locally. See Interacting with Filters as a Viewer for more information.
Interacting with Filters as a Viewer

Viewers can interact with filters for analyzing data. Through the Filters pane, you can modify filters in a dashboard and change the way values are sorted in a visualization. This is useful for focusing on data displayed in dashboards that have been shared by the dashboard’s owner.

As a Viewer, you can select and deselect filter visuals on a dashboard. Each time you interact with a filter, for example by selecting or entering a value, the filter is immediately applied to your dashboard. After you have configured the relevant filters, you can save it as your default filters view.

Viewers can interact with filters as follows:

- Make a different selection in the filter controls displayed in the Filter pane on the right side of the dashboard, as shown below:
OR
Click on the pencil icon next to the filter name (shown above) in the dashboard to display the Filter Definition window.

In addition, Viewers can perform the following procedures:
Switching Filters On and Off
Saving your Default Filters View

To create a filter you must be a Designer.
Click here to learn more about filters.
Interacting with Filters as a Designer

If you are a Dashboard Designer, you can define both dashboard filters and widget filters, as follows:

**Dashboard Filters:** This type of filter affects all the widgets in the dashboard (except the widgets that you configured as independent). Designers can create dashboard filters and Viewers can view and interact with them.

**Widget Filters:** This type of filter affects only the specific widget it was applied to. A Designer can define which subset of data values is displayed in a widget. Viewers do not see these filters and cannot change them.

**Note:** Dashboard editing rights are assigned by Dashboard Designers and Designers.

The following topics include procedures for:
- Creating Dashboard Filters
- Editing and Deleting a Dashboard Filter
- Creating Widget Filters
- Creating Dependent Filters
- Configuring how Filters Affect the Dashboard and Widgets
- Limiting Filters to Specific Values with Background Filters
- Locking Filters
- Restoring Filters
- Switching Filters On and Off
- Saving your Default Filters View
Creating Dashboard Filters

This topic explains how to create dashboard filters using the different filtering options.

In this section you will learn how to:

- Create a Dashboard Filter
- Create a List Filter
- Create a Text Filter
- Create a Numeric Value Filter
- Create a Top/Bottom Ranking Filter
- Create a Time Filter
- Wildcards
Creating a Dashboard Filter

To create a dashboard filter:
Click **Filter Your Dashboard** in the dashboard's right panel (if it's your first filter), or **+** if you are adding another filter. The Add Filter dialog box is displayed.

<table>
<thead>
<tr>
<th>Add Filter</th>
<th>Sample Healthcare</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type to search for fields</td>
<td></td>
</tr>
<tr>
<td>Admissions</td>
<td></td>
</tr>
<tr>
<td>Admission_Time</td>
<td></td>
</tr>
<tr>
<td>Cost_of_admission</td>
<td></td>
</tr>
<tr>
<td>Death</td>
<td></td>
</tr>
<tr>
<td>Diagnosis_ID</td>
<td></td>
</tr>
<tr>
<td>Discharge_Time</td>
<td></td>
</tr>
<tr>
<td>Doctor_ID</td>
<td></td>
</tr>
<tr>
<td>HAI</td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td></td>
</tr>
<tr>
<td>Patient_ID</td>
<td></td>
</tr>
<tr>
<td>Room_ID</td>
<td></td>
</tr>
<tr>
<td>SSI</td>
<td></td>
</tr>
<tr>
<td>Surgical_Procedure</td>
<td></td>
</tr>
<tr>
<td>Time_Of_Stay</td>
<td></td>
</tr>
<tr>
<td>Conditions Time Of Stay</td>
<td></td>
</tr>
<tr>
<td>Average time of stay</td>
<td></td>
</tr>
</tbody>
</table>

(Optional) If you have multiple data sources, select the data source that contains the fields you want to filter. Filters applied to fields from one data source do not affect fields from another data source even on the same dashboard.
Select the field by which to filter.
Select the type of filter from the pane on the left, such as **List**, **Criteria**, **Calendar** or **Ranking**. The options provided in this window may vary according to the selected field type (Date, Numeric or Descriptive).
Select the required filtering details in the panel on the right to create one of the following types of filters:

- Creating an Include/Exclude Filter
- Creating a Text/Numeric/Date Filter
- Creating a Top/Bottom Ranking Filter
- Creating a Calendar Filter

These are explained in depth below.

Click the **Star Favorite** button to save this filter. You can give your starred filter a unique name by clicking in the text field and editing the given name. This will be useful when you need to quickly add a new filter (with same field type) or update a filter to reflect the unique parameters that you defined in your starred filter.

Note: Your favorite filters are visible to shared users.

Click **OK** to filter the data in the Dashboard and add this filter to the Filters panel on the right of the Dashboard.

After a filter has been added to the Dashboard, it stays displayed on the Dashboard’s Filters panel. Any changes that you make to the filter are immediately reflected in the Dashboard.

Tip: You can rearrange the filters in the Filters panel by dragging a filter above or below another filter.

For a short video overview of dashboard filters, see below.
Creating an Include/Exclude Filter (List Filter)

This type of filter enables you to define whether selected field values are included or excluded.

To define a List filter:
Follow Steps 1 – 3 of Creating Dashboard Filters.
Select the List option in the left panel.

Select/clear the field values to be included/excluded in the Dashboard. Click OK to filter the data in the Dashboard and add a List filter to the Filter panel of the Dashboard.
Selecting Single Selection or Multi Selection

By default, list filters are set to multi-selection. To toggle between single and multi-selection, click the button on the upper right of the list filter:

**Multi Selection:**

<table>
<thead>
<tr>
<th>Add Filter</th>
<th>Sample Healthcare</th>
</tr>
</thead>
<tbody>
<tr>
<td>List</td>
<td></td>
</tr>
<tr>
<td>Values</td>
<td></td>
</tr>
<tr>
<td>Ranking</td>
<td></td>
</tr>
<tr>
<td>Starred</td>
<td></td>
</tr>
<tr>
<td>Advanced</td>
<td></td>
</tr>
</tbody>
</table>

- Values: 1
- Ranking: 2
- Starred: 3, 4, 5, 6, 7, 8, 9, 10
- Advanced: 11

OK
<table>
<thead>
<tr>
<th>List</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Values</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Ranking</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Starred</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11</td>
<td></td>
</tr>
</tbody>
</table>

Single Selection:
Creating a Text Filter

This type of filter enables you to filter according to text matching.

To define a Text filter:
Select the Text tab.
Follow steps 1 – 3 of Creating Dashboard Filters. Select a Textual field.
Select the operator, for example Starts with or Containing.
Type in the value by which to filter.
If you require additional conditions, click + Add condition. Select AND or OR to define how filter operators are combined.
Creating a Numeric Value Filter

This type of filter enables you to limit your dashboards to specific value ranges.

For example:
- Keep only sales above 100$.
- Keep only product IDs between 1000020 and 1000030.

To define a Numeric filter:
Select the Values tab.
Follow steps 1 – 3 of [Creating Dashboard Filters](#).
Select the operator, for example Equals or Greater than.
Type in the value by which to filter.
If you require additional conditions, click + Add condition. Select AND or OR to define how filter operators are combined.
Creating a Top/Bottom Ranking Filter

This type of filter enables you to select whether to include only the top/bottom ranking fields.

To define a Ranking filter:
Follow steps 1 – 3 of Creating Dashboard Filters.
Select the Ranking filter tab.
Select Top or Bottom and the number of items to include.
In the By field, select a field by which to rank the fields. A list of the fields that match these definitions is displayed so that you can verify that the results are as you expect.
Click OK to filter the data in the Dashboard and add a Ranking filter to the Filter panel of the Dashboard.
Creating a Time Filter

There are two types of time filtering options, Calendar and Dynamic Time filters. **Note:** Unlike other filters that you define, viewers will be able to change the type of date filter, for example, from Time Frame to Calendar, etc., providing more flexibility for viewing date ranges.
Calendar Filter

This filter enables you to select custom date ranges from a calendar.

**To define a Calendar filter:**
Follow steps 1 – 3 of [Creating Dashboard Filters](#). Select a date field. Select the Calendar filter tab. Use the quick navigation menu to jump to a point in time. Select a date range using any of the following methods:
- Use the calendar to select a date range to filter by. Use the left calendar to select the starting date of the range, and the right calendar to select the end date.
- Type in the dates.
- Use a shortcut such as Earliest Date or Today.

**Note:** Earliest Date and Latest Date refer to the earliest and latest dates with data.

Click **OK** to filter the data in the Dashboard and add a date filter to the Filter panel of the Dashboard.
Dynamic Time Filter

This filter enables you to filter dashboards to preset dynamic time frames, such as 'Last Year', 'Last 2 Years', '2 Year ago', etc.

**To define a dynamic time filter:**
- Follow steps 1 – 3 of [Creating Dashboard Filters](#). (Select a date field)
- Select the **Time Frame** filter tab.
- Use the desired option.
- Click **OK** to filter the data in the Dashboard and add a date filter to the Filter panel of the Dashboard.
Advanced Filtering Criteria

If you require advanced filtering options that are not included in the provided settings (for dashboard or widget), you can add or edit filters in the Advanced section.

To add or edit filtering criteria:
- Click Advanced in the left panel.
- Add or edit filtering criteria. Click Test to see the results of your new/edited filter.
- Click OK to add the filter.

Example 1 – Adding an OR statement
The example below shows an OR statement being used to display multiple names. After clicking Test, the matches are displayed to the right of the script.

Examples using JAQL Queries
Example 2 – Filtering by Measures
This option lets you filter dimensions by measures, returning only the filtered members and associated values.

In this example, all products with a total price equal or greater than 1 million will be returned.

The query:

```
"datasource": "laptopsales",
"metadata": [
  {
    "dim": "product"
  },
  {
    "dim": "price",
    "agg": "sum",
    "filter": {
      ">=": 1000000
    }
  }
]
```

Results include the relevant product names and price, for example:

- Product A, 1000000
- Product B, 1551246

**Example 3 – Filtering by Dimension Attributes**

This option lets you filter dimensions by filtered attributes of other dimensions, returning only the filtered values.

In this example, all customers that purchased Product A and Product B will be returned.
{  
  "datasource": "laptopsales",
  "metadata": [  
    {  
      "dim": "customers",
      "filter": {  
        "attributes": [  
          {  
            "dim": "products",
            "filter": {  
              "members": ["Product A", "Product B"]
            }
          }
        ]
      }
    }
  ]
}

Note that you can add as many nested filters as needed.
Updating on Every Change

When this option is on, any changes you make to the dashboard filters are updated immediately, and the dashboard is refreshed to reflect the changes. If you are working with large or complex data sets that result in longer refresh times, then switch off this option. Make all your changes, and click the Update button when you are ready to update your dashboard.

**Note**: This option is available for Designers only. Sisense viewers can make changes that will update automatically on very change.
Wildcards

To filter lists or text quickly, you can use wildcards such as ‘_’ or ‘%’, which will be interpreted as literal characters or an empty string.

Sisense supports the following wildcards:

**Single Characters**

You can enter an underscore ‘_’ as a placeholder for a single unknown character.

**Multiple Characters**

You can use a percent % symbol to search for an undefined length of string.
Escape Characters

If you are searching for a string with an underscore or percent symbol, add a backslash (\) in front of the special character. Backslashes are treated as an escape character by Sisense.

For example, to locate results for the string “_te”, use a backslash before the string:
Edit A Filter

List

Text

Ranking

Starred

Advanced

Filter field items:

Containing \_\_te

Add condition

Previewing all 1 results

my_text

All items containing "\_\_te"
Editing and Deleting a Dashboard Filter

This topic explains how to edit a dashboard or delete a dashboard filter.
Editing a Dashboard Filter

To edit a dashboard filter:

Make a different selection in the filter controls displayed in the Filter panel on the right side of the dashboard, as shown below:

OR

Click on the pencil icon next to the filter name (shown above) in the dashboard to display the Filter Definition window, as described in Creating
Dashboard Filters.
Deleting a Dashboard Filter

To delete a filter:

Click the bin icon. To temporarily disable a widget’s filter, you can toggle the filter on or off using the toggle switch. Delete the widget only if you do not need it any longer.
Creating Widget Filters

This topic describes how you can use the Widget Designer to create widget filters and add widget filters to an existing field.

In the Widget Designer, you can define a filter for the data in a widget. For example, you can create a filter that only shows the fields of relevant countries. The filters are not visible in the dashboard, nor can they be edited directly from the dashboard.

Note about duplicate filtering: When creating a widget filter for a field that is already filtered in a dashboard filter, the widget filter will override the dashboard filter. Example: You have a dashboard filter, filtering the months field, with the values January and February selected. At the same time you create a filter for a widget, also for months, but this time March and April are selected. When both filters are applied in the dashboard, you will see data for March and April in the widget, according to the widget’s filter. If your widget is filtered using measured values, then the measured value will override any other widget or dashboard filters you have for the same fields.
Creating a Widget Filter

To add a widget filter:

Open (edit) this Widget in the Widget Designer, as described in Adding Widgets to a Dashboard.

Click the Filters tab in the Widget Designer. The Filters panel is displayed showing the filters defined for this Widget. Click on the Add Filter button to display the Data Browser, which lists the fields in the ElastiCube.

Click on a field in the Data Browser to add it as a filter. You can define multiple filters, each for a different field. These filters have an AND relationship between them. Each filter that you define appears on the Widget Filters panel on the right, enabling you to easily edit it if needed. The procedure for defining a Widget filter is similar to the procedure for defining a Dashboard filter, as described in Creating Dashboard Filters. The difference is that the widget filter only affects its hosting widget, and is accessible only via the widget designer, not directly from the dashboard.
Adding a Widget Filter to an Existing Field

Adding a new widget filter, as described above, lets you use any field as a widget filter. Sometimes you want to directly filter a field that is already part of the widget and visible in the data panel on the left.

To add a Widget Filter to an existing field:
Hover over the field you want to filter, and click the FILTER icon.
Creating Dependent Filters

This topic describes dependent filters. Dependent Filters combine multiple dashboard filters into a hierarchy of dependent filters that affect each other from parent, to child, to grandchild and so on. When you filter the parent filter, fewer options will be available in the child filter.

**Example Use Case**

The following example shows how using a dependent filter helps you easily focus information in a dashboard.

This example describes an online travel agency with 250,000 hotels in its database. An agent wants to find a specific hotel, but does not remember its name. To narrow the results, the agent can define a dependent filter. This dependent filter is comprised of four sub-dependent filters defined with the following hierarchy:

- **Destination**: Displays a list of States and Countries. In this example, the agent will select the Destination Florida.
City: Because Florida was selected, only Destinations in Florida are displayed in the child filter, such as: Miami, Orlando and Tampa. In this example, the agent will select the City Orlando.

Resort: Because Orlando was selected, only resorts in Orlando are displayed in the grandchild filter, such as Floridays, Buena Vista, Walt Disney World. In this example, the agent will select the Walt Disney World resort.

Hotel: Because Walt Disney World was selected, only the Hotels in Walt Disney World are displayed in the great-grandchild filter.

Note: By default, filters are not dependent. Dependent Filters can be defined only from selection type filters.

To define a Dashboard Dependent Filter:

The top level (most inclusive) filter of the Dependent Filter must first be defined in the usual manner (see Creating Dashboard Filters). This newly added filter then appears on the dashboard.

On the parent filter, click + or click ➕→ Add a dependent filter.
Define the child filter in the usual manner (see Creating Dashboard Filters). The number of available options for selection in the child filter are less because of the selections in the parent filter. The child filter appears on the dashboard with a down arrow indicating the dependent filter and its parent...
Repeat this step for each new child filter to create as many sub-levels of Dependent Filters as you need.
Configuring How Filters Affect the Dashboard and Widgets

This topic explains how dashboard filters work together with widget filters, and how you can configure different filtering behaviors and their effects on the dashboard.

The following procedures are explained below:

- Configuring how Dashboard Filters Affect a Widget
- Defining how a Widget Behaves when Filtered
- Defining how a Selection on the Widget Affects the Dashboard
Configuring how Dashboard Filters Affect a Widget

Dashboard filters affect all the widgets in the dashboard, except widgets that you have defined to be independent. An independent widget is not affected by dashboard filters.

**Note:** Dashboard filters are case-sensitive. For example, if you have two columns with the same name, but one is uppercase and the other is lowercase, a dashboard filter applied to one will not be applied to the other.

**To make a widget filter-independent:**
Open (edit) this Widget in the Widget Designer, as described in Adding Widgets to a Dashboard. The top part of the Widget Filters panel on the right of the Widget Designer lists the Dashboard Filters that are defined for this Dashboard.

![Filters Panel]

Turn off the Dashboard filters that you do not want to affect this Widget:
The toggle on icon (default) indicates that the Widget is affected by this Dashboard filter.
The toggle off icon indicates that the widget is not affected by this Dashboard filter. You also have the option to make this Widget independent of all current and future Dashboard Widgets by turning the Dashboard
Filters option off, as shown below:

<table>
<thead>
<tr>
<th>Filters</th>
<th>Design</th>
</tr>
</thead>
</table>
| Dashboard Filters | ![Filter Off](image)
| Slice/Filter | ![Filter On](image) |
| Highlight | ![Highlight On](image) |
Defining how a Widget Behaves when Filtered

When fields that are visible in a widget are filtered, it can be displayed in two ways – Slice/Filter or Highlight. For example, this column chart shows revenue for different age categories:

If the chart is set to Highlight mode, filtering a subset of age categories will highlight them in the chart:
If the chart is set to **Slice/Filter** mode, the same filter will remove all non-selected age categories from the chart:

To define the filter behavior mode:
Select **Slice/Filter** or **Highlight** in the Filters panel on the right side of the dashboard.
Defining how a Selection on the Widget Affects the Dashboard

When making a selection on a widget, a filter is added to the dashboard with the selected part of the chart as the applied filter.
To disable this option for a widget, open the widget in editing mode, and from the menu options, disable **Widget Affects Dashboard Filters**.
Limiting Filters to Specific Values with Background Filters

This topic describes how you can leverage background filters to limit what data is exposed to your Viewers or to reduce the amount of filtering a Viewer needs to do.

As a Designer, you can define which of the fields in a filter will be visible to Viewers. After defining which fields will be included in the filter, viewers will be able to use the updated and restricted filter to further filter the dashboard or widget using the available fields.

This can be useful in cases where Viewers only require selected data, and then they do not have to search through a long list of fields, or for security purposes in cases where you do not want to expose all the data.

**Note:** You must have access to edit a dashboard to view this option.

**Example 1 – Simple List Filter**

A filter includes a list of 10 countries. As a designer, you can limit the filter to include only five specified countries. Viewers will see the filter as having only five countries to include or exclude in the filter.

**Example 2 – Conditional Filter**

Another example requires having only the top 5 countries by population included in the available list of countries. In this case, Viewers will only see five countries, however, the list of countries will depend on the population sizes and any changes that may affect which countries are in the top 5 at the given time.
To create a background filter:
Create a background filter using one of the following methods:
In a filter without selected values, open the filter's menu and click **Create background filter**. Define your filter settings, and click **OK**.
OR
If you made changes to your filter (either by editing the filter or by selecting values directly in the filter), you can open the filter's menu, and click **Set as background filter**.
The background filter is indicated by a filter ▼ icon.

**Note:** The icon is not visible to viewers.

To edit a background filter:
In the filter’s menu, click **Edit background filter**.
Update the filter settings, and click **OK**.

To remove a background filter:
In the filter’s menu, click **Remove background filter**.
Locking Filters

As a Designer, you can lock a filter, preventing Viewers from making any changes. The Viewer will be able to see the defined filter settings, but the filter will be disabled for editing.

**Note:** You must have access to edit a dashboard to view this option.

**To lock a filter:**
- Click the Filter menu.
- Click **Lock**.
  A lock icon appears to indicate that the filter is locked.
- To unlock the filter, click on the menu, and select **Unlock**.
Restoring Filters

As a Designer, if you changed the dashboard’s filters, you can revert to the latest copy shared by the dashboard’s owner at any time. When you restore a dashboard, local changes to a dashboard by Viewers are overridden and the default dashboard is restored for all users.

Under the Dashboards list, click on the menu of the dashboard and select **Restore Dashboard**.
Switching Filters On and Off

You can easily toggle filters on and off using the toggle switch. Use this option to compare states (with and without the filter), or to temporarily disable a filter, rather than deleting it.
Saving Your Default Filters

You can save the current state of your filters and their settings at any time. The current state includes the existing filters, their configuration, and the order in which they appear in the filters panel. After making changes to any of the above settings, you will be able to restore your filters to their previously saved state.

To save your current set of filters:
In the Filters menu, click Set as My Default Filters.

To restore a saved filters set:
Click the restore icon next to the Filters menu.
Introduction to Formulas

This topic provides an overview to working with formulas in Sisense along with important tips and examples.

Formulas are custom calculations performed on one or more fields in your data. They offer an important way to analyze results and express business logic.

Sisense’s formula capabilities are designed around several principles;
- Create complex business calculations without IT or technical knowledge.
- Easily combine fields from different data sources together.
- Customize formulas to reflect specific criteria and conditions.
- Work with raw data without the need to summarize data sets before creating formulas.
- Instantly recalculate formulas based on any filter, variable or level of granularity.

The table below provides a reference to the main formula functions available in Sisense. Click here to see a complete reference.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Function</th>
<th>Types and Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perform calculation based on criteria</td>
<td>Measured value</td>
<td><strong>Value Filters:</strong> ≠, =, &gt;, &lt;, between</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Text Filters:</strong> Contains, Doesn't Contain, Doesn't End With, Doesn't Start With, Ends With, Start With, Equals, Not Equal</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>List Filter:</strong> Include, Exclude</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Ranking Filters:</strong> Top, Bottom Ranking</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Time Filter:</strong> Date and Calendar</td>
</tr>
<tr>
<td>Combine data/apply simple mathematics</td>
<td>Aggregate functions</td>
<td><strong>Operator:</strong> +,-,*,/</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Aggregate:</strong> Sum()</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Average:</strong> Avg()</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Count:</strong> Count(), DupCount()</td>
</tr>
<tr>
<td>Goal</td>
<td>Function</td>
<td>Types and Syntax</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>Summarize data</td>
<td>Statistical function</td>
<td><strong>Range</strong>: Max()/Min()</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Central Tendency</strong>: Median(), Model, Largest()</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Std Deviation and Variance</strong>: Stdev(), Stdevp(), Varp(), Var()</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Quartile and Percentile</strong>: Quartile(), Percentile()</td>
</tr>
<tr>
<td>Accumulate data</td>
<td>Rolling sum/average</td>
<td><strong>Sum to Date</strong>: YTDSum(), QTDSum(), MTDSum()</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Avg to Date</strong>: YTDAvg(), QTDAvg(), MTDAvg()</td>
</tr>
<tr>
<td>Compare Time or Trends</td>
<td>Time functions</td>
<td><strong>Past Periods</strong>: PastYear(), PastQuarter(), PastMonth(), Next(), Prev()</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Growth Trend</strong>: Growth(), GrowthRate()</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Time Difference</strong>: YDiff(), QDiff(), MDiff(), DDiff(), HDiff(), MnDiff(), SDiff()</td>
</tr>
</tbody>
</table>

The following topics explain how to create formulas, and describe the features of the formula editor:

- Using the Formula Editor
- Creating and Editing a Formula
- Reusing Formulas
- Using Quick Functions
- Creating Formulas Based on Criteria and Conditions (Filters)
- Building Formulas with Functions
- Function Reference
Formula Editor

The Formula Editor is where Designers define formulas for a dashboard's widgets.

If you are familiar with the Formula Editor, see the Quick Reference Guide. For a detailed guide with examples, continue reading below.

**To open the Formula Editor:**

Do one of the following:

- For a new widget, click **Select Data**, and then $\text{fx}$.
- For an existing widget, click on the edit formula button $\text{笔}$.  

The Formula Editor has two tabs, the Data Browser to select fields and the Functions tab to select formula operations. You can create a formula combining one or more function, field and filters. The diagram below
Functions are operations which perform different calculations, for example a sum. Use the 'Jump To' menu or the search box to quickly find the formula you need.

Fields in the Data Browser are variables contained in the data set (ElastiCube). Clicking on a field in the data browser will include it as part of the formula.

Filters can be applied to restrict formulas based on criteria.

Starring is a way to save a formula for later use.
The Formula Editor window can be expanded by clicking the expand button at the top right.
Creating and Editing a Formula

The Data Browser enables you to define formulas (freeform expressions) that define the values and filters of a widget. A rich variety of functions are provided for you to use in the formula that you define.

To define a formula:

Open the formula editor in the Data Browser:

For a new widget, click Select Data, and then $\text{fx}$.  
For an existing widget, click on the edit formula button.

The Data Browser then changes to display the Formula Editor, which has two tabs: Data Browser and Functions. The Data Browser tab provides fields to choose from. The Functions tab lists the functions that you can include in your formula by selecting them. You can read a description of each function in a tooltip by hovering over it.

Define the formula as follows:

From the **Data Browser** tab, select one or more fields.
From the **Functions** tab, select the required functions.

Type in the required parts of the formula. To see examples, see [Formulas Based on Criteria and Conditions](#), and [Functions to Build Formulas](#).

Click **OK**.

**To edit a formula:**

You can easily edit formulas using the right-click options. They include:

- **Rename**: Rename the formula, for example, give a name that represents a real-life task or expected result from the formula, or include in the name filters that you have added to the formula.
- **Filter**: Add filters to the formula.
- **Type**: Change the default aggregation method, for example, from Sum to Average.

The following image shows the right-click options.
The following image shows right-click options for a filtered formula. Read more below.
Reusing Formulas

This topic describes how you can reuse formulas that you have previously marked as a favorite (starred).

Formulas are saved per ElastiCube.

**Important:** Changing a starred formula **does not** affect previous uses of that formula. These will continue to use the old version of the formula. Only future uses of the starred formula will implement your latest formula.

**To mark a formula as a Favorite:**
- While defining a formula, click the Favorite (Star) button.
- Enter a name for this Formula.
Click **OK**.

To reuse a favorite formula:
Favorite Formulas appear in the Data Browser under the title Formulas, as shown below. Simply select it to use it.

```
SUM(CYMDDC)
```
Using R in Formulas

R is a software environment for statistical computing and graphics. Sisense supports the integration of R functions in your formulas. You can write R code directly in the formula editor, and send fields as parameters. You can also combine R functionality with the regular functions to create advanced and tailored formulas. (For general help with the formula editor and functions, click here.)

Before you can integrate R into your formulas, you must have an R server set up in your organization.

This document will cover the following:
- Understanding How R works in Sisense
- Connecting Sisense to Your R Server
- Using R functions in Formulas
- Guidelines for using R code in Sisense
- Example 1: Simple R Function – Kmeans Clustering
- Example 2: R Logistic Regression via Sisense
- Example 3: Loading an Existing R Model
- Example 4: Building and Saving a R Model using Data from Sisense
Understanding How R works in Sisense

Within a widget, R code is used and assigned fields as parameters, the following happens:

- The selected fields are sent fully to R as a list of arrays.
- The R code is executed on the R server, referencing the fields as needed.
- The numeric results from R are sent back to the widget for visualization.

Each request from the R server is done in a self-contained namespace, thus R variables and results cannot be reused across requests unless they are loaded from a saved location.

R calculations are computed in the R server, and not in the ElastiCube, this means that for larger data sets results will not return as fast as regular widgets.

Running R on Windows

If your R server is running on Windows you will be bound to the concurrency limitations the R distribution has on Windows. This means that concurrent requests to R may return wrong results. Sisense recommends using R on a Linux machine for production environment with concurrent user activity.
Connecting Sisense to Your R Server

To connect your Sisense instance to your R server:
Open the Sisense Server Console from the start menu.
Click on the Server Preferences icon to open the server settings.
In the Rserve field, enter the IP address of your R server, and select the checkbox to enable the connection.

You can test the connection by clicking on the Test button.

Installing Rserve
Sisense can only connect to a running Rserve instance. For more information about installing and running Rserve follow this link.
Please note that if you’re using RStudio, Rserve still has to be installed.
Using R functions in Formulas

Using R code in Sisense is achieved by using two dedicated formulas – RINT & RDOUBLE. Depending on the result type of your R code, use the appropriate function within the formula editor.

For either formula, RINT or RDOUBLE, the return type has to be an exact match.

Multiple numeric field values can be passed as parameters to the R functions, which within your R code can be accessed via the ‘args’ argument (See K-means example below).

**Syntax:**

RINT([recycle (true)], [Ordering], <R expression>, [numeric value 1>, ..., <numeric value n>])

RDOUBLE([recycle (true)], [Ordering], <R expression>, [numeric value 1>, ..., <numeric value n>])
Sisense does not support the use of a final ‘;’ in R statements. If you add a semi-colon at the end of your R statement, the code will fail.

Parameters:

**Recycle true/false (default = true)** – This is an optional parameter that controls whether the results from R should be recycled (cached), so that consequent queries will not have to be recalculated unless they or the data have changed. Generally, this behavior is automatically managed by the ElastiCube automatically for all functions. However, since R code might have non-deterministic components to it (such as randomality functions or date-specific functions), the ElastiCube cannot rely on a data-set and function that hasn’t changed not to return a different result in multiple executions.
By default, the Recycle value is set to true. Use ‘off’ if your R code contains randomality or other non-deterministic content.

**Ordering** – This is an optional parameter that defines the sort order in which numeric data is sent to R. The argument of the Ordering parameter can be an index in your data source or you can use the ORDERING() function to determine the order of your fields. This function arranges the values of the arguments into ascending or descending order, breaking ties by further arguments.
For example:
ORDERING([Total Sales], -1*[COUNT Salesman], MIN(<Office Name>))
For more information about the ORDERING() function, click here.

**R expression** – Your R code needs to be passed here, wrapped in double quotes. R expects the return type to be an array with the same size as widget’s row count. Nulls will be used to make up for shorter arrays, and longer arrays will be trimmed.
Use single quotes to wrap strings within your R code when using the Rint/Rdouble functions, so that there will be no double-quote collision with the quotes wrapping your R code within the Rint/Rdouble function.

**Numeric Value Arguments** – Numeric values can be passed as arguments to your R code.
All arguments are passed to R as a 1-based list named “args”. Each item in the list contains an array that represents the field.
For example:
args[[1]] will return an array which represents the first field that was used as an argument.
args[[2]][3] will return the 3rd data value within the 2nd field that was used as an argument.
Guidelines for using R code in Sisense

In order for your R code to work well in Sisense you need to make sure of the following:

- The resultset returned from R into Sisense has to be equal in size to the dataset that is sent to R as parameters.
- The order of the resultset return from R into Sisense has to match the incoming order. No assumptions can be made about the incoming order – This means that you cannot assume that the incoming order matches the order you visually see in the widget.
Example 1: Simple R Function – Kmeans Clustering

**Description:** In the example below the Total Cost and Total Revenue fields from Sisense are being used to cluster data via a Kmeans function. The result will cluster the data based on the kmeans settings, in this case 4 clusters.

**Sisense Syntax:**

```r
RINT(TRUE, "m<- log(matrix(unlist(args), ncol = 2)); kmeans(m,4)$cluster", [Total Cost],[Total Revenue])
```

**Result:** The R expression will return a result from 1 to 4. The widgets color settings can be configured to color corresponding values based on if they equal 1, 2, 3 or 4.

**Tip:** You can save (star) the complex formula above, and use it again in additional formulas and charts.

In the chart’s BREAK BY/COLOR panel, you can change the color of the clusters as well as define the number of clusters and their break points.
Example 2: R Logistic Regression via Sisense

**Description:** In the example below website traffic data is used to predict if a conversion is likely to occur based on the source of web traffic. The following fields contained in the ElastiCube are passed to the R logistic model: Average Outcome (1=conversion, 0=no conversion), Average Time on Page, Average Pages (viewed) and Average Bounce Rate. This data is passed to a model to predict the outcome and likelihood the traffic will convert.

[For the sake of illustration both the training and test data set are the same in the example]

**Sisense Syntax:**

```R
RDOUBLE(TRUE,"mydata<-data.frame(convert= args[[1]], time= args[[2]], pages= args[[3]], bounce= args[[4]]); lrmodel <- glm(convert ~ time + pages + bounce, data = mydata, family = 'binomial'); prob <- predict(lrmodel, newdata = mydata, type = 'response')", [Average Outcome], [Average Time on Page], [Average Pages], [Average Bounce Rate])
```

**Result:** The R expression will return a result from 0 to 1 showing the likelihood of a conversion occurring. Note a similar process can be applied to create a linear regression.
Predicted outcome of website conversion based on logistic regression.
Example 3: Loading an Existing R Model

Description: In the example below a saved logistic regression built on previous training data is called and used with new website traffic data to predict if a conversion occurs. The following fields contained in the ElastiCube are passed to the saved R logistic model: Average Outcome (1=conversion, 0= no conversion), Average Time on Page, Average Pages (viewed) and Average Bounce Rate. This data is passed to a model to predict the outcome and likelihood the traffic will convert.

Sisense Syntax:
RDOUBLE(TRUE,"mydata<-data.frame(convert=arg[1],
time=arg[2], pages=arg[3],
bounce=arg[4]);load('C:\rdata');prob <-
predict(lrm;model, newdata = mydata, type = 'response')",
[Average Outcome], [Average TimeonPage], [Average Pages],
[Average BounceRate])

Result: The R expression will return a result from 0 to 1 showing the likelihood of a conversion occurring based on the loaded R model in this case stored in ‘rdata’.
Example 4: Building and Saving a R Model using Data from Sisense

Description: In the example below a logistic regression to predict if a website conversion will occur is built using data from Sisense and then saved as an R model. The following fields contained in the ElastiCube are passed to create the R logistic model: Average Outcome (1=conversion, 0= no conversion), Average Time on Page, Average Pages (viewed) and Average Bounce Rate. This data is saved as an rdata file and can be used on other data sets to predict outcomes.

Sisense Syntax:
RDOUBLE(TRUE,"mydata<-data.frame(convert=args[[1]],
time=args[[2]], pages=args[[3]],
bounce=args[[4]]);lrmodel <- glm(convert ~ time + pages +
bounce, data = mydata, family = 'binomial');
save(lrmodel, file = 'C:\rdata') ;1", [Average Outcome],
[Average TimeonPage], [Average Pages], [Average BounceRate])

Result: The R expression will use Sisense data to build and save a logistic model in the specified file in this case 'rdata'.
Creating Formulas Based on Criteria and Conditions (Filters)

Often formulas must take into account specific criteria. To do this Sisense provides a feature called **Measured Value**, which like the SUMIF function in Excel, only performs a calculation when the values meet a set of criteria. Criteria for Measured Values may be based on any logical operators in a filter.

*Measure Value Syntax:*
(Measure, Scope1, Scope2...ScopeN)

*Parameters*
Measure: A field measure or formula.
Scope: A filter including Value, Text, List, Ranking and Time filters.

To filter the formula:
In the Data Browser, create your formula from the Data Browser and Functions, as explained in [Formula Editor](#). Add the field (criteria) by which you want to filter the formula. Right-click the field and select **Filter**.
You can then filter the formula by listed items, text options, ranking, etc. When done, click **OK**.

A simple example of Measured Value is the use of a list filter. A marketing team may need to count leads generated for a specific region such as North America. Even if leads come from many different countries, the measured value calculates leads generated only when the lead originates from the United States or Canada.

<table>
<thead>
<tr>
<th>Country</th>
<th># Leads</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>15</td>
</tr>
<tr>
<td>France</td>
<td>20</td>
</tr>
<tr>
<td>Canada</td>
<td>5</td>
</tr>
<tr>
<td>UK</td>
<td>30</td>
</tr>
</tbody>
</table>

**Example 1: Measured Value with List Filter**

(Sum(Leads), (List Filter: Country = United States, Canada))

**Result**

15 + 5 = 20

The above example as defined in the Formula Editor.
A more sophisticated case is the use of a ranking filter, for example a sales team may want to track the contribution of best-selling products to total revenue. However, what constitutes a popular product may change over time. A measured value can be created for sales which includes a condition that only shows sales for the top products for any month. This simultaneously filters the data but also takes into account changes in what classifies as a top product over time.

<table>
<thead>
<tr>
<th>Month</th>
<th>Product</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>A</td>
<td>10</td>
</tr>
<tr>
<td>Jan</td>
<td>B</td>
<td>5</td>
</tr>
<tr>
<td>Jan</td>
<td>C</td>
<td>15</td>
</tr>
<tr>
<td>Feb</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>Feb</td>
<td>B</td>
<td>25</td>
</tr>
<tr>
<td>Feb</td>
<td>C</td>
<td>5</td>
</tr>
</tbody>
</table>

**Example 2: Measured Value with Top Ranking Filter**  
(Sum(Sales), (Top Ranking Filter: Top 2 Products by Sales))

<table>
<thead>
<tr>
<th>Results</th>
<th>Month</th>
<th>Total Sales Top 2 Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>(A/C)</td>
<td>10 + 15 = 25</td>
</tr>
<tr>
<td>Feb</td>
<td>(B/C)</td>
<td>25 + 5 = 30</td>
</tr>
</tbody>
</table>

The above example as defined in the Formula Editor.
Measured Values are a powerful feature to take into account business logic and quickly perform calculations only when a specific set of criteria is met. **Note:** If your widget is filtered using measured values, then the measured value will override any other widget or dashboard filters you have for the same fields.
Calculating Contributions Using the ALL Function

The All() function returns the total amount for a dimension, and can be used for various use cases. In the following example, we will use the All function to calculate how much each country contributed towards the total cost of a campaign.

Our final widget includes the following information:

<table>
<thead>
<tr>
<th>Country</th>
<th>Total Cost</th>
<th>Total Cost per Countries</th>
<th>Contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Australia</td>
<td>9,643.09</td>
<td>283,755.53</td>
<td>3.40%</td>
</tr>
<tr>
<td>Brazil</td>
<td>9,638.80</td>
<td>283,755.53</td>
<td>3.40%</td>
</tr>
<tr>
<td>China</td>
<td>9,607.4</td>
<td>283,755.53</td>
<td>3.39%</td>
</tr>
<tr>
<td>England</td>
<td>19,492.8</td>
<td>283,755.53</td>
<td>6.87%</td>
</tr>
<tr>
<td>France</td>
<td>19,440.59</td>
<td>283,755.53</td>
<td>6.85%</td>
</tr>
<tr>
<td>Germany</td>
<td>19,377.61</td>
<td>283,755.53</td>
<td>6.83%</td>
</tr>
<tr>
<td>Greece</td>
<td>9,842.84</td>
<td>283,755.53</td>
<td>3.47%</td>
</tr>
<tr>
<td>India</td>
<td>9,603.16</td>
<td>283,755.53</td>
<td>3.38%</td>
</tr>
<tr>
<td>Kazakhstan</td>
<td>9,899.33</td>
<td>283,755.53</td>
<td>3.49%</td>
</tr>
<tr>
<td>Nigeria</td>
<td>9,764.44</td>
<td>283,755.53</td>
<td>3.44%</td>
</tr>
<tr>
<td>Norway</td>
<td>9,681.69</td>
<td>283,755.53</td>
<td>3.41%</td>
</tr>
<tr>
<td>Portugal</td>
<td>9,919.33</td>
<td>283,755.53</td>
<td>3.50%</td>
</tr>
<tr>
<td>South Africa</td>
<td>19,211.15</td>
<td>283,755.53</td>
<td>6.77%</td>
</tr>
<tr>
<td>Ukraine</td>
<td>9,693.03</td>
<td>283,755.53</td>
<td>3.42%</td>
</tr>
<tr>
<td>United States</td>
<td>99,241.60</td>
<td>283,755.53</td>
<td>34.97%</td>
</tr>
<tr>
<td>Vietnam</td>
<td>9,698.67</td>
<td>283,755.53</td>
<td>3.42%</td>
</tr>
<tr>
<td>Grand Total</td>
<td>283,755.53</td>
<td>283,755.53</td>
<td>100%</td>
</tr>
</tbody>
</table>

**Step 1:** The second column above represents a formula that sums up the total cost for all countries and does not represent the breakdown per country. The formula includes the calculation (total cost) followed by the all function (filter), followed by the dimension (country) in parenthesis. It looks like this:

$$([\text{Total Cost}], \text{all}([\text{Country}]))$$
We can save (star) the above formula and call it Total cost for Countries, which will be used in the next step.

**Step 2:** We can now use the above formula in another formula to calculate the contribution, like this:

\[
\frac{\text{SUM}[\text{Cost}]}{\text{Total Cost for Countries}}
\]

The result is the third column above (plus formatting the results as percentages).
Conditional Statements

Conditional statements are formulas that enable you to create additional categories in your data set. When the condition is met, the new category is added to your visualization without modifying the data source itself.

For example, if you want to calculate only certain values according to conditions you define, you can use a conditional statement to include only the values of interest.

Sisense provides several conditional statements that enable you to evaluate criteria and display only the relevant results in your dashboard.

Note: Conditional states work on measures and aggregations only and return numeric values.
IF

IF (Boolean_expression, true value, false value)
Evaluates a boolean expression and returns the first value when evaluated as true, and returns the second value when evaluated as false.
For example:
(if(Sum(Sales) < 10, 0, Sum(Sales) )
Nested IF statements are also supported.
For example:
if (Sum(Sales) < 100 , 1 ,(if (Sum(Sales) < 1000 , 2, 3) )}
isNull

isNull (<numeric value>)

Returns true if the expression doesn't contain data (Null). Can be used as a condition when writing conditional statements.

For example:

If (isNull(sum(deals)), 0, sum(deals) )
Case

```sql
CASE
WHEN Boolean_expression THEN result_expression
[ ... ]
[ ELSE else_result_expression ]
END
```

The CASE function evaluates boolean expressions, and when evaluated as true, returns the corresponding result_expression. If no match is found, the else_result_expression is returned. If there is no default returned and no values match, then Null is returned.

For example:
```sql
CASE
WHEN Sum(Sales) < 100 THEN 1
WHEN Sum(Sales) < 1000 THEN 2
ELSE 3
END
```
**Working with Functions**

Sisense provides a large number of built-in functions that you can use to perform specific calculations or return information about your data. You can use the functions on their own or within formulas.

The functions are organized into four categories: statistical, mathematical, time-related, and other.

**To access functions:**
- Hover over any of the fields in the Values section and click on the Edit (pencil) icon. The Formula Editor is displayed.
- Open the **Functions** tab and scroll down to select the desired function.
- To find a specific function in the **Functions** tab, enter the function’s name (for example, Average) or syntax (AVG) in the Search field. As you begin typing, the functions that contain the string in either their name or syntax will appear below.
Quick Functions

Just like reusing formulas, quick functions is another feature to make working with formulas easier. The Widget Designer provides a variety of predefined commonly used functions that you can easily apply in the Data Browser when selecting a Value to be included in a widget.

Quick Functions instantly add a time dimension to any existing value and formula. These functions include calculations for past values, change over time, contribution and running totals. Quick Functions include all the Time Functions previously discussed but they can only be accessed by clicking on a formula that is already present in a widget.

A simple example of a Quick Function is a finance manager who is reviewing total costs per month but needs to track the accumulated annual costs. They can simply use a quick function to calculate the year to date total for costs.

To use a quick function:
Hover and click on the menu icon of a numeric field in the data panel of the widget designer, and select Quick Functions from the menu:

A list of commonly used functions is displayed. Select a function. The widget will be updated immediately.
Adding Aggregate Functions

You can also add aggregate functions to your formula without opening the formula editor.
Click on the Value icon to open a list of aggregate functions, and select the function to apply to your formula.
Starring Formulas with Quick functions

A more complex example uses starring with multiple quick functions. Let’s say a finance manager also wants to compare year to date costs to the same period for the previous year. They can first apply the year to date function to total cost and then save it to the formula repository. They can then add the saved year to date formula but apply another quick formula for past values which will perform the same calculation but on data from the previous year.

Note: Starred (favorite) formulas will be shared with other users.
Function Reference

This page contains a list of all the functions you can use in Sisense’s formula editor.
Statistical Functions
Average

Avg(<numeric Field>)
Calculates the mean average of the given values.
For example – AVG(Score) will calculate the mean average of the given scores.

Avg(<group by field>, <aggregation>)
Calculates the average of the given aggregation grouped by another field.
For example – Avg( Product, Total Sales) calculates the average of the total sales per product.
Contribution

Contribution(<numeric field>)

Calculates the percentage of total.

For example – Contribution( Total Sales ) will calculate the percentage of total sales per group (for example per day or per product) out of total sales (for all days or all products).
Correlation

CORREL(<Numeric Field a>, <Numeric Field b>)

Returns the correlation coefficient of two numeric fields.
For example – CORREL(Revenue, Cost) will return the correlation between revenue and cost.

CORREL(<group by field>, <aggregation a>, <aggregation b>)

Returns the correlation coefficient of two fields aggregations grouped by another field.
For example – CORREL(Products, AVG(Revenue), AVG(Cost)) will return the correlation between the average of revenue and cost per product.
Count

Count(<Numeric Field>)
Counts the number of unique values within the given values.
Count All

`DupCount(<Numeric Field>)`

Returns the actual item count of the given list of items, including duplicates.
Covariance (Population)

COVARP(<Numeric Field a>, <Numeric Field b>)
Returns the population covariance of <Numeric Field a> and <Numeric Field b>. For example – COVARP(Revenue, Cost) will return the population covariance of revenue and cost.

COVARP(<group by field>, <aggregation a>, <aggregation b>)
Returns the population covariance of two fields aggregations grouped by another field.
For example – COVARP(Products, AVG(Revenue), AVG(Cost)) will return the population covariance of the average revenue and the average cost per product.
Covariance (Sample)

COVAR(<Numeric Field a>, <Numeric Field b>)
Returns the sample covariance of <Numeric Field a> and <Numeric Field b>. For example – COVAR(Revenue, Cost) will return the sample covariance of revenue and cost.

COVAR(<group by field>, <aggregation a>, <aggregation b>)
Returns the sample covariance of two fields aggregations grouped by another field.
For example – COVAR(Products, AVG(Revenue), AVG(Cost)) will return the sample covariance of the average revenue and the average cost per product.
Exponential Distribution

EXPONDIST(<numeric value>, <lambda>, <Cumulative (true/false)>)

Returns the exponential distribution for a given value and a supplied distribution parameter lambda. Cumulative: TRUE = Cumulative distribution function, FALSE = Probability density function. For example – EXPONDIST( Count(Leads), 2, False ) will return the exponential distribution density of the number of leads per country where lambda is 2.
Intercept

\[
\text{INTERCEPT(<field>, <numeric value>)}
\]

Returns the intercept of the linear regression line through a supplied series of x- and y-values.

For example – \text{INTERCEPT(Date.Quarter, Total Sales)} will return the intercept of the regression line that represents the trend over quarter of the sum of sales.
Largest

LARGEST(<Numeric Field>, <k>)

Returns the k-th largest value in a field.
Maximum

Max(<Numeric Field>)

Returns the maximum value among the given values.
Median

MEDIAN( <Numeric Field> )

Calculates the median of the given values. The median of a set of data is the middlemost number in the set. The median is also the number that is halfway into the set.
Minimum

\text{Min(<Numeric Field>)}

Returns the minimum value among the given values.
Mode

MODE(<Numeric Field>)

Returns the most frequently occurring value from the column.
Normal Distribution

NORMDIST(<Numeric Field>, <Mean>, <Standard Deviation>, <Cumulative (true/false)>)

Returns the standard normal distribution for a given value, a supplied distribution mean and standard deviation. Cumulative: TRUE = Cumulative Normal Distribution Function, FALSE = Normal Probability Density Function. For example – NORMDIST(Score, ( Mean(Score), All(Score)), ( STDEV(Score), All(Score) ), False ) will return the normal probability density of a given score.
Percentile

PERCENTILE(<Numeric Field>, <k>)

Returns the k-th percentile value from the given field.

k is any number between 0..1 (inclusive).
Poisson Distribution

POISSONDIST( <numeric value>, <mean>, <Cumulative (true/false)>)

Returns the poisson distribution for a given value and a supplied distribution mean. Cumulative: TRUE = Cumulative distribution function, FALSE = Probability mass function.

For example – POISSONDIST( Score, ( Mean(Score), All(Score) ), ( STDEV(Score), All(Score) ), False ) will return the poisson probability density of a given number of sales.
Quartile

QUARTILE(<Numeric Field>, <k>)

Returns the k-th quartile for the given field.
   k = 0 returns the Minimum value
   k = 1 returns the first quartile (25th percentile)
   k = 2 returns the Median value (50th percentile)
   k = 3 returns the third quartile (75th percentile)
   k = 4 returns the Maximum value
Rank

RANK(<numeric value>, [DESC/ASC], [Rank Type], [<group by field 1>,... , <group by field n>])

Returns the rank of a value in a list of values.

[DESC/ASC] – Optional. By default sort order is descending.

[Rank Type] – Optional. By default the type is standard competition ranking (“1224” ranking). Support also modified competition ranking (“1334” ranking), dense ranking (“1223” ranking) and ordinal ranking (“1234” ranking).

[<Group by field 1>,... , <Group by field n>] – Optional. Rank partitions fields.

For example – RANK(Total Cost, “ASC”, “1224”, Product, Years) will return the rank of the total annual cost per each product, sorted in ascending order.
Skewness (Population)

SKEWP(<numeric value>)

Returns the skewness of the distribution of a given value in the population. For example – SKEWP(Revenue) will return the skewness of the distribution of revenue in the population.
Skewness (Sample)

SKEW(<numeric value>)

Returns the skewness of the distribution of a given value.
For example – SKEW(Revenue) will return the skewness of the distribution of revenue.
Slope

\[ \text{SLOPE(<field>, <numeric value>)} \]

Returns the slope of the linear regression line through a supplied series of x- and y-values.

For example – SLOPE(Date.Quarter, Total Sales) will return the slope of the regression line that represent the trend over quarter of the sum of sales.
Standard Deviation (Population)

STDEVP( <Numeric Value> )

Returns the Standard Deviation of the given values (Population). Standard deviation is the square root of the average squared deviation from the mean. The standard deviation of a population gives researchers the amount of dispersion of data for an entire population of survey respondents.
Standard Deviation (Sample)

STDEV( <Numeric Value> )

Returns the Standard Deviation of the given values (Sample). Standard deviation is the square root of the average squared deviation from the mean. A standard deviation of a sample estimates the amount of dispersion in a given data set, based on a random sample.
T Distribution

TDIST( <numeric value x>, <degrees_freedom>, <Cumulative (true/false)>)

Returns the student’s T-distribution for a given value and a supplied number of degrees of freedom (must be ≥ 1). Cumulative: TRUE = Cumulative Distribution Function, FALSE = Probability Density Function.

For example – TDIST( Score, 3, TRUE ) will return the student’s T-distribution of a given score, with 3 degrees of freedom.
Variance (Population)

VARP( <Numeric Value> )

Returns the Variance of the given values (Population). Variance (Sample) is the average squared deviation from the mean, based on an entire population of survey respondents.
Variance (Sample)

VAR( <Numeric Value> )

Returns the Variance of the given values (Sample). Variance (Sample) is the average squared deviation from the mean, based on a random sample of the population.
Mathematical Functions
Absolute

Abs(<Numeric value>)

Returns the absolute value of the given value.

For example – ABS(Cost), where the absolute result for the value ‘2’ or ‘-2’ is ‘2’.
Acos

ACOS(<numeric value>)

Returns the angle, in radians, whose cosine is the given numeric expression. Also referred to as arccosine.

For example – ACOS(Total Revenue) will return the angle, in radians, whose cosine is the given total revenue.
Asin

ASIN(<numeric value>)

Returns the angle, in radians, whose sine is the given numeric expression. Also referred to as arcsine.

For example – ASIN(Total Revenue) will return the angle, in radians, whose sine is the given total revenue.
Atan

$\text{ATAN}(\text{numeric value})$

Returns the angle in radians whose tangent is the given numeric expression. Also referred to as arctangent.

For example – ATAN(Total Revenue) will return the angle in radians whose tangent is the given total revenue.
Ceiling

$$\text{CEILING}(<\text{numeric value}>\})$$

Returns a number rounded up away from zero, to the nearest multiple of significance.

For example – CEILING(Total Cost), where the result of ‘83.2’ is rounded up to ‘84’.
Cos

\[ \text{COS(<numeric value>)} \]

Returns the trigonometric cosine of the given angle (in radians).
For example – COS(Average Angle) will return the trigonometric cosine of the average angle.
Cosh

COSH(<numeric value>)

Returns the hyperbolic cosine of the given value.

For example – COSH(Total Revenue) will return the hyperbolic cosine of the total revenue.
Cot

\[ \text{COT}(\text{<numeric value}>) \]

Returns the trigonometric cotangent of the given angle (in radians).
For example – COT(Average Angle) will return the trigonometric cotangent of the average angle.
Exp

EXP(<numeric value>)

Returns the exponential value of the given value.

For example – EXP(Sales) will return the exponential value of sales.
Floor

FLOOR(<numeric value>)

Returns number rounded down, toward zero, to the nearest multiple of ‘1’.
For example – FLOOR(Revenue), where the result of ‘88.6’ rounded down is ‘88’.
Ln

LN(<numeric value>)

Returns the base-e logarithm of the given value.

For example – LN(Cost) will return the base-e logarithm of cost.
Log10

\text{LOG10(<numeric value>)}

Returns the base-10 logarithm of the given value.

For example – \text{LOG10(Revenue)} will return the base-10 logarithm of revenue.
Mod

\[
\text{MOD(}<\text{numeric value}, \text{ divisor}>)
\]

Returns the remainder after a number is divided by a divisor.
For example – MOD(Cost, 10), where the reminder of ‘255’ divided by ‘10’ is ‘5’.
Power

Power(value, power)

Returns the results of the given value raised to a supplied power.

For example – POWER(Revenue, 2) will return revenue raised by the power of 2.
Quotient

QUOTIENT(<numeric value>, divisor)

Returns the integer portion of a division.

For example – QUOTIENT(Cost, 2), where the integer portion of ‘5’ divided by ‘2’ is ‘2’.
Round

ROUND(<numeric value>, num_digits)

Returns number rounded to a specified number of digits.

For example – ROUND(Revenue, 2) will return the revenue rounded to two decimal places.
Sin

\[ \text{SIN}(<\text{numeric value}>) \]

Returns the trigonometric sine of the given angle (in radians).

For example – SIN(Average Angle) will return the trigonometric sine of the average angle.
Sinh

\[ \text{SINH(<numeric value>)} \]

Returns the hyperbolic sine of the given value.
For example – \( \text{SINH(Total Revenue)} \) will return the hyperbolic sine of the total revenue.
Square root

\[ \text{SQRT(}<\text{Numeric value}>\text{)} \]

Returns the square root of the given value.

For example – SQRT(Cost) will return the square root of cost.
Sum

Sum(<Numeric Field>)

Calculates the total of the given values.
Tan

\[ \text{TAN}(\text{numeric value}) \]

Returns the trigonometric tangent of the given angle (in radians).
For example – TAN(Average Angle) will return the trigonometric tangent of the average angle.
Tanh

\[ \text{TANH(<numeric value>)} \]

Returns the hyperbolic tangent of the given value.

For example – TANH(Total Revenue) will return the hyperbolic tangent of the total revenue.
Time Related Functions
Day Difference

DDiff( <Start Time>, <End Time> )

Returns the difference between <Start Time> and <End Time> in days.
Past Month Difference

DiffPastMonth( <numeric value> )

Returns the difference between this month's data and the data from the previous month. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

DiffPastMonth( <Total Sales> )

Returns the difference between this month’s sales and previous month’s sales, for the displayed time resolution. For example, for day resolution: (sales in current day - sales in same day one month back).
Past Period Difference

\[
\text{DiffPastPeriod} ( \text{<numeric value>} )
\]

Returns the difference between this period's data and the data from the previous period. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

\[
\text{DiffPastPeriod} ( \text{<Total Sales>} )
\]

Returns the difference between this period's sales and previous period's sales, for the displayed time resolution. Formula: (current value - compared value).
Past Quarter Difference

DiffPastQuarter( <numeric value> )

Returns the difference between this quarter's data and the data from the previous quarter. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:
DiffPastQuarter( <Total Sales> )

Returns the difference between this quarter's sales and previous quarter's sales, for the displayed time resolution. For example, for month resolution: (sales in current month - sales in same month one quarter back).
Past Week Difference

\textbf{DiffPastWeek( <numeric value> )}

Returns the difference between this week’s data and the data from the previous week. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

\textbf{DiffPastWeek( <Total Sales> )}

Returns the difference between this week’s sales and previous week’s sales, for the displayed time resolution. For example, for day resolution: (sales in current day - sales in same day one week back).
Past Year Difference

`DiffPastYear( <numeric value> )`

Returns the difference between this year's data and the data from the previous year. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

`DiffPastYear( <Total Sales> )`

Returns the difference between this year's sales and previous year's sales, for the displayed time resolution. For example, for month resolution: (sales in current month - sales in same month one year back).
Growth

Growth( <Numeric Value> )

Calculates growth over time. The time dimension to be used is determined by the time resolution in the widget/dashboard.

Formula: (current value – compared value) / compared value.

For example:
- If this month your value is 12, and last month it was 10, your Growth for this month is 20% (0.2).
  Calculation: (12 – 10) / 10 = 0.2
- If this year your value is 80, and last year it was 100, your Growth for this year is -20% ( -0.2).
  Calculation: (80 – 100) / 100 = -0.2
Growth Rate

GrowthRate( <Numeric Value> )

Calculates growth rate over time. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:
  If this month your value is 12, and last month it was 10, your Growth Rate for this month is 12/10 = 120% (1.2).
  Calculation: 12 / 10 = 1.2
  If this year your value is 80, and last year it was 100, your Growth for this year is 80/100 = 80% (0.8).
  Calculation: 80 / 100 = 0.8
Growth Past Month

GrowthPastMonth( <Numeric Value> )

Calculates the growth from the past month to the current month. The time
dimension to be used is determined by the time resolution in the
widget/dashboard.

For example:
GrowthPastMonth([Total Sales])

Calculates the difference between this month's sales and previous month's sales,
for the displayed time resolution. For example, for day resolution: (sales in
current day - sales in same day one month back) / sales in same day one month
back.
Growth Past Quarter

GrowthPastQuarter( <Numeric Value> )

Calculates the growth from the past quarter to the current quarter. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

GrowthPastQuarter([Total Sales])

Calculates the difference between this quarter's sales and previous quarter's sales, for the displayed time resolution. For example, for month resolution: (sales in current month - sales in same month one quarter back) / sales in same month one quarter back.
Growth Past Week

\texttt{GrowthPastWeek(<Numeric Value>)}

Calculates the growth from the past week to the current week. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

\texttt{GrowthPastWeek([Total Sales])}

Calculates the difference between this week's sales and previous week's sales, for the displayed time resolution. For example, for day resolution: \((sales \text{ in current day} - sales \text{ in same day one week back}) / sales \text{ in same day one week back} \).
Growth Past Year

GrowthPastYear( <Numeric Value> )
Calculates the growth from the past year to the current year. The time dimension to be used is determined by the time resolution in the widget/dashboard.
For example:
GrowthPastYear([Total Sales])
Calculates the difference between this year’s sales and previous year’s sales, for the displayed time resolution. For example, for month resolution: (sales in current month - sales in same month one year back) / sales in same month one year back.
Hour Difference

HDiff( <Start Time>, <End Time> )

Returns the difference between <Start Time> and <End Time> in hours.
Prev

Prev( <Time Field>, [<N>] )

Returns the Time period Member in <Time Field> which is N periods back from the current Member. This function only works as a scope function and not by itself.

For example – This formula will return the numeric value 2 months ago:

(<Numeric Value>, Prev(<Month Field>, 2))
Minute Difference

MnDiff( <Start Time>, <End Time> )

Returns the difference between <Start Time> and <End Time> in minutes.
Month Difference

MDiff( <Start Time>, <End Time> )

Returns the difference between <Start Time> and <End Time> in months. Returns whole numbers.
Month to Date Average

MTDAvg( <Numeric Value> )

Returns the running average starting from the beginning of the month up to the current time period member.
The time dimension to be used is determined by the time resolution in the widget/dashboard.
Returns 0 if the active time resolution is quarters or years or weeks.
Month to Date Sum

MTDSum( <Numeric Value> )

Returns the running total starting from the beginning of the month up to the current time-period member.
The time dimension to be used is determined by the time resolution in the widget/dashboard.
Returns 0 if the active time resolution is quarters or years or weeks.
Next

Next( <Time Field>, [N] )

Returns the Time period Member in <Time Field> which is N periods after the current Member. This function only works as a scope function and not by itself. For example – This formula will return the numeric value 2 months ahead of now:

(<Numeric Value>, Next(<Month Field>, 2))
Now

Now(<Day from Date field>)

Returns the value for the current time period. Supports day, month, quarter or year. The Now function receives a date dimension and its level and returns all the members in that dimension which match the current query execution time.

**Note:** This function only works as a scope function and not by itself.
The following example will return the value for the current day.

([Total Sales], Now([Days in Datefield]))
Past Year

PastYear( <Numeric Value> )

Calculates the value for the same period in the past (previous) year.

For example:
  - If you’re looking at a specific day, you will see the value of the same day one year back.
  - If you’re looking at a specific month, you will see the value of the same month one year back.

Note: When using the Past Year function in a weeks table and using a week filter, no results are returned.
Past Quarter

`PastQuarter(<Numeric Value>)`

Calculates the value for the same period in the past (previous) quarter.

For example:
- If you’re looking at a specific day, you will see the value of the same day one quarter back.
- If you’re looking at a specific month, you will see the value of the same month one quarter back.
Past Month

`PastMonth( <Numeric Value> )`

Calculates the value for the same period in the past (previous) month.

For example:
   If you’re looking at a specific day, you will see the value of the same day one month back.
Past Week

PastWeek( <Numeric Value> )

Calculates the value for the same period in the past (previous) week.

For example:
   If you’re looking at a specific day, you will see the value of the same day one week back.
Past Day

PastDay( <Numeric Value> )

Calculates the value for the same period in the past (previous) day.

For example:
   If you're looking at a specific day, you will see the value of the same day one day back.
Quarter Difference

QDiff( <Start Time>, <End Time> )

Returns the difference between <Start Time> and <End Time> in quarters. Returns whole numbers.
Quarter to Date Average

QTDAvg( <Numeric Value> )

Returns the running average starting from the beginning of the quarter up to the current time-period member.

The time dimension to be used is determined by the time resolution in the widget/dashboard.

Returns 0 if the active time resolution is years or weeks.
Quarter to Date Sum

QTDSum( <Numeric Value> )

Returns the running total starting from the beginning of the quarter up to the current time-period member.
The time dimension to be used is determined by the time resolution in the widget/dashboard.
Returns 0 if the active time resolution is years or weeks.
Range

`range( <Field1>, <Field2> )`

Returns a graphical range selector for a data set where two members of the same dimension and level define the minimum and maximum values of the range.

**Note:** The second argument `<field2>`, must be wrapped in `current()`.
Second Difference

SDiff( <Start Time>, <End Time> )

Returns the difference between <Start Time> and <End Time> in seconds.
Week to Date Average

WTDAvg( <Numeric Value> )

Returns the running average starting from the beginning of the week up to the current time period member.
The time dimension to be used is determined by the time resolution in the widget/dashboard.
Returns 0 if the active time resolution is years, quarters, or months.
Week to Date Sum

WTDSum( <Numeric Value> )

Returns the running total starting from the beginning of the week up to the current time period member.
The time dimension to be used is determined by the time resolution in the widget/dashboard.
Returns 0 if the active time resolution is years, quarters, or months.
Year Difference

YDiff( <Start Time>, <End Time> )

Returns the difference between <Start Time> and <End Time> in years. Returns whole numbers.
Year to Date Average

YTD Avg( <Numeric Value> )

Returns the running average starting from the beginning of the year up to the current time period member.

The time dimension to be used is determined by the time resolution in the widget/dashboard.
Year to Date Sum

YTDSum( <Numeric Value> )

Returns the running total starting from the beginning of the year up to the current time period member.
The time dimension to be used is determined by the time resolution in the widget/dashboard.
Other
All

All(<Field>)

Ignores the scope set on the dimension.
CASE

(WHEN <condition> THEN <result_expression> [...] [ELSE <result_expression>] END)

Returns the result_expression of the first condition evaluated as true. When no condition is true, else_expression is returned, if one is defined.

For example, the below function will return '1' when the Total Sales value is between 100 and 1000. It will return '2' if the Total Sales value is above 1000. It will return '3' in any other case (meaning, when Total Sales are below 100).

CASE

WHEN Sum(Sales) < 100 THEN 1
WHEN Sum(Sales) < 1000 THEN 2
ELSE 3
END
IF

IF (<condition>, <numeric expression 1>, <numeric expression 2>)

Returns numeric expression '1' when the condition is true, and expression '2' when the condition is false. Nested conditional statements are supported.

For example, if the number of unique values within the Sales values is larger than 100, the below function will return the Total Sales x 1.1 (sales increase of 10%). Otherwise - if the number of unique values within the Sales values is lower than 100, the function will return only the Total Sales, without an increase.

IF(Count(Sales)>100, Sum(Sales)*1.1, sum(Sales))
IsNull

(<numeric value>)

Returns true if the expression doesn't contain data (Null). Can be used as a condition when writing conditional statements.
Ordering

`ORDERING(<expression1>,<expression2>)`

Returns the numeric order position of rows sorted into ascending or descending order, breaking ties with further arguments.

The expressions must be aggregated by applying the MIN/MAX functions as in the example below:

`ORDERING(MIN([Sales Person Name]), MIN([Days in Transaction_Date]), -1*Sum([Sales]))`
Rdouble

RDOUBLE(<R expression>, <numeric value 1>, [<numeric value 2>, ..., <numeric value n>] )

Returns a numeric result for a given R expression and a list of numeric values
(use ‘args[[i]]’ in the R expression to reference numeric values parameters).
The R expression is passed to the running Rserve.
The optional ordering expression determines the order in which the rows are
sent to R. The argument of the Ordering parameter can be an index in your data
source or you can use the ORDERING() function to determine the order of the
field. For more information about the ORDERING() function, click here.
For example – RDOUBLE(“m <- log(matrix(unlist(args), ncol=2));
kmeans(m,3)$cluster”, [Total Cost], [Total Revenue]) will return the k-means
cluster (R expression) of the args: [Total Cost] and [Total Revenue].
RDOUBLE(<recycle>, <R expression>, <numeric value 1>,
[<numeric value 2>, ..., <numeric value n>] )
recycle = TRUE (default) – Results will be cached for unchanged functions and
data.
recycle = FALSE – Results will not be cached. Use this option if your R code
contains randomness.
Rint

\[
\text{RINT}(\text{< expression>}, \text{< numeric value 1>}, \ldots, \text{< numeric value n>])}
\]

Returns an integer result for a given R expression and a list of numeric values (use ‘args[[i]]’ in the R expression to reference numeric values parameters). The R expression is passed to the running Rserve.

The optional ordering expression determines the order in which the rows are sent to R. The argument of the Ordering parameter can be an index in your data source or you can use the ORDERING() function to determine the order of the field. For more information about the ORDERING() function, click here.

For example – RINT("m <- log(matrix(unlist(args), ncol=2)); kmeans(m,3)$cluster", Total Cost, Total Revenue)) will return the k-means cluster (R expression) of the args: [Total Cost] and [Total Revenue].

\[
\text{RINT(\text{< recuryle>}, \text{< expression>}, \text{< numeric value 1>}, \ldots, \text{< numeric value n>])}
\]

\begin{itemize}
\item \texttt{recycle = TRUE} (default) – Results will be cached for unchanged functions and data.
\item \texttt{recycle = FALSE} – Results will not be cached. Use this option if your R code contains randomness.
\end{itemize}
Running Sum (RSUM)

RSUM ( <numeric value> ), RSUM ( <numeric value> , <continuous> )

Returns the running total of the measure by the defined dimension according to the current sorting order in the widget.

By default, RSUM accumulates a measure by the sorting order of the dimension. To accumulate by another order, the relevant measure should be added as an additional column and sorted.

<continuous> is a boolean value that accumulates the sum continuously when there are two or more dimensions. The default value is False.

Note: Filtering the RSUM column by Values will filter the dimensions and recalculate the RSUM from the first filtered value.
Building Formulas with Functions

Functions can be used to build formulas.
Combine Data: Aggregate Functions

Aggregations are used to perform mathematical calculations on data. This being such an essential function, Sisense offers an advantage in the ability to run multiple aggregations on several fields simultaneously – this makes it easy to summarize data based on multiple factors.

| Aggregate Syntax: |
| Function(Numeric Field) |
| Parameters |
| Function: Sum(), Avg(), Count(), DupCount(), Max(), Min() |
| Numeric Field: A numeric field or formula. |

An example of a **simple aggregation** is a sales manager who wants to calculate the average sales revenue for each sales rep. The manager can create a pivot table, which shows the sales rep and their average sales revenue.

A more complex example is a **multi-pass aggregation** (or grouping) that is an aggregation that performs multiple calculations simultaneously. Following our first example, let's assume the sales manager wants to also see average sales per day for each sales rep. Instead of having to add an additional column for day in the pivot table, the manager can create a multi-pass aggregation that first performs a sum of sales per day and then averages the results for each rep. This requires two fields: a day from a date field and the revenue field, as well as two aggregations: sum of sales and average. The result is that sales manager does not need to add a column for days in the pivot.
The above example as shown in the formula editor.
Summarize Data: Statistical Functions

Descriptive statistics provide meaningful summaries of data and help make more informed decisions. This is particularly important for large data sets where descriptive statistics can help to focus analysis.

An example of statistical functions is a marketing team that has a large data set on leads generated from various channels and want to understand where to focus their budget. Descriptive statistics can be used to summarize valuable insight about each channel, such as the central tendency or median leads generated, along with standard deviations to assess typical lead volume.
Accumulate Data: Running Total and Average

Often, to measure performance, data must be viewed in a continuous and accumulative format over extended periods such as years, quarters or months. Sisense provides functions to create running totals and averages over standard or custom time periods.

For example, a support team has a goal to reduce the average monthly cost to resolve open issues. A Year to Date Average can be used to track progress towards reducing the average cost of support.
Change over Time: Time Functions

Time is critical for business decisions. Time functions make it easy to compare outcomes at different points in time, determine growth rates and calculate the time differences. Functions can be set for common time periods such as year, month or day, as well as for custom periods.

*Time Syntax: Function(Numeric Field) *

**Parameters**

**Function:**
- PastYear()
- PastQuarter()
- PastMonth()
- Growth()
- GrowthRate()
- Contribution()
- YDiff()
- QDiff()
- MDiff()
- DDiff()
- HDiff()
- MnDiff()
- SDiff()
- Next()
- Prev()
- DiffPastYear()
- DiffPastMonth()
- DiffPastQuarter()
- DiffPastDay()

**Numeric Field:** A numeric field or formula.

* To calculate time functions require a time filter applied on the dashboard or widget.

For example, an executive team wants to compare growth in revenue to the same period in the previous year. A Difference in Past Year function can be used to compare past values based on the current month to the same month in the previous year.

<table>
<thead>
<tr>
<th>Date</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-2013</td>
<td>5</td>
</tr>
<tr>
<td>02-2013</td>
<td>10</td>
</tr>
<tr>
<td>03-2013</td>
<td>15</td>
</tr>
<tr>
<td>01-2014</td>
<td>25</td>
</tr>
<tr>
<td>02-2014</td>
<td>5</td>
</tr>
<tr>
<td>03-2014</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Month</th>
<th>Difference to Past Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-2013</td>
<td>25-5 = 20</td>
</tr>
<tr>
<td>02-2013</td>
<td>5-10 = -5</td>
</tr>
<tr>
<td>03-2014</td>
<td>10-15 = -5</td>
</tr>
</tbody>
</table>
Fiscal Years

By default, Sisense calculates the beginning of the fiscal year as January 1st. In some cases, you might need to express your company’s date fields according to another fiscal month, such as April 1st. Sisense allows you to easily shift your fiscal calendar to start at a different month according to your company’s requirements.

This can be defined at the system level, by the Administrator, and also at the ElastiCube level, by the Data Designer. This means that you can support one fiscal calendar month for your system, such as January 1st, while a specific ElastiCube can be set according to the US tax year which begins October 1st.

When you change the fiscal calendar, this affects how time is calculated in your dashboard and you are also getting an indication, for example when filtering your widgets and dashboards by time, through the FY tag.

<table>
<thead>
<tr>
<th>Date Level</th>
<th>Effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>Changes the calendar year to fiscal. By default, the name</td>
</tr>
<tr>
<td>Date Level</td>
<td>Effects</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>reflects the year in which it ends. For example, if the fiscal year</td>
</tr>
<tr>
<td></td>
<td>starts in April, the year for the date June 1, 2014 would be shown as</td>
</tr>
<tr>
<td></td>
<td>FY 2015.</td>
</tr>
<tr>
<td>Quarter</td>
<td>Changes the calendar quarter to that of the quarter in the fiscal year.</td>
</tr>
<tr>
<td></td>
<td>For example, if the fiscal year starts in April, the quarter for the</td>
</tr>
<tr>
<td></td>
<td>date June 1, 2014 would be Q1.</td>
</tr>
<tr>
<td>Month</td>
<td>Not affected by fiscal year settings. The calendar month is the same as</td>
</tr>
<tr>
<td></td>
<td>the fiscal month.</td>
</tr>
<tr>
<td>Week</td>
<td>Changes the calendar week to the week of the fiscal year. For example,</td>
</tr>
<tr>
<td></td>
<td>if your fiscal year begins April 1st, April 3rd would fall under Week</td>
</tr>
<tr>
<td></td>
<td>1.</td>
</tr>
<tr>
<td>Day/Minute/Hour</td>
<td>Not affected by fiscal year settings.</td>
</tr>
</tbody>
</table>

To set your fiscal year on the system level, see [System Settings](#).

Administrators and Data Designers can set the fiscal year on the ElastiCube level.

**To set your fiscal year on the ElastiCube level:**

In the Sisense Web Application, select **Admin > Data Sources**.
For the relevant ElastiCube, select the ElastiCube menu > Fiscal Year Start, and then select the month.
Formatting Fiscal Dates

When you apply fiscal years, the default format for your dates is 2018 FY for years or Q1 2018 FY for quarters.

You can customize the format of the date through placeholders such as “y” and “p” where “y” represents the year and “p” the previous year.

The examples below illustrate how you can format your dates:
- 2017/18: yyyy/y
- 17/18: yp/yy
- FY17/18: FY yp/yy
- Q1 2018 FY: Q yyyy FY

By default, Sisense applies FY at the end of the year when fiscal years are applied. However, if you make any changes, Sisense will no longer add the FY automatically, so you must define this with the placeholder FY in the location you want as shown above.

**To format your fiscal dates:**
- On the Dashboard, click the Pencil (Edit) button that appears in the top-right corner of a widget with a date dimension.
Select the Calendar icon to define your formatting.

Select the relevant time period and enter the relevant placeholders in the format you want to be displayed in your widget. A preview of the custom format is displayed below.

Click **OK**. The format of your dates is updated in the widget.
Formatting Fiscal Dates through the REST API

In addition to formatting Fiscal dates through Sisense, you can also define the format through the Sisense REST API.

The settings/system endpoint contains the Fiscal object that has two keys, `month` and `format`.

```json
{
    "id": "string",
    "alias": "string",
    "firstday": "string",
    "fiscal": {
        "month": "jan",
        "format": "string"
    },
    "dashboardAdministration": {
        "enabled": true,
        "email": {
            "enabled": true,
            "senderEmail": "string",
            "senderName": "string"
        },
        "webServer": {
            "enableSSl": true,
            "cors": {
                "enabled": true,
                "allowedOrigins": ["string"]
            }
        }
    }
}
```

The `month` key determines the first month of your fiscal year. The `format` key determines how your fiscal dates are displayed.

For example:

- `years`: `fiscalFormatRestAPI + 'yyyy';`
- `quarters`: `fiscalFormatRestAPI + 'yyyy Q';`
- `months`: `'MM/yyyy' fiscalFormatRestAPI;`
- `weeks`: `'ww ' + fiscalFormatRestAPI + ' yyyy';`
days: 'MM/dd/yyyy ' + fiscalFormatRestAPI;

If you were to set the format to the following, 'EEE d/MMM//QQ//yp-yy', the format would be displayed in Sisense as follows:

FY Mon 1/Apr/Quarter 2/96-97 1997 Q2

In addition, you can add strings to the format as long as they are in single quotes.

To modify your fiscal date formats through the REST API, you can send a POST request to the settings/system API.
Date and Time Fields

Many widgets include data that is relative to date and time fields. Sisense enables you to apply date and time filters so you can easily identify trends in your data. When you create a widget in your dashboard that includes a date field, Sisense automatically breaks the data down by year, however, you can change this to a shorter resolution. For example, a Date field included in a widget can be separated into Years, Quarters, Months, Weeks, Days, and Time from your widget’s filter.
Dashboard Viewers have the option to drill down the widget according to the Date level as shown in the following image:

Viewers can drill down into the lowest date or time resolution available in your data.

The highest resolution available is by Year and the lowest resolution is Time, which includes Hour and Minute intervals.

For more information about filters, see Creating Dashboards Filters.
Customizing the Dashboard Layout

By default, widgets are arranged in a dashboard one underneath the other. When you add a new widget, it is added at the bottom of the dashboard. You can drag widgets one on top of another by selecting the widget's title and moving the widget to the relevant location.

The dashboard can also be organized in columns. By default, a dashboard is created with a single column, but you can add up to four columns. Each column can contain multiple widgets, but a widget cannot span more than one column. You can rearrange the order of the widgets, reorganize the widgets in columns, add/remove widgets from columns, and resize columns/widgets.

For a short video overview of customizing the dashboard layout, see below:
Layout Mode and View Mode

A dashboard can be viewed in either Layout mode or in View mode.

**Layout Mode**
Layout mode is the default mode in which you can add new widgets and rearrange them on the dashboard, as described throughout this section.

☐ is displayed in the top-right corner of the screen when the dashboard is in Layout mode. Click this button to display View mode.

**View Mode**
View mode enables you to see what the dashboard looks like when a user is only viewing the dashboard. In View mode, the dashboard cannot be rearranged or edited.

шение is displayed when the dashboard is in View mode. Click this button to display Layout mode.
Adding Dashboard Columns

To change the number of columns:
Click on the dashboard's menu, and select Columns and the number of columns.
Moving a Widget on the Dashboard

To move a widget to a different column or position:

Make sure that you are in Layout mode (default mode). The Layout mode indicator is displayed in the top-right corner of the dashboard. If not, click to go from View Mode to Layout mode.

Drag and drop the widget into the desired position by dragging it by the top part of the widget.

The position of a widget can be split to contain multiple widgets within a single column by dragging one widget on top of another.
Placing Widgets Side-by-Side

You can place widgets side-by-side using one of the following methods:
   By putting each widget in a different column.
   OR
   By dragging one widget on top of the other. Both these widgets will then be in the same column.
Resizing Widgets

Resizing the column height or width automatically resizes the widgets in the best possible way.

To resize a widget:

- Resize the dashboard column in which the widget is located by dragging the edge of the column right or left. All the widgets in this same column are resized accordingly in an optimal manner.
- Resize the widget by dragging its edges right, left, up or down.

Refer to Ticker widgets below for a description of how an Indicator Widget turns into a ticker type widget when you reduce its height.
Adding Widget Titles

You can add titles to your widgets directly in the dashboard view.

If you already added a title when you created the widget in the Widget Designer, you can edit the title directly in the dashboard.

**To add a widget title from the dashboard:**

To add a title, click **Add Title** at the top of the widget, and type in the title. If you are editing a title, click on the existing title, and type in the new title.

Click on the green ✔️ icon to save the title.
Working with Ticker Widgets

If you resize the height of an Indicator widget, it automatically turns into a Ticker-type widget. An Indicator widget that was a numeric Indicator appears differently to an Indicator widget that was a gauge.

**Regular Indicators**

<table>
<thead>
<tr>
<th>GROWTH OF VISITORS</th>
<th>REVENUE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>13.2%</strong></td>
<td><strong>2M</strong></td>
</tr>
<tr>
<td># of Visitors</td>
<td>Growth</td>
</tr>
<tr>
<td>220K</td>
<td></td>
</tr>
</tbody>
</table>

Full size Indicator widget | Full size Gauge widget

**Ticker Indicators**

<table>
<thead>
<tr>
<th>GROWTH OF VISITORS: <strong>13.2%</strong></th>
<th># of Visitors: 220K</th>
<th>REVENUE: <strong>2M</strong></th>
<th>Growth: 0.2%</th>
</tr>
</thead>
</table>

Indicator widget after resizing down | Gauge widget after resizing down
Changing the Dashboard’s Color Palette

You can easily change the color scheme of the dashboard to suit your professional or brand preferences.

To change the color palette:
   Click 🔄 in the top menu, and select the preferred color palette. The default palette is Vivid.

Note: Three of the palettes are suitable for color blindness: ColorBlind 1, 2, and 3.

For a short video overview of the process, see below:

Note: When you change the color of an individual widget to a new color that is not one of the palette colors, the new color will be applied to the widget regardless of the applied palette. If you select a new color from one of the palette colors, then when changing the dashboard’s palette, the color will change according to the new palette.
Examples
Creating a Custom Palette via the REST API

You can create your own brand custom palette through the REST API. Click here to learn more about using the REST API. To learn more about creating a custom palette using the REST API, see this support article.

Sisense provides the following endpoints:

- **GET /palettes**
  - Returns a list of available palettes in the Sisense web app.
- **GET /palettes/default**
  - Returns the default color palette.
- **POST /palettes**
  - Adds a new color palette.
- **PUT /palettes/{name}**
  - Updates the dashboard's color palette.
- **DELETE /palettes/{name}**
  - Deletes a color palette from the color palettes.

The table below describes the parameters you can return and update through the palettes endpoints:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>colors</td>
<td>array</td>
<td>The colors included in the palette. Your custom palette can include up to 16 colors.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the new palette.</td>
</tr>
<tr>
<td>isDefault</td>
<td>boolean</td>
<td>Determines whether the new palette is the default. New dashboards are created with the default palette. There can only be one default palette at any time.</td>
</tr>
<tr>
<td>sortOrder</td>
<td>number</td>
<td>Determines the order of the palette in the palettes list.</td>
</tr>
</tbody>
</table>

To add an existing palette:
Access the Sisense REST API.

In the API Documentation, select version 0.9 on the top-right of the screen. Click **palettes** to access the API operations related to roles.

Go to PUT/palettes/default and click **Try It Out**.

Indicate the current name of the palette, and make your changes. You can change the palette name, colors, etc.

Click **Execute** and scroll down to the Responses section to verify that the response does not contain errors.

**To create a new palette:**

Go to **POST / palettes** and click **Try It Out**.

Enter all the relevant values.

Click **Execute**.

The new palette will be added to your Dashboard Palettes.

**To delete a palette:**

Go to **DELETE/ palettes**, click **Try It Out**, and enter the name of the palette to delete into the Value input box. Then click **Execute**.

The Palette will be deleted from the list of your dashboard Palettes.
Creating a Dashboard with Multiple Sources

Dashboards can support multiple data sources including ElastiCube and Live data models, which means you can have widgets built on several data sources within a single dashboard. For example, if you have widgets from an ElastiCube on a dashboard, you can add a data set to the dashboard and create widgets built on that data set's live data source. The widgets from both data sources continue to function independently of each other in the same dashboard. In addition, filters applied to fields from each data source, do not affect the fields from another data source.

To create a widget based on another data source, when you creating the widget, in the top-right corner of the New Widget window, modify the default data source collected.
Changing a Dashboard’s Data Source

Changing a dashboard’s data source is useful when you have recently changed your server and you need to reassign your dashboard’s data source or in cases where you have deleted a data source and you want to reassign the dashboards. Dashboards can also support multiple data sources including ElastiCube and Live data models, for more information, see Creating Dashboards with Multiple Sources.

To change your dashboard’s data source:

In the top-left corner of your dashboard, click the ElastiCube link.

Hover over Change Data Source and select the new data source.

The source is added to your list. You now add new widgets from any of your data sources to the dashboard.
Exploring Dashboards

In the **Analytics** page, you can see dashboards that you created or were shared with you on the left or the main dashboard area below.

To open a dashboard, you can click the title of the dashboard on the left side or click the dashboard tile below. This opens up the dashboard inside the **Analytics** page.
A dashboard that was shared with you appears with this icon in the Dashboards list.

To open a dashboard, you can click the title of the dashboard on the left side or click the dashboard tile below.

Alternatively, if you have a lot of dashboards, you can search for dashboards by typing the title in the Search field. As you begin typing, any relevant results are displayed.
To locate a dashboard by name, source, or owner, you can use search operators within the **Search** field. Sisense supports the following operators:

<table>
<thead>
<tr>
<th>What You Can Search By</th>
<th>Search Operator and Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dashboard name</td>
<td>Name:</td>
</tr>
<tr>
<td></td>
<td><strong>Example</strong>: Name:Sales</td>
</tr>
<tr>
<td>Data source</td>
<td>Source:</td>
</tr>
<tr>
<td></td>
<td><strong>Example</strong>: Source:Sample eCommerce</td>
</tr>
<tr>
<td>Dashboard Owner</td>
<td>Owner:</td>
</tr>
<tr>
<td></td>
<td><strong>Example</strong>: Owner:John</td>
</tr>
</tbody>
</table>
Interacting with Filters as a Viewer

Once you’ve found the dashboard you’re looking for, you can start exploring your data.
Sisense dashboards usually have one or more filters that affect the dashboard widgets.
This topic describes how Viewers can interact with filters for analyzing data.
You can interact with filters for analyzing data, either through the Filters pane or by simply selecting values by left-clicking the widget visualization. Each time you interact with a filter, for example by selecting or entering a value, the filter is immediately applied to your dashboard. You will not affect anyone else by changing the filters. In addition, you can restore the original state of the dashboard at any given point by selecting the dashboard menu item and clicking **Restore Dashboard**.

As a Viewer, you can interact with filters as follows:
Make a different selection in the filter controls displayed in the Filters pane on the right side of the dashboard, as shown below:
OR

Click on the pencil icon next to the filter name (shown below) in the dashboard to display the Filter Definition window.

In addition, you can perform the following procedures:
Switching Filters On and Off
Saving your Default Filters View
To create a filter, you must be a Designer. Click here to learn more about filters.
Changing Sisense’s Language

This topic discusses how Sisense Administrators can change the language of the Sisense Web Application for all of their users.

Administrators can set the default language for a Sisense server from the Admin section. If you are an Administrator and want to set the language for your system, users groups, or users, see Changing the Sisense Web Application’s Language.

If your users would like to set another language for the Sisense Web Application other than the language defined by the Administrator, they can select a language from the User Profiles or from the Login page.

Sisense supports the following languages:
- English
- Chinese
- Dutch
- French
- German
- Italian
- Japanese
- Portuguese
- Russian
- Spanish (LA)
- Spanish (Spain)

In addition, any languages that your company have translated will appear in the list according to the name defined in the translated system files.

Note, the following content is not being translated:
- Built-in Javascript editor
- ElastiCube Manager and Server Console
- Sisense documentation and online help
- REST API and documentation
- Sisense automated emails

To change your system's language:
Select the language from the Login page of the Sisense Web Application.

OR

In the Sisense User Options, open the languages list and select the language you want to display.

After setting the user’s language, Sisense will automatically be displayed in the selected language in future sessions.
Making Selections in a Widget

You can click on a specific portion of a widget to select it. This filters the dashboard according to the selected data by adding a filter to the dashboard’s Filters panel.

You can also use the selection to drill down in the widget (see Drilling Down in a Widget).

**To select an item in a widget for filtering:**

Left-click an item in a widget to automatically select it and add a dashboard filter according to the selection.

For multiple selection, use the Ctrl key. When released, a menu will appear. Click Select.

In chart widgets, you can drag and draw the area to be selected, as shown below:
When you select a portion of the widget, \( \uparrow \) appears in the widget's menu bar. Click \( \uparrow \) to expand your view of the selection.

You can refine your data further by toggling the switches on the left side as shown in the example below.
Drilling Down in a Widget

This topic describes how you can drill down into your data in most widgets to get an in-depth view of a selected value.

In the following example, the pie chart on the left shows a breakdown by gender. This is the original chart as created by the dashboard’s Designer. On the right side, the chart shows a breakdown or drill down showing age groups of the ‘female’ segment from the original pie chart.

With Sisense you can drill down from any field to any field, unless disabled by the dashboard’s Designer.

For information about enabling or disabling the drilling feature, see Drill Hierarchies.

To drill down into a chart:

Right-click on the item in the widget into which you want to drill down.

To manually select the drill hierarchy path, select **Drill Into**, and then in the Data Browser, select the field into which to drill down. If you have already drilled into this chart, then you will have shortcuts to previously
selected fields, or select **Choose Another Field** to select a different field for the first time.

**OR**

Select a predefined drill hierarchy (if available). Predefined drill hierarchies are defined by the dashboard's owner.

<table>
<thead>
<tr>
<th>Drill</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calendar Hierarchy</td>
</tr>
<tr>
<td>OrderDate</td>
</tr>
<tr>
<td>Recent</td>
</tr>
</tbody>
</table>

To drill up to a higher level, click on a breadcrumb. To drill all the way up, click on the X icon.
Leverage AI to Uncover Hidden Insights

Sisense is looking for customers who are willing to provide feedback about this feature as part of our early-adopter program. If you would like to influence the development of this feature, contact your CSM.

Sisense’s AI (Artificial Intelligence) Exploration Paths automatically generate visualizations and insights that anticipate your Viewers’ questions without requiring a Dashboard Designer to create more widgets. Sisense leverages AI to uncover hidden insights and suggest new visualizations, all within the context of the data model that your dashboard is based on. Exploration widgets offer a great way to answer ad-hoc questions that naturally arise when analyzing a KPI. They enable you to drill down into widgets in cases when it is challenging to set up drill hierarchies to answer all the potential business questions. Now you can build a lean and highly focused dashboard including the KPIs most important to the business, and Sisense AI will generate exploration widgets for your users to go deeper.

The suggestions are taken from across all dashboards based on the data model that your dashboard is connected to. To provide Viewers with a right context throughout the exploration, the exploration widgets inherit the dashboard and widget filters, and go through the same security rules as any other widget in the dashboard.

Note: To enable Viewers to receive exploration widgets in the dashboard, your Sisense Administrator must first activate the Exploration Path feature for the
entire system and then enable it on each particular dashboard where this feature needs to be available.

For a short video overview of exploration widgets, see the video below:
What Determines which AI-driven Suggestions Viewers Get?

Sisense automatically suggests visualizations that anticipate the next questions a Viewer might have, without involving a Dashboard Designer.
Sisense employs an AI algorithm that runs in the background and scans the entire dashboard base associated with a particular data model. The algorithm studies the analysis and the usage patterns (such as drilling down and filtering) performed by all dashboard users, and based on that suggests new exploration widgets. With time, as the AI algorithm picks up more input from user activity, the suggestions become more accurately targeted to the Viewer needs.
How Does This Work?

The purpose of the process is to produce new break-by and trend-over-time analysis of the data. To do that, the AI exploration algorithm suggests additional fields (for break-by analysis) and date fields (for trend-over-time analysis) by which to slice and dice the metric.

At the first stage, the AI algorithm compares formulas across the entire dashboard set. It looks at all the unique combinations of components of each formula, such as function, columns(s) and filter(s). Then it utilizes the fuzzy version of the Jaccard similarity coefficient to calculate the similarity between the formula you want to explore to all other formulas in the system. The output of this is a similarity score.

Next, the algorithm collects the fields and data fields that are used together with formulas in other widgets in the system. Finally, it employs the Pointwise Mutual Information (PMI) measure, which is a measure of association used in Information Theory and Statistics. PMI ranks the fields and date fields according to their frequency of use with similar formulas in other widgets.

The result is a ranked list of the fields and date fields that frequently appear with formulas most similar to the formula we want to explore. The fields and date fields with the highest ranking are suggested to the user as exploration widgets to provide additional insights about the data.
Viewing Exploration Widgets

To view exploration widgets:

Click 🌟 in your dashboard widget.

A window is displayed with suggested exploration widgets.

Switch between the two tabs to view all the exploration widgets.

To download an exploration widget to a PNG file, click the ⬇️ button at the bottom of the widget.

(Optional) To provide positive feedback to the algorithm about the exploration widgets, click 🌟. This provides more input to the algorithm, enabling it to optimize its results over time. Note that clicking 🌟 would not influence the results already upon your next session with Sisense, as it takes time to refine the results.
When Sisense does not Suggest Exploration Widgets

In some cases, Sisense is unable to suggest exploration widgets. Below are some possible reasons:

**No previous user activity**
The AI algorithm makes suggestions based on the analyses and usage patterns previously performed by users in the dashboard. When a new dashboard is created, it has no previous user activity, so the algorithm cannot make any suggestions. In this case, 👇 is not displayed for widgets on the new dashboard.

**What you can do:** Work in the dashboard as usual. When the algorithm has picked up user-activity patterns (this might take a day or two), it will start suggesting exploration widgets.

**No data to display**
This might be due to the data security rules or filters applied to the dashboard, or when there are less than two members to display. In these cases, when you click 👇, you receive the following message:

What you can do: Work in the dashboard as usual. When the algorithm
detects some new patterns in user activity, or when data security rules or filters change, it will start suggesting exploration widgets.

When the algorithm detects any changes in the system, the exploration widgets will reflect these changes on the next day, because the algorithm processes the new information overnight.
Enabling AI-driven Exploration Widgets

To enable Viewers to view AI-driven exploration widgets in their dashboards, the Sisense Administrator must activate the Exploration Path feature system-wide. Then dashboard Owners can enable it for each dashboard separately. It is recommended to test-drive this feature for your Viewers by enabling it on a single dashboard before enabling it on all dashboards. When you enable or disable this feature, the dashboard refreshes and the feature is enabled/disabled immediately for all users without any message being issued to the users.
Activating System-wide

Click **Admin** in the top menu, and then **Settings** on the left. Enable **Exploration paths**.
Enabling for a Single Dashboard

From the Navigation Pane, click on a dashboard to open it or create a new dashboard.

**Note:** You can enable this feature only if you are the Owner of the dashboard.

Click ∗: in the top menu, and select **Exploration Paths**.

Click **Got It!** in the confirmation message.
Frequently Asked Questions

Q: Does the AI engine need a warm-up period before showing suggestions?
A: After turning on the feature at the account level, you need to wait 24 hours to let the engine generate suggestions. The other option is to restart the “Sisense.Intelligence” service. This initializes the AI engine, after which the engine will start generating suggestions immediately.

Q: I embed Sisense. Can my viewers use Exploration Paths in an embedded widget or dashboard?
A: The Exploration icon is the entry point to this feature. It is accessible only via the Widget header. Therefore, only when embedding a full dashboard using an iFrame, the title bar appears on widgets, and with it the Exploration icon. When embedding a single widget, the title bar does not appear and therefore, currently, this feature is inaccessible to the Viewer.

Q: What does the ♡ button (“I found this useful”) do?
A: In the initial version (7.4.2), the heart button sends a heartbeat to the Sisense monitoring system. Sisense is tracking usage patterns and user feedback through this button. In the future this button will be connected to the suggestions ranking algorithm. This means that users will be able to influence the order by which suggestions are presented for each measure.

Q: Can the dashboard owner influence the content of exploration widgets presented to users?
A: No, but Sisense plans to support it in the future. In the initial version (7.4.2), the dashboard owner can decide whether to enable the feature on their dashboard, but cannot control exactly what the algorithm suggests to the user.

Note that since the algorithm learns from existing widgets and dashboards in the account, and abides to the security measures and filters applied on each dashboard, the system will learn mainly from the widgets that you and other owners on the account have created in the past. This helps avoiding suggestions that are out of the context of any dashboard.

Q: Can a customer use Exploration Paths in the Sisense Mobile BI App?
A: Not at this time. Sisense is still evaluating what would be the right way to integrate exploration on-the-go. Sisense is keen to hear your feedback. If you have feedback to provide, contact your CSM.
Q: Does this feature work over widgets connected to a Live model?
A: Yes. The feature works on both ElastiCube and Live data models.

Q: For this feature to operate as expected, does the Sisense server need to access a cloud service?
A: No. The feature is run locally on the Sisense instance.
Reporting in Sisense

After you have viewed your dashboards, you may have discovered some insights you want to share. Sisense supports a variety of ways you can share your data with other Sisense users.

Downloading widgets and dashboards as various file types is useful for sharing your data with users or including in your reports and presentations, however, Sisense makes it easy to share dashboards directly with over Sisense users by allowing you to share the dashboard’s URL from the Sisense Web Application or sending your insights via email.

In addition, you can schedule automatic reports to be sent to yourself or other Sisense users. See the Sending Email Dashboard reports for more information.

The topics below describe how you can share widgets and dashboards with other Sisense users:

- Exporting Widgets
- Exporting and Sharing Dashboards
- Downloading Dashboards as an Image
- Creating PDF Reports
- Customizing PDF Reports
Exporting Widgets

You can download widgets in one of the following formats depending on the widget type:

**CSV:** Download the data described by the widget in a CSV file. If you have applied any filters, the filtered data is the data included in the file.

**Excel:** Download the data described by the widget in an Excel file. When you export a pivot table to Excel, the data that is exported is the data that is displayed in your widget. This means that any active filters, layout structure, and masks that you have applied in Sisense to your data are exported as well. For example, if you have modified how currency, percentages, or numbers are formatted in the widget in Sisense, these will be reflected in your exported Excel file. Exporting to Excel maintains your pivot’s layout and structure, and values such as sub-totals, which are not maintained when you export a pivot table to CSV.

**Image:** You can download a widget as an image in PNG format. The image size will reflect the size of the widget on the screen when you download it. To create a larger image, enlarge the widget space in the dashboard, or open the widget in edit view, to get a bigger picture.

**PDF:** Download the widget as a PDF. For Table widgets, you can customize how your table is displayed in your PDF including the table’s orientation and page size. In the PDF Report Settings, the first 14 pages of your Pivot table are displayed in the preview window, however, when you export your Pivot table to PDF, the entire table is included, up to 10,000 rows on multiple pages. For more information, see [Customizing PDF Reports](#).

**To download a widget:**

In dashboard view, click on the widget’s menu, and select **Download** and select the relevant file type.

In edit widget view, click on the download icon, and select **Download** and select the relevant file type.
Exporting and Sharing Dashboards

You can share your Sisense dashboards and widgets with other Sisense users to distribute your Sisense insights.

The topics below describe how to share your dashboards in a variety of formats:

- Exporting Pivot Tables to PDF
- Exporting Pivot Tables to Excel
- Export Tables to PDF
- Exporting CSVs
- Downloading Dashboards
- Downloading Widgets as Images
- Sharing Dashboards
- Sending Email Dashboard Reports
- Email Troubleshooting
Creating PDF Reports

When you need to take copies of your dashboards with you for meetings or sharing with others, you can generate a PDF report of your dashboard. By clicking the PDF icon on your dashboard, you can customize your dashboard’s appearance through the PDF Report Settings page. After you have defined how your dashboard is to be displayed, you can download your dashboard locally as a PDF.

For a quick explanation, see the tutorial below.

To create a PDF report:
From your dashboard menu, click the PDF icon PDF. The PDF Report Settings page is displayed.
Customize how your dashboard is displayed in the PDF. For more information, see Customizing PDF Reports.

Click \(\text{Download}\). The dashboard is downloaded locally as a PDF file.
Customizing PDF Reports

If you need to share dashboards with other users, or create a hardcopy for yourself, Sisense allows you to send an email report to your users or export your dashboard to PDF. Dashboards in email reports or PDFs though look different compared to online dashboard as they serve different purposes. Through the Sisense PDF Report Settings, you can customize and create formatted email reports and PDFs according to your requirements. The Sisense PDF Report Settings allow you to define both the content and design of your report quickly and easily.
From the PDF Report Settings page, you have two modes, Edit mode, for customizing your PDF, and View mode, for seeing how the dashboard will be displayed in the PDF. In Edit Mode, all the settings you can apply to your report are displayed in the left menu.

In Edit mode, you can set the size of your widgets by selecting and dragging the borders of each widget. These borders are displayed when Edit mode is toggled on.

In View and Edit mode, you can define your dashboard’s orientation, its layout, and what is displayed in the header and footer on your report.

After you have customized your report, you can save the settings and use them when sharing reports, or download the report as a PDF.

**To customize a PDF report:**
In your dashboard’s menu, click PDF. The PDF Report Settings page is displayed.

Edit the dashboard by selecting any of the following options:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paper Size</td>
<td>The size of the email report in the PDF.</td>
</tr>
<tr>
<td>Orientation</td>
<td>The orientation of the dashboard in the PDF, landscape or portrait. Landscape orientations display the dashboard horizontally while portrait orientations display the dashboard vertically.</td>
</tr>
<tr>
<td>Header</td>
<td>The design and content of your report’s header. The header is displayed on the top of each page or your PDF report. From the Header list, you can define how your header is to be displayed: Compact, Medium, Large, or No Header. After you have set the design, you can determine what information is included in the header. <strong>Title</strong>: Select to display the dashboard name at the top of the PDF. <strong>Dashboard As Of</strong>: Select to display the ElastiCube name and last build time at the top of the PDF. <strong>Dataset Name</strong>: Select to display the name of the ElastiCube that contains the dashboard’s data. <strong>Dashboard Filters</strong>: Select to display dashboard filter selections.</td>
</tr>
<tr>
<td>Footer</td>
<td>The content displayed in the footer section of each page of your report. You can display the following information: <strong>Page Number</strong>: Select to display the page number on each page of the report.</td>
</tr>
</tbody>
</table>

Click **Save** to save your settings when sharing reports or **download** to download your report.
Managing Dashboards

This section is for Sisense users who have dashboard editing rights (Designers). The topics in this section describe how you can manage your dashboards within Sisense:

- Viewing Dashboards
- Locating Dashboards
- Duplicating Dashboards
- Organizing Dashboards
- Sharing Dashboards
- Deleting Dashboards
- Exporting and Importing Dashboards

For information on improving dashboard performance, see Maximize Dashboard Performance.
Viewing Dashboards

In the **Analytics** page, you can view all the dashboards that you own or that were shared with you.

There are two locations from where you can access your dashboards, on the left side in the Navigation Pane, and the Dashboard area below. In the Dashboard area, your dashboards are displayed as clickable tiles. There are two dashboard views you can view your dashboards with.
Tile View

The Tile view displays the name of the dashboard and its owner. You can click a tile to open the dashboard, or click its menu to see a variety of options. The options displayed are determined by your role and whether or not you own the dashboard.

When you see a yellow tag attached to the tile this indicates that the dashboard is in your system, but you are viewing the owners version, and not your version. This tag is only visible for Administrators. For more information see Managing Your Users Dashboards.

<table>
<thead>
<tr>
<th>Sample - Lead Generation</th>
</tr>
</thead>
<tbody>
<tr>
<td>JB  James B</td>
</tr>
</tbody>
</table>
**List View**

The Dashboard List provides information about your system adoption, such as the number of Viewers of each dashboard, and the last time a dashboard was modified.

The Dashboard List provides the following information:

- **Name**: The name of the dashboard.
- **Owner**: The name of the dashboard’s owner.
- **Data Sources**: The data sources that are being queried by the dashboard.
- **Last Published**: The last time the dashboard was published.
- **Last Modified**: The last time the dashboard was modified. Any changes to the dashboard or to the dashboard metadata (share list, and reporting schedule) affect the “last modified” timestamp.
- **Total Viewers**: This column is only displayed to Administrators. The total number of Viewers who have opened the dashboard. This number may be lower than the number of users the dashboard is shared with if some of them have not opened the dashboard. This number does not include Administrators who opened a dashboard that was not owned by or shared with them.
- **No. of Widgets**: The number of widgets in the dashboard.
Toggling between Views

You can toggle between these views by clicking the View icon of the right-side of the Dashboard area.
Locating Dashboards

You can quickly and easily locate dashboards without having to navigate through your entire list of dashboards. When you enter any characters in the **Search** field, the list of dashboards is filtered according to the characters found within your dashboard titles.

To locate a dashboard by name, source, owner, or who the dashboard is shared with, you can use search operators within the **Search** field. Sisense supports the following operators:

<table>
<thead>
<tr>
<th>What You Can Search By</th>
<th>Search Operator with Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dashboard name</td>
<td>Name:</td>
</tr>
<tr>
<td></td>
<td>Example: Name: Sales</td>
</tr>
<tr>
<td>Data source</td>
<td>Source:</td>
</tr>
<tr>
<td></td>
<td>Example: Source: Sample eCommerce</td>
</tr>
<tr>
<td>Users the dashboard is Shared with</td>
<td>Shared with Example:</td>
</tr>
<tr>
<td></td>
<td>“Shared with: John”</td>
</tr>
<tr>
<td>Dashboard Owner</td>
<td>Owner:</td>
</tr>
<tr>
<td></td>
<td>Example: Owner: John</td>
</tr>
</tbody>
</table>

In addition, you can use a combination of search operators to refine your results further. For example, Name: Sales Source: Sample - eCommerce

**To locate a dashboard:**

In the Navigation Pane, click 📑 to display the **Search** field.
In the **Search** field, enter the title of your dashboard. Any dashboards that contain the characters you entered are displayed.

🔍 Sam Cancel

|🔍  Sample - Ecommerce |
|🔍  Sample - Healthcare |
|🔍  Sample - Lead Generation |
Duplicating Dashboards

Copying a dashboard (even one that was shared with you) creates a new copy of the dashboard with a different name. This new dashboard is totally separate from the original and you are its owner, meaning that you have full editing rights. When you copy a dashboard, the new dashboard is added below the copied dashboard in your Navigation Pane with the same name plus a number in the order it was created.

For example, Sample Ecommerce(1) is a copy of Sample Ecommerce, while Sample Ecommerce(1)(1) is a copy of Sample Ecommerce(1):

- Sample - Ecommerce
- Sample - Ecommerce (1)
- Sample - Ecommerce (1) (1)
- Sample - Healthcare
- Sample - Lead Generation

To copy a Dashboard:
Right-click the dashboard's name in the Navigation Pane and select Duplicate.
Organizing Dashboards

You can organize your dashboards into folders and sub-folders within any of your folders to improve how you navigate the Navigation Pane.

To create a folder:

In the Navigation Pane, click + and select **New Folder**.

The new folder is added to the dashboards list.

Hover over the new folder and click > **Rename** to name the folder.

To create additional sub-folders within your folders, hover over the folder and select > **New Folder** and repeat Step 2.

You can drag and drop your dashboards from other folders or the Navigation Pane to any of your folders.

For information on improving dashboard performance, see [Maximize Dashboard Performance](#).
Deleting Dashboards

To delete a dashboard:
Click on the menu of the Dashboard to be deleted in the list in the Home page, and select **Delete**.

OR
Click the Dashboard Menu and select **Delete Dashboard**.

To delete multiple dashboards:
You can delete multiple dashboards by opening list view, selecting the relevant dashboards, and clicking .

OR
In your Navigation Pane, click the checkbox icon [✓], select the relevant dashboards, and click [✓].

**Dashboards**

- [ ] Sample - Ecommerce
- [✓] Sample - Ecommerce (1)
- [✓] Sample - Healthcare
- [✓] Sample - Lead Generation

---

3 dashboards selected
Exporting and Importing Dashboards

Sisense supports exporting and importing of dashboards. This is useful for copying dashboards between separate Sisense installations, and also for making backups.

Dashboards are exported as files with a .dash extension, which can then be imported.
Exporting a Dashboard

Administrators and Designers can export dashboards and import them into multiple instances of Sisense. When you export a dashboard or multiple dashboards, a single .dash file is created. Afterwards, when the .dash file is imported into another instance of Sisense, all the dashboards are added to the main Analytics page.

To export dashboards to a .dash file:

Option #1 (Single dashboard)

In the Navigation Pane on the left of the Sisense environment, click the dashboard’s menu and click Export Dashboard.

OR

Option #2 (Single dashboard)
Open the dashboard’s menu and click **Export**.

**Option #3 (Multiple dashboards)**

In the Navigation Pane on the left of Sisense, click ✅. This displays checkboxes next to all of your dashboards and folders. Select the relevant dashboards and folders to be exported and click ✅.

In List view, select all the relevant dashboards and click ✅.
All of these methods prompt you to download the .dash file and save it to disk. After you have downloaded the exported file, you or your users can import it into their instance of Sisense. For more information, see Importing Dashboards below.
Importing a Dashboard

To import a dashboard:
Hover over Dashboards list in the home page and click the ‘+’ symbol. Select Import Dashboard from the menu. You can also right-click anywhere in the dashboards list, or click on the menu of one of your existing folders to import a dashboard into a folder.
Select your exported dash file. The dashboard will be added to your list of dashboards or folder. You can drag it from there to your desired location in the tree.
Best Practices for Promoting Dashboard Adoption

One of the pitfalls for successful implementation of BI Analytics in companies is adoption. A company may invest in all the modern technology and create attractive-looking analytics, but still struggle with long-term adoption of analytics. Many Sisense clients have successfully tackled this problem. Below are some creative ideas that might help you drive analytics implementation in your company at the different stages of your journey toward data-driven culture.
Think Ahead at Rollout

When introducing analytics into the organization, you have probably appointed a BI champion or set up an Analytics team. Their job is to gather requirements from all stakeholders, to design the BI solution, and finally, to execute the design. An additional, parallel, objective of this team should be to build a User Adoption Plan. Start thinking about the rollout of the dashboard right from the first phases of the workflow. Making these decisions in the initial planning phases helps to ensure your final dashboards serve your ultimate goal of achieving maximum BI value in your organization.

The best User Adoption Plans are built alongside the dashboards and adjusted as your BI project expands, to ensure full adoption and utilization of BI resources by all of your stakeholders and target audiences.
Provide Ongoing User Training

So your BI person or team has created a brand new set of dashboards to serve your organization. If the rollout of the analytics was successful, pretty soon your BI champion(s) will have their hands full with user requests to create more dashboards and tweak the widgets to answer additional business questions. This can hinder or even derail the further development of your analytics solution. And this is not how it should work. Modern BI solutions are all about self-service, allowing all users to use them to get insights.

That being said, there is still a learning curve. How steep the curve is depends on your BI tool and the users’ technical level, especially if the bulk of your team isn’t tech-savvy. You will have to train users to use the tool and to think analytically.

Below are some suggestions for user training:

**Use the training courses offered by your BI vendor.** Sisense provides a rich offering of courses that cover all stages of the BI flow and cater to all types of uses, from newbies to advanced:
- Business Requirements Gathering Framework
- Business Requirements Recon: The Questions
- Dashboard Planning Document
- Meet Si-Cling (Case Study Introduction) Session
- Know the Data Session
- Identifying Relationships Session

**Conduct quarterly workshops to train your users** - and not just the newcomers. It’s best to include an objective-oriented component in the workshop, which means that users should come with a business question they struggle with. By the end of the workshop, they will have created a dashboard exploring their actual business questions, but also will have learned a lot about the system and built their confidence with an experience of success.

**Offer weekly office hours** for one-on-one training and support of your users.

**Provide resources online** - dashboard documentation, Q&A on the most frequently asked questions, etc. You can provide this as part of your internal user portal, web site, etc.
When the users across the organization are all inspired analytics users, do you still need a dedicated BI resource? Of course. Business users are not familiar with your data and data sources. They do not know much about data cleansing, ETL processes, many-to-many joins, etc. It is good practice to leave the data governance to professionals, and allow business users to take it from there. And when all these training efforts are in place, they are guaranteed to save hundreds of hours for your BI resources.

Learn how one of Sisense clients does this in this webinar: Increasing Adoption of BI Within Your Organization.
Promote Across all Levels of the Organization

When your BI team rolls out a new dashboard, how can you promote it across the organization and make sure that users are aware and integrating it into their ongoing, long-term work processes? Obviously, the number of paths to take depends only on the creativity of those involved. Below are a few ideas that were implemented successfully with Sisense clients:

**Promote dashboards through internal email announcements.** In Sisense, that’s easy to do by sharing the new dashboard with all or just some users across your organization in one action. For details, see [here](#).

**Prepare training on each new dashboard** - this can be a short video or one-pager that you upload to your internal BI portal. State clearly who the audience is for this dashboard, and how the audience will benefit from using it. Communicate clearly all the changes you make to existing dashboards. It’s crucial that users understand the outcome of each change. Otherwise, when they come to the dashboard and do not see what they saw last week and what they expect to see today, their notion of the dashboard as a single source of truth might be undermined.

**Engage users with dashboard contests.** This solution is as simple as it is brilliant. Dashboard contests can encourage users across the organization to make practical use of business intelligence, becoming well-versed in using the system in the process. It can also help you identify the specific datasets that most people in the organization need to use. To learn how this benefited one of Sisense clients, see [here](#).

**Create an end-user steering committee (COE).** For most organizations, one of the main stumbling blocks to BI adoption is building consensus between the stakeholders. To overcome this problem, make sure to invite the three key groups of organizational stakeholders in relation to analytics: business users, business analysts, and data engineers. To learn more about building organizational consensus for a BI platform, see [here](#).

**Appoint executive sponsorship to enforce usage.** An effective sponsor for a BI-related initiative would be a senior-level executive with a data-driven outlook, whose team will benefit from the introduction of this initiative. This sponsor will have to define the business scope of the
initiative, set up business priorities and delineate a business strategy required to adopt analytics in the organization.
Monitor Usage

Once analytics are in use across the organization, it is important to monitor the usage of the different dashboards. This includes tracking the amount of users using each dashboard, the amount of users creating their own dashboards, tracing what the uses do in dashboards, and so on.

If usage is low, the next step is to gather feedback from your users to understand why this is so. This might happen due to varying reasons, such as users see no value in analytics, lack the knowledge to use your analytics tool, aren't clear on the objective of certain dashboards, lack the executive sponsorship to enforce usage, etc. When you know the reasons, you can correct .... and repeat for as many times as necessary, till the time comes to retire the dashboard. A handy feedback questionnaire for users is suggested in our BI Implementation Methodology handbook.

Is there an easy way to monitor analytics usage? Yes. in Sisense, usage analytics is a built-in dashboard that enables you to monitor your Sisense activity across all users and business dashboards. With Sisense Usage Analytics, you can better understand how users interact with Sisense and optimize your configuration accordingly. For more information, see here.
Introduction

Sisense Pulse

Sisense Pulse is a centralized location where you can stay on top of your most important KPIs across multiple dashboards or manage your data and build alerts. For example, you can consolidate and monitor important aspects of your business by adding important values from widgets from various dashboards to the Sisense Pulse page.

Then, you can create alerts to notify you when certain thresholds are met or anomalies in your data are detected. Sisense Pulse provides you with access to your most important data and notifies you when to take action.

The following sections describe Sisense alerts and managing alerts and KPIs in Sisense Pulse.

Sisense Pulse

Overview

Managing Pulse Tiles
Alerts

- Overview
- Creating Data Alerts
- Creating Build Alerts
- Advanced Alert Settings
Sisense Alerts

This topic describes the two types of alerts you can define, data alerts and system alerts in Sisense.

Data alerts are triggered by changes to your data. Sisense provides several options for defining the type of changes that trigger data alerts including threshold based alerts, automatic anomaly detection, and Always which is triggered by any change to your data. For example, if you have an Indicator widget that represents total sales, Sisense can notify you when a threshold you defined is met or exceeded.

Build alerts can be configured to be sent following build attempts. When creating an alert, you can customize your notifications, the channel through which they are sent, and how often a notification is sent following a data or build.

After you create an alert, the alert is added to the Sisense Pulse page where you can manage the alert and its notifications. For example, you can delete alerts or enable and disable notifications.

If you want to make a similar alerts, for example, alerts with different notifications for different groups of users, you can duplicate the alert and modify its settings.

When an alert is triggered, you and your users are notified through the channel you defined when creating the alert and through the Pulse Feed. The Pulse Feed is displayed in the Sisense Web Application and shows a list of your most recent triggered alerts. From the Pulse Feed you can open the dashboard from where
<table>
<thead>
<tr>
<th>Message</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The alert originated or open the Sisense Pulse page.</td>
<td></td>
</tr>
<tr>
<td>Build Alert - Success</td>
<td>10/13/16 9:20 AM</td>
</tr>
<tr>
<td>The ElastCube Test on localhost was built successfully.</td>
<td></td>
</tr>
<tr>
<td>Converted (%)</td>
<td></td>
</tr>
<tr>
<td>Converted (%) = 7.29%</td>
<td>10/31/16 9:35 AM</td>
</tr>
<tr>
<td>Converted (%)</td>
<td></td>
</tr>
<tr>
<td>Converted (%) = 7.29%</td>
<td>10/25/16 3:55 PM</td>
</tr>
<tr>
<td>Average Cost</td>
<td></td>
</tr>
<tr>
<td>Average Cost = $15.65</td>
<td>10/31/16 9:35 AM</td>
</tr>
<tr>
<td>Average Cost</td>
<td></td>
</tr>
<tr>
<td>Average Cost = $15.65</td>
<td>10/25/16 3:55 PM</td>
</tr>
<tr>
<td>Build Alert - Success</td>
<td>10/25/16 10:12 AM</td>
</tr>
<tr>
<td>The ElastCube Test on localhost was built successfully.</td>
<td></td>
</tr>
</tbody>
</table>
Related Topics

Creating Data Alerts
Creating Build Alerts
Creating Data Alerts

This topic describes data alerts and how you can add them to Sisense Pulse. Data alerts are triggered by conditions you define for your widgets. For example, if you have an Indicator widget that represents your company’s revenue, you can define an alert that notifies you when this revenue reaches or falls below a certain threshold.

Sisense supports alerts for the following widgets:
- Indicator Widgets
- Area Chart Widgets
- Bar Chart Widgets
- Column Chart
- Line Chart

For each type of widget you can add to Pulse, you can monitor a single value. For example, for Column charts you can monitor a single section (value) of a column. To monitor multiple values, you can add each value you want to monitor to Pulse separately.

When creating data alerts, you define the condition that triggers the alert. Sisense provides three alert conditions that can trigger an alert:

 Threshold
Sisense notifies you when a threshold is reached following a build. This condition is useful for monitoring metrics that follow a regular, cyclical schedule.

 Automatic
Sisense automatically notifies you when the Sisense alert engine identifies an anomaly in your data based on machine learning algorithms. The Sisense algorithm actively learns from your data and eliminates noise to identify anomalies in your data. Sisense’s anomaly detection is based on a smoothing algorithm that takes the latest values into account. The more times an ElastiCube
is built, the more accurate Sisense becomes at identifying any anomalies and notifying you.

Sisense notifies you after each build, regardless if there is a change in your data or not.

In addition, you can configure advanced settings that determine who receives the alert and what channel the alert is sent through. For more information see Advanced Settings.

**To create a data alert:**

For an Indicator widget, from the widget’s menu, select **Add To Pulse**. For other types of widgets, right-click any value in the chart and select **Add To Pulse**. The Add to Pulse window is displayed.

**Add to Pulse**

<table>
<thead>
<tr>
<th>Name</th>
<th>Total Cost of admission</th>
</tr>
</thead>
</table>

**Alert Condition**

- **Threshold**
- **Automatic**
- **Always**

The system will monitor this value and notify you when anomalies are detected.

**Advanced**

In the **Name** field, enter a meaningful name that represents your alert. In the Alert Condition area, select the condition that triggers the alerts. **Threshold**: Sisense defines the threshold value as the current value displayed in the widget. You can increase or decrease this value and define the equality and relational operators that determine when the threshold is
Automatic: Sisense automatically identifies anomalies in your data and alerts you.

Always: Sisense notifies you every time the data changes.

Click **Add** to create the alert. The alert is added to Sisense Pulse. For additional advanced configuration options, see [Advanced Settings](#).
Creating Build Alerts

Administrators can set build alerts to notify themselves or other users when a build fails or succeeds. Build alerts are created from Sisense and are sent following an ElastiCube build.

Build alerts allow you to monitor the status of ElastiCube builds.

Administrators can create build alerts from the **Admin** page. After a build alert is created, it is added to Sisense Pulse from where you can manage it.

**To create a build alert:**

- In Sisense, select **Admin > Data Sources > Add Build Alert**.
- OR
- From the ElastiCube menu, **Add Build Alert**.

The Add Build Alert window is displayed.
In **Name**, enter a name for the alert.

<table>
<thead>
<tr>
<th>Add Build Alert</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td><strong>Server</strong></td>
</tr>
<tr>
<td><strong>ElastiCube</strong></td>
</tr>
<tr>
<td><strong>Notify when</strong></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

From the **Server** list, select the server where the ElastiCube is hosted. From the ElastiCube list, select the ElastiCube(s) that triggers an alert following a build.

Select when to notify you following a build. There are three conditions that when met, can trigger an alert:

- **Build failed**: The alert is sent after the ElastiCube build fails.
- **Build succeeded**: The alert is sent after the ElastiCube is built successfully.
- **Build succeeded after failure**: The alert is sent following a successful build after the ElastiCube previously failed to build.

Click **Add**. The alert is added to your Sisense **Pulse** page.

For additional advanced configuration options, see [Advanced Settings](#).
Sisense Pulse enables you to stay on top of your most important KPIs across all your dashboards and your latest ElastiCube builds. By adding important KPIs from your dashboards to Sisense Pulse, you can get a comprehensive picture of your data from a single location.

Sisense Pulse contains tiles that display information from your dashboards and the status of your ElastiCube builds.

Within Sisense Pulse, there are two types of tiles, data tiles that display your data's current status and recent history and build tiles that display the latest status of your ElastiCube builds.

You can access Sisense Pulse from the menu bar of the Sisense Web Application.

By hovering over the header of any tile in Sisense Pulse, you can view additional options for your tiles.

The icon opens a new tab in your browser. For data tiles the dashboard where the widget originated is displayed and for build tiles the Data Sources section in the Admin page is displayed.

The icon is a tooltip that describes the tile in more detail. For data tiles the tooltip describes who added the tile to Pulse and when it was added. Build tile tooltips describe when the tile was created, which ElastiCube is being monitored, and under what conditions an alert is triggered when alerts are enabled.
The icon displays a menu for managing tiles in Sisense Pulse. For more information regarding managing tiles in Sisense Pulse, see Managing Tiles in Sisense Pulse.
Adding Tiles to Sisense Pulse

To keep track of your important KPIs and ElastiCube builds, you add the relevant data and build tiles to Sisense Pulse. The flow for adding tiles to Sisense Pulse is the same for creating alerts for Indicator widgets and ElastiCube builds. For more information about these flows, see the following topics:

For data tiles, see Creating Data Alerts.
For build tiles, see Creating Build Alerts.

Each of these flows adds tiles to Sisense Pulse and creates an alert. If you want to monitor your data and ElastiCube builds without alerts, see Deactivating Notifications.
Managing Tiles in Sisense Pulse

This topic describes how to manage your Pulse tiles and notifications when an alert is triggered.

After you have created alerts or added tiles to Sisense Pulse, you can manage those tiles from Sisense Pulse. Through the Sisense Pulse page, you can duplicate alerts and modify them, disable notifications, and delete alerts.
Duplicating Tiles in Sisense Pulse

You can create copies of alerts and modify them to quickly generate new alerts with different notification messages or notify various groups across separate channels.

For example, if you want to notify Group 1 via email and Group 2 via Slack when the same event occurs, you can duplicate an alert and modify the notification channel.

To duplicate a tile:

In the Sisense Web Application, select **Pulse**.

In the relevant alert’s menu, select **Duplicate**. A copy of the alert is created and the Add to Pulse window is displayed.
You can modify the alert’s settings from the **Add to Pulse** window. For more information about these settings, click [here](#).
Deactivating Your Notifications

Each time you add a tile to Sisense Pulse or create an alert, by default, you will receive notifications when the event is triggered. If you do not want to receive any notifications, you can disable notifications from being sent by toggling the notifications. When you toggle the notifications, you no longer receive notifications when an alert is triggered, however, other users with whom the alert was shared will continue to be notified. See Deactivating Notifications for All Users to turn off notifications for everyone associated with an alert.

To deactivate notifications:

In Sisense, select Pulse.

From the Tile menu of the relevant alert, toggle Receive Notifications.
Deactivating Notifications for All Users

If you have added an alert that is no longer relevant or you want to deactivate the alert for all users, you can turn off the alert.

When you turn off an alert, no users are notified when the event that triggers the alert occurs.

To deactivate notifications for all users:
In the Sisense Web Application, select Pulse.
From the Tile menu of the relevant alert, select Turn Off. A confirmation dialog box is displayed.

Click Yes to confirm that you want to turn off all notifications for the alert. The alert is turned off. Alerts that have been turned off appear in the Pulse page as shaded tiles. You can reactivate the alert by selecting Turn On from the Tile menu.
Deleting Tiles in Sisense Pulse

When a tile is no longer needed, you can delete the tile through the Sisense Pulse page. Deleting an alert removes it from the Sisense Pulse page and users are no longer notified.

To delete an alert:
   In Sisense, select Pulse.
   In the relevant alert’s menu, select Delete.
Removing Sisense Pulse

You can remove the Pulse page from Sisense and deactivate all the alerts defined in Sisense Pulse from the Admin page. This is useful if you have embedded Sisense and your end users do not need to receive alerts.

To remove Sisense Pulse:

In Sisense, in the top menu, open the Admin page and select Settings.

Toggle the Pulse switch to disable Sisense Pulse. The Sisense Pulse tab on the top menu is no longer displayed and no new notifications will be sent.
Advanced Settings

By default, when an alert is triggered, you are notified via your email address registered with Sisense.

You can also send alerts using more advanced configurations that can let you notify additional users when an alert is triggered, notify them you across multiple channels, and allows you to customize the message displayed in the notification.

The following topics describe the functionality available in the Advanced Settings options:

- "Notifying Additional Users" on page 800
- "Setting Notification Channels" on page 802
- "Customizing Notification Messages" on page 816
- "Configuring Additional Options" on page 814
Notifying Additional Users

When an alert is triggered, Administrators and Designers can notify additional Sisense users and groups by configuring the alert’s advanced settings. Viewers cannot share alerts with additional users.

Users who are notified receive a notification through Sisense Pulse and any other additional channels you define.

**To notify additional users:**

For Build Alerts, select the `

<table>
<thead>
<tr>
<th>Add Build Alert</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Type name...</td>
</tr>
<tr>
<td>Server</td>
<td>LocalHost</td>
</tr>
<tr>
<td>ElastiCube</td>
<td>All</td>
</tr>
<tr>
<td>Notify when</td>
<td>Build failed, Build succeeded after failure</td>
</tr>
</tbody>
</table>
Enter the email address, Sisense username, or group name of the users you want to notify when an alert is triggered. Entering Everyone specifies that all Sisense users and user groups associated with your account are notified. The contact information of the people you add is listed in the center of the window and is displayed the next time you open this window. Click Add to add the alert to Pulse.
Setting Notification Channels

By default, your users are notified via their email addresses registered with Sisense. In addition, Sisense supports a variety of workflows by sending notifications across multiple applications and services. For example, your users can receive notifications through the Sisense mobile application, Zapier, Slack, and other 3rd services via webhooks.

**To define how users are notified:**

For Build Alerts, select the 📣 tab.

For Data Alerts, select **Advanced Options** 📣.

---

### Add to Pulse

#### Message

Type a message to be sent with the alert...

#### Notifications

- Email
- Mobile
- Slack
- Zapier

شعاع Add Webhook

---

### Additional Options

#### Basic

Add | Cancel
In the Notifications area, select the relevant channel for your notification. The default channel, Email, is already selected.
You can select one or more of the following channels:
   Email
   Mobile
   Slack
   Zapier
   Webhook
   mobile
Click **Save**.
Sisense Mobile

Sisense Mobile enables users to view Sisense dashboards through their mobile devices. You or your users can receive notifications through Sisense Mobile. After selecting the users you want to notify, any users with a Sisense user account and
Sisense Mobile will receive a notification when an alert is triggered.
Slack

Slack is a real-time team collaboration messaging platform that allows you to receive incoming Webhooks from external sources such as Sisense. To send notifications through Slack, you need to enter the URL and optionally, your Slack channel or username, where the notification is to be displayed. You can ask your Administrator to provide you with the Slack URL and channel. You can obtain these details by adding the Incoming WebHooks app into your Slack configuration. For a quick walkthrough of configuring Pulse notifications in Slack, see the tutorial below.

To obtain a Webhook URL and channel from Slack:
Add the Incoming WebHooks app to Slack. You can find this app in Slack’s App Directory. In the Incoming WebHooks app page, click **Add Configuration**. After you add the Incoming WebHooks app, you can select which channel to post the notifications to. The channel name is optional and displays the notification in the channel associated with your WebHook URL. In the Post to Channel area, select the channel where the notification is to be displayed.
and click **Add Incoming WebHooks integration.**

In the Setup Instructions the Webhook URL is displayed. Copy the Webhook URL.
In Sisense when sending notifications through Slack, in the **URL** field, enter the URL copied in the previous step and in the `#channel/@username` field, enter the channel name if you have defined one.
Zapier

Zapier is an online service that lets you create automated actions connecting disparate business apps and services such as Sisense. To receive notifications through Zapier, you must create a Zap and provide Sisense with the relevant URL when configuring your alerts’ notifications.

When an alert is triggered, Sisense sends the notification to the Webhook you define in Zapier.

You can ask your Administrator to provide you with the Zapier URL.

**To obtain a Webhook URL from Zapier:**

Log in to Zapier and click **Make a Zap**. The Trigger App page is displayed. From the Trigger App page select Webhooks. The Connect to Webhooks by Zapier page is displayed.

In Step 1 > Set Up Webhook, click **Copy to clipboard** to copy the Webhook URL you need to provide to Sisense.

In Sisense, when sending notifications through Zapier, in the **URL** field, enter the URL copied in the previous step.
Webhooks

Webhooks are a system of automated notifications that indicate that an event occurred, in this case an alert being triggered. To send notifications through additional 3rd party channels, you can select Webhook and enter the Webhook name and URL.
Your Webhook server listens for incoming messages from Sisense across the URL specified in the URL field.
You can ask your system administrator to provide you with the Webhook URL. When Sisense sends notifications to your custom Webhook, they are sent within a JSON payload. Your Webhook provider must interpret the payload and display the notification to your users. The following are examples of notifications sent in a JSON payload and a description of the contents of the payload:

```
...

Data Events


Build Events

| { type: 'build', name: 'Success Event', measure: null, value: 1, |
```

value: 39759625.51502721,
condition: 'none',
filters:
    [ { asDimension: false,
        title: 'Category',
        value: 'Include all',
        jaql: [Object],
        panel: 'scope' },
     { asDimension: false,
        title: 'Gender',
        value: 'Include all',
        jaql: [Object],
        panel: 'scope' } ]

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>String</td>
<td>Type of alert. ‘kpi’ for data events and ‘build’ for build events.</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>Name of the alert.</td>
</tr>
<tr>
<td>Message</td>
<td>String</td>
<td>Notification message.</td>
</tr>
</tbody>
</table>
| originDashboardLink   | String   | Dashboard ID provided by Sisense. You can find the dashboard ID from the URL of the dashboard. For example, the ID of the dashboard below if: http://localhost:8081/app/main#/dashbo
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Double</td>
<td>For data events when a threshold condition is applied, this is the value of the threshold. For build events, the value is 0 for failed builds and 1 for successful builds.</td>
</tr>
<tr>
<td>condition</td>
<td>String</td>
<td>Type of condition that generated the alert.</td>
</tr>
<tr>
<td>filters</td>
<td>Object</td>
<td>Object that defines filters applied to the dashboard.</td>
</tr>
</tbody>
</table>
Configuring Additional Options

You can configure how often and when a notification is sent after an event is triggered. For example, you can limit the amount of notification sent to one when a data alert is triggered. This is useful if you build your ElastiCubes frequently, but do not need multiple notifications.

To configure additional options:
When adding or editing a widget to Pulse, in the Advanced Options window, select `Additional Options`.

Select any of the following options:

**For data alerts:**
Select **Notify only once after a condition is met** to limit how many notifications are sent after an event occurs.
Select **Notify when a condition is no longer met** to notify users after the event is no longer triggered.

**For build alerts:**
Select **Notify once when there is a status change** to notify users only once when the latest build status is different than the previous build status.

Click **Add** to save the alert.
Customizing Notification Messages

By default, the following message is provided in your email alert notifications:

Hi,
The latest value in the KPI that you are monitoring is:
Value

You can provide customized messages for your notifications when configuring your alert through the alert’s Advanced Settings. The message you provide replaces the string: The latest value in the KPI that you are monitoring is:

For example, if your message is “The last value of my KPI is:” the message in your alert’s notification will be:

Hi,
The last value of my KPI is:
Value
To customize your notification’s text:
For Build Alerts, select the 📣 tab.

Message
Type a message to be sent with the alert...

Notifications
- Email
- Mobile
- Slack
- Zapier

Add Webhook

Additional Options

Basic
Add Cancel
For Data Alerts, select **Advanced**.

In the Message area, enter your custom text. Click **Save**. The next time a notification is sent, your custom message is displayed in the notification.

You can modify your alert notifications further including customizing the text and images displayed. For more information, see [Rebranding Sisense Automated Emails](#).
Administration

Sisense supports several "Sisense User Roles" on page 46 with Administrative privileges for customizing Sisense for your company. The following roles: System Administer, Data Administrator, and Administrator have varying levels of access to your system including servers, users, ElastiCubes and more. The Admin page is your dedicated location in your web application for managing your system.

This section describes how your Sisense Administrator can configure and manage users, groups, data sources, and system settings from the Admin page.

Note: The Admin page is only displayed to users with the relevant privileges. In addition, certain settings are only available to certain user roles except for Administrators who have access to all settings.

"Managing Sisense Users" on page 821
"Managing Your Users Dashboards" on page 893
Data Source Management
"System Settings" on page 907
"Embedded Analytics" on page 955
"Sisense Security Overview" on page 1026
"Sisense Migration" on page 934
"Multi-Node Deployments" on page 1056
Managing Sisense Users

Sisense Administrators have access to the User Management section of the Admin Page where users, groups, and advanced user settings can be configured. The topics below describe how to create users and groups, and allow users you authenticate to access your implementation of the Sisense Web Application.

- Sisense User Roles
- Managing Dashboard Users
- Managing User Groups
- Managing User Sessions
- Customizing User Roles
- License Utilization
- Integrating Active Directory
- Introduction to SSO
Managing Dashboard Users

This topic describes how to add and manage Sisense users that can access Sisense dashboards.

Users are managed from the Admin page through the Users tab.

In the Users table, you can edit and delete users in your system. The table provides the following information about your users:

**Username**: The user's username in Sisense.

**Email**: The user's email address.

**Name**: The name of the user.

**Groups**: The groups, if any, that the user belongs to. By default, the System Administrator belongs to a group called Admins.

**Role**: The user role assigned to the user.

**Origin**: The origin of the user. This is typically your company's name.

**Last Active**: The date of the last time the user was active. Activity includes opening any of the Sisense tabs (Data, Analytics, Pulse, and Admin), viewing dashboards, working with widgets, or exporting reports. This field is blank if the user has never been active in Sisense.
**Days Since Last Active:** The number of days that have passed since the user last was active. This field is blank if the user has never been active in Sisense.
Adding a New User

You can add users using the following methods:

- Add new users by entering them one by one or in bulk
- Add new users from your Active Directory database. Click here to learn how to configure your Active Directory in Sisense

To add a new user:

Click + Users. If a connection with Active Directory also exists, an additional Import Many option is available. Click it to select a user from Active Directory.

Fill in the user’s details in the window.

(Optional) The Language for your account is the language set by an Administrator for your group or server. From the Language list, you can override those settings and select the language you prefer to be displayed in the Sisense Web Application.

Select the Role of the user:

**Admin**: Admins can access the Admin page of the Sisense Web Application from where they can manage ElastiCubes, users, user groups and more. Sisense supports multiple Admin users.

**Data Admin**: Data Admins can manage all data sources in the system. They have access to the Admin page of the Sisense Web Application from where they can manage ElastiCubes and Live connections. Data Admins can also add and manage servers. This is useful for migrating
Sisense across environments. Data Admins do not have access to User Management and System Configuration.

**Data Designers:** Data Designers can create and edit ElastiCubes they created or were shared with them. Data Designers can access the Admin page of the Sisense Web Application from where they can manage the ElastiCubes and Live Connections they have access to in the Data Sources section. In addition, Data Designers can see servers, but can not add new servers. Data Designers do not have access to User Management and System Configuration.

**Designers:** Designers can create, design, edit and share Dashboards. Sisense Designers determine whether the user with who they share a Dashboard has editing rights (is a Designer) or only viewing rights (Viewer).

**Viewers:** Viewers can view, explore, drill down, make selections and filter the Dashboards that Designers share with them. Viewers only need a standard web browser; no plugins or downloads. They cannot create new Dashboards or edit existing ones.

Click **Save**. The user then receives an email from Sisense that redirects the user to activate their account and set a password.

**To manually set a password for the new user:**
Set the **Define Password** option to **ON** and enter a password before saving.

<table>
<thead>
<tr>
<th>Define Password</th>
<th>OFF</th>
<th>ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set Password</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Confirm Password</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In this case, the user receives an invitation email. You must personally notify the user of the password that you set for them.
Adding Multiple Users

To add multiple users:

Click the **Import Many** option at the top right of the Add users window. In the displayed text box, paste a comma separated list of emails and click **Next**.

The next screen enables you to assign User rights to the users that you entered and to verify that the user count and permissions match your license.

After you click **Add**, each user receives an activation email.
Editing an Existing User

To edit an existing user:
   Click Admin and select the Users tab on the left.
   Click ✂ on the right of the user’s entry in the list.
Deleting a User

To delete an existing user:
   Click **Admin** in the upper right corner and select the **Users** tab on the left.
   Click at the right of the user’s entry in the list.
Managing User Groups

This topic describes how to add and use user groups in Sisense.

**How Do User Groups Work?**

When building and sharing dashboards, wherever you can select a user, you can also select a user group. For example, you can share a dashboard with an entire user group at once.

When you add an additional user to an existing user group, the user receives the role you assigned to them when you created that user.

When users are added in Active Directory to a previously imported Active Directory user group, then those users are automatically added to the corresponding Sisense user group.

User groups can only contain users. They do not support nested user groups. Additional users cannot be added to a user group that was imported from Active Directory.

You can add users to a user group using the following methods:

- Add existing Sisense users to a user group
- Add a user group from Active Directory
Accessing User Group Management

You can add, edit and delete user groups, and assign users to them.

To access the user group settings:
   Click Admin and select the Groups tab on the left. A list of the previously defined user groups is displayed. The Origin column displays Sisense or Active Directory to indicate whether the user group was defined in Sisense or imported from Active Directory.

If no user groups have yet been defined, the following message is displayed: No Groups Found
Adding a New User Group

The following procedure describes how to add a new user group by selecting users from existing users in Sisense.

See Adding a User Group from Active Directory for a description of how to add a user group from Active Directory.

**To add a new user group:**

Click **Admin** and select the **Groups** tab on the left.

Click **Add Group**.

In a **Group Name** field, enter the name for this user group.

In the **Group Users** field select each of the users that belong to this group. To add each user, start typing in the user’s name. An auto-complete list is displayed showing the users that already exist in Sisense. Select a username to add to the group.

**Note:** If you want, you can leave the group empty and add members to it later.

(Optional) The **Default Language** for a group is the language set by an Administrator for that server. From the Default Language list, you can select another default language for a group. For example, if your server supports groups located in different countries, you can set the language for each group.
(Optional) Select the default user role. If you have implemented SSO over SAML, each member of the group is assigned this role when they log in for the first time.

(Optional) The **Session Inactivity Timeout** options are displayed only if you have selected Session Inactivity for your session management. You can use the system default for a group or manually set how many minutes must pass before the group’s users are logged out.

**Note**: If the user belongs to multiple groups, the highest timeout settings are applied to that user.

For more information about these settings, see [Managing User Sessions](#).

Click **Create**.
Adding Users to an Existing User Group

The following procedure describes how to add one or more users to an existing user group.

**To add multiple users to a user group:**
- Click **Admin** and select the **Users** tab on the left.
- Select the relevant users.
- Click **+ Add to Group**.
- In the **Group Users** field, you can type additional usernames to add more users. An auto-complete list is displayed showing the users that already exist in Sisense. Select a username to add to the group.
- Click **Add**.
Adding Users to an Existing User Group

The following procedure describes how to add one or more users to an existing user group.

**To add multiple users to a user group:**
- Click **Admin** and select the **Users** tab on the left.
- Select the relevant users and click **Edit** to edit their profile.

![Edit AD User: Adrienne Valvo](image)

Click **+ Add to Group**.

In **Group Users**, you can type additional groups to add the user to the groups. An auto-complete list is displayed showing the Sisense and Active Directory groups that already exist in Sisense. Select a group to add the user to it.

Click **Save**.
Editing an Existing User Group

To edit an existing user group:
Click Admin and select the Groups tab on the left.
Click on the right of the user group’s entry in the list.

Note: To remove a user from the Group, hover over the user name in the list and click .
Deleting a User Group

When a user group that was entered in Sisense is deleted, its users are not deleted. They remain active users in Sisense.

When a user group that was imported from Active Directory is deleted, a prompt is displayed asking whether you want to also delete all the users in this user group from Sisense.

**To delete an existing user group:**

Click **Admin** and select the **Groups** tab on the left.

Click 🗑️ at the right of the user group’s entry in the list.
User Sessions

You can view a list of all the users currently logged into your Sisense account in the Session Table.

The Sessions Table provides the following information about the users currently logged into your Sisense account:

- **Username**: The Sisense username
- **Name**: The user's first and last name
- **Domain**: The part of the user’s email address after the @ sign
- **Current Sessions**: The number of sessions the user has open
- **Session Duration**: The amount of time from the first session until now

The Session Table is useful for monitoring who is currently logged in and who might be abusing their license. For example, if you need to declare some downtime, you can view the currently logged in users and end their sessions manually by revoking their session.
If you believe that a user is abusing their license, you can check the **Current Sessions** column to see how many sessions they currently have open. Users can open multiple tabs in their browser to open multiple sessions, or by sharing their account information with other users. To view which users have the most sessions currently open, click the **Current Sessions** table heading.

**Note:** User session in Sisense Mobile are not included in the Sessions table.
Revoking a Session

From the Session Table, you can revoke a user's session. Revoking the session ends the user's current session. It does not revoke the user’s API tokens. When a user’s session ends, the next time the user tries to perform an action in Sisense, a message is displayed that the session has ended and your users are prompted to log in again.

To revoke a session:

In the Admin page, select Sessions Table. Locate the relevant user in the Session table. You can click ⏰ and enter the user’s information in the Search field to locate the user quickly. As you begin to type, the results are filtered.

For the relevant user, click ⏰ and confirm that you want to revoke the session by clicking Revoke. The next time the user performs an action in Sisense, they are redirected back to the Login page.
Setting Session Inactivity Timeouts

When a user signs into Sisense, a session cookie is stored in their browser. The session cookie allows the user to remain logged in and authenticated even after ending a session for 7 days.

For security reasons, you may want to end your user’s session sooner. Sisense provides two methods for ending a user session, according to the user’s cookie or by the amount of session inactivity as recorded by Sisense.

When a user’s session ends, the next time the user tries to perform an action in Sisense, a message is displayed that the session has ended and your users are prompted to log in again.

After making any changes to your users’ session timeout settings, all of your users’ sessions are terminated and the user is asked to log in again.

If you have implemented SSO, your users will be logged out and redirected to the IdP to reauthenticate.

To set the session inactivity timeouts:

In Sisense, open the Admin page and select Settings.

In the Session Management area, select the method you want to use for defining when a user’s session ends:

**Cookie:** Allows you to define how much time must pass before a user’s session is ended according to the user’s cookie. This is the default behavior for Sisense.

**Expiration(days):** The amount of days that must pass for a user’s session to end. The default is 7 days. You can enter 0 so your users' sessions do no expire so long as they have the cookie.

**Show "Remember Me" checkbox on Login screen:** Toggle to remove the Remember Me checkbox from the Login screen. Disabling this option means that if your user ends a session, they must log in again
the next time they try to access Sisense.

Session Inactivity: Allows you to define how many minutes of inactivity must pass before a user or an Administrator is logged out.

Note: By selecting this option, you can set a custom session timeout for your Sisense user groups. See Managing User Groups for more information.

Default Inactivity Timeout (min.): Determines how much time must pass before a user is logged out. The default is 30 minutes.

Admin Inactivity Timeout (min.): Determines how much time must pass before an Administrator is logged out. The default is 30 minutes.

Click Save to save your changes.
Customizing User Roles

This topic describes how to customize the permissions given to the Sisense user roles, using Sisense’s REST API.

Click here to learn more about each role type.

For a short video tutorial, see below:
Customizing Roles

You can customize the default roles with Sisense’s REST API, by defining which permissions to enable or disable for each role. For example, you can prevent a user from drilling into widgets, or exporting data to a CSV file. On the other hand, you may want to let viewers change the widget type and format the style.

The following table lists all the actions that you can customize using the API. 'True' and 'False' indicate values that you can change. 'Never' indicates values that you cannot change.

### Dashboards

<table>
<thead>
<tr>
<th>Action</th>
<th>Admin</th>
<th>Data Admin</th>
<th>Data Designer</th>
<th>Viewer</th>
</tr>
</thead>
<tbody>
<tr>
<td>create</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>Never</td>
</tr>
<tr>
<td>Create a new dashboard.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>duplicate</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>Duplicate an existing dashboard.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>change_owner</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>Change the dashboard' owner.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>toggle_edit_mode</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>Select between Layout and View modes in the dashboard. Layout mode is the default mode in which you can add new widgets and rearrange them on the dashboard. In View mode, the dashboard cannot be rearranged</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Admin</td>
<td>Data Admin</td>
<td>Data Designer</td>
<td>Viewer</td>
</tr>
<tr>
<td>------------------</td>
<td>-------</td>
<td>------------</td>
<td>---------------</td>
<td>-------</td>
</tr>
<tr>
<td>edit_script</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>Never</td>
</tr>
<tr>
<td>Edit a dashboard using the Sisense JavaScript API.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>export_dash</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>Never</td>
</tr>
<tr>
<td>Export a dashboard as a .dash file that can be imported back into Sisense.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>export_jpeg</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>Download a widget as an image in PNG format. The downloaded image represents the current view.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>export_pdf</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>Export the dashboard to PDF format. The exported file includes the current view.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>restore</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>Revert changes made in the dashboard to the latest dashboard that was shared.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>copy_to_server</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>Copy the dashboard to another server.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>import</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>Never</td>
</tr>
<tr>
<td>Import a .dash file into the Sisense Web Application.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Dashboards</strong></td>
<td><strong>Admin</strong></td>
<td><strong>Data Admin</strong></td>
<td><strong>Data Designer</strong></td>
<td><strong>Viewer</strong></td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-----------</td>
<td>----------------</td>
<td>-------------------</td>
<td>------------</td>
</tr>
<tr>
<td>select_palette</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>Never</td>
</tr>
<tr>
<td>Change the selected color palette.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>replace_datasource</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>Never</td>
</tr>
<tr>
<td>Select a new data source for a dashboard.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>undo_import_dash</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>Filters</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>create</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>Create a new filter.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>on_off</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>Switch filters on or off.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>toggle_expansion</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>Enable the expansion of filter settings to see more filter parameters.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>toggle_auto_update</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>Include the option to update the dashboard following every change</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>modify_type</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>Include in the filter options 'Ranking' and 'Starred', in addition to 'List' and 'Text'.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>set_defaults</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>Include in the dashboard menu the option to</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Dashboards</strong></td>
<td>Admin</td>
<td>Data Admin</td>
<td>Data Designer</td>
<td>Viewer</td>
</tr>
<tr>
<td>----------------</td>
<td>-------</td>
<td>------------</td>
<td>--------------</td>
<td>--------</td>
</tr>
<tr>
<td>set the dashboard filters as the default filters for the dashboard.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>advanced Include the Advanced menu option for creating custom filters.</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>use_starred Include the option to use filters that have been starred</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Widgets</strong></th>
<th>Widgets</th>
<th>Admin</th>
<th>Data Admin</th>
<th>Data Designer</th>
<th>Viewer</th>
</tr>
</thead>
<tbody>
<tr>
<td>create Create a new widget.</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>Never</td>
</tr>
<tr>
<td>delete Delete a widget.</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>rename Rename a widget.</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>duplicate Duplicate a widget.</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>Never</td>
</tr>
<tr>
<td>copy_to_dashboard</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>Widgets</td>
<td>Admin</td>
<td>Data Admin</td>
<td>Data Designer</td>
<td>Viewer</td>
<td></td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------</td>
<td>------------</td>
<td>---------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>Copy a widget to a dashboard. This</td>
<td>e</td>
<td>e</td>
<td></td>
<td>e</td>
<td></td>
</tr>
<tr>
<td>functionality allows you to drag and drop</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>widgets to additional dashboards in your</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Navigation Pane.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>edit_script</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>Never</td>
<td></td>
</tr>
<tr>
<td>Edit a widget using the JavaScript API.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>change_type</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>Change the chart type.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>export_csv</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td></td>
</tr>
<tr>
<td>Export a widget's data to a CSV file.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>export_png</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td></td>
</tr>
<tr>
<td>Download a widget as an image in the PNG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>format. The downloaded image represents</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>the current view.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>modify_selection_mode</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>Enable the option to change the filter</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>behavior between 'Slice' (only filtered</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>items are included in the chart) and 'Highlight'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(filters items are highlighted).</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>drill_to_anywhere</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td></td>
</tr>
<tr>
<td>Enable the option to drill down into a filed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>to get an in-depth view of a selected value.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Widgets</td>
<td>Admin</td>
<td>Data Admin</td>
<td>Data Designer</td>
<td>Viewer</td>
<td></td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-------</td>
<td>------------</td>
<td>--------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>False, the user can only drill down with predefined drill hierarchies (if defined).</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td></td>
</tr>
<tr>
<td>add_to_pulse</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td></td>
</tr>
<tr>
<td>Include the option to add a widget to Pulse if the widget type is supported.</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td></td>
</tr>
<tr>
<td><strong>Items</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reorder</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>Reorder the fields and values in the data panel and their representation in the chart.</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td></td>
</tr>
<tr>
<td>modify_type</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>Never</td>
<td></td>
</tr>
<tr>
<td>Enables the option to edit values and categories in the data panel.</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td></td>
</tr>
<tr>
<td>on_off</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td></td>
</tr>
<tr>
<td>Enable the option to turn fields on or off in the widget.</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td></td>
</tr>
<tr>
<td>select_hierarchies</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>Include the option to select pre-defined hierarchies.</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td></td>
</tr>
<tr>
<td><strong>Filters</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>on_off</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>Switch filters on or off.</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td></td>
</tr>
<tr>
<td>toggle_expansion</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td></td>
</tr>
<tr>
<td>Widgets</td>
<td>Admin</td>
<td>Data Admin</td>
<td>Data Designer</td>
<td>Designer</td>
<td>Viewer</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>-------</td>
<td>------------</td>
<td>---------------</td>
<td>----------</td>
<td>--------</td>
</tr>
<tr>
<td>Enable the expansion of filter settings to display more filter parameters.</td>
<td>e</td>
<td>e</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>modify</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>modify_type</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>Include in the filter options 'Ranking' and 'Starred', in addition to 'List' and 'Text'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>use_starred</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>Enables the option to star (bookmark as favorite) a filter selection.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ElastiCubes**

This object is not displayed in the Sisense REST API, however, you can still toggle these features through the `POST /roles/{idOrName}/manifest/{path}` endpoint. The value of path should be in the format of `manage/elasticubes/{value}`, where `value` is the feature listed in the table below.

<table>
<thead>
<tr>
<th>ElastiCubes</th>
<th>Admin</th>
<th>Data Admin</th>
<th>Data Designer</th>
<th>Designer</th>
<th>Viewer</th>
</tr>
</thead>
<tbody>
<tr>
<td>idOrName</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>string * required</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(path)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>path * required</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(path)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For more information, refer to the documentation available at: [ElastiCubes](#).
<table>
<thead>
<tr>
<th>ElastiCubes</th>
<th>Admin</th>
<th>Data Admin</th>
<th>Data Designer</th>
<th>Designer</th>
<th>Viewer</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>Allows the user to add new ElastiCubes.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>schedule_build</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>Allows the user to schedule builds for an ElastiCube.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>modify_data_security_rights</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>Allows editing the row-based data security permissions.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Changing Permissions

To change permissions for a user role:
In the Sisense Web Application, click Admin at the top-right of the screen, and then REST API in the left menu.
Click REST API Reference to view the API documentation.
In the API Documentation, select version 0.9 on the top-right of the screen.
Click roles to access the different API operations related to roles.
Open the operation that you require (see "Examples" on page 854 below), and then click Test It Out.

In the list of fields that opens, enter or select the required parameters.

Note: The API uses a previous terminology for Designers and Viewers. When entering the role name into the operation parameters, enter 'contributor' for the Designer role, and 'consumer' for the Viewer role.
When changing a permission for a role, you must use the Path parameter, which defines the relative path to the permission you want to change in the hierarchy of permissions. To view the current hierarchy of permissions, scroll down to the Example Value in the Responses section. For example, to change the 'Create' setting for dashboard filters, enter 'dashboards/filters/'. To create dashboards, enter 'dashboards/'.

When changing a permission for a role, go right into the list of permissions in the Manifest parameter, and type in the desired permission (for example, change 'true' to 'false').

Then remove everything else from the list of permissions, including the
heading ("dashboards"), the brackets and the extra commas.

Click **Execute** to apply the changes.
Scroll down to the Responses section to verify that the response has no errors.

Once you have no errors, your change is applied to all users assigned the relevant role.
Examples

Here are some examples of what you can do:

| See all permissions for all roles | Method: **Get /roles**  
Parameters  
**includeManifest** – Use the default setting (true)  
**compiledRoles** – Use the default setting (true)  
Example API Call  
/api/roles?includeManifest=true&compiledRoles=true |
| See all settings for a specific role | Method: **GET /roles/{idOrName}**  
Parameters  
**idOrName** – The role name can be one of the following: admin, contributor, consumer  
**compiledRole** – Use the default setting (true)  
Example API Call  
/api/roles/consumer?compiledRole=true |
| Change a specific setting for a specific role | Method: **PUT /roles/{idOrName}/manifest/{path}**  
Parameters  
**idOrName** – The role name can be one of the following: admin, contributor, consumer.  
**Path** – Enter the path to the setting that you want to change as it appears in the hierarchy. Do not include the setting in the path. Example: To change the ‘Create’ setting for dashboard filters, enter dashboards/filters/. For creating dashboards, enter /dashboards. See the reference table above for more settings.  
**Manifest** – Defines whether to enable or disable the setting. For example, to enable ‘create’, type in: {”create”:true} |
Example API Call
/api/roles/viewer/manifest/dashboards/filters

**Note:** You can use POST to do the same action, but for POST all other options will be reset to their default settings.

<table>
<thead>
<tr>
<th>Restore a specific setting for a specific role</th>
<th>Method: DELETE /roles/{idOrName}/manifest/{path}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td></td>
</tr>
<tr>
<td><strong>idOrName</strong> – The role name can be one of the following: super, dataDesigner, dataAdmin, admin, contributor, consumer.</td>
<td></td>
</tr>
<tr>
<td><strong>Path</strong> – Enter the full path of the setting you want to change, as it appears in the hierarchy. Example: dashboards/filters/modify_type. See the reference table above for more settings.</td>
<td></td>
</tr>
</tbody>
</table>

Example API Call
/api/roles/viewer/manifest/dashboards/filters/modify_type

<table>
<thead>
<tr>
<th>Restore all settings for a specific role</th>
<th>Method: DELETE /roles/{idOrName}/manifest/{path}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td></td>
</tr>
<tr>
<td><strong>idOrName</strong> – The role name can be one of the following: super, dataDesigner, dataAdmin, admin, contributor, consumer.</td>
<td></td>
</tr>
<tr>
<td><strong>Path</strong> – / Enter just a slash to restore all settings.</td>
<td></td>
</tr>
</tbody>
</table>

Example API Call
/api/roles/viewer/manifest/
Integrating Active Directory

Sisense supports integrating Active Directory users and groups with your current Sisense users, so that you can share dashboards and email reports with any of your users. This works in parallel with Sisense’s existing User Management, so you can continue to add users and user groups that are not part of the Active Directory domain.

To enable Sisense to identify your Active Directory users, you import your Active Directory users and groups into Sisense. When you import an Active Directory user, Sisense automatically creates a Sisense user. When you import an Active Directory group, Sisense does not automatically create a Sisense user for each member. The first time a member of an Active Directory group logs into Sisense, then Sisense automatically creates a Sisense user. Active Directory users that do not log in, are not assigned a Sisense user until their first login.

For secure connections to Active Directory, Sisense supports LDAP-S (LDAP over SSL), which encrypts communication between Sisense and your Active Directory. If your Active Directory supports SSL, you can upload your PFX certificate and PFX password for each Active Directory domain while defining your Active Directory settings in Sisense. PFX files contain the public key file (SSL certificate file) and the associated private key file in a single file. These files are generated and provided by you.

This topic describes how to configure Active Directory in Sisense, and how to import users and user groups from Active Directory.
Adding an Active Directory Domain to Sisense

From the Active Directory page, you can add Active Directory domains to your Sisense configuration. Each Active Directory domain you add to your Sisense configuration is added to the Active Directory list displayed in the Active Directory page. The first time you open the Active Directory page, the Add Domain link is displayed, which you can click to begin adding Active Directory domains.

After you have added your first Active Directory domain, it is displayed in the Active Directory list.

From the Active Directory list, you can add additional Active Directory domains. Each Active Directory domain you add to Sisense enables you to share your dashboards and reports with users from any of the domains as well as Sisense users not part an Active Directory.

You can filter the list of the Active Directory domains by entering the name of the relevant domain in the Search field.

To add an Active Directory:

Click Admin and select the Active Directory tab on the left.
In Active Directory, click Add Domain.
In the Add Domain window, fill in the Active Directory configuration fields to point to your Active Directory instance:

Name: The name of your Active Directory domain. This name is used to locate your Active Directory domain in Sisense when you have multiple domains.
Connection String: The full path to your Active Directory domain. Example: ldap://dc.domain.com (Or ldaps://dc.domain.com for secure connections)
Base DN: The Active Directory location from where a server searches for users. Enter the domain by its distinguished name (DN) with its
domain components (DC) connected by commas. For example: For the domain domain.sisense.test, use
DC=domain,DC=sisense,DC=test.

**Note:** If you have a hierarchy of user groups in your organization, you can define the hierarchy in your string as follows:

If your domain is domain.sisense.com/OU1/OU2, then use **OU=OU2, OU=OU1,DC=domain,DC=sisense,DC=com**. Only users and user groups from OU2 will be available. Users/groups under OU1 will not be available.

**Username:** Enter the username with domain. Example: domain\username

**Password:** Enter the password for your Active Directory instance.

**Sync data every:** Enter how often Sisense automatically synchronizes with the users and groups in this Active Directory domain in milliseconds.

**SSL Enabled:** Select to enable a secure connection to your Active Directory. When you select **SSL Enabled**, the **PFX Upload** area and **PFX Password** field are displayed. In the **PFX Upload** area, drag the PFX file into the browser or click **Browse** and navigate to the file. In **PFX Password**, enter the password you received after your PFX was generated.

Click **Test** to verify that your configuration successfully connects to your Active Directory domain.

Click **Save**.
Editing Active Directory Domain Settings

To edit an Active Directory domain:

From the Active Directory list, select 🗂️ for the Active Directory settings you want to edit. The Edit Active Directory window is displayed.

In the Edit Active Directory window, edit the relevant fields. See **Adding an Active Directory in Sisense** for more information about each field.

Click **Save**.
Disabling an Active Directory Domain

Supports allows you to disable Active Directory domains in Sisense. An Active Directory domain that has been disabled is still displayed in the Active Directory list, however, users from this domain will not be able to log in to Sisense, and you will not be able to share dashboards or reports with them. Disabled Active Directory domains have the status Disabled in the Status column in the Active Directory list.

**To disable an Active Directory domain:**
- From the Active Directory list, select `≡ > Disable` for the Active Directory domain you want to disable. The Active Directory domain is disabled.
- You can enable the Active Directory domain by selecting `≡ > Enable`. 
Deleting an Active Directory

You can delete an Active Directory domain from your configuration. When you delete an Active Directory domain, the users included in that domain are no longer displayed when sharing dashboards or reports. Deleted Active Directory domains cannot be restored, however you can add the Active Directory domain again from the Active Directory page.

To delete an Active Directory domain:

From the Active Directory list, select for the Active Directory settings you want to delete. The Delete Active Directory window is displayed. Click Delete. The Active Directory domain settings are removed from Sisense.
Adding a User from Active Directory

Once a connection with Active Directory has been established, an additional **Import from External Server** option is added to the **Add Users** button in the User tab.

The user’s username and email will be added from Active Directory.

**Note:** You can add a user from Active Directory, even if you have not imported that user’s user group from Active Directory.

See [Managing Dashboard Users](#) for general information about working with users/user groups in Sisense.
Associating a Sisense User with an Active Directory User

Once a connection with Active Directory has been established, you can associate Sisense users to your Active Directory users in the User tab. Associating a Sisense user with an Active Directory allows you to manage your Sisense users from Active Directory. For example, after associating your users, if you need to delete a user, you can remove them from Active Directory and they will no longer be able to access Sisense. Another example is if you have an Active Directory user and a Sisense user with their own dashboards or ElastiCubes, you can associate the accounts so you don’t need to create a new Active Directory user for your Sisense account.

To associate a Sisense user:

From the **Users** tab of the **Admin** page, click 🆕️ to edit the relevant Sisense user.

If Active Directory is not enabled, toggle the switch to enable it.

Select the relevant Active Directory domain and locate your Active Directory user.

An auto-complete list is displayed showing the Active Directory users. Select a username to add to the group.
Click **Save** to associate the accounts.
Adding a User Group from Active Directory

Once a connection with Active Directory has been established, an additional Import from Active Directory option is added to the Add Group button in the Groups tab.

The following procedure describes how to import a list of users from a group in Active Directory to Sisense.

Users that are added in this way only appear in the Sisense users list after each user logs in to Sisense. In this way, your Sisense license slots are only consumed by users who actually log in to Sisense.

**To add a user group from Active Directory:**

Click Admin and select the Groups tab on the left.

Click the Add Group button and then select the From Active Directory option from the menu.

In the Search Active Directory Group field start typing in the name of the desired Active Directory Group. An auto-complete dropdown menu is displayed from which you can select a user group.

If an Active Directory user group has already been imported into Sisense, the words **Group already selected** appear next to it, as shown above. You can add multiple Active Directory groups, by selecting one group after another. The groups appear in this window as follows:

Select the default Role for the members of this user group: Viewer or Designer. It will be assigned to new users from this group, when the users are created. Subsequently, user’s role can be changed in the Sisense Users list.
Changes to a group’s default role will affect users created after that time, but will not affect existing users. Click Add.
Account Lockout Thresholds

You can prevent brute-force account attacks in Sisense by limiting the number of failed login attempts that can be performed before an account is locked. In the Settings tab of the Admin page, you can set your account lockout thresholds for your users. Account lockout thresholds determine after how many failed login attempts a user is locked out of Sisense and for how long. A locked account cannot be used until you unlock it or until the number of minutes specified by the Account lockout duration policy setting expires.
Setting Account Lockout Thresholds

To set your account lockout thresholds:

In the Admin page, click Settings. The Account Lockout settings are displayed on the bottom of your Settings.

By default, Account Lockout is enabled. If it is not, toggle the Account Lockout switch to turn on account lockout thresholds.

In **Number of failed login attempts before lockout**, enter how many attempts a user may fail before they are locked out.

In **Lockout duration (minutes)**, enter the amount of time, in minutes, that a user is locked out for after exceeding the amount of failed login attempts.

(Optional) You can define the period of time where if the user exceeds the amount of failed login attempts, they are locked out through the Sisense REST API. See [Managing Account Lockout Settings](#) via the REST API for more information.

Click **Save**.
Unlocking Accounts

To unlock a locked user account:

In the Admin page, click Users. For the relevant locked user, click 🛠️. The user's account settings are displayed.

Click Account Locked to unlock the user. Click Save. The user can now log in.
Managing Account Lockout Settings via the REST API

You can manage your Account Lockout thresholds through the REST API. Sisense provides two endpoints:

GET /settings/login_lockout

PATCH /settings/login_lockout

The GET endpoint returns your current threshold settings while PATCH enables you to update these settings.

The table below describes the parameters you can return and update through the login_lockout endpoints.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enabled</td>
<td>boolean</td>
<td>Determines if the Account Login threshold is active. The default value is true.</td>
</tr>
<tr>
<td>maxFailedAttempts</td>
<td>number</td>
<td>How many failed login attempts must occur before a user is locked out.</td>
</tr>
<tr>
<td>failedAttemptsPeriod</td>
<td>number</td>
<td>The period of time, in minutes, where if the user exceeds the amount of failed login attempts, they are locked out.</td>
</tr>
<tr>
<td>lockoutPeriod</td>
<td>number</td>
<td>The amount of time in minutes that a user is locked out for after exceeding the allowed amount of login attempts.</td>
</tr>
</tbody>
</table>

To access the login_lockout endpoints:
Access the Sisense REST API.

In version 1.0 of the REST API, the login_lockout endpoints are located under /Settings.
Introduction to SSO

Single Sign-On (SSO) is a mechanism that allows a system to authenticate users and subsequently tell Sisense that the user has been authenticated. The user is then allowed to access Sisense without being prompted to enter separate login credentials.

The SSO security mechanism allows Sisense to trust the login requests it gets from your corporate authentication system, and grant access to the users that have been authenticated by it. An SSO session begins when the authenticated user requests a secured resource from Sisense while logged into your site or application. The user’s browser sends an HTTP request to Sisense that includes a cookie which contains session and authentication information. This information is then used for session validation.

Users who already have Sisense accounts can continue to access Sisense through the Sisense Login page with their current accounts. To prevent users from directly logging in to Sisense instead of your login page, your Sisense administrator can change the passwords of your current users forcing them to log in with your company's credentials in your company’s login page.

Sisense recommends that administrators always keep a Sisense password, so that the administrator can access Sisense in case the SSO server is not available. Sisense SSO supports two SSO protocols for securing the exchange of user authentication data, JSON Web Token (JWT) and SSO via SAML 2.0 (Security Assertion Markup Language). For more information, see SSO via JWT or SSO via SAML 2.0.
SSO via JWT

JWT is a technique that can be used for single sign-on (SSO) between your site and Sisense. JWT is a token that represents your users credentials wrapped in a single query string. In addition, Sisense uses the jti parameter (see below), which adds a unique ID to the token that prevents the token from being used more than once, thus preventing attacks on the system (also known as replay attacks).

The Sisense SSO via JWT authentication flow is explained in the following diagram.

Note: See the Developers Community for a tutorial that shows how to implement SSO via JWT.
SSO Authentication Flow

The following is a diagram of the SSO authentication flow from your site or application to Sisense.

Your user requests a resource from Sisense, typically a dashboard. Sisense recognizes that no authenticated cookie is present. If you have enabled SSO in Sisense, the SSO handler redirects the user to your Remote Login URL defined in the Sisense Web Application.

![Diagram of SSO Authentication Flow]

Your user is challenged to authenticate their account. Your Remote Login application authenticates your user and generates a JWT (JSON Web Token). You redirect the user back to Sisense with the encoded JWT in a query string. Sisense sets a cookie that authenticates the user’s session until they end it.
or you log them out via the Sisense REST API. For more information see [Logging Users Out](#).

Sisense provides the authenticated user with the request resource. A common scenario that illustrates SSO is when an unauthenticated user navigates to your site in which Sisense is embedded via an iFrame. Sisense redirects this user to your SSO script. Your script authenticates the user through your login process and builds a JWT request with all the relevant credentials wrapped together. You then redirect the customer back to Sisense with the JWT payload. Sisense then decodes the user details from the JWT payload and then grants the user a session.
Configuring SSO in Sisense

While SSO is highly customizable, there are generally four steps you should complete when configuring SSO:

**Note:** Configuring SSO requires technical expertise and should be conducted by an administrator or developer with SSO experience.

- **Enabling SSO in Sisense:** Through the Sisense Web Application, an administrator can enable SSO in Sisense and define the relevant Login and Logout URLs.
- **Creating a JWT:** After you authenticate a user, you generate a JWT with the user’s credentials to Sisense, so Sisense knows this user is allowed to access resources from Sisense through your site.
- **Logging Users Out:** A user can access Sisense so long as a session is maintained. To end a session, the user’s cookie that Sisense provides must be deleted. To delete this cookie, you can use the Sisense REST API.
Enabling SSO in Sisense

For Sisense to recognize that your users should be authenticated through SSO, you must enable SSO in the Sisense Web Application. In the SSO menu of the Admin page of the Sisense Web Application, you define the URL where Sisense redirects users to authenticate on your side and where Sisense redirects users after they log out from Sisense.

When you access the SSO menu of the Admin page, Sisense displays the Shared Secret key. The Shared Key is a JWT encryption public key used to encrypt theJWT payload. It is generated once when the SSO configuration is saved. You include this key in the JWT payload when redirecting the user back to Sisense after authenticating them on your side.

To access and set up SSO:

Log into Sisense.
Select Admin screen and click Single Sign On in the left menu.
Fill in the following SSO configuration fields:

**Remote Login URL:** This is the URL that Sisense will invoke to attempt remote authentication. In that endpoint the participating application user authentication script is triggered and the JWT payload is generated.

**Remote Logout URL:** This is the URL that users will be redirected to after they log out from Sisense (i.e. the participating application’s home page).

Click Save.
Creating a JWT

Your script builds a JWT request that contains the user data. The table below provides a list and descriptions of the attributes your JWT should contain.

In addition, several samples are provided below in various languages.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>typ</td>
<td>Yes</td>
<td>Define the type of token with the attribute typ. You should also specify HS256 as the JWT algorithm in the header of your JWT payload.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;typ&quot;:&quot;JWT&quot;,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;alg&quot;:&quot;HS256&quot; }</td>
</tr>
<tr>
<td>iat</td>
<td>Yes</td>
<td>Issued at the time the token was generated. This is used to help ensure that a given token gets used shortly after it is generated. The value must be the number of seconds since UNIX epoch. Sisense allows up to five minutes clock skew. <strong>Note:</strong> The date must be an integer and not a float.</td>
</tr>
<tr>
<td>sub</td>
<td>Yes</td>
<td>Email of the user being signed in, used to uniquely identify the user in Sisense. If the user does not exist in Sisense, it will be created with default viewer privileges.</td>
</tr>
<tr>
<td>jti</td>
<td>Yes*</td>
<td>A unique string added to the token that is used to prevent replay attacks, by making sure the token</td>
</tr>
<tr>
<td>Attribute</td>
<td>Mandatory</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>is used only once.</td>
</tr>
</tbody>
</table>

* You can set this attribute as optional in the Sisense REST API v1.0 through the POST settings/SSO endpoint.

SSO Code Samples

- [C#](#)
- [Java](#)
- [Javascript](#)
- [PHP](#)
- [Ruby](#)
- [Python](#)
return_to URL

After your users successfully authenticate, Sisense returns them to a URL defined as the return_to URL.

For example:
https://yourcompany.sisense.com/dashboards/

To define the return_to URL in Sisense:
   In your browser, open the Configuration Manager located at http://localhost:3030.
   In SSO Return to, enter your base URL.
   Click Save to save your changes.
**Logging Users Out**

When a user is logged in, anyone using that browser can access the session, or users may encounter an issue where they remain logged in until the Sisense cookie is cleared.

Users are logged out when the session ends. A session ends when the user closes their browser or according to the value of the attribute exp you send in the JWT payload.

You can log the user out through the Sisense REST API.

To manually log a user out, access version .9 of the REST API. Through the Auth method, you can issue a get request to log out specific users.

While the logout REST API can delete the SSO authentication cookie, it can only delete it when the call is made from within the Sisense domain. Scripts on different pages can access each other only if the pages that executed them are at locations with the same protocol.

If you have embedded Sisense in an iFrame and you want to log out the user from your application and Sisense, you can use the window.postMessage method to call the logout when the users asks to logout from your application. This method overcomes any cross-origin communication limitations. Sisense has created a
plugin that implements a listener, which calls the Logout API when the 
postMessage method() is called. For more information, click [here](#).
SSO via SAML 2.0

The Sisense SAML authentication process is based on the SAML 2.0 protocol, and is explained in the following diagrams:

**Sisense SAML auth workflow (1st phase)**

1. Sisense resource required
2. Save-hash page
3. Resource hash
4. Redirect to IdP Server command
5.
Phase 1

The first phase of this process begins when a user requests a resource from Sisense via their browser (1). The browser generates a resource request to the Sisense server. The server processes this request (2), and for unauthenticated users, returns a special save-hash page to save the requested hash data. After the browser loads the save-hash page from the Sisense server, it runs a script, which loads the requested resource hash from the URL and sends it to the Sisense Server (3). At the end of this phase, the Sisense Server converts the requested resource URL and sends it to the browser as a part of the redirect to IdP Server command (4).
Phase 2

The second phase of the authentication process starts after the browser retrieves the redirect command from the Sisense Server, and sends the authentication request to the IdP Server (1) as described in the diagram below.

The base URL for this request is taken from the Remote Login URL field in the Single Sign On section of the Admin page in the Sisense Web Application. The requested resource address is passed to this request as a RelayState parameter. All other data is provided as a SAMLRequest parameter. The next step of this phase depends on the user authentication state and the IdP implementation.

If the current user isn’t logged in as an IdP user, the IdP server redirects the browser to your Login page (2), where the user enters their IdP credentials. After logging in, the IdP Server sends the browser the Auto-Sign-In page with encoded data about the currently logged-in user (4).

**Note:** Two-factor authentication for Sisense is supported for SSO providers that support two-factor authentication.
After this document is loaded in the user’s browser, it runs a script which creates the POST-query to the Sisense Server API, and passes the SAML response to this query (1). At this time, the third phase of the authentication process begins.
Phase 3

The Sisense Server handles the POST-query (2), decrypts it with the certificate specified in the Public X.509 Certificate field defined in the Single Sign On section of the Admin page. The Sisense Server uses the decrypted User ID as a key to locate the Sisense user in the internal database.

If a user is not found in the system, Sisense creates a new user (3).
Associating SSO Users with Sisense Groups

When Sisense creates a new user, Sisense analyzes the “memberOf” field to locate one or more groups related to the logged-in user.

If the “memberOf” field is empty, the user is assigned the role: “Viewer”. If the “memberOf” field contains one or more groups, and the groups were previously defined in Sisense, the newly created users’ Role is taken from the groups’ default role. When multiple Sisense Groups are found, the user is assigned the role with the maximum privileges.

After a user is created in the system, an administrator can modify the user role, if needed.

Below, is an example of an SAML XML where the “Test” group is specified:

```xml
<ns0:Attribute
   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic"
   Name="memberOf">
   <saml:AttributeValue
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:type="xs:string">
     Test
   </saml:AttributeValue>
</ns0:Attribute>
```

After the user is located or created, a new session is initialized, and the Sisense Server redirects the user browser to the requested resource (4).
Enabling SSO via SAML in Sisense

After you have configured your SAML server, sign in to Sisense as an Administrator and follow the instructions below.

**To enable SAML in Sisense:**

In the Sisense Web Application, click **Admin** and select **Single Sign On**.

In the Single Sign On page, select **SAML 2.0**.

In the **Remote Login URL** field, enter the SAML Login endpoint. Sisense redirects the user to this field when they sign in. This value should be provided by the IdP Service.

In the **Remote Logout URL** field, enter the SAML Logout endpoint. Sisense redirects the user to this field when they sign out. This value should be provided by the IdP Service.

In the **Public X.509 Certificate** field, enter your public key for your SAML configuration. This value should be provided by the IdP Service.

Click **Save**. SSO via SAML 2.0 is configured.
Sisense Default Role Set-Up

When an authenticated user is not found in the Sisense database, a new account is created. The user role is specified based on the user group/groups default role.

To define a group’s role:
In the Sisense Web Application, click Admin and select Groups.

Click Add Group. The Create a New Group window is displayed.
In the **Create a New Group** window, select the default role of the group.

Click **Save**.

**Note:** Changes to the group’s default role are applied when new users are created, and do not affect existing users. After a user is created in the system, an administrator can modify the user role, if needed.
Related Topics

For instructions on setting up Sisense with some 3rd-party providers, see:

- Setting Up SSO SAML 2.0 with Okta
- Setting Up SSO SAML 2.0 with OneLogin
- Setting Up SSO SAML 2.0 with Auth0
- Setting Up SSO SAML 2.0 With Salesforce
- Setting Up SSO SAML 2.0 With ADFS
- Setting Up SSO SAML 2.0 With G Suite
Managing Your Users
Dashboards

As a Sisense Administrator, you can manage and interact with all dashboards in the system, including your dashboards and dashboards of other users across your system from the Analytics page.

For Administrators, the Analytics page is a centralized location for managing all the dashboards in your system and performing administrative actions on them.

As an Administrator, you can perform the following operations:
- Sharing the dashboard with other users in the system, or change the dashboard’s owner
- Accessing a user’s dashboard as a Viewer
- Copying the dashboard to another server
- Downloading a PDF of the dashboard
- Deleting the dashboard

This is in addition to the functionality available to you for your own dashboards. When you click the All Dashboards button on the Analytics page, you can see all your dashboards and the dashboards of your users. A yellow border displayed on top of the dashboard tile indicates that you are looking at the dashboard version of one of your users, and not your copy of the dashboard. This can be
useful to view a dashboard that the owner modified, but has not yet republished the changes.

You can click the dashboard tile to open the dashboard. When opening a dashboard not shared with you, you will see the dashboard as a Viewer without editing rights. If you need to modify the dashboard, you must first make yourself the owner of the dashboard. You will then have full edit permissions for the dashboard.

The menu on the right of the dashboard tile is from where you can manage your or your users’ dashboards. The example below includes the options available to you when the dashboard belongs to another user.

Keep in mind that the All Dashboards button is displayed only if the grid-view area of the Analytics page is available to you. If you have rebranded the whole Analytics landing page, the All Dashboards button will not be displayed. If you rebrand just the top part of the Analytics landing page, the All Dashboards button will be available. For Administrators, if you have enabled the Dashboard Admin feature, the default Sisense Analytics page is displayed while all non-admin users continue to see the rebranded page.

You can rebrand the top part of the Analytics and Data pages by defining the URL of the marketingPart key, which is part of the POST /branding endpoint. The
marketingPart key is an iFrame that displays the Welcome banner and Sample dashboards and ElastiCubes.

**Note:** This feature can be disabled over REST API, if you are not interested in exposing the system dashboards to Administrators. See the isAdminAccessAllDashboards key of the `POST /settings/system` endpoint for more information.

For information on improving dashboard performance for your users, see [Maximize Dashboard Performance](#).
Data Management

The topics below describe how you can share data models with other users and you can create drill hierarchies that you want to make available for Designers:

- Adding and Removing ElastiCube Servers
- Sharing ElastiCubes
- Managing Drill Hierarchies
Adding and Removing ElastiCube Servers

This topic describes how to manage your ElastiCubes.
By default, the Sisense Web Application is connected to the ElastiCube server that is installed with it on the same machine.
By default, all ElastiCubes on that machine are accessible to all dashboard users, and are visible under LOCALHOST.
Opening an ElastiCube in Sisense

You have a number of ways to open your ElastiCube in Sisense:

From the Sisense Web Application:
If you have an open dashboard, click on the ElastiCube name next to the dashboard name. Select View in ElastiCube's Page to open the ElastiCube management page in the web app. In the ElastiCubes page, select the ElastiCube and click Open. To open the ElastiCube of the open dashboard, click Open Sisense.

From Windows: From the Start menu, open the ElastiCube Manager.
Adding Another ElastiCube Server

To add another ElastiCube Server to Sisense:

Click **Admin** and select the **Data Sources** tab on the left. Click **Add Server**.

Select whether this ElastiCube Server is intended for Admins only. This selection also applies to all new ElastiCubes that will be added to this ElastiCube Server. A user that has access to an ElastiCube Server can access all the ElastiCubes on it.

Add the IP of the ElastiCube Server to which you want to connect and click **Add Server**. The new ElastiCube Server is added to the list with all its ElastiCubes.

Click **Save**. Once a new ElastiCube Server has been added, its ElastiCubes immediately become available to all designers (by default) when they create a new Dashboard.

To define which users are allowed to access the data of an ElastiCube Server, see **Assigning Rights to an ElastiCube Server**.

To define which users are allowed to access each specific ElastiCube, see **Assigning Rights to an ElastiCube**.
Removing an ElastiCube Server

To remove an ElastiCube server from the list:

In the Data Sources page, for the relevant server, click the Trash button.

**Note:** The LOCAL MACHINE server cannot be removed.
Sharing ElastiCube Models

You can share ElastiCube models with other Sisense users through the Admin page.

There are two types of access rights that can be assigned to a Sisense user, edit rights and query rights. Edit rights allow the user to modify the ElastiCube model, while query rights allow the user to query or create a dashboard based on the shared model, but not modify it.

The user's role determines what access they have and what access can be assigned to them when the ElastiCube model is shared.

Administrators and Data Administrators have edit rights to the ElastiCube model, even without sharing the ElastiCube model with them. Data Designers can have edit or query rights, while Designers can only have query rights.

When an ElastiCube model is created on the Data page, it is not shared with anyone. This allows the model's creator to work on the model and publish it only once it's ready.

When you assign specific users/user groups access rights to this ElastiCube, then only the users/user groups that you assign can access the ElastiCube.

When an ElastiCube is shared with other users, the ElastiCube is displayed in the Data page with a list of the other shared users.
To share an ElastiCube:
In the Sisense, select **Admin > Data Sources** and then the menu icon for the relevant ElastiCube.

Click **Share**.

Enter the names or groups of users with whom to share the ElastiCube. Selecting **Everyone** specifies that this ElastiCube is shared with all Sisense users and user groups.
The contact information of the people you add is listed in the center of the window and is displayed the next time you open this window. In the dropdown menu next to each user/user group select what rights the user has. (You can only select the rights for a Data Designer as all other roles have predefined permissions.) Click **Save**.
Managing Drill Hierarchies

Dashboard viewers can drill down in most widgets to get an in-depth view of a selected value. To make it easier for viewers to select drill-down paths, Administrators can add predefined drill hierarchies to their widgets. As an Administrator, you can create a pool of drill hierarchies, from which you can select the drill hierarchies they want to include in the widgets they build. This section describes how to create and manage the drill hierarchies that you want to make available for designers.

To create a drill-down hierarchy:
Click Admin in the Sisense Web Application. In the left menu, select Data Sources. For the relevant ElastiCube, click and select Manage Hierarchies.
If this is the first hierarchy you are adding, select Add Hierarchy. If you already have hierarchies, click Add Hierarchy in the top-right corner. Select the field. This will be the first field in the drill hierarchy, and the same field used in the widget.
Click Add Field to add the next level in the drill hierarchy. Repeat this step for as many levels as you need.

Additional options include:
**Always include with field:** Select this checkbox to always include the drill hierarchy in the widget for this field.
**Editing the Hierarchy name:** To edit the name of the drill hierarchy, click on the pencil icon, and type in the name. Click on the tick icon to confirm the change.

**Editing the field name:** To edit the name of a field in the hierarchy, hover over the field name and click on the menu > Rename. Type in the name, and click on the tick icon to confirm the change.

Add as many hierarchies as you need for the ElastiCube. When done, dashboard designers will have access to the hierarchies in the widget.
designer, and will be able to select and add them to the widget.

To change a field in a hierarchy: Hover over the field in the hierarchy, and click on the menu that appears. Click **Change** and select a different field.

To delete a hierarchy: Click on the bin to the right of the hierarchy.
System Settings

Sisense Administrators can perform a variety of tasks related to implementing and maintaining an implementation of Sisense.

The following topics describe system-related tasks that Sisense Administrators are responsible for handling:

- System Configuration
- Internationalization and Localization
- Managing Plug-ins
- Setting Up a Custom Email Server
- Sisense Narratives
Changing the Sisense Web Application’s Language

Sisense supports multiple languages in the Sisense Web Application making it easier to extract meaningful insights from your data in your users’ local language. The following images display two interfaces, one in English and the other in Spanish.
<table>
<thead>
<tr>
<th><strong>English</strong></th>
<th><strong>Spanish</strong></th>
</tr>
</thead>
</table>

What has been Translated?

Sisense has translated text that your users might see including menu names, buttons, messages, and other elements in the Sisense Web Application. Translating the Sisense Web Application does not affect your data or how it is displayed.

The Sisense Web Application has been localized to the following languages:
- Chinese (Simple)
- French (France)
- German
- Italian
- Japanese
- Portuguese (BR)
- Russian
- Spanish (ES)
- Spanish (LA)
- Dutch

Note, the following content is not being localized:
- Built-in Javascript editor
- ElastiCube Manager and Server Console
- Sisense documentation and online help
- REST API and documentation
- Sisense automated emails

**Note:** You can use your own translation of Sisense automated emails by defining your own. Click [here](#) for more information.
Changing Sisense’s Default Language

Administrators can select the default language for the Sisense Web Application. Setting the default language affects all users and groups who access the Sisense dashboard. Additionally, Administrators can then set the language for specific groups and users when adding new users or by editing a group or user account. After selecting a new language, each string with a translation is displayed in that language across all your Sisense users’ interfaces.

If you have defined a custom translation, the name of the language as defined in the settings.json is displayed in this list. See Embedding Custom Languages below for more information.

Sisense’s default language is English. After changing the system language, you can revert back to the English or any other language by selecting it from the list.

To modify your default language:
- In the Sisense Web Application, click Admin in the top menu, and then Settings in the left menu.
- Select the relevant language from the Language list. After selecting the language, the UI is immediately translated.
- Click Save. The Sisense Web Application is translated in the selected language.
Embedding Customized Languages

Sisense enables you to display your own translations in the Sisense Web Application by replacing the values of strings in any language you choose. You can embed customized languages by modifying configuration and resource files replacing the string values according to your requirements.

For example, in the homepage of the Sisense Web Application, the word “Explore” is displayed. In the resource file home.js, you can modify the value of this string to display your own text as shown below.

```
"welcome_user": "Welcome",
"explore": {
  "explore": "TEST",
  "learn": "Learn & find new tips & tricks",
  "links": {
```

In addition to modifying the resource files, you must modify the settings.json file and specify your language's name and update the version. In this example, the language has been changed to “test” and the version increased from 1.0.0 to 1.0.1.

```
{  "displayName": "test",  "version": "1.0.1" }
```
The value of “displayName” is displayed as your language’s name in Default Language list on the System Configuration page. In the example below, the displayed was set to Test.

Upgrading Sisense may replace the bundled translation files. If you wish to modify a bundled language, copy of all of the translation files to a new directory, and change the copied versions to prevent your translation from being overwritten.

To customize the Sisense Web Application’s language:

Navigate to the directory ...\Program Files\Sisense\app\translations\.

In the \translations directory, copy the en-us(default) directory, paste it in the \translation directory and rename it.

In the new directory, open each resource file and translate the strings into the desired language saving each file after your changes are made.

Open the settings.json file and replace the following key values:

- **displayName**: Enter the name of the language.
- **version**: Enter the name of the new version. Currently, the version is 1.0.0. For each version, increase this value, for example, 1.0.1.

Save the changes in a file called settings.json in your new language folder.

In the Sisense Web Application, click Admin in the top menu, and then Settings in the left menu.

Select your customized translation from the Language list. The Sisense Web Application is displayed in your customized translation.

Click **Save**.
Internationalization and Localization

Sisense supports localization and internationalization by displaying your users’ native language in the Sisense Web Application. You can select the language displayed in the Sisense Web Application from a list of languages provided by Sisense or display your own translation. Each time you select a language, the Sisense Web Application interface is displayed in that language.

When you change the default language displayed in the Sisense Web Application, words that are part of the Sisense framework such as menu items, buttons, and messages are displayed in the selected language. Words specific to your dashboard, such as table names, field names, and filters, are not automatically translated when you select a new language to display. These words are called metadata and you can provide your own translation of these terms and display those translations to your users through a Dashboard script.

For instructions on how to change the language displayed in the Sisense Web Application, see [Changing the Sisense Web Application’s Language](#).

For instructions on how to translate your metadata, see [Translating Sisense Metadata](#).
Translating Sisense Metadata

Sisense enables you to translate metadata in your users’ native language. Sisense metadata is displayed in your users’ dashboards and includes the following:

- ElastiCube fields
- ElastiCube tables
- Widget Titles
- Dashboard Names
- Folder Names
- Saved formulas
- Default and custom hierarchies
- Customized renamed field titles

The following images display two interfaces with metadata, one in English and the other in Russian.
As fields such as ElastiCube names and tables are defined by you, these fields are not translated when changing the default language of the Sisense Web Application.

You can translate your metadata by modifying the default values using a JavaScript plugin. Through the plugin, you define the new values to be displayed. You may also include additional logic to determine which users or user groups the translated values should be displayed to.

When you create a JavaScript plugin, you should create a folder for the plugin and place it inside the plugins folder in the following location:

**In Windows:**
```
...\Program Files\Sisense\app\plugins
```

**In Linux:**

See [Uploading Files through the File Manager](#) and upload the files to the **plugins** folder.

OR

Upload all the folders to your Sisense Server through the File Manager. See [Uploading Files through the File Manager](#) for more information.

If the `\plugins\` folder does not exist, create it and then create a new folder within the `\plugins\` folder with the relevant files.

You can download an example of a Metadata Translation plugin [here](#). This is just an example, you define how your metadata is to be translated.

Extract the enclosed folder into the plugins folder.

This file contains a folder called Metadata and two files:

- **plugin.json**: The plugin.json file contains the name of the plugin and a link to the file that describes the plugin’s functionality, in this example, run.js.
- **run.js**: The run.js file contains the plugin’s functionality and describes which metadata is to be translated and how it is to be translated.
This example, and any metadata translation JavaScript file you write should include three parts:

Provider Function: These functions provide the mapping context contained in the Mapping object.

Event Subscription: This event is triggered when a datasource changes, and calls the Provider function.

Mapping Object: Contains key/value pairs with your default and translated metadata.
Provider Functions

The code sample below is an example of two Provider functions, the first is a global provider function for folder and dashboard aliasing. The second is the datasource provider function for translating ElastiCube and formula names. You can replace the parameters in this example to provide the mapping context you define in the Mapping object.

**Global Provider**

```javascript
function (resolve, reject) {
  // an ajax request or any other logic.
  setTimeout(function () {
    resolve(globalAliasing);
  }, 0);
};
```

The following table describes the Provider function and its arguments:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>resolve</td>
<td>A callback function which should be called with the provided aliasing context object or without any parameter if aliases shouldn't be provided for the datasource.</td>
</tr>
<tr>
<td>reject</td>
<td>A callback function to be called in case of error, so the system can properly handle it.</td>
</tr>
</tbody>
</table>

**Datasource Provider**

```javascript
function (datasourceId, resolve, reject) {
  // an ajax request or any other logic.
  setTimeout(function () {
    resolve(datasourceAliasing);
  }, 0);
};
```
The following table describes the Provider function and its arguments:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>datasourceId</td>
<td>An object that represents the ID of the datasource with the following properties: address, database, fullname, ID, title</td>
</tr>
<tr>
<td>resolve</td>
<td>A callback function which should be called with the provided aliasing context object or without any parameter if aliases shouldn't be provided for the datasource.</td>
</tr>
<tr>
<td>reject</td>
<td>A callback function to be called in case of error, so the system can properly handle it.</td>
</tr>
</tbody>
</table>

**Grouping Provider**

The config object allows you to segment translated content per group. For example, Group 1 can see dashboards translated while group two can see the Data page translated. In the config object, you define the groupId and what alias is assigned to that group.

```javascript
var config = [
{
"groupId":"5bbb4d8fd8528537d5604777",
"aliasName":datasourceAliasing
},
{
"groupId":"5bf3c5035f56071a1cf1b37c",
"aliasName":datasourceAliasing_1
}];
```
The following table describes the Provider function and its arguments:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupId</td>
<td>Defines the group that an alias is to be assigned to. You can retrieve the groupid through the GET /groups endpoint in the Sisense REST API.</td>
</tr>
<tr>
<td>aliasname</td>
<td>Defines which alias object to use for the group defined in the groupId.</td>
</tr>
</tbody>
</table>
Event Subscription

To trigger the Provider functions, your plugin should include a handler for a global event ‘beforealiascontextinit’.

In the example below, datasourceProviderFunction and globalAliasingProvider are the Provider functions. The timeout parameter is optional and configures the amount of time in milliseconds which the system waits for the aliasing context to be resolved by the provider function. By default the system uses a timeout of 2000 ms

```javascript
prism.on("beforealiascontextinit", function (ev, args) {
    var timeout = 2000;
    args.register(datasourceProviderFunction,
                  globalAliasingProvider, timeout);
});
```
Mapping Object

The translation context is a simple JavaScript object which contains key value pairs for tables, columns, hierarchies, formulas and custom item titles, all case-insensitive.

The sample below is a skeleton of a Mapping object that defines what metadata is to be translated and its new value.

```javascript
/**
 * tables - aliases for table names
 * "tables": {
 * "table_name_1": "alias_1",
 * "table_name_2": "alias_2",
 * }
 * columns - aliases for column names in scope of table
 * "columns": {
 * "table_name_1": {
 * "column_name_1": "alias_for_column_name_1",
 * "column_name_2": "alias_for_column_name_2",
 * "column_name_3": "alias_for_column_name_3"
 * },
 * "table_name_2": {
 * "column_name_1": "alias_for_column_name_1",
 * "column_name_2": "alias_for_column_name_2",
 * "column_name_3": "alias_for_column_name_3"
 * }
 * }
 * titles - aliases for custom dimension titles, etc
 * "titles": {
 * "Custom title": "Alias",
 * "Sample Revenue": "Sample Profit"
 * },
 * formulas - aliases for custom formula titles
 * "formulas": {
 * "Custom title": "Alias"
 * },
 * hierarchies - aliases for hierarchy titles
* "hierarchies": { 
  * "Custom title": "Alias" 
  * } 
* /

var datasourceAliasing = { 
  "tables": { 
  "brand": "бренд", 
  "category": "категория", 
  "country": "страна", 
  "commerce": "коммерция" 
  },
  "formulas": { 
  "Revenue": "Чистый доход" 
  },
  "hierarchies": { 
  "Category by Brand and Age Range": "Категория через Бренд и Год" 
  },
  "titles": { 
  "CUSTOM REVENUE": "ДОХОДЫ" 
  },
  "columns": { 
  "Brand": { 
  "Brand": "Бренд", 
  "Brand ID": "Бренд Id" 
  },
  "Category": { 
  "Category": "Категория", 
  "Category ID": "Id Категории" 
  },
  "Widgets": { 
  "Revenue": "Чистый доход", 
  },
  "Commerce": { 
  "Age Range": "Возрастные диапазоны", 
  "Brand ID": "Id Бренда", 
  "Category ID": "Id Категории", 
  "Condition": "Состояние", 
  "Cost": "Затраты", 
  "Country ID": "Id Страны" 
  } 
}
"Date": "Дата",
"Gender": "Пол",
"Quantity": "Количество",
"Revenue": "Доходы",
"Visit ID": "Id Посетителя"
},
"Country": {
"Country ID": "Id Страны",
"Country": "Страна"
}
};
/**
 * Global object example
 * folder - aliases for folder names
 * "folder": {
 * "folder_name_1": "alias_1",
 * "folder_name_2": "alias_2",
 * }
 * dashboards- aliases for dashboard names
 * "dashboards": {
 * "dashboard_name_1": "alias_1",
 * "dashboard_name_2": "alias_2",
 * }
 */
var globalAliasing = {
"folders": {
"New Folder": "Folder renamed",
"my folder": "my folder renamed",
"Any name": "любое название"
},
"dashboards": {
"Sample - Ecommerce (1)": "Коммерческий Dashboard",
"Sample - Healthcare": "Sample Healthcare renamed",
"Sample - Healthcare (1)": "Healthcare renamed",
"my dashboard": "my dashboard renamed"
};
System Configuration

The System Configuration settings let you define the web server’s alias, the first day of the week, email settings, and more.

To update one or more System Configuration settings:
Click Admin in the top menu, and then Settings on the left.

Update one or more of the following settings:

**Alias**: Define the web server’s alias or IP address. This is especially important when you have customized your domain name (alias), for example, to prevent example, sharing dashboards from the incorrect IP address. Reports will be sent from the alias entered in this field.

**First Day of the Week**: Select the first day of the week so all of your Sisense users’ queries will be calculated according to the day you define. By default, the first day of the week is Monday when querying an ElastiCube. Sisense defines Monday as the first day according to ISO 8601. This means that when selecting filters such as Week 1, Week 2, etc., the first day of the week is not that defined by first day of that year, but rather the first Monday of the year. Each week starts on a Monday and ends on a Sunday for 52-53 full weeks in a year.

**Note**: Changing the first day of week after using a dashboard or widget filter/drill with Weeks filter in use might lead to bad results. To resolve the issue, the Weeks filter should be removed and added again.

**First Month of Fiscal Calendar**: Select the first month of your company’s fiscal calendar. By default, the first month of the fiscal calendar is January.

**Note**: Changing the first month of the fiscal calendar after using a dashboard or widget filter/drill with the related date fields is in use might lead to bad results. To resolve the issue the date filter should be removed and added again.

**Default Language**: Select the language to be displayed in Sisense. If you have defined a custom translation, the name of the language as defined in the settings.json is displayed in this list. Select it to display your custom translation. For more information about custom translations, click here.

**Send Emails**: You can disable sending emails. This will disable emails sent when sharing dashboards as well as any scheduled email reports.
**Pulse:** Toggle to remove the Pulse page from the top menu bar and the Pulse Notification Feed. This option hides the Pulse page from all users in your system and stops sending Pulse notifications.

**Dashboard Administration:** Toggle to enable/disable the dashboard administration feature. Turn this feature off if you do not want Administrators to have access to all of the dashboards of every user in the system in the Analytics page. For more information, see Managing Your Users Dashboards.

**Widget Narrative:** Toggle to allow your Designers to add narratives to widgets. For more information about this feature, see Sisense Narratives.

**Field Suggestions:** You can toggle on field suggestions. Field suggestions are displayed to Dashboard Designers to help them add the relevant fields to widgets quickly and easily. For more information, see Field Suggestions.

**Exploration Paths:** Toggle to enable Viewers to receive exploration widgets in the dashboard, see Leverage AI to Uncover Hidden Insights.

**End-User Scheduling:** Toggle to allow the dashboard owners to enable dashboard recipients to manage their own subscription setting, see Sharing and Distributing Dashboards.

**Session Management:** You can choose which method to use for handling the expiration of your users' sessions: Cookie or Session Inactivity. For more information, see Managing User Sessions.

**Embedded Domain White List:** You can define which domains can embed your dashboards into iFrames on their site. This is useful for controlling where your dashboards can be embedded. In the Add Domain field, enter each domain where your dashboards can be embedded and click Add. If you do not add any domains, then your dashboards can be embedded into any site. After adding a domain, your dashboards can be embedded only in those domains.

**Number of failed login attempts before lockout:** Enter the number of times a user can fail to log in before they are locked out of Sisense.

**Lockout duration (minutes):** Enter the number of minutes that a user is locked out of Sisense.

Click Save to update your system settings.
License Utilization

Sisense provides your licensing information in the **Admin** page of the Sisense Web Application.

In the License Utilization section, you can see when your license expires, how many user licenses are in use for your account, and additional features that are supported in your account.

The License Utilization section includes the following information:

<table>
<thead>
<tr>
<th>Details</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>License Expiration Date</td>
<td>The date when your Sisense license expires in the format of Month/Day/Year.</td>
</tr>
<tr>
<td>Trial License</td>
<td>Indicates if your license is for a trial or full version of Sisense. True indicates that you are using a trial license.</td>
</tr>
<tr>
<td>Administrators</td>
<td>The number of Administrator licenses in use out of the number of available licenses in your account. Administrator licenses include Admin, Sys.Admin, and Data Admin.</td>
</tr>
<tr>
<td>Designers</td>
<td>The number of Designer licenses in use out of the number of available licenses in your account. Designer licenses include Designers and Data Designers.</td>
</tr>
<tr>
<td>Viewers</td>
<td>The number of Viewer licenses in use out of the number of available licenses in your account.</td>
</tr>
<tr>
<td>High Availability</td>
<td>Indicates if your account supports high availability.</td>
</tr>
<tr>
<td>Big Data</td>
<td>Indicates if your account supports big data. Big data as defined by Sisense is typically considered to be data sets above ~1 billion.</td>
</tr>
</tbody>
</table>
To access the License Utilization section:

**Note:** This section is accessible to System Administrators only.

In the Sisense Web Application, click **Admin** and select License Utilization in the left menu.
Configuration Manager

This topic provides an overview of the Sisense Configuration Manager, which you can use to configure and administer a variety of functionality and services in Sisense.

For example, you can use the Configuration Manager to perform the following tasks:

- Configure SSL in your Sisense environment
- Configure your application database for high availability
- Define where log files are stored and how detailed your logs are

The Configuration Manager can be accessed from your browser at http://localhost:3030.

Below you can find descriptions of the fields that appear in the Configuration Manager and how they relate to tasks you can perform in Sisense. When relevant, links are provided to topics that provide a more in depth explanation of the field and the tasks you typically use the field for.
Web Server

The following fields are used to configure SSL for your Sisense environment. You should enable SSL if you have configured your web environment with SSL security (HTTPS) and certificates. This will enable HTTPS in the URLs of shared dashboards. For a complete explanation of these fields and their purpose, see Setting up SSL.

- **Enable SSL**: Toggle to enable/disable SSL.
- **Upload Certificate**: Select the type of SSL certificate you are using.
- **SSL Certificate**: Enter your SSL certificate.
- **SSL Key**: Enter your SSL key.

**Port**: The port to be used when connecting to Sisense. The default value is 8081. For traffic over a secure connection, typically, this port is 443.

After setting a new port, restart the following services in Windows Services:

- Sisense.Identity
- Sisense.Galaxy
- Sisense.ECMServer
- Sisense.Gateway

**Proxy URL**: This field is used for implementing Base URLs. See Configuring Base URLs for more information.

**Domain Binding**: This field is used to allow external domains to access Sisense, for example, if you have implemented iFrames. You can add multiple domains separated by a comma and click **Save** to save your changes.

**Build Node Server**: When implementing a multi-node configuration, for example for high availability, you must define which node is your build node in the Configuration Manager. The address of the build node should be the server IP address and not the server name.
Application Database

The following fields are used to communicate with your Sisense application database. Typically, this is only necessary when implementing a high availability environment. For more information, see [High Availability](#).

**Connection String:** The connection string contains all the necessary attributes for accessing the Sisense application database.

**Encryption Key:** The value of this field is the encryption key used by Sisense to encrypt your application database users. This value is used when you need to create users for your application database. See [Hosting the Application Database](#) for more information.
Logs

Sisense stores logs locally on your Sisense server. These logs are typically used by Sisense Support when troubleshooting any issues with your Sisense environment. You can define the location of these logs and the level of logging.

**Path:** The current path when Sisense stores logs. You can enter a new path for your logs on your Sisense server.

**Level:** Select the level of logging.
General

The following fields are used to communicate with your Sisense Configuration database and the Sisense Message Broker. Typically, this is only necessary when implementing a high availability environment. For more information, see High Availability.

**ZooKeeper Connection String:** The connection string to the configuration database. Click **Test** to verify the connection. See [Replicating Zookeeper](#) for more information.

**RabbitMQ Connection String:** The connection string to the Sisense message broker. Click **Test** to verify the connection. See [Replicating the Message Broker](#) for more information.

**Manual Discovery Connections:** This is used internally by Sisense Support and does not need to be configured.
Sisense Migration

Many times, you may want to set up Sisense or upgrade Sisense in a development environment before migrating it to your production environment. In a development environment, you can test and verify your dashboard behavior and data accuracy before publishing dashboards to a production environment.

The migration process for Sisense involves migrating your ElastiCubes, dashboards, and Sisense system configuration.

Additionally, if you are using plugins, you will need to replace the Plugins folder with your current plugins located at C:\Program Files\Sisense\app\plugins.

Backing up your installation is recommended before a migration process, but also on a regular basis for securing your data and Sisense configurations.

Note: The original server and the target servers to which Sisense is being migrated must be running the same version of Sisense.

For a complete walk-through of the migration process, see the video below:
Next Steps

Migrating ElastiCubes
Migrating Dashboards
Migrating Sisense System Configurations
Migrating ElastiCubes

There are two ways to migrate your data to a new server, by exporting and importing .ecube files or exporting and importing .ecdata files. An .ecube file includes the model metadata: model schema and data connection settings. An .ecdata file includes both the model metadata and the data itself. Both .ecdata and .ecube files can be used to make backups and transfer data between ElastiCube Servers, depending on whether you want to move the metadata only, or the metadata and data. After building your ElastiCube from an .ecube file, your ElastiCube with the metadata and the raw data combined is stored in an .ecdata file.

You can migrate ElastiCubes with .ecube files or .ecdata files. As .ecdata files can be quite large depending on how much data you imported, it may be easier to migrate your .ecube files. The steps below describe how to migrate your Sisense ElastiCubes via .ecube files. If you prefer to migrate the .ecdata files, see Migrating .ecdata Files for more information.
Exporting .ecube Files

You can export .ecube files from the Sisense Desktop Manager. Once these files have been exported, you can back them up with your source control and import them into your target environment.

To export .ecube files:
In the Sisense Desktop Manager in the source environment, open the ElastiCube you want to migrate. Select File > Save ElastiCube As... and download the .ecube file to your machine.

Save the .ecube file to your target environment.
Importing .ecube Files

After exporting .ecube files, you can import them into Sisense through the Sisense Desktop Manager. Once the .ecube file have been imported, you must build the ElastiCube to use it with your dashboards.

**To import .ecube files:**

In the Sisense Desktop Manager in the target environment, select **File > Open ElastiCube File** and navigate to your .ecube file.

After selecting the relevant .ecube file, the schema is displayed, you can build the ElastiCube or if you are importing it in a new environment, see [Deploying ElastiCubes](#).
Deploying ElastiCubes

As .ecube files only contain the metadata for your ElastiCube in your source environment, you need to **build** the ElastiCube in the target environment to bring in the data. If your target environment has different data sources from your source environment, for example, the source environment uses development databases whereas the production environment uses production databases, then the .ecube files need to be updated with the new data sources.

There are two ways to edit the ElastiCube with the new data sources:

- Sisense PSM Shell to automate the process
- Manually open the .ecube file and edit it

Both methods are outlined below.
Automating ElastiCube Deployment

The following instructions describe a common scenario for deploying ElastiCubes in a new environment where .ecube files have been imported into a production environment and the data source connections need to be updated.

Editing .ecube Files

Using the Sisense Shell, or Sisense PSM, it is easy to edit the database connection attributes. This set of APIs allows you to edit Elasticube parameters via a command line interface. The most commonly used command in a dev-to-prod scenario redirects .ecube build queries to a production database. Below is an example where “mycube.ecube” is redirected from a development database called “tracking” to a production database called “trackingX”.

```plaintext
psm ecube edit connection database
ecube="c:\users\myuser\documents\mycube.ecube"
server="10.0.0.1" database="tracking"
newserver="localhost" newdatabase="trackingX"
newusername="mynewuser" newpassword="newpassword"
```

The server and database parameters contain the details of the previous database and the newserver and newdatabase parameters should contain the location and name of your updated data source on the production server.

Building ElastiCubes with the PSM

The build can also be initiated on the production server through the Sisense interface or through the PSM. The following example code builds the Elasticube on the production server with the PSM:

```
Full Build, Local Server: psm ecube build name="<elasticube name>"
mode=restart serverAddress=localhost
```
Accumulative Build, Remote Server: psm ecube build name="<elasticube name>" mode=full serverAddress=192.168.1.134
Manually Deploying Elasticubes

If you want to manually update your ElastiCube’s connection settings to reflect your new data sources, on your target server, navigate to the imported .ecube file and double-click to open it. This will open the Elasticube Manager on the server. From here, you can manually edit the connection settings for each data source. This is only necessary if you need to update the connection settings. If your data sources have not changed between environments, you can build the ElastiCube without having to edit it.

Editing the .ecube File

If you are pointing to different data sources in your production environment (e.g., a production vs development database) you can edit the connection settings in the Elasticube Manager.

**To edit .ecube file:**
In the Desktop ElastiCube Manager, select **Elasticube** and then **Change Connectivity Settings**.

![Elasticube Manager]
For each data source that needs to be changed, select the **Change Source Database** icon.

Choose your Database server location, then click **Connect to Server**. Make sure the correct database is selected under **Select Database** and then
select **OK** and **Close**.
Next Steps

Migrating Dashboards
Migrating Dashboards in Sisense

There are several ways to migrate dashboards to other Sisense environments:
- Migrating Dashboards in Sisense (Recommended - described below)
- Migrating Dashboards via REST API
- Exporting and importing .dash files

The easiest way is to migrate the dashboards through Sisense as described below, however, if you want to implement version control for your dashboards, you may prefer to export .dash files that represent your dashboards in Sisense. Administrators and Designers with access to a remote Sisense server can copy dashboards from one Sisense server to another from the Sisense Web Application. When you copy a dashboard or a folder to a remote server, all the dashboards, including those in the selected folders are duplicated into the main Navigation Pane in the target server. Folders are not duplicated on the target server.

Before copying a dashboard to a target server, the Administrator of the target server must provide access to target server. The procedure below explains how to provide access to another server and then how to copy a dashboard from the source server to the target server through the Sisense Web Application.

**Note:** When migrating a dashboard to another server, you must be logged in to your local server with the IP address as the URL. You cannot copy dashboards from localhost.

**To copy your dashboard to another Sisense server:**
On the target server, the Administrator needs to provide access for the source server to the target server. On the target Sisense server, select **Admin > Trusted Server List.**
The Trusted Server List page is displayed. In the Trusted Server List page, click **Add Server**. The Create a New Server Access dialog box is displayed.

### Add Server

<table>
<thead>
<tr>
<th>Server</th>
<th>Type URL or Hostname (Including Port)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Optional</td>
</tr>
</tbody>
</table>

**Note**: Some of the dialog boxes may be considered as pop-ups by your ad blockers. Make sure that Sisense is whitelisted or that your ad blocker is disabled when performing this procedure.

In the **Server** field, enter the IP address or hostname of the source server that contains the original dashboard.

In the **Name** field, enter a name for the source server. This is an optional identifier to help you keep track of your servers.
Click **OK**. The server is added to the list and now has copy access to the target server.

On the source server, from the Navigation Pane, click the dashboard’s menu and click **Copy to Server**.

---

**OR**

To copy multiple dashboards, in the Dashboards list, click the Select Dashboards button to display checkboxes next to the dashboard and folders, which allows you to select multiple dashboard to be exported. Then, select the relevant dashboards and folders to be exported and click **OK**. The Copy Dashboards to Server dialog box is displayed.

---

In the Copy Dashboards to Server dialog box, enter the following information:

**Server**: The URL or Hostname of the remote server. Sisense supports SSL
connections so dashboards can be transferred securely when SSL is activated.

**Republish dashboards after copying:** Select to automatically republish the dashboard on the target server after the selected dashboards are copied. Republishing a dashboard refreshes the dashboard definitions (widget types, titles, scripts etc.) for each of the users who share the dashboard.

Click **OK**.

If the dashboards already exist on the recipient server, Sisense prompts you to select how you want to handle each dashboard that already exists. You can select one of the following options for each dashboard:

- **Skip:** No changes are made to the dashboard on the recipient server.
- **Overwrite:** The new dashboard writes the existing dashboard on the recipient server.
- **Duplicate:** A new dashboard is created on the recipient server with a number added to the end of the dashboard’s title, for example, “Sample Dashboard (1)”.

After you have defined how the dashboards are to be handled, the selected dashboards are copied to the remote server and displayed in the Dashboard’s list.

Once the copy is completed, you may need to update the data source for your dashboard or set data security rules for your ElastiCube (see the links below for detailed documentation):

- [Change the Data Source to the Production ElastiCube](#)
- [Set Security on the Production ElastiCube](#)
- [Share or Republish the dashboard to users](#)
Next Steps

Migrating the Sisense Configuration
Migrating Sisense System Configurations

To migrate your Sisense Web Application configuration, you must copy several directories from the current environment and migrate them to the new environment.

In addition, your Sisense application database has an identifier associated with it when you install Sisense. This identifier must be deleted when migrating Sisense as each database should have its own unique identifier. A new identifier will be generated in the environment the next time Sisense is opened. You can access the database through RoboMongo and remove the identifier as described below.

To migrate the Sisense Web Application:
Back up the old environment, open your machine’s Control Panel, and go to System and Security > Administrative Tools > Services, and stop the
SisenseRepository service (this will make the website inaccessible).

Create a copy of the entire Repository directory located at:

%ProgramData%\Sisense\PrismWeb\Repository

Replace the existing Repository directory in the new environment with the one you backed up.

Create a copy of the entire Discovery directory located at:

%ProgramData%\Sisense\Infra

Replace the existing Discovery directory in the new environment with the one you backed up.

In Windows Services, start the SisenseRepository service and Sisense.Discovery service.

Open Robomongo and connect to your application database with a WriteUser. For more information, see Accessing the Application Database.
Under **PrismWebDB**, go to the **servers** collection and edit the file with address **LocalHost**.
Remove the entire **identity** row including the value.

Open Windows Services and restart the following services:

- Sisense.Configuration
- Sisense.Galaxy
- Sisense.Gateway
- Sisense.Identity
- Sisense.Plugins
- Sisense.EMCServer
- Sisense.ECMLogs
- Sisense.Jobs
- Sisense.StorageManager

Open the IIS Manager and restart the website SisenseWeb.
Embedded Analytics

Sisense’s embedded analytics solution is an end-to-end, single stack BI solution embedded within your application that lets your customers easily prepare, analyze and visualize complex data. Sisense’s embedded analytic solution offers built-in features and exposes a variety of functionality through several interfaces, such the Sisense JavaScript and REST APIs for customizing your integration.

You can accomplish the following through the Sisense embedded analytics solution:
- Seamlessly integrate Sisense with your current interface and branding
- Manage data, reporting and visualization
- Integrate SSO and Active Directory
- Govern user and data security
- Support high availability

This documentation is intended for two audiences, product managers and developers. For product managers, those who want to enhance their products and applications with Sisense embedded analytics, this section describes the features and functionality Sisense offers as part of an embedded analytics solution. For developers, those who embed Sisense in the application, this section describes how to embed Sisense analytics through the relevant APIs.
Overview

When you embed Sisense analytics, you can provide your users with Sisense’s analytical capabilities such as data management, reporting and visualization, built into your business applications and solutions.

Each OEM has their own requirements when embedding Sisense’s analytic solution. This section covers the multiple ways in which Sisense’s OEM features can be embedded to provide BI capabilities in your applications. Sisense exposes functionality through several interfaces. For example, Sisense’s REST API allows you to white label Sisense and Sisense’s command shell allows you to couple Sisense with your on-premise installation to modify connection settings. You determine which features you want to embed according to the type of OEM services you provide to your customers.

The following table lists the various OEM-related tasks you can perform with Sisense and the description with links to the relevant documentation for completing each tasks:

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rebranding Sisense</td>
<td>Sisense enables you to white label the Sisense Web Application to display your company's logos and colors and mask the URL to display your company's domain. For documentation regarding how you can customize and rebrand your Sisense solution, see the following topics: <a href="#">White Labeling Sisense</a> <a href="#">Rebranding Sisense Automated Emails</a> <a href="#">Displaying Custom URLs</a></td>
</tr>
<tr>
<td>Embedding Sisense</td>
<td>Sisense also allows you to embed widgets and dashboards directly into your site or application through the use of iFrames</td>
</tr>
<tr>
<td>Task</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>or SisenseJS, Sisense's custom JavaScript library for embedding widgets.</td>
<td>For documentation regarding how you can embed Sisense, see the following topics: <a href="#">Embedding Dashboards and Widgets via iFrames</a> <a href="#">Embedding Dashboards and Widgets via JavaScript</a></td>
</tr>
</tbody>
</table>
Considerations

Before embedding Sisense analytics, you should keep in mind several considerations that may affect your design and deployment:

**Users**: Who will be using your application? This determines what interfaces need to be white labeled and which API calls you should leverage.

**Performance**: As with any deployment, you must provide sufficient CPU and memory on the client and the server for your anticipated user load.

**Availability**: What is your desired up-time and how many resources can you dedicate to maintaining system availability.
OEM Architecture

Sisense has many customers who use Sisense as an OEM Business Intelligence platform. This means that a customer is using Sisense to provide analytics to their own external customers. Customers’ customers, can be referred to as “Tenants”.
Rebranding and Embedding Sisense

OEM deployments comprise of two main features: rebranding and embedding. In the white-labeled use-case, a Sisense server is completely rebranded. Customers replace the Sisense logo and name, change the color palette used by the application, customize the emails sent by the system, and more. The tenants access the Sisense server directly, and experience Sisense analytics as customized by the customer. They are not necessarily aware that the analytics server they are accessing is actually a Sisense server.

In the embedded use-case, Sisense customers have their own web application. They embed Sisense within it. Sisense supports embedding the full Sisense application, including the data modeling, analytics and administration areas, or embedding specific dashboard and widgets using iFrames. Widgets can also be embedded within web pages using the SisenseJS infrastructure.

Both of these use-cases, the white-labeled server and the embedded offering, rely on strict data governance to ensure tenants get access only to their own assets and data.

This type of governance can be achieved using a few different OEM architectures. The selected architecture depends on the customer requirements and needs, and their own preferences. Sisense provides the flexibility to support OEM deployments in the way that customers want to work.

In this topic, a few different ways to support OEM use-cases are outlined, and describe the benefits and disadvantages of each.
Option 1: Shared ElastiCubes with Row-Based Data Security

**Advantages:** Low hardware costs, high resource utilization, simple asset change management

**Disadvantages:** Tenant resource usage may affect other tenants

**Typically Best for:** Tenants with identical data models and dashboard requirements

The first type of architecture for OEM deployments utilizes shared Sisense servers for multiple tenants, and shared ElastiCubes and dashboards. Segregation between tenants is achieved by using row-based data security within ElastiCubes. All of the customers’ data resides in a shared ElastiCube, but each of the tenants gets access only to their own data.

This methodology makes it easy to maintain system assets, and handle changes that occur during the lifecycle of the asset. Modifying the ElastiCube schema is done only once in the shared ElastiCube for all users. Changing the shared dashboards is also done once for all users.
This methodology is well suited for customers providing a service based on identical data sources and reports. For example, a customer analyzing shopping statistics for tenants on a shared shopping portal will take data for all of the customers from the shopping portal analytics data using each tenant’s credentials. But all of the data has the same exact structure, so the generated dashboards are common analytics of this data.

This methodology provides good utilization of the server resources, and ensures that asset maintenance remains easy.

Note that as the server is shared by multiple tenants, they are also sharing resources. High resource usage by one of the tenants, for example, generating multiple reports or heavy builds that require a lot of CPU, may affect other tenants.
Option 2: Dedicated Cube per Tenant

**Advantages:** Low hardware costs for small number of tenants, high resource utilization

**Disadvantages:** High hardware costs for large number of tenants, more complicated asset change management, tenant resource usage may affect other tenants

**Typically Best for:** Tenants who require customized data models

The second architecture for OEM deployments utilizes shared Sisense servers for multiple tenants, together with providing a dedicated ElastiCube and dashboards for each tenant. In this deployment, multiple tenants use the same server. Typically, the OEM has default ElastiCubes and dashboards, and creates a dedicated copy of them for each of the tenants. The ElastiCubes and dashboards can be identical copies for each of the tenants, or customized per tenant. The OEM uses access control for ElastiCubes and dashboards to ensure each of the tenants only has access to their own data. Typically a user group is created for each of the tenants. All of the tenants users are assigned to the same group. The relevant ElastiCubes and Dashboards are shared with the tenants group. In this
way, the asset access control layer ensures that users of each tenant only gets access to their own data.
This methodology is well suited for customers whose data sources are not identical. It allows for customizing the data import process so that the data preparation and ETL processes can handle the specific customers data structure, and transform it to the desired target structure. This methodology is also well suited for tenants who have different reporting needs. While the tenant is initially provisioned with default ElastiCubes and default dashboards, customizations can be made per tenant, without affecting other tenants. It is easier to accommodate the specific needs and requirements of each tenant.
In this use case, the OEM needs to manage multiple copies of ElastiCubes and dashboards. Making a change to an ElastiCube schema or to dashboards needs to be replicated across all of the tenants. When there are many tenants, the cost of making changes to assets is high.
This methodology provides better utilization of the server, and can reduce the cost of ownership as multiple tenants are sharing the same server. But as with the previous option, shared servers means that multiple tenants are also sharing the server resources. The behavior of one tenant may affect other tenants using the server.
With this solution, you need to consider how the system scales to support your future needs, to support additional tenants. There are limitations to the number of ElastiCubes that can be deployed on a single machine. While initially, you enjoy shared server resources, as you add more tenants, you may have to provision additional servers, increasing the hardware costs of this solution.
Option 3: Dedicated Sisense Server per Tenant

**Advantages**: Highest level of security, dedicated resources per tenant

**Disadvantages**: Low resource utilization, high hardware costs, complicated asset change management

**Typically Best for**: Tenants with strict security regulations, such as financial or healthcare institutes and tenants that need a high level of schema and dashboard customizations

The third architecture for OEM deployment is to provide a dedicated Sisense server for each tenant. Typically an OEM customer will have a server image including default ElastiCubes and dashboards. Each of the customers receives their own instance of the server. The ElastiCubes and dashboards can be identical between the servers, or customized for each of the tenants. The data for each tenant is completely separate as each server has its own assets, including configuration, users, ElastiCubes and dashboards.
This option offers a high level of customization per tenant. Each of the tenants has their own ElastiCubes and dashboards, and they can be modified to accommodate the tenants specific needs, without affecting any other customers. This option also provides a very high level of governance and security for each of the tenants, as the OEM can be assured that no tenant can access assets of another tenant. Additionally, the OEM is assured that the system behavior of one tenant will not affect other tenants. If there is a heavy system load caused by multiple queries or reports generated by one tenant, it will not have any effect on the other tenants.

This deployment architecture is suitable for cases where there may be legal restrictions that mandate that data of different tenants cannot be located on the same physical machine.

However, for multiple reasons, this option can be expensive and not very cost effective to maintain. A tenants server may be underutilized if it’s not used throughout the day. Or if most of the users reside in the same timezone, the server may be idle for long periods of the time. Additionally, a Sisense server is a high performance server and capable of supporting a large number of concurrent users. If a tenant does not have many users, the server may be underutilized even during work hours of the tenant, and could have possibly been used by additional users.

Another issue is the configuration maintenance of the ElastiCubes and dashboards. If an OEM needs to make a change to the ElastiCubes and dashboards that are provided to the tenants, then all the changes will have to be made on each one of the servers. For example if the OEM wants to create a new dashboard, and share it with all of their tenants, they will have to copy the dashboard to each one of the tenants servers. This can require quite a lot of work.
In this deployment option there’s a trade-off between complete segregation and dedicated resources, and the cost of ownership and maintenance.
Summary

Sisense supports a range of flexible architectures for OEM deployment. The type of architecture suitable for a specific customer depends on the use-case, the needs of the customer, the resources that can be dedicated to the deployment (both allocated people, and allocated hardware), and preferences.
Embedding Dashboards and Widgets

This article is about getting started with embedded Sisense dashboards. If you’re new to Sisense, check out our [embedded analytics solutions](#).

You can embed Sisense dashboards and widgets in non-Sisense environments, such as in your own websites or applications by concatenating parameters to the URL of your dashboard or widget. In addition, you can customize what aspects of the dashboard and widget are displayed in your environment.

For example, if you want to remove the Sisense header from the dashboard, you can concatenate the “?embed=true&h=false” parameter in the URL of the dashboard to hide the header.

```
mysite.com/app/main#/dashboards/5541dc7a80a4e2181e00011a?embed=true&h=false
```

In this example, the “embed=true” embeds the dashboard into your environment. When you embed a dashboard or widget, viewers can only view the dashboard or widget and do not have any editing privileges. The “h=false” parameter hides the Sisense header allowing you to whitelabel the dashboard.

This article provides a full list of the parameters you can concatenate, their functionality, and examples.
Embedding Dashboards

To embed a dashboard, add “?embed=true” to the URL of the dashboard, and refresh the page. If your dashboard is within a folder, then first delete “?folder=” and everything that follows, before adding “?embed=true”.

This will open the dashboard without the surrounding Sisense environment. For example:

Dashboard URL:
myosite.com/app/main#/dashboards/5541dc7a80a4e2181e00011a

Embedded URL:
myosite.com/app/main#/dashboards/5541dc7a80a4e2181e00011a?embed=true

Dashboard URL for dashboard within folder:
myosite.com/app/main#/dashboards/550952417404b2981a000029?folder=550955a27404b2981a00003b

Embedded URL for dashboard within folder:
http://localhost:8081/app/main#/dashboards/550952417404b2981a000029?embed=true

You can use this URL to embed the dashboard in an iframe, for example:

```html
<iframe id="ifm" name="ifm" width="100%" height="100%" frameborder="0"
src="http://myosite.com/app/main#/dashboards/536f2b70d093e26c280000d5?embed=true" scrolling="auto"></iframe>
```
Dashboard Embedding Configuration Options

You can control which aspects of the Sisense environment are available when embedding by concatenating the following parameters to the embedded URL:

**Show/Hide the left Navigation Panel**
Parameter: "l" (left) or "i" to hide it completely
Default: false
Example: ?embed=true&l=true

**Show/Hide the right filter panel**
Parameter: “r” (right)
Default: true
Example: ?embed=true&r=false

**Show/Hide the toolbar**
Parameter: “t” (toolbar)
Default: false
Example: ?embed=true&t=true

**Show/Hide the environment header**
Parameter: “h” (header)
Default: false
Example: ?embed=true&h=true
Embedding Widgets

To embed a single widget, add “?embed=true” to the URL of the widget when it is in edit mode.

This will open the widget without the surrounding Sisense environment.

For example:

**Dashboard URL:**
http://mysite.com/app/main#/dashboards/536f3a54d093e26c28000114/widgets/536f3a54d093e26c2800011b

**Embedded URL:**
http://mysite.com/app/main#/dashboards/536f3a54d093e26c28000114/widgets/536f3a54d093e26c2800011b?embed=true

You can use this URL to embed the widget in an iframe, for example:

<iframe id="ifm" name="ifm" width="100%" height="100%" frameborder="0"
src="http://mysite.com/app/main#/dashboards/536f3a54d093e26c28000114/widgets/536f3a54d093e26c2800011b?embed=true"
scrolling="auto"></iframe>
Widget Embedding Configuration Options

You can control which aspects of the Sisense environment are available when embedding by concatenating the following parameters to the embedded URL:

**Show/Hide the left data panel**
Parameter: “l” (left)
Default: false
Example: ?embed=true&l=true

**Show/Hide the right design panel**
Parameter: “r” (right)
Default: false
Example: ?embed=true&r=false

**Show/Hide the toolbar**
Parameter: “t” (toolbar)
Default: false
Example: ?embed=true&t=true

**Show/Hide the environment header**
Parameter: “h” (header)
Default: false
Example: ?embed=true&h=true
Authentication in Embedded Dashboards and Widgets

All Sisense dashboards and widgets require authentication. When embedding dashboards and widgets outside of the Sisense environment, usually the use case is to not enforce the Sisense environment authentication, but rather use SSO (Single Sign On) with existing corporate authentication. To read more on configuring Single Sign On in Sisense, [click here](#).
Displaying Custom URLs

You can display your company’s URL in the address of your Sisense dashboards by defining an alias URL from the Sisense Admin page.

This is useful when you are providing OEM services and you want to mask Sisense’s URLs and branding.

To implement an alias URL in Sisense:

Click Admin in the top menu, and then Settings in the left menu.

In the Alias field, enter the web server’s alias or IP address. This is especially important when you have configured your IIS with a domain name (alias), but when you share a dashboard, the IP address and/or the incorrect port is sent instead. Reports will be sent from the alias entered in this field.

Click Save.
Rebranding Sisense Automated Emails

Sisense automated emails are sent to users according to predefined scenarios. When a predefined scenario such as a build alert is triggered, an automated email is sent to the relevant Sisense users. You can customize these emails to match your company's branding. This page describes the Sisense automated emails and how you can replace them with your own so each time a predefined scenario is triggered, Sisense sends your customized emails to your Sisense users.
Sisense Automated Emails

You can rebrand the following automatic emails:

- **Build Alert**: Email sent each time a build alert is triggered.
- **Dashboard Errors Reports**: Email sent when a dashboard report fails.
- **Dashboard Errors Reports Bulk Fail**: Email sent when a large amount of email reports fail.
- **Dashboard Reports**: Email sent to the recipient of a shared dashboard when you share a dashboard.
- **Empty Dashboard Report**: Email sent to an administrator when an empty dashboard report is sent.
- **KPI Alert**: Email sent each time a KPI alert is triggered.
- **New User Invitation**: Email sent to a new user invited to Sisense.
- **Password Recovery**: Email sent to a user who requested a new password.
- **Share ElastiCube**: Email sent to the recipient of a shared ElastiCube.
- **Share With**: Email sent to the recipient of a shared dashboard.
- **Transfer Ownership**: Email sent to a user who received ownership of a dashboard from another owner.
- **User Created**: Email sent to a user when a user is created.
- **User Created AD**: Email sent when a user is created in Sisense after being added from Active Directory.

For each automated email, Sisense provides an email template in the format of EJS and LESS files. These files are located within template folders inside the Template Parent folder located at:

**In Windows:**

```
...\Program Files\Sisense\app\translations\{LOCALIZATION_CODE}\email-templates.js
```

**In Linux:**

See [Uploading Files through the File Manager](#) and upload the files to the `emails` folder.
LESS Files

All templates contain the style.less file that refers to a stylesheet that defines the branding used in Sisense’s automated emails. The content of every style.less is as follows:

@import '../styles.less';

The stylesheet referred to by all the email templates is called styles.less located at:

In Windows:

...\Program Files\Sisense\app\galaxy-service\src\features\emails\templates

In Linux:

/opt/sisense/storage/emails

By replacing the content of the styles.less file with your own CSS, you can redefine the branding used in all of the email templates to your custom branding.
EJS Files

Each template folder contains an EJS file, which contains HTML and JavaScript that describe the content of the email.
You can use any text/HTML editor to modify the HTML.ejs files according to your needs.
The following is an example of an HTML.ejs file.

```html
<% include ../header %>
<tr>
    <td class="contentText">
        <p class="userText" style="margin: 0;">% i18nContent.hi % <%= newOwnerUserName %>,</p>
        <p class="generalInfo" style="margin: 0;">% i18nContent.generalInfo %</p>
    </td>
</tr>
<% include ../footer %>
```
Each EJS file has three sections, a header, body, and footer; each of which describes various areas in the email your users receive. Section 1 is the Header section defined in the EJS file as `<% include ../header %>`. This is a reference to the header.ejs file located at:

`...\Program Files\Sisense\app\galaxy-service\src\features\emails\templates`

The header.ejs file defines the title and image used in the header section of each email template. You can modify the header by replacing the header.ejs file with or replace the `<% include ../header %>` tag from each template with your own HTML and JavaScript.
Section 2 is the Body section, which contains the content displayed in the email your users receive. There are two types of content displayed in automated emails, tokens and strings. Tokens refer to Sisense variables such as the Sisense username or dashboard owner’s name. For example, the `<%= newOwnerUserName %>` token displays the new owner of a dashboard’s Sisense username in the automated email.

```html
<tr>
  <td>
    <div class="contentText">
      <p class="userText" style="margin: 0;">%= i18nContent.hi %<%= newOwnerUserName %>,</p>
      <p class="generalInfo" style="margin: 0;">%= i18nContent.generalInfo %</p>
    </div>
  </td>
</tr>
```

When you replace the HTML.ejs file with your own files, you can still use the Sisense tokens. For more information about tokens you can add to your automated emails, see Applying Sisense Tokens.

In addition, each template contains an object such as `i18nContent.hi` that displays strings predefined by Sisense. These strings cannot be modified, however, you can remove and replace the object with your own content.

The final section, Section 3 contains the footer. Like the header, the footer, `<% include ../footer %>`, is a reference to a footer.ejs file located at:

**In Windows:**
```
...\Program Files\Sisense\app\galaxy-service\src\features\emails\templates
```

**In Linux:**
```
/opt/sisense/storage/emails
```
The footer.ejs file defines the copyright text and image used in the footer section of each email template. You can modify the footer by replacing the footer.ejs file with or replace the `<% include ../footer %>` tag from each template with your own HTML and JavaScript.

Images that are displayed in the Header and Footer are stored in the following location:

**In Windows:**

`...\Program Files\Sisense\app\galaxy-service\src\features\emails\templates\images`

**In Linux:**

`/opt/sisense/storage/emails/images`

You can replace these with your own images by maintaining the same file names used, however, it is recommended to replace the content of the Header and Footer sections altogether with your own content.
Replacing Sisense Automated Emails

If you prefer to implement your own email templates for Sisense’s predefined scenarios, you can replace Sisense’s emails with your own through the Sisense REST API.

**Note:** Rebranding Sisense emails must be enabled for your license. Contact your Account Manager for more information.

For each automated email, Sisense provides an email template in the format of EJS and LESS files. These files are located within template folders inside the Template Parent folder located at:

**In Windows:**

```bash
...\Program Files\Sisense\app\galaxy-service\src\features\emails\templates
```

**In Linux:**

```
/opt/sisense/storage/emails
```

Some template folders may contain an html-org.ejs file. These files are legacy templates and are maintained for backwards compatibility. Newer developments should only modify or replace the html.ejs files.

**Note:** While it is possible to modify the content of your automated emails through the language.js file located in the templates folder, the recommended method is modify the email templates as described in the procedure below.

After you have prepared your new automated emails, through the /branding endpoint of the REST API, you can configure Sisense to send the customized automated emails when the relevant event is triggered.

**To define the location of your branded emails:**

Copy the contents of the directory:

**In Windows:**

```bash
...\Program Files\Sisense\app\galaxy-service\src\features\emails\templates
```
In Linux:
/opt/sisense/storage/emails
OR
Upload all the folders to your Sisense Server through the File Manager. See Uploading Files through the File Manager for more information.
Paste all the folders in the following directory:
...\Program Files\Sisense\app\galaxy-service\src\features\emails\

In Linux:
/opt/sisense/storage/emails
OR
Upload all the folders to your Sisense Server through the File Manager. See Uploading Files through the File Manager for more information.

Note: You will modify the templates in this folder and configure Sisense to send them instead of the original Sisense automated emails.
For each template you want to modify, change the contents of html.ejs files and styles.less as described in Sisense Automated Emails.
Access the Sisense REST API.
In version .9 of the REST API, select the POST /branding.

In the email object define the following information:

**Note:** If you have already rebranded your Sisense Web Application, copy the current settings through the GET /branding endpoint and paste them into the POST /branding endpoint modifying or adding the emails object. If you only modify the emails object, the remaining objects and keys are set to their default Sisense and will overwrite any existing settings.

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>senderEmail</td>
<td>The email address from which the automated email is to be sent. When you define the senderEmail value, you must verify your domain in Sisense's Mandrill email service. Click <a href="#">here</a> for more information.</td>
</tr>
<tr>
<td>senderName</td>
<td>The name to be displayed as the Sender in the automated email.</td>
</tr>
<tr>
<td>templates_directory</td>
<td>The directory location of your template folders. Sisense accepts /resources/branding/emails as a custom location for automated email templates. This directory is located on your Sisense</td>
</tr>
<tr>
<td>Key</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| server at                    | server at ...
| resources/branding/emails    |server at ...
| passwordRecoverySubject      | The subject line of the password recovery email.                            |
| newUserInviteSubject         | The subject line of the new user email.                                     |
| createdUserSubject           | The subject line of the new user invite email.                              |
| shareWithNewUserSubject      | The subject line of the shared dashboard with a new user email.             |
| shareWithExistingUserSubject | The subject line of the shared dashboard email.                              |
| transferOwnership            | The subject line of the dashboard transfer ownership email.                 |

For example:

```
"emails": {
    "senderEmail": "sys-admin@company.com",
    "senderName": "sys-admin",
    "templates_directory": "/resources/branding/emails",
    "passwordRecoverySubject": "Recover Password",
    "newUserInviteSubject": "new User Account Activation",
    "shareWithNewUserSubject": "A Dashboard has been shared with you:"
    "shareWithExistingUserSubject": "A Dashboard has been shared with you:"
},
```
Click **Run**. The next time an automated email is sent, your templates are sent instead of Sisense’s predefined emails.
Applying Sisense Tokens

Most Sisense automated emails include tokens that you can apply to your automated emails. When the email is sent, Sisense populates the token with the relevant value. For example, the `<% owner %>` token represents the owner of the dashboard.

Each template has tokens defined by Sisense for each automated email that you can apply listed below. Tokens from one automated email cannot be applied to another.

**build_alert**: Sent when a build alert is triggered.

No Template Tokens Available

**Dashboard_errors_report**: Sent when a Dashboard email report fails.

Template Tokens Available

  `<% errors[i].error %>`
  `<% errors[i].widgetFailed %>`
  `<% errors[i].widgetCount %>`

**Dashboard_errors_report_bulk_fail**: Sent when a Dashboard email report sent to a large amount of users fails.

Template Tokens Available

  `<% errors.groups[i] %>`

**Dashboard_report**: An automated email that contains a dashboard report sent to users.

Template Tokens Available

  `<% url %>`
  `<% images[i] %>`

**Empty_dashboard_report**: An automated email that contains an empty dashboard report sent to users.

No Template Tokens Available
**Kpi_alert:** Sent to shared users when a KPI alert is triggered.

Template Tokens Available

```
<% measure %>
<% value %>
<% message %>
```

**New_user_invitation:** Sent to a new user that you have invited to join.

Template Tokens Available

```
<% owner %>
```

* The folder for this template contains a file called HTML-ORG.ejs. This file is for older versions of Sisense and is maintained for backward compatibility.

**Password_recovery:** Sent when a user requests their password from the Forgot Password page.

No Template Tokens Available

* The folder for this template contains a file called HTML-ORG.ejs. This file is for older versions of Sisense and is maintained for backward compatibility.

**Share_with:** Sent to a new recipient of a dashboard when the dashboard is shared with them.

No Template Tokens Available

* The folder for this template contains a file called HTML-ORG.ejs. This file is for older versions of Sisense and is maintained for backward compatibility.

**transfer_ownership:** Sent to the new owner of a dashboard when ownership is transferred.

Template Tokens Available

```
<% newOwnerUserName %>
<% oldOwnerUserName %>
<% sharingDashboardName %>
```

**User_created:** Sent to the recipients of new Sisense accounts.

No Template Tokens Available
**User_created_ad**: Sent when a user is created in Sisense after being added from Active Directory.

No Template Tokens Available
White Labeling Sisense

Sisense enables you to rebrand (white label) the Sisense Web Application into your native application or site by using the Sisense REST API to replace existing icons and images with your own.

In addition, you can rebrand the automated emails Sisense sends to match your company’s branding.

White labeling must be enabled in your license. To enable this feature, please contact your Sisense representative or open a request through our Help Center.

In addition, Sisense now offers a premium white labeled version of the Sisense Mobile app that can be customized according to your branding, for more information see Rebranding Sisense Mobile.
Rebranding Sisense

You can rebrand the Sisense Web Application and the automated emails sent by Sisense through Sisense’s REST API.

There are two steps to rebranding the Sisense Web Application. The first step is to place your branded content into the Sisense branding directory where Sisense is installed. For a list of the files you can rebrand and their specific dimensions, see White Labeling Configuration Options.

The second step is to access the Sisense REST API, where you can embed your content directly into Sisense and modify text displayed to customers.

To rebrand the Sisense Web Application:

Create a subfolder with your brand name in this directory (on the machine where Sisense is installed):

**In Windows:**
```
“…Program Files\Sisense\app\resources\<YourBrand>”.
```

**In Linux:**
See Uploading Files through the File Manager and upload the files to the branding folder.

OR

Upload all the folders to your Sisense Server through the File Manager. See Uploading Files through the File Manager for more information.

Put your branded files in this directory. The following files are supported:

- Desktop Logo (for example: Logo-Desktop.png/.jpg) – This is the main icon that is visible at the top left of the Sisense Web Application.
- Tablet Logo (for example: Logo-Tablet.png/.jpg)
- Mobile Logo (for example: Logo-Mobile.png/.jpg)
- Favicon (for example: favicon.ico)

**Note:** Optimal dimensions for logo images: 92×26

In the Sisense Web Application, click **Admin**, and then **Rest API**, followed by REST API Reference.

In the REST API Reference section, select version 0.9 in the top right corner.

Click /branding and then click the POST operation /branding.

To get the required JSON format for the request, click the sample model schema to place the sample text in the body parameter.
Replace the sample values with your required values. For an explanation of each parameter and how it modifies the Sisense Web Application, see White Labeling Configuration Options.

Click **RUN**. Refresh the Sisense Web Application page to view your changes.

**Note:** You can change just some of the parameters by leaving out those you do not want to update.
White Labeling Configuration Options

The following table provides a list of the features you can rebrand within the Sisense Web Application and the JSON code that must be modified to rebrand it. Click the thumbnail image of the feature to see the default branding provided by Sisense and where your branding will appear.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Sisense Logo     | The Sisense logo can be replaced with your own. You need to supply two sizes of the logo, for each device (Desktop / Tablet / Mobile). These logos affect the following screens:  
                          - Login Screen  
                          - Account Activation Screen  
                          - ‘Forgot Password’ Screen  
                          Below is a list with the ideal logo dimensions (pixels): |
|                  | Small  |
|                  |    Desktop: 120x48  
|                  |    Tablet: 92x24  
|                  |    Phone: 92x24  |
|                  | Large  |
|                  |    Desktop: 204x60  
|                  |    Tablet: 163x74  
|                  |    Phone: 198x168  |

Important Notes:
Images must be in URI format (web address)
No spaces are allowed in the address of the images
Transparent .png format is preferred

"logo":{
  "desktop":{
    "small":"/resources/<YourBrand>/logoDesktopSmall.png",
    "large":"/resources/<YourBrand>/logoDesktopLarge.png"
  },
  "phone":{
    "small":"/resources/<YourBrand>/logoPhoneSmall.png",
    "large":"/resources/<YourBrand>/logoPhoneLarge.png"
  },
  "tablet":{
    "small":"/resources/<YourBrand>/logoTabletSmall.png",
    "large":"/resources/<YourBrand>/logoTabletLarge.png"
  }
},

<table>
<thead>
<tr>
<th>Page Title and Favicon</th>
</tr>
</thead>
<tbody>
<tr>
<td>The page title that appears in the browser tab/page and the Sisense favicon.</td>
</tr>
</tbody>
</table>

**Note:**
| **Homepage** | The home screen when your users log into Sisense. Rebranding the homepage does not affect the view of the Administrator. Administrators see the default homepage, while all other users see the rebranded page. **Note**: The prefix http:// or https:// must be included. “homePage” : “http://www.mysite.com”, |
| **Homepage Config** | The Welcome banner of the homepage displayed when users log into Sisense. You define the following properties in the homePageConfig object: **hideSearchInNavver**: Indicate if you want to display the Search bar below or hide it. By default, the Search bar is |
marketingPart: The URL of the Marketing Banner iFrame. This iFrame is displayed on the top part of the Analytics and Data pages where the Welcome message and Sample Dashboards and ElastiCubes are displayed. You can replace this iFrame with your own content by defining the URL as the value of this key.

tutorialsLink: The Tutorials button. Enter a URL that opens your site.

documentationLink: The Documentation button. Enter a URL that opens your documentation site.

Login Page

The Login page title and subtitles. “loginTitle” : “WELCOME TO SISENSE”,”loginSubtitle” : “Sign in & start exploring”

Note: To change the color of the Login button, click here.
Copyright Text

The copyright text that appears in Sisense. "copyrightText":"Copyright © 2018 Sisense inc. All rights reserved", 
Activation Titles

The titles on the activation page for new users signing in to Sisense.
“activationTitle”: “First Time Activation”,
“activationSubtitle”: “Please choose a password to begin using Sisense”,

Welcome to Sisense
Sign in & start exploring...

gideon@sisense.com

Remember me

Forgot your password? Click here

Copyright © 2018 Sisense inc. All Rights Reserved
<table>
<thead>
<tr>
<th>Contact Us</th>
<th>The Contact Us text is displayed when you request a password and enter your email on the Recover Password page. This text displays a message and provides a link to your Support services. “contactUsText” : “For further assistance, please contact our success management team!: &lt;a href='mailto:support@sisense.com'&gt;<a href="mailto:support@sisense.com">support@sisense.com</a>&lt;/a&gt;”,</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Support</td>
<td>The User Support text is displayed in the</td>
</tr>
</tbody>
</table>
To top right corner of the Sisense Web Application.

By default, the links redirect customers to the Sisense user forums and documentation site. You can modify these URLs to redirect users to your user forums or documentation. This is useful for example when providing OEM services or if you have special forums or instructions to display to your users.

```json
{  "documentationUrl": "string",  "forumUrl": "string"}
```

Replace the string with the URL to the relevant locations to redirect your users to your support sites.

### Powered by Sisense

This string is displayed by default in the User Profile for rebranded servers. You can remove this string by setting the boolean `PoweredBySisense` value to `false`.

### Emails

The sender information and subject line text displayed in automated emails. For further customization, see [Rebranding Sisense Automated Emails](#).

Note: Branded emails (for example, using...
the senderEmail and senderName parameters) will be used for new user/group invitation emails and password recovery emails. When sharing a dashboard report, or for subscription emails, the email will be sent without the original name and email of the dashboard’s creator.

"emails" :
{"senderEmail":"sys-admin@company.com","senderName":"sys-admin","templates_directory" :
"/resources/<YourBrand>/emails","passwordRecoverySubject" :
"Sisense password recovery", "newUserInviteSubject" :
"Sisense account activation", "shareWithNewUserSubject" :
"A Dashboard has been shared with you:", "shareWithExistingUserSubject" :
"A Dashboard has been shared with you:" , "transferOwnership":}
"string"}
Rebranding Sisense Mobile

Sisense enables you to provide your users with a fully white-labeled version of Sisense Mobile that matches your brand's look and feel.
The following objects can be rebranded:

- App icon and title
- Splash screen
- Mobile notification logo image and colors
- Mobile color palette
- App description and information that appears in the app store

Rebranding Sisense Mobile is a premium Sisense offering. Contact your CSM if you are interested in your own white-labeled version of Sisense Mobile.

**Note:** iOS does not support embedded IFrames.
Sisense Mobile Checklist

Once you have purchased a rebranded version of Sisense Mobile, you need to download and complete this checklist, which details all the information and resources you must provide Sisense.

The resources include things like icons in different sizes (iPhone, Android), specific colors, mobile app name, and other configurable options within Sisense Mobile. These resource files should be zipped together, and the zipped file and your completed checklist should be sent to rebranded_mobile@sisense.com with your CSM CCed in the email.

After receiving all the assets and your completed checklist, Sisense will create a rebranded mobile app and provide you with a version to try out. The white-labeled mobile app will be fully tested by Sisense QA, but Sisense recommends that you test it out for yourself before uploading it to the app stores.

Sisense will provide instructions for signing the application and loading it up to the app store.

Rebranding Sisense Mobile typically takes up to one month.
## Resources Required for Rebranding Sisense Mobile

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>The date you completed the form.</td>
<td>None. Mandatory</td>
</tr>
<tr>
<td>Company Name</td>
<td>The name of your company.</td>
<td>None. Mandatory</td>
</tr>
<tr>
<td>Contact Person</td>
<td>The full name of the person responsible for managing the rebranding of Sisense Mobile.</td>
<td>None. Mandatory</td>
</tr>
<tr>
<td>Contact Email</td>
<td>Your email address.</td>
<td>None. Mandatory</td>
</tr>
<tr>
<td>CSM Name</td>
<td>The name of your Sisense CSM.</td>
<td>None. Mandatory</td>
</tr>
<tr>
<td>Display Name</td>
<td>The application name that is displayed below the icon on the phone home screen.</td>
<td>None. Mandatory</td>
</tr>
<tr>
<td>Android package ID</td>
<td>Google Play identifier. The ID must match the criteria described <a href="#">here</a>. This ID must be unique. You can verify whether or not the ID is taken by opening this URL</td>
<td>None. Mandatory</td>
</tr>
<tr>
<td>Information</td>
<td>Description</td>
<td>Default Value</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td></td>
<td><strong><a href="https://play.google.com/store/apps/details?id=com.mycompany.myapp">https://play.google.com/store/apps/details?id=com.mycompany.myapp</a></strong> replacing com.mycompany.myapp with your preferred app ID.</td>
<td></td>
</tr>
<tr>
<td>iOS Bundle ID</td>
<td>App Store identifier. This ID must be unique. For iOS, you need to sign in to your account at <a href="https://developer.apple.com">https://developer.apple.com</a> and click Certificates, IDs &amp; Profiles &gt; App IDs &gt; Registering an App ID (+ sign) &gt; select Explicit App ID.</td>
<td>None. Mandatory</td>
</tr>
<tr>
<td>Server IP Address/hostname</td>
<td>If this option is provided, the rebranded mobile app will be bound to this specific IP address. This means that users of your app will be able to skip the screen for providing the server IP address/hostname. Keep in mind that if you do provide this address, your app can be used only for this IP address. If you intend to use the mobile app for different Sisense servers, or you are not sure your IP address/hostname will remain your permanent identifier, you should not provide it.</td>
<td>None. Optional</td>
</tr>
<tr>
<td>Information</td>
<td>Description</td>
<td>Default Value</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Primary background color</td>
<td>Background color of the top navigation bar, the splash screen and the buttons.</td>
<td>#FFCB05 (Sisense yellow)</td>
</tr>
<tr>
<td>Primary text (foreground) color</td>
<td>The color of the text that is positioned on top of the components with primaryBGColor background.</td>
<td>#000 (black)</td>
</tr>
<tr>
<td>Title highlight color</td>
<td>The color of highlighted text that is displayed on a white background (e.g. Recent Dashboards and My Dashboards).</td>
<td>#F2B900 (dark yellow)</td>
</tr>
<tr>
<td>Title border color</td>
<td>The color of separators and pipes that are displayed around titleHighlightColor components.</td>
<td>#FFCB05 (Sisense yellow)</td>
</tr>
</tbody>
</table>
Icon Resource Files

The following icon resources are used by the mobile app for push notifications and for the store listing in iOS. All sizes are mandatory. Please use the exact file names specified below.

<table>
<thead>
<tr>
<th>Platform</th>
<th>Size (Width x Height)</th>
<th>File Name</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>iOS</td>
<td>40x40</td>
<td>ios_icon_40x40.png</td>
<td></td>
</tr>
<tr>
<td>iOS</td>
<td>80x80</td>
<td>ios_icon_80x80.png</td>
<td></td>
</tr>
<tr>
<td>iOS</td>
<td>60x60</td>
<td>ios_icon_60x60.png</td>
<td></td>
</tr>
<tr>
<td>iOS</td>
<td>120x120</td>
<td>ios_icon_120x120.png</td>
<td></td>
</tr>
<tr>
<td>iOS</td>
<td>180x180</td>
<td>ios_icon_180x180.png</td>
<td></td>
</tr>
<tr>
<td>iOS</td>
<td>76x76</td>
<td>ios_icon_76x76.png</td>
<td></td>
</tr>
<tr>
<td>iOS</td>
<td>152x152</td>
<td>ios_icon_152x152.png</td>
<td></td>
</tr>
<tr>
<td>iOS</td>
<td>167x167</td>
<td>ios_icon_167x167.png</td>
<td></td>
</tr>
<tr>
<td>iOS</td>
<td>29x29</td>
<td>ios_icon_29x29.png</td>
<td></td>
</tr>
<tr>
<td>iOS</td>
<td>58x58</td>
<td>ios_icon_58x58.png</td>
<td></td>
</tr>
<tr>
<td>iOS</td>
<td>87x87</td>
<td>ios_icon_87x87.png</td>
<td></td>
</tr>
<tr>
<td>iOS</td>
<td>1024x1024</td>
<td>ios_icon_1024x1024.png</td>
<td>App Store Icon</td>
</tr>
<tr>
<td>Platform</td>
<td>Size (Width x Height)</td>
<td>File Name</td>
<td>Note</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------</td>
<td>----------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>Android</td>
<td>36x36</td>
<td>android_icon_36x36.png</td>
<td></td>
</tr>
<tr>
<td>Android</td>
<td>48x48</td>
<td>android_icon_48x48.png</td>
<td></td>
</tr>
<tr>
<td>Android</td>
<td>72x72</td>
<td>android_icon_72x72.png</td>
<td></td>
</tr>
<tr>
<td>Android</td>
<td>96x96</td>
<td>android_icon_96x96.png</td>
<td></td>
</tr>
<tr>
<td>Android</td>
<td>144x144</td>
<td>android_icon_144x144.png</td>
<td></td>
</tr>
<tr>
<td>Android</td>
<td>192x192</td>
<td>android_icon_192x192.png</td>
<td></td>
</tr>
<tr>
<td>Android</td>
<td>24x24</td>
<td>android_notice_24x24.png</td>
<td>Notifications Icon</td>
</tr>
<tr>
<td>Android</td>
<td>36x36</td>
<td>android_notice_36x36.png</td>
<td>Notifications Icon</td>
</tr>
<tr>
<td>Android</td>
<td>48x48</td>
<td>android_notice_48x48.png</td>
<td>Notifications Icon</td>
</tr>
<tr>
<td>Android</td>
<td>72x72</td>
<td>android_notice_72x72.png</td>
<td>Notifications Icon</td>
</tr>
<tr>
<td>Android</td>
<td>96x96</td>
<td>android_notice_96x96.png</td>
<td>Notifications Icon</td>
</tr>
</tbody>
</table>
Splash Screen

Splash screen images are displayed on the full screen until the app is loaded when the app is cold-launched. All sizes are mandatory.
<table>
<thead>
<tr>
<th>Platform</th>
<th>Size (Width x Height)</th>
<th>File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>iOS</td>
<td>320x480</td>
<td>ios_splash_320x480.png</td>
</tr>
<tr>
<td>Platform</td>
<td>Size (Width x Height)</td>
<td>File Name</td>
</tr>
<tr>
<td>----------</td>
<td>----------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>iOS</td>
<td>640x960</td>
<td>ios_splash_640x960.png</td>
</tr>
<tr>
<td>iOS</td>
<td>768x1024</td>
<td>ios_splash_768x1024.png</td>
</tr>
<tr>
<td>iOS</td>
<td>1536x2048</td>
<td>ios_splash_1536x2048.png</td>
</tr>
<tr>
<td>iOS</td>
<td>1024x768</td>
<td>ios_splash_1024x768.png</td>
</tr>
<tr>
<td>iOS</td>
<td>2048x1536</td>
<td>ios_splash_2048x1536.png</td>
</tr>
<tr>
<td>iOS</td>
<td>2208x1242</td>
<td>ios_splash_2208x1242.png</td>
</tr>
<tr>
<td>iOS</td>
<td>1242x2208</td>
<td>ios_splash_1242x2208.png</td>
</tr>
<tr>
<td>iOS</td>
<td>750x1334</td>
<td>ios_splash_750x1334.png</td>
</tr>
<tr>
<td>iOS</td>
<td>640x1136</td>
<td>ios_splash_640x1136.png</td>
</tr>
<tr>
<td>iOS</td>
<td>1125x2436</td>
<td>ios_splash_1125x2436.png</td>
</tr>
<tr>
<td>iOS</td>
<td>2436x1125</td>
<td>ios_splash_2436x1125.png</td>
</tr>
<tr>
<td>Android</td>
<td>320x240</td>
<td>android_splash_320x240.png</td>
</tr>
<tr>
<td>Android</td>
<td>480x320</td>
<td>android_splash_480x320.png</td>
</tr>
<tr>
<td>Android</td>
<td>800x480</td>
<td>android_splash_800x480.png</td>
</tr>
<tr>
<td>Android</td>
<td>1280x720</td>
<td>android_splash_1280x720.png</td>
</tr>
<tr>
<td>Android</td>
<td>1600x960</td>
<td>android_splash_1600x960.png</td>
</tr>
<tr>
<td>Android</td>
<td>1920x1280</td>
<td>android_splash_1920x1280.png</td>
</tr>
<tr>
<td>Android</td>
<td>240x320</td>
<td>android_splash_240x320.png</td>
</tr>
<tr>
<td>Android</td>
<td>320x480</td>
<td>android_splash_320x480.png</td>
</tr>
<tr>
<td>Android</td>
<td>480x800</td>
<td>android_splash_480x800.png</td>
</tr>
<tr>
<td>Android</td>
<td>720x1280</td>
<td>android_splash_720x1280.png</td>
</tr>
<tr>
<td>Android</td>
<td>960x1600</td>
<td>android_splash_960x1600.png</td>
</tr>
<tr>
<td>Platform</td>
<td>Size (Width x Height)</td>
<td>File Name</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>Android</td>
<td>1280x1920</td>
<td>android_splash_1280x1960.png</td>
</tr>
</tbody>
</table>
## Additional File Resources

<table>
<thead>
<tr>
<th>Platform</th>
<th>Purpose</th>
<th>Filename</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>iOS+Android</td>
<td>Logo image that is displayed after the Splash screen.</td>
<td>login_screen_logo.svg</td>
<td>Must be a vector file (svg) to fit all resolutions.</td>
</tr>
<tr>
<td>iOS+Android</td>
<td>Login screen background.</td>
<td>login-bg-image.svg</td>
<td>Must be a vector file (svg) to fit all resolutions.</td>
</tr>
<tr>
<td>iOS</td>
<td>Firebase project configuration file for pulse alerts (push notifications)</td>
<td>GoogleService-Info.plist</td>
<td>Unique for each app. Generate the file on <a href="https://firebase.google.com">https://firebase.google.com</a> If not provided, notifications from Pulse alerts will not work.</td>
</tr>
<tr>
<td>Android</td>
<td>Firebase project configuration file for pulse alerts (push notifications)</td>
<td>google-services.json</td>
<td>Unique for each app. Generate the file on <a href="https://firebase.google.com">https://firebase.google.com</a> If not provided, push notifications from Pulse alerts will not work.</td>
</tr>
</tbody>
</table>
Publishing Your App

Before Sisense provides you with your rebranded mobile app, Sisense signs the app for you. To sign your app, you must provide Sisense with the relevant certificates as described below. Once the app has been signed, Sisense returns the signed app to you and you can publish the app in Google Play or the Apple App store. Apple and Android require that all apps be digitally signed with a certificate.
Android

For instructions on how to sign your rebranded Android-based mobile app, see Sign your app.
iOS

Go to [https://developer.apple.com](https://developer.apple.com) and create an identifier (App ID) for the new app.

Under **App Services**, select **Push Notifications**.

If you do not have an iOS production certificate, create one on [https://developer.apple.com](https://developer.apple.com).

Download the certificate to the Mac that has your private key, and add it to the key chain (double click).

Open Keychain Access on your Mac, and under **My Certificates** locate the iPhone Distribution Certificate you have created for the app.

Select **File > Export Item**. Make sure the file format is p12.

In developer.apple.com, create a Distribution Provisioning Profile for your app, using the same certificate you created above.

Download the certificate to your Mac.

Double click the certificate. This will start the process to install it.

For more information, see [Code Signing](https://developer.apple.com).
Sisense Mobile Upgrades

Sisense regularly releases updates for Sisense Mobile. Sisense will periodically (about once a quarter) notify when an upgraded rebranded version is available, and you can choose whether to upload the app to your store or not.
Security at Sisense

Sisense provides a variety of ways in which you can control which data is exposed to users and secure your connections to Sisense.

The following topics describe how you can configure and maintain the security of your data in Sisense:

- Security Overview
- Security Bulletin
- Data Security
- SSL
- Optional Security Features
Security Bulletin

Security vulnerability in NodeJS library used by Sisense 7.2.1 and 7.2.1 Service Pack 1
We have recently discovered a low severity security vulnerability in the NodeJS libraries used in version 7.2.1. The issue is resolved in 7.2.1 service pack 3 (7.2.1.13003), released Dec. 6th, 2018. This issue is not relevant for any other versions.

We have found that a third party package used by Sisense, NodeJS, included malicious code that could be used to steal Bitcoin from Bitpay and Copay wallets. The malicious code targeted developers at the Copay company that had a very specific development environment setup: **running the payload in any other environment has no effect.** This specific type of targeting means that, ultimately, most applications are not affected even if the malicious module is mistakenly deployed. **Specifically to Sisense, this vulnerability could not be used to access any Sisense data, and was not used by Sisense code, but antivirus programs are identifying it as malware.**

For more information about this vulnerability, see [here](#).

Remediation:

If you are using one of the builds mentioned above and are concerned about the AntiVirus alerts we recommend you upgrade to the latest 7.2.1 version, available for download [here](#).

Update regarding a security vulnerability resolved in Sisense V6.7.1 on May 23, 2018
Starting from Sisense V6.7, there was an issue with data security rules. This rare issue occurred only when a single dashboard contained at least two widgets from entirely disconnected tables - meaning that there was no relationship path leading from one table to the other.

If a filter was applied on one of the tables, data security rules of the second table were disabled. The widget showing data from the second table would display all data, without data security rule restrictions.

Note that the issue did not occur when a dashboard only used widgets related to a single data security field. The issue also did not occur if there was any relationship path between the tables of the data security fields. The relationship path did not have to be a direct relationship between the tables, it could be a relationship that went through other tables.

The issue is resolved in Sisense V6.7.1 (build 6.7.1.17004), and in Sisense V7.1.2.

**Update regarding Meltdown (CVE-2017-5754) and Spectre (CVE-2017-5753 and CVE-2017-5715)**

The Meltdown/Spectre vulnerabilities are the recent vulnerabilities found in Intel’s processors. These are vulnerabilities in all of the Windows operating systems. Sisense provides an application and does not provide the server hardware or the Windows OS. These are provided by customers. As such, it’s the customers’ responsibility to secure their OS.

We recommend customers follow all security recommendations of the vendors who provided the customers OS. The current Windows recommendation is to install the latest available security updates for the OS. Sisense has run extensive performance tests on the OS patch (Windows patch) and found a minimal impact on performance.

For more information about these vulnerabilities and how to negate them, see: [Meltdown and Spectre](#): A high-level description of the vulnerabilities.
Project Zero: Describes the theory behind the vulnerabilities by those who discovered them.

Intel Official Announcement: Describes Intel’s response to the vulnerabilities.

AMD Processor Security: Describes AMD’s response to the vulnerabilities.
Sisense Security Architecture

Sisense is built around a robust and flexible security architecture that is both comprehensive and intuitive. This architecture has been designed to ensure security processes are enforced while scaling to enterprise deployments of Sisense. This includes the ability to secure dashboards and data as well as implement custom security requirements that suit your organization. This section provides a general overview of the main security features.

Security is based around three levels associated with sets of security features. The diagram below maps this security architecture on a system, data and object level.

### System Level
- User Management
- SSO
- Active Directory
- REST API

### Object Level
- ElastiCube Access
- Dashboard Access

### Data Level
- Row Based Security
- Row Level Defaults
System Level Security

System-level security encompasses security features for role-based settings and integration options. This includes user and server management, connection to an active directory, Single Sign-On (SSO) implementation, and use of the security REST API.

User Management
You can assign five primary roles to Sisense users:
- Administrators
- Data Admin
- Data Designer
- Designer
- Viewer

These roles can be defined on either a user or group level to determine sharing, access and security. To learn more about user roles, click here.

ElastiCube Server Access Rights
You can assign access rights to different ElastiCube servers for individual users, groups or to everyone. These settings allow the management of different environments such as a testing and production server, or servers for specific projects or departments. See also ElastiCube Server and Data Model Security.

Note: Following improvements to ElastiCube security in Sisense V7.0 and later, ElastiCubes created prior to Sisense V7.0 are accessible to everyone by default, unless you have defined the ElastiCube's access rights. For ElastiCubes created after Sisense V7.0, the default access is only for the ElastiCube owner. When the ElastiCube is ready, it should be shared with the relevant people, or with "Everyone" if that is the desired access policy.

Active Directory
Connect existing users and groups from your organization’s Active Directory to define security and sharing properties and reduce deployment time. This
removes ‘password fatigue’ as users can rely on existing credentials while organizational policies around security credentials such as updates can be enforced. See also Integrating Active Directory.

**Single Sign-On (SSO)**

SSO facilitates seamless integration between Sisense and other systems in your organization while offering standardization of authentication policies across your organization. This can improve user productivity by avoiding password fatigue and reduce support overhead. See also Configuring SSO.

**REST API**

The Security REST API provide access to parameters to integrate and automate restrictions and access control based on existing settings and standards. Specify access rights and security to dashboards, data models and data. Manage users via the API to create, edit and assign new users or groups. Click here to visit the API documentation site.
Object Level Security

Object security defines access rights for different users and groups to various components within Sisense. The two main objects are dashboards and data models.

**Dashboards**
You can share dashboards on either a user or group level. The sharing options include the configuration of access rights for all users as well as whether users defined as designers may edit the dashboard. The sharing options also include subscription settings that define which users and groups will receive email reports. See also [Sharing Dashboards](#).

**Data Models**
You can define access rights to different Data Models on a user or group level. This enables flexibility to create models for specific user or group needs while offering strict access control. See also [Sharing ElastiCubes](#).
Data Level Security

Data access must provide data to people only to the extent that they need to complete their jobs. Data Level Security provides the necessary control to enforce varying degrees of data visibility and access to support the separation of duties. A single dashboard can be shared with many users, but each viewer sees only data relevant to their needs. This reduces both development time and provides for security.

Security on the Row Level
You can grant user and group permissions to specific rows in the data. For each data model, you can apply multiple rules to enforce granular access control.

Row Level Defaults
Control which data is accessible for users or groups that do not have explicit security rules. For example, enable new employees to access a restricted data set until they are added to relevant groups. You can set defaults to include everything, nothing or view based on a security rule.

See more in Data Security below.
Security Levels

Sisense provides two levels of security:

Data Model Security
Data Security
Data Model Security

What is Data Model Security?
Sisense enables you to define access rights to control which users can access which models, whether they are creating new dashboards or trying to access shared dashboards.

Data Model Security – Use Case Example
You may have a data model named Marketing and only want the CEO and Marketing team to have access to it. You can grant rights only to them using Data Model Security, thus denying anyone else access.

How Does Data Model Security Work?
You can define which users/user groups have access to a data model. By default, only the data model's creator, Administrator and Data Administrator can access a data model. Once you start assigning users/user groups access rights to a data model, then those users/user groups will have access to the model. The type of access is determined their role and what access you assign to the user.
When a user attempts to access a dashboard using a direct link and that dashboard is based on a data model to which that user does not have access rights, a security message is displayed.
Data Security

Data security in Sisense can be divided into two types, data encryption and data access. Data communication is related to how data is secured by Sisense while being imported into Sisense and written on your server’s disk. When data is imported into Sisense or when you connect directly to a data source, the protocol used depends on the protocols supported by the data source. Sisense supports importing data over SSL, if the source supports it. Sisense supports SSL for data retrieval, for example, when viewing data in dashboards. Configuration data, such as account credentials and authorization profiles, are encrypted prior to being written to the disk. The encryption technology used by Sisense includes:

1. SHA-256
2. TripleDES
3. AES-256

For data at rest, Sisense supports OS based disk encryption, Windows file system encryption. For more information, click here.

The second type of data security is data access. This type of data security refers to who can access your data after it’s imported into Sisense and displayed in a dashboard.

What is Data Access Security?

Sisense enables you to define Data Security Rules that control which users can access which portions of the raw data in a data model, down to row granularity. For example:

- Each widget only shows the data permitted by the Data Security Rules that apply, including totals, averages and so on.
- The data browsers used while building dashboards and widgets only show the data permitted by the Data Security Rules that apply.
You can define a single dashboard that automatically displays different results for each user (or user group), based on the rows that user is permitted to see.

**Use Case Example**

A Sales Order table has a column representing the salesperson that closed a deal.

You created a quarterly performance dashboard for your salespeople, but want each of them to see only their own data.

You do not want any of them to be exposed to data that represents the performance of others.

**How Does Data Access Security Work?**

The Sisense security model is designed to work in a 'Grant access' model, and not in a 'Deny access' model.

By default, when applying a data security rule, access is blocked for everyone, and the best practice is to leave 'Everyone else' set to 'Nothing', while granting groups and users with access to specific data values. The model accumulates grants, meaning that the most permissive combination wins. So if both a group and one of its members have conflicting rules, the permissive combination wins.

Each data model contains tables and each table contains fields.

A Data Security Rule defines that a specific user can only see any data of an entire row of a table, if a specific field in that row has a specific value(s).

For example, in a Sales widget a salesperson (for example, Dan) will only see the sales amounts from the rows of a Sales model whose Salesperson field contains the value Dan (rows 1 and 4).

**Sales Table**

<table>
<thead>
<tr>
<th></th>
<th>Salesperson</th>
<th>Product</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dan</td>
<td>HD-TV</td>
<td>$100</td>
</tr>
<tr>
<td>2</td>
<td>Matthew</td>
<td>TV</td>
<td>$300</td>
</tr>
<tr>
<td>3</td>
<td>Amber</td>
<td>Media Center</td>
<td>$700</td>
</tr>
<tr>
<td>4</td>
<td>Dan</td>
<td>Player</td>
<td>$200</td>
</tr>
</tbody>
</table>
Dan will not see any part of a row in the data model that does not contain the value 'Dan' in the Salesperson field, nor will any amounts from this row be included in totals.

**Note:** The entire row of data is not seen by the relevant user even when the field to which the rule applies does not appear in the widget.

If a widget that shows the amount spent per product is shared with Dan, then he will only see HD-TV and Player and the sales total will be 300$. 

|   | Matthew | Air Conditioner | $600 |
Defining Data Access Security for a Data Model

Each Data Security Rule applies to a specific field in a data model and to specific user(s)/user group(s). It enables you to define the values that must be contained in a specific field to enable that entire row of data to be available to a user. If you want to exclude or hide data from certain users, you can define these rules in the Sisense REST API. For more information, see Restricting Data Access for Data Models.

To access Data Security:
Click Admin and select the Data Sources tab in the menu. For the relevant data model, select and click Data Security.
If no data security rules have yet been defined for this data model, then the following message is displayed:
Click **Add Field** to display a list of the fields in this data model.

![Sample Ecommerce](image-url)

Note: Row-based data security rules are may caused reduced performance when applied to floats.

Select a field. For example, **Brand**. The following window is then displayed in which you can define rules.

Note: You cannot select date type fields.

The left side of this table enables you to define which users/user groups can access this data. Click + **Add Restriction** and start typing into the **Restricted User/Groups** field to get a drop-down list. Add as many users/user groups, as necessary.

The right side of this window enables you to define which values the
specified users/user groups are permitted to see. Start typing into the Values field to get a list. Multiple values can be selected. The value of numeric type fields must be typed into this field, as no auto-complete option appears for numeric type fields. Alternatively, you can start typing in one of these values (in both text and numeric type fields):

**Everything**: To specify that the selected users/user groups can see this data no matter what the value is in this field.

**Nothing**: To specify that the selected users/user groups cannot see this data no matter what the value is in this field.

For example, you can define that the following Users/User Groups must have the following values in the Product Category column to enable them to see their data row in a widget.

<table>
<thead>
<tr>
<th>#</th>
<th>User/User Group</th>
<th>Product Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Management</td>
<td>Everything</td>
</tr>
<tr>
<td>2</td>
<td>Bob</td>
<td>Apple Mac Desktops</td>
</tr>
<tr>
<td>3</td>
<td>Don</td>
<td>Calculators, Camera Flashes</td>
</tr>
<tr>
<td>4</td>
<td>Everyone else</td>
<td>Nothing</td>
</tr>
</tbody>
</table>

This means that management can see the data of all Product Categories, Don can only see the data of Calculators and Camera Flashes, Bob can only see the data of Apple Mac Desktops, and Everyone else won’t see anything.

**How Does Data Level Security Work for Tables with Relationships?**

Tables in a data model may have a relationship between them. As described above, each widget only shows any data of an entire row of a table, if a specific field in that row has a specific value.
In addition, a widget may further restrict the data shown to a specific user when a rule is defined for a table that has a relationship to a table that has a field in the widget.

This means that a widget only shows the data permitted by the combined Data Security Rules assigned to all the tables that have any field in the widget. As described above, the entire data row is restricted even when the field to which the rule applies does not appear in the widget. The entire row of data is also restricted even when the field of the relationship between the two tables does not appear in the widget.

**Use Case Example – Expanding Upon the Example Above**

The Sales table has a column that has a relationship with a Deal Contacts table that holds the contacts that were involved in each deal.
You created a Deal Contacts widget for your salespersons.
As described in the example above, the Sales table has a Data Security Rule that maps each user to their matching field value, so that each sales person only sees their own data.
Even though the Deal Contacts table does not have any Data Security Rules defined for it, the Deal Contacts widget only enables each sales person to see the contacts associated with their own sales, because of the Data Security Rule assigned to the Sales table.
Restricting Data Access for Data Models

In Sisense, all users who have access to your data models can see all of the data. If you define any data security rules, the default behavior is inclusionary, which means you define what values of a field a user is allowed to see. For example, you can allow each Sales rep to see transactions for their own customers, and prevent Sales reps from seeing the transactions of other customers. In this case, you define a row-based data security rule for each Sales rep, based on the customer IDs of each customer.

In some cases, you may want to allow all of your users to see your data except for a specific user or group of users. In this case, exclusionary rules are preferred. For example, let’s assume that your company has thousands of customers, and your policy is that all Sales reps can see information for most of your customers, not only for their own customers. You may have certain customers whose data is sensitive and should only be accessed by certain authorized Sales reps. In this case, it’s easier to manage a definition that allows access to everything, except the few restricted customers, than to manage a list of the thousands of customers whose data is freely available to all Sales reps.

To restrict certain rows of data to a specific user or group of users, through the Sisense REST API, you can change the default data security behavior to exclusionary, which allows you to hide or restrict access to data to certain users. You can combine inclusionary rules with exclusionary rules. In case the rules conflict, the exclusionary rules take precedence.

**Note:** When multiple data security rules exist for a specific field-user or field-group combination, the “inclusionary” rules will be combined with “OR” logic between them. “Exclusionary” rules will be combined with “AND” logic between them.

**To change the data security behavior for a rule:**
In Sisense, click Admin, and then Rest API, followed by REST API Reference.

In Version 0.9 of the REST API, select **elasticubes**, then POST `/elasticubes/datasecurity`.

In the body of your call, update the value of "**exclusionary**" to **true**. Click **Try it out** and then **Execute**.
Setting Up SSL

SSL (Secure Sockets Layer) is a security protocol that encrypts the server-client data channel. In Sisense, SSL secures the link between the Sisense server and the clients’ browsers (where dashboards are created and viewed). With SSL in place, you can access Sisense via an HTTPS secure connection, which is password protected. This is recommended when you want users to be able to access dashboards from outside the organization.

Keep in mind, configuring SSL on your Sisense server and in your IT environment should be performed by an IT Specialist or Web Administrator.

**Note:** If you are using a reverse proxy or load balancer, you should ensure that the communication between the proxy and Sisense is HTTPS.

The procedure below describes how to set up SSL in Sisense from Sisense V7.2 and later. If you are upgrading from an earlier version to Sisense V7.2 or later, see [Configuring SSL after Upgrading](#) at the end of this topic.

The tutorial below describes how to set up a simple SSL connection in Sisense.

**To configure your Sisense SSL settings:**

In your browser, open the Configuration Manager located at http://localhost:3030.

Toggle the **Enable SSL** switch to **Enabled**.

Upload or enter your SSL certificate. Sisense supports two types of certificates, PFX and CERT files. Typically, these files are provided by a third party provider.

**When using SSL, the certificate must include the root certificate and any intermediate certificates.**

**PFX:** PFX files contain the public key file (SSL certificate file) and the associated private key file in a single file. If you select **PFX**, drag the PFX file into the browser or click **Browse** and navigate to the file. In **PFX Password**, enter the password you received after your PFX was generated.
**CERT-Key:** A .cert file is the public key, which is used to verify client authentication requests. It is what is received by an HTTP client from a server in the SSL handshake. If you select **CERT-KEY**, two boxes are displayed **CERT File** and **Key File**. In **CERT File**, drag the .cert file into the browser or click **Browse** and navigate to the file. In **Key File**, drag the key file into the browser or click **Browse** and navigate to the file. The .key file is the private key to the certificate.

Alternatively, if the files are not provided, and you have received a coded certificate and key, you can enter these values in the **SSL Certificate** and **SSL Key** fields.

In **Port**, enter the port to be used when accessing Sisense. By default, this is 8081, however, if you are implementing SSL, typically the port is set to 443.

After you have finished defining these settings, in the Configuration Manager, click **Save**.

If you have any problems, see the following tutorial on troubleshooting SSL in Sisense:
Self-Signed Certificates

If you are using a self-signed certificate, you may experience problems connecting to data sources. This is caused by the self-signed certificate being rejected. To fix this, you need to give the NODE_EXTRA_CA_CERTS environment variable a file path to a file containing your certificate.

To prevent self-signed certificates from being automatically rejected:
On your Sisense server, right-click on This PC (Or My Computer in older versions of Windows) and select Properties.
On the left side, click Advanced system settings.
In System Properties, click Environment Variables.
In **Environment Variables**, under the **System variables** area, click **New**.

In the **New System Variable** dialog box, in **Variable name**, enter **NODE_EXTRA_CA_CERTS**.

In **Variable value**, enter the address of your .ca file.

OR

Select **Browse file** and navigate to the .ca file.

Click **OK**.
Restart your computer. This should resolve connection problems caused by sign certificates.
Configuring SSL After Upgrading

Earlier versions of Sisense used IIS, which meant that part of your SSL configuration was stored in the IIS Manager. As NodeJS is now used as the application server, your Sisense SSL bindings must be removed when you are upgrading to Sisense V7.2 and later from an earlier version for SSL to continue to work.

To configure SSL after upgrading:

On the Sisense Server, open the IIS Manager.
Remove from the IIS Manager the bindings to port 443.

Reset IIS.
In your browser, open the System Configuration at http://localhost:3030/.
Verify that Enable SSL is enabled.
Upload or enter your SSL certificate. Sisense supports two types of certificates, PFX and CERT files. Typically, these files are provided by a third party provider.
When using SSL, the certificate must include the root certificate and any intermediate certificates.
PFX: PFX files contain the public key file (SSL certificate file) and the
associated private key file in a single file. If you select PFX, drag the PFX file into the browser or click Browse and navigate to the file. In PFX Password, enter the password you received after your PFX was generated. CERT-Key: A .cert file is the public key, which is used to verify client authentication requests. It is what is received by an HTTP client from a server in the SSL handshake. If you select CERT-KEY, two boxes are displayed CERT File and Key File. In CERT File, drag the .cert file into the browser or click Browse and navigate to the file. In Key File, drag the key file into the browser or click Browse and navigate to the file. The .key file is the private key to the certificate.

Alternatively, if the files are not provided, and you have received a coded certificate and key, you can enter these values in the SSL Certificate and SSL Key fields.

In Port, enter the port to be used when accessing Sisense. By default, this is 8081, however, if you are implementing SSL, typically the port is set to 443. Click Save.

Click Yes to restart the Sisense services.
Optional Security Hardening for Sisense Web Pages

This topic provides additional security options that can be applied for hardening the security of Sisense web pages for your needs.
Cookie Security

By default, Sisense web pages have cookies that contain a secureFlag. This flag instructs the browser that the cookie should only be returned to the Sisense Web Application over encrypted connections (HTTPS).
Account Lockout Thresholds

To prevent brute-force attacks, you can configure account lockout thresholds.
For more information, see Account Lockout Thresholds.
Strict Transport Security

HTTP Strict Transport Security (HSTS) is a method for preventing any communications from being sent over HTTP to the specified domain and allows only communication over HTTPS. This is useful for preventing man-in-the-middle attacks or users with invalid certificates from accessing your dashboards. This is automatically applied when you enable SSL for the Sisense Web Application.
Allowed Domains for Embedded Dashboards

If you are embedding a dashboard on your website, you can control who can access the website by adding allowed domains to a whitelist. Allowed Domains enable you to limit where your embedded dashboards can be viewed, even if someone takes the embed code from your page. When you add a domain to the whitelist, Sisense includes the domain in the X-Frame Options header of the dashboard web page.

For example:

```html
<add name="X-Frame-Options" value="ALLOW-FROM https://dashboardurl.com" />
```

The header is not included by default. You can enable it from the Configuration Manager located at http://localhost:3030.

To add your domain to a white list:
In the Admin page, select Settings. Under Security Settings, enter your domain and the port.

Click Add.
Click **Save** at the bottom of the page.
Multi-Node Deployments
Overview

Sisense provides flexible design choices for supporting high availability (HA) and scalability for your Sisense deployments. Typically, you want to implement multi-node deployments when you want to optimize performance or build in redundancy. Before implementing a multi-node environment, you can ask yourself the following questions:

Do you want to improve dashboard load time?
Are your ElastiCubes taking a long build time to build?
Do you need to support a lot of concurrent users?
Do you have a lot of ElastiCubes?
Do you need high availability for your system?

If you answered yes to any of these questions, you may want to consider scaling out Sisense.

The basic Sisense architecture looks like the diagram below where Sisense is installed on a single machine. Your users connect to your machine and you import or connect to your data sources from the same machine. This machine supports the Sisense web application, your data models, and all of your users.
One way to scale out Sisense is to improve your hardware with more storage, CPU, and memory; however, this can get quite expensive and doesn’t provide any redundancy.
Another way to scale out Sisense is to add more machines, thus improving performance and building in redundancy into the system as shown in the diagram below.
While Sisense is fully-functional in a single node environment, a multi-node deployment is necessary for scalability to support large amounts of concurrent users and redundancy in case of a failure of one of the Sisense components. Replicating each of these components provides redundancy and fault tolerance against the failure of any single component. The replicated components are combined into nodes. There are three types of nodes, a build node, query nodes, and an application node. These nodes and their components are described below.
Nodes
Build Node

The build node is responsible for building ElastiCubes and distributing the build to query nodes via the Sisense Orchestrator Service. The Sisense Orchestrator Service is an automated service that you configure on the build node to synchronize and distribute built ElastiCubes to the query nodes. For more information, see Distributing ElastiCube Builds to Query Nodes. Build nodes include an ElastiCube Server, application database, Sisense plugins, and the Sisense Orchestrator Service.

The build node is not replicated as its failure only prevents building new ElastiCubes, not issuing queries from Sisense.

Note: Sisense plugins must be located on the build node.
Query Nodes

Query nodes are responsible for supporting queries from Sisense dashboards on the application layer. ElastiCube models are distributed by the build node to the query nodes. The query nodes’ ElastiCube models are then combined into ElastiCube Sets to increase redundancy by separating the web and ElastiCube servers across multiple query nodes. If a build node is distributing a build to one ElastiCube server, Sisense automatically directs any queries to the other ElastiCubes in the ElastiCube Set.

The query node can be configured as a single application stack where each node hosts Sisense, ElastiCubes, and an application database. In this configuration, if the machine hosting the components fails, the whole query node will fail. Queries will then be redirected to the next available query node. Another option is to host each component of the query node separately in a distributed application stack. In this configuration, if a component of the query node fails, the rest of the query node is not affected. For an example of a single application stack, see Scenario 1 and for an example of a distributed application stack, see Scenario 2.
Application Nodes

The application node supports your Sisense application. This is the interface you see when you log into Sisense, including the Model Editor, dashboards, etc. In some models, this resides on the same node as the query node.
Components

Sisense has many components that reside on each of the nodes. These components are highlighted in the diagram below. Some of these components are responsible for supporting Sisense, such as the application database, configuration database, and message broker. Other components, such as a load balancer, Multi-Node Deployment Wwizard, and ElastiCube Sets are responsible for supporting high availability in Sisense. Each of these components is described in more detail below.
Application Database

The application database is installed with Sisense and supports Sisense. The application database is a central repository for Sisense metadata including user information, permissions, data sources, dashboards, jobs, etc.

If the application database fails, the Sisense web application will fail.

To achieve redundancy and high availability of the application database, a minimum of three nodes is required.
Message Broker

The message broker is a component of Sisense and is responsible for the communication of events between various Sisense services across your Sisense configuration. Sisense availability and functionality are heavily dependent on the broker service. It should be replicated with at least two nodes to ensure that the services can continue to communicate with each other in case the message broker fails.
Configuration Database

The configuration database provides a single representation for the cluster regarding the topology, configurations, and state.

To achieve redundancy and high availability, the configuration service should be replicated with at least three nodes to ensure that your configuration is up-to-date across your entire deployment.
Load Balancer

To support a multi-node configuration, you must handle load balancing on your side prior to directing traffic to one of your Sisense nodes. Load balancing spreads requests across multiple query nodes according to an algorithm you define and the current status of the query node.

When implementing ElastiCube Sets, Sisense’s query nodes operate in active-active mode. This means that each of the query nodes is active and can handle requests when the node is not building and its components are available. For example, traffic could be spread 50-50 across two web servers and if a component fails, a load balancer should redirect traffic to the other available web server.
ElastiCube Sets

Sisense ElastiCube Sets are collections of identical ElastiCube models that allow you to query running ElastiCubes within the ElastiCube Set while other ElastiCubes are building.
Components that Cannot be Replicated

Several Sisense components are deployed as single services that cannot be replicated and do not have redundancy:

- ElastiCube web management
- Plug-ins
- Scheduled reporting jobs
- Sisense Orchestrator
- Build nodes
Multi-Node Configuration

To support more concurrent users and queries, and build in redundancy into your deployment, you must provide additional machines and configure the orchestration between the various Sisense nodes and their services. Sisense makes it easy to implement a multi-node deployment with the Multi-Node Deployment Wizard. This wizard automates the configuration of your nodes. If you want to implement high availability, after you have run the wizard, you configure the Sisense Orchestrator service that manages the distribution of ElastiCubes across multiple machines.

The following pages describes the multi-node deployments Sisense supports, how to configure them in the Multi-Node Deployment wizard, and how to configure the distribution of ElastiCubes with the Sisense Orchestrator.
Next Steps

Supported Deployments
Setting Up Multi-Node Deployments
Distributing ElastiCubes to Query Nodes
Setting Up ElastiCube Sets
Securing the Message Broker’s Communication
Supported Deployments

Previous Steps

Overview

Sisense supports four different multi-node deployment models:

- **Model 1**: 1 Application Node, 1 Query Node/Build Node
- **Model 2**: 1 Application Node, 1 Query Node, 1 Build Node
- **Model 3**: 2 Application Nodes/Query Nodes, and 1 Build Node
- **Model 4**: 2 Application Nodes, 2 Query Nodes, and 1 Build Node

Which model should you choose is determined by these three parameters:

- **Redundancy**: Which components require redundancy?
- **Concurrency**: How many users do you need to support?
- **Resources**: How many servers can you add and maintain to your configuration?

The differences between each model are the amount of redundancy they provide, how much scale they support, and how many resources you need to implement them.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Model 1</th>
<th>Model 2</th>
<th>Model 3</th>
<th>Model 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enhanced Security</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>High availability &amp; Redundancy</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Scale Performance</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Dedicated Hardware</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Scale as You Grow</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

The advantages of each model are described in more detail below.
In Model 1, Sisense is vertically scaled out, which means that the application node is separated from the query and build node. This deployment provides the following benefits:

**Enhanced Security:** When the application node and query/build nodes are separated, Viewers who interact with just the application node, are less vulnerable.

**Scale Performance:** Vertical scaling enables more users to query data while the build node can continue to build ElastiCube models.

**Dedicated Servers per Task:** Support greater concurrency by dedicating one machine to supporting Sisense, while another machine supports queries and building ElastiCube models.

For more information, see:
- [Model 1 Architecture, Services, Ports](#)
- [Deploying Model 1](#)
In Model 2, Sisense is vertically scaled out with the application, query, and build node separated. This deployment provides the following benefits:

**Enhanced Security**: When the application node, query, and build nodes are separated, Viewers who interact with just the application node, are less vulnerable

**Scale Performance**: Vertical scaling enables more users to query data while the build node can continue to build ElastiCube models

**Dedicated Servers per Task**: Support greater concurrency by dedicating one machine to supporting Sisense, while another machine supports queries and building ElastiCube models

For more information, see:

- [Model 2 Architecture, Services, Ports](#)
- [Deploying Model 2](#)
Model 3

In Model 3, Sisense is horizontally scaled out with the application/query nodes replicated. This deployment provides the following benefits:

**High Availability**: Multiple application/query nodes ensures that if a single node fails, performance will not be affected and your Viewers won’t experience any downtime

**Scale Performance**: Horizontal scaling enables more users to query data and build models at the same time

**Dedicated Servers per Task**: Support greater concurrency by dedicating one machine to supporting Sisense, while another machine supports queries and building ElastiCube models

**Scales as You Grow**: Add additional application/query nodes as needed

For more information, see:

- Model 3 Architecture, Ports, Services
- Deploying Model 3
In Model 4, Sisense is horizontally scaled out with the application and query nodes replicated. This deployment provides the following benefits:

**High Availability:** Multiple application and query nodes that have been replicated mean that if a single-server fails, your Viewers will not affect performance or lead to downtime

**Scale Performance:** Horizontal scaling enables more users to query data and build models at the same time

**Dedicated Servers per Task:** Support greater concurrency and redundancy by dedicating multiple machines to supporting Sisense, while other machines support queries and building ElastiCube models

**Scales as You Grow:** Add additional application/query nodes as needed

For more information, see:
- Model 4 Architecture, Services, Ports
- Deploying Model 4

**Note:** If you want to implement a topology not supported by Sisense, contact your CSM.
Next Steps

- Installing the Multi-Node Deployment Wizard
- Distributing ElastiCubes to Query Nodes
- Setting Up ElastiCube Sets
- Securing the Message Broker's Communication
Installing the Multi-Node Deployment Wizard

**Previous Steps**

[Supported Deployments](#)

With a multi-node deployment, you can increase the reliability of your configuration further by adding additional query and build nodes, and an application node.

To simplify the process of setting up a Multi-Node environment, Sisense provide a Multi-Node Deployment Wizard that automates most of the setup process for you.

**Note:** If you are expanding a single-node deployment to a multi-node deployment, see [Expanding Single-Node Deployments to Multi-Node Deployments](#).

**Prerequisites**

Windows 8, 10, and Windows Server 2012, 2016
Local System Admin account where the wizard is to be installed and deployed

**To install the Multi-Node Deployment Wizard:**

Download the [Sisense High Availability Multi-Node Deployment Wizard](#) to your build node. When saving the file, make sure there are no spaces in the file name. This can happen if you download the file to the same directory multiple times, for example, MultiServerDeploymentWizard.zip(1).

Extract the contents of the zip file.

Download the [Cygwin script](#) and [Sisense-install-ansible.bsh](#) and save the files to the directory `...MultiServerDeploymentWizard/resources/` located in the directory where you extracted the contents of the Multi-Node Deployment Wizard zip file.

Run `install-cygwin.ps1` with Powershell. This process can take between 20-30 minutes.
During the process, Cygwin is downloaded and installed. Once the installation is complete, you are prompted to press any button to close the script.
The Multi-Node Deployment Wizard is installed.
Next Steps

**Model 1:** 1 Application Node, 1 Query Node/Build Node
**Model 2:** 1 Application Node, 1 Query Node, 1 Build Node
**Model 3:** 2 Application/Query Nodes and 1 Build Node
**Model 4:** 2 Application Nodes, 2 Query Nodes, and 1 Build Node
Model 1: 1 Application Node, 1 Query Node/Build Node

Previous
Installing the Multi-Node Deployment Wizard
In Model 1, your application node is on a separate server from the query/build node. The benefit of Model 1 over a single node is that by separating the nodes, your data can be secured on a different node than your application node. In addition, you can utilize your hardware to support more different tasks; one node for queries and builds, and one node for supporting the Sisense application. The disadvantage of Model 1 compared to other models is that high availability is not supported and workloads for the application and query nodes cannot be scaled. If your query/build node fails, the Sisense application will not be usable as the data models cannot be queried.

To learn how to implement this model in the Deployment Wizard, see Deploying Model 1.

**Limitations**
- Maximum of four concurrent builds on one build node
- Maximum of 40 ElastiCubes per query node

<table>
<thead>
<tr>
<th>Components</th>
<th>High Availability</th>
<th>Load Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>App Node</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>


# Components

<table>
<thead>
<tr>
<th>Components</th>
<th>High Availability</th>
<th>Load Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Node</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Build Node</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Application Database</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

## Services Mapping

Sisense utilizes multiple communication ports. Some of the ports are used for communication between Sisense modules, and others are used for communication with remote locations.

The table below describes the ports used for communication between the Sisense modules that need to be opened in a firewall to support this architecture.

**Note:** "N/A" means the ports are not relevant (internal process).

<table>
<thead>
<tr>
<th>Service Type</th>
<th>Service</th>
<th>Ports</th>
<th>Query/Build</th>
<th>Application</th>
<th>Directions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infra</td>
<td>Sisense.Oxygen</td>
<td>31112</td>
<td>Yes</td>
<td>Yes</td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.Orchestrator</td>
<td>445, 137-139</td>
<td>N/A</td>
<td>N/A</td>
<td>Inbound</td>
</tr>
<tr>
<td>Infra-Core</td>
<td>Sisense.Broker</td>
<td>5672, 15672, 25672, 4369</td>
<td></td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.Discovery</td>
<td>2181, 2888, 3888</td>
<td></td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.Repository</td>
<td>27018</td>
<td></td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td>Infra-Logs</td>
<td>Sisense.Collector</td>
<td>N/A</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Sisense.Shipper</td>
<td>5052</td>
<td>Yes</td>
<td>Yes</td>
<td>Outbound</td>
</tr>
<tr>
<td>Service Type</td>
<td>Service</td>
<td>Ports</td>
<td>Query/Build</td>
<td>Application</td>
<td>Directions</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------------</td>
<td>-------------</td>
<td>------------</td>
</tr>
<tr>
<td>Infra Connectors</td>
<td>Sisense.HouseKeeper</td>
<td>N/A</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Sisense.CLRCConnectors Container</td>
<td>8090, 8098</td>
<td>Yes</td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.JVMConnectors Container</td>
<td>8095, 9000-9010</td>
<td>Yes</td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td>App - ECM</td>
<td>Sisense.ECMServer</td>
<td>15000-15050</td>
<td>Yes</td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.ECMLogs</td>
<td>N/A</td>
<td>Yes</td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td>App - ElasticCube</td>
<td>Sisense.ECMS</td>
<td>811,812</td>
<td>Yes</td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>IIS</td>
<td>14996</td>
<td>Yes</td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.Galaxy</td>
<td>15000-15050</td>
<td>Yes</td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.Gateway</td>
<td>15000-15050</td>
<td>Yes</td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.Configuration</td>
<td>15000-15050</td>
<td>Yes</td>
<td>Yes</td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.Identity</td>
<td>15000-15050</td>
<td>Yes</td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.Jobs</td>
<td>N/A</td>
<td>Yes</td>
<td></td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Sisense.Plugins</td>
<td>15000-15050</td>
<td>Yes</td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td>Service Type</td>
<td>Service</td>
<td>Ports</td>
<td>Query/Build</td>
<td>Application</td>
<td>Directions</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------</td>
<td>----------------</td>
<td>-------------</td>
<td>-------------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td>Sisense.SPE</td>
<td>15000-15050</td>
<td>Yes</td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.StorageManager</td>
<td>15000-15050</td>
<td>Yes</td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.Usage</td>
<td>N/A</td>
<td>Yes</td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.Pivot2</td>
<td>15000-15050</td>
<td>Yes</td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.QueryProxy</td>
<td>N/A</td>
<td>Yes</td>
<td></td>
<td>Inbound</td>
</tr>
</tbody>
</table>
Deploying Model 1

Note: If you are expanding a single-node deployment to a multi-node deployment, see Expanding Single-Node Deployments to Multi-Node Deployments.

After you have installed the Multi-Node Deployment Wizard, you can log in with your Sisense credentials and begin configuring your nodes to implement a Model 1 deployment. In the Multi-Node Deployment Wizard, you define which servers act as build/query nodes and application nodes. The Multi-Node Deployment Wizard then configures the nodes for you.

In addition, you can define advanced settings for determining where to host the Sisense infrastructure, which includes the following components:

**Sisense Application Database:** The application database contains metadata for ElastiCubes, dashboards, and users. You must have an odd number of machines to host the application database as Sisense combines them into a replica set. The goal of a replica set is to ensure that each application database has the same metadata about ElastiCubes, dashboards, and filters. If one database fails on one server, the next available database can replace it.

**Sisense Message Broker:** The message broker is responsible for communicating events across Sisense components. As the message broker is not part of a replica set you can have it located on an even number of machines.

**Sisense Configuration Database:** The Sisense configuration database is responsible for storing your configuration data. This ensures that your Sisense configuration is up-to-date across your entire configuration. You must have an odd number of machines to host the configuration database as Sisense combines them into a replica set. The goal of a replica set is to ensure that each configuration database has the same data. If one database fails on one server, the next available database can replace it.

To deploy Model 1:
After installing the Deployment Wizard, in your High Availability Configuration directory, run SisenseHAWizard.exe. If you are setting up
your deployment behind a firewall, see Multi-Node Deployment Offline Installation.
Enter your Sisense login credentials and click **Sign In**.

If you are installing a multi-node deployment, select **Install New Deployment**, and click **Next**. If you are modifying an existing deployment, select **Upgrade Existing Deployment** and click **Next**. If you are
Click **Download Agent Installation** to install the Sisense Agent. This agent must be installed on all the Sisense servers that are part of your deployment. It allows Sisense to communicate with each of the other machines in your deployment. To install the agent, right-click on the file and...
select **Run with PowerShell.** Click **Next.**

**Agent Installation**

Before you continue with this deployment wizard, you must install a Sisense agent on every server you want to configure. Download the agent installation file and install it on each of your servers. Once you have installed all of the agents, click "Next" to continue.

[Download Agent Installation]
Select the version of Sisense you are using. You can view your version of the Sisense Server by clicking 🕵️‍♀️ in the top right corner of Sisense.
Select **Model 1** and click **Next**.

In the Multi-Node Deployment Wizard, enter your Windows server credentials. These credentials will be used to connect to each of the servers in your environment, so the user name and password should be the same for each server and the user must have Admin privileges for each machine.
**Note:** The password should not contain any special characters.

In **App**, enter the IP address of your application server. In **Query/Build**, enter the IP address of your query/build server. (Optional) Click **Advanced** to configure the support the Sisense infrastructure.
Select **Long Index Edition** to install the Long Index version of Sisense. Long Index determines how Sisense transforms your data when loading it into an ElastiCube. For more information, see [Long Index](#).

In the Infra Configuration area, select additional servers to define where Sisense hosts additional copies of the Sisense Message Broker, Sisense configuration database, and Sisense application database.

In the Application Database Configuration area, select **Read Strategy** to define the read preferences of each member of your application database's replica set. The default is **primaryPreferred**, which means that Sisense reads are sent to the primary, but if it is unavailable, the reads come from secondary members instead. If your application databases are geographically spread out, you may want to select **Nearest** where reads are performed on the nearest available node, disregarding if it is a primary or secondary member. The nearest node is determined by sending pings to all members and measuring the response time.

**OR**

Select **Use External Application Database** if you host your Sisense application database externally on a separate machine. In the
Connection String area, create a replica set for supporting your application database. This ensures that each application database has the same metadata about ElastiCubes, dashboards, and filters. Modify the value of **Connection String** to:

```
mongodb://AppUser:we6jBUYGoOrh0K6l+XpTmA==@localhost:27018,XX.XX.XX.XX:27018,XX.XX.XX.XX:27018/?replicaSet=<replicasetname>&readPreference=primaryPreferred
```

replacing ‘@localhost:27018/’ with the IP addresses of each of your nodes in the replica set separated by a comma and replacing `<replicasetname>` with the name of your replica set.

Click **Install**. Sisense will automatically complete the setup of your Multi-Node environment. Close the Deployment Wizard once the setup is complete. After you have created your environment, the next step is to define how builds are distributed across all your servers.
Next Steps

Model 2: 1 Application Node, 1 Query Node, 1 Build Node
Model 2: 1 Application Node, 1 Query Node, 1 Build Node

Previous

Installing the Multi-Node Deployment Wizard
Model 2 Architecture, Services, Ports

In this model, your application, query, and build node are located on separate servers. The benefit of Model 2 over a single node is that by separating each node, your data can be secured on a different node than your application node. In addition, you can utilize your hardware to support different tasks; one node for queries, one node for builds, and one node for supporting the Sisense application.

The disadvantage of Model 1 compared to other models is that high availability is not supported and workloads for the application and query nodes cannot be scaled. If your query node fails, the Sisense application will not be usable as the data models cannot be queried.

To learn how to implement this model in the Deployment Wizard, see Deploying Model 2.

Limitations
- Maximum of four concurrent builds on one build node
- Maximum of 40 ElastiCubes per query node
- No component replication

<table>
<thead>
<tr>
<th>Components</th>
<th>High Availability</th>
<th>Load Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>App Node</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Components</td>
<td>High Availability</td>
<td>Load Distribution</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Query Node</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Build Node</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Application Database</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

**Services Mapping**

Sisense utilizes multiple communication ports. Some of the ports are used for communication between Sisense modules, and others are used for communication with remote locations.

The table below describes the ports used for communication between the Sisense modules that need to be opened in a firewall for this architecture.

Note: "N/A" means the ports are not relevant (internal process).

<table>
<thead>
<tr>
<th>Service Type</th>
<th>Service</th>
<th>Ports</th>
<th>Build</th>
<th>Query</th>
<th>Application</th>
<th>Directions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infra</td>
<td>Sisense.Oxygen</td>
<td>31112</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.Orchestrator</td>
<td>445, 137-139</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td>Infra-Core</td>
<td>Sisense.Broker</td>
<td>5672,15672,25672,4369</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.Discovery</td>
<td>2181,2888,3888</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.Repository</td>
<td>27018</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Inbound</td>
</tr>
<tr>
<td>Infra - Logs</td>
<td>Sisense.Collector</td>
<td>5052</td>
<td>N/A</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------</td>
<td>------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td></td>
<td>Sisense.Shipper</td>
<td></td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Sisense.HouseKeeper</td>
<td>N/A</td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Infra - Connectors</td>
<td>Sisense.CLRConnectors</td>
<td>8090, 8098</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sisense.JVMConnectors</td>
<td>8095</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>App - ECM</td>
<td>Sisense.ECMServer</td>
<td>15000-15050</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sisense.ECMLogs</td>
<td>N/A</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>App - Elasticube</td>
<td>Sisense.ECMS</td>
<td>811,812</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>App</td>
<td>IIS</td>
<td>14996</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sisense.Galaxy</td>
<td>15000-15050</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sisense.Gateway</td>
<td>15000-15050</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sisense.Configuration</td>
<td>15000-15050</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sisense.Identity</td>
<td>15000-15050</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sisense.Jobs</td>
<td>N/A</td>
<td>Yes</td>
<td>N/A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>-------</td>
<td>-----</td>
<td>--------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sisense.Plugins</td>
<td>15000-15050</td>
<td>Yes</td>
<td>Inbound</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sisense.SPE</td>
<td>15000-15050</td>
<td>Yes</td>
<td>Inbound</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sisense.StorageManager</td>
<td>15000-15050</td>
<td>Yes</td>
<td>Inbound</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sisense.Usage</td>
<td>N/A</td>
<td>Yes</td>
<td>Inbound</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sisense.Pivot2</td>
<td>15000-15050</td>
<td>Yes</td>
<td>Inbound</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sisense.QueryProxy</td>
<td>N/A</td>
<td>Yes</td>
<td>Inbound</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Deploying Model 2

**Note:** If you are expanding a single-node deployment to a multi-node deployment, see [Expanding Single-Node Deployments to Multi-Node Deployments](#).

After you have installed the Multi-Node Deployment Wizard, you can log in with your Sisense credentials and begin configuring your nodes to implement a Model 2 deployment. In the Multi-Node Deployment Wizard, you define which servers act as build/query nodes and application nodes. The Deployment Wizard then configures the nodes for you.

In addition, you can define advanced settings for determining where to host the Sisense infrastructure, which includes the following components:

- **Sisense Application Database:** The application database contains metadata for ElastiCubes, dashboards, and users. You must have an odd number of machines to host the application database as Sisense combines them into a replica set. The goal of a replica set is to ensure that each application database has the same metadata about ElastiCubes, dashboards, and filters. If one database fails on one server, the next available database can replace it.

- **Sisense Message Broker:** The message broker is responsible for communicating events across Sisense components. As the message broker is not part of a replica set you can have it located on an even number of machines.

- **Sisense Configuration Database:** The Sisense configuration database is responsible for storing your configuration data. This ensures that your Sisense configuration is up-to-date across your entire configuration. You must have an odd number of machines to host the configuration database as Sisense combines them into a replica set. The goal of a replica set is to ensure that each configuration database has the same data. If one database fails on one server, the next available database can replace it.

To deploy Model 2:

After [installing the Deployment Wizard](#), in your High Availability Configuration directory, run `SisenseHAWizard.exe`. If you are setting up
your deployment behind a firewall, see Multi-Node Deployment Offline Installation. Enter your Sisense login credentials and click **Sign In**.

**Sisense Deployment**

Welcome!
Enter your Sisense credentials to begin

**Sign In**

- **Email**
- **Password**

**Behind a firewall?**

If you are installing a multi-node deployment, select **Install New Deployment**, and click **Next**. If you are modifying an existing deployment, select **Upgrade Existing Deployment** and click **Next**. If you are
Click **Download Agent Installation** to install the Sisense Agent. This agent must be installed on all the Sisense servers that are part of your deployment. It allows Sisense to communicate with each of the other machines in your deployment. To install the agent, right-click on the file and...
select **Run with PowerShell**. Click **Next**.

Agent Installation

Before you continue with this deployment wizard, you must install a Sisense agent on every server you want to configure. Download the agent installation file and install it on each of your servers. Once you have installed all of the agents, click “Next” to continue.

Download Agent Installation
Select the version of Sisense you are using. You can view your version of the Sisense Server by clicking 🔄 in the top right corner of Sisense.

Sisense Deployment

Select the version to install:

- Latest

Contact Support  Back  Next
Select **Model 2** and click **Next**.

In the Multi-Node Deployment Wizard, enter your Windows server credentials. These credentials will be used to connect to each of the servers in your environment, so the user name and password should be the same for each server and the user must have Admin privileges.
Note: The password should not contain any special characters.

In **App**, enter the IP address of your application server.
In **Query**, enter the IP address of your query server.
In **Build**, enter the IP address of your build server.
(Optional) Click **Advanced** to configure the support the Sisense infrastructure.
Select **Long Index Edition** to install the Long Index version of Sisense. Long Index determines how Sisense transforms your data when loading it into an ElastiCube. For more information, see *Long Index.*

In the Infra Configuration area, select additional servers to define where Sisense hosts additional copies of the Sisense Message Broker, Sisense configuration database, and Sisense application database.

In the Application Database Configuration area, select **Read Strategy** to define the read preferences of each member of your application database's replica set. The default is **primaryPreferred**, which means that Sisense reads are sent to the primary, but if it is unavailable, the reads come from secondary members instead. If your application databases are geographically spread out, you may want to select **Nearest** where reads are performed on the nearest available node, disregarding if it is a primary or secondary member. The nearest node is determined by sending pings to all members and measuring the response time.

OR

Select **Use External Application Database** if you host your Sisense application database externally on a separate machine. In the
Connection String area, create a replica set for supporting your application database. This ensures that each application database has the same metadata about ElastiCubes, dashboards, and filters. Modify the value of **Connection String** to:

```
mongodb://AppUser:we6jBUYGoOrh0K6l+XpTmA==@localhost:27018,XX.XX.XX.XX:27018,XX.XX.XX.XX:27018/?replicaSet=<replicasetname>&readPreference=primaryPreferred
```

replacing ‘@localhost:27018/’ with the IP addresses of each of your nodes in the replica set separated by a comma and replacing `<replicasetname>` with the name of your replica set.

Click **Install**. Sisense will automatically complete the setup of your Multi-Node environment. Close the Deployment Wizard once the setup is complete. After you have created your environment, the next step is to **define how builds are distributed across all your servers**.
Next Steps

Model 3: 2 Application/Query Nodes and 1 Build Node
Model 3: 2 Application/Query Nodes and 1 Build Node

Previous
Installing the Multi-Node Deployment Wizard
Model 3 Architecture, Services, Ports

In Model 3, your application node and query nodes reside on the same servers and are separate from the build node. You can add additional query and application nodes as needed. The benefit of this architecture over Model 4 is that it requires fewer machines, while still providing redundancy in case one of the nodes fails.

The disadvantage of Model 3 compared to Model 4 is that fewer concurrent users are supported and larger queries may burden the system.

To learn how to implement this model in the Deployment Wizard, see Deploying Model 3.

**Limitations**
- Maximum of four concurrent builds on one build node
- Maximum of 40 ElastiCubes per query node

<table>
<thead>
<tr>
<th>Components</th>
<th>High Availability</th>
<th>Load Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web/Application Server</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Query Service</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MongoDB</td>
<td>Yes</td>
<td>N/A</td>
</tr>
<tr>
<td>Components</td>
<td>High Availability</td>
<td>Load Distribution</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Build</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Plugins</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Jobs</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ECM Server (Web)</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ECM Server Build Logs</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Orchestrator</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

**Services Mapping**

Sisense utilizes multiple communication ports. Some of the ports are used for communication between Sisense modules, and others are used for communication with remote locations.

The table below describes the ports used for communication between the Sisense modules that need to be opened in a firewall for this architecture.

*Note:* "N/A" means the ports are not relevant (internal process).

<table>
<thead>
<tr>
<th>Service Type</th>
<th>Service</th>
<th>Ports</th>
<th>Build</th>
<th>App/Query 1</th>
<th>App/Query 2</th>
<th>Directions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infra</td>
<td>Sisense.Oxygen</td>
<td>31112</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.Orchestrator</td>
<td>445, 137-139</td>
<td>Yes</td>
<td>See comments</td>
<td>See comments</td>
<td>Inbound</td>
</tr>
<tr>
<td>Infra-Core</td>
<td>Sisense.Broker</td>
<td>5672,15672,25672,4369</td>
<td>See comments</td>
<td>See comments</td>
<td>See comments</td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.Discovery</td>
<td>2181,288</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Inbound</td>
</tr>
<tr>
<td>Service Type</td>
<td>Service</td>
<td>Ports</td>
<td>Build</td>
<td>App/Query 1</td>
<td>App/Query 2</td>
<td>Directions</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------</td>
<td>---------------------</td>
<td>-------</td>
<td>-------------</td>
<td>-------------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8,3888</td>
<td></td>
<td></td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.Repository</td>
<td>27018</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Inbound</td>
</tr>
<tr>
<td>Infra - Logs</td>
<td>Sisense.Collector</td>
<td>N/A</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Sisense.Shipper</td>
<td>5052</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Outbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.HouseKeeper</td>
<td>N/A</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
</tr>
<tr>
<td>Infra - Connectors</td>
<td>Sisense.CLRConnectorsContainer</td>
<td>8090, 8098</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.JVMConnectorsContainer</td>
<td>8095 9000-9010</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Inbound</td>
</tr>
<tr>
<td>App - ECM</td>
<td>Sisense.ECMServer</td>
<td>15000-15050</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.ECMLogs</td>
<td>N/A</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td>App - Elasticcube</td>
<td>Sisense.ECMS</td>
<td>811,812</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>IIS</td>
<td>14996</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.Galaxy</td>
<td>15000-15050</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.Gateway</td>
<td>15000-</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td>Service Type</td>
<td>Service</td>
<td>Ports</td>
<td>Build</td>
<td>App/Query 1</td>
<td>App/Query 2</td>
<td>Directions</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------</td>
<td>-------------</td>
<td>-------</td>
<td>-------------</td>
<td>-------------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td>Sisense.Configuration</td>
<td>15000-15050</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.Identity</td>
<td>15000-15050</td>
<td></td>
<td></td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.Jobs</td>
<td>N/A</td>
<td>Yes</td>
<td></td>
<td></td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Sisense.Plugins</td>
<td>15000-15050</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.SPE</td>
<td>15000-15050</td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.StorageManager</td>
<td>15000-15050</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.Usage</td>
<td>15000-15050</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.Pivot2</td>
<td>15000-15050</td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Inbound</td>
</tr>
<tr>
<td></td>
<td>Sisense.QueryProxy</td>
<td>N/A</td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Inbound</td>
</tr>
</tbody>
</table>
Deploying Model 3

**Note:** If you are expanding a single-node deployment to a multi-node deployment, see [Expanding Single-Node Deployments to Multi-Node Deployments](#).

After you have installed the Multi-Node Deployment Wizard, you can log in with your Sisense credentials and begin configuring your nodes to implement a Model 3 deployment. In the Multi-Node Deployment Wizard, you define which servers act as build/query nodes and application nodes. The Deployment Wizard then configures the nodes for you.

In addition, you can define advanced settings for determining where to host the Sisense infrastructure, which includes the following components:

**Sisense Application Database:** The application database contains metadata for ElastiCubes, dashboards, and users. You must have an odd number of machines to host the application database as Sisense combines them into a replica set. The goal of a replica set is to ensure that each application database has the same metadata about ElastiCubes, dashboards, and filters. If one database fails on one server, the next available database can replace it.

**Sisense Message Broker:** The message broker is responsible for communicating events across Sisense components. As the message broker is not part of a replica set you can have it located on an even number of machines.

**Sisense Configuration Database:** The Sisense configuration database is responsible for storing your configuration data. This ensures that your Sisense configuration is up-to-date across your entire configuration. You must have an odd number of machines to host the configuration database as Sisense combines them into a replica set. The goal of a replica set is to ensure that each configuration database has the same data. If one database fails on one server, the next available database can replace it.

**To deploy Model 3:**

After [installing the Deployment Wizard](#), in your High Availability Configuration directory, run `SisenseHAWizard.exe`. If you are setting up
your deployment behind a firewall, see Multi-Node Deployment Offline Installation.
Enter your Sisense login credentials and click **Sign In**.

If you are installing a multi-node deployment, select **Install New Deployment**, and click **Next**. If you are modifying an existing deployment, select **Upgrade Existing Deployment** and click **Next**. If you are
Click **Download Agent Installation** to install the Sisense Agent. This agent must be installed on all the Sisense servers that are part of your deployment. It allows Sisense to communicate with each of the other machines in your deployment. To install the agent, right-click on the file and
select **Run with PowerShell.** Click **Next.**

**Agent Installation**

Before you continue with this deployment wizard, you must install a Sisense agent on every server you want to configure. Download the agent installation file and install it on each of your servers. Once you have installed all of the agents, click "Next" to continue.

**Download Agent Installation**
Select the version of Sisense you are using. You can view your version of the Sisense Server by clicking 🕵️‍♂️ in the top right corner of Sisense.

Sisense Deployment

Select the version to install:

Latest
Select **Model 3** and click **Next**.

In the Multi-Node Deployment Wizard, enter your Windows server credentials. These credentials will be used to connect to each of the servers in your environment, so the user name and password should be the same for each server and the user must have Admin privileges.
Note: The password should not contain any special characters.

In **App**, enter the IP address of your application server.
In **Query**, enter the IP address of your query server.
In **Build**, enter the IP address of your build server.
(Optional) Click **Advanced** to configure the support for the Sisense infrastructure.
Select **Long Index Edition** to install the Long Index version of Sisense. Long Index determines how Sisense transforms your data when loading it into an ElastiCube. For more information, see [Long Index](#).

In the Infra Configuration area, select additional servers to define where Sisense hosts additional copies of the Sisense Message Broker, Sisense configuration database, and Sisense application database.

In the Application Database Configuration area, select **Read Strategy** to define the read preferences of each member of your application database's replica set. The default is **primaryPreferred**, which means that Sisense reads are sent to the primary, but if it is unavailable, the reads come from secondary members instead. If your application databases are geographically spread out, you may want to select **Nearest** where reads are performed on the nearest available node, disregarding if it is a primary or secondary member. The nearest node is determined by sending pings to all members and measuring the response time.

OR

Select **Use External Application Database** if you host your Sisense application database externally on a separate machine. In the
Connection String area, create a replica set for supporting your application database. This ensures that each application database has the same metadata about ElastiCubes, dashboards, and filters. Modify the value of **Connection String** to:

```
mongodb://AppUser:we6jBUYGoOrh0K6l+XpTmA==@localhost:27018,XX.XX.XX.XX:27018,XX.XX.XX.XX:27018/?replicaSet=<replicasetname>&readPreference=primaryPreferred
```

replacing ‘@localhost:27018/’ with the IP addresses of each of your nodes in the replica set separated by a comma and replacing `<replicasetname>` with the name of your replica set.

Click **Install**. Sisense will automatically complete the setup of your Multi-Node environment. Close the Deployment Wizard once the setup is complete. After you have created your environment, the next step is to **define how builds are distributed across all your servers**.
Next Steps

Model 4: 2 Application Nodes, 2 Query Nodes, and 1 Build Node
Model 4: 2 Application Nodes, 2 Query Nodes, and 1 Build Node

Previous
Installing the Multi-Node Deployment Wizard
Model 4 Architecture, Services, Ports

In Model 4, your application node is on a separate server from the query and build node. The application and query nodes are replicated for redundancy. You can add additional query and application nodes as needed. The benefit of Model 4 over a single node is that by you can support more concurrent users, and if one of the application or query nodes fails, the other can take on the load to ensure availability of the Sisense application.

To learn how to implement this model in the Deployment Wizard, see Deploying Model 4.

Limitations
Maximum of four concurrent builds on one build node
Maximum of 40 ElastiCubes per query node

<table>
<thead>
<tr>
<th>Components</th>
<th>High Availability</th>
<th>Load Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web/Application Server</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Query Service</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MongoDB</td>
<td>Yes</td>
<td>N/A</td>
</tr>
<tr>
<td>Build</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Components</td>
<td>High Availability</td>
<td>Load Distribution</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Plugins</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Jobs</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ECM Server (Web)</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ECM Server Build Logs</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Orchestrator</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

**Services Mapping**

Sisense utilizes multiple communication ports. Some of the ports are used for communication between Sisense modules, and others are used for communication with remote locations.

The table below describes the ports used for communication between the Sisense modules that need to be opened in a firewall for this architecture.

*Note:* "N/A" means the ports are not relevant (internal process).

<table>
<thead>
<tr>
<th>Service Type</th>
<th>Service</th>
<th>Ports</th>
<th>Build Quer y 1</th>
<th>Quer y 2</th>
<th>Application 1</th>
<th>Application 2</th>
<th>Comments</th>
<th>Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infra-</td>
<td>Sisense.Oxygen</td>
<td>31112</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Inbound</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sisense.Orchestrator</td>
<td>445, 137-139</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td>Inbound</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sisense.Discovery</td>
<td>2181,2888,3888</td>
<td>See Com</td>
<td>See Com</td>
<td>See Com</td>
<td>See Com</td>
<td>Inbound</td>
<td></td>
</tr>
<tr>
<td>Service Type</td>
<td>Service Type</td>
<td>Ports</td>
<td>Build</td>
<td>Query 1</td>
<td>Query 2</td>
<td>Application 1</td>
<td>Application 2</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------</td>
<td>-------</td>
<td>---------</td>
<td>---------</td>
<td>---------------</td>
<td>---------------</td>
<td>---------</td>
</tr>
<tr>
<td>Infra - Logs</td>
<td>Sisense.Collector</td>
<td>N/A</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
</tr>
<tr>
<td>Infra - Logs</td>
<td>Sisense.Shipper</td>
<td>5052</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Outbound</td>
</tr>
<tr>
<td>Infra - Logs</td>
<td>Sisense.HouseKeeper</td>
<td>N/A</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
</tr>
<tr>
<td>Infra - Connectors</td>
<td>Sisense.CLRCConnectorsContainer</td>
<td>8090, 8098</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td>Infra - Connectors</td>
<td>Sisense.JVMConnectorsContainer</td>
<td>8095-9000-9010</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td>App - ECM</td>
<td>Sisense.ECMServer</td>
<td>15000-15050</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td>App - ECM</td>
<td>Sisense.ECMLogs</td>
<td>N/A</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td>App - Elasti cube</td>
<td>Sisense.ECMS</td>
<td>811,812</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Inbound</td>
</tr>
<tr>
<td>Service Type</td>
<td>Service</td>
<td>Ports</td>
<td>Build Query 1</td>
<td>Build Query 2</td>
<td>Application 1</td>
<td>Application 2</td>
<td>Comments</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------</td>
<td>---------------</td>
<td>---------------</td>
<td>---------------</td>
<td>---------------</td>
<td>---------------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>App</td>
<td>IIS</td>
<td>14996</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Inbound</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sisense.Galaxy</td>
<td>15000-15050</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Inbound</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sisense.Gateway</td>
<td>15000-15050</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Inbound</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sisense.Configuration</td>
<td>15000-15050</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Inbound</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sisense.Identity</td>
<td>15000-15050</td>
<td></td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Inbound</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sisense.Jobs</td>
<td>N/A</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sisense.Plugins</td>
<td>15000-15050</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td>Inbound</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sisense.SPE</td>
<td>15000-15050</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Inbound</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sisense.Storage Manager</td>
<td>15000-15050</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td>Inbound</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sisense.Usage</td>
<td>N/A</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td>Inbound</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sisense.Pivot2</td>
<td>15000-15050</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Inbound</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sisense.QueryPr</td>
<td>N/A</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Inbound</td>
<td></td>
</tr>
<tr>
<td>Service Type</td>
<td>Service</td>
<td>Ports</td>
<td>Build Query 1</td>
<td>Build Query 2</td>
<td>Application 1</td>
<td>Application 2</td>
<td>Comments</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>---------</td>
<td>-------</td>
<td>---------------</td>
<td>---------------</td>
<td>---------------</td>
<td>---------------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>oxy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>nd</td>
<td></td>
</tr>
</tbody>
</table>
Deploying Model 4

Note: If you are expanding a single-node deployment to a multi-node deployment, see Expanding Single-Node Deployments to Multi-Node Deployments.

After you have installed the Multi-Node Deployment Wizard, you can log in with your Sisense credentials and begin configuring your nodes to implement a Model 4 deployment. In the Multi-Node Deployment Wizard, you define which servers act as build/query nodes and application nodes. The Deployment Wizard then configures the nodes for you.

In addition, you can define advanced settings for determining where to host the Sisense infrastructure, which includes the following components:

**Sisense Application Database:** The application database contains metadata for ElastiCubes, dashboards, and users. You must have an odd number of machines to host the application database as Sisense combines them into a replica set. The goal of a replica set is to ensure that each application database has the same metadata about ElastiCubes, dashboards, and filters. If one database fails on one server, the next available database can replace it.

**Sisense Message Broker:** The message broker is responsible for communicating events across Sisense components. As the message broker is not part of a replica set you can have it located on an even number of machines.

**Sisense Configuration Database:** The Sisense configuration database is responsible for storing your configuration data. This ensures that your Sisense configuration is up-to-date across your entire configuration. You must have an odd number of machines to host the configuration database as Sisense combines them into a replica set. The goal of a replica set is to ensure that each configuration database has the same data. If one database fails on one server, the next available database can replace it.

**To deploy Model 4:**
After installing the Deployment Wizard, in your High Availability Configuration directory, run SisenseHAWizard.exe. If you are setting up
your deployment behind a firewall, see Multi-Node Deployment Offline Installation. Enter your Sisense login credentials and click **Sign In**.

If you are installing a multi-node deployment, select **Install New Deployment**, and click **Next**. If you are modifying an existing deployment, select **Upgrade Existing Deployment** and click **Next**. If you are
Click **Download Agent Installation** to install the Sisense Agent. This agent must be installed on all the Sisense servers that are part of your deployment. It allows Sisense to communicate with each of the other machines in your deployment. To install the agent, right-click on the file and
select Run with PowerShell. Click Next.

Agent Installation

Before you continue with this deployment wizard, you must install a Sisense agent on every server you want to configure. Download the agent installation file and install it on each of your servers. Once you have installed all of the agents, click “Next” to continue.
Select the version of Sisense you are using. You can view your version of the Sisense Server by clicking in the top right corner of Sisense.
Select **Model 4** and click **Next**.

In the Multi-Node Deployment Wizard, enter your Windows server credentials. These credentials will be used to connect to each of the servers in your environment, so the user name and password should be the same for each server and the user must have Admin privileges.
**Note:** The password should not contain any special characters.

In **App**, enter the IP address of your application server.  
In **Query**, enter the IP address of your query server.  
In **Build**, enter the IP address of your build server.  
(Optional) Click **Advanced** to configure the support the Sisense infrastructure.
Select **Long Index Edition** to install the Long Index version of Sisense. Long Index determines how Sisense transforms your data when loading it into an ElastiCube. For more information, see [Long Index](#).

In the Infra Configuration area, select additional servers to define where Sisense hosts additional copies of the Sisense Message Broker, Sisense configuration database, and Sisense application database.

In the Application Database Configuration area, select **Read Strategy** to define the read preferences of each member of your application database's replica set. The default is **primaryPreferred**, which means that Sisense reads are sent to the primary, but if it is unavailable, the reads come from secondary members instead. If your application databases are geographically spread out, you may want to select **Nearest** where reads are performed on the nearest available node, disregarding if it is a primary or secondary member. The nearest node is determined by sending pings to all members and measuring the response time.

OR

Select **Use External Application Database** if you host your Sisense application database externally on a separate machine. In the
Connection String area, create a replica set for supporting your application database. This ensures that each application database has the same metadata about ElastiCubes, dashboards, and filters. Modify the value of **Connection String** to:

```
mongodb://AppUser:we6jBUYGoOrh0K6l+XpTmA==@localhost:27018,XX.XX.XX:27018,XX.XX.XX:27018/?replicaSet=<replicasetname>&readPreference=primaryPreferred
```

replacing ‘@localhost:27018/’ with the IP addresses of each of your nodes in the replica set separated by a comma and replacing `<replicasetname>` with the name of your replica set.

Click **Install**. Sisense will automatically complete the setup of your Multi-Node environment. Close the Deployment Wizard once the setup is complete. After you have created your environment, the next step is to **define how builds are distributed across all your servers**.
Next Steps

Distributing ElastiCube Builds to Query Nodes
Distributing ElastiCube Builds to Query Nodes

The Sisense Orchestrator Service is an automated service that distributes builds and balances queries across an ElastiCube Set. The Sisense Orchestrator Service is installed during the default installation of Sisense. To distribute ElastiCube builds across all query nodes, you must configure the Sisense Orchestrator Service. When an ElastiCube is building, or receiving a build from the build node, the Sisense Orchestrator Service redirects requests to another available ElastiCube in the set.

After you have set up all of your nodes, you can verify that each query node is working by stopping the Sisense.ECMS service. When this service is stopped, queries should be redirected to the other query nodes in your configuration.

Note: After configuring the Sisense Orchestrator Service, the directory \C:\ProgramData\SiSense\PrismServer\ElastiCubeData\ and the relevant ElastiCube folders within it must be shared with permissions for ‘Everyone’ so Sisense can build or update the ElastiCubes in those folders. Keep in mind that each of your nodes must have sufficient disk space to support the build node.

In addition, the user of the build node must have Administrator access to the service Sisense.Orchestrator on each of the query nodes, so the build node can write to them. You can set this in the properties of the Sisense.Orchstrator...
The Sisense Orchestrator Service is defined in a JSON file called config.json in the Sisense.Orchestration Config folder (C:\ProgramData\Sisense\Sisense.Orchestrator\config.json) of your Sisense Installation folder of your build node.

To distribute your ElastiCube builds through the Sisense Orchestrator Service, you define and save the config.json file. After saving the file, the Sisense Orchestrator Service automatically begins to build ElastiCubes in your ElastiCube Set. You can review the progress of the build in the C:\ProgramData\Sisense\application-logs\orchestrator-service folder in a log file whose file name you define in the JSON file.

The default config.json file has a basic structure with no values.
To learn more about how to configure the config.json file, see the following tutorial:

Sisense supports a wide range of objects and values and several examples and their descriptions are provided in the table below:

<table>
<thead>
<tr>
<th>Key</th>
<th>Example</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>general</td>
<td>{</td>
<td>The General object contains two objects, logLevel and emails that define what types of logs Sisense generates and where to send them to.</td>
</tr>
<tr>
<td></td>
<td>&quot;general&quot;:{</td>
<td>}</td>
</tr>
<tr>
<td></td>
<td>&quot;logLevel&quot;:&quot;INFO&quot;,</td>
<td>}</td>
</tr>
<tr>
<td></td>
<td>&quot;emails&quot;:[]</td>
<td>}</td>
</tr>
<tr>
<td></td>
<td>&quot;email&quot;:&quot;<a href="mailto:john.test@sisense.com">john.test@sisense.com</a>&quot;,</td>
<td>}</td>
</tr>
<tr>
<td></td>
<td>&quot;name&quot;:&quot;Example&quot;</td>
<td>}</td>
</tr>
<tr>
<td>logLevel</td>
<td>&quot;logLevel&quot;:&quot;INFO&quot;,</td>
<td>The type of info returned in the log. The possible values you can enter are Info, debug, and</td>
</tr>
</tbody>
</table>

```json
{
  "servers": {
    "servername":{
      "ip":"",
      "port":"",
      "apiKey":""
    },
    "tasks":[]
  }
}
```
<table>
<thead>
<tr>
<th>Key</th>
<th>Example</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>emails</td>
<td>&quot;emails&quot;: [</td>
<td>This object defines who receives an email and their email address when an email is triggered according to the events you define in the mail object.</td>
</tr>
<tr>
<td></td>
<td>{</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;email&quot;:&quot;<a href="mailto:john.test@sisense.com">john.test@sisense.com</a>&quot;,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;name&quot;:&quot;Example&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>]</td>
<td></td>
</tr>
<tr>
<td>email</td>
<td>&quot;emails&quot;: [</td>
<td>The email address where emails are to be sent depending on the value of the When key.</td>
</tr>
<tr>
<td></td>
<td>{</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;email&quot;:&quot;<a href="mailto:john.test@sisense.com">john.test@sisense.com</a>&quot;,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;name&quot;:&quot;Example&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>]</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>&quot;emails&quot;: [</td>
<td>The name of the recipient of the email.</td>
</tr>
<tr>
<td></td>
<td>{</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;email&quot;:&quot;<a href="mailto:john.test@sisense.com">john.test@sisense.com</a>&quot;,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;name&quot;:&quot;Example&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>]</td>
<td></td>
</tr>
<tr>
<td>cubes</td>
<td>&quot;cubes&quot;: {</td>
<td>The cubes object contains all your ElastiCubes in your configuration and</td>
</tr>
<tr>
<td></td>
<td>&quot;HA_QueryWeb1&quot;: {</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;ecube&quot;:&quot;Sample ECommerce&quot;,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>}</td>
<td></td>
</tr>
</tbody>
</table>
### Build ElastiCubes

<table>
<thead>
<tr>
<th>Key</th>
<th>Example</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;url&quot;:&quot;\127.128.129.91\C:\ProgramData\Sisense\PrismServer\ElastiCube Data&quot;</td>
<td>their location. As the Sisense Orchestrator Service is installed on the build node, the build cubes are local, so you must define the name of the ElastiCube. For ElastiCubes on a query node, you must define the ElastiCube name, URL of the remote server, and its directory.</td>
<td></td>
</tr>
<tr>
<td>&quot;localPath&quot;:&quot;C:\ProgramData\Sisense\PrismServer\ElastiCubeData&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;HA_QueryWeb2&quot;:{</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ecube&quot;:&quot;Sample ECommerce&quot;,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;url&quot;:&quot;\127.128.129.93\C:\ProgramData\Sisense\PrismServer\ElastiCubeData&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;localPath&quot;:&quot;C:\ProgramData\Sisense\PrismServer\ElastiCubeData&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;HA_QueryWeb1&quot;:{</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ecube&quot;:&quot;Sample ECommerce&quot;,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;url&quot;:&quot;\127.128.129.91\C:\ProgramData\Sisense\PrismServer\ElastiCubeData&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;localPath&quot;:&quot;C:\ProgramData\Sisense\PrismServer\ElastiCubeData&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Key</td>
<td>Example</td>
<td>Value</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>-------</td>
</tr>
</tbody>
</table>
| ecube | sense\\PrismServer\\ElastiCubeData" },
      | "HA_QueryWeb1":{
      |       | "ecube":"Sample ECommerce",
      |       | "url":"\\\127.128.129.91\\C:\\ProgramData\\Sisense\\PrismServer\\ElastiCubeData",
      |       | "localPath":"C:\\ProgramData\\Sisense\\PrismServer\\ElastiCubeData" },
|       |         | The name of the ElastiCube. |
| url   | "HA_QueryWeb1":{
      |       | "ecube":"Sample ECommerce",
      |       | "url":"\\\127.128.129.91\\C:\\ProgramData\\Sisense\\PrismServer\\ElastiCubeData",
      |       | "localPath":"C:\\ProgramData\\Sisense\\PrismServer\\ElastiCubeData" },
<p>|       |         | The URL address of the ElastiCube server. If the URL is to a secure address, the value should include the username and password credentials. For example, “ssh://username:password@10.50.1.128:/ecubes”. See Scenario 1 for an |</p>
<table>
<thead>
<tr>
<th>Key</th>
<th>Example</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>localPath</td>
<td>&quot;HA_QueryWeb1&quot;: {</td>
<td>Directory of the ElastioCube.</td>
</tr>
<tr>
<td></td>
<td>&quot;ecube&quot;: &quot;Sample ECommerce&quot;,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;url&quot;: &quot;\127.128.129.91\C:\ProgramData\Sisense\PrismServer\ElastiCubeData\&quot;,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;localPath&quot;: &quot;C:\ProgramData\Sisense\PrismServer\ElastiCubeData\&quot;</td>
<td></td>
</tr>
</tbody>
</table>

<p>| tasks     | &quot;tasks&quot;: { &quot;task1&quot;: [ { &quot;build&quot;: { &quot;cube&quot;: [ &quot;HA_QueryWeb1&quot;, &quot;HA_QueryWeb2&quot; ] }, &quot;queue&quot;: [ &quot;entire&quot; ] } ], | The Tasks object contains a task array that defines which ElastioCube should be built, the type of build, and to which ElastioCubes the build should be distributed to. In high availability scenarios, running multiple concurrent builds is not recommended and |
|           |   }                                                                 |                                                                      |</p>
<table>
<thead>
<tr>
<th>Key</th>
<th>Example</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>task</td>
<td>&quot;tasks&quot;:{</td>
<td>can result in stability issues. Sisense recommends that tasks are spaced out so each build is complete before the next one begins.</td>
</tr>
<tr>
<td></td>
<td>&quot;task1&quot;:[</td>
<td></td>
</tr>
<tr>
<td></td>
<td>{</td>
<td>The name of the task array. Currently, you can define only one task. Defining multiple tasks in the config.json file may cause builds to fail.</td>
</tr>
<tr>
<td></td>
<td>&quot;build&quot;:{</td>
<td>In high availability scenarios, running multiple concurrent builds is not recommended and can result in stability issues. Sisense recommends that tasks are spaced out so each build is complete before the next one begins.</td>
</tr>
<tr>
<td></td>
<td>&quot;cube&quot;:[</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;HA_QueryWeb1&quot;,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;HA_QueryWeb2&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>],</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;queue&quot;:['entire']</td>
<td></td>
</tr>
<tr>
<td></td>
<td>]</td>
<td></td>
</tr>
<tr>
<td>Key</td>
<td>Example</td>
<td>Value</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>build</td>
<td>&quot;build&quot;: {</td>
<td>The build object defines the ElastiCube to be built and distributed.</td>
</tr>
<tr>
<td></td>
<td>&quot;cube&quot;: [</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;HA_QueryWeb1&quot;,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;HA_QueryWeb2&quot;]</td>
<td></td>
</tr>
<tr>
<td>cube</td>
<td>&quot;build&quot;: {</td>
<td>The name of the cube to be built.</td>
</tr>
<tr>
<td></td>
<td>&quot;cube&quot;: [</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;HA_QueryWeb1&quot;,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;HA_QueryWeb2&quot;]</td>
<td></td>
</tr>
<tr>
<td>queue</td>
<td>&quot;queue&quot;: [</td>
<td>The value of this key is the type of build Sisense should attempt and in what order. There are two possible values delimited by a comma:</td>
</tr>
<tr>
<td></td>
<td>&quot;accumulate&quot;,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;entire&quot;</td>
<td></td>
</tr>
</tbody>
</table>

**accumulate**: Attempts an accumulative build. **entire**: Attempts an entire build. **schemaChanges**: Attempts to update the build only if changes were made to the
<table>
<thead>
<tr>
<th>Key</th>
<th>Example</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>distribute</td>
<td>{</td>
<td>Determines which cubes the latest build should be distributed to. The value should be the ElastiCube name for your ElastiCubes on your query nodes. For example, [“cube1, cube2, cube3”]. You can add multiple ElastiCubes delimited by a comma separator.</td>
</tr>
<tr>
<td></td>
<td>&quot;distribute&quot;: [</td>
<td>schema since the previous build. Sisense recommends the following value: &quot;accumulate&quot;,&quot;entire&quot; In this example, Sisense attempts to do an accumulative build first, and if it fails, Sisense attempts an entire build. If the build fails for all builds, Sisense retries again according to the next scheduled build.</td>
</tr>
<tr>
<td>Key</td>
<td>Example</td>
<td>Value</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>reattach</td>
<td>&quot;reattach&quot;:{</td>
<td>Reattach is an object that contains two objects, DeleteOldDbfarm and Cube.</td>
</tr>
<tr>
<td></td>
<td>&quot;deleteOldDbfarm&quot;:true,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;cube&quot;: [</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;HA_QueryWeb1&quot;,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;HA_QueryWeb2&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>},</td>
<td></td>
</tr>
<tr>
<td>deleteOldDbfarm</td>
<td>&quot;reattach&quot;:{</td>
<td>A boolean value that determines if an old ElastiCube is to be deleted after it has been updated.</td>
</tr>
<tr>
<td></td>
<td>&quot;deleteOldDbfarm&quot;:true,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;cube&quot;: [</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;HA_QueryWeb1&quot;,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;HA_QueryWeb2&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>},</td>
<td></td>
</tr>
<tr>
<td>cube</td>
<td>&quot;reattach&quot;:{</td>
<td>The name of the ElastiCube to be reattached or not.</td>
</tr>
<tr>
<td></td>
<td>&quot;deleteOldDbfarm&quot;:true,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;cube&quot;: [</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;HA_QueryWeb1&quot;,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;HA_QueryWeb2&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>]</td>
<td></td>
</tr>
<tr>
<td>Key</td>
<td>Example</td>
<td>Value</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>mail</td>
<td>`{</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;mail&quot;:&quot;build&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>}</td>
<td></td>
</tr>
</tbody>
</table>

Indicates when an email alert is triggered. Email alerts can be triggered for the following scenarios:

**None**: No email is ever sent.

**Build**: An email is sent when a build is complete regardless if the build was successful or not.

**Fail**: An email is sent when a build fails.

<table>
<thead>
<tr>
<th>scheduler</th>
<th>&quot;scheduler&quot;:[</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;task&quot;:&quot;task1&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;schedule&quot;:&quot;15 14 * *&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;enabled&quot;:true</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td></td>
<td>]</td>
</tr>
</tbody>
</table>

The Scheduler object contains an array of tasks that define when a task is to be initiated.

<table>
<thead>
<tr>
<th>task</th>
<th>&quot;scheduler&quot;:[</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;task&quot;:&quot;task1&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;schedule&quot;:&quot;15 14 * *&quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
</tbody>
</table>

Defines which task to perform and the order. Currently, Sisense only supports
<table>
<thead>
<tr>
<th>Key</th>
<th>Example</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>*&quot;, &quot;enabled&quot;:true</td>
<td>one task.</td>
</tr>
<tr>
<td></td>
<td>}</td>
<td></td>
</tr>
<tr>
<td>schedu</td>
<td>&quot;scheduler&quot;: [</td>
<td>The time that a task is to be initiated in Cron format. Some examples: -To run a build each night at midnight, enter the value &quot;0 0 * * * *&quot; -To run a build every hour, enter the value &quot;0 * * * * *&quot; See the full Cron format reference here.</td>
</tr>
<tr>
<td>le</td>
<td>{</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;task&quot;:&quot;task1&quot;, &quot;schedule&quot;:&quot;15 14 * *&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;enabled&quot;:true }</td>
<td></td>
</tr>
<tr>
<td></td>
<td>}</td>
<td></td>
</tr>
<tr>
<td>enabled</td>
<td>&quot;scheduler&quot;: [</td>
<td>A boolean value that indicates if the task is to be executed or not.</td>
</tr>
<tr>
<td></td>
<td>{</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;task&quot;:&quot;task1&quot;, &quot;schedule&quot;:&quot;15 14 * *&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;enabled&quot;:true }</td>
<td></td>
</tr>
<tr>
<td></td>
<td>}</td>
<td></td>
</tr>
</tbody>
</table>
Use Cases

The following section presents two use cases and an example config.json file that demonstrates how to support these use cases. For assistance with additional HA configurations, contact your Customer Service Manager.

Scenario 1 – Single Stack

In this example, there is one build node and two query nodes. Each query node is hosted on a separate machine while the components that make up the query node are hosted on the same machine as a single application stack.
In this example, the build cube is called “Sample eCommerce”. This ElastiCube is distributed to two ElastiCube servers defined in the cube1 and cube2 objects. The build cube is distributed first to cube1 and then to cube2 as defined in the tasks object. The first time the Sisense Orchestrator Service tries to build the BuildTest1 ElastiCube, it attempts an accumulative build and if that fails, an entire build. After each build is replicated, the previous version of the ElastiCube
is deleted as specified in the reattach object. This task takes place hourly. If the
distribution fails, Sisense sends an email to johndoe@Sisense.com with a debug
log attached.

```json
{
    "general":{
        "logLevel":"INFO",    //Determines type of log
events recorded
        "emails":[
            {
                "email":"john.test@sisense.com",    //Who
to send reports
                "name":"Example1"
            }
        ]
    },
    "cubes":{
        "localCubeBuildTest1":{
            "ecube":"Sample ECommerce"      //Name of the
ecube to be distributed
        },
        "cube1":{
            "ecube":"Sample ECommerce",    //Name of the
ecube on the query node
            "url":"\\\127.128.129.91\\C:\\ProgramData\\Sisense\\PrismServer\\ElastiCubeData\\Sample ECommerce",
            "localPath":"C:\\ProgramData\\Sisense\\PrismServer\\ElastiCubeData\\Sample ECommerce"
        },
        "cube2":{
            "ecube":"Sample ECommerce",    //Name of the
ecube on the query node
            "url":"\\\127.128.129.92\\C:\\ProgramData\\Sisense\\PrismServer\\ElastiCubeData\\Sample ECommerce",
            "localPath":"C:\\ProgramData\\Sisense\\PrismServer\\Ela"
"stiCubeData":
{
"Sample ECommerce":
{
"tasks":{
"task1":{
"build":{
"cube":[
"cube1",
"cube2"
],
"queue":[
"accumulate",  //The first type of build to be attempted
"entire"   //The second type of build to perform if the first fails
]

"distribute":[
//The order of how ecubes on the build node is distributed to query nodes
"cube1",
"cube2"
]

"reattach":{
"deleteOldDbfarm":true,
"cube":[
"cube1",
"cube2"
]
}
}
}
"mail":"build"   //What event triggers an email
}
Scenario 2 – Distributed Stack

In this example, there are three query nodes in which each component is located in a different remote location. This scenario, while costly to implement, demonstrates a distributed application stack configuration where each component is replicated and located on a separate server independent of the other components. If any component fails, the rest of the architecture remains intact. For example, the web server is on one server, the ElastiCubes are stored on another server, and the MongoDB is located on another server.
The build cube in this example is called “BuildTest1”. This ElastiCube is distributed to three remote ElastiCube servers defined in the cube1, cube2, and cube3 objects. The build cube is distributed first to cube1, then cube2, and then cube3 as defined in the tasks object. The URLs are standard URLs for cube1 and
cube2 while the URL for cube3 is secured through SSH. When the URL is secured, you must provide the URL and the required credentials to access it. The first time the Sisense Orchestrator Service tries to build the BuildTest1 ElastiCube, it attempts an accumulative build and if that fails, an entire build. If both builds fail, Sisense attempts to update the schema only. After each build is replicated, the previous version of the ElastiCube is deleted as specified in the reattach object. This task takes place at 14:15 on the 1st of every month. After a build is successful, Sisense sends an email to johndoe@Sisense.com.

```json
{
    "general": {
        "logLevel": "DEBUG",
        "emails": [
            {
                "email": "john.test@sisense.com",
                "name": "Example1"
            }
        ]
    },
    "cubes": {
        "localCubeBuildTest1": {
            "ecube": "Sample ECommerce"
        },
        "cube1": {
            "ecube": "Sample ECommerce",
            "url": "\\127.128.129.91\\C:\ProgramData\Sisense\PrismServer\ElastiCubeData\\Sample ECommerce",
            "localPath": "C:\ProgramData\Sisense\PrismServer\ElastiCubeData\\Sample ECommerce"
        },
        "cube2": {
            "ecube": "Sample ECommerce",
            "url": "\\127.128.129.92\\C:\ProgramData\Sisense\PrismServer\ElastiCubeData\\Sample ECommerce",
            "localPath": "C:\ProgramData\Sisense\PrismServer\ElastiCubeData\\Sample ECommerce"
        }
    }
}
```
"localPath":"C:\\ProgramData\\Sisense\\PrismServer\\ElastiCubeData\\Sample ECommerce"
  }
  "cube3":{
    "ecube":"Sample ECommerce",
  "url":"\\\127.128.129.93\\C:\\ProgramData\\Sisense\\PrismServer\\ElasticCubeData\\Sample ECommerce",
  "localPath":"C:\\ProgramData\\Sisense\\PrismServer\\ElastiCubeData\\Sample ECommerce"
}
},
"tasks":{
  "task1":[
  {
    "build":{
      "cube":[
        "cube1",
        "cube2",
        "cube3"
      ],
      "queue":[
        "accumulate",
        "entire",
        "schemachanges"
      ]
    }
  },
  {
    "distribute":[
      "cube1",
      "cube2",
      "cube3"
    ]
  },
  {
    "reattach":{}
"deleteOldDbfarm":true,
"cube":[
  "cube1",
  "cube2",
  "cube3"
],
"mail":"fail",
"scheduler":[
  {
    "task":"task1",
    "schedule":"15 14 1 * *",
    "enabled":true
  }
]
Add Query Nodes

In some configurations, replicating components can improve performance by scaling out and preventing any potential bottlenecks. Each component you replicate must be added to the ElastiCube Set and the config.json file. You can continue to add query nodes according to your requirement so long as the nodes are included in the config.json file.

To add a query node:
Stop the Sisense Orchestrator Service. Open Windows Services, select Sisense.Orchestrator, and click .

In the Sisense Web Application, click Admin and select the Data Sources tab on the left.
Hover over the ElastiCube you want to add to the ElastiCube Set and click on the menu that appears. Click Add to ElastiCube Set and select the set that you want to add the ElastiCube to.
Edit the config.json located at Sisense/Sisense.Orchestration/Config/ and add the new ElastiCube to the Cubes object.
Save the config.json file.
Restart the Sisense Orchestrator Service.
Remove Query Nodes

You can remove query nodes by removing them from the ElastiCube Set and the config.json file.
To remove a query node:

Stop the Sisense Orchestrator Service. Open Windows Services, select Sisense.Orchestrator, and click 🔄.

In the Sisense Web Application, click Admin and select the Data Sources tab on the left.

In the ElastiCube Set table, click and clear the checkboxes of the ElastiCubes to be removed. Click Save after you have selected the relevant ElastiCubes.

Edit the config.json located at Sisense/Sisense.Orchestration/Config/ and delete the relevant ElastiCubes from the Cubes object.

Save the config.json file.

Restart the Sisense Orchestrator Service.
Next Steps

Setting Up ElastiCube Sets
Setting Up ElastiCube Sets

Previous Steps

Distributing ElastiCubes

Sisense ElastiCube Sets are collections of ElastiCube models with identical schemas that allow you to query running ElastiCubes within the ElastiCube Set while other ElastiCubes are building.

**Note:** From Sisense V6.5 onwards, you can query ElastiCubes during accumulative builds without ElastiCube Sets. However, to ensure that your users can query ElastiCubes after a failed build, you can implement ElastiCube Sets to handle your queries while Sisense rebuilds the failed ElastiCube.

Grouping ElastiCubes into ElastiCube sets has several benefits:
- Allowing viewers to query the most up-to-date cubes within an ElastiCube Set.
- Reducing server load during builds by using multiple nodes in which only the non-building node is queried.
- Achieving data redundancy by running builds interchangeably across multiple nodes.

To maximize availability and reduce build and dashboard response times, administrators can distribute resource allocation across multiple Sisense nodes. By adding ElastiCubes across multiple servers to a single ElastiCube Set, administrators can configure ElastiCube Sets to run builds interchangeably via the Sisense Orchestrator Service so the most up-to-date and complete ElastiCubes are queried.

The Sisense Orchestrator Service is an automated service that can be configured to synchronize builds across the ElastiCube Set. If you change your schemas within any of the ElastiCubes, you must manually change the schemas in all of the ElastiCubes and rebuild the cube entirely.
This page provides the following information:
Instructions on Creating and Deleting ElastiCube Sets.
A walkthrough for how you can create a set and configure the Sisense Orchestrator Service. See Working with ElastiCube Sets. (Optional) Instructions on how to implement and configure the Sisense Orchestrator Service. See Activating the Sisense Orchestrator Service. How to update an ElastiCube within an ElastiCube Set.
Creating ElastiCube Sets

From the **Admin** page, Administrators can create ElastiCube Sets. ElastiCube Sets require at least two ElastiCubes. The Administrator can add multiple ElastiCubes to a set, however, ElastiCubes can only be part of a single ElastiCube Set at a time. After an Administrator creates an ElastiCube Set or adds an ElastiCube to an existing set, the ElastiCube is disabled in the original server and no further actions can be performed outside of the set.

Sisense recommends that you develop ElastiCubes in a development environment or separate ElastiCube to ensure they work before adding them to an ElastiCube Set.

**Note:** Build nodes used in high availability configurations should not be included in your ElastiCube Set. If you have implemented high availability, only query nodes should be included in the ElastiCube Set.

**To create an ElastiCube Set:**
Click **Admin** and select the **Data Sources** tab on the left. For the relevant ElastiCube you want to add to the ElastiCube Set, select and click **Create ElastiCube Set**.

In the Create ElastiCube Set window, enter a name for the set and select the ElastiCubes you want to include in the set.
From the Routing Strategy list, select a strategy for how you want to route traffic to your ElastiCube Set. See [Routing Queries in ElastiCube Sets](#) for more information.

From the Failover ElastiCube list, select which ElastiCube you prefer to use if the other ElastiCubes in the set are not available. Failover ElastiCubes only handle queries when an ElastiCube that is part of the ElastiCube Set fails. (Note: this feature is available in Sisense V6.7 onwards).

**Create ElastiCube Set**

**ElastiCube Set Name**

Type in a descriptive name for the new ElastiCube Set.

**Routing Strategy**

- **Single ElastiCube**

**ElastiCubes**

- **Search ElastiCubes...**
  - □ Facebook Ads Demo 4.01MB
  - □ new 30.46MB
  - □ Sample ECommerce 59.09MB
  - □ Sample Healthcare 1.56MB
  - ✔ Sample Lead Generation 2.25MB
  - □ Training 814.28KB

1 ElastiCube Selected

**Failover ElastiCube**

- **None**

Click **Create**.

In the Existing Dashboards popup message, indicate if you want to associate your ElastiCube’s dashboards with the ElastiCube Set or maintain the
association with the original ElastiCube.

Click **OK**. The ElastiCube Set is created and displayed in the Admin page. After you have created an ElastiCube Set, you can configure the ElastiCube Set's Hierarchies, Access Rights, and Data Security.
Adding ElastiCubes to ElastiCube Sets

After you have created an ElastiCube Set, you can add more ElastiCubes to the set.

When you add an ElastiCube to an existing ElastiCube Set, any dashboards associated with that ElastiCube are passed on to the ElastiCube Set.

**Note:** You cannot add ‘localhost’ to an ElastiCube Set. Instead, you can add your localhost as a new server with its IP address as the server name and then add ElastiCubes from it to an ElastiCube Set.

**To add an ElastiCube to the ElastiCube Set:**

Click **Admin** in the upper right corner and select the **ElastiCubes** tab on the left.

For the relevant ElastiCube you want to add to the ElastiCube Set select ‒ and click **Add to ElastiCube Set**.

**OR**

In the ElastiCube Set table, click ‒ and select **Edit** and select the ElastiCubes to be added to the set. Click **Save** after you have selected the relevant ElastiCubes.
The ElastiCube is added to the ElastiCube Set and displayed in the ElastiCube Set table on the **Admin** page.
Deleting ElastiCube Sets

Administrators can delete ElastiCube Sets. Deleting an ElastiCube Set removes the set from the Admin page and any dashboards associated with that set.

**To delete an ElastiCube Set:**

- Hover over its title and click the Trash icon.
- Click **Delete** to confirm that you want to delete the ElastiCube Set.
Overcoming Build Failures in ElastiCube Sets

In Sisense, if an accumulative build fails, you must rebuild the ElastiCube that failed entirely to ensure the validity of the ElastiCube. If you use the Sisense Orchestrator Service, Sisense recommends that you define the value of the **Queue** key as “FullUpdateExisting, Full”

```
"queue": ["FullUpdateExisting","Full"]
```

In this configuration, the Sisense Orchestrator Service attempts to build the set using accumulative builds and in the event of a failure, a full build is initiated.
Retrieving a Token

As part of the config.json file, you must provide a token that Sisense uses for authenticating your ElastiCube Sets.

You can retrieve this token through Sisense’s API documentation. After you retrieve the token, save it as the value of apiKey in the config.json file.

To retrieve a token:
In Sisense, access the API documentation, select Admin > REST API > REST API Reference.

In the REST API Reference site, select Version 1.0 in the top-right corner of the page.
Open the authentication method.
Enter your login credentials and click **Run**. The access_token is displayed in the Response Body below.

Copy the token and paste it into the value of apiKey in the config.json file.
Updating ElastiCubes within an ElastiCube Set

The Sisense Orchestrator Service automatically updates your data across ElastiCubes within an ElastiCube Set. If you want to modify the schema, for example, by adding new tables, you must manually rebuild the ElastiCubes and then import that ElastiCube to the other machines hosting your ElastiCube Set. While rebuilding your ElastiCube, you should deactivate the Sisense Orchestrator Service to prevent it from updating the ElastiCubes with different schemas. After the schemas are identical for all the ElastiCubes across the ElastiCube Set, you can reactivate the Sisense Orchestrator Service to automatically update your ElastiCubes.

**To update an ElastiCube within ElastiCube Set:**
Stop the Sisense Orchestrator Service. Open Windows Services, select Sisense.Orchestrator, and click .
Apply your schema changes to the ElastiCube and run a full build. Export the ElastiCube and import it on your other machines. For more information, see Importing and Exporting ElastiCube Data. In Windows Services, restart the Sisense Orchestrator Service by selecting Sisense.Orchestrator and clicking ▶.
Next Steps

Routing Queries in ElastiCube Sets
Routing Queries in ElastiCube Sets

When a user queries an ElastiCube that is part of an ElastiCube Set, Sisense routes the query to a single ElastiCube with the latest ElastiCube build. By default, ElastiCube Sets operate in active-passive mode. In active-passive mode, one ElastiCube, by default the ElastiCube with the latest build, handles incoming queries. This ElastiCube is active while the remaining ElastiCubes are on standby in passive mode.

Sisense also supports active-active mode where queries are spread across available ElastiCubes within an ElastiCube Set. In active-active mode, each ElastiCube that is not currently building can handle queries.

You can modify how queries are routed to ElastiCubes within ElastiCube Sets by selecting one of the following routing strategies:

**Single** (Default): ElastiCubes operate in active-passive mode where the ElastiCube with the freshest data receives all the queries while the remaining ElastiCubes are on standby.

**Multiple ElastiCubes – Fresh Data**: ElastiCubes operate in active-active mode where queries are spread across all ElastiCubes with the latest data.

**Multiple ElastiCubes – Best Spread**: ElastiCubes operate in active-active mode where queries are spread across the largest group of ElastiCubes with identical data.

**To set a query routing strategy:**

Click **Admin** and select the **Data Sources** tab on the left.
For the relevant ElastiCube Set, select **Edit**.

From the Routing Strategy list, select the relevant strategy. Click **Save**.
Securing the Message Broker's Communication

**Note:** The procedure described below is an optional step for hardening the security of the Sisense Message Broker.

The message broker is responsible for communicating events across Sisense components. To ensure that your communication is secure, RabbitMQ, which Sisense uses as the message broker, supports Transport-Layer-Security (TLS) connections. TLS encrypts traffic between Sisense components and verifies that various Sisense components are authentic.

The communication is encrypted through the use of certificates. You are responsible for generating your certificate bundle, which includes the Certificate Authority (CA file), a certificate (public key) file and a private key file. To encrypt and decrypt traffic, you must provide Sisense with certificate/key pairs in the Sisense Configuration Manager. The following article explains how you can generate TLS certificates.

**To enable TLS-connections for the Sisense message broker:**

Open the Configuration Manager in your browser located at http://localhost:3030/.
Under the section Message Broker, toggle **TLS Enabled** to enable TLS support.

Enter the following details:

**Connection String:** The connection string to the Sisense message broker. The RabbitMQ uses the [AMQP URI Specification](#) for connections.

**CA Certificate:** Enter your CA certificate. This is provided when you generate your certificate bundle.

**Private Certificate:** Enter your private certificate. This is provided when you generate your certificate bundle.

**Private Key:** Enter your private key. This is provided when you generate your certificate bundle.
Click **Save**.
Expanding Single-Node Deployments to Multi-Node Deployments

If you want to expand an existing single-node deployment to a multi-node deployment retaining your existing assets, you need to take the following into consideration:

- Migrating your application database (MongoDB) containing the system metadata (users, groups, dashboard definitions, and much more)
- Usage of ElastiCube Sets for query load balancing
- Migrating resource files (rebranding files, email templates, languages, etc.)
- Migrating plugin files
- And more.

Contact Sisense Support to assist with this process.
Using ElastiCube Models in Multi-Node Environments

You can manage your ElastiCube models in multi-node deployments with the following configuration and behavior:
Step 1

The application must know the IP address of the build server. You can specify the build node server IP address in the Configuration Manager:

In your browser, open the Configuration Manager at: [server address]:3030, for example, 127.61.98.2030:3030.

In **Build Node Server**, enter the IP address of the build node.
Step 2

To edit and build ElastiCubes in the **Data** page, you must add the build node as a remote server.

In the **Admin** page, select **Data Sources** and click **Add Server**.

![Add Server Page]

Enter your IP address. Make sure to use the server IP address and not the server name when you add it.
Note: If you initiate a build from the Data page at the same time that a build is initiated by the Orchestrator service or when a build is already in progress, one of the builds may fail. If this happens, initiate another build.

You will now see the ElastiCubes from the build server on your Data page. This ElastiCube is displayed with the IP address of the build node.

You can now edit and build ElastiCubes that reside on your build server from the Data page.
Step 3

If you want to use multiple query nodes, configure the Orchestrator to distribute your ElastiCube models from the build node to the query nodes.
For more information, see Activating the Sisense Orchestrator Service.
Step 4

Connect your dashboards to an ElastiCube or ElastiCube Set.

If your dashboard is new, you can select the data source when creating the dashboard. For existing dashboards, see Changing a Dashboard’s Data Source.

If you want to use ElastiCube Sets, perform the following steps:
- Add the query nodes as remote servers in the Admin page. See Step 2 for information on how to set up a query node as a remote server.
- Make sure the Orchestrator has completed distributing the Elasticubes to the query nodes. You will be able to see the Elasticubes from the query nodes on the Data page.
- Create the ElastiCube Sets using the Elasticubes on the query nodes.
- Connect the dashboards to the ElastiCube sets. See Changing a Dashboard’s Data Source for more information.

Note: You must edit only the ElastiCube used for the build, meaning the ElastiCube that is located on the build node. You should not edit the Elasticubes that are on the query nodes, and are part of the ElastiCube sets, as their data is overwritten upon the next build.
Supporting Usage Analytics in Multi-Node Environments

Usage Analytics is supported in multi-node deployments. The information is collected from all the nodes in the system, stored on the build node, and then displayed in your dashboard on the application node.

If you are using the Usage Analytics feature in a multi-node deployment, these are the steps you need to perform to support Usage Analytics. In some cases, you may have defined these settings already, if so, verify that each of these steps is complete.
Step 1

Sisense must know the IP address of the build server. You can specify the build node server IP address in the Configuration Manager on your application node:

In your browser, open the Configuration Manager at: [server address]:3030, for example, 127.61.98.2030:3030.

In **Build Node Server**, enter the IP address of the build node.

---

**Build Node Server**

---

**Application Database**

Connection String: `mongodb://AppUser:5s5bQGkDPUY6zjXRuEAgg==@localhost:27018/`
Step 2

On the application node, enable Usage Analytics.
In the Admin page, select Usage Analytics.

Toggle Collect Usage Analytics to enable.
Enabling Usage Analytics creates the Usage Analytics Model ElastiCube on the build node.
Step 3

Add the build node as a remote server to the application node.

In the Admin page, select Data Sources and click Add Server.

Enter your IP address. Make sure to use the server IP address and not the server name when you add it.
Then, on the build node, schedule builds for the Usage Analytics Model ElastiCube.

See [Scheduling Builds](#) for more information.
Step 4

Configure the Orchestrator on your build node to distribute the Usage Analytics Model ElastiCube to the query nodes.

See [Distributing ElastiCube Builds to Query Nodes](#) for more information.
Step 5

On your application node, add your query nodes as remote servers. In the Admin page, select Data Sources and click Add Server.

Enter your IP address. Make sure to use the server IP address and not the server name when you add it.

Add Server

IP or Server Name

Who can access this server?  ○  Admins Only  ○  Everyone

Add Server  Cancel
Step 6

Create ElastiCube Sets for the Usage Analytics Model ElastiCubes for the query nodes.

See Creating ElastiCube Sets for more information.
Step 7

Update your default Usage Analytics dashboards to use the Usage Analytics Model ElastiCube Set.

See [Changing a Dashboard’s Data Source](#) for more information.

You are now ready to view your Usage Analytics data in a Multi-Node deployment topology.
Overview

Sisense Usage Analytics is a set of pre-defined dashboards and ElastiCube that enable you to monitor your Sisense user and dashboard activity. With Sisense Usage Analytics, you can better understand how users are interacting with Sisense dashboards and optimize your configuration accordingly. For example, you can understand which dashboards are making an impact and how fast the dashboards are loading.

Sisense collects usage data for the following actions:
- Opening a dashboard
- Changing a filter
- Selecting areas (filter)
- Drilling down into widgets
- Exporting a dashboard to PDF
- Exporting a dashboard to images
- Adding KPIs to Pulse

The data displayed in your Usage Analytics dashboards is collected once Usage Analytics is enabled. When you first enable the feature, there will only be a small amount of data. Data is accumulated while the feature is enabled, and stored on your Sisense Server for 30 days.

Sisense Usage Analytics includes the following dashboards:

**Usage - Dashboards**: Provides insights regarding the number of dashboards in your system, how often they are used, and their performance.

**Usage - Domains**: Provides insights regarding the behavior of your users, from different email domains in your system. This is useful for OEMs who have tenants from different email domains. From this dashboard you can monitor system usage for each tenant.
**Usage - General:** Provides a general summary of dashboard performance, dashboard usage, and who is viewing your dashboards.

**Usage - Groups:** Provides insights regarding group activity in your system and their dashboard usage.

**Usage - Users:** Provides insights regarding the behavior of specific users in the system.

The Usage Analytics dashboards included with Sisense are visible after you activate the Usage Analytics feature in the Admin page. After activating Usage Analytics, Sisense displays the default Usage Analytics ElastiCube in the Data page and your default Usage Analytics dashboards in the Analytics page.

You can fully customize the default ElastiCube and dashboards as you like.

**Note:** The Usage Analytics ElastiCube is built on a CSV file where your data is stored. This CSV should not be modified in any way or the ElastiCube will not build. To update the Usage Analytics data, you should perform a full build. If you modify your ElastiCube and it no longer builds, or if you ever need to revert back to the original ElastiCube and dashboards, you can do so through the Admin page or through the REST API. See [Restoring Usage Analytics](#) for more information.
If you have implemented a multi-node environment and want to view usage data across all your nodes, there are additional steps you must take to set it up. See Supporting Usage Analytics in Multi-Node Environments for more information. **Note:** Groups with ‘;’ in their name will lead to incorrect data being displayed in your Usage Analytics as this causes the groups to be parsed as two separate groups.
Related Topics

Activating Usage Analytics
Dashboard Analytics
Domain Analytics
General Analytics
Groups Analytics
Users Analytics
Restoring Usage Analytics
Supporting Usage Analytics in Multi-Node Environments
Activating Usage Analytics

There are two steps to activating Usage Analytics. The first step is to enable Usage Analytics from the Admin page, and the second step is to build the model called Usage Analytics Model that is automatically created when Usage Analytics is enabled.

Once Usage Analytics is enabled, your server’s activity is recorded on your Sisense Server. Enabling Usage Analytics adds a Usage Analytics ElastiCube to the Data page and a Usage Analytics folder to your Dashboards list where all the pre-defined usage dashboards are stored.

These dashboards are initially empty until you complete the second step, which is to build the pre-defined Usage Analytics model. After you build your Usage Analytics model, your Usage Analytics dashboards will display the updated usage data. The data displayed in your Usage Analytics dashboards is collected when
Usage Analytics is enabled. All the data displayed in your Usage Analytics is stored on your Sisense Server for 30 days.

If you have implemented a multi-node environment and want to view usage data across all your nodes, there are additional steps you must take to set it up. See [Supporting Usage Analytics in Multi-Node Environments](#) for more information.

**Note:** To display the latest usage analytics data, you need to perform a full build on the Usage Analytics model. You can automate this step by [scheduling periodic builds](#).

**To activate Usage Analytics:**

In the **Admin** page, select Usage Analytics.

Toggle **Collect Usage Analytics** to enable. Sisense automatically adds the **Usage Analytics Model** to your **Data** page.

In the **Data** page, open the **Usage Analytics Model**.
Build the Usage Analytics Model. Your Usage Analytics dashboards display all the information collected since you began collecting usage analytics data. To update your dashboards with the latest data, you need to run a full build each time.
Dashboard Analytics

The Usage - Dashboard dashboard provides you with the ability to see how often your dashboards are being used, how often Viewers are interacting with them, and how quickly your dashboards are loading. You can filter the information displayed in your widgets with a set of filters on the right. With these filters, you can filter the details about your dashboard by various parameters, such as a specific dashboard or user or group.

This dashboard is useful for answering the following questions:

<table>
<thead>
<tr>
<th>Question</th>
<th>Widget</th>
</tr>
</thead>
<tbody>
<tr>
<td>How many dashboards have been used?</td>
<td>Dashboards Use in Period</td>
</tr>
<tr>
<td>How many Viewers have been looking at my dashboards?</td>
<td>Avg Viewers per Dashboard</td>
</tr>
<tr>
<td>Question</td>
<td>Widget</td>
</tr>
<tr>
<td>---------------------------------------------------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>How many times have my dashboards been refreshed?</td>
<td>Avg Refreshes per Dashboard, Dashboard Refreshes and Viewers</td>
</tr>
<tr>
<td>How long does it take to load a dashboard?</td>
<td>Average Dashboard Load Time, Avg Action per Dashboard, Dashboard Load Time</td>
</tr>
<tr>
<td>How much are Viewers interacting with my dashboards?</td>
<td>Avg Actions per Dashboard, Dashboard Actions, Dashboard Usage</td>
</tr>
<tr>
<td>Who is looking at my dashboards?</td>
<td>Dashboard Viewers</td>
</tr>
</tbody>
</table>
Domain Analytics

Domain Analytics provide you with insights regarding users from different email domains active in your account. A domain represents each unique email domain of your users in your Sisense system. For example, if you have users with the email addresses john@acme.com and sally@sisense.com, these represent two different domains. This dashboard is useful for OEMs who want to monitor and track their tenants' activity in their Sisense system.

The Usage - Domain dashboard contains data regarding how many domains were active for the recorded period of time, the number of Viewers per domain, how often they are using your dashboards, and what type of dashboard load performance your customers are experiencing.

You can filter the information displayed in your widgets with a set of filters on the right. With these filters, you can filter the details displayed in your dashboard by each domain.
With the Usage - Domain dashboard, you can keep track of how many users you have per domain. For example, in the Users per Domain and Number of Dashboard Viewers per Domain tables, you can view how many users your customers have, and monitor their license utilization.

This dashboard is useful for answering the following questions:

<table>
<thead>
<tr>
<th>Question</th>
<th>Widget</th>
</tr>
</thead>
<tbody>
<tr>
<td>How many domains are using my dashboards?</td>
<td>Active Domains in Period</td>
</tr>
<tr>
<td>How many dashboards are users from each domain viewing?</td>
<td>Avg Viewed Dashboards per Domain, Avg Viewers per Domain, Users per Domain</td>
</tr>
<tr>
<td>How often are dashboards being refreshed by users from the domain?</td>
<td>Avg Refreshes per Dashboard per Domain, Dashboard Refreshes and Viewers</td>
</tr>
<tr>
<td>How many Viewers do I have per domain?</td>
<td>Avg Viewers per Domain, Number of Dashboard Viewers per Domain</td>
</tr>
<tr>
<td>How much are users from each domain interacting with my dashboards?</td>
<td>Avg Actions per Domain, Actions over Time</td>
</tr>
<tr>
<td>How well are my dashboards loading?</td>
<td>Avg Load Time, Dashboard Load Time</td>
</tr>
<tr>
<td>Who is viewing my dashboards?</td>
<td>Dashboard Viewers per Domain</td>
</tr>
</tbody>
</table>
General Analytics

The Usage - General dashboard provides you with insights regarding your Sisense system, including how often Sisense is being used and its performance. You can filter the information displayed in your widgets with a set of filters on the right. With these filters, you can filter the details about your system activity by an individual dashboard or per user or group.

This dashboard is useful for answering the following questions:

<table>
<thead>
<tr>
<th>Question</th>
<th>Widget</th>
</tr>
</thead>
<tbody>
<tr>
<td>How often are dashboards being refreshed?</td>
<td>Avg Refreshes per Day, System Usage</td>
</tr>
<tr>
<td>How many Viewers are active each day?</td>
<td>Avg Viewers per Day, System Usage</td>
</tr>
<tr>
<td>Have many actions occur in my dashboards per day?</td>
<td>Avg Actions per Day, Actions, System Usage</td>
</tr>
<tr>
<td>How long does it take to load a dashboard?</td>
<td>Avg Load Time, Dashboard Load Time,</td>
</tr>
<tr>
<td>Question</td>
<td>Widget</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Which dashboards are being viewed?</td>
<td>Dashboard Viewers</td>
</tr>
<tr>
<td>Who is viewing my dashboards?</td>
<td>Dashboard Viewers</td>
</tr>
<tr>
<td></td>
<td>System Usage</td>
</tr>
</tbody>
</table>
Groups Analytics

The Usage - Group dashboard provides you with insights regarding your Sisense groups. This dashboard is useful to see which groups are active and check the system performance for dashboards for your groups.

The Usage Domain dashboard contains data regarding how many groups were active for the recorded period of time, how often they are using your dashboards, and what type of dashboard load performance your users are experiencing. You can filter the information displayed in your widgets with a set of filters on the right. With these filters, you can filter the details displayed in your dashboard by each user, group, or dashboard.

With the Group Analytics dashboard, you can keep track of your groups and remove or edit any groups that aren’t being used.

This dashboard is useful for answering the following questions:

<table>
<thead>
<tr>
<th>Question</th>
<th>Widget</th>
</tr>
</thead>
</table>

<p>| Active Groups in Period       | 2                        |
| Avg Active Viewers per Group  | 7                        |
| Avg Actions per Group         | 10M                      |
| Avg Dashboard Refreshes per Group | 29,004.59               |
| Avg Load Time per Group       | 31,504.14                |</p>
<table>
<thead>
<tr>
<th>Question</th>
<th>Widget</th>
</tr>
</thead>
<tbody>
<tr>
<td>How many active groups do I have?</td>
<td>Active Groups in Period, Dashboard Refreshes and Viewers</td>
</tr>
<tr>
<td>Can unnecessary groups be deleted?</td>
<td>Active Groups in Period, Avg Active Viewers per Group, Dashboard Viewers per Group</td>
</tr>
<tr>
<td>How many active users are in a group?</td>
<td>Avg Actions per Group, Actions, Dashboard Refreshes and Viewers</td>
</tr>
<tr>
<td>How many actions does a group perform?</td>
<td>Avg Actions per Group, Actions</td>
</tr>
<tr>
<td>How many dashboards were viewed by a group?</td>
<td>Avg Dashboards Viewed per Group</td>
</tr>
<tr>
<td>How many times was a dashboard refreshed by a group?</td>
<td>Avg Dashboards Refreshed per Group</td>
</tr>
<tr>
<td>Whats the dashboard load time for my groups?</td>
<td>Avg Load Time per Group, Load Time</td>
</tr>
<tr>
<td>Who is in my groups?</td>
<td>Dashboard Viewers per Group</td>
</tr>
</tbody>
</table>
User Analytics

The Usage - User dashboard provides you with insights regarding your Sisense users. This dashboard is useful to see which users are active and which users are using a license unnecessarily.

The User Analytics dashboard contains data regarding how many users were active for the recorded period of time, how often they are using your dashboards, and what type of dashboard load performance your users are experiencing. You can filter the information displayed in your widgets with a set of filters on the right. With these filters, you can filter the details displayed in your dashboard by each user, group, or dashboard.

This dashboard is useful for answering the following questions:

<table>
<thead>
<tr>
<th>Question</th>
<th>Widget</th>
</tr>
</thead>
<tbody>
<tr>
<td>How many active users do I have?</td>
<td>Active Users in Period, Dashboard Refreshes and Viewers, User Activity</td>
</tr>
<tr>
<td>Question</td>
<td>Widget</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Can unnecessary users be deleted?</td>
<td>Active Users in Period, Viewed Dashboards per User, User Activity</td>
</tr>
<tr>
<td>How often are my users viewing dashboards?</td>
<td>Avg Refreshes per User</td>
</tr>
<tr>
<td>How many times is a dashboard being refreshed?</td>
<td>Dashboard Refreshes and Viewers</td>
</tr>
<tr>
<td>How many times are users interacting with my dashboards?</td>
<td>Avg Actions per User, User Actions</td>
</tr>
<tr>
<td>How fast are my dashboards loading?</td>
<td>Dashboard Load Time</td>
</tr>
<tr>
<td>Who is viewing my dashboards?</td>
<td>Dashboard Viewer, Viewed Dashboards per Users</td>
</tr>
</tbody>
</table>
Restoring Usage Analytics

You can work with your Usage Analytics ElastiCube and dashboards like you would with any other ElastiCube or dashboard you own.

If you have modified the Usage Analytics ElastiCube and cannot build it anymore, you can restore the default ElastiCube from the Admin page or the REST API. Restoring the ElastiCube or a dashboard returns it to the original state. Any changes you made will be deleted.

To restore your Usage Analytics ElastiCube and dashboards in Sisense:

In the Admin page, click Usage Analytics.
If Collect Usage Analytics is not enabled, toggle the switch to enable it.
Click Restore to for the relevant assets or click both Restore buttons to restore the ElastiCube and dashboards.

To restore your Usage Analytics ElastiCube and dashboards through the REST API:

In the Admin page, click REST API and then select API Reference.
For the ElastiCube, run the following endpoints to restore your Usage Analytics assets:
admin, POST /usageanalytics/restore/cube
For your dashboards, run the following endpoint:
admin, POST /usageanalytics/restore/dashboards
Extracting Usage Analytics Data

You can run SQL queries to extract data from ElastiCubes, including the Usage Analytics ElastiCube. This data is collected and stored in a CSV file. You can download this CSV file with the SQL Runner with the following call:

http://localhost:8081/api/datasources/LocalHost/Usage%20Analytics%20Model/sql?query=select%20*%20from%20usage&format=csv

In this example, LocalHost is the name of the server where the Usage Analytics ElastiCube is located. If you are connecting remotely, change this to the IP address or name of the server as defined in the Data Sources table in the Admin page of Sisense.

Note: Following security enhancements made in Sisense V7.0 and later, this feature is only available to Administrators. You can allow additional user roles to access it by customizing that user's role.

For more information, see SQL Runner.
Managing Plug-ins

Sisense plugins enable you to extend the functionality and capabilities of your Sisense dashboard through standard web development languages such as JavaScript and HTML.

For more information about developing Sisense Plug-ins, see JavaScript API & Plugins.

In the Sisense Admin page, you can view a list of your current plugins, their version as defined in your plug-in’s script, and their status, enabled or disabled.

From the Plug-ins page, you can enable or disable your plugins. This is useful for testing new plugins or troubleshooting issues with your dashboard. When a plug-in is disabled, its functionality is no longer available in the Sisense Web Application.

To enable or disable a plug-in:

In the Sisense Web Application, select Admin > Plugin-Ins.
Toggle the Enable/Disable switch to activate or deactivate a plug-in.

OR

To activate/deactivate multiple plugins, select the checkbox of the relevant plugins. When you select any of the checkboxes, an Enable/Disable switch is displayed in the menu bar. Toggle the switch to enable or disable all the selected plugins.

After enabling or disabling a plug-in, Sisense rebuilds your Plug-ins’ list. The next time you refresh your dashboard, the plugins will be enabled or disabled as defined in the Plug-ins page.
Setting Up a Custom Email Server

Sisense includes a default email server configuration, which uses Mandrill by MailChimp to send automated emails such as dashboard reports or password reset emails. This configuration can be used for POC and testing environments. For production environments, you should provide your own email server. Custom email servers allow all your email transactions to be completely managed on your side.

Through the Sisense REST API, you can send an HTTP POST request to issue Sisense emails through your own custom email server instead of the default Sisense email server. The REST API exposes properties that you can configure for the custom email servers such as security, timeout, and logging properties. When authenticating an HTTP request, Sisense only supports the basic authentication through the Authorization method using plain text.

**To set up a custom email server:**
- In the Sisense Web Application, click Admin, and then Rest API, followed by REST API Reference.
- In version 1.0 of the REST API, select Settings, then POST /settings/email_server.
- In the body of your call, define the values of the following keys:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>enable</td>
<td>If True, allows access to the custom email server instead of the default Sisense email server. <strong>Note:</strong> This property is different from the enable property found in version .9 of the REST API POST /settings/system request. In .9, enable activates Sisense</td>
</tr>
<tr>
<td>Key</td>
<td>Value</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>emails. In version 1.0, enable sends Sisense emails through your custom server.</td>
</tr>
<tr>
<td>host</td>
<td>The hostname or IP address to connect to. By default, this is 'localhost'.</td>
</tr>
<tr>
<td>port</td>
<td>The port to connect to (defaults to 25 or 465).</td>
</tr>
<tr>
<td>secure</td>
<td>When you send an email, some email servers check if secure transport (TLS) is available for that address or domain.</td>
</tr>
<tr>
<td></td>
<td>This value determines if the connection should use TLS (if True) or not (if False). The default value is false. The connection may be upgraded to TLS if the email server requires it.</td>
</tr>
<tr>
<td>ignoreTLS</td>
<td>If this is True and secure is False, TLS will not be used even if the email server attempts to upgrade the connection to TLS-secure.</td>
</tr>
<tr>
<td></td>
<td>This field should be set to True when connecting to a non-secure SMTP server.</td>
</tr>
<tr>
<td>requireTLS</td>
<td>If this is True and secure is False, it forces the connection to be upgraded to TLS-secure even if the email server does not advertise support for TLS.</td>
</tr>
<tr>
<td>connectionTimeout</td>
<td>How many milliseconds to wait for the connection to establish.</td>
</tr>
<tr>
<td>greetingTimeout</td>
<td>How many milliseconds to wait for the greeting after connection is established.</td>
</tr>
<tr>
<td>socketTimeout</td>
<td>How many milliseconds of inactivity to allow.</td>
</tr>
<tr>
<td>Key</td>
<td>Value</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>user</td>
<td>The user name of the user.</td>
</tr>
<tr>
<td></td>
<td>As non-secure or restricted SMTP servers do not require authentication, do not include this field when connecting to a non-secure SMTP server.</td>
</tr>
<tr>
<td>pass</td>
<td>The user’s password for the email server.</td>
</tr>
<tr>
<td></td>
<td>As non-secure or restricted SMTP servers do not require authentication, do not include this field when connecting to a non-secure SMTP server.</td>
</tr>
<tr>
<td>logger</td>
<td>If set to True, Sisense logs events such as which server you connected to. If the value is not defined or is False, then nothing is logged.</td>
</tr>
<tr>
<td>debug</td>
<td>If set to True, then Sisense logs SMTP traffic. When False, Sisense logs only transaction events. Debug logs record everything and may increase in size quickly and impact performance. You should only set activate debug logging when troubleshooting a specific issue and then reset the logging level back to info.</td>
</tr>
</tbody>
</table>

Click **Run**.

After you have set up the email server, you need to change the sender email address to the domain of your custom email server so they are consistent. You can update the sender email through the POST 0.9 /branding endpoint in the SisenseREST API.

```json
{
    "emails": {
        "senderEmail": "username@newdomain.com"
    }
}
```
Click Run.
Activating a Sisense AMI from the AWS Marketplace

Sisense provides a BYOL (Bring Your Own License) AMI (Amazon Machine Image) on the Amazon AWS Marketplace. You can utilize your image to run your Sisense instances on AWS. Before you create an instance, make sure you have your Sisense account user and password available to initialize the instance.

**Note:** The payment to AWS covers the price of the machine on AWS, using an existing Sisense license. Your Sisense license payment is part of the existing Sisense account billing, and is not charged to your AWS account.

**To initialize your Sisense instance:**
Instantiate the Sisense 7.x AMI from the AWS Marketplace to create your Sisense server.
Connect to your Sisense server using Remote Desktop. The Sisense Activation screen is displayed.
Sign in using the your Sisense credentials: the email address and password you provided when acquiring your Sisense license.
Click Sign In to continue. Your account information will be verified, and your server automatically initialized. After completing the initialization, your browser will automatically be launched, displaying the Sisense login window (http://localhost/app/account#/login).
Enter the same user name and password you provided to activate the server, and click Login to login to Sisense server. Your Sisense server is now active and available for your use. The Sisense Web UI is available - and displays the Analytics landing page.

Troubleshooting:
In case you activate Sisense server, and login to it, and the Sisense WebUI isn't displayed, restart your Sisense server. As you have already activated the Sisense instance, your Sisense application is ready. Click Start to launch the application, and display the login screen.
Welcome!
Your Sisense application is ready

Start

Contact Support

Provide your Sisense user and password to begin working with the application.
Getting Started with Linux

Welcome to the Linux Getting Started guide.

Before you begin...

Sisense supports many different types of business users, but this guide is designed for DevOps and IT users.

This guide explains all the steps you need to provision Sisense on Linux, but you should have at least some experience configuring server software and with Linux.

Below are the primary steps required to get Sisense up and running on Linux.

- "Step 1. Verifying System Requirements" on page 1236
- "Step 2. Planning Your Configuration" on page 1240
- "Step 3: Configuring Port Settings" on page 1246
- "Step 4: Initializing Sisense on Linux" on page 1252
- "Step 5: Creating Data Groups" on page 1268
- "Step 6: Monitoring Sisense on Linux" on page 1272

In addition, here are some additional topics you might be interested in:

- Linux FAQs
- Supported Connectors
- Sisense on Windows versus Linux
- Migrating Sisense from Windows to Linux
- Debugging Cloud-Native Sisense on Linux
- Upgrading Sisense on Linux
- Uninstalling Sisense from Linux
- Uploading Files through the File Manager
- Rebranding Sisense Automated Emails
- Translating Sisense Metadata
- White Labeling Sisense
Step 1. Verifying System Requirements

Previous Step
Getting Started with Linux

The following minimum hardware requirements and recommendations apply to all servers running Sisense, including physical hardware and virtual machines. The Sisense Linux deployment is certified to run on the following operating systems:

- Ubuntu 18.04 LTS
- RHEL 7.x
- Centos 7
- Amazon Linux 2.0

The Sisense Linux Cloud-native deployment is currently certified to run on local data centers and on AWS. Certification on Azure and GCP will be provided in the future.

Your servers should meet or exceed the minimum of these hardware recommendations. Note the Linux cloud-native deployments require an additional 4 GB RAM beyond the Sisense requirements for Windows deployments, to handle the cluster infrastructure.

The actual hardware requirements of the cluster depends on many parameters, including the data size, the models created in the system, and the dashboard complexity, and needs to be determined per specific deployment. Additionally, using custom queries in model definitions adds additional compute load to the system, and needs to be considered when planning your deployment.
<table>
<thead>
<tr>
<th>Deployment Type</th>
<th>Processor</th>
<th>CPU</th>
<th>RAM</th>
<th>DISK SPACE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single node</td>
<td>64-bit</td>
<td>8-core, 2.0 GHz or higher</td>
<td>16 GB</td>
<td>50 GB for the first partition. For the second partition or disk /opt/sisense, the space should be two times the amount of your data</td>
</tr>
<tr>
<td>Multi-Node (per server)</td>
<td>64-bit</td>
<td>8-core, 2.0 GHz or higher</td>
<td>16 GB</td>
<td>50 GB for the first partition. For the second partition or disk /opt/sisense, the space should be two times the amount of your data</td>
</tr>
</tbody>
</table>
Additional Requirements

- Root Linux file system ("/") must have at least 50GB of free space
- Sisense recommends that your Linux deployment include AVX2 which is introduced in Haswell
- The default network plugin Calico is required for cluster support for IP-in-IP tunneling in your network. Optionally, you can use the network plugin Weave with encrypted VXLAN
- Servers must be connected to the Internet and must have network access to Dockerhub
- The installation requires the user/password or SSH key. In a multi-node deployment, the same user needs to be defined on all of the servers
- If your deployment includes shared storage with GlusterFS, each node must have a second unformatted unmounted unpartitioned hard disk with at least 30GB disk space available, and additional storage for the ElastiCubes. Sisense recommends that the second disk size be at least the size of all ElastiCubes multiplied by 2.
Next Steps

Step 2. Planning Your Configuration
Step 2. Planning Your Configuration

Previous Step:
Step 1. Verifying System Requirements

Sisense supports two deployment topologies on Linux: a single-node deployment, and a multi-node deployment. This page describes the two supported deployments and the differences between them.

In both deployments, the Sisense application is provided as Docker containers. The orchestration between these containers is managed through Kubernetes as displayed in the diagram below.

Kubernetes Cluster
This diagram illustrates how Sisense utilizes Kubernetes for redundancy and load balancing regardless of the number of machines. When you initialize Sisense, it is installed as a Kubernetes namespace. The services and ElastiCubes that make up the Sisense application are pods spread across the node or nodes in your deployment. In addition, Sisense supports shared storage with GlusterFS or Amazon EFS for data distribution, consistency, and resiliency.
Single-Node

In its most basic configuration, Sisense can run on a single Linux machine. The deployment scenario is straightforward to set up, maintain, and upgrade. The single server deployment will deploy all Sisense nodes on a single server using Kubernetes orchestration. It can support a sandbox environment for development and testing, but it is also a valid configuration for some production environments.

The deployment will utilize the storage available locally on the host server.
**Multi-Node**

Sisense supports the configuration of multiple machines, or nodes. In a multiple node deployment, you have one build-node, two application/query nodes. You can add additional query nodes as needed. In addition, Sisense supports GlusterFS and EFS for shared storage.

As with a single-node deployment, Kubernetes is used to manage the orchestration of Docker containers across a namespace. Kubernetes is also used for load balancing within the Sisense deployment.

The minimum deployment size for a Sisense multi-server deployment is three servers. In three server deployments, two servers will each run as an application and query node. The third server will run as the build node.
In a basic three server deployment, one of the servers is a dedicated build node. The other two servers both provide the functionality of an application and a query node. The infrastructure services must be deployed on each of the three servers, to adequately provide high-availability and redundancy of the infrastructure modules.

Additional query nodes can be added to the deployment, to support additional query load. They will function only as query nodes.

Sisense uses Helm as the installer for the Sisense application. One of the Helm charts labels a Kubernetes node for the application. Helm charts have been configured to add labels to the Kubernetes nodes for application and build nodes. Based on the label, Kubernetes loads the Sisense pods on the correct node on the cluster. Kubernetes also places ElastiCubes on each node according to dynamic labels given in the data group.
Next Steps

Step 3: Configuring Port Settings
Step 3: Configuring Port Settings

Previous Step
Step 2. Planning Your Configuration

Sisense uses certain ports to communicate with machines on the Internet and within your Sisense namespace. Below is a description of the ports that you may need to allow in your deployment.

Note: In cluster deployments, open all traffic between the nodes (TCP and UDP).
Inbound Rules for Sisense

The following ports should be opened to your network so you will be able to access the Sisense application, SSH and Kubernetes dashboard:

<table>
<thead>
<tr>
<th>Ports</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP 443/30845</td>
<td>HTTPS/HTTP WEB (SSL/non-SSL mode). These ports should be open to allow your users to access Sisense.</td>
</tr>
<tr>
<td>TCP 22</td>
<td>SSH. This port should be opened when your Administrator needs to deploy or upgrade Sisense.</td>
</tr>
<tr>
<td>TCP 6443</td>
<td>This port should be opened when your Administrator needs to access the Kubernetes dashboard.</td>
</tr>
</tbody>
</table>
Cluster Mode

When deploying multiple nodes, the following ports should be opened between each node:

<table>
<thead>
<tr>
<th>Ports</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP 2379 - 2380</td>
<td>etcd</td>
</tr>
<tr>
<td>TCP 10248 - 10259</td>
<td>Kubernetes</td>
</tr>
<tr>
<td>TCP 9100</td>
<td>Node exporter</td>
</tr>
</tbody>
</table>
Cluster Network Plugin

Sisense support two cluster network plugins, Calico and Weave. The default network plugin used by Sisense is Calico. Calico and Weave secure the communication between your nodes. The following ports should be opened:

**Calico**

<table>
<thead>
<tr>
<th>Ports</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP 9099</td>
<td>Calico</td>
</tr>
<tr>
<td>TCP 179</td>
<td>Calico - bird</td>
</tr>
</tbody>
</table>

**Weave**

<table>
<thead>
<tr>
<th>Ports</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP 6783</td>
<td>Weave’s control and data</td>
</tr>
<tr>
<td>UDP 6783/6784</td>
<td>Weave’s control and data</td>
</tr>
<tr>
<td>UDP 4789</td>
<td>VXLAN</td>
</tr>
<tr>
<td>TCP 111</td>
<td>rpcbind</td>
</tr>
<tr>
<td>TCP 179</td>
<td>bird</td>
</tr>
</tbody>
</table>

Cluster Shared File System Implementation

Sisense supports GlusterFS or Amazon EFS for shared storage. Depending on which you use, the following ports should be opened:

**GlusterFS**

<table>
<thead>
<tr>
<th>Ports</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP 24007</td>
<td>Gluster Daemon</td>
</tr>
<tr>
<td>TCP 111</td>
<td>rpcbind</td>
</tr>
<tr>
<td>Ports</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>TCP 49152-49251</td>
<td>GlusterFS</td>
</tr>
<tr>
<td>TCP 2049</td>
<td>NFS</td>
</tr>
<tr>
<td>TCP and UDP 24007 - 24008</td>
<td>Gluster servers</td>
</tr>
</tbody>
</table>

**Amazon EFS**

<table>
<thead>
<tr>
<th>Ports</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP 2049</td>
<td>NFS</td>
</tr>
</tbody>
</table>

When using Amazon EFS, make sure that Sisense instances VPC is opened to the EFS security group.

Amazon EFS should be exposed to your server

**Load Balancer**

If you are using an external load balancer, make sure that the load balancer supports WebSockets.

If you are using Amazon AWS with load balancing, ALB supports WebSockets, ELB does not.
Next Steps

**Step 4: Initializing Sisense on Linux**
Step 4: Initializing Sisense on Linux

Note: Your Linux non-root user must have sudo permissions to initialize Sisense. The steps below describe how to initialize Sisense on Linux in a single or multi-node deployment.

To initialize Sisense on Linux, you must download and extract the Sisense Linux archive on your Linux machine. This archive includes a config.yaml, which contains all the configuration settings that are needed to deploy a single-node or multi-node cluster.

From the config.yaml file, you can customize your installation by using various parameters described below. Once the parameters have been defined, you run the script and Sisense will be deployed. You can then retrieve the URL to access the Sisense application online.

**Click here for instructions for Single Node Deployments**

Before you set up Sisense on your single Linux machine, you must mount /opt on a dedicated disk. For instructions on how to mount /opt, [click here](#).

Make sure you have enough disk space on the disk you mount to include the 20GB + 2*(size of the ElasticCube data), No less than 50GB.

**To initialize Sisense in a Linux environment:**

In your Linux CLI, enter the following command to download the Sisense tar.gz file.

```
wget [sisense-linux-deployment-link]
```

Contact Sisense to receive the latest Linux archive file.
Extracting the tar.gz file:
```
tar zxf [sisense-linux-deployment-package-name]
```

Navigate to the directory where you extracted the tar.gz file.
```
cd sisense-[sisense-version]
```

Edit the config.yaml file.
```
vim config.yaml
```

After running this command, the parameters of the config.yaml file are displayed.

In config.yaml file, insert values for the following parameters:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>k8s_nodes</td>
<td>K8S node/nodes are the set of machines that are used to run Sisense. The installation machine is used only during installation to run the installation scripts. The Installation machine can be one of the K8S nodes, but it can also be a different machine (remote installation). This parameter defines the following: <strong>node</strong>: name of your node. <strong>internal_ip</strong>: The internal IP addresses define the location within your Kubernetes cluster. <strong>external_ip</strong>: The external IP address defines how your cluster can be accessed from outside, or the DNS name of the node. It can also be the same value as the internal IP in case this is the only IP used to access the node.</td>
</tr>
<tr>
<td>remote_installation</td>
<td>If you are running your installation from a Bastion machine, part of the K8S nodes, or remotely, enter true.</td>
</tr>
<tr>
<td>update</td>
<td>If you are upgrading Sisense, enter true, otherwise, keep this value as the default. If you enter true, Sisense skips the Kubernetes installation. If</td>
</tr>
</tbody>
</table>
### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>your Kubernetes cluster is already configured and running, keep <strong>yes</strong> to skip cluster installation. Staging and dev environments can run on the same cluster as a separate namespace. For installing on a separate namespace, the value of <strong>namespace_name</strong> needs to be update.</td>
<td></td>
</tr>
<tr>
<td>application_dns_name</td>
<td>Enter your DNS name. The default is the first node of your external IP. If you have not defined a secure connection (No SSL), Sisense uses the external IP of your first node when accessing Sisense.</td>
</tr>
<tr>
<td>linux_user</td>
<td>Enter the name of your Linux user. This user must not be the &quot;root&quot; user, but should have sudo privileges, and all the other privileges as a root user.</td>
</tr>
<tr>
<td>password</td>
<td>Enter the password of your Linux user or ssh_key.</td>
</tr>
<tr>
<td>ssh_key</td>
<td>If you have a secure connection to your server, enter the SSH key of the Linux user defined in <strong>sisense_user</strong>. The SSH key should be in .pem format.</td>
</tr>
<tr>
<td>is_kubernetes_cloud</td>
<td>For single node deployments, leave this field empty.</td>
</tr>
<tr>
<td>storage_type</td>
<td>For single deployments, leave this field empty as Sisense use the host path /opt/sisense.</td>
</tr>
<tr>
<td>nfs_server</td>
<td>For single node deployments, leave this field empty.</td>
</tr>
<tr>
<td>nfs_path</td>
<td>For single node deployments, leave this field empty.</td>
</tr>
<tr>
<td>efs_file_system_id</td>
<td>For single node deployments, leave this field empty.</td>
</tr>
<tr>
<td>Parameters</td>
<td>Value</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>efs_aws_region</td>
<td>For single node deployments, leave this field empty.</td>
</tr>
<tr>
<td>fsx_file_system_id</td>
<td>For single node deployments, leave this field empty.</td>
</tr>
<tr>
<td>fsx_region</td>
<td>For single node deployments, leave this field empty.</td>
</tr>
<tr>
<td>sisense_disk_size</td>
<td>For single node deployments, leave this field empty.</td>
</tr>
<tr>
<td>mongodb_disk_size</td>
<td>For single node deployments, leave this field empty.</td>
</tr>
<tr>
<td>zookeeper_disk_size</td>
<td>For single node deployments, leave this field empty.</td>
</tr>
<tr>
<td>namespace_name</td>
<td>Enter the name of the Kubernetes namespace.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Kubernetes Ports should be released (Non-listening mode).</td>
</tr>
<tr>
<td>gateway_port</td>
<td>Enter the port of the API gateway for your deployment.</td>
</tr>
<tr>
<td></td>
<td>If you are not implementing SSL, this will be the port used to</td>
</tr>
<tr>
<td></td>
<td>connect to Sisense.</td>
</tr>
<tr>
<td>is_ssl</td>
<td>Enter true for secure connections to Sisense. Enter false if you</td>
</tr>
<tr>
<td></td>
<td>have not implemented SSL.</td>
</tr>
<tr>
<td>ssl_cn_name</td>
<td>Enter the Host name (CN name) the certificate is issued for.</td>
</tr>
<tr>
<td></td>
<td>Example: test.sisense.com</td>
</tr>
<tr>
<td>ssl_key_path</td>
<td>If you connect to your server securely, enter the SSL keypath.</td>
</tr>
<tr>
<td></td>
<td>When SSL is defined, the Sisense API Gateway Port will be 443 and</td>
</tr>
<tr>
<td></td>
<td>not the value defined in gateway_port.</td>
</tr>
<tr>
<td>ssl_cer_path</td>
<td>If you connect to your server securely, enter the SSL certificate</td>
</tr>
<tr>
<td>Parameters</td>
<td>Value</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>path (.crt file).</td>
<td></td>
</tr>
<tr>
<td>weave_enabled</td>
<td>For single node deployments, leave this field empty.</td>
</tr>
<tr>
<td>weave_password</td>
<td>For single node deployments, leave this field empty.</td>
</tr>
<tr>
<td>new_node</td>
<td>For single node deployments, leave this field empty.</td>
</tr>
<tr>
<td>remove_node</td>
<td>For single node deployments, leave this field empty.</td>
</tr>
<tr>
<td>node_to_remove</td>
<td>For single node deployments, leave this field empty.</td>
</tr>
<tr>
<td>recover_kubernetes</td>
<td>Enter <strong>true</strong> if one of your nodes is broken. Sisense removes the node and recovers it.</td>
</tr>
<tr>
<td>http_proxy</td>
<td>If you must use a web proxy, set this to the unsecured proxy value.</td>
</tr>
<tr>
<td>https_proxy</td>
<td>If you must use a web proxy, set this to the secured proxy value.</td>
</tr>
<tr>
<td>no_proxy</td>
<td>If you must use a web proxy, to exclude certain addresses from your proxy, enter the IP addresses separated by a comma.</td>
</tr>
<tr>
<td>internal_monitoring</td>
<td>Enter <strong>false</strong> to disable a Grafana dashboard that is supported by Prometheus for monitoring your deployment.</td>
</tr>
<tr>
<td>external_monitoring</td>
<td>Enter <strong>false</strong> to disable external monitoring with your Logz.io account. If you set this value to yes, provide values for sisense_ownerid and logz_key.</td>
</tr>
<tr>
<td>is_efk</td>
<td>Enter <strong>true</strong> to add Docker Logging via EFK (Elasticsearch + Fluentd + Kibana). Sisense preconfigures these to work</td>
</tr>
</tbody>
</table>
Parameters | Value
--- | ---
 together, but does not provide a dashboard. For more information, see [Monitoring Sisense in Linux](#).
uninstall_cluster | Enter true to remove Kubernetes infrastructure and Sisense services.
uninstall_sisense | Enter true to uninstall Sisense services, but leave your Kubernetes infrastructure in case it's needed in the future.
remove_user_data | Enter true to delete all user data. This deletes your ElastiCube models, application database, message broker, and plug-ins.

Run the configuration script.
`. /sisense.sh config.yaml`

Your configuration settings are displayed with a message to confirm that you want to deploy Sisense with these settings.

Enter **Yes** to confirm that you want to deploy Sisense. Enter **No** to abort the deployment. If you entered **Yes**, Sisense will be deployed. If there are any issues, you can view the installation logs here: [installation-dir]/sisense-ansible.log. When this installation is complete a list of endpoints are displayed for accessing Sisense and managing your deployment. The URLs are listed below. In addition, you can run the following command to return the URL to access Sisense.

`kubectl cluster-info`

This displays the URL of your Sisense application. You can enter this address into your browser to access Sisense. To verify that all your services are running as expected, you can enter the URL of Sisense with the port and `/app/test` to the end of the address in your browser. This displays the status of each of your services. For example, `0.0.0.0:PORT/app/test`.

**List of endpoints in the single installation:**

To connect to Sisense, in your browser, enter the following in your browser:

**For non-secure connections:**
`http://{IP}:30845/`

**For secure connections:**
`https://{IP}/`
To connect to your Kubernetes dashboard, enter the following in your browser:
https://{IP}:6443/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy

Click here for instructions for Multi-Node Deployments

To initialize Sisense in a Linux environment:
In your Linux CLI, enter the following command to download the Sisense tar.gz file.

```bash
wget [sisense-linux-deployment-link]
```

Contact Sisense to receive the latest Linux archive file.

Extracting the tar.gz file:

```bash
tar zxf [sisense-linux-deployment-package-name]
```

Navigate to the directory where you extracted the tar.gz file.

```bash
cd sisense-[sisense-version]
```

Edit the config.yaml file.

```bash
vim config.yaml
```

After running this command, the parameters of the config.yaml file are displayed.

In the config.yaml file, insert values for the following parameters:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>k8s_nodes</td>
<td>K8S node/nodes are the set of machines that will be used to run Sisense.</td>
</tr>
<tr>
<td></td>
<td>The installation machine is used only during installation to run the installation scripts.</td>
</tr>
<tr>
<td></td>
<td>The Installation machine can be one of the K8S nodes, but it can also be a different machine.</td>
</tr>
<tr>
<td></td>
<td>This parameter defines the IP address of your internal and external IP address.</td>
</tr>
<tr>
<td></td>
<td>If you are implementing a multi-node deployment, remove the comments from # k8s_nodes and</td>
</tr>
<tr>
<td></td>
<td>define the internal IP, external IP, disk_volume_device, and role for each node.</td>
</tr>
<tr>
<td></td>
<td>The internal IP addresses define the location within your</td>
</tr>
<tr>
<td>Parameters</td>
<td>Value</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Kubernetes cluster.</td>
<td>The <strong>external IP</strong> address defines how your cluster can be accessed from outside, or the DNS name of the node. It can also be the same value as the internal IP in case this is the only IP used to access the node.</td>
</tr>
<tr>
<td></td>
<td>If you are using GlusterFS for shared storage, <strong>disk_volume_device</strong> is the location where ElastiCubes and other shared assets are stored. You need to define this value for each node. To retrieve the value, run the <code>lsblk</code> command on each node and enter the location returned for each node.</td>
</tr>
<tr>
<td></td>
<td>The <strong>role</strong> defines the role of the node. There are three possible values: <strong>query, application, and build.</strong></td>
</tr>
<tr>
<td>remote_installation</td>
<td>If you are running your installation from a Bastion machine, part of the K8S nodes, or remotely, enter <strong>yes.</strong></td>
</tr>
<tr>
<td>update</td>
<td>If you are upgrading Sisense, enter <strong>yes</strong>, otherwise, keep this value as the default.</td>
</tr>
<tr>
<td></td>
<td>If you enter <strong>yes</strong>, Sisense skips the Kubernetes installation. If your Kubernetes cluster is already configured and running, keep <strong>yes</strong> to skip cluster installation.</td>
</tr>
<tr>
<td></td>
<td>Staging and dev environments can run on the same cluster as a separate namespace. Make sure you have enough storage in the GlusterFS for a second namespace. For installing on a separate namespace, the value of <strong>namespace_name</strong> needs to be update.</td>
</tr>
<tr>
<td>application_dns_name</td>
<td>Enter your DNS name. The default is the first node of your external IP.</td>
</tr>
</tbody>
</table>
|                          | If you have not defined a secure connection (No SSL), Sisense
<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>uses the external IP of your first node when accessing Sisense.</td>
</tr>
<tr>
<td>linux_user</td>
<td>Enter the name of your Linux user. This user must not be the &quot;root&quot; user, but should have sudo privileges, and all the other privileges as a root user.</td>
</tr>
<tr>
<td>password</td>
<td>Enter the password of your Linux user or ssh_key.</td>
</tr>
<tr>
<td>ssh_key</td>
<td>If you have a secure connection to your server, enter the SSH key of the Linux user defined in sisense_user. The SSH key should be in .pem format.</td>
</tr>
<tr>
<td>is_kubernetes_cloud</td>
<td>Enter true if you already have a Kubernetes cluster. Sisense supports the following Kubernetes clusters:</td>
</tr>
<tr>
<td></td>
<td>Amazon EKS</td>
</tr>
<tr>
<td></td>
<td>Google GKE</td>
</tr>
<tr>
<td></td>
<td>Azure AKS</td>
</tr>
<tr>
<td></td>
<td>Click the links above for instructions on how to set up Sisense on the relevant cluster type.</td>
</tr>
<tr>
<td></td>
<td>If you do not have a Kubernetes cluster, enter false and skip the next few parameters until storage_type.</td>
</tr>
<tr>
<td>storage_type</td>
<td>Enter one of the following values:</td>
</tr>
<tr>
<td></td>
<td><strong>If you are using the Sisense installed Kubernetes (kubespray)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>glusterfs</strong>: For using GlusterFS with the three second disks, you defined in disk_volume_device on the first 3 machines.</td>
</tr>
<tr>
<td></td>
<td><strong>nfs</strong>: For using an external NFS server as the shared storage. You must define nfs_server and nfs_path and have root read/write permissions on the path from the cluster hosts.</td>
</tr>
<tr>
<td></td>
<td><strong>efs</strong>: For using Amazon EFS, you must enter values for</td>
</tr>
<tr>
<td>Parameters</td>
<td>Value</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>efs_file_system_id</td>
<td>and efs_aws_region.</td>
</tr>
<tr>
<td>fsx:</td>
<td>For using Amazon FSx luster, you must enter values for</td>
</tr>
<tr>
<td>fsx_file_system_id</td>
<td>and fsx_region.</td>
</tr>
<tr>
<td>If you provide your own Kubernetes</td>
<td></td>
</tr>
<tr>
<td>azurefile:</td>
<td>For Azure AKS, use “Azure Files” storage service and enter azurefile. Define the values nfs_server and nfs_path below.</td>
</tr>
<tr>
<td>nfs:</td>
<td>For Google GKE, use “Google Cloud Filestore” storage service and enter nfs. You must have your own Filestore and enter values for nfs_server and nfs_path.</td>
</tr>
<tr>
<td>efs:</td>
<td>For Amazon EKS, enter efs. You must enter values for</td>
</tr>
<tr>
<td>efs_file_system_id</td>
<td>and efs_aws_region.</td>
</tr>
<tr>
<td>fsx:</td>
<td>For Amazon EKS, enter fsx for using Amazon FSx luster. You must enter values for</td>
</tr>
<tr>
<td>fsx_subnet_id</td>
<td>and fsx_security_group_id.</td>
</tr>
</tbody>
</table>

<p>| nfs_server         | Enter the address (IP or DNS) of your NFS server.                                       |
| nfs_path           | Enter the location of your NFS server.                                                 |
|                    | The mounting point for each logical disk (Sisense app, MongoDB, and Zookeeper) will be created under this path. |
| efs_file_system_id  | Enter your EFS File System ID.                                                         |
| efs_aws_region     | Enter your EFS AWS region.                                                              |
| fsx_file_system_id  | Enter your Amazon FSx luster file system ID.                                            |
| fsx_region         | Enter your Amazon FSx luster file system region.                                       |</p>
<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>sisense_disk_size</td>
<td>If you have implemented GlusterFS for storage, enter the amount of disk space in gigabytes to be allocated for Sisense. <strong>Important!</strong> You need to provide enough space to support your Sisense ElastiCube models. At least two times the amount of data in all ElastiCubes. GlusterFS is set for two replicas so assume the same logical disk is in two of the physical disks you set in the disk_volume_device. Every namespace takes additional logical disk space from the physical disks. If you have not allocated enough space, the installation will fail. You can use the following equation: sisense_disk_size = 50GB x 3. (In a three node deployment) If you allocate space for the application DB and configuration DB (the values of mongodb_disk_size and zookeeper_disk_size) this should be considered as well. If you enter 150GB as the value, this allocates 50GB in a 3 node deployment minus 3 times (the values of mongodb_disk_size and zookeeper_disk_size). Sisense also recommends that you specify an additional 5GB free space for the value of sisense_disk_size.</td>
</tr>
<tr>
<td>mongodb_disk_size</td>
<td>The amount of disk space allocated for the Sisense application database. This value should be multiplied by the number of nodes your deployment has. It is recommended to leave the default of 3 GB. If only metadata is stored in the MongoDB, there is no need to increase the size.</td>
</tr>
<tr>
<td>zookeeper_disk</td>
<td>The amount of disk space allocated for the ZooKeeper service.</td>
</tr>
<tr>
<td>Parameters</td>
<td>Value</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>_size</td>
<td>This value should be multiplied by the number of nodes your deployment has. It is recommended to leave the default of 1 GB. If only metadata is stored in Zookeeper, there is no need to increase the size.</td>
</tr>
<tr>
<td>namespace_name</td>
<td>Enter the name of the Kubernetes namespace. If you have multiple deployments, for example, for a development and production environment, you should have a unique namespace for each deployment. In addition, for multiple deployments, each should have a unique gateway_port value, and for each deployment after the first, the value of update should be set to true. <strong>Note:</strong> Kubernetes Ports should be released (Non-listening mode)</td>
</tr>
<tr>
<td>gateway_port</td>
<td>Enter the port of the API gateway for your deployment. If you are not implementing SSL, this will be the port used to connect to Sisense.</td>
</tr>
<tr>
<td>is_ssl</td>
<td>Enter <strong>true</strong> for secure connections to Sisense. Enter <strong>false</strong> if you have not implemented SSL.</td>
</tr>
<tr>
<td>ssl_cn_name</td>
<td>Enter the Host name (CN name) the certificate is issued for. Example: test.sisense.com</td>
</tr>
<tr>
<td>ssl_key_path</td>
<td>If you connect to your server securely, enter the SSL keypath. When SSL is defined, the Sisense API Gateway Port will be 443 and not the value defined in gateway_port.</td>
</tr>
<tr>
<td>ssl_cer_path</td>
<td>If you connect to your server securely, enter the SSL certificate path (.crt file).</td>
</tr>
<tr>
<td>Parameters</td>
<td>Value</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>weave_enabled</td>
<td>Enter <strong>true</strong> if your cluster uses Azure virtual machines or any other place in your deployment does not support IP in IP network protocol. The default network plugin is calico. Sisense supports the Weave network plugin for communication between nodes. You must include your Weave password in the value for <strong>weave_password</strong> below. Enabling weave will change the network protocol between the nodes to secured vXLAN. If the weave_password is empty, the vXLAN will not be encrypted.</td>
</tr>
<tr>
<td>weave_password</td>
<td>Enter your Weave password. To disable encryption on the vXLAN enter an empty string.</td>
</tr>
<tr>
<td>new_node</td>
<td>If you have already provisioned Sisense and have a working cluster, you can add additional nodes. Enter <strong>true</strong> to add new nodes. The new nodes should be defined under <code>k8s_nodes</code> above in addition to your existing nodes. Adding new nodes does not affect your existing nodes.</td>
</tr>
<tr>
<td>remove_node</td>
<td>Enter <strong>true</strong> to remove a node from your cluster. Enter the node to be removed in the value of <strong>node_to_remove</strong> below.</td>
</tr>
<tr>
<td>node_to_remove</td>
<td>Enter the name of a node to be removed. You can remove multiple nodes by entering each node separated by a comma; <code>node_1, node_2</code>.</td>
</tr>
<tr>
<td>recover_kubernetes</td>
<td>Enter <strong>true</strong> if one of your nodes is broken. Sisense removes the node and recovers it.</td>
</tr>
<tr>
<td>http_proxy</td>
<td>If you must use a web proxy, set this to the unsecured proxy.</td>
</tr>
<tr>
<td>Parameters</td>
<td>Value</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>https_proxy</td>
<td>If you must use a web proxy, set this to the secured proxy value.</td>
</tr>
<tr>
<td>no_proxy</td>
<td>If you must use a web proxy, to exclude certain addresses from your proxy, enter the IP addresses separated by a comma. You need to include the internal IP address of the host in the list.</td>
</tr>
<tr>
<td>internal_monitoring</td>
<td>Enter <strong>false</strong> to disable a Grafana dashboard that is supported by Prometheus for monitoring your deployment. For more information, see <a href="#">Monitoring Sisense in Linux</a>.</td>
</tr>
<tr>
<td>external_monitoring</td>
<td>Enter <strong>false</strong> to disable external monitoring with your Logz.io account. If you set this value to <strong>true</strong>, provide values for sisense_ownerid and logz_key. For more information, see <a href="#">Monitoring Sisense in Linux</a>.</td>
</tr>
<tr>
<td>is_efk</td>
<td>Enter <strong>false</strong> to disable Docker Logging via EFK (Elasticsearch + Fluentd + Kibana). Sisense preconfigures these to work together, but does not provide a dashboard. For more information, see <a href="#">Monitoring Sisense in Linux</a>.</td>
</tr>
<tr>
<td>uninstall_cluster</td>
<td>Enter <strong>true</strong> to remove Kubernetes infrastructure and Sisense services.</td>
</tr>
<tr>
<td>uninstall_sisense</td>
<td>Enter <strong>true</strong> to uninstall Sisense services, but leave your Kubernetes infrastructure in case it's needed in the future. This can also be used if you need to remove Sisense from your own cloud-based cluster without impacting the cluster itself. Only Sisense the application is removed.</td>
</tr>
<tr>
<td>remove_user_data</td>
<td>Enter <strong>true</strong> to delete all user data. This deletes your ElastiCube models, application database, message broker, and plug-ins.</td>
</tr>
</tbody>
</table>
Run the configuration script.
./sisense.sh config.yaml

Your configuration settings are displayed with a message to confirm that you want to deploy Sisense with these settings.
Enter **Yes** to confirm that you want to deploy Sisense. Enter **No** to abort the deployment. If you entered **Yes**, Sisense will be deployed. If there are any issues, you can view the installation logs here: [installation-dir]/sisense-ansible.log. When this installation is complete a list of endpoints are displayed for accessing Sisense and managing your deployment. The URLs are listed below. In addition, you can run the following command to return the URL to access Sisense.
kubectl cluster-info
This displays the URL of your Sisense application. You can enter this address into your browser to access Sisense. To verify that all your services are running as expected, you can enter the URL of Sisense with the port and /app/test to the end of the address in your browser. This displays the status of each of your services. For example, 0.0.0.0:PORT/app/test.

**List of endpoints in the single installation:**

To connect to Sisense, in your browser, enter the following in your browser:

**For non-secure connections:**
http://{IP}:30845/

**For secure connections:**
https://{IP}/

To connect to your Kubernetes dashboard, enter the following in your browser:
https://{IP}:6443/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy
Next Steps

Step 5: Creating Data Groups
Step 5: Creating Data Groups

Previous Step
   Step 4: Initializing Sisense on Linux

If you choose to set up a multi-node deployment, your ElastiCubes are stored in a shared storage. For each ElastiCube, there is one instance stored on your shared storage resource. You can create multiple instances which are loaded in memory on the other machines in your deployment. Creating multiple instances of ElastiCubes can improve performance.

To create and manage each of the instances of your ElastiCubes, you can create Data Groups in Sisense. Data Groups are collections of ElastiCubes that make it easy to manage your resources for multiple instances of ElastiCubes. Data Groups allow you to assign and split a system’s resources between the ElastiCubes according to your use-case.

For example, let’s assume you have several high priority ElastiCubes with a lot of concurrent users, and some other ElastiCubes with fewer users with data that doesn’t change very often. You can group your high priority ElastiCubes into one data group, and your lower priority ElastiCubes in another group. For the first group, you can create multiple instances of your ElastiCubes on each of the machines in your deployment and allocate more resources to this data group. For the lower priority group, you may want to create fewer instances and provision these instances in memory on one of your slower machines.

To create a data group:
   In the Sisense Admin page, select Data Groups.
In the Data Groups area, click + Data Group.

**Add Data Group**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Name</td>
<td>Enter a name for your data group.</td>
</tr>
<tr>
<td>Build Node</td>
<td>Select the build node where you want to host an instance of an ElastiCube.</td>
</tr>
<tr>
<td>Query Nodes</td>
<td>Select the query nodes where you want to host instances of your ElastiCubes.</td>
</tr>
<tr>
<td>ElastiCubes</td>
<td>Select the ElastiCubes that are to be assigned to your data group.</td>
</tr>
</tbody>
</table>

(Optional) Click More to display advanced options for specifying resource allocation for your nodes. For the relevant nodes, enter values for the following:

- **Reserved Cores**: The amount of CPU cores dedicated to an ElastiCube on a node.
- **Max Cores**: The maximum amount of CPU cores dedicated to an ElastiCube on a node.
- **Reserved RAM**: The amount of RAM dedicated to an ElastiCube on a node.
- **Max RAM**: The maximum amount of RAM dedicated to an ElastiCube on a node.

Select Set as Default to add new ElastiCube models to this group.
Click **Apply** to create the data group. After the group is created, you can monitor its status under the **System Management** tab in the **Admin** page of Sisense.
Next Steps

Step 6: Monitoring Sisense on Linux
Step 6: Monitoring Sisense on Linux

Previous Step
  Step 5: Creating Data Groups

Sisense comes with several tools for monitoring the health of your deployment that allow you to troubleshoot any issues. These tools allow you to monitor the following areas:
  - Dashboard performance and ElastiCube build times
  - Sisense cluster status
  - Sisense containers in your Kubernetes namespace
Sisense Monitor

Sisense Monitor provides information about dashboard performance and your ElastiCube builds. For example, you can identify if too many concurrent builds are affecting query performance and then take corrective action such as scheduling builds throughout the day. For more information, see Sisense Monitor. To access Sisense Monitor, contact your CSM.
System Management

In the **Admin** page of Sisense, you can view the health of your nodes from the **System Management** tab. Within the System Management tab, you can view which nodes are active, the status of your system, and a description of the system's hardware specifications.
Grafana

Sisense provides a Grafana dashboard on Prometheus for monitoring your Sisense Linux cluster. This dashboard allows you to track performance and usage of different pods across different namespaces, investigate performance issues from a centralized location.

You can access your Grafana from the System Management tab:

OR

In your browser, type http://{IP}:{(random _port*)}/login

You can retrieve the port with the following command: kubectl get svc -n monitoring prom-operator-grafana
EFK (Elasticsearch, FluentD and Kibana)

Elasticsearch, FluentD and Kibana (EFK) is a combined logging mechanism that lets you build dashboards for monitoring important parts of your deployment from metrics collected by FluentD. If you provisioned EFK (is_efk = true) when you initialized Sisense on Linux, Sisense activated EFK. This means that Sisense began collecting metrics about your deployment that you can access through Kibana. Keep in mind that Sisense does not provide a Kibana dashboard, only access to Kibana with metrics collected by FluentD. You can access Kibana and create your own dashboard with the collected metrics by entering the IP of your node, and port 30561 in your browser.

NodeIPaddress:30561
Kubernetes Dashboard

The Kubernetes Dashboard provides an interface for troubleshooting Sisense containers, and manage the cluster's resources. You can use the dashboard to get an overview of how Sisense is running on your cluster as well as monitoring Sisense resources.

To access your Sisense Kubernetes dashboard, enter the following URLs into your browser:

Single Node Deployments
https://{IP}:6443/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy

Multi-Node Deployments
https://{IP}:6443/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxyqueue

Note: {IP} should be replaced with the IP address you defined in the config.yaml file when you provisioned Sisense.
# Linux Connectors

The table below contains a list of connectors supported in Linux and what version their support started in:

<table>
<thead>
<tr>
<th>Connector</th>
<th>Support ed From</th>
<th>ElastiCube</th>
<th>Live Models</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSV</td>
<td>V7.3 Linux Beta</td>
<td>Documentati on</td>
<td></td>
<td>ElastiCube models only. Live not currently supported</td>
</tr>
<tr>
<td>Excel</td>
<td>V7.3 Linux Beta</td>
<td>Documentati on</td>
<td></td>
<td>ElastiCube models only. Live not currently supported</td>
</tr>
<tr>
<td>Generic JDBC</td>
<td>V7.3 Linux Beta</td>
<td>Documentati on</td>
<td></td>
<td>ElastiCube models only. Live not currently supported</td>
</tr>
<tr>
<td>Connector</td>
<td>Supported From</td>
<td>ElastiCube</td>
<td>Live Models</td>
<td>Comments</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------</td>
<td>------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Google Ads</td>
<td>V7.4.3 Linux</td>
<td>Documentati</td>
<td></td>
<td>ElastiCube models only. Live not currently supported</td>
</tr>
<tr>
<td>Google Analytics</td>
<td>V7.4.1 Linux</td>
<td>Documentati</td>
<td></td>
<td>ElastiCube models only. Live not currently supported</td>
</tr>
<tr>
<td>Google BigQuery</td>
<td>V7.3 Linux Beta</td>
<td>Documentati</td>
<td>Documentati</td>
<td></td>
</tr>
<tr>
<td>Google Sheets</td>
<td>V7.3 Linux Beta</td>
<td>Documentati</td>
<td></td>
<td>ElastiCube models only. Live not currently supported</td>
</tr>
<tr>
<td>Connector</td>
<td>Supported From</td>
<td>ElastiCube</td>
<td>Live Models</td>
<td>Comments</td>
</tr>
<tr>
<td>----------------</td>
<td>----------------</td>
<td>------------</td>
<td>-------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>MemSQL</td>
<td>V7.4.3 Linux</td>
<td>Documentati on</td>
<td>Documentati on</td>
<td>d</td>
</tr>
<tr>
<td>MySQL</td>
<td>V7.3 Linux Beta</td>
<td>Documentati on</td>
<td>Documentati on</td>
<td></td>
</tr>
<tr>
<td>Oracle</td>
<td>V7.3 Linux Beta</td>
<td>Documentati on</td>
<td>Documentati on</td>
<td></td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>V7.3 Linux Beta</td>
<td>Documentati on</td>
<td>Documentati on</td>
<td></td>
</tr>
<tr>
<td>Redshift</td>
<td>V7.3 Linux Beta</td>
<td>Documentati on</td>
<td>Documentati on</td>
<td></td>
</tr>
<tr>
<td>SalesForce</td>
<td>V7.3 Linux Beta</td>
<td>Documentati on</td>
<td>Documentati on</td>
<td>ElastiCube models only. Live not currently supported</td>
</tr>
<tr>
<td>Snowflake</td>
<td>V7.3 Linux</td>
<td>Documentati on</td>
<td>Documentati on</td>
<td></td>
</tr>
<tr>
<td>Connector</td>
<td>Support ed From</td>
<td>ElastiCube</td>
<td>Live Models</td>
<td>Comments</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------</td>
<td>------------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>SQL Server</td>
<td>Beta</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>V7.3 Linux Beta</td>
<td><a href="#">Documentation</a></td>
<td><a href="#">Documentation</a></td>
<td></td>
</tr>
</tbody>
</table>
Linux Frequently Asked Questions

Does the Linux deployment support single and multi-server deployments?
The Linux deployment supports both single and multi-server deployments, and such deployments have already being installed in the Linux program.

Which type of multi-server deployments are supported on Linux?
Sisense supports: 1 build node, 2 app/query nodes, and 0-n additional query/build nodes. Other multi-server deployment models are not supported.

Which flavor of Linux is supported?
Currently Ubuntu 18.04 has been certified for Sisense deployments. We may certify additional Linux flavors in the future.
Reference: http://releases.ubuntu.com/18.04.2/ubuntu-18.04.2-live-server-amd64.iso

Will we continue to support Windows deployments after the Linux release?
We plan to continue to support Windows deployments following the Linux release.

Do Linux deployments support all features of the Windows deployment?
The Linux release supports the same general feature set as Windows, with some differences:
  Only the Data page is available for creating and editing Sisense models. The desktop ElastiCube manager is not supported on Linux at all.
Only connectors supported by the Data page (web-based ElastiCube manager) are supported by Linux. PSM.exe is not supported on Linux. The functionality provided by PSM.exe is available over new Linux CLI and/or REST API. ElastiCube to ElastiCube import is not supported in the initial release.

**Which connectors are supported on Linux?**

The following connectors are already supported in Sisense Linux:

- MySQL
- SQL Server
- PostgreSQL
- Redshift
- Oracle
- Generic JDBC (ElastiCube models only, no Live models)
- CSV (ElastiCube models only, no Live models)
- Excel (ElastiCube models only, no Live models)
- Salesforce (ElastiCube models only, no Live models)
- Google Sheets (ElastiCube models only, no Live models)
- Google BigQuery
- Snowflake

Additional connectors will be supported in future releases.

**What will our users experience when their server is moved to Linux?**

Sisense on Linux provides a web UI similar to the Windows web UI. The changes users experience depends on their roles, as follows:

- **Viewers:** No change at all.
- **Designers:** Minor changes. The concept of ElastiCube Sets doesn’t exist in Linux. Dashboards are created on ElastiCubes, also in multi-server deployments. Except for that, no change.
- **Data Designers:** No concept of remote servers. All Models are created on the same cluster. Except for that, no change.
- **Data Admins:** No concept of remote servers. All Models are created on the same cluster.
- **Administrators:** Administrator users are the main role who will experience a difference when working on Linux. The main difference is the installation and upgrade, which are completely different in Linux. The method for installing all extensions (plugins, email templates, rebranding files) is different. Some administrative activities are different, for example the mechanism for configuring and defining high-availability has completely
changed. There is a new concept of Data Groups for managing ElastiCube orchestration and resources replacing the Windows High-Availability orchestrator. Some administrative activities are exactly the same on Linux as on Windows, for example, User Management, Group creation, Active Directory, and more.

**Is there a migration process from Windows to Linux?**

There is an automatic migration process from Windows Sisense V7.2 and later to Windows. The process migrates almost all the assets customers created, including: User management (users, groups, data security rules), ElastiCubes, dashboards, plugins, email templates, languages, and rebranding files. Customers will need to upgrade to Sisense V7.1 or later to use the migration tool.

**How long does it take for the migration process to run?**

The migration process takes between 10 minutes to a few hours depending on the size of the data that needs to be migrated.

Sisense recommend that you migrate only the ElastiCube metadata without the ElastiCube data (.ecdata files) to shorten the migration duration. In that case, following the migration, “build entire” is required on the ElastiCubes migrated to the Linux system before the dashboards can be viewed.

**How long does it take to provision a Linux deployment?**

The installation of a single node and of multi-node deployments takes up to 10 minutes as long as the environment has a good internet connection.

**How does the deployment work?**

The installation is a relatively small tar.gz file containing a set of scripts/ansible/helm configuration files and definitions for installing the system. Most of the system is installed from a Docker hub and public Ubuntu repositories using apt get.

**What is the required servers size?**

The Linux installation requires a similar machine spec to the Windows servers used by existing deployments, with an additional 4GB of memory for the cluster management.
How should a single server Linux be installed?

Sisense recommends the Linux server be installed with 2 disks or at least two partitions:
- The Linux disk should have at least 50GB.
- The second disk is used for the ElastiCubes. It’s size depends on the data size mounted on /opt/sisense/storage.
  - **Minimal Size**: 100GB
  - **Recommended**: At least the size of all the ElastiCubes * 2.5

How should the servers of a multi-server Linux deployment be installed?

Sisense recommends that Linux be installed with 2 disks or at least two partitions:
- The Linux disk should have at least 50GB.
- The second disk is used for the ElastiCubes. Its size depends on the data size mounted on /opt/sisense.
  - **Minimal Size**: 100GB
  - **Recommended**: At least the size of all the ElastiCubes * 2.5

For more information on the GlusterFS, see: [https://docs.gluster.org/en/latest/](https://docs.gluster.org/en/latest/) and [https://github.com/heketi/heketi](https://github.com/heketi/heketi)

Which user is needed in order to configure and deploy the Linux server?

Any user is good enough as long as they have sudo permissions. The installation requires the user/password or SSH key. In a multi-node deployment, the same user needs to be defined on all of the servers.

How do I backup and restore the system? Can I use mirror disk?

**Single Server:**
- Backup the machine
- Use a mirror disk with a regular Linux mechanism.
- Use a backup and restore script

**Multi-Node:**
As Sisense uses Kubernetes and GlusterFS, mirror cannot be used. GlusterFS saves 2 copies of each page (GlusterFS uses the term brick).

If any of the servers die, Sisense continues to be highly available.

In addition, Sisense is currently developing backup and restore scripts that will be part of the installation.

**Does the Linux deployment require specific CPU feature?**

Sisense recommends that your Linux deployment includes AVX2 which was introduced in Haswell.

**Which plugins are supported?**

All certified plugins are now supported on Linux and automatically installed as part of the default installation, including:

- Sisense BloX
- Insight Miner
- Embed Images
- Histogram
- JAQL

Every version upgrade will update all certified plugins to the version compatible with the release.

**Note:** Any manual change that you make to certified plugins will be overwritten. Sisense recommends not modifying certified plugins. If you want to change them, duplicate the plugin, rename it (including the internal label), and modify the duplicate version of the plugin. The upgrade process creates a backup copy of all plugins, which can be restored if needed.
Uploading Files through the File Manager

Sisense enables you to customize your Linux deployment with custom connectors, plugins, translations, and rebranded dashboards stored in your Linux shared storage. You can easily manage all of these files through the File Manager in the Sisense Admin page with your Sisense Administrator credentials.
Accessing the File Manager

**Note:** Only Sisense Administrators can access the File Manager.

**To access the File Manager:**
In the System Management tab of the *Admin* page, click *File Management*.

The File Manager is displayed in a new window.
Managing Files in the File Manager

In the File Manager, you can upload your static customized files for your Sisense deployment. The File Manager displays a list of relevant directories where the relevant customized files can be stored in your shared storage. You can upload the files by opening the relevant directory and dragging the files into your browser.

The list below describes which files should be placed in the directory with links to the relevant pages where you can find more information about the files that should be stored in each folder.

- **branding**: This directory contains customized files for rebranding Sisense. See Rebranding Sisense for more information.
- **connectors**: This directory contains Sisense connectors and any customized connectors you have developed. See Custom REST Connector for more information.
- **emails**: This directory contains customized automated Sisense emails. See Rebranding Sisense Automated Emails.
- **plugins**: This directory contains plugins and is the location where new plugins should be stored.
- **samples**: This directory contains sample ElastiCubes included in your deployment.
- **translations**: This directory contains translations of the Sisense interface.
Migration from Sisense Windows Deployment

Sisense provides a migration tool for migrating existing Windows deployments to Linux deployment. The Windows2Linux migration tool migrates the following Sisense assets from a Sisense deployment in Windows to a Sisense deployment in Linux:

- Data models (ElastiCube and Live) (.ecdata files)
  - **Note**: ElastiCubes in draft mode will not be migrated. All ElastiCubes you intend to migrate should be built before running the migration tool
- Dashboards
- User definitions: users, groups, data security rules
- System resources: plugins, translation files, email templates, rebranding assets

In a single node deployment, the Windows2Linux migration tool should be run from your production machine. If you have implemented a multi-node deployment, the tool should be run on one of your query nodes. This assumes that your query nodes are identical. If they are not identical, then you should run the tool on each query node you want to migrate. When you run the migration tool on the second query node, you should perform a partial migration of just the .ecdata files. In this scenario, you should consult with Sisense Technical Support for more information.

As the migration tool is run on one of your query nodes and not a build node, if you have any customized files, such as rebranded emails or translated language files, you must migrate these manually to the following directories in your Linux machine:

- **Rebranded Files**: /opt/sisense/storage/rebranding
- **Automated Emails**: /opt/sisense/storage/emails
**Customized Languages**: /opt/sisense/storage/languages

**Plugins**: /opt/sisense/storage/plugins/

**Note**: In multi-node deployments, these files are saved in shared storage that you can access with the command:

```
kubectl -n [NAMESPACE] cp [FILENAME] management-
<TAB>:/opt/sisense/storage/[shared asset name - plugins, languages, emails]
```

When you run the migration tool in Windows, Sisense duplicates all the assets to be migrated in a new directory automatically created on your Windows machine. Keep in mind that depending on the size of your .ecdata files (your data models and their data), this assets directory can be quite large and you must insure that you have enough space to support these files. Once the migration is complete, the assets directory is deleted, while your original files remain on the machine.

**Note**: SSO will be disabled automatically in Linux after migrating the Sisense assets.

**Prerequisites**
- Sisense V7.2 and higher. For Sisense V7.1 and earlier, contact Sisense
- Data is transferred through SSH through port 22.
- Downtime is required for each ElastiCube only while exporting the ElastiCube data
- To migrate to multi-node environments, contact Sisense
- Moving assets from Windows to Linux uses SCP, which uses Secure Shell (SSH) for data transfer. You should ensure that such communication possible from Windows to Linux machines

**Limitations**
- Sisense does not migrate static data source files (Excel, CSV) for ElastiCubes
- The following assets are not currently migrated:
  - Pre/Post build plugins
  - Scheduled builds
  - Custom REST connectors
  - Rebranding files
  - Email templates
  - Translation files

**Single Node Migrations**
The instructions below apply to single node deployments where all your Sisense assets reside on a single machine.
In Windows:

Download the Win2Linux file to your Windows-based Sisense query nodes. Contact Sisense for the latest version of this file.

Unzip the file. Make sure that the files are extracted to a folder with no spaces in the folder's name.

In the directory where you extracted the migration file, open the config.json file in a text editor and define the following:

**Note:** In most cases, the default value provided by Sisense is correct and does not need to be changed.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEMP_DIR_FOR_ASSETS</td>
<td>Enter the location on your Windows machine where your Sisense assets are stored before they are sent to an archive file and migrated to Linux.</td>
</tr>
<tr>
<td>ARCHIVE_OUTPUT_PATH</td>
<td>Enter the location on your Windows machine where your Sisense assets are stored before they are sent to an archive file and migrated to Linux.</td>
</tr>
<tr>
<td>EXPORT_ECDATA</td>
<td>Set the value to true to export your ecdata and JSON schema files or false to export only the JSON schemas.</td>
</tr>
</tbody>
</table>

**Section: STATIC_ASSETS**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLUGINS_PATH</td>
<td>Enter the location of your plugins directory on your Windows machine.</td>
</tr>
<tr>
<td>TRANSLATIONS_PATH</td>
<td>Enter the location of your translations directory on your Windows machine.</td>
</tr>
<tr>
<td>REBRANDING_PATH</td>
<td>Enter the location of your rebranding directory on your Windows machine.</td>
</tr>
<tr>
<td>EMAIL_TEMPLATES_PATH</td>
<td>Enter the location of your email templates directory on your Windows machine.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Section: SISENSE_API</strong></td>
<td></td>
</tr>
<tr>
<td>USERNAME</td>
<td>Enter the Sisense Administrator’s user name for Windows.</td>
</tr>
<tr>
<td>PASSWORD</td>
<td>Enter the Sisense Administrator’s password for Windows.</td>
</tr>
<tr>
<td>WIN_BASE_URL</td>
<td>Enter the BASE URL for Sisense including the port.</td>
</tr>
<tr>
<td><strong>Section: LINUX_SCP</strong></td>
<td></td>
</tr>
<tr>
<td>HOST</td>
<td>Enter your host or the IP address of your Linux machine.</td>
</tr>
<tr>
<td>USERNAME</td>
<td>Enter the SSH username of your Linux machine</td>
</tr>
<tr>
<td>PASSWORD</td>
<td>Enter the password of your Linux machine</td>
</tr>
<tr>
<td>AMAZON_RSA_PEM_KEY_PATH</td>
<td>If your Linux machines are hosted by Amazon, enter the directory where your PEM key is located.</td>
</tr>
<tr>
<td>DESTINATION_DIRECTORY</td>
<td>Enter the Linux directory where all assets and scripts will be transferred to.</td>
</tr>
<tr>
<td><strong>Section: LINUX_TOKEN_LOGIN</strong></td>
<td></td>
</tr>
<tr>
<td>PORT</td>
<td>Enter the port number for your Linux API Gateway.</td>
</tr>
<tr>
<td>URL_SUFFIX</td>
<td>The location of your Sisense API token. This value should not be changed.</td>
</tr>
<tr>
<td><strong>Section: WIN_MONGODB</strong></td>
<td></td>
</tr>
<tr>
<td>MONGODB_IP</td>
<td>Enter the address of the Sisense application database on your Windows machine.</td>
</tr>
<tr>
<td></td>
<td>If you have not changed this address, you can use the default value.</td>
</tr>
<tr>
<td>MONGO</td>
<td>Enter the port number for your Sisense application</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DB_PORT</td>
<td>database.</td>
</tr>
<tr>
<td></td>
<td>If you have not changed this, you can use the default value.</td>
</tr>
<tr>
<td>MONGODB_DATABASES_ES</td>
<td>Enter the names of any additional Sisense databases associated with your Sisense application database. Each additional database should be separated by a comma.</td>
</tr>
<tr>
<td><strong>Section: WIND_ZOOKEEPER</strong></td>
<td></td>
</tr>
<tr>
<td>ZOOKEEPER_IP</td>
<td>Enter the address of the Sisense configuration database on your Windows machine.</td>
</tr>
<tr>
<td></td>
<td>If you have not changed this address, you can use the default value.</td>
</tr>
<tr>
<td>ZOOKEEPER_PORT</td>
<td>Enter the port number for your Sisense configuration database.</td>
</tr>
<tr>
<td></td>
<td>If you have not changed this, you can use the default value.</td>
</tr>
<tr>
<td><strong>Section: LINUX_MONGODB</strong></td>
<td></td>
</tr>
<tr>
<td>NAMESPACE</td>
<td>Name of the Sisense namespace where the application database is to be migrated to.</td>
</tr>
<tr>
<td>POD_NAME</td>
<td>For single node deployments, enter &quot;mongod-0&quot; as the value. For multi-node deployments, enter &quot;sisense-mongodb-replicaset-0&quot;.</td>
</tr>
<tr>
<td><strong>Section: LINUX_ZOOKEEPER</strong></td>
<td></td>
</tr>
<tr>
<td>SERVICE_NAME</td>
<td>Enter the name of service responsible for the configuration database.</td>
</tr>
<tr>
<td>NAMESPACE</td>
<td>Name of the Sisense namespace where the configuration database is to be migrated to.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DISABLE_HTTPS</td>
<td>Enter true to disable secure connections within your Zookeeper replica set.</td>
</tr>
<tr>
<td>DISABLE_SSO</td>
<td>Enter true to disable SSO within your Zookeeper replica set.</td>
</tr>
<tr>
<td>LINUX_PLUGINS_PATH</td>
<td>Enter the location where plugins are to be stored on Linux.</td>
</tr>
<tr>
<td>STORAGE_PATH</td>
<td>Enter the location where ecdata files should be stored on your Linux machine or shared storage.</td>
</tr>
<tr>
<td>NAMESPACE</td>
<td>Enter the name of the Sisense namespace where ecdata files should be stored.</td>
</tr>
<tr>
<td>URL_SUFFIX (LINUX_IMPORT_ECDATA)</td>
<td>Enter the address of the API endpoint used to import ElastiCubes. This value does not need to be changed.</td>
</tr>
<tr>
<td>URL_SUFFIX (LINUX_IMPORT_JSON_SCHEMA)</td>
<td>Enter the address of the API endpoint used to import ElastiCube schemas. This value does not need to be changed.</td>
</tr>
</tbody>
</table>

Open the Command Prompt (CMD) as an Administrator from within the win2linux_migration folder and run the following command to validate your changes in the config.json file:
`win2linux_migration.exe -h`

In the Command Prompt, run the command below to migrate Sisense to Linux.
`win2linux_migration.exe all_steps`

Once the script is complete, a confirmation message is displayed in the Command Prompt. Continue to the next part, migrating in Linux.
In Linux:

**Note**: Before completing the migration process, ensure that you have a working Sisense installation running in Linux.

In Linux, open the Linux host shell and go to `/home/sisense`.

Open the `config.json` file.

Verify that the parameters under `LINUX_MONGODB` are correct. `POD_NAME` should be set to the name of the primary pod name in cases where you have created a replica set for the application database. If you have a replica set for your application database, make sure to enter the primary pod name.

The following command can return the primary pod name:

```
kubectl -n sisense -c mongod-container exec -it sisense-mongodb-replicaset-0 -- mongo prismWebDB --eval "rs.status()"
```

Verify that the parameters under `LINUX_ZOOKEEPER` are correct. `SERVICE_NAME` should be set to the name of the primary pod name in cases where you have created a replica set for the configuration database (ZooKeeper).

Execute the following command to validate your configuration:

```
sudo python win2linux_sisense.py -h
```

Execute the following command to complete the migration your configuration:

```
sudo python win2linux_sisense.py all_steps
```

While the script is running, a message may appear asking you to confirm if your ZooKeeper configuration should be overwritten. Enter "y" to confirm that it should be overwritten.

Once the script is complete, a confirmation message is displayed in the command line. The migration is complete and you should be able to access Sisense through your browser as before.

**Multi-Node Migrations**

The instructions below apply to multi-node deployments where your Sisense assets are replicated and reside on multiple machines.

**Prerequisites**

- Make sure that there are no ElastiCubes builds running during the migration in the query machine
Make sure that no ElastiCubes are transferred from the build to the query node during the migration process
In Windows:

Download the Win2Linux file to your Windows-based Sisense machine. Contact Sisense for the latest version of this file.

Unzip the file. Make sure that the files are extracted to a folder with no spaces in the folder’s name.

Create the directory `C:\assets_to_linux` with the same name and syntax.

Copy the static Sisense assets folders to the new folder you created in Step 3, `C:\assets_to_linux`. These directories are typically located in `C:\Program Files\Sisense\app`. For example, `C:\Program Files\Sisense\app\plugins` should be moved to `C:\assets_to_linux\plugins`.

In the directory where you extracted the migration file, open the `config.json` file in a text editor and define the following:

**Note:** In most cases, the default value provided by Sisense is correct and does not need to be changed.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEMP_DIR_FOR_Assets</td>
<td>Enter the location on your Windows machine where your Sisense assets are stored before they are sent to an archive file and migrated to Linux.</td>
</tr>
<tr>
<td>ARCHIVE_OUTPUT_PATH</td>
<td>Enter the location on your Windows machine where your Sisense assets are stored before they are sent to an archive file and migrated to Linux.</td>
</tr>
<tr>
<td>EXPORT_ECDATA</td>
<td>Set the value to <strong>true</strong> to export your ecdata and JSON schema files or false to export only the JSON schemas.</td>
</tr>
</tbody>
</table>

**Section: STATIC_ASSETS**

| COLLECT       | In multi-node migrations, enter **true** to collect the static |

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>assets described in Step 4 and migrate them.</td>
<td></td>
</tr>
<tr>
<td>PLUGINS_PATH</td>
<td>Enter the location of your plugins directory on your Windows machine.</td>
</tr>
<tr>
<td>TRANSLATIONS_PATH</td>
<td>Enter the location of your translations directory on your Windows machine.</td>
</tr>
<tr>
<td>REBRANDING_PATH</td>
<td>Enter the location of your rebranding directory on your Windows machine.</td>
</tr>
<tr>
<td>EMAIL_TEMPLATES_PATH</td>
<td>Enter the location of your email templates directory on your Windows machine.</td>
</tr>
</tbody>
</table>

**Section: SISENSE_API**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>USERNAME</td>
<td>Enter the Sisense Administrator's user name for Windows.</td>
</tr>
<tr>
<td>PASSWORD</td>
<td>Enter the Sisense Administrator's password for Windows.</td>
</tr>
<tr>
<td>WIN_BASE_URL</td>
<td>Enter the BASE URL for Sisense including the port.</td>
</tr>
</tbody>
</table>

**Section: LINUX_SCP**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOST</td>
<td>Enter your host or the IP address of your primary Linux machine.</td>
</tr>
<tr>
<td>USERNAME</td>
<td>Enter the SSH username of your primary Linux machine</td>
</tr>
<tr>
<td>PASSWORD</td>
<td>Enter the password of your primary Linux machine</td>
</tr>
<tr>
<td>AMAZON_RSA_PEM_KEY_PATH</td>
<td>If your Linux machines are hosted by Amazon, enter the directory where your PEM key is located.</td>
</tr>
<tr>
<td>DESTINATION_DIRECTORY</td>
<td>Enter the Linux directory where all assets and scripts will be transferred to.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Section: LINUX_TOKEN_LOGIN</strong></td>
<td></td>
</tr>
<tr>
<td>PORT</td>
<td>Enter the port number for your Linux API Gateway.</td>
</tr>
<tr>
<td>URL_SUFFIX</td>
<td>The location of your Sisense API token. This value should not be changed.</td>
</tr>
<tr>
<td><strong>Section: WIN_MONGODB</strong></td>
<td></td>
</tr>
<tr>
<td>MONGODB_IP</td>
<td>Enter the IP address of the Sisense application database on your primary Linux machine. If you have not changed this address, you can use the default value.</td>
</tr>
<tr>
<td>MONGODB_PORT</td>
<td>Enter the port number for your Sisense application database.</td>
</tr>
<tr>
<td></td>
<td>If you have not changed this, you can use the default value.</td>
</tr>
<tr>
<td>MONGODB_DATABASES</td>
<td>Enter the names of any additional Sisense databases associated with your Sisense application database. Each additional database should be separated by a comma.</td>
</tr>
<tr>
<td><strong>Section: WIN_ZOOKEEPER</strong></td>
<td></td>
</tr>
<tr>
<td>ZOOKEEPER_IP</td>
<td>Enter the address of the Sisense configuration database on your Windows machine.</td>
</tr>
<tr>
<td></td>
<td>If you have not changed this address, you can use the default value.</td>
</tr>
<tr>
<td>ZOOKEEPER_PORT</td>
<td>Enter the port number for your Sisense configuration database.</td>
</tr>
<tr>
<td></td>
<td>If you have not changed this, you can use the default value.</td>
</tr>
<tr>
<td><strong>Section: LINUX_MONGODB</strong></td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NAMESPACE</td>
<td>Name of the Sisense namespace where the application database is to be migrated to.</td>
</tr>
<tr>
<td>POD_NAME</td>
<td>For single node deployments, enter &quot;mongod-0&quot; as the value. For multi-node deployments, enter &quot;sisense-mongodb-replicaset-0&quot;.</td>
</tr>
</tbody>
</table>

**Section: LINUX_ZOOKEEPER**

<table>
<thead>
<tr>
<th>SERVICE_NAME</th>
<th>Enter the name of service responsible for the configuration database.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAMESPACE</td>
<td>Name of the Sisense namespace where the configuration database is to be migrated to.</td>
</tr>
<tr>
<td>DISABLE_HTTPS</td>
<td>Enter true to disable secure connections within your Zookeeper replica set.</td>
</tr>
<tr>
<td>DISABLE_SSO</td>
<td>Enter true to disable SSO within your Zookeeper replica set.</td>
</tr>
<tr>
<td>LINUX_PLUGINS_PATH</td>
<td>Enter the location where plugins are to be stored on Linux.</td>
</tr>
<tr>
<td>STORAGE_PATH</td>
<td>Enter the location where ecdata files should be stored on your Linux machine or shared storage.</td>
</tr>
<tr>
<td>NAMESPACE</td>
<td>Enter the name of the Sisense namespace where ecdata files should be stored.</td>
</tr>
<tr>
<td>URL_SUFFIX</td>
<td>Enter the address of the API endpoint used to import ElastiCubes. This value does not need to be changed.</td>
</tr>
<tr>
<td>URL_SUFFIX</td>
<td>Enter the address of the API endpoint used to import</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>LINUX_IMPORT_JSO N_SCHEMA</td>
<td>ElastiCube schemas. This value does not need to be changed.</td>
</tr>
</tbody>
</table>

Open the Command Prompt (CMD) as an Administrator from within the win2linux_migration folder and run the following command to validate your changes in the config.json file:
```
win2linux_migration.exe -h
```

In the Command Prompt, run the command below to migrate Sisense to Linux.
```
win2linux_migration.exe all_steps
```
Once the script is complete, a confirmation message is displayed in the Command Prompt. Continue to the next part, migrating in Linux.
**In Linux:**

**Note:** Before completing the migration process, ensure that you have a working Sisense installation running in Linux.

In Linux on your master machine, open the Linux host shell and go to `/home/sisense`.

Open the `config.json` file.

Verify that the parameters under **LINUX_MONGODB** are correct. **POD_NAME** should be set to the name of the primary pod name in cases where you have created a replica set for the application database. If you have a replica set for your application database, make sure to enter the primary pod name. The following command can return the primary pod name:

```
kubectl -n sisense -c mongod-container exec -it sisense-mongodb-replicaset-0 -- mongo prismWebDB --eval "rs.status()"
```

Verify that the parameters under **LINUX_ZOOKEEPER** are correct. **SERVICE_NAME** should be set to the name of the primary pod name in cases where you have created a replica set for the configuration database (ZooKeeper).

Execute the following command to validate your configuration:

```
sudo python win2linux_sisense.py -h
```

Execute the following command to complete the migration your configuration:

```
sudo python win2linux_sisense.py all_steps
```

While the script is running, a message may appear asking you to confirm if your ZooKeeper configuration should be overwritten. Enter "y" to confirm that it should be overwritten. Once the script is complete, a confirmation message is displayed in the command line. The migration is complete and you should be able to access Sisense through your browser as before.
Upgrading Sisense on Linux

The process for upgrading to the latest version of Sisense is similar to the process of provisioning Sisense. The key difference is that the value of upgrade in the config.yaml file must be changed to Yes.

If you are upgrading a multi-node environment, Sisense recommends you create a new namespace for the upgraded environment to ensure uptime. In this case, you must also define a new namespace value for the parameter namespace_name.

Backup existing version using the following guide - Linux k8s Sisense assets collector
Upgrade Sisense using the following guide - How to Install Sisense in Linux - Kubespray Cluster/Single Installation (tar.gz)
Restore Sisense Data from the previous version using the following guide - Linux k8s Sisense assets restore

To upgrade Sisense in a Linux environment:

Navigate to the directory where you extracted the tar.gz file.
   `cd sisense-[sisense-version]`
Edit the config.yaml file.
   `vim config.yaml`
After running this command, the following parameters are displayed.
In config.yaml file, define values for the following parameters:
Uninstalling Sisense from Linux

If you need to remove Sisense, there are several options available to you from the Sisense config.yaml file you configured when you originally set up Sisense on Linux. You can remove Sisense, remove the Kubernetes cluster, or delete all your Sisense data.

To remove Sisense, you need to edit the config.yaml file from the Linux terminal and modify any of the following values:

- **Uninstall_cluster**: Enter true to remove the Kubernetes infrastructure and Sisense services.
- **Uninstall_sisense**: Enter true to uninstall Sisense services, but leave your Kubernetes infrastructure in case it's needed in the future. This deletes Sisense, but leaves your Sisense user data and ElastiCube models.
- **Remove_user_data**: Enter true to delete all user data. This deletes your ElastiCube models, application database, message broker, and plug-ins.

To remove Sisense completely, enter true for each of the parameters described above in the config.yaml file. For more information, see [Step 4: Initializing Sisense on Linux](#).
Debugging Cloud-Native Sisense on Linux

This page provides a list of commands and log locations that you can use to debug your Sisense deployment on Linux.

**Get kubectl tab completions:**
source <(kubectl completion bash) 2>/dev/null

**Get helm tab completions:**
source <(helm completion bash) 2>/dev/null

**Get a list of pods:**
kubectl -n sisense get pods
OR
kubectl -n sisense get pods -o wide

**Access management logs:**
kubectl -n sisense logs $(kubectl -n sisense get pods -l app="management" -o custom-columns=":.metadata.name")

**Tail the log and print the last 10 lines:**
kubectl -n sisense -f --tail=10 logs $(kubectl -n sisense get pods -l app="management" -o custom-columns=":.metadata.name")

**Get Kubernetes events:**
kubectl -n sisense get events

**Monitor Kubernetes events:**
kubectl -n sisense get events --watch

**Restart and build the Sisense service:**
kubectl -n sisense delete pod $(kubectl -n sisense get pods -l app="build" -o custom-columns=":.metadata.name")

**Restart all services:**
kubectl -n sisense delete pods $(kubectl -n sisense get pods -o custom-columns=":.metadata.name")

**Shut-off Sisense single node deployment:**
kubectl scale -n sisense deployment --all --replicas=0

**Restore Sisense single node deployment:**
kubectl scale -n sisense deployment --all --replicas=1

**Note:** This is not recommended in multi-node deployments as some of the services have more than 1 replica. Sisense recommends you use the installer to update the load system.

**The location of log directorys:**
On the first installed node.
/var/log/sisense/namespace/
/var/log/sisense/sisense/combined.log -- logs of all services

For each service, there is a log file that you can retrieve, for example:
/var/log/sisense/sisense/query.log
/var/log/sisense/sisense/api-gateway.log

**Get Sisense CLI:**
kubectl -n sisense cp $(kubectl -n sisense get pods -l app="management" -o custom-columns=":.metadata.name"):/etc/sisense.sh .
source sisense.sh
login_sisense <server-ip>:30845 <sisense-admin>

**Get a list of ElasticCubes from the CLI:**
si elasticubes list
Build an ElastiCube from the Sisense CLI:

```bash
si elasticubes build -name ariel -type full
```

Get GlusterFS topology information:

```bash
kubectl exec -it $(kubectl get pods -l name="heketi" -o custom-columns=":.metadata.name") heketi-cli topology info
```

View disk usage for shared storage:

```bash
kubectl -n sisense exec -it $(kubectl -n sisense get pods -l app="management" -o custom-columns=":.metadata.name") -- bash -c "df -H /opt/sisense/storage"
```

See all device allocations in the GlusterFS:

```bash
for i in $(kubectl exec -it $(kubectl get pods -l name="heketi" -o custom-columns=":.metadata.name") heketi-cli node list | awk '{gsub(/Id\:\:/,""); print $1}'); do kubectl exec -it $(kubectl get pods -l name="heketi" -o custom-columns=":.metadata.name") heketi-cli node info $i; done
```

Expand Sisense disk volume by 1GB:

```bash
VOL=$(kubectl get persistentvolumes $(kubectl get persistentvolumeclaims -n sisense storage -o custom-columns=":.spec.volumeName") -o custom-columns=":.spec.glusterfs.path" | grep vol | awk '{gsub('/vol_/,""); print $1}')
kubectl exec -it $(kubectl get pods -l name="heketi" -o custom-columns=":.metadata.name") -- bash -c "heketi-cli volume expand --volume=$VOL --expand-size=1"
```

Get all services:

```bash
kubectl get services -n sisense
```
Execute bash on a pod:
kubectl -n sisense exec -it $(kubectl -n sisense get pods -l app="management" -o custom-columns=":.metadata.name") bash

List all blocked devices in the system:
lsblk

Get Kubernetes dashboard URL:
kubectl cluster-info

Get an Admin user token:
kubectl describe secret -n kube-system admin-user-token

List helm releases:
helm list --all

Expose the Message Broker Management UI:
kubectl port-forward -n sisense pod/sisense-rabbitmq-ha-0 30999:15672 --address=0.0.0.0

Expose the Sisense application database internal communication port:
kubectl port-forward --address 0.0.0.0 -n sisense sisense-mongodb-replicaset-0 30846:27017

Kill all evicted pods:
kubectl get po --all-namespaces | grep Evicted | awk '{print $2, "--namespace", $1}' | xargs kubectl delete pod

Get the Grafana of the cluster:
kubectl -n monitoring get svc prom-operator-grafana
Sisense CLI Commands

In addition to the REST API, Sisense includes a CLI (Command Line Interface) commands that support executing various Sisense-related commands using the Linux command line.
Accessing the Sisense CLI

To access the Sisense CLI:

- Connect to your Linux host using ssh.
- Run the command ".add_completion-ns-sisense.sh".

After running the command, you can then run Sisense commands in the format si <group name> <command> <parameters>.

For example, if your command is part of the DataGroups group, your command would like this: si datagroups list

The table below lists the Sisense commands, descriptions, related parameters, and examples.
## DataGroups

<table>
<thead>
<tr>
<th>Commands</th>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
</table>
| list     | Displays a list of all data groups.  
**Example:**  
si datagroups list |            |
| create   | Create a new data group.  
**Example:**  
si datagroups create -name example | **Mandatory:** name  
**Optional:** connector-mode  
index-size  
default  
query-instances  
query-cpu-limit  
query-cpu-request  
query-memory-limit  
query-memory-request  
query-cores-usage-percentage  
built-cpu-limit  
built-cpu-request  
built-memory-limit  
built-memory-request |
<table>
<thead>
<tr>
<th>Commands</th>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>build-cores-usage-percentage enable-recycler</td>
</tr>
<tr>
<td>rename</td>
<td>Rename a data group.</td>
<td><strong>Mandatory:</strong> name new-name</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> si datagroups create -name example -new-name example</td>
<td></td>
</tr>
<tr>
<td>update</td>
<td>Update a data group.</td>
<td><strong>Mandatory:</strong> name</td>
</tr>
<tr>
<td></td>
<td><strong>Optional:</strong> connector-mode index-size default query-instances query-cpu-limit query-cpu-request query-memory-limit query-memory-request query-cores-usage-percentage build-cpu-limit build-cpu-request build-memory-limit build-memory-</td>
<td></td>
</tr>
<tr>
<td>Commands</td>
<td>Description</td>
<td>Parameters</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>------------</td>
</tr>
<tr>
<td>assign</td>
<td>Assigns an ElastiCube to a data group.</td>
<td>request&lt;br&gt;build-cores-usage-percentage&lt;br&gt;enable-recycler&lt;br&gt;&lt;br&gt;Mandatory: name cube&lt;br&gt;Example: si datagroups assign -name example -cube examplecube</td>
</tr>
<tr>
<td>attach</td>
<td>Attaches a node to a data group.</td>
<td>Mandatory: name node&lt;br&gt;roles&lt;br&gt;Example: si datagroups attach -name example -node myhost -roles BUILD,QUERY</td>
</tr>
<tr>
<td>detach</td>
<td>Detaches a node to a data group.</td>
<td>Mandatory: name node&lt;br&gt;Example: si datagroups detach -name example -node myhost</td>
</tr>
<tr>
<td>delete</td>
<td>Deletes a data group.</td>
<td>Mandatory: name&lt;br&gt;Example: si datagroups detach -name example</td>
</tr>
</tbody>
</table>
## ElastiCube

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Parameter</th>
<th>Parameter</th>
</tr>
</thead>
</table>
| list    | Displays a list of all ElastiCubes.  
**Example:**  
si elasticube list | | |
| start   | Starts an ElastiCube.  
**Example:**  
si elasticube start -name example | **Mandatory:** name | |
| stop    | Stops an ElastiCube.  
**Example:**  
si elasticube stop -name example | **Mandatory:** name | |
| delete  | Deletes an ElastiCube.  
**Example:**  
si elasticubes delete -name example | **Mandatory:** name | |
| detach  | Detaches an ElastiCube.  
**Example:**  
si elasticube detach [-name example] [-delete false] | **Mandatory:** name  
**Optional:** delete | |
| attach  | Attaches an ElastiCube.  
**Example:**  
si elasticubes attach -path /path/to/cube/files | **Optional:** path  
title | |
| build   | Builds an ElastiCube.  
**Example:**  
si elasticube build -name example -type full -sync | **Mandatory:** type  
**Optional:** | |
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Parameter(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>[e cm_model.json]</td>
<td>sync name</td>
</tr>
<tr>
<td>import</td>
<td>Import an ElastiCube.</td>
<td>Optional:</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> si elasticube import -path &quot;/opt/sisense/storage/path/to/NameOfCube.ecdata&quot; -start false</td>
<td>path start</td>
</tr>
<tr>
<td>reset</td>
<td>Deletes all ElastiCubes from a database and storage</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> si elasticube reset</td>
<td></td>
</tr>
<tr>
<td>export</td>
<td>Exports an ElastiCube.</td>
<td>Optional:</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> si elelasticube export -name MyCube -path &quot;/opt/path/to/NameOfCube&quot; -file cubeName.ecdata2</td>
<td>name path file</td>
</tr>
<tr>
<td>console</td>
<td>Starts mclient console for a specific ElastiCube.</td>
<td>Optional:</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> si elasticube console -name xxx -language sql or</td>
<td>language name</td>
</tr>
<tr>
<td></td>
<td>elasticube console -name xxx -language mal</td>
<td></td>
</tr>
</tbody>
</table>
## Deployment

<table>
<thead>
<tr>
<th>Commands</th>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Displays a list of all nodes.</td>
<td>Optional: id</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>si deployment list</td>
<td></td>
<td></td>
</tr>
<tr>
<td>add</td>
<td>Adds a node to the system.</td>
<td>Mandatory: id, hostname, roles</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>si deployment add -id myhost -hostname myhost -roles BUILD,QUERY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td>Removes a node from the system.</td>
<td>Mandatory: id</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>si deployment remove -id myhost</td>
<td></td>
<td></td>
</tr>
<tr>
<td>instances</td>
<td>Lists all instances in the deployment.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>si deployment instances</td>
<td></td>
<td></td>
</tr>
<tr>
<td>version</td>
<td>List all the deployment nodes' versions.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>si deployment version</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Datasources

<table>
<thead>
<tr>
<th>Commands</th>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
</table>
| list     | Displays a list of all data sources.  
**Example:**
si datasources list | Optional: id |
| query    | Query JAQL on data source.  
**Example:**
cat myJaql.txt | si datasources query -name myEcube | Mandatory: name  
Optional: format |
| sql      | Executes an SQL query on data source.  
**Example:**
echo 'select * from MyTable' | si datasources sql -name myEcube | Mandatory: name |
## System

<table>
<thead>
<tr>
<th>Commands</th>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Displays versions of your services.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>si system</td>
<td></td>
</tr>
</tbody>
</table>
## Configuration

<table>
<thead>
<tr>
<th>Commands</th>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Gets the configuration value.</td>
<td><strong>Mandatory:</strong> key <strong>Optional:</strong> type</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>si configuration get -key category.entry -type key</td>
<td></td>
</tr>
<tr>
<td>set</td>
<td>Sets a configuration value.</td>
<td><strong>Mandatory:</strong> key <strong>Optional:</strong> new-value</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>si configuration set -key category.entry -new-value 1412</td>
<td></td>
</tr>
<tr>
<td>reset</td>
<td>Resets a configuration to the default value if possible.</td>
<td><strong>Optional:</strong> key</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>si configuration reset -key category.entry, or to reset all configuration of a specific category: configuration reset -key category, to change all configurations: configuration reset -key ALL</td>
<td></td>
</tr>
</tbody>
</table>
Breaking Changes

Deleting or renaming a column in a model that has a row-based data security rule defined on it prevents all dashboards from loading. The row-based data security rule must be corrected or deleted for your dashboards to load. The ElastiCube Set concept is not supported on Linux, and therefore the API /elasticubes/sets is no longer supported. The “Data Group” entity can be used for ElastiCube resource management. ElastiCubes imported from Windows must be fully built on Linux after importing them. Afterwards, you can perform accumulative builds.
Deprecated Features

The PSM.exe functionality has been replaced by the REST API
The desktop ElastiCube Manager has been replaced by the Data page available in the Sisense web application
7.4.3

**7.4.3.215** - Official release of Sisense V7.4.3 - August 4, 2019
New Features and Enhancements

More Supported Deployment Options

Version 7.4.3 is certified for the following deployment options:

<table>
<thead>
<tr>
<th>Single/Multi Location(Cloud Provider)</th>
<th>Kubernetes</th>
<th>OS(*)</th>
<th>Storage</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Node</td>
<td>ALL Sisense kubescape</td>
<td>Ubuntu, RHEL, CentOS, Amazon Linux 2</td>
<td>Local second disk</td>
<td></td>
</tr>
<tr>
<td>Multi-Node On Prem</td>
<td>Sisense kubescape</td>
<td>Ubuntu, RHEL, CentOS</td>
<td>GlusterFS/NFS</td>
<td></td>
</tr>
<tr>
<td>Multi-Node AWS</td>
<td>Sisense bundled kubescape</td>
<td>Ubuntu, RHEL, CentOS</td>
<td>GlusterFS/NFS/EFS/FSx</td>
<td>FSx uses EBS for</td>
</tr>
<tr>
<td>Single/Multi Location (Cloud Provider)</td>
<td>Kubernetes</td>
<td>OS(*)</td>
<td>Storage</td>
<td>Notes</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>------------</td>
<td>-------</td>
<td>------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Amazon Elastic Kubernetes Service (EKS)</td>
<td></td>
<td></td>
<td>EFS/FSx</td>
<td>Mono ZK</td>
</tr>
<tr>
<td>Azure Sisense kubespray</td>
<td>Ubuntu,</td>
<td></td>
<td>Gluster/Azure file share</td>
<td></td>
</tr>
<tr>
<td>Azure Kubernetes Service</td>
<td>RHEL,</td>
<td></td>
<td>Azure Files</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CentOS,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Amazon</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Linux 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single/Multi Location (Cloud Provider)</td>
<td>Kubernetes</td>
<td>OS(*)</td>
<td>Storage</td>
<td>Note</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>------------</td>
<td>-------</td>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>(AKS)</td>
<td>Centos</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Google Cloud Platform (GCP)</td>
<td>Sisense kubespray</td>
<td>Ubuntu, RHEL, Centos</td>
<td>GlusterFS/NFS/Google Cloud Filestore</td>
<td></td>
</tr>
<tr>
<td>Google Kubernetes Engine (GKE)</td>
<td>Google Cloud Filestore</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(*) The following OS versions are certified:
- Ubuntu 18.04 LTS
- RHEL 7.x
- Centos 7
- Amazon Linux 2.0

**Expanded Support for Plugins**

All certified plugins are now supported on Linux and automatically installed as part of the default installation, including:
- Sisense BloX
- Insight Miner
- Embed Images
- Histogram
- JAQLine
Every version upgrade will update all certified plugins to the version compatible with the release.

**Note:** Any manual change that you make to certified plugins will be overwritten. Sisense recommends not modifying certified plugins. If you want to change them, duplicate the plugin, rename it (including the internal label), and modify the duplicate version of the plugin. The upgrade process creates a backup copy of all plugins, which can be restored if needed

**Pivot Enhancements:**
The Pivot widget now supports export to PDF. The first 1000 rows of the pivot are exported. To export the full pivot result set, use the Export to Excel feature

**Connectors**
The following additional connectors are now supported on Linux:
  - Google Ads
  - MemSQL

For the full list of supported data sources, [click here](#).

**Other**
  - Usage Analytics is now supported on Linux
  - You can now import and export ElastiCubes and Live data models only to a local machine or to the Sisense server
Selected Bug Fixes

Some security issues were resolved
Emails from systems configured to use SSL will be generated correctly
The password recovery feature now works correctly when SSL is configured
Unsupported functions for Live data sources were removed including, running sum, running average, and rank functions

Plugins:
- When the Dynamic ElastiCube plugin was enabled, other plugins didn't operate correctly
- When the Accordion and Aggregated plugins were enabled, dashboards didn't return results
- The JAQLine plugin displayed a blank window

Accumulative builds on Snowflake didn't always work correctly
The “Add Build Alert” action added alerts to ElastiCubes that were not selected

Data Security:
- Defining data security over REST API is now supported
- Data security attributes are now applied without delay

The Pulse page is now translated to Italian as expected
Limitations

Widgets
Linux supports the latest version of the Pivot widget, replacing the previous Pivot widget available on Windows. All existing widgets using the Windows Pivot widget are migrated automatically to the new Pivot widget. The following features are not yet supported on Linux Pivot widgets:
- Plugins on pivots (front-end plugins)
- Sisense Narratives
- Changing the “First day of week” doesn’t affect all the weeks in a pivot table
- The Scatter Map widget isn’t fully functional Linux
- Pie and Scatter charts do not highlight filtered slices

ElastiCubes
ElastiCubes imported from Windows must be fully built on Linux after importing them. Afterwards, you can perform accumulative builds

Connectors
- Excel with static ranges cannot be built
- In PostgreSQL, CSV, and MySQL, you need to import time fields as text, and then convert them to a time field
- When importing an ElastiCube that was exported from Windows, the file paths are not automatically converted to Linux paths ("/opt/storage/sisense/..."). You need to change the connectivity settings for each table to a valid CSV file
- Sisense Live is supported on MySQL, excluding MySQL 8.0
- Accumulative build by date using SalesForce connector fails
- Google Ads and Google Analytics connectors’ table customizations are not supported

Admin Page
- Data Designers cannot see their own Live models in the Admin page.
- Session inactivity timeout always uses the system global parameter and doesn’t enforce separate Admin inactivity configuration values

Data Security
- Following importing an ElastiCube, you must open the ElastiCube on the data page at least once, before you can add row-based data security rules

Plugins
The Jump to Dashboard plugin functions on Linux, but once you associate the plugin with a dashboard, you cannot edit it. You need to delete it and redefine it.

Other

Sisense Narratives is not supported on this version
Pre/post build plugins (This is planned for a future release)
After upgrading from Sisense Linux beta, you need to restart the management service using the CLI command:

```bash
kubectl -n sisense delete pods $(kubectl -n sisense get pods | grep management | awk '{print $1}')
```
7.4.1

7.4.1.571 - Selected bug fixes - July 11, 2019
7.4.1.564 - Selected bug fixes and security hardening - June 25, 2019
7.4.1.561 - Selected bug fixes - June 19, 2019
7.4.1.555 - Official release of Sisense V7.4.1 - June 11, 2019
Announcing Breaking Changes

The ElastiCube Set concept is not supported on Linux, and therefore the API /elasticubes/sets is not supported. The “Data Group” entity can be used for ElastiCube resource management.
New Features and Enhancements

Dashboard responsiveness was significantly improved when navigating between dashboards.

Google BigQuery connector supports worldwide query location for improved performance.

New informative tooltips for ElastiCube and Live tiles in the Data page.

The “return to” parameter for SSO via JWT is now configurable in the Configuration Manager (Documentation).

Scheduled reports are sent even if some of the widgets are not loaded in the dashboard (Documentation).

Find functions quickly in the Formula Editor by name or syntax (Documentation).

Non-Admin Redshift users can now connect to specific databases in Redshift (Documentation).

Performance improvements for Live models on top and bottom ranking filters. This is relevant for all databases and MySQL from v5.8 and above.
Supported Connectors

The following data source is now supported in this Linux release:
  Google Analytics

For the full list of supported data sources, click here.
Deprecated Features

The PSM.exe functionality has been replaced by the REST API
The desktop ElastiCube Manager has been replaced by the Data page available in the Sisense web application
Limitations

**ElastiCubes**
ElastiCubes imported from Windows must be fully built on Linux after importing them. Afterward, you can perform accumulative builds.
You cannot import ElastiCubes over REST API

**Widgets**
Linux supports the new Pivot widget, replacing the previous Pivot widget available on Windows. All existing widgets using the Windows Pivot widget are migrated automatically to the new Pivot widget.
The following features are not yet supported on Linux Pivot widgets:
- Plugins on pivots (front-end plugins)
- Sisense Narratives
- Export to PDF (Export to excel and CSV is supported)
The Scatter Map widget isn’t supported on Linux

**Connectors**
Excels with static range cannot be built
In Postgres, CSV, and MySQL, you need to import time fields as text, and then convert to a time field
When importing an ElastiCube that was exported from Windows, the file paths are not automatically converted to linux paths ("/opt/storage/sisense/"). You need to change the connectivity settings for each table to a valid CSV file
Sisense Live is supported on MySQL, excluding MySQL 8.0

**Data Security**
Defining data security over REST API is not supported
Data security attributes are applied only after a 5 minute delay
Data security values are not populated correctly for a live connection with direct query toggled off

**Plugins**
All of the certified plugins are supported on Linux, except for the following plugins which are not yet supported on the Linux release:
- Sisense BloX
- Insight Miner
- Embed Images
- Histogram
- JAQLine
The Jump to Dashboard plugin functions on Linux, but once you associate the plugin with a dashboard, you cannot edit it. You need to delete it and redefine it.

Other
Usage Analytics is not currently supported for Linux (planned to be introduced in a future release).

After upgrading from Sisense Linux beta, you need to restart the management service using the CLI command: `kubectl -n sisense delete pods $(kubectl -n sisense get pods | grep management | awk '{print $1}')`
7.3.1

7.3.1.632 - Initial release of Sisense Linux V7.3 - March 31, 2019
New Features and Enhancements

Monitor your Sisense Linux deployment with Prometheus, Grafana, and FluentD bundled with Sisense
Supported Connectors

The following data sources are supported by the Linux release: (Documentation)

MySQL
SQL Server
PostgreSQL
Redshift
Oracle
Google BigQuery
Snowflake
Generic JDBC (ElastiCube models only, no live models)
CSV (ElastiCube models only, no live models)
Excel (ElastiCube models only, no live models)
SalesForce (ElastiCube models only, no live models)
Google Sheets (ElastiCube models only, no live models)
Limitations

**ElastiCubes**
ElastiCubes imported from Windows must be fully built on Linux after importing them. Afterward, you can perform accumulative builds.
You cannot import ElastiCubes over REST API.

**Widgets**
Linux supports the new Pivot widget, replacing the previous Pivot widget available on Windows. All existing widgets using the Windows Pivot widget are migrated automatically to the new Pivot widget.
The following features are not yet supported on Linux Pivot widgets:
- Plugins on pivots (front-end plugins)
- Sisense Narratives
- Export to PDF (Export to excel and CSV is supported)
The Scatter Map widget isn’t supported on Linux.

**Connectors**
Excel with static range cannot be built
In Postgres, CSV, and MySQL, you need to import time fields as text, and then
convert to a time field
When importing an ElastiCube that was exported from Windows, the file
paths are not automatically converted to Linux paths
(“/opt/storage/sisense/…”). You need to change the connectivity settings
for each table to a valid CSV file
Sisense Live is supported on MySQL, excluding MySQL 8.0.

**Data Security**
Defining data security over REST API is not supported
Data security attributes are applied only after a 5 minute delay
Data security values are not populated correctly for a live connection with
direct query toggled off.

**Plugins**
All of the certified plugins are supported on Linux, except for the following plugins which are not yet supported on the Linux release:
- Sisense BloX
- Insight Miner
- Embed Images
- Histogram
- JAQLine
The Jump to Dashboard plugin functions on Linux, but once you associate the plugin with a dashboard, you cannot edit it. You need to delete it and redefine it.

**Other**

Usage Analytics is not currently supported for Linux (planned to be introduced in a future release)

After upgrading from Sisense Linux beta, you need to restart the management service using the CLI command: `kubectl -n sisense delete pods $(kubectl -n sisense get pods | grep management | awk '{print $1}')`

Pre/post build plugins (planned to be introduced in a future release)
The Sisense Cloud Managed Service

Sisense’s Cloud Managed Service is a cloud-based service that provides a secure, high-performance BI environment. Sisense’s Cloud Managed Service enables organizations to focus on their businesses and BI results, without having to manage the technical aspects of deploying a BI project. The Sisense Cloud is based on the same scalable, performant, and agile solution that is available to Sisense’s entire client community through our on-premises offering.
Cloud Security

Sisense Cloud customers take advantage of the robust security and available certifications provided by Sisense and industry’s cloud computing leaders, such as Amazon Web Services. A description of the Sisense security measures can be found here.
Cloud Managed Service SLA (Service Level Agreement)

The Sisense Managed Service is a robust and reliable solution. It is hosted on AWS, one of the most reliable computing platform with the highest SLA standard, as set out in the AWS SLA. Based on the AWS SLA, Sisense defines the BI Service SLA, as set out in the Sisense Cloud Terms and Conditions.
Cloud Managed Service Administration

Sisense deploys your system in a dedicated virtual private network on AWS. The Sisense Cloud Operations Team sets up the cloud environment and works with your Administrative users on establishing access to the cloud environment and securely connecting to your data sources.

In addition, there are a few other steps you must take:

- **Complete the Cloud Setup Form**: This form should be completed by you and provides Sisense with the necessary information for configuring your Cloud environment on Sisense's side.
- **Setting Up the Cloud Environment**: On your side, there are few things you must configure including access to Sisense, and verifying that the relevant ports are open.
- (Optional) **Complete the Cloud Migration Form**: If you need to migrate your existing production/trial/POC system, complete this form. Once complete, return the form to your Customer Success Manager. The Cloud Operations team reviews the form, requests clarifications as needed, and assists you with migrating Sisense.
Setting up the Cloud Environment

While Sisense configures your cloud environment, there are a few steps you must complete to manage Sisense. The Sisense Cloud Operations team will assist you to complete the following tasks described below.

- [Access to the Cloud Environment](#)
- [Data Transfer](#)
- [Configuring your Network](#)
- [Optional Configuration](#)
Access to the Cloud Environment

Based on the details provided in the Cloud Setup Form, the Sisense Cloud Operations Team deploys the cloud environment. How your users access the environment is described in the diagram below.

All users can access dashboards and Sisense via secure HTTPS from any location. The default Sisense site is https://<customer_name>.sisense.com. Should a custom domain be required, Sisense supports custom URLs using SSL certificates provided by you, as described here.

If necessary, you can access the Desktop ElastiCube Manager as described in Accessing the Desktop ElastiCube Manager.
Data Transfer

In some cases, you may want to transfer data to Sisense, for example, when uploading Excel files or customized add-ons.

Sisense supports the following data source connections to connect to the Sisense application database:

- Direct connection to a database with or without SSL
  - If selected, you will provide Sisense with your public/external IP and port of the database. If SSL encryption is required, you will provide an SSL certificate and send it to Sisense
- Site-2-site VPN (IPSec) tunnel
  - If you choose this option, Site-2-site VPN, Sisense will create a dual-tunnel AWS VPN. You must configure a firewall to connect to the database via internal IP routing based on the AWS configuration instructions provided by Sisense.
- SSH Tunnel
  - The SSH server should be set up on the database server or via a bastion gateway.
- VPC Peering
  - For databases deployed on an AWS VPC, you need to provide the VPC details. Once the cloud environment is deployed, Sisense will issue a peering request or, alternatively, you can issue a peering request from your VPC.
- Access to AWS RDS, Athena, etc. with or without SSL
- Secure FTPS as described here: [Transferring Files to Sisense Cloud](#).
- Sisense Customer Data Proxy
  - If none of the above methods is applicable, the Sisense Customer Data Proxy (CDP) can be deployed on a virtual machine behind a firewall to locally connect to your data source(s). The CDP is an OVA VM image to be deployed within your network where it should have at least one interface on a subnet with the source DB, and at least one interface routable to the internet, and open for outgoing connections on port TCP 22. The CDP maintains the outbound connection to the Sisense cloud to transfer the data via an encrypted SSH tunnel. The CDP supports multiple data sources within your network. For a detailed description of the CDP, click [here](#).
Configuring your Network

Sisense’s Cloud Managed Service is deployed on secure AWS server(s) specifically dedicated to you. Access to the server is restricted to designated applications only. No other programs/processes can be run on the Sisense Server.

To connect the Sisense Managed Cloud Service to your database, the following ports should be open for outbound traffic:

- 990
- 8443-8444
- 4000-4100

The relevant IP addresses are provided to you by Sisense from the Sisense Cloud Operations Team.

After setting up your cloud environment, Sisense will contact you to schedule an online Cloud Access Verification Session.
Optional Configuration

**Custom DNS**: Review to implement a rebranded URL for Sisense.

**SSO**: Review to implement SSO in your cloud environment.

**Drivers**: Review to connect to various data sources with the Sisense JDBC connector.
Accessing the Desktop ElastiCube Manager

While the Sisense web application has most of the features of the desktop version of the ElastiCube Manager, some features have not been migrated yet. If you need to connect to the desktop ElastiCube Manager for any reason, you can connect through Microsoft's Remote Desktop Web Access. Once connected, you can access the Desktop ElastiCube Manager for your environment.

**To connect to the desktop ElastiCube Manager with Microsoft Remote Desktop Web Access:**

Type the URL of the RDWeb Access Web site into your browser. It can only be entered using the verified IP's given to Sisense. By default, you can access the RDWeb Access website at the following location.

https://server_name:8443/rdweb

Where `server_name` is the fully qualified domain name of the web server.
Download the ElastiCube Manager RDP file by clicking it.

Once downloaded, you may continue to use the file to access the ElastiCube Manager. You are not required to download a new one each time except for rare cases where there is a Windows update.

Open the file and re-enter your credentials to allow the certificate. Mac users must download the Windows Remote Desktop application from the app store to open the RDP file.

After you have accessed the ElastiCube Manager, you can open and modify your settings as needed.
Troubleshooting

**You cannot access the RDWeb page**
- Make sure your network is open to outbound traffic to your dedicated Sisense machine on port 8443
- Make sure that your IP is whitelisted by Sisense Support. You can find your IP by searching "what is my IP" in Google and sending the information to Sisense Support. If your IP is not a static IP, mention this in the support ticket.

**When trying to open the RDP file you get an error**
- Make sure that you are able to enter the Rdweb site to eliminate possible network issues
- Open a ticket with the screenshot of the error

**Adding an additional application**
- If you'd like to add any additional application such as ODBC or the orchestrator wizard, please open a ticket.

If access from a static IP address is not possible, for example, you do not have a permanent work location, dynamic access can be provided by Sisense as described [here](#).
Accessing your Sisense Server from a Dynamic Location

When connecting to your Sisense Server through RDWeb, FTP, or to restart services, only whitelisted IPs with Sisense can be used. If you do not have a static IP address or are working from multiple machines or IP addresses, the best option to connect to your Sisense Server is to request your IT team to route the request to your Cloud Sisense Server so it uses the office IP. If this is not possible, by request, Sisense can provide a proxy machine for you and provide you with a VPN.

The flow for connecting the Sisense backend through a Sisense proxy is as follows:
1. Connect to the VPN. See the steps below for instructions on how to connect to the VPN from a Mac or in Windows.
2. Use the remote desktop connection to connect to the proxy server.
3. Log in to the proxy server where you can use the FTP and RDWeb application.
Connect to the VPN

For Mac:

Open the Terminal as an Administrator and run this command:

```
sudo chmod 776 /Users
```

Uninstall any old Java versions from your computer, and download and install the [latest Java SE Development Kit (JDK)](https://www.oracle.com/). Ensure that Safari is not blocking popups by clicking Safari > Preferences then navigate to the Websites tab. Select Pop-up Windows and then set the site to Allow.

Open Safari and go to [https://nydcvpn.sisense.com](https://nydcvpn.sisense.com). Log in using your VPN credentials. Click Connect under Native Application.
Click the link in the new pop-up window to download the Check Point Mobile agent.

Install the agent by double-clicking it.

Restart your Mac.
Click **Trust server** when you're prompted with this screen.
Enter your computer password, and install the SNX_Install_Tool.

After this is installed you should see Connected under the Status line.

Windows VPN
Download the Check Point Capsule VPN from the Microsoft Store.
Extract and install the Check Point Capsule VPN.
Create the VPN connection profile by going into your Network Settings and adding a new VPN connection:
- **VPN Provider**: CheckPoint Capsule VPN
- **Server name**: nydcvpn.sisense.com
- **User Name and Password**: Enter your user name and password.
Connect to the VPN you created.

Add a VPN connection

VPN provider
- Check Point Capsule VPN

Connection name
- Sisense

Server name or address
- nydcvpn.sisense.com

Type of sign-in info
- User name and password

User name (optional)

Password (optional)
sisense

Connect  Advanced options  Remove
Transferring Files to Sisense Cloud

After you have opened the relevant ports, you can transfer files to the Sisense Server using an FTP site.

The types of files that you need to transfer are static data files, custom connectors, and plugins.

Sisense shares the following folders on the FTP site:
- **C:\Customer Data**: This folder is used mainly for uploading Excel and CSV files that you want to import into the ElastiCube.
- **C:\Program Files\Sisense\DataConnectors\DotNetContainer\Connectors**: This folder is used to add and manage REST Connectors.
- **C:\Program Files\Sisense\DataConnectors\JVMContainer**: This folder contains XML files responsible for connecting to web-based data sources, such as Facebook.
- **C:\program files\Sisense\app\plugins**: This folder contains your web plugins and add-ons.
- **C:\program files\Sisense\app\Resources**: This folder contains rebranding files and additional images to be displayed in your Sisense application.
- **C:\Program Files\Sisense\Prism\Server\Plugins**: This folder contains Pre and Post plugins for the ElastiCube Manager.
- **C:\program files\Sisense\app\galaxy-service\src\features\emails\templates**: This folder contains automated emails sent to your users from Sisense, for example, when you create an account for a user. See [Rebranding Sisense Automated Emails](#) for more information. If you are making a change in this folder, notify your Cloud administrator so changes can be tracked to prevent any problems when upgrading Sisense.

The following is an example of how to connect with [FileZilla](#):
Enter the following information and click Quickconnect:

**Host:** Your domain name for the Sisense Machine. For example mysite.sisense.com

**Username:** Your cloud user name, for example cloud\user.

**Password:** Your cloud password.

**Port:** The port, typically, 990.

After you log in, the following folders are displayed:

<table>
<thead>
<tr>
<th>Filename</th>
<th>Filesize</th>
<th>Filetype</th>
<th>Last modified</th>
<th>Perm</th>
</tr>
</thead>
<tbody>
<tr>
<td>..</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customer Data</td>
<td></td>
<td>File folder</td>
<td>1/21/2018 7:44:...</td>
<td></td>
</tr>
<tr>
<td>DotNetContainer_Connectors</td>
<td></td>
<td>File folder</td>
<td>1/21/2018 7:44:...</td>
<td></td>
</tr>
<tr>
<td>JVMContainer_Connectors</td>
<td></td>
<td>File folder</td>
<td>1/21/2018 7:44:...</td>
<td></td>
</tr>
<tr>
<td>PrismWeb_plugins</td>
<td></td>
<td>File folder</td>
<td>1/21/2018 7:44:...</td>
<td></td>
</tr>
<tr>
<td>PrismWeb_Resources</td>
<td></td>
<td>File folder</td>
<td>1/21/2018 7:44:...</td>
<td></td>
</tr>
<tr>
<td>Server_plugins</td>
<td></td>
<td>File folder</td>
<td>1/21/2018 7:44:...</td>
<td></td>
</tr>
<tr>
<td>templates</td>
<td></td>
<td>File folder</td>
<td>1/21/2018 7:44:...</td>
<td></td>
</tr>
</tbody>
</table>

Drag and drop the file from your File Explorer or Finder. For plugins, you need to download them locally, unzip them, and move the unzipped folder into PrismWeb_plugins FTP directory.
Uploading CSV and Excel Files

When uploading CSV or Excel files to the Customer Data FTP directory, how you access this folder from the desktop ElastiCube Manager differs from the Sisense Web Application. The instructions for both are provided below.

**Desktop ElastiCube Manager**

Click **Add Data** in the top menu of Sisense. Under the **Files** category select **Microsoft Excel File**.

In the window that opens, locate the Excel file.

**Sisense Web Application**

In the **Data** page, open an ElastiCube or create a new ElastiCube.
In the ElastiCube, click + Data. The Add Data dialog box is displayed.

Click **Microsoft Excel**. The Microsoft Excel Connect area is displayed.

Select **Server Access** to define the location of your files on the Sisense Server. If your Excel files frequently change, or you have scheduled builds, you should place your files on your server as the data is taken from the Excel files each time the ElastiCube is built.

**Note:** To use this option, you must have remote access to the server.

Select **Input Folder Path** and enter the full file path with the file name where your Excel files are located. This will display each Excel file in the
folder in the next screen where you select what tables to add to the ElastiCube.

OR

Select **Input File Path** and enter the full file path with the file name and its extension of your Excel file. For example, C:\Example.xlsx. This file displays all the tables in the Excel file on the next screen where you select what tables to add to the ElastiCube.

If you need to access any other folder or are having issues connecting and using the application, submit a ticket through Support.sisense.com.
Restarting Sisense Services

There may be scenarios where you need to restart or stop a service on the Sisense Server. You can reset the Sisense services through the following site:
https://mysite.sisense.com:8444

**Note:** Your firewall must allow traffic to the Sisense machine on port 8444.

**To restart a Sisense service:**

Open your Restart Service page at the following address:
https://mysite.sisense.com:8444, where mysite is your domain.

Enter your cloud user name without the cloud\ domain in the beginning. For example, if its cloud\username enter just username.

Enter the cloud password.

To restart a service, click **Restart** and wait for 30 seconds.

Click **Check Status** to make sure the service is back up.

To learn more about these services, click **here**.
Welcome to Sisense Labs

Sisense Labs offers you opportunities to evaluate and implement our latest, most exciting technical capabilities and product ideas – while they are still being conceived and developed. Sisense Labs provides benefits for both you and the Sisense Labs team. Data driven companies get to try out cool new features before they are available anywhere else. Just as important, when you test out a Sisense Labs product, you become part of the development process. Your feedback helps Sisense shape emerging products to best suit your needs and the market.

With a strong emphasis on innovation, short development cycles and open feedback channels, the projects in Sisense Labs are intended to showcase our most promising new ideas. The Sisense Labs team coordinates and assists with bringing the innovation from an idea to proof of concept, to a supported product with full documentation and tutorials.

Currently, you can test drive the following Sisense Labs’ products:

- Sisense BloX
- Sisense Boto
- Sisense Bulb
- Amazon Echo
Overview
Breaking Changes

Sisense BloX 2.0 includes major infrastructure changes. These changes will affect how your Sienese BloX widgets from earlier versions are displayed. Sisense encourages you to upgrade to the latest version of Sisense BloX as this version provides more flexibility, allowing you to customize every element in your widgets.

The following are a list of breaking changes that you can expect when upgrading to the latest version of Sisense BloX:

- Styling and layout changes. You should verify that the styling of your Sisense BloX widgets works as expected
- The value of `<style>` is no longer a string and should now include CSS
- Values of attributes are now case-sensitive
- Conditional formatting includes a max value
- `JumpToDashboard` and lightboxes are not supported with SisenseJS

If you have an earlier version of Sisense BloX or want to install an earlier version, you can find the documentation [here](https://example.com).

Sisense’s BloX lets you create dynamic, rich widgets that can turn your dashboard into an interactive business app. BloX are JSON objects rendered as HTML inside widgets on your dashboard. The idea behind Sisense BloX is to allow you to create dynamic and interactive content in your widgets, thus allowing you to gather insights and take immediate action from within your widget or dashboard. For example, you can create widgets that visualize an important KPI, while also displaying buttons that allow you to push a webhook to take an action you define.
Sisense BloX offers much more than just a customizable design. With Sisense BloX, you can:

- Add images to your widget
- Embed iFrames
- Add custom scripts
- Organize your content in containers and columns
- Add multiple types of actions
- Collect input from your users
Who Should Try This

Sisense BloX are JSON objects that render HTML and CSS. Therefore, to create your own interactive BloX, you should have a basic understanding of how JSON works, and have knowledge of HTML and CSS.

The following pages will show you how you can create interactive dashboards with Sisense BloX:

- Tutorials
- Installing Sisense BloX
- How Sisense BloX Works
- Creating Interactive Dashboards
- Customizing Your BloX Widget’s Look and Feel
- Embedding BloX Widgets
- Sisense BloX Reference

Limitations

Sisense BloX requires Sisense V7.2 or later

If you have users using Internet Explorer or if you need to export Sisense BloX to PDF, use JS ES5 rather than ES6 in your custom actions as well as IE-compliant CSS syntax

BloX responsiveness is limited, meaning the sizes will be adjusted with background image size or the original widget size, but have some compatibility limitation due to the Sisense overall grid. You should test your BloX widgets in different screen sizes

Images from an external source may not load properly in the Excel or PDF reports. See Adding Images for information on how to load images locally.

BloX are not compatible with Amazon’s Echo. This will be available in a future release

If you use the Custom-Style Plugin, this will overwrite the conditional color of BloX

Items and values are limited to 20 panels each

List view (Carousel: false) is limited to 100 first items

Sisense BloX v1.1 does not support any native Sisense visualization

Once BloX is installed, refresh plugin list on the Admin page to see it displayed

Base64 as an image will cause the widget to load slowly
**BloX for Mobile**

BloX responsiveness is limited. For optimal view, Sisense suggests you design widgets for desktop or widgets for mobile,

Mobile best practice:
- Choose images that can fit mobile screens or keep the background as a color
- Keep all text below 10 font size
- Make a clear Call-to-action to be displayed on the center
- Limit your items into 1 column and no more than 5 different text/data elements
- Limit all icons to Small
- Social feed template is not supported

Click [here](#) for Terms and Conditions.
Installing Sisense BloX

Download and install Sisense BloX 2.1.0. If you want to install an earlier version, you can find the documentation here. Once the installation is complete, the BloX widget is added to your list of widgets.

Note: If Sisense BloX does not appear in your list of plugins, in Windows,
restart the Sisense.Plugins service.
Installation in Multi-Node Environments

If you have implemented a multi-node environment and want to install Sisense BloX, you should run the BloX installer on the server that runs the plugin service. Typically, this is the build node.

If you have multiple plug-in services running on multiple machines, you should install BloX on each machine that runs the Sisense.Plugin service.

**Note:** This may affect your saved templates and in some cases will show only limited saved templates in Sisense BloX. Contact your CSM and [ron.oren@sisense.com](mailto:ron.oren@sisense.com) if you experience this issue.
BloX Design Panel

This page describes the features and functionality within the BloX Design Panel. After installing Sisense BloX, the first step is to create a new widget or edit an existing widget and to choose the widget type BloX. This transforms the Widget Editor into the BloX Design Panel, which contains the following features and functionality:

**Layout**: Displays the current state of your visualization as defined in the BloX Editor and Configuration Editor.

**Filters**: Add dashboard and widget filters to your Sisense BloX widgets. For more information about filters, see [Creating Widget Filters](#) and [Creating Dashboard Filters](#).

**Templates**: Contains a list of predefined templates Sisense has designed that you can edit to fit your use case. The easiest way to design BloX widgets is to choose the template closest to your use case and modify it. However, you can create one by deleting the content from the editor and using the code snippets to create your own templates. After you create a template, they can be exported and imported.

**Editor**: Contains the HTML JSON objects that determine the content of your BloX, including text and interactive actions such as buttons. This is where
you provide the content of your BloX and describe what actions your BloX can do. See Creating Interactive Dashboards for more information.

**Configuration Editor:** Contains CSS JSON objects that determine the look and feel of your cards. See Customizing Your BloX Widget's Look and Feel for more information.

**Snippets:** Contains predefined code snippets for BloX elements that you can copy into the editor to add those elements to your widget. From the BloX Design Panel, you can create your own cards by customizing predefined templates created by Sisense or by creating your own templates from scratch.

**Actions:** Contains predefined code samples for creating actions. You can create your own actions as well.
Creating Interactive Dashboards

While you can continue using Sisense's default widgets in your BloX dashboard, the real value of BloX is turning your dashboard into an interactive business app. The typical work flow for creating BloX widgets is to take a pre-made template, and then modify it by adding content, snippets, and actions. After you have finished modifying the template, you can save it and start creating new widgets from it.

The sections below explain how you can use in Sisense BloX to transform your dashboard.

**BloX Widgets**: Describes how to build BloX widgets

**Templates**: Describes how to modify and create templates

**Adding Content to Widgets**: Describes how to add content to your widgets using snippets and actions.

If you haven't already, you may want to review the [tutorials](#) for more information.

**BloX Widgets**

Interactive dashboards in Sisense consist of BloX widgets. In the Editor, you define the content of your BloX widgets.
BloX widgets are built on schemas that have the following basic structure:

**Body**: The body of your BloX is made up of building-blocks known as elements. Elements can be composed in a variety of arrangements to create many types of BloX.

**Actions**: Many BloX have a set of actions that you can take. This property describes the actions that typically get rendered in an “action button” at the bottom of your widget.

The example below illustrates the JSON structure common to all BloX:

```json
{
    "style": "",
    "script": "",
    "title": "",
    "body": [
        {
            "type": "TextBlock",
            "text": "This is my Sisense InfoCard"
        },
        {
            "type": "Image",
            "Url": "https://cdn.sisense.com/dotcom/images/sisense-logo.png"
        }
    ],
    "conditions": [
        {
            "minRange": "-Infinity",
            "maxRange": 2000,
            "backgroundColor": "#fd6e69",
        }
    ],
    "carouselAnimation": {
        "delay": 0
    },
    "showCarousel": true
}
```
At the top are two keys, style and script, for adding your own JavaScript and CSS if you want to use custom classes in your BloX widgets. Below is the body element that behaves just as a body element does in a typical web page. Within the body element, each element has a **type** property that determines what type of object it is. In the example above, there are two elements; a TextBlock and an Image:

- **TextBlock**: Adds a block of text with properties to control what the text looks like.
- **Image**: Adds an image with properties to control what the image looks like.

As in web pages, in BloX widgets, every element stacks vertically and expands to the width of its parent. You can use snippets like containers, columns, and column sets to add multiple elements side-by-side. Containers act as HTML divs where you can define a collection of elements. Column Sets contain columns. Within a Column Set, each column can be placed next to another column to allow you to add multiple elements on the same row instead of stacking them one on top of the other.

Other elements you may use are Input and Action elements for collecting and submitting data. These elements and more are described in the Reference page. You can build a widget based on this structure, but each template pre-defined by Sisense already uses this structure, so Sisense recommends using these templates and modifying them for your use case.

**Templates**

In the right corner of the BloX Design Panel, under the Design tab you can see your templates.
Templates are created by Sisense, but you can modify these to create your own templates. Templates are pre-built widgets with content and actions.

The easiest way to work with Sisense BloX is to modify templates that are the closest to your use case. After selecting a template, the Layout, Editor, and Configuration Editor are updated with the content of your template. In the Editor, you can add the content that is displayed in your widget. You can create this yourself as described in the BloX Widgets section, however, it is much easier to use snippets and actions to modify them as describe in sections Snippets and Actions.

The procedures below describe how to save your templates after editing them and import and export them.
Saving Templates

From the menu in the right corner of the editor you have two options for saving templates:

**Save Changes**: Saves your changes to the current template you are working on.

**Save As New Template**: Saves your changes in a new template. If you select this option you are prompted to enter a name for the template. The template is then added to your list of templates above.
Exporting Templates

You can export your templates and import them later or share them with other Sisense users so they can add them to their dashboards. Templates are exported as JSON files that include the content under the Editor and Configuration tabs (HTML and CSS).

To export a template,

In the BloX Design Panel, click the download icon and select Export Template.

You are then prompted to download the JSON file that represents your template.
Importing Templates

You can import templates (JSON files) that you have exported or that you downloaded.

**To import a template:**

In the BloX Design Panel, click the menu and select **Import Template**.

Navigate to your exported template JSON file and select it. Click **OK** to upload it. The template is uploaded and added to your list of templates.
Deleting Templates

If you no longer need a template, you can delete it. Before deleting a template, you may want to export it in case you need to use it in the future.

To delete a template:
In the BloX Design Panel, click the menu and select Delete Template. The template is removed and no longer visible in the BloX Design Panel.

Adding Content to Widgets

Sisense makes it easy to add content to your widgets by providing code snippets for the supported elements described above. With snippets you can build a complete widget or edit an existing BloX template by inserting the relevant elements.
If you want to create a new BloX widget from scratch, you can begin with the BloX Main snippet, and then add additional snippets to it. This snippet contains the mandatory keys for any BloX widget.

```json
{
    "title": "BloX",
    "showCarousel": true,
    "backgroundImage": "",
    "body": [
    {
        "type": "Container",
        "items": [
        {
            "type": "TextBlock",
            "text": "This is my Sisense BloX Card"
        }
        ]
    }
}
```
From this basic snippet, more snippets can be copied and pasted into the BloX Editor and then modified according to your needs.
For example, the TextBlock snippet can be added to your template to add a text object. You can then modify the snippet as required to fit the style of the rest of your BloX.

```json
{
  "type": "TextBlock",
  "text": "{panel:Date}",
  "spacing": "medium",
  "horizontalAlignment": "center",
  "color": "default",
  "size": "medium",
  "weight": "light",
  "lineColor": "#414241"
}
```

It’s important to remember that snippets are JSON objects that while they can be added anywhere to your template, they must not break the structure of the template. For example, if you add the text block snippet shown above to an existing template, you need to add a comma to the previous object, as you would in any JSON file:

```
{
  "type": "TextBlock",
  "size": "light",
  "weight": "light",
  "horizontalAlignment": "center",
  "spacing": "medium",
  "color": "default",
  "text": "{panel:Google} views
```
Text blocks display text in your widget and give you a way to provide context to your users.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>string</td>
<td>No</td>
<td>Text to be displayed. You can use placeholders to display values dynamically from your data. For example, {panel:Date} refers to the value of the Date column.</td>
</tr>
</tbody>
</table>
Adding Actions

You can add actions to your widgets. Sisense supports a variety of actions such as applying filters to a dashboard or submitting a form through a webhook. You can read more about supported actions [here](#).

You can create your own custom actions. Actions that you create consist of custom JavaScript you provide, see [Creating Custom Actions](#) for more information. In addition, you can leverage [Sisense’s JavaScript API](#) for adding more functionality to your dashboards.

Once you create an action, it's added to your code snippets where Designers can add the action to their widgets.

**To create your own actions:**

Open a BloX widget.

In the Editor, open the menu and select **Create Action**.

The Action Editor is displayed.

Enter a name for your action. This name is displayed when your action is added to the **My Action** in the Snippets section.
Click **Next**. Sisense opens an Action template with two parameters, type and title.

```json
{
  "type": "Action",
  "title": "Title"
}
```

In the Action Editor, enter the code for your action. Click **Create**. The new action is added to the My Actions section of your snippets.

After the action has been added, you can open its menu to edit or delete the action.
Adding Images

Images are useful for providing context or for branding your dashboard. Like Text blocks, images have their own element, as shown in the snippet below.

```json
{
   "type": "Image",
   "url": "/plugins/BloX/blox-images/ConditionalCard/green_boto_bg.png",
   "size": "auto"
}
```

There are two options for hosting images, internally on the Sisense Server or externally. If you host the images externally, the value of "url" is the full external address of the image.

When exporting your dashboard as a PDF or image, images hosted externally may not be displayed properly in your report. You can host images locally on your Sisense Server in the /resources/ directory located at C:\Program Files\Sisense\app\resources. Images hosted here will be displayed properly in your dashboard and when exported to your reports. You can refer to this images with a shortcut as shown below:

```
"url": "/resources/sample.png",
```

If you want to use as a background image, you can do so by defining the URL of backgroundImage:

```
"backgroundImage":
"https://s3.amazonaws.com/sisenseblox/Sample+-SenseBUZZ+/Background+Imgs/Header.png",
```

However, the background image will only cover up to the title area in your widget. If you want to display a background image that covers the title area of your widget as well, set `display` to "none".
```
"titleStyle": [
{
  "display": "none"
},
],
```

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;Image&quot;</td>
<td>Yes</td>
<td>Must be Image.</td>
</tr>
<tr>
<td>url</td>
<td>string</td>
<td>Yes</td>
<td>The URL to the image. To add your own images to BloX directly from the Sisense Server, you can upload them to C:\Program Files\Sisense\app\resources and insert them in BloX by referring to the /resources/ directory. &quot;backgroundImage&quot;: &quot;/resources/sample.png&quot;,</td>
</tr>
<tr>
<td>size</td>
<td>object</td>
<td>No</td>
<td>Controls the approximate size of the image. The physical dimensions will vary per host. Specify &quot;auto&quot; for true image dimension, or &quot;stretch&quot; to force it to fill the container.</td>
</tr>
</tbody>
</table>
Adding Sparklines

Sparklines allow you to add a tiny simple graph over a widget that conveys the basic shape of your data in a meaningful, but compact way. Sparklines are useful for showing trends in your data.

You can add sparklines to any BloX widget by defining the values of the sparkline.

```json
{
    "type": "TextBlock",
    "text": "<span class='sparklines' width='200' height='50' lineWidth='3' fillColor='' minSpotColor='#000' maxSpotColor='#000' sparkType='line' Color='#000'>{spark:Data}</span>
}
```

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;TextBlock&quot;</td>
<td>Yes</td>
<td>Must be TextBlock</td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Yes</td>
<td>Span with the class 'sparklines' that defines the dimensions and values of your sparkline. You can use a placeholder as the values of your sparkline, {spark:Data}, or enter each value hardcoded separated by a comma, 3, 15, 32, 75.</td>
</tr>
</tbody>
</table>
Embedding iFrames

You can embed iFrames into your widgets. This is useful for display websites, social media feeds, or streaming videos. To display content in an iFrame, you add a Text block with the value of the text field the URL you want to display in an iFrame element.

Note: Each attributes included in the value of iframe src must be wrapped with single quotation marks 'x'. See the example below for more information.

```
"type": "TextBlock",
ps%3A%2F%2Fwww.facebook.com%2Fsisense&tabs=timeline&width=500&height=600&small_header=true&adapt_container_width=ture&hide_cover=false&show_facepile=true&appId=22667597433047' width='500' height='600' style='border:none;overflow:hidden' scrolling='no' frameborder='0' allowTransparency='true' allow='encrypted-media'></iframe>"
```

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;TextBlock&quot;</td>
<td>Yes</td>
<td>Must be TextBlock.</td>
</tr>
<tr>
<td>text</td>
<td>string</td>
<td>Yes</td>
<td>The URL of the iFrame you want to open.</td>
</tr>
</tbody>
</table>
Adding Carousels

In many cases, there is more than one item in the result set. If you want to show them all, you can display them as a list as shown below:

![Carousel Example]

**Note:** Due to performance concerns there is a limit of 100 list items. Another option is to display your data in a carousel that your users can scroll through as shown in the example below:
In the Card Editor, you can change the value of the boolean key “showCarousel” to `true` to display a single set of items that users can toggle through or `false` to display a list.

```json
{
    "title": "Multi Purpose Card",
    "showCarousel": true,
    "panels": [
    
}
```

In addition, you can modify the amount of time it takes to cycle between values in your carousel with the Carousel Animation object or whether or not to display arrows. The default speed is 500 milliseconds. You can increase or decrease by adding the Carousel Animation snippet and changing the value of `delay` as needed.

To hide the carousel arrows, set the boolean “showButtons” to `false`.

```json
{
    "title": "BloX",
    "showCarousel": true,
    "carouselAnimation": {
        "delay": 500,
        "showButtons": true
    },
}
```
Using Conditionals

Conditionals allow you to create thresholds that trigger events in your widgets when met. In the example below, the background color is set to gold when the placeholder {panel:Total Revenue} is greater than 1000.

```json
{
    "title": "BloX",
    "showCarousel": true,
    "conditions": [
        {
            "minRange": "-Infinity",
            "maxRange": 1000,
            "backgroundColor": "#fd6e69",
            "image": "https://image.ibb.co/d5vPpe/red_boto_bg.png"
        },
        {
            "minRange": 1000,
            "maxRange": 10000,
            "backgroundColor": "#ffcb05",
            "image": "https://image.ibb.co/dNuvvK/yellow_boto_bg.png"
        },
        {
            "minRange": 10000,
            "maxRange": "Infinity",
            "backgroundColor": "#3adcca",
            "image": "https://image.ibb.co/bY68FK/green_boto_bg.png"
        }
    ]
}
```

There are two steps to implementing conditionals. The first step is to add the condition_data class to an object. In the Conditional Formatting Target snippet, you can see several examples like the one below. In this example, the
The condition_data class has been added to a TextBlock. The condition is triggered by the value of "text" in this case, a dynamic value taken from the dashboard panel.

```json
{
  "type": "TextBlock",
  "class": "condition_data",
  "text": "\{panel:Data\}"
}
```

The second step is to define what is the condition that must be met to trigger an action, and its results. Sisense supports two types of conditions, numeric and strings.

**Numeric Conditions**

Numeric conditions include a range of numbers. In the Conditional Range snippet, the min and max ranges are defined and what happens when the value of the Conditional Formatting Target reaches a certain number. In the example above, if Total Revenue ("text": "\{panel:Total Revenue\}") equals 1000 or less, then a red background image is displayed.

There are two types of values you can enter, integers, such as 10, 20, 30 etc. or percentages, such as 20%, 30%, 40%, etc. For integers, simply add the value. If you choose to enter a percentage, add the value as a decimal, for example, the value 0.2 represents 20% in your widget.

**Note:** For the best results, your min and max range should be continuous. For example, for negative conditions, set your minRange to "-Infinity" and your maxRange to 0. For neutral conditions, set minRange to 0 and maxRange to 10, and for positive conditions, minRange 10 and maxRange "Infinity". You can set the values as you like, but ideally, all the ranges of all conditions should create a continuous range with no gaps and no overlaps.

**String Conditions**
In addition, you can define strings that also trigger events. For example, in the Conditional Text snippet, you can redefine how your widget appears when “textEquals” equals a string you define as shown in the sample below, “Blox”. The results of the action are then defined below in the same object. The important thing to remember is that the value must be a string and placed within quotes “"".

```json
{
    "conditions": [
    {
        "textEquals": "Blox",
        "backgroundColor": "",
        "hide": false,
        "fontSize": "",
        "fontWeight": "",
        "color": "",
        "image": ""
    },
    ]
}
```
Customizing Your BloX Widget's Look and Feel

You can customize the look and feel of your Sisense BloX widgets. The look and feel of your templates are defined in the Configuration Editor. In addition, you can customize any element in your widgets by adding the <style> key and using inline CSS as the value of the <style> key. Both methods are described below.
Configuration Editor

The Configuration Editor contains a JSON structure that renders the key:value pairs as CSS in your card. From the Configuration Editor, you can define the look and feel of your template.

Each template has its own JSON structure and elements. Some of these elements can be referenced from the Card Editor, but most can be set in the Configuration Editor.

For example, the color key inside the Card Editor of the multiple_indicators card refers to the “attention” object. This object defines the font color of the text displayed in the card and is located in the Configuration Editor.

Layout Editor

"type": "Column",
    "items": [ 
        
    
},

Configuration Editor

"emphasis": { 
  "foregroundColors": { 
  "default": { 
  "normal": "#333333", 
  "subtle": "#EE333333" 
  }, 
  
}
You can modify the hex values of these colors in the Configuration Editor, or remove the reference in the BloX Editor and define your own colors specific for each element, such as TextBlocks.

Another example is redefining the background color of your BloX. The following is an example taken from the ‘Sparkline’ template:
The ‘Sparkline’ template has a white background color by default. You can change this to yellow by modifying the value of the backgroundColor key.

"backgroundColor": ":FFFFF"

To

"backgroundColor": "yellow"

The Configuration Editor provides most of the CSS you need to redefine your BloX, but with Sisense BloX, you have complete freedom to customize the look and feel to meet your use-case using custom CSS written in JSON key:value pairs.
Customizing Elements

For each element in your Sisense BloX widgets, you can customize the look and feel by adding the <style> key with some inline CSS as the value. This is useful if you want to specify custom styling that is distinct from the rest of your template or dashboard.

The example below shows a text block that has been customized through the <style> key:

```
"body": [ 
    {
      "type": "Container",
      "items": [
      {
        "type": "TextBlock",
        "text": "❤️️ Welcome to BloX ❤️️ ",
        "style": {
          "text-align": "center",
          "font-weight": "bold",
          "font-size": "24px",
          "margin": "100px"
        }
      }
      ]
    }
  ]
```
Embedding BloX Widgets

After you have created a BloX widget, you can embed the widget directly into your dashboard, website, or application.

Sisense provides three easy ways to embed a BloX widget, through an iFrame, QR code, or direct URL.

**To generate the code to embed a BloX Widget:**
Open the widget you want to embed in the BloX editor.
Click the download icon.

Select how you want to embed your BloX widget, iFrame, QR code, or direct URL.

After you select the relevant option, the code is displayed for you to copy.
Sisense BloX Reference

The following tables describe the properties for the JSON objects that are used or can be used in the Sisense BloX.
Snippets
Containers

**Blox Main**
Card schema for BloX.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>style</td>
<td>string</td>
<td>No</td>
<td>Space to enter your own internal CSS stylesheet.</td>
</tr>
<tr>
<td>script</td>
<td>string</td>
<td>No</td>
<td>Space to enter your own custom JavaScript, similar to &lt;script&gt; &lt;/script&gt; tags in an HTML header.</td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>No</td>
<td>Name displayed as the title of the widget.</td>
</tr>
<tr>
<td>body</td>
<td>array[]</td>
<td>Yes</td>
<td>The Card elements to show in the primary card region.</td>
</tr>
<tr>
<td>actions</td>
<td>Action[]</td>
<td>No</td>
<td>The Actions to show in the BloX's action bar.</td>
</tr>
<tr>
<td>showCarousel</td>
<td>boolean</td>
<td>No</td>
<td>Indicates if an element should be displayed as a carousel instead of a list.</td>
</tr>
</tbody>
</table>

**Container**
Containers group items together.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;Container&quot;</td>
<td>Yes</td>
<td>Must be Container.</td>
</tr>
<tr>
<td>style</td>
<td>string</td>
<td>No</td>
<td>Style hint for Container.</td>
</tr>
<tr>
<td>items</td>
<td>array[]</td>
<td>Yes</td>
<td>The Card Elements to include in the Column.</td>
</tr>
</tbody>
</table>

**Column Set**
ColumnSet divides a region into Columns allowing elements to sit side-by-side.
<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;ColumnSet&quot;</td>
<td>Yes</td>
<td>Must be ColumnSet.</td>
</tr>
<tr>
<td>columns</td>
<td>Column[]</td>
<td>No</td>
<td>The array of Columns to divide the region into.</td>
</tr>
</tbody>
</table>

**Column**

Defines a container that is part of a ColumnSet.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;Column&quot;</td>
<td>Yes</td>
<td>Must be Column.</td>
</tr>
<tr>
<td>items</td>
<td>array[]</td>
<td>Yes</td>
<td>The Card Elements to include in the Column.</td>
</tr>
</tbody>
</table>
## Attributes

*Click here to expand*

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>backgroundColor</td>
<td>string</td>
<td>No</td>
<td>Determines the background color of a widget.</td>
</tr>
<tr>
<td>backgroundImage</td>
<td>string</td>
<td>No</td>
<td>Adds a background image to the widget. For more information, see <a href="#">Adding Images</a>.</td>
</tr>
<tr>
<td>horizontalAlignment</td>
<td>string</td>
<td>No</td>
<td>Controls how Items are horizontally positioned within their container.</td>
</tr>
<tr>
<td>verticalContentAlignment</td>
<td>string</td>
<td>No</td>
<td>Controls how Items are vertically positioned within their container.</td>
</tr>
<tr>
<td>spacing</td>
<td>string</td>
<td>No</td>
<td>Controls the amount of spacing between this element and the previous element.</td>
</tr>
<tr>
<td>separator</td>
<td>boolean</td>
<td>No</td>
<td>The Separator object type describes the look and feel of a separation line between two elements.</td>
</tr>
<tr>
<td>carousel</td>
<td>object</td>
<td>No</td>
<td>Converts your list into a carousel. See <a href="#">Adding Carousels</a>.</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>conditional range</td>
<td>string</td>
<td>No</td>
<td>Allows you to create thresholds that trigger events in your widgets when met. For more information see <a href="#">Using Conditionals</a>.</td>
</tr>
<tr>
<td>conditional text</td>
<td>string</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>conditional formatting</td>
<td>string</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>conditional formatting target</td>
<td>string</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>widget title style</td>
<td>object</td>
<td>No</td>
<td>Defines how widget titles are displayed.</td>
</tr>
<tr>
<td>hide widget title style</td>
<td>string</td>
<td>No</td>
<td>If you want to display a background image that covers the title area of your widget as well, set display to &quot;none&quot;.</td>
</tr>
</tbody>
</table>
Text Styling

*Click here to expand.*

| Items      | Type    | Required | Element                                                        |
|------------|---------|----------|                                                               |
| Font Size  | string  | No       | Controls the size of the text.                                 |
| Font Color | string  | No       | Controls the color of TextBlock Items.                        |
| Font Weight| string  | No       | Controls the weight of TextBlock Items.                       |
| Text Wrap  | boolean | No       | True if text is allowed to wrap.                              |
Components

**Sparkline Charts**

Sparklines allow you to add a tiny simple graph over a widget that conveys the basic shape of your data in a meaningful, but compact way. Sparklines are useful for showing trends in your data. You can add sparklines to any BloX widget by defining the values of the sparkline.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;TextBlock&quot;</td>
<td>Yes</td>
<td>Must be &quot;TextBlock&quot;.</td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td>No</td>
<td>A unique ID associated with the element.</td>
</tr>
<tr>
<td>class</td>
<td>object</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>text</td>
<td>object</td>
<td>Yes</td>
<td>Span with the class 'sparklines' that defines the dimensions and values of your sparkline. You can use a placeholder as the values of your sparkline, {spark:Data}, or enter each value hardcoded separated by a comma, 3, 15, 32, 75.</td>
</tr>
</tbody>
</table>

**iFrame**

You can embed iFrames into your widgets. This is useful for display websites, social media feeds, or streaming videos. To display content in an iFrame, you add a Text block with the value of the text field the URL you want to display in an iFrame element.

**Note:** Each attributes included in the value of iframe src must be wrapped with single quotation marks 'x'. See the example below for more information.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property</td>
<td>Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>type</td>
<td>&quot;TextBlock&quot;</td>
<td>Yes</td>
<td>Must be TextBlock.</td>
</tr>
<tr>
<td>text</td>
<td>string</td>
<td>Yes</td>
<td>The URL of the iFrame you want to open.</td>
</tr>
</tbody>
</table>

**Image**

The Image Item allows for the inclusion of images in BloX.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;Image&quot;</td>
<td>Yes</td>
<td>Must be Image.</td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td>No</td>
<td>A unique ID associated with the element.</td>
</tr>
<tr>
<td>class</td>
<td>string</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>url</td>
<td>string</td>
<td>Yes</td>
<td>The URL to the image.</td>
</tr>
<tr>
<td>altText</td>
<td>string</td>
<td>No</td>
<td>Alternate text for the image for accessibility.</td>
</tr>
<tr>
<td>size</td>
<td>string</td>
<td>No, default: auto</td>
<td>Controls the approximate size of the image. The physical dimensions will vary per host. Specify &quot;auto&quot; for true image dimension or &quot;stretch&quot; to force it to fill the container.</td>
</tr>
</tbody>
</table>

Additional Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>horizontalAlignment</td>
<td>string</td>
<td>No</td>
<td>Controls how Items are horizontally positioned within their container.</td>
</tr>
<tr>
<td>opacity</td>
<td>object</td>
<td>No</td>
<td>Controls how much opacity (0-1 in %) to apply to images. “type”: “image”, “opacity”:0.7</td>
</tr>
<tr>
<td>selectAction</td>
<td>object</td>
<td>No</td>
<td>An Action that will be invoked when</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>---------</td>
<td>----------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>the Image is tapped or selected.</td>
</tr>
<tr>
<td>style</td>
<td>string</td>
<td>No</td>
<td>Controls the way Images are displayed.</td>
</tr>
<tr>
<td>spacing</td>
<td>string</td>
<td>No</td>
<td>Controls the amount of spacing between this element and the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>previous element.</td>
</tr>
<tr>
<td>separator</td>
<td>boolean</td>
<td>No, default: false</td>
<td>The Separator object type describes the look and feel of a separation line between two elements.</td>
</tr>
</tbody>
</table>

**Text Block**

The TextBlock Item allows for the inclusion of text with various font sizes, weight and color, in BloX.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;TextBlock&quot;</td>
<td>Yes</td>
<td>Must be TextBlock.</td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td>No</td>
<td>A unique ID associated with the element.</td>
</tr>
<tr>
<td>class</td>
<td>string</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>text</td>
<td>string</td>
<td>Yes</td>
<td>The actual text to display.</td>
</tr>
</tbody>
</table>

**Additional Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>color</td>
<td>string</td>
<td>No</td>
<td>Controls the color of TextBlock Items.</td>
</tr>
<tr>
<td>horizontalAlignment</td>
<td>string</td>
<td>No</td>
<td>Controls how Items are horizontally positioned within their container.</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>------------------------------------------------------------</td>
</tr>
<tr>
<td>size</td>
<td>string</td>
<td>No</td>
<td>Controls the size of the text.</td>
</tr>
<tr>
<td>weight</td>
<td>string</td>
<td>No</td>
<td>Controls the weight of TextBlock Items.</td>
</tr>
<tr>
<td>wrap</td>
<td>boolean</td>
<td>No</td>
<td>True if text is allowed to wrap.</td>
</tr>
<tr>
<td>spacing</td>
<td>string</td>
<td>No</td>
<td>Controls the amount of spacing between this element and the previous element.</td>
</tr>
<tr>
<td>separator</td>
<td>boolean</td>
<td>No, default: false</td>
<td>The Separator object type describes the look and feel of a separation line between two elements.</td>
</tr>
</tbody>
</table>

### 2-Column List
The FactSet Item makes it simple to display a series of facts (e.g. name/value pairs) in a tabular form.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>facts</td>
<td>Fact[]</td>
<td>Yes</td>
<td>The array of Facts.</td>
</tr>
<tr>
<td>type</td>
<td>&quot;FactSet&quot;</td>
<td>No</td>
<td>Must be FactSet.</td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td>No</td>
<td>A unique Id associated with the element.</td>
</tr>
<tr>
<td>spacing</td>
<td>string</td>
<td>No</td>
<td>Controls the amount of spacing between this element and the previous element.</td>
</tr>
<tr>
<td>separator</td>
<td>boolean</td>
<td>No, default: false</td>
<td>The Separator object type describes the look and feel of a separation line between two elements.</td>
</tr>
</tbody>
</table>

**Fact**

Describes a Fact in a FactSet as a key/value pair.
### Property Types

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;Fact&quot;</td>
<td>Yes</td>
<td>Must be Fact.</td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Yes</td>
<td>The title of the fact.</td>
</tr>
<tr>
<td>value</td>
<td>string</td>
<td>Yes</td>
<td>The value of the fact.</td>
</tr>
</tbody>
</table>

**HTML Text Block**

The TextBlock Item allows for the inclusion of text with various font sizes, weight and color, in BloX.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;TextBlock&quot;</td>
<td>Yes</td>
<td>Must be TextBlock.</td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td>No</td>
<td>A unique ID associated with the element.</td>
</tr>
<tr>
<td>class</td>
<td>string</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>text</td>
<td>string</td>
<td>Yes</td>
<td>The actual text to display.</td>
</tr>
</tbody>
</table>

**Additional Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>color</td>
<td>string</td>
<td>No</td>
<td>Controls the color of TextBlock Items.</td>
</tr>
<tr>
<td>horizontalAlignment</td>
<td>string</td>
<td>No</td>
<td>Controls how Items are horizontally positioned within their container.</td>
</tr>
<tr>
<td>isSubtle</td>
<td>boolean</td>
<td>No</td>
<td>Indicates whether the color of the text should be slightly toned down to appear less prominent.</td>
</tr>
<tr>
<td>maxLines</td>
<td>number</td>
<td>No</td>
<td>When Wrap is true, you can specify the maximum number of lines to allow the textBlock to</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>----------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>size</td>
<td>string</td>
<td>No</td>
<td>Controls the size of the text.</td>
</tr>
<tr>
<td>weight</td>
<td>string</td>
<td>No</td>
<td>Controls the weight of TextBlock Items.</td>
</tr>
<tr>
<td>wrap</td>
<td>boolean</td>
<td>No</td>
<td>True if text is allowed to wrap.</td>
</tr>
<tr>
<td>spacing</td>
<td>string</td>
<td>No</td>
<td>Controls the amount of spacing between this element and the previous element.</td>
</tr>
<tr>
<td>separator</td>
<td>boolean</td>
<td>No, default: false</td>
<td>The Separator object type describes the look and feel of a separation line between two elements.</td>
</tr>
</tbody>
</table>
Input Fields

**Text Input**

Input.Text collects text from the user.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;Input.Text&quot;</td>
<td>Yes</td>
<td>Must be Input.Text.</td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td>Yes</td>
<td>ID for the value (will be used to identify collected input when SUBMIT is clicked).</td>
</tr>
<tr>
<td>class</td>
<td>string</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>No</td>
<td>Title of the button.</td>
</tr>
<tr>
<td>placeholder</td>
<td>string</td>
<td>No</td>
<td>Title Description of the input desired.</td>
</tr>
</tbody>
</table>

**Additional Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isMultiline</td>
<td>boolean</td>
<td>No</td>
<td>Do you want to allow multiple lines of input.</td>
</tr>
<tr>
<td>maxLength</td>
<td>number</td>
<td>No</td>
<td>Hint of maximum length characters to collect (may be ignored by some clients).</td>
</tr>
<tr>
<td>style</td>
<td>string</td>
<td>No</td>
<td>Style hint for Input.Text.</td>
</tr>
<tr>
<td>value</td>
<td>string</td>
<td>No</td>
<td>The initial value for a field.</td>
</tr>
<tr>
<td>spacing</td>
<td>string</td>
<td>No</td>
<td>Controls the amount of spacing between this element and the previous element.</td>
</tr>
<tr>
<td>separator</td>
<td>boolean</td>
<td>No, default: false</td>
<td>The Separator object type describes the look and feel of a separation line between two elements.</td>
</tr>
</tbody>
</table>

**Date Input**

Input.Date collects a date from the user.
### Property
<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;Input.Date&quot;</td>
<td>Yes</td>
<td>The type must be Input.Date.</td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td>Yes</td>
<td>ID for the value (will be used to identify collected input when SUBMIT is clicked).</td>
</tr>
<tr>
<td>class</td>
<td>string</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

**Additional Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>max</td>
<td>string</td>
<td>No</td>
<td>Hint of maximum value expressed in ISO-8601 format (may be ignored by some clients).</td>
</tr>
<tr>
<td>min</td>
<td>string</td>
<td>No</td>
<td>Hint of minimum value expressed in ISO-8601 format (may be ignored by some clients).</td>
</tr>
<tr>
<td>placeholder</td>
<td>string</td>
<td>No</td>
<td>Title Description of the input desired.</td>
</tr>
<tr>
<td>value</td>
<td>string</td>
<td>No</td>
<td>The initial value for a field expressed in ISO-8601 format.</td>
</tr>
<tr>
<td>spacing</td>
<td>string</td>
<td>No</td>
<td>Controls the amount of spacing between this element and the previous element.</td>
</tr>
<tr>
<td>separator</td>
<td>boolean</td>
<td>No, default: false</td>
<td>The Separator object type describes the look and feel of a separation line between two elements.</td>
</tr>
</tbody>
</table>

**Time Input**

Input:Time collects the time from the user.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;Input.Date&quot;</td>
<td>Yes</td>
<td>The type must be Input.Date.</td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td>Yes</td>
<td>ID for the value (will be used to identify collected input when SUBMIT is clicked).</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>class</td>
<td>string</td>
<td>No</td>
<td>class</td>
</tr>
<tr>
<td>Additional Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>max</td>
<td>string</td>
<td>No</td>
<td>Hint of maximum value expressed in ISO-8601 format (may be ignored by some clients).</td>
</tr>
<tr>
<td>min</td>
<td>string</td>
<td>No</td>
<td>Hint of minimum value expressed in ISO-8601 format (may be ignored by some clients).</td>
</tr>
<tr>
<td>placeholder</td>
<td>string</td>
<td>No</td>
<td>Title Description of the input desired.</td>
</tr>
<tr>
<td>value</td>
<td>string</td>
<td>No</td>
<td>The initial value for a field expressed in ISO-8601 format.</td>
</tr>
<tr>
<td>spacing</td>
<td>string</td>
<td>No</td>
<td>Controls the amount of spacing between this element and the previous element.</td>
</tr>
<tr>
<td>separator</td>
<td>boolean</td>
<td>No, default: false</td>
<td>The Separator object type describes the look and feel of a separation line between two elements.</td>
</tr>
</tbody>
</table>

**Number Input**

Input.Number collects a number from the user.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;Input.Number&quot;</td>
<td>Yes</td>
<td>The type must be Input.Number.</td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td>Yes</td>
<td>ID for the value (will be used to identify collected input</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>class</td>
<td>string</td>
<td>No</td>
<td>Title Description of the input desired.</td>
</tr>
<tr>
<td>placeholder</td>
<td>string</td>
<td>No</td>
<td>Title Description of the input desired.</td>
</tr>
</tbody>
</table>

**Additional Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>max</td>
<td>number</td>
<td>No</td>
<td>Hint of maximum value (may be ignored by some clients).</td>
</tr>
<tr>
<td>min</td>
<td>number</td>
<td>No</td>
<td>Hint of minimum value (may be ignored by some clients).</td>
</tr>
<tr>
<td>value</td>
<td>number</td>
<td>No</td>
<td>The initial value for a field.</td>
</tr>
<tr>
<td>spacing</td>
<td>string</td>
<td>No</td>
<td>Controls the amount of spacing between this element and the previous element.</td>
</tr>
<tr>
<td>separator</td>
<td>boolean</td>
<td>No, default: false</td>
<td>The Separator object type describes the look and feel of a separation line between two elements.</td>
</tr>
</tbody>
</table>

**Radio Buttons**

Allows a user to input a Choice.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;Input.ChoiceSet&quot;</td>
<td>Yes</td>
<td>The type must be Input.ChoiceSet.</td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td>Yes</td>
<td>Must be radiotVal.</td>
</tr>
<tr>
<td>class</td>
<td>string</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------</td>
<td>--------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>style</td>
<td>style</td>
<td>No, default: &quot;compact&quot;</td>
<td>Style hint for Input.ChoiceSet.</td>
</tr>
<tr>
<td>placeholder</td>
<td>string</td>
<td>No</td>
<td>Title Description of the input desired.</td>
</tr>
<tr>
<td>choices</td>
<td>Input.Choice[]</td>
<td>Yes</td>
<td>Choice options.</td>
</tr>
</tbody>
</table>

**Additional Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isMultiSelect</td>
<td>boolean</td>
<td>No, default: false</td>
<td>Allow multiple choices to be selected.</td>
</tr>
<tr>
<td>value</td>
<td>number</td>
<td>No</td>
<td>The initial choice (or set of choices) that should be selected. For multi-select, specify a comma-separated string of values.</td>
</tr>
<tr>
<td>spacing</td>
<td>string</td>
<td>No</td>
<td>Controls the amount of spacing between this element and the previous element.</td>
</tr>
<tr>
<td>separator</td>
<td>boolean</td>
<td>No, default: false</td>
<td>The Separator object type describes the look and feel of a separation line between two elements.</td>
</tr>
</tbody>
</table>

**Checkbox**

Allows a user to input a Choice.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;Input.ChoiceSet&quot;</td>
<td>Yes</td>
<td>The type must be Input.ChoiceSet.</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------</td>
<td>----------</td>
<td>-------------------------------------------------------------------</td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td>Yes</td>
<td>Must be checkboxVal.</td>
</tr>
<tr>
<td>class</td>
<td>string</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>style</td>
<td>style</td>
<td>No, default: &quot;compact&quot;</td>
<td>Style hint for Input.ChoiceSet.</td>
</tr>
<tr>
<td>placeholder</td>
<td>string</td>
<td>No</td>
<td>Title Description of the input desired.</td>
</tr>
<tr>
<td>choices</td>
<td>Input.Choice[]</td>
<td>Yes</td>
<td>Choice options.</td>
</tr>
</tbody>
</table>

**Additional Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isMultiSelect</td>
<td>boolean</td>
<td>No, default: false</td>
<td>Allow multiple choices to be selected.</td>
</tr>
<tr>
<td>value</td>
<td>number</td>
<td>No</td>
<td>The initial choice (or set of choices) that should be selected. For multi-select, specify a comma-separated string of values.</td>
</tr>
<tr>
<td>spacing</td>
<td>string</td>
<td>No</td>
<td>Controls the amount of spacing between this element and the previous element.</td>
</tr>
<tr>
<td>separator</td>
<td>boolean</td>
<td>No, default: false</td>
<td>The Separator object type describes the look and feel of a separation line between two elements.</td>
</tr>
</tbody>
</table>

**Dropdown**

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
</table>

---

**Sisense Icon**
<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;Input.ChoiceSet&quot;</td>
<td>Yes</td>
<td>The type must be Input.ChoiceSet.</td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td>Yes</td>
<td>Must be selectVal.</td>
</tr>
<tr>
<td>class</td>
<td>string</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>style</td>
<td>style</td>
<td>No, default: &quot;compact&quot;</td>
<td>Style hint for Input.ChoiceSet.</td>
</tr>
<tr>
<td>placeholder</td>
<td>string</td>
<td>No</td>
<td>Title Description of the input desired.</td>
</tr>
<tr>
<td>choices</td>
<td>Input.Choice[]</td>
<td>Yes</td>
<td>Choice options.</td>
</tr>
<tr>
<td><strong>Additional Properties</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>isMultiSelect</td>
<td>boolean</td>
<td>No, default: false</td>
<td>Allow multiple choices to be selected.</td>
</tr>
<tr>
<td>value</td>
<td>number</td>
<td>No</td>
<td>The initial choice (or set of choices) that should be selected. For multi-select, specify a comma-separated string of values.</td>
</tr>
<tr>
<td>spacing</td>
<td>string</td>
<td>No</td>
<td>Controls the amount of spacing between this element and the previous element.</td>
</tr>
<tr>
<td>separator</td>
<td>boolean</td>
<td>No, default: false</td>
<td>The Separator object type describes the look and feel of a separation line between two elements.</td>
</tr>
</tbody>
</table>

**Dynamic Inputs**
Dynamic inputs allow you to display dynamic options for dropdowns, radio buttons, and checkbox inputs.

The Dynamic Input snippet displays a dropdown list and when you change the value of “choices” to match your Items, a list of values based on your data is displayed. The default value included in the snippet is choices:Title,Value where title is the name of the option displayed and Value is the ID of the option selected that is input when selected.

```json
{
    "type": "Input.ChoiceSet",
    "id": "selectVal",
    "class": "",
    "displayType": "compact",
    "value": "1",
    "choices": "{choices:Category,Id}"
}
```

For example, if your Items display Brand as a list of strings, these can be displayed in a dropdown by changing the value of “choices” to “{choices:Brand}” where Brand equals the title of the options to be displayed.
While a dropdown list is the default option for the Dynamic Inputs snippet, you can use the same functionality for radio buttons and checkboxes. For example, by changing the value of choices in the Checkbox or Radio Button snippets from an array to a string, you can display input options based on your data.

```json
{
    "type": "Input.ChoiceSet",
    "id": "checkboxVal",
    "class": "",
    "layout": "vertical",
    "isMultiSelect": true,
    "value": "1,3",
    "choices": "{choices:Brand}"
}
```

**Toggle**

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;Input.Toggle&quot;</td>
<td>Yes</td>
<td>Must be Input.Toggle.</td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Yes</td>
<td>Text to be displayed alongside the toggle option.</td>
</tr>
<tr>
<td>valueOn</td>
<td>string</td>
<td>No</td>
<td>Enter &quot;true&quot; to set the toggle to selected or enabled by default.</td>
</tr>
<tr>
<td>valueOff</td>
<td>string</td>
<td>No</td>
<td>Enter &quot;false&quot; to set the toggle to unselected or disabled by default.</td>
</tr>
<tr>
<td>value</td>
<td>string</td>
<td>Yes</td>
<td>A value you want to send when the toggle is selected.</td>
</tr>
</tbody>
</table>

**Slider**

<p>| Property | Type | Required | Description |
|----------|------|----------|-------------|-------------|
| type     |      |          |             |             |</p>
<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;TextBlock&quot;</td>
<td>Yes</td>
<td>Must be TextBlock.</td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>class</td>
<td>string</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>text</td>
<td>Slider array</td>
<td>Yes</td>
<td>List of values in the slider. Use 'sliderData' prefix within slider configuration block to connect it to metadata item from 'Items' panel.</td>
</tr>
</tbody>
</table>

**Additional Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>items</td>
<td>array</td>
<td>No</td>
<td>Connection to metadata item.</td>
</tr>
<tr>
<td>range</td>
<td>string</td>
<td>No</td>
<td>Enable slider range mode.</td>
</tr>
<tr>
<td>point-color</td>
<td>string</td>
<td>No</td>
<td>Changes color of slider draggable button.</td>
</tr>
<tr>
<td>line-color</td>
<td>string</td>
<td>No</td>
<td>Changes color of slider (horizontal line).</td>
</tr>
<tr>
<td>value</td>
<td>number</td>
<td>No</td>
<td>Default value (for single selection slider mode).</td>
</tr>
<tr>
<td>value-start</td>
<td>number</td>
<td>No</td>
<td>Default start value (for range slider).</td>
</tr>
<tr>
<td>value-end</td>
<td>number</td>
<td>No</td>
<td>Default end value (for range slider).</td>
</tr>
</tbody>
</table>

**Discrete Slider**

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;TextBlock&quot;</td>
<td>Yes</td>
<td>Must be TextBlock.</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------</td>
<td>----------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>class</td>
<td>string</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>text</td>
<td>Slider array</td>
<td>Yes</td>
<td>List of values in the slider. Use 'sliderData' prefix within slider configuration block to connect it to metadata item from 'Items' panel.</td>
</tr>
<tr>
<td>items</td>
<td>array</td>
<td>No</td>
<td>Connection to metadata item.</td>
</tr>
<tr>
<td>range</td>
<td>string</td>
<td>No</td>
<td>Enable slider range mode.</td>
</tr>
<tr>
<td>point-color</td>
<td>string</td>
<td>No</td>
<td>Changes color of slider draggable button.</td>
</tr>
<tr>
<td>line-color</td>
<td>string</td>
<td>No</td>
<td>Changes color of slider (horizontal line).</td>
</tr>
<tr>
<td>value</td>
<td>number</td>
<td>No</td>
<td>Default value (for single selection slider mode).</td>
</tr>
<tr>
<td>value-start</td>
<td>number</td>
<td>No</td>
<td>Default start value (for range slider).</td>
</tr>
<tr>
<td>value-end</td>
<td>number</td>
<td>No</td>
<td>Default end value (for range slider).</td>
</tr>
</tbody>
</table>
Actions

Actions turn your BloX into interactive business applications by letting your users perform certain actions inside a widget. Actions are located after your body element, unless you add an Action.Set that allows you to place Actions within the body.

```json
{
    "type": "ActionSet",
    "actions": [
        {
            "type": "Action.OpenUrl",
            "action": "lightbox",
            "title": "Share",
            "url": "http://www.sisense.com"
        }
    ]
}
```

The snippets in this section can be applied to elements to add actions to them. For example, you can convert images into clickable elements with the Clickable Elements snippets.
Action Buttons

**Button**

You can create a button with the Action Section that when clicked, triggers an action.

```json
{
    "type": "Action.OpenUrl",
    "title": "View in SalesForce",
    "url": "https://sisense.salesforce.com/{panel:Id}"
}
```

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>title</td>
<td>string</td>
<td>Yes</td>
<td>Label for button or link that represents this action.</td>
</tr>
<tr>
<td>url</td>
<td>string</td>
<td>Yes</td>
<td>The URL to open.</td>
</tr>
</tbody>
</table>

**Inline Button**

When Action.OpenUrl is invoked, a URL you define is opened in your browser.

```json
{
    "type": "Action.OpenUrl",
    "title": "View in SalesForce",
    "url": "https://sisense.salesforce.com/{panel:Id}"
}
```

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
</table>
### Property | Type | Required | Description
--- | --- | --- | ---
title | string | Yes | Label for button or link that represents this action.
url | string | Yes | The URL to open.

**Clickable Element**

When Action.OpenUrl is invoked, a URL you define is opened in your browser.

```json
{
  "type": "Action.OpenUrl",
  "title": "View in SalesForce",
  "url": "https://sisense.salesforce.com/{panel:Id}"
}
```
Action Types

**Open URL**

When `Action.OpenUrl` is invoked, a URL you define is opened in your browser.

```json
{
    "type": "Action.OpenUrl",
    "title": "View in SalesForce",
    "url": "https://sisense.salesforce.com/{panel:Id}"
}
```

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>title</td>
<td>string</td>
<td>Yes</td>
<td>Label for button or link that represents this action.</td>
</tr>
<tr>
<td>url</td>
<td>string</td>
<td>Yes</td>
<td>The URL to open.</td>
</tr>
</tbody>
</table>

**Open URL in Popup**

When `Action.OpenUrl` is invoked, a URL you define is opened in a lightbox popup.

```json
{
    "type": "Action.OpenUrl",
    "action": "lightbox",
    "title": "View in SalesForce",
    "url": "https://sisense.salesforce.com/{panel:Id}"
}
```

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property</td>
<td>Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>action</td>
<td>&quot;lightbox&quot;</td>
<td>Yes</td>
<td>Must be lightbox. Determines that the URL should be opened in a pop-up.</td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Yes</td>
<td>Label for button or link that represents this action.</td>
</tr>
<tr>
<td>url</td>
<td>string</td>
<td>Yes</td>
<td>The URL to open.</td>
</tr>
</tbody>
</table>

**Submit**

Submit action takes input fields and merges them with optional data fields. It then generates an event to a client asking for data to be submitted. It is up to the client to determine how that data is processed. For example, the action below submits an ID to a webhook defined as the value of the “url” key.

```json
{
    "type": "Action.Submit",
    "action": "post",
    "title": "Submit",
    "data": {"id": "007007"}
    "url": "https://hooks.zapier.com/hooks/catch/171572/kay5ad/
```

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;Action.Submit&quot;</td>
<td>Yes</td>
<td>Must be Action.Submit.</td>
</tr>
<tr>
<td>action</td>
<td>&quot;post&quot;</td>
<td>Yes</td>
<td>Must be &quot;post&quot;.</td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Yes</td>
<td>Label for button or link that represents this action.</td>
</tr>
<tr>
<td>data</td>
<td>string,object</td>
<td>No</td>
<td>Initial data that the input fields will be combined with. This is essentially</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>----------------</td>
<td>----------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>'hidden' properties.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>url</td>
<td>string</td>
<td>Yes</td>
<td>The URL where the data is to be sent to.</td>
</tr>
</tbody>
</table>

**BloX Popup**

Action.ShowCard defines an inline BloX that is shown to the user when it is clicked.

```json
{
    "type": "Action.ShowCard",
    "title": "ShowCard",
    "card": {
        "showCarousel": true,
        "backgroundImage": "",
        "body": []
    }
}
```

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>title</td>
<td>string</td>
<td>Yes</td>
<td>The title to be displayed in the pop-up's window.</td>
</tr>
<tr>
<td>card</td>
<td>object</td>
<td>Yes</td>
<td>Defines the content to be displayed in the pop-up.</td>
</tr>
</tbody>
</table>

**Dashboard Filters (by Panel Name)**

You can add dashboard filters to your widget according to their panel name as it appears in the left panel when editing a widget.
For example, your users can apply a filter by its panel name, even if the filter wasn't previously defined in a dashboard, by clicking a button when you add this action.

```
{
  "type": "Filters",
  "title": "{{{category.text}}}",
  "data": {
    "filters": [
      {
        "panelName": "Category",
        "filterJaql": {
          "explicit": true,
          "members": [
            "{{{category.value}}}"
          ]
        }
      }
    ]
  }
}
```

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;Filter&quot;</td>
<td>Yes</td>
<td>Must be Filters.</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-----------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>s&quot;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Yes</td>
<td>Title of the action.</td>
</tr>
<tr>
<td>data</td>
<td>object</td>
<td>Yes</td>
<td>Contains the filters to be applied to the dashboard.</td>
</tr>
<tr>
<td>filters</td>
<td>object</td>
<td>Yes</td>
<td>Defines the filters to be applied to the dashboard.</td>
</tr>
<tr>
<td>panelName</td>
<td>string</td>
<td>Yes</td>
<td>Name of the filter as its displayed in the Panel.</td>
</tr>
<tr>
<td>filterJaql</td>
<td>object</td>
<td>No</td>
<td>Add custom advanced filters to your widget. In the filterJaql object, paste the code snippet created under the Advanced filters tab when creating a filter. For more information, see <a href="#">Advanced Filtering Criteria</a>.</td>
</tr>
</tbody>
</table>

**Dashboard Filters (by Filter Name)**

You can add dashboard filters to your widget according to their filter name. For example, your users can filter a dashboard by clicking a button when you add this action to that button.
```json
{
  "type": "Filters",
  "title": "Last Month",
  "data": {
    "filters": [
      {
        "filterName": "Months in Date",
        "filterJaql": {
          "last": {
            "count": 1,
            "offset": 1
          }
        },
        "dim": {
          "title": "Year",
          "table": "Lead Generation",
          "column": "Date",
          "datatype": "datetime",
          "level": "Years"
        }
      }
    ]
  }
}
```

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;Filters&quot;</td>
<td>Yes</td>
<td>Must be Filters.</td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Yes</td>
<td>Title of the filter to be displayed in the right dashboard panel.</td>
</tr>
<tr>
<td>data</td>
<td>object</td>
<td>Yes</td>
<td>Contains the filters to be applied to the dashboard.</td>
</tr>
<tr>
<td>filters</td>
<td>object</td>
<td>Yes</td>
<td>Defines the filters to be applied to the dashboard.</td>
</tr>
<tr>
<td>filterJaql</td>
<td>object</td>
<td>No</td>
<td>Add custom advanced filters to your widget. In</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>type</td>
<td>&quot;Filters&quot;</td>
<td>Yes</td>
<td>Must be Filters.</td>
</tr>
</tbody>
</table>

**Dashboard Filters (by Dimension)**

You can add dashboard filters to your widget according to dimension (dim).

For example, your users can filter a dashboard by clicking a button when you add this action to that button.

```json
{
    "type": "Filters",
    "title": "Filter (by Dimension)",
    "data": {
        "filters": [
            {
                "filterJaql": {
                    "last": {
                        "count": 1,
                        "offset": 0
                    }
                },
                "dim": {
                    "title": "Year",
                    "table": "Lead Generation",
                    "column": "Date",
                    "datatype": "datetime",
                    "level": "Years"
                }
            }
        ]
    }
}
```
<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>title</td>
<td>string</td>
<td>Yes</td>
<td>Title of the filter to be displayed in the right dashboard panel.</td>
</tr>
<tr>
<td>data</td>
<td>object</td>
<td>Yes</td>
<td>Contains the filters to be applied to the dashboard.</td>
</tr>
<tr>
<td>filters</td>
<td>object</td>
<td>Yes</td>
<td>Defines the filters to be applied to the dashboard.</td>
</tr>
<tr>
<td>filterJaql</td>
<td>object</td>
<td>No</td>
<td>Add custom advanced filters to your widget. In the filterJaql object, paste the code snippet created under the Advanced filters tab when creating a filter. For more information, see Advanced Filtering Criteria.</td>
</tr>
</tbody>
</table>

**Jump to Dashboard**

Adds a an action to an object in your widget that when clicked, opens a dashboard according to the Dashboard ID.

```json
{
    "type": "JTD",
    "title": "Jump To Dashboard",
    "data": {
        "dashboardId": ",",
        "panelsToInclude": [
            {
                "panel": "Category",
                "value": "{{category.text}}"
            }
        ],
        "args": {
            "displayDashboardsPane": false,
            "displayFilterPane": true
        }
    }
}
<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>&quot;JTD&quot;</td>
<td>Yes</td>
<td>Must be JTD.</td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Yes</td>
<td>Title of the action.</td>
</tr>
<tr>
<td>dashboardId</td>
<td>string</td>
<td>Yes</td>
<td>The ID of the dashboard found at the end of the URL of your dashboard. For example, <a href="http://localhost:8081/app/main#/dashboards/58184c2f295b2b342b000023">http://localhost:8081/app/main#/dashboards/58184c2f295b2b342b000023</a> - the set of characters after the last '/ ' is your Dashboard ID.</td>
</tr>
<tr>
<td>panelsToInclude</td>
<td>object</td>
<td>No</td>
<td>Enter the any panels you want to add to your dashboard when it is opened.</td>
</tr>
<tr>
<td>args</td>
<td>object</td>
<td>No</td>
<td>Add additional arguments and functionality to the action when a dashboard is opened. For a complete list of all the args you can apply, see the Technical Details for the Jump to Dashboard addon.</td>
</tr>
</tbody>
</table>

**Collected Product and Usage Data**

**User Information**

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sisense Owner ID</td>
<td>The email address of the owner of the Sisense account.</td>
</tr>
</tbody>
</table>

**Sisense-Related Information**
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sisense Version</td>
<td>The version of Sisense installed on the machine.</td>
</tr>
<tr>
<td>Sisense Monitoring</td>
<td>The version of Sisense’s remote support module (monitoring).</td>
</tr>
</tbody>
</table>

**Action Logs**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>The type of action performed by the user, for example: openURL, LightBox, Post API etc.</td>
</tr>
<tr>
<td>Area</td>
<td>The area of the application the action is related to, for example: Setup, Dashboard View, widget</td>
</tr>
<tr>
<td>Feature</td>
<td>The name of the feature being used, for example: Plugin Admin, Dev to Production, Pulse, Export to Excel</td>
</tr>
</tbody>
</table>

**Note:** The action log does not include licensee data, and does not attribute actions to identifiable users.

**Logs**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Exception Log</td>
<td>Product exception data such as when an exception occurred, which application, the message returned and the error type.</td>
</tr>
<tr>
<td>Event Log</td>
<td>Record when an error occurred, in which application, and the error message. The error messages within the event logs may contain information such as IP address, and ElastiCube, dashboard, or widget names.</td>
</tr>
</tbody>
</table>
Sisense BloX Tutorials

The following tutorials describe how to create an interactive dashboard that includes features such as embedded iFrames, action buttons for submitting data, and carousels for comparing products.
Introduction
Installing BloX
Creating Banners
Using Columns
Designing BloX Widgets
Adding Sparklines and Action Buttons
Embedding iFrames
Adding Carousals
Adding Forms
Embedding Social Media Feeds
Using Conditionals
Using Webhooks
Exporting and Importing BloX Templates
Saving and Deleting BloX Templates
Creating Actions
Embedding BloX Widgets
BloX 2.0 Course

This course includes an updated use case relevant for BloX 2 features.

At the end of the course, you will be familiar with new capabilities such as inline styling, custom actions, template sharing, integrating QR codes and much more.

Part 1 - Introduction and Warehouse Employee Solution
Part 2 - Supply Chain Manager Solution
Sisense Boto

Sisense Boto™ Terms of Use

Sisense Boto™ is an experimental product of Sisense Ltd. (collectively with its affiliates, “Sisense” “we”, “our” or “us”) created to give the members of the public (“you”) free access to powerful Artificial Intelligence analytics on big datasets. Sisense does not offer any support for Sisense Boto. Any questions or issues may be directed to discussion forums on the Sisense website. Sisense does not guarantee that Sisense Boto will remain available in the future and may modify or remove it at any time.

By using Sisense Boto, you agree to comply with these Terms of Use. Sisense Boto and the Documentation pertaining to it are provided “as-is” without warranty of any kind. Sisense and its licensors and suppliers disclaim all express, implied or statutory warranties, including without limitation any implied warranties of title, non-infringement of third party rights, merchantability, or fitness for a particular purpose to the extent permitted under applicable law. You alone bear the entire risk arising out of the use of Sisense Boto. We specifically disclaim any liability for your use of or reliance upon any results or outputs of Sisense Boto. We do not guarantee Sisense Boto will produce results or output for all data inputs. **In no event shall Sisense or its directors, officers, employees, shareholders, subsidiaries, affiliates, agents, or licensors be liable for any damages arising out of or related to the use, inability to use, performance, or nonperformance of Sisense Boto except to the extent such liability may not be disclaimer under applicable law.**

Sisense reserves the right to limit your access or block you from using Sisense Boto at its discretion if you violate these Terms of Use or abuse the service.
Accessing Sisense Boto

Sisense Boto may be accessed through: (i) the web interface or (ii) through bots running on platforms such as Facebook Messenger, Skype, and Slack. Sisense Boto may only be used by a natural human using the interfaces or bots provided by Sisense. Any attempt to use a script, bot, other interface, or any other automated or manual means to circumvent the interfaces or bots or repeatedly accessing Sisense Boto could damage or overload the system and is strictly forbidden. You may not frame or embed the Sisense Boto interfaces or bots in any other website or interface nor may you modify Sisense Boto in any way. Sisense may terminate access to Sisense Boto by anyone who jeopardizes the availability or integrity of our system.
Accessing Sisense Boto through bots

If you access Sisense Boto through bots on another platform, your use is subject to the terms of use of those platforms in addition to these Terms of Use. Please refer to and comply with the terms of use on such websites.

Sisense Boto bots may be used to connect to a Sisense Server software for which you have an active license. Alternatively, you can upload your data as a .csv file through the Sisense Boto web interface or bot.

If you use the Sisense Boto bots with your licensed Sisense Server software, your use of the Sisense Server software is governed by the license agreement between you and Sisense (the “License Agreement”). However, none of the representations, warranties or indemnities provided by Sisense in the License Agreement shall apply to Sisense Boto or the bots. For purposes of the limitation of liability provisions of the License Agreement, Sisense Boto is not provided under the terms of a Sales Order and no amount has been paid or is payable to Sisense in connection with the Sisense Boto or the bots.
Uploading your data to Sisense Boto

You may upload your data to Sisense Boto and copy and use the outputs of the Sisense Boto for noncommercial purposes only, subject to these Terms of Use. Uploaded data is limited to a single .csv file of 10Mb or smaller at a time.
Ownership of your input data

By uploading data or inputting data into Sisense Boto, you warrant and represent that you have all the rights and authority to use the data with Sisense Boto. You further grant Sisense a worldwide irrevocable license to store, copy, analyze, and generate the results from the data that we return to you. We may also use your inputs in accordance with the Privacy Policy below.
Children

You must be 13 or older (or whatever age your local jurisdiction requires for providing your data without parental consent, whichever is older) to use Sisense Boto. Sisense will block you from using Sisense Boto if we discover you are under age.
Use Restrictions

Sisense Boto is intended to be used solely for lawful purposes. Sisense does not screen or review the inputs or the data you use with Sisense Boto. You are solely responsible for ensuring that your use of Sisense Boto complies with all applicable laws, regulations, and rules that apply to you or your data and does not violate any third party rights.

You may not use Sisense Boto to:

- attempt to reverse engineer Sisense Boto or use data inputs or queries to discover the algorithms or technologies underlying Sisense Boto.
- attempt to circumvent any security protections, probe for any vulnerabilities, penetrate, or otherwise violate the security of the system or tamper, interfere, or disrupt networks or systems.
- defame, abuse, harass, or otherwise violate the rights of privacy and publicity of others;
- share inappropriate content, advertising, spam, spyware or malware;
- infringe upon the rights of others.

You may not input into Sisense Boto personally identifiable data of others, protected health information (PHI), payment card information, personal finance information, export-controlled data, or any other data that is subject to any laws, regulations, industry standards, or contractual obligations that restrict the use or onward transfer of the data or require the recipient to protect the data in any way that is inconsistent with use of Sisense Boto under these Terms of Use.
Export Restrictions

You represent and warrant that you are not located in and will not export or re-export, either directly or indirectly, Sisense Boto to any country or entity under United States restrictions or to any country or entity subject to trade sanctions.
Sisense Boto Privacy Policy

We take reasonable measures to secure Sisense Boto and any communications with you, but cannot guarantee complete security of your data when you use Sisense Boto. We don’t encrypt the data stored on our servers.

**Sisense Boto may not be used to analyze or process personal information of others. You are exclusively responsible for ensuring that your use of Sisense Boto does not violate the privacy rights of others.**

Sisense is operated from servers in the United States. By using Sisense Boto, you agree to transmit and process your data in the United States and that you have the required permission to transmit such data to United States locations. When you use the Sisense Boto interface on the Sisense website, your use is also subject to the Sisense Privacy Policy and website Terms of Use.
What data we collect

When you interact with Sisense Boto through bots, Sisense collects your chat address (the name or other identifier you use on the platform) only as long as you keep Sisense Boto in your contact list on the platform you use. If you delete Sisense Boto from your contact list, the chat record is deleted as well and Sisense does not retain a copy of the chat.

When you connect your own Sisense Server software to the Sisense Boto bots, the data on your Sisense Server isn’t disclosed to Sisense. However, outputs of the data are retained with the chat record which is visible to Sisense as long as you keep Sisense Boto in your contact list on the platform you use. You can delete your data and the chat record by removing the Sisense Boto bot from your contact list.

When you upload your data to Sisense Boto (as a .csv file or otherwise) (“uploaded data”), Sisense does not keep such data longer than 48 hours. The outputs and analysis results of the uploaded data are retained by Sisense as part of the chat dialogue only while the session remains active. The chat dialogue is deleted when you close the browser window or tab you used to access Sisense Boto.
How we may use your data and inputs

We will not disclose, sell, or provide your information to third parties, except in the following circumstances:

To operate Sisense Boto and provide you the results of the analysis.

To test and improve the performance of Sisense Boto, study usage patterns, and develop other Sisense products. We may compile information and generate reports related to our users’ use of Sisense Boto, and may share aggregated statistical information with third parties.

If the information is required by law, we will share information in response to legal process, court orders, subpoenas, or to establish or exercise our legal rights or defend against legal claims;

Information collected from users of Sisense Boto could be transferred as part of or in connection with a corporate merger, consolidation, restructuring, the sale of substantially all of Sisense’s stock and/or assets relating to Sisense Boto or Sisense’s business or other corporate change, including during the course of any due diligence process.

If you use Sisense Boto, we may contact you to interest you in learning more about our products and services, send you surveys, or seek user feedback.
General

Sisense may revise these Sisense Boto Terms of Use at any time by updating this posting. You should visit this page from time to time to review the current Terms of Use.

Any claim relating to Sisense Boto will be governed by and interpreted in accordance with the laws of the State of Israel, without reference to its conflict-of-laws principles and you hereby consent to exclusive jurisdiction and venue in the civil courts sitting in Tel Aviv, Israel. You agree to waive all defenses of lack of personal jurisdiction and forum non-conveniens. If any term or provision of these Sisense Boto Terms of Use is for any reason held to be invalid, such invalidity shall not affect any other term or provision.

Last updated: June 11, 2018
Introduction

While Sisense provides you with a centralized location for consuming data in the Sisense Web Application, sometimes you need quick access to your data without having to jump across applications to ask data questions. Sisense Boto analyzes your data and returns natural language insights within 3rd party applications such as Skype and Slack. In addition, you can upload CSV files and Sisense Boto will provide you with insights directly in your chat.

**Note:** To participate in the Sisense Boto Beta Program, your Sisense server must be accessible over the Internet without a VPN.

Sisense Boto supports the following 3rd party applications:

- Skype
- Slack
- Facebook

Click the links above to add Sisense Boto to your applications.
How does it Work?

Sisense Boto analyzes your data and returns insights to you in natural language in 3rd party applications.

After setting up Sisense Boto, all you need to do is chat with it and tell Sisense Boto which dashboards you would like to get insights for, or upload a CSV file for Sisense Boto to analyze.

Throughout your chat, Sisense Boto will provide instructions with how interact with it. The one thing to keep in mind is that as you dive deeper into your data, each dive acts as a level so, to go to previous insights, you can type **back** until you get to the desired level.

The easiest way to learn about Sisense Boto is to see it in action:
Setting Up Sisense Boto

To interact with Sisense Boto, you must add Sisense Boto to your application.

Skype
Slack
Facebook

Click the links above to add Sisense Boto to your applications.

After adding Sisense Boto to your application, greet Sisense Boto by typing, ‘Hi’.

The first time you message Sisense Boto, you will be asked to authenticate your account.

To authenticate, click the link and enter the following details:

Sisense server address: The address of your Sisense account including the protocol (“http://”) and your server’s port, for example http://192.168.1.0:8081.
Email address: Your Sisense account email address.
Password: The password of your Sisense account.

After you have entered your details, click Authenticate and return to your application. Greet Sisense Boto again by typing ‘Hi’. Sisense Boto will get your environment ready, which takes just a few seconds. Once it’s ready, you can now begin to consume data from with your application.

Sisense Boto will offer some advice to get started. You can select Skip to pass on the tutorial.
Interacting with Sisense Boto

Sisense Boto provides an interface for returning insights, so no need to ask questions. Just select your dashboard from the carousel and begin diving in.

Hold tight while I connect to your data.
You have 662 widgets across 44 dashboards.
Choose a dashboard to explore:

To get an idea of just how easy it is to work with Sisense Boto, check out some of our tutorials below:

Share Insights Instantly with Anyone

Unveil Insights from Your CSV Files

Get Deeper Insights

Once you have started exploring a dashboard, Sisense Boto will display the following buttons:

Show my Dashboards: Display a list of your dashboards in a carousel in your 3rd party application
Get Whisper: Return a natural language explanation of your widget.
Go Deeper: Return more explanations of your widget.
Thank you Boto: Return to the main menu.
Remove from List: Remove a dashboard from current dashboard carousel. This does not delete the dashboard, just removes it from the carousel.
Add to List: Add a removed dashboard to dashboard carousel.
Open in Web: Open your Sisense application filtered by the selected widget.
Filtered by: Show the list of filters applied to the selected dashboard.

Sisense Advanced Commands
As you begin exploring your dashboards and their widgets, you will see a button called Advanced Options.
Below are a list of advanced commands you can use and their meanings:

Show my dashboards: Display a list of your dashboards in a carousel in your 3rd party application.
Get [WIDGET NAME]: Return the last widget you explored to the chat and provides some natural language details of the data.
Get [DASHBOARD NAME] dashboard: Return a list of widgets your dashboard and its data.
Analyze [WIDGET NAME]: Return a natural language explanation of your widget.

Share [WIDGET NAME]: Share the widget with another user. After Sisense Boto returns your insights, select the Sisense user's name or type Other to share a widget with a non-Sisense user, for example, an email address. Sisense Boto will then forward the widget and its insights to the external user.

CleanUP: Scans your activity for the past 30 days and suggests removing any unused dashboards from the dashboard carousel. This does not delete the dashboard, just prevents it from being displayed. You can add the dashboard again by using the Add to List button.

Refresh: Clears your metadata. This is useful if Sisense Boto seems to be running slow or you are experiencing a bug.

Back: Returns you to the previous step in your data exploration. For example, if you explore a dashboard, then select Go Deeper, selecting back would return you to the dashboard.

Goodbye: Ends your Sisense Boto session. This clears all the metadata. To restart a new session, say 'Hi'.
Analyzing CSV Files

In addition to analyzing your dashboards, you can upload CSV files (up to 10mb in size) and Sisense Boto will analyze the data and return natural language insights.

After uploading the file, you select the column in the CSV file you want to Sisense Boto to analyze. Uploaded files are deleted by Sisense automatically several hours after the upload takes place.

To analyze your data:

Drag your CSV file into your chat with Sisense Boto. Sisense Boto will request that you select a column to analyze.

Select a category. Sisense Boto will begin to analyze it. This may take several minutes depending on the size of your file and how many insights are extracted. The maximum amount of time is 5 minutes.

When Sisense Boto is done, your insights will be displayed in chat. For more insights, click Next and Sisense Boto will display any more insights if any are available.
Sharing Insights

You have several options for sharing insights, by typing ‘Share’ or click the **Share** button. If you use the advance command, Share [Widget Name], you will be prompted to select the Sisense user. If you want to share some insights with a non-Sisense user, just select Other and you will be able to enter that user’s email address. Sisense Boto will then forward your insights via email to the user.
Group Chat

You can begin a group chat with a colleague and then invite Sisense Boto to join the conversation by typing, '@sisense_boto'. Remember, when Sisense Boto returns insights, anyone in the conversation will be able to see those insights, however, only users who you shared the dashboard with can click the **Explore** button to access the dashboard from within the chat.

In your conversation, interact with Sisense Boto as you would within a private session.
Sisense Enabled Bulb

Beta License Agreement
This product is still in beta and may undergo further changes. The following are instructions for using various equipment with Sisense’s BI Everywhere software functionality (the equipment and the BI Everywhere software is collectively referred to herein as the “Product”). This notice is to remind users that the Product is a beta release offering and is not warranted to be at the level of performance of a commercially available product offering. In addition, the Product may be substantially modified prior to first commercial release, or at Sisense’s option may not be released commercially in the future. As such, by using the equipment along with the Sisense Software, each user agrees (i) to follow all instructions provided by Sisense with respect to the installation and use of the equipment with the Sisense Software, including the below, and (ii) that such use continues to be for testing and evaluation purposes only.

The Product and Documentation are provided “AS IS” without warranty of any kind, and Sisense and its licensors and suppliers disclaim all warranties, express, implied or statutory, including without limitation any implied warranties of title, non-infringement of third party rights, merchantability, or fitness for a particular purpose. The entire risk arising out of the use or performance of the Product remains with the user.

For the sake of clarity, the provisions in the license agreement between the licensee of the Sisense Software and Sisense (the “License Agreement”), providing any representations, warranties or indemnities shall not apply to the Product. For purposes of the limitation of liability provisions of the License Agreement, no amount has been paid or is payable to Sisense in connection with
the Product. However, the license provisions continue to apply to the Sisense Software and the confidentiality provisions of the License Agreement shall continue to apply to those portions of the information shared with you by Sisense that are Confidential Information as defined in the License Agreement or any confidentiality agreement in effect between the licensee and Sisense.
Introduction

Traditional dashboards provide comprehensive access to vast amounts of data, however, sometimes you need quick, actionable insights. The Sisense-Enabled Bulb provides an immediate visual representation of a KPI. Sisense recommends choosing a KPI that is meaningful to your business and you expect might change frequently. KPIs that are relatively static are less meaningful to monitor with the Sisense-Enabled Bulb.

To see how Sisense’s Marketing department has implemented the Sisense-Enabled Bulb, click here.

If you have not purchased a bulb, for US customers you can purchase one from the LIFX US site and for European customers, you can visit the LIFX Europe site.

This topic provides the following information:
- **How does it Work**: Describes how the Sisense-Enabled Bulb works.
- **Connecting the Sisense-Enabled Bulb to Your Wi-Fi**: Describes how to connect your Sisense-Enabled Bulb to your Wi-Fi.
- **Retrieving a Sisense-Enabled Bulb Token**: Describes how to retrieve a LIFX token for your bulb.
- **Connecting Your Bulb to Sisense**: Describe how to connect Sisense to your Sisense-Enabled Bulb.
- **Troubleshooting the Sisense-Enabled Bulb**: Describes how to troubleshoot potential issues with the Sisense-Enabled Bulb.
How does it Work?

The Sisense-Enabled Bulb is connected to your Wi-Fi connection and dynamically updates when a data alert is triggered. You can create data alerts for a wide array of conditions. To learn more, see Data Alerts.

Sisense can be configured to send a webhook to Zapier each time a data alert is triggered. When the webhook arrives, Zapier sends a request, also known as a ‘Zap’, to LIFX API. The Zap contains the ID of your bulb and defines how your bulb should respond to an alert. For example, how many times the bulb should blink or what color to display.

To set up an LIFX bulb, you need to connect your LIFX bulb to your Wifi, retrieve an ID for your bulb, and finally, create a Zap that is sent to the LIFX API when your data alert is triggered.
Connecting the Sisense-Enabled Bulb to Your WiFi

To connect to the Sisense-Enabled Bulb to your wifi, you must download the LIFX app. The LIFX provides an interface in Apple or Android-based mobile devices for defining your bulb’s color and connections settings.

**To connect to the Sisense-Enabled Bulb:**
- Before plugging the lamp into the power outlet, screw the bulb into the lamp.
- Plug the lamp into the power outlet and turn it on.
  - **Note:** You can reset the lamp by turning the lamp on and off 5 times. When you reset the lamp you must complete this procedure again.
- Download and install LIFX. For Apple-based mobile devices, click here, and for Android, click here.
- Open the LIFX app.
- In LIFX, tap **Register** and create an account.

Enter an email address and password to be used for your LIFX account and tap **Sign In**.
In the top-right corner, tap + and then **Connect light**. LIFX locates the bulb.
Tap the name of the bulb located by LIFX.

Connect the bulb to your Wi-Fi.
LIFX will connect to your Sisense-Enabled Bulb, and display a list of all your Sisense-Enabled Bulbs. The current color of your bulb is displayed.
Retrieving a Sisense-Enabled Bulb Token

To allow Sisense to communicate with your LIFX bulb, you must provide Sisense with access to your LIFX bulb. After you provide access, LIFX returns a token that you include in your Zap that enables your bulb to authenticate requests from Sisense when updating your bulb’s color.

**To retrieve a token:**

In your web browser, open the Sisense LIFX authorization site located [here](#).

Enter your LIFX account email address and password in the relevant fields. Click **Sign in**. The Authorization page is displayed.

In the Authorization page, click **Authorize**. Your Sisense BulbiToken is displayed. Save this token for later.
Connecting Your Bulb to Sisense

Now that you have set up your bulb and retrieved a token, you can create a Zap that defines how your bulb responds to a data alert.

When connecting your bulb to Sisense, you need Sisense V6.5 or later and a Zapier account. If you do not have a Zapier account, you can make one for free here.

**Note:** Repeat this procedure for every color you want your bulb to support. For example, if you use colors green and red to represent alerts, repeat this procedure twice, once for each color.

**To configure Zapier to work with the Sisense-Enabled Bulb:**

1. Log in to your Zapier account.
2. Open this [Zap template](#) prepared by Sisense.
3. Click **Create this Zap**.
4. Click **Continue**. The Webhook Setup options are displayed.
5. Click **Continue**. The Test Webhooks page is displayed.
6. In the Test Webhooks page, a webhook generated by Zapier is displayed. Click **Copy to clipboard** to copy this URL.
7. Test the webhook by pasting it into your browser and add ?data= to the end of the URL to verify that the webhook works. For example: https://hooks.zapier.com/hooks/catch/1779449/5f9jj7/?data=
8. If the webhook is working as expected, you should see “status”: “success” and some additional IDs in your browser. Keep the URL saved in your clipboard as you need to paste it again later.
9. In the Test Webhooks page, click **I did this** to confirm that you have tested the webhook. A message is displayed confirming that the test was successful.
10. Open Sisense.
11. Create a new data alert or edit an existing alert. For more information, see [Creating Data Alerts](#).
12. In the Notifications Options of your alert, select **Zapier** and paste the webhook URL copied from Step 6 into the **URL** field (without the ?data=)
In the bottom left corner of the window, click **Additional Options** and select **Notify only once after a condition** is met to limit how many times your bulb refreshes itself after an alert is triggered.

Return to Zapier and click **Continue**.

Create a new webhook by clicking **Continue**. The **Set Up Webhooks** page is displayed.

In the **URL** field, enter your LIFX URL. The URL has the following structure: `https://api.lifx.com/v1/lights/BULB_ID/effects/pulse`. Replace **BULB_ID** with your LIFX bulb ID. **The ID should be written in**
lowercase letters.
Select **Show advanced options**. Additional fields are displayed.

- **Payload Type (optional)**
  Pay special attention to the proper mapping of the data below.
  - `form`

- **Data (optional)**
  If you leave this empty, it will default to including the raw data from the previous step. Key, value pairs sent as data. Do not place raw JSON or form encoded values here!

- **Wrap Request In Array (optional) ● ○**
  Wraps the request in an array if that is something the service requires.
  - `no`

- **File (optional)**
  A file object to be attached to the request. The request will automatically be converted to `multipart/form-data`, with the above payload type defining the sub content-type for the data being passed.

- **Unflatten (optional) ● ○**
  By default we convert fields with underscores into nested dictionaries. Select no here to turn this behavior off.
  - `no`

- **Basic Auth (optional)**
  A pipe (`|`) separated username, password combo for standard HTTP authentication.

- **Headers (optional)**
  Key, value pairs to be added as headers in all requests.

From the **Payload Type** list, select **Form**.
In the Data section you need to add several key-value pairs that determine the behavior of your bulb when an alert is triggered. The values of these keys are passed as parameters through your webhook to the bulb. Click + to add five fields. Entering the following details into the fields:

Key 1: Enter **from_color** in the Key field and in the **Value** field, enter the default color for your bulb.
Key 2: Enter **color** in the **Key** field and in the **Value** field, enter the color you want the bulb to be when an alert is triggered.
Key 3: Enter **power_on** in the **Key** field, and in the **Value** field, enter **true**. When true, Sisense turns the bulb on even if it is not already on.
Key 4: Enter **cycles** in the **Key** field and in the **Value** field enter the number of cycles. Cycles are the number of times a light pulsates when an alert is triggered.
Key 5: Enter **period** in the **Key** field and in the **Value** field enter the number of seconds for each period. Periods are the time in seconds for a single cycle to complete.
Key 6: Enter persist in the **Key** field and in the **Value** field, enter **true**. When true, the last color displayed in the bulb is the color displayed until your next alert.

For example:

```
<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>from_color</td>
<td>red</td>
</tr>
<tr>
<td>color</td>
<td></td>
</tr>
<tr>
<td>power_on</td>
<td>true</td>
</tr>
<tr>
<td>cycles</td>
<td>5</td>
</tr>
<tr>
<td>period</td>
<td></td>
</tr>
<tr>
<td>persist</td>
<td>true</td>
</tr>
</tbody>
</table>
```

In the **Unflatten** field, enter **yes** to convert fields with underscores into nested dictionaries.

In the **Headers** section, enter the following key-value pair:

- In the **Key** field, enter **Authorization**.
  - In the **Value** field, enter **Bearer** and your LIFX token with a space in between, for example, **Bearer 23Sk5fj23934fLMNs**. Click here to retrieve your LIFX token.

Click **Continue**. The Test Webhook page is displayed.

Click **Create & Continue**. The next time the alert is triggered, Sisense fires a webhook that activates your Sisense-Enabled Bulb.
Troubleshooting the Sisense-Enabled Bulb

Bulbs are designed to communicate on channels 1 through 11 on the 2.4GHz (802.11 b,g,n) spectrum. If your router is in auto-channel mode, disable it and ensure you are somewhere between channels 1 through 11. We suggest channels 1, 6, or 11, as these are the only channels that are non-overlapping.

You can reset the bulb by turning it on and off 5 times in a row. When you reset the lamp you must reconnect the lamp as described [here](#).
**Sisense Alexa Skill**

**Beta License Agreement**

The following are instructions for using various equipment with Sisense’s BI Everywhere software functionality (the equipment and the BI Everywhere software is collectively referred to herein as the “Product”). This notice is to remind users that the Product is a beta release offering and is not warranted to be at the level of performance of a commercially available product offering. In addition, the Product may be substantially modified prior to first commercial release, or at Sisense’s option may not be released commercially in the future. As such, by using the equipment along with the Sisense Software, each user agrees (i) to follow all instructions provided by Sisense with respect to the installation and use of the equipment with the Sisense Software, including the below, and (ii) that such use continues to be for testing and evaluation purposes only.

The Product and Documentation are provided “AS IS” without warranty of any kind, and Sisense and its licensors and suppliers disclaim all warranties, express, implied or statutory, including without limitation any implied warranties of title, non-infringement of third party rights, merchantability, or fitness for a particular purpose. The entire risk arising out of the use or performance of the Product remains with the user.

For the sake of clarity, the provisions in the license agreement between the licensee of the Sisense Software and Sisense (the “License Agreement”), providing any representations, warranties or indemnities shall not apply to the Product. For purposes of the limitation of liability provisions of the License Agreement, no amount has been paid or is payable to Sisense in connection with the Product. However, the license provisions continue to apply to the Sisense
Software and the confidentiality provisions of the License Agreement shall continue to apply to those portions of the information shared with you by Sisense that are Confidential Information as defined in the License Agreement or any confidentiality agreement in effect between the licensee and Sisense.
Introduction

Sisense provides you with the means to interact with your data through traditional point-and-click interfaces as well as through voice commands. With a single voice command, you can retrieve and hear your numeric KPIs. The Sisense Alexa Skill searches your widgets and dashboards to provide you with the best result according to your command.

Sisense enables Amazon’s Echo, which is a voice-powered, wireless-enabled speaker that interfaces with Alexa, to retrieve your data in Sisense. Alexa is a cloud-based voice service that powers the Echo. Alexa’s default capabilities such as playing music can be extended through skills. Sisense has developed a Sisense skill that extends the Echo’s functionality and returns numeric KPIs through voice commands.

This page provides the following information:
- **How does it Work**: Describes how the Sisense Alexa Skill works.
- **Setting up the Sisense Alexa Skill**: Describes how to set up the Sisense Alexa Skill.
- **Querying the Sisense Alexa Skill**: Describes how to query the Sisense Alexa Skill.
- **Troubleshooting**: Describes how to troubleshoot potential issues with the Sisense Alexa Skill.
How does it Work?

After launching Sisense via voice command, you can retrieve numeric data quickly by asking Alexa about your KPIs similar to issuing traditional queries through Sisense. Your Sisense Alexa Skill listens for the command and forwards it to Alexa.

Alexa handles your requests and routes the request to the Sisense ElastiCube where your information is retrieved. Alexa then translates this data into an audio response played back by your Sisense Alexa Skill.
Setting Up Sisense Alexa Skill

Sisense has developed a skill that enables you to retrieve your data via voice commands. This skill takes requests from the Alexa service and leverages the Sisense APIs to retrieve your data.

To enable your Sisense Alexa Skill to accept your voice commands and pass them through the Alexa service and the Sisense APIs, you will need to enable the Sisense skill.

Before setting up your Sisense Alexa Skill, you may want to create a new user with a single dashboard associated with their account so you can easily test your Echo without returning unexpected results from various dashboards.

**Note:** To participate in the Sisense Alexa Skill Beta Program, your Sisense server must be accessible over the Internet without a VPN.

**To set up the Sisense Alexa Skill:**

1. Download the Alexa app from the app store and activate it using your email address.
   - Android
   - iTunes

2. Turn on your Echo and connect it to your WIFI.

3. Open the Sisense Skill page and click **Enable**.

After you have enabled the skill, the **Link Account** button is displayed. Click **Link Account** to display the Sisense Insights page in a new window. If you have a pop-up blocker enabled, you may need to allow the website...
insights.sisense.com to be opened in a new window.

In the Sisense Insights page you link the Sisense skill to your Sisense account. Enter the following details:

**Sisense server address**: The address of your Sisense account including “http://.”

**Email address**: Your Sisense account email address.

**Password**: The password of your Sisense account.

Click **Authenticate**. Your account is now authenticated and you can begin to work with the Sisense Alexa Skill.
Querying the Sisense Alexa Skill

The table below provides a list of supported commands, the objects they refer to, and the results you can expect when you give the command:

<table>
<thead>
<tr>
<th>Command</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refresh metadata</td>
<td>Refreshes the dashboard list.</td>
</tr>
<tr>
<td>List my dashboards</td>
<td>Lists your top five most frequently used dashboards.</td>
</tr>
<tr>
<td>List all widgets from &lt;dashboard title&gt;</td>
<td>Returns all widgets' names from that specific dashboard.</td>
</tr>
<tr>
<td>List all widgets from &lt;dashboard title&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;dashboard title&gt; dashboard</td>
<td></td>
</tr>
<tr>
<td>Analyze &lt;dashboard title&gt; dashboard</td>
<td>Returns a summary of all your Indicator widgets.</td>
</tr>
<tr>
<td>Analyze &lt;widget title&gt; from &lt;dashboard title&gt; title</td>
<td>Returns an NLP analysis of that specific widget.</td>
</tr>
<tr>
<td>Help</td>
<td>Returns all available commands.</td>
</tr>
<tr>
<td>Logout</td>
<td>Disables the skill. If you want to access this skill again, you must enable the skill with Amazon.</td>
</tr>
</tbody>
</table>
Security

There are two potential security issues that arise when working with the Sisense Alexa Skill. The first issue is what happens to data when you ask a question. When you ask questions and receive answers, your data is sent to and returned from the Alexa service. This data transfer takes place across secured SSL ensuring that no one can access your data.

The second issue is securing the Sisense Alexa Skill itself, so no one can access it and retrieve your data. When you say to Alexa, “Deactivate Account”, this disconnects your Sisense Alexa Skill from your Sisense account. Alexa will ask you to confirm that you want to deactivate your account. If you respond “Yes”, the account is deactivated. No one will be able to access your data through the Sisense Alexa Skill until the next time you manually reactivate your account.
Troubleshooting

For Amazon Tap and Echo devices, you can use Voice Training. Voice Training helps Alexa understand your speech patterns. During a Voice Training session, the Alexa app shows 25 different phrases, which you say to your device. To learn more, go to Voice Training.

If your selected language doesn’t match the language settings for your Amazon account, Alexa may not always understand what you say and may have trouble pronouncing things. For example, if you are located in Canada and are having language problems, change the language setting to US English.

If you have any issues authenticating your Echo, you can try to enable the device from the desktop through the following site.
Viewing Dashboards on Mobile Devices

Sisense Mobile connects you to your dashboard anytime, anywhere on your Android or iPhone mobile devices. Sisense Mobile decreases time to insight with an intuitive UI, search bar for locating dashboards, and responsive visualizations; You can explore your dashboards with simple gestures optimized to let you dive deep into your data or apply filters to view the data you want. Check out a demo video below to see how you can tap, zoom, and swipe to interact with your data.

Sisense now offers a premium white-labeled version of the Sisense Mobile app that can be customized according to your branding. For more information, see Rebranding Sisense Mobile.
Getting Started

Supported Phones and Operating Systems
  iPhone – Requires iOS 9 or later, Supported iPhone 5 device or higher.
  Android – Requires Android versions 4.4 or later.
Get Sisense Mobile

To download the Sisense Mobile App for your device, click the relevant link below according to your version of Sisense.

**Note:** Sisense Mobile is supported by Sisense V6.5 and later.
Logging in to Sisense Mobile

After you have downloaded Sisense Mobile, open the app to launch the login screen and enter the server address and your Sisense user credentials. The server address should include the server IP or domain name (alias). If you are not using the default port (80), you will need to add the port as well, for example: http://xx.xxx.xx.xx:8081. When working with SSL security protocol, use the ‘https’ prefix.

You can sign out by opening the navigation menu and tapping Sign Out.

If you have problems connecting, check the following:
- If your server is not on port 8081, check that the port has been added correctly, for example, 192.168.1.1:8080
- Try to access the server from your phone's web browser
- Try to access the server from other computers connected to the same WIFI connection as your phone

**Navigating Sisense Mobile**
When you first log in to Sisense Mobile, a list of your folders, dashboards, and recently opened dashboards is displayed.

Tap a dashboard to open the dashboard, or if your dashboard is located in a folder, tap the folder then tap the relevant dashboard to open it.

You can also locate a dashboard quickly by tapping the search icon and typing the name of the dashboard. As you begin to type, any dashboards that match the
characters you entered are displayed.

| Sales Current | Sales |
| Sales Prospects | Sales |
| Open Issues | Support |
| Response Time | Support |
Once the dashboard is displayed, you can tap the dashboard’s title to display more details.
By rotating your device horizontally, you can view your dashboard in landscape view.

**Exploring Data**

In Sisense Mobile, you can filter and drill down into your data, much like you do from your PC.
Filtering Widgets

To filter a dashboard, you can tap the filter icon. This displays the list of available filters.

<table>
<thead>
<tr>
<th>Country</th>
<th>Dashboards Filters</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Include</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Years</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2013</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sales Revenue Range</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>All items Greater than 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
You can disable your filters by swiping left on the filter to display a toggle button. Tap it to disable a filter.
A gray background behind the filter indicates the filter has been disabled.

To dive deeper into your filters, tap the relevant filter to open all of its available options. Then, tap any of the filter options to toggle the filters to your data.
Drilling into your Data

You can drill down in most widgets to get an in-depth view of a selected value. For example, tap on a slice in a pie chart to see the absolute value in addition to the percentage.

In the following example, the left chart shows the lead breakdown by campaign source, and on the right side, the chart shows a breakdown of visits from various social media sources.

Any time you select a value, a tooltip is displayed at the bottom of your screen, which provides a description of the selected value and the drill down
icon. In this example, you see the total lead and its percentage.

Note: You cannot drill down while in landscape mode.
To return to a higher level, click on a breadcrumb. To drill all the way up, click on the X icon.
Viewing More Details

Sisense Mobile supports a variety of gestures that let you dive deeper in your data and see more details.

<table>
<thead>
<tr>
<th>Highlight a Value</th>
<th>Swipe across your Visualizations</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Highlight a Value" /></td>
<td><img src="image2.png" alt="Swipe across your Visualizations" /></td>
</tr>
</tbody>
</table>

- **Pinch in to Zoom**
- **Pinch out to Expand**
**Sharing Widgets**

You can share widgets as PNG images from Sisense Mobile across various apps supported on your device.

In your dashboard, tap a widget to open Widget Mode, then tap the menu icon and tap Share. Your device will display a list of apps that you can use to send the
**Sisense Pulse**

Sisense Pulse is a centralized location where you can stay on top of your most important KPIs across multiple dashboards or manage your data and build alerts. By adding important KPIs from your dashboards to Sisense Pulse, you can get a comprehensive picture of your data from a single location.
Sisense Pulse contains tiles that display information from your dashboards and the status of your ElastiCube builds. Within Sisense Pulse, there are two types of tiles, data tiles that display your widgets’ current status and recent history and build tiles that display the latest status of your ElastiCube builds.

In Sisense Mobile you can manage your alerts from Sisense Pulse or create data alerts based on widgets in your dashboards.
**Note:** Build alerts can only be created from the Sisense Web Application.
You can access Sisense Pulse from Sisense Mobile by opening the navigation menu and tapping Pulse.
Creating Data Alerts

Data alerts are triggered by conditions you define for your widgets. For example, if you have an Indicator widget that represents your company’s revenue, you can define an alert that notifies you when this revenue reaches or falls below a certain threshold.

For each type of widget you can add to Pulse, you can monitor a single value. For example, for Column charts, you can monitor a single section (value) of a column. To monitor multiple values, you can add each value you want to monitor to Pulse separately.

When you create an alert for a specific value, Sisense applies any filters that were already applied, for example, if you create a data alert for the total revenue of a given age range, the pulse alert created will be called Total Revenue and will be filtered by the value of the age range you selected.

To create a data alert, from your dashboard, tap the relevant widget to open it in Widget Mode. In Widget Mode, tap the value you want to monitor, then tap the menu icon and tap Add to Pulse. This opens the Alert Conditions menu where you select the type of data alert you want to create. For example, if you create a
threshold data alert, the following settings are displayed.

<table>
<thead>
<tr>
<th>Alert Name</th>
<th>Total Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alert Condition</td>
<td></td>
</tr>
<tr>
<td>Threshold</td>
<td>&gt; 691797</td>
</tr>
</tbody>
</table>

Notification Methods

| Mobile | | |
| Email | | |

Notification Limitations

| Notify once | | |
| Notify when no longer met | | |

After you have set the conditions, tap **Save** and the new alert will be added to Sisense Pulse.
Managing Alerts

You can manage your alerts from Sisense Pulse. In Sisense Pulse, tap the relevant tile to open Tile Mode. Once you're in Tile Mode, the menu icon is displayed in the top right corner. Tap it to display a list of alert management options.

- Open Dashboard
- Edit Alert
- Receive Notifications
- Turn Off
- Delete

From the options area, you can edit alerts, delete alerts, deactivate alerts, and toggle notifications.
Pulse Notifications

If your Sisense user has been configured to receive Pulse alerts to your mobile device, when an alert is triggered, you will receive a notification from the Sisense Mobile app.

You can swipe left on an alert to clear the alert or to view the alert in Sisense Pulse.

To clear all alerts, click the trash can icon at the top of the notification feed. **Note:** You must have Sisense Mobile installed to receive Pulse notifications.

**Single Sign On**

Sisense supports SSO for Sisense Mobile. If your Sisense server is configured to use SSO, Sisense Mobile will use the same SSO configuration. No additional configuration is required on the Sisense server.

When you log in to Sisense Mobile, you are redirected to the IdP login URL using web view. Following authentication by the IdP, the SSO token is returned to the Sisense Server, which returns a web cookie to Sisense Mobile, and the application
is opened. For more information about how to implement SSO, see [Single Sign On](#).

**Plugins**

Sisense Mobile V2.3 and above supports Sisense plugins and supports plugins made with the Sisense JavaScript API.

For information on developers plugins for Sisense Mobile, [click here](#).
Supported Sisense Plugins

The following table lists Sisense plugins that are supported on Sisense Mobile.

<table>
<thead>
<tr>
<th>Plugin</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregated Table</td>
<td></td>
</tr>
<tr>
<td>Auto Add Filter Based On User</td>
<td></td>
</tr>
<tr>
<td>Comments</td>
<td>Always use an absolute URL for the form.</td>
</tr>
<tr>
<td>Dynamic Buckets</td>
<td></td>
</tr>
<tr>
<td>Embed Images</td>
<td>Always use absolute URLs for the image</td>
</tr>
<tr>
<td>Filtered Measure (Parameter)</td>
<td></td>
</tr>
<tr>
<td>Forecasting</td>
<td></td>
</tr>
<tr>
<td>Jump to Dashboard</td>
<td>From Sisense Mobile BI app version 2.5.2</td>
</tr>
</tbody>
</table>

**Release Notes**

**April 4, 2019**  
Support added for Jump to Dashboard plugin

**April 26, 2018**  
Sample dashboards are now available  
Bug fixes

**March 3, 2018**  
Support for Hours and Minutes time resolutions in date filters (Sisense 7.0.1 and above)  
Ability to collapse/expand Recent Dashboards  
Improved support for SSO connectivity  
Bug fixes