ThoughtSpot

ThoughtSpot Data Integration Guide

Release 6.0 December 13, 2019

910 Hermosa Court, Sunnyvale, California 94085

Table of Contents

Overview

You can install a ThoughtSpot cluster on a hardware appliance, cloud service, or VMware appliance. Your ThoughtSpot installation cannot mix node types. For example, you can have either hardware or VMware nodes, but not both. You can, however, have a cloud cluster for development and use an appliance for production.

This guide instructs you how to prepare each of the following:

- Hardware appliance
- Amazon Web Services (AWS) EC2
- Microsoft Azure
- Google Cloud Platform (GCP)
- VMware

After you configure your nodes, you can contact ThoughtSpot Support by phone, mail, email, or by filing a support ticket.

Hardware appliance overview

Summary: What is in the box.

The ThoughtSpot appliance hardware will be installed in a rack in your data center. This section describes the typical physical configuration.

Hardware provided by ThoughtSpot

When your ThoughtSpot appliance arrives, the following items will be included:

Item Name	UOM	Qty
Round Hole to Sq Hole Adapter Kit (For Slide Rail Management)	Each	1
Power Cord, C13 to C14, 6 feet	Each	2
Power Cord, C13 to NEMA 5-15, 6 feet[1] **This power cord is not included with the Haswell platform.**	Each	2
Document, Rack Rail Installation, TS-2000	Each	1
TS-2000 Quick Start Guide	Each	1
Bezel Assembly, TS-2000	Each	1
Slide Rail Kit	Each	1
Appliance (containing 1-4 nodes, depending on ordered configuration)	Each	1
SFP+ Connector per ordered node (data connection)	Each	1
5m Fiber cable per ordered node (data connection)	Each	1
5m Network cable per ordered node (management connection)	Each	1

1: The supply voltage, 120 VAC, available when using a NEMA-15 power cord is an insufficient input to achieve the full power output required by the Haswell power supply. Only the C13 to C14 power cord should be used with the Haswell platform.

Additional hardware requirements

You must supply the following items, as they will not be included with your ThoughtSpot appliance:

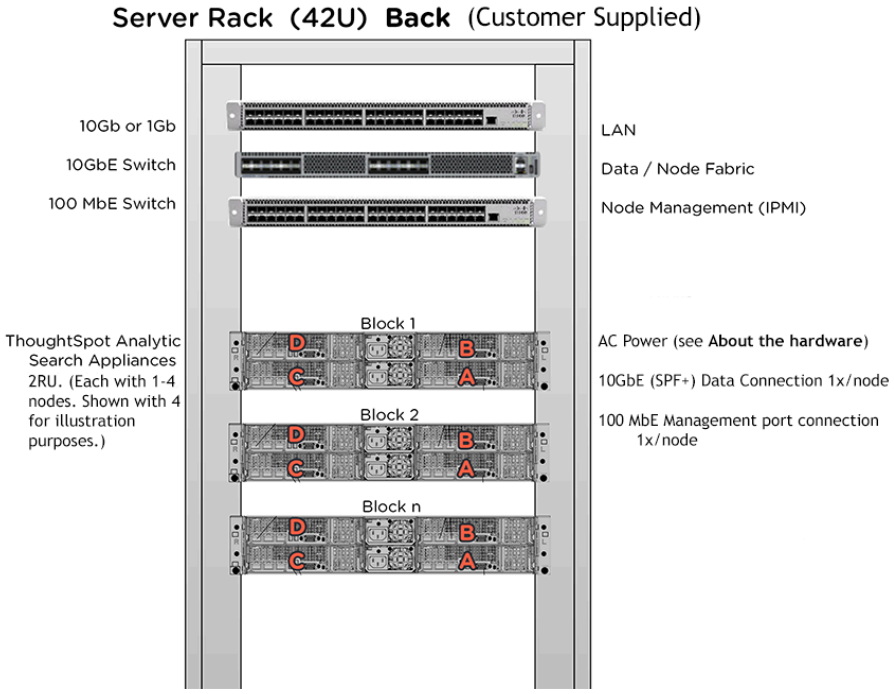
- Data center with proper cooling
- 2U of rack space per appliance (post depth 26.5” - 36.4”)
- AC power **Attention:** Refer to Hardware details for power input requirements.
- 10GbE infrastructure (switch) - 1x port required / node
- 100MbE infrastructure (switch) - 1x port required /node
- Network cable Cat 5e/6 (node management)[1]
- 10G connection: SFP+ for switch side[2]

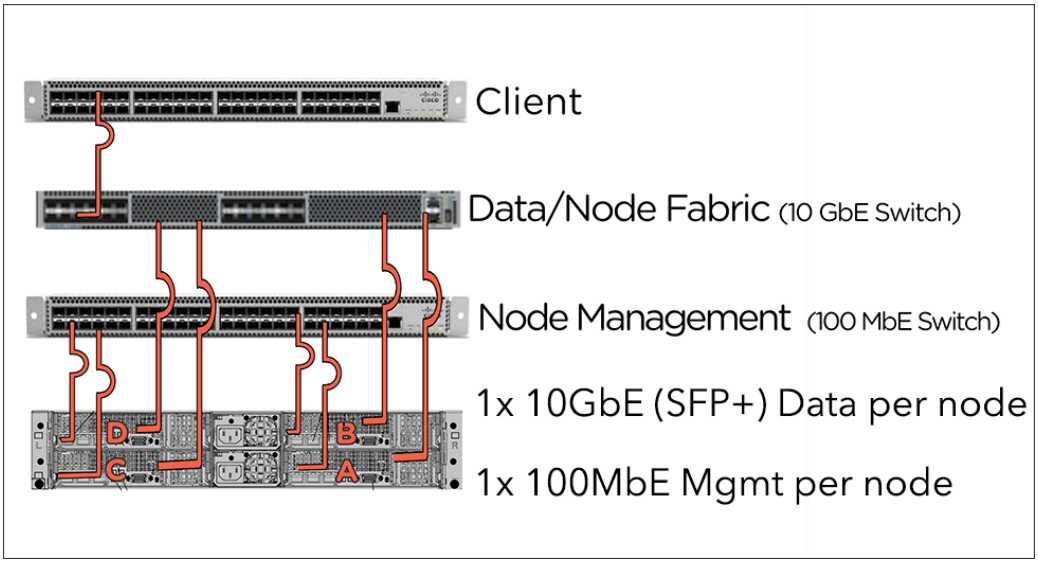
1. One 5m CAT 5e/6 network cable, per node, is provided with the appliance for management port connection. Customer supplied cable can be used if preferred.

2. One SFP+ connector is provided, per node, for the node side data connection. One 5m fiber cable is also provided. The customer must provide switch side SFP+ that is compatible with their switch. Customer supplied DAC cables or fiber cables can be used if preferred.

Typical physical deployment

These diagrams show a physical configuration with three blocks of four nodes each. Your appliance can have 1-4 nodes, depending on the ordered configuration.


Server Rack (42U) Back (Customer Supplied)
10Gb or 1Gb
10GbE Switch
100 MbE Switch
LAN
Data / Node Fabric
Node Management (IPMI)
ThoughtSpot Analytic Search Appliances 2RU. (Each with 1-4 nodes. Shown with 4 for illustration purposes.)
Block 1
Block 2
Block n
D
C
B
A
AC Power (see About the hardware)
10GbE (SPF+) Data Connection 1x/node
100 MbE Management port connection 1x/node


Client
Data/Node Fabric (10 GbE Switch)
Node Management (100 MbE Switch)
1x 10GbE (SFP+) Data per node
1x 100MbE Mgmt per node
D
C
B
A

Supported hardware

Summary: Required and provided installation hardware.

This section lists all required hardware that is needed to successfully install your ThoughtSpot appliance in your data center. Some hardware will be provided with your appliance, while the rest must be provided on-site.

The ThoughtSpot instance hardware is configured for fast data searching and reliability. This overview details the hardware specification and installation. The system is made up of compute nodes, which form a cluster. The 2U system includes up to 4 nodes and can hold up to 1TB of data. This can be scaled out.

Network connection

Before you can access ThoughtSpot, you need a network connection.

Refer to Network Ports in the Administrator's Guide to see which ports must remain open to outside traffic for handling certain network requests and for inter-cluster communication. The Administrator's Guide also provides information on network security and how to test your network connectivity between nodes.

Here are some more details on ports and node communication:

- Port redundancy (bonding) is not supported. Only one 10G port is active per node.
- Nodes communicate with each other through the 10G connection (data ports).
- All nodes should be on the same VLAN – ideally connected to the same top of rack switch.
- IPMI ports are used for management functions of the nodes.

Appliance hardware platforms

You can deploy the ThoughtSpot Analytical Search engine on Haswell appliance hardware platforms, with the following specifications:

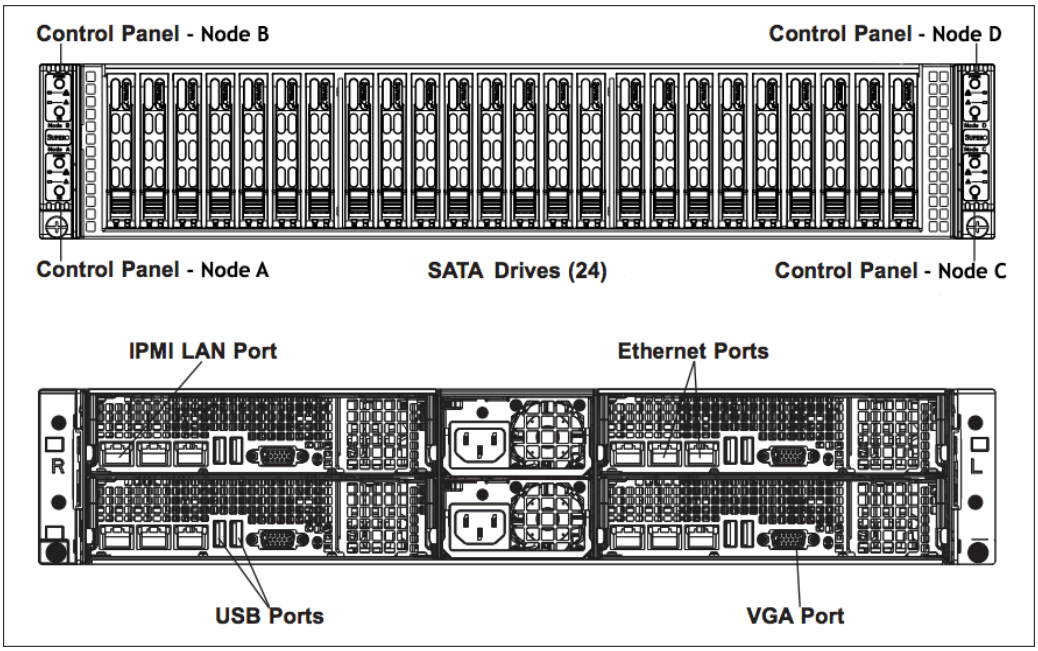
Details	Haswell
Dimensions	2 RU chassis (17.25 x 3.47 x 28.5 in.)
# of nodes	Populated with 1 to 4 nodes
Node specifications	Each node is independent and consists of a server board (removable from rear), 1x 200GB SSD, 3x 2TB HDD
Max power consumption	2000 W
Required power input	200-240 / 11.8 - 9.8A / 50-60Hz

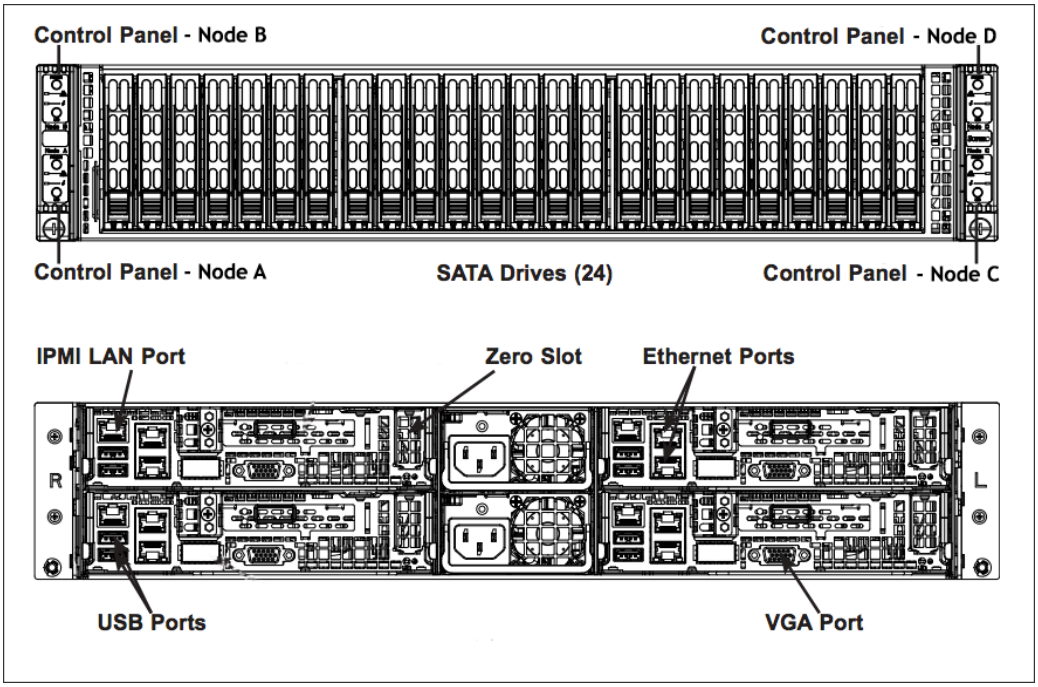
Note: ThoughtSpot deployments are no longer offered on Ivy Bridge platforms.

Chassis views

These diagrams show the front and rear chassis views. The marked features are present on all four nodes on the rear of the chassis even though they are only pointed out on one node in the diagrams.

The chassis appear fully populated (4-nodes). Your appliance may be populated with 1-4 nodes, depending on the ordered configuration. If less than 4-nodes were ordered, the empty slot will be filled with a filler panel.


Control Panel - Node B
Control Panel - Node D
Control Panel - Node A
SATA Drives (24)
Control Panel - Node C
IPMI LAN Port
Ethernet Ports
R
L
USB Ports
VGA Port


Control Panel - Node B
Control Panel - Node D
Control Panel - Node A
SATA Drives (24)
Control Panel - Node C
IPMI LAN Port
Zero Slot
Ethernet Ports
R
L
USB Ports
VGA Port

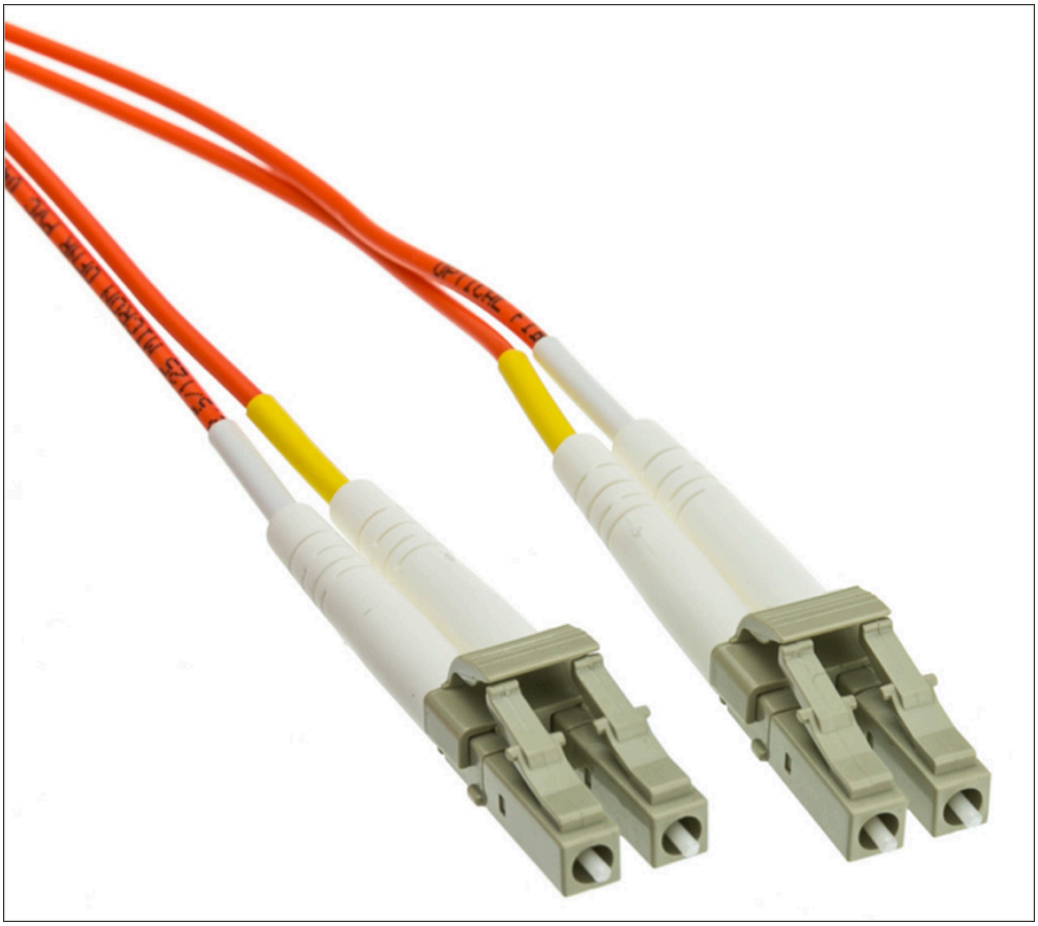
Cable networking

This section reviews the types of cables needed for 10GbE networking and how to plug them in. There are three types of cables to consider for 10GbE networking:

- Fiber
- Direct Attach Copper (DAC)
- Category 6a (not supported by ThoughtSpot)

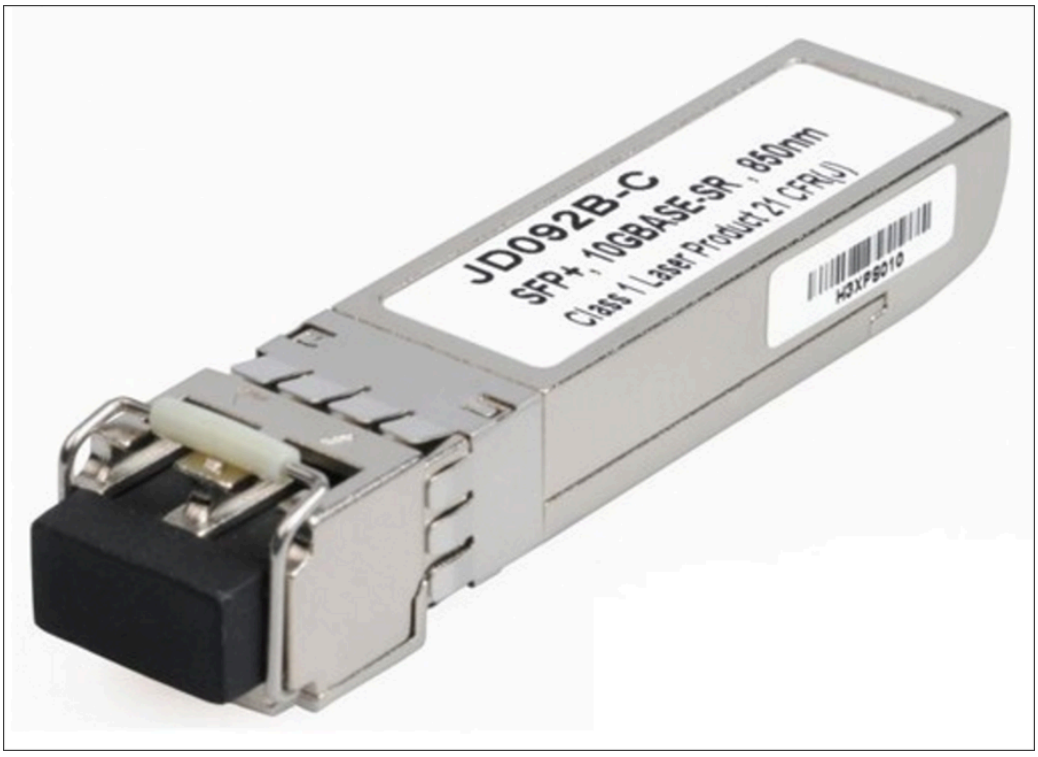
Option 1 - Fiber cables

Fiber can be run long distances to the switch.

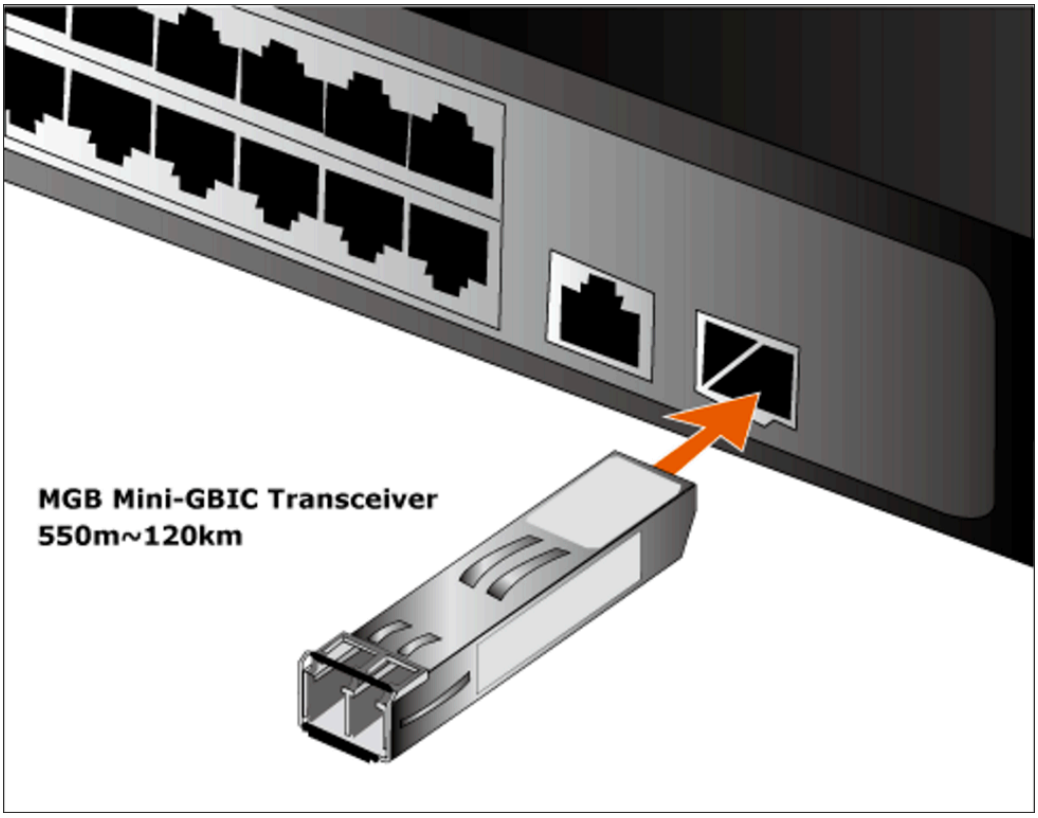


These cables require gigabit interface converters (GBICs), SFP+ form factor.

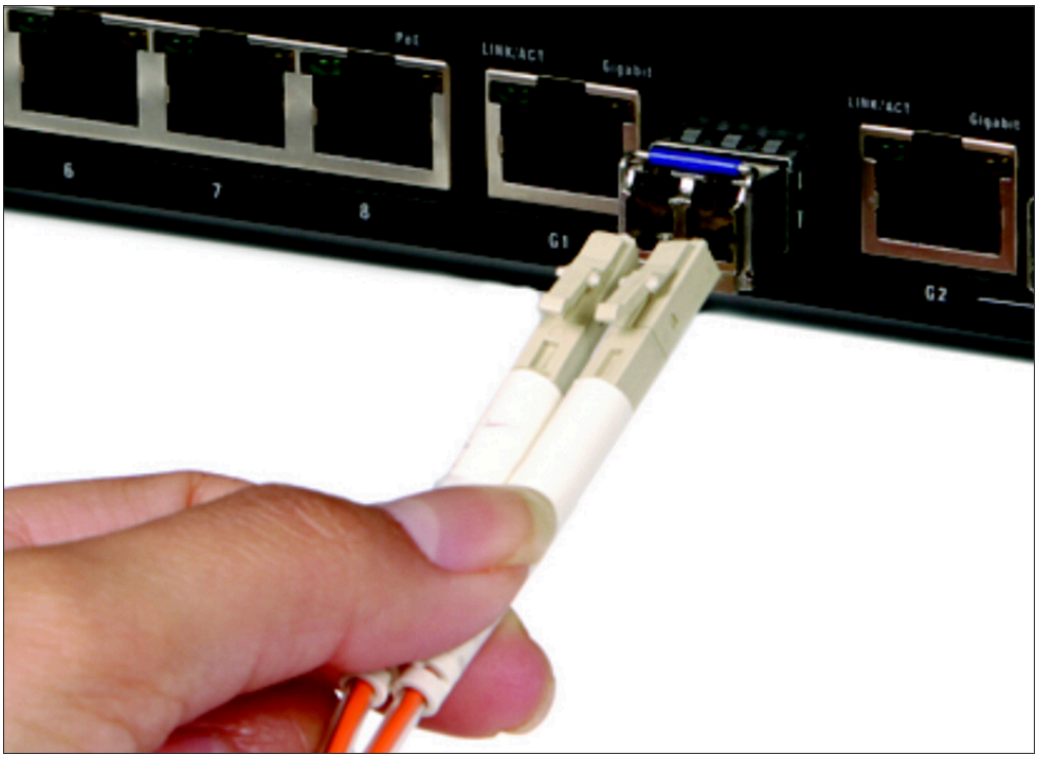
Remember: ThoughtSpot does not supply cables or GBICs



The GBIC must be plugged into a data port on the back of the appliance before plugging in the fiber cables.

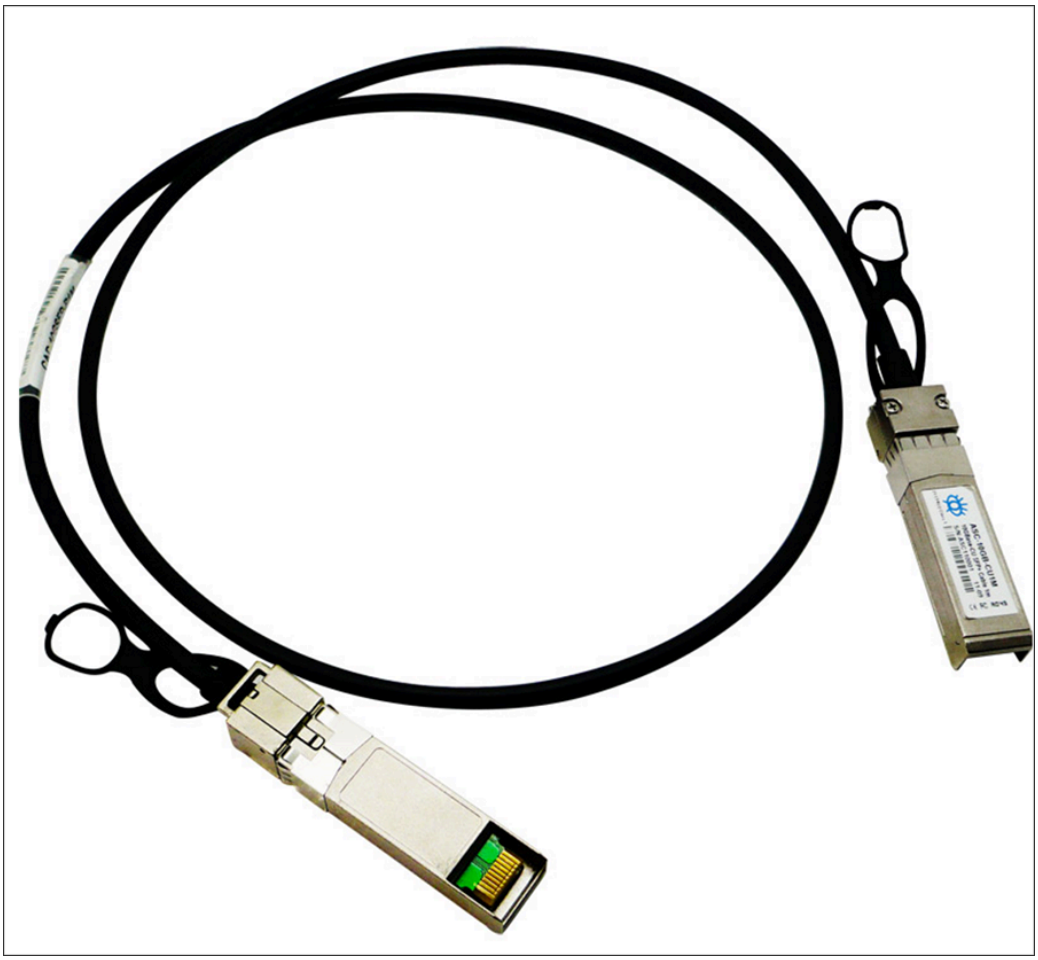


The fiber cables must then be plugged into the GBIC.

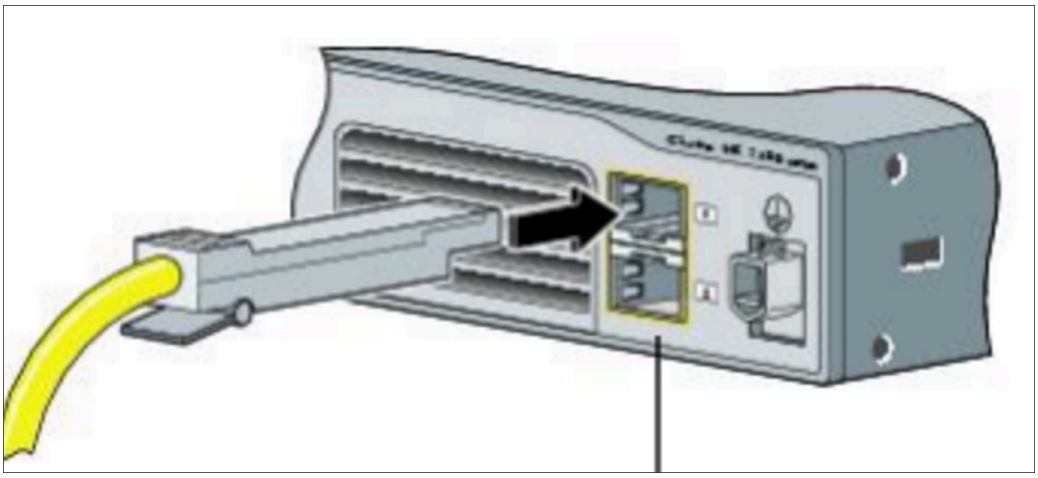


Option 2 - DAC/Twinax cables

Copper can only be run short distances to the switch. An SFP+ is attached to the cable.



Here is how you would plug in a DAC cable.



Non-option - Category 6a cables (not supported by ThoughtSpot)

There are no adapters for these cables. The 10GbE NIC (Network Interface Card) used on the ThoughtSpot appliance is not compatible with this type of cable/connection.

Setup and start the appliance

This section explains how to install and start the appliance.

Before you begin

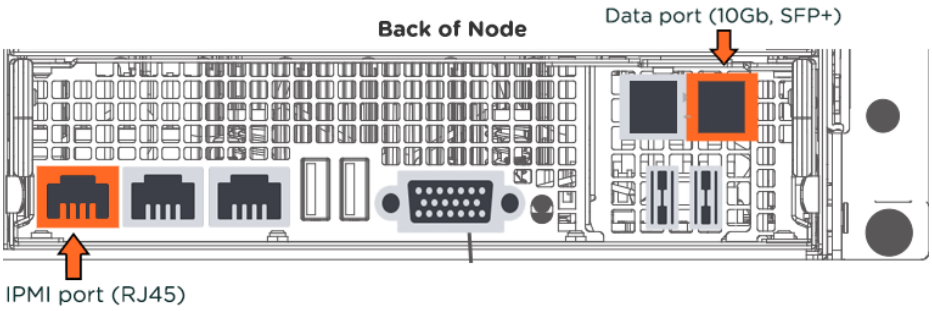
The ThoughtSpot appliance comes pre-installed with all the required software. Network settings on the appliance are required prior to using the appliance. Reference ThoughtSpot's site survey for the information specific to the customer's network environment that is required to configure the appliance.

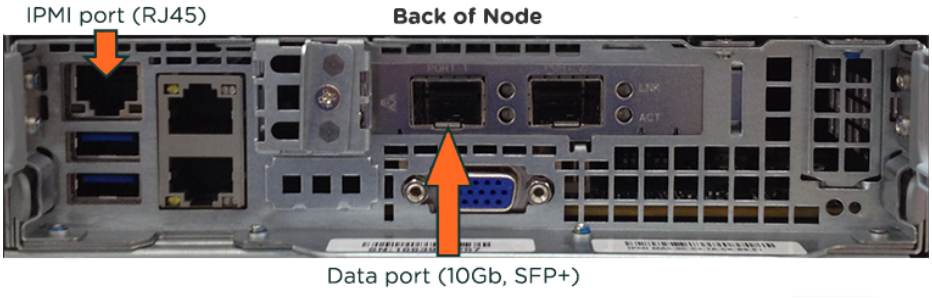
- If ThoughtSpot's site survey form was completed and returned to ThoughtSpot prior to the appliance being shipped, the appliance may be pre-configured for your network environment and ready to install and connect to your network.
- If the network configuration was not pre-set, then this step must be done as part of the installation process.
- If assistance is needed to determine the configuration status of the appliance, please contact ThoughtSpot Support.

Installation procedure

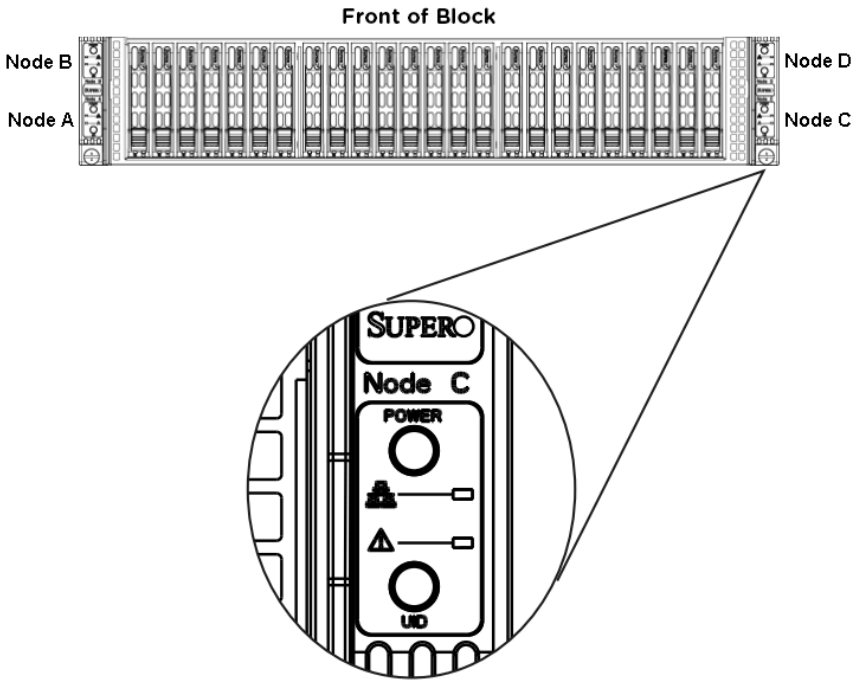
To install and start the appliance and connect to your network:

1. Refer to the Rack Install Guide to install the appliance securely in your data center.
2. Plug in the power cord, but do not turn the appliance on yet. See the figure of the **Location of the power and UID buttons on the control panel** for the power button location.
3. Connect the IPMI dedicated LAN port to a dedicated LAN for system management.
4. Connect the data port(s) on the back of the appliance to your 10GbE network switch. Only the one 10GbE port shown in the following **Back of Node** figures is active. Only one 10GbE port connection is needed.





5. Turn on the appliance by pressing and releasing the power button for each node and allow time for the nodes to boot up completely.



Each node has its own power and UID buttons. Turning the system off using the power button removes the main power, but keeps standby power supplied to the system. Therefore, you must unplug the AC power cord from any external power source before servicing. The power button for each node has a built-in LED which will turn green when the power is on.

There is also a UID button, which is used to turn on or off the blue light function of the LED. After the blue light is activated, the unit can be easily located in very large racks and server banks. A blue LED is also illuminated on the corresponding node, visible from the rear of the chassis.

6. After the appliance has been turned on, verify that both LEDs (IPMI and data NICs) on each network card are lit.
7. Connect a keyboard and monitor to each node in turn. You should see a login prompt on the screen. If you don't see one or the screen isn't responsive, press the key combination control, alt, and F2 on your keyboard, which should allow you to attempt to log in.
8. Log in as username admin, using the default password.
9. Run the following commands and capture the output at every stage:

```
 sudo ipmitool lan print 1
 sudo ipmitool lan set 1 ipsrc static
 sudo ipmitool lan set 1 defgw ipaddr <IPMI_GATEWAY_ADD
R>
 sudo ipmitool lan set 1 netmask <IPMI_VLAN_SUBNET_MASK>
 sudo ipmitool lan set 1 ipaddr <IPMI_NIC_IP_ADDR>
 sudo ipmitool lan print 1
```

10. Run and capture the output of the following commands as well:

```
ifconfig eth0
ifconfig eth1
ifconfig eth2
ifconfig eth3
sudo ethtool eth0
sudo ethtool eth1
sudo ethtool eth2
sudo ethtool eth3
```

11. Share the output of all commands with the ThoughtSpot team who will then determine the next steps.

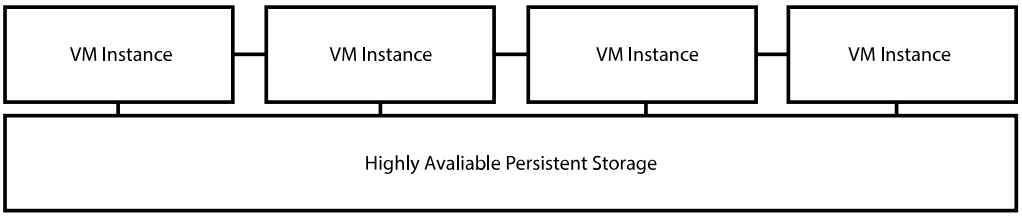
Cloud overview

ThoughtSpot can currently be deployed in the following cloud provider environments:

- Amazon Web Services (AWS) EC2
- Microsoft Azure
- Google Cloud Platform (GCP)

The ThoughtSpot cloud deployment consists of cloud compute (VM) instances and an underlying persistent storage layer. The number of instances required for a cloud deployment is based on the size of the data that needs to be analyzed in ThoughtSpot. The instances act as a distributed cluster of nodes to serve query responses.

	AWS	Azure	GCP
Compute	Virtual machines deployed in your AWS VPC	Virtual machines in your Azure VNET	Virtual machines in your GCP VPC
Persistent storage	Deployment options: 1. Elastic Block Storage 2. S3 + Elastic Block Storage	Premium SSD Managed Disks	Zonal SSD persistent disk



To determine the number of instances and the persistent storage requirements to provision your cluster, please refer to the available instance types for your cloud service provider in the next section.

ThoughtSpot cloud instance types

Refer to the following guidelines for how to set up ThoughtSpot on each cloud service:

- AWS instance types
- Azure instance types

- GCP instance types

Reducing your cloud infrastructure costs

ThoughtSpot recommends following these guidelines to help reduce the cost of your cloud deployment.

Use small and medium instance types when applicable

For ThoughtSpot customers who are deploying their instance with lower data sizes (<=100 GB), ThoughtSpot supports “small” (20 GB data) and “medium” (100 GB data) instance types, as provided at the links above, to help reduce the costs of cloud infrastructure. These are instances with lower CPU/RAM sizes (16/32 vCPU and 128 GB/256 RAM). Advanced lean configuration is required before any data can be loaded onto these instances.

Please contact ThoughtSpot support for assistance with this configuration.

Shut down and restart your cluster

If you do not need your ThoughtSpot cluster to be up and running 24/7, you can shut down your cluster and restart it during normal usage hours to save on the infrastructure costs of running ThoughtSpot instances in cloud provider environments.

To shut down and restart your cluster, do the following in the tscli:

1. Ensure there are no issues with the cluster by running: `$ tscli cluster check`

   The above command should return no failure messages.

2. Stop the cluster by running: `$ tscli cluster stop`

   Wait until you see the message: “Done stopping cluster”

3. Go to your cloud provider’s console and shut down all of the ThoughtSpot VMs in your cluster.

4. When you are ready to use ThoughtSpot again, start up your node VMs.

5. Restart your cluster by running: `$ tscli cluster start`

You should see the message: ”Started pre-existing cluster”

Depending on the size of your cluster, you may need to wait several minutes before the system is up and running. Make sure you budget for this startup time to ensure that the system is fully operational before you expect people to use it.

6. Ensure that your cluster is ready for use by running: `$ tscli cluster status`

   The following messages are displayed to indicate your cluster is up and running:

   `... Cluster: RUNNING`

   `Database: READY`

   `Search Engine: READY`

Automating your cloud deployment

You can automate your deployment, using the free tools in the ThoughtSpot Cloud Deployment GitHub repository.

For more information about automating your cloud deployment, read Deploying ThoughtSpot in the Cloud Using Terraform and Ansible.

AWS configuration options

Summary: Your instances require specific configurations of memory, CPU, storage, and networking capacity.

ThoughtSpot can be deployed in your AWS environment by deploying compute (VM) instances in your Amazon VPC as well as an underlying persistent storage infrastructure. Currently two configuration modes are supported by ThoughtSpot:

- Mode 1: Compute VMs + EBS-only persistent storage
- Mode 2: Compute VMs + EBS and S3 persistent storage

The cost of infrastructure for deploying ThoughtSpot is cheaper when using S3. However, there are differences in where data is loaded, as well as in the backup and restore procedure. For assistance in choosing the best mode for your organization, contact your ThoughtSpot representative. For more information on purchasing ThoughtSpot on AWS, see: ThoughtSpot Pricing.

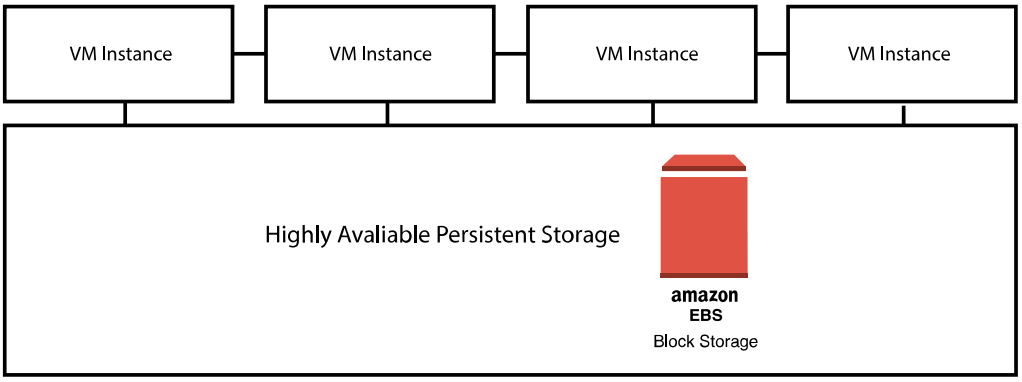
All AWS VMs in a ThoughtSpot cluster must be in the same availability zone (and therefore, also in the same region). ThoughtSpot does not support deploying VMs in the same cluster across availability zones. For more information, see Regions and Availability Zones in Amazon's AWS documentation.

ThoughtSpot AWS instance types

The following sections contain the supported and recommended instance types for a ThoughtSpot AWS deployment. When setting up your cluster in AWS, use the information here to select an instance type, configure the number of instances required for the storage you need, and add data volumes to your cluster.

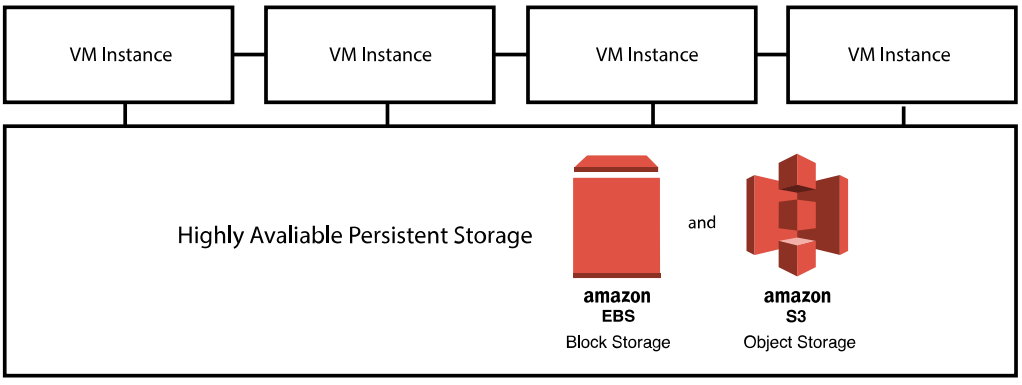
For example: If you were deploying a total cluster data size of 1 TB using the standard r5.16xlarge instance type, you would need 4 instances (VMs), because the per-VM user data capacity of that instance type is 250 GB. If you were deploying EBS-only data volumes, you would need 2x1 TB data volumes per VM.

VMs with EBS-only persistent storage



Per VM user data capacity	Instance type	CPU/RAM	Recommended per-VM EBS volume
20 GB	r4.4xlarge, r5.4xlarge	16/122, 16/128	2X 400 GB
100 GB	r4.8xlarge, r5.8xlarge	32/244, 32/256	2X 400 GB
192 GB	m5.24xlarge	96/384	2X 1 TB
250 GB	r4.16xlarge, r5.16xlarge	64/488, 64/512	2x 1 TB
384 GB	r5.24xlarge	96/768	2X 1.5 TB

VMs with EBS and S3 persistent storage



Per VM user data capacity	Instance type	CPU/RAM	Recommended per-VM EBS volume
20 GB	r4.4xlarge, r5.4xlarge	16/122, 16/128	1x 500 GB
100 GB	r4.8xlarge, r5.8xlarge	32/244, 32/256	1x 500 GB
192 GB	m5.24xlarge	96/384	1x 500 GB
250 GB	r4.16xlarge, r5.16xlarge	64/488, 64/512	1x 500 GB
384 GB	r5.24xlarge	96/768	1x 500 GB

Note: The S3 bucket size is approximately equal to the size of the user data.

Related information

- EC2 instance types
- EC2 pricing
- EBS pricing
- Placement groups

Set up ThoughtSpot in AWS

Summary: After you determine your configuration options, you must set up your virtual machines (VMs) on AWS using a ThoughtSpot Amazon Machine Image (AMI).

Overview of ThoughtSpot setup in AWS

The high-level process for setting up ThoughtSpot in AWS involves these steps:

1. Gain access to ThoughtSpot AMIs.
2. Choose a VM instance configuration recommended by ThoughtSpot.
3. Set up your Amazon S3 bucket (optional).
4. Set up your ThoughtSpot cluster in AWS.
5. Contact ThoughtSpot to finish setting up your cluster.
6. Open the required network ports for communication for the nodes in your cluster and end users.

About the ThoughtSpot AMI

An Amazon Machine image (AMI) is a preconfigured template that provides the information required to launch an instance. You must specify an AMI when you launch an instance in AWS.

To make deployment easy, the ThoughtSpot AMI includes a custom ThoughtSpot image, with the following components:

- A template for the root volume for the instance, such as an operating system, an appliance server, and applications.
- Launch permissions that control which AWS accounts can use the AMI to launch instances.
- A block device mapping that specifies the volumes to attach to the instance when it launches.

The ThoughtSpot AMI has specific applications on a CentOS base image. The AMI includes the EBS volumes necessary to install ThoughtSpot in AWS. When you launch an EC2 instance from this image, it automatically sizes and provisions the EBS volumes. The base AMI includes 200 GB (xvda), 2X400 GB (xvdb), and SSD (gp2). It contains the maximum number of disks to handle a fully loaded VM.

Prerequisites

To install and launch ThoughtSpot, you must have the following:

- Familiarity with Linux administration, and a general understanding of cloud deployment models.
- The necessary AWS Identity and Access Management (IAM) users and roles assigned to you to access and deploy the various AWS resources and services as defined in the Required AWS components section that follows.
  For more information about IAM, see: What Is IAM? in Amazon's AWS documentation.

Required AWS components

- An AWS VPC. For details, see VPC and Subnets in Amazon's AWS documentation.
- A ThoughtSpot AMI. For details, see the next section.
- AWS security groups. For required open ports, see network policies.
- AWS VM instances. For instance type recommendations, see ThoughtSpot AWS instance types.
- EBS volumes.
- (Optional) If deploying with S3 persistent storage, one S3 bucket dedicated to each ThoughtSpot cluster.

Guidelines for setting up your EC2 instances

- Sign in to your AWS account.
- Copy the following ThoughtSpot public AMI which has been made available in N. California region to your AWS region:
  AMI Name: thoughtspot-image-20190718-dda1cc60a58-prod
  AMI ID: ami-0b23846e4761375f1
  Region: N. California

  > ℹ **Note:** The AMI is backward-compatible with ThoughtSpot releases 5.1.x - 5.2.x.

- Choose the appropriate EC2 instance type: See ThoughtSpot AWS instance types for supported instance types.
- Networking requirements: 10 GbE network bandwidth is needed between the VMs. This is the

default for the VM type recommended by ThoughtSpot.

- Security: The VMs that are part of a cluster need to be accessible by each other, which means they need to be on the same Amazon Virtual Private Cloud (VPC) and subnetwork. Additional external access may be required to bring data in/out of the VMs to your network.
- Number of EC2 instances needed: Based on the datasets, this number varies. Please check ThoughtSpot AWS instance types for recommended nodes for a given data size.
- Staging larger datasets (> 50 GB per VM), may require provisioning additional attached EBS volumes that are SSD (gp2).

Setting up your Amazon S3 bucket

If you are going to deploy your cluster using the S3-storage option, you must set up that bucket before you set up your cluster. Contact ThoughtSpot Support to find out if your specific cluster size can benefit from the S3 storage option.

To set up an Amazon S3 bucket in AWS, do the following:

1. In AWS, navigate to the S3 service dashboard by clicking **Services**, then **S3**.
2. Make sure the selected region in the top-right corner of the dashboard is the same region in which you plan to set up your cluster.
3. Click **Create bucket**.
4. In the **Name and region** page, enter a name for your bucket, select the region where to set up the cluster, and click **Next**.
5. On the **Properties** page, click **Next**.
6. On the Configure options page, make sure **Block *all* public access** is selected, and click **Next**.
7. On the Set permissions page, click **Create bucket**.

Encrypting your data at rest

ThoughtSpot makes use of EBS for the data volumes to store persistent data (in the EBS deployment model) and the boot volume (in the EBS and S3 deployment models). ThoughtSpot recommends that you encrypt your data volumes prior to setting up your ThoughtSpot cluster. If you are using the S3 persistent storage model, you can encrypt the S3 buckets using SSE-S3. ThoughtSpot does not currently support AWS KMS encryption for AWS S3.

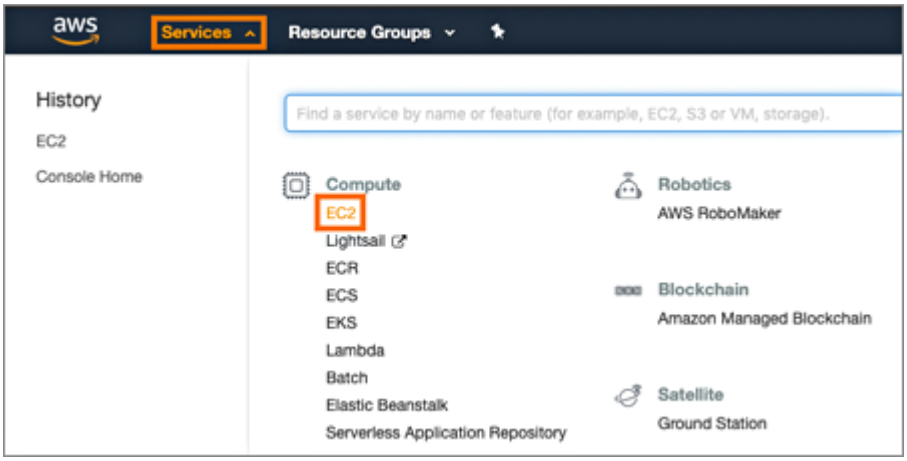
For more information on encryption supported with AWS:

- For EBS, see Amazon EBS Encryption in Amazon's AWS documentation.
- For S3, see Amazon S3 Default Encryption for S3 Buckets in Amazon's AWS documentation.

Setting up your ThoughtSpot cluster

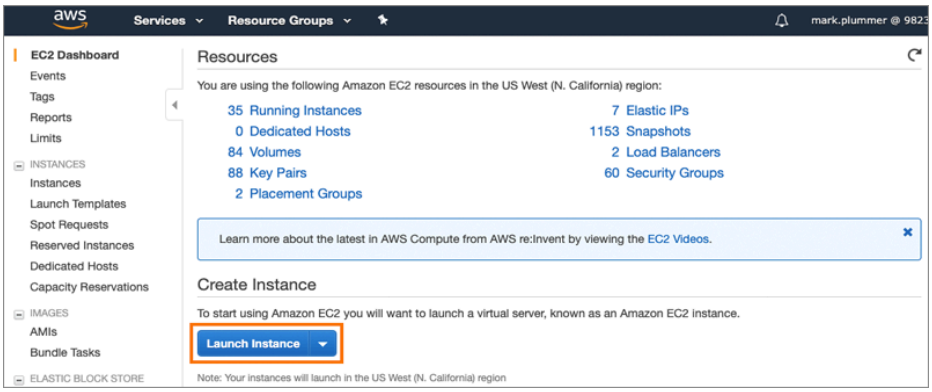
To set up a ThoughtSpot cluster in AWS, do the following:

1. In AWS, navigate to the EC2 service dashboard by clicking **Services**, then **EC2**.



2. Make sure your selected region is correct in the top-right corner of the dashboard. If not, select a different region you would like to launch your instance in. Let ThoughtSpot support know if you change your region.

3. Start the process of launching a VM by clicking **Launch Instance**.



4. Click the **My AMIs** tab, find the ThoughtSpot AMI from the list, and click **Select**.

5. On the Choose an Instance Type page, select a ThoughtSpot-supported instance type. (See ThoughtSpot AWS instance types.)

6. Click **Next: Configure Instance Details**.

7. Configure the instances by choosing the number of EC2 instances you need. The instances must be on the same VPC and subnetwork. ThoughtSpot sets up the instances to be in the same ThoughtSpot cluster.

   S3 storage setting: If you are going to use the S3 storage option, you must go to the **IAM role** menu and select **ec2rolewithfulls3access**. This setting gives your instance access to all S3 buckets in your account's region.

   To restrict the access to a specific bucket, you must create a new IAM role that provides access to the specific bucket, and select it instead. For details on that, click **Create new IAM role**.

8. Click **Next: Add Storage**. Add the required storage based on the storage requirements of the instance type you have selected, and the amount of data you are deploying. For specific storage requirements, refer to ThoughtSpot AWS instance types.

9. When you are done modifying the storage size, click **Next: Add Tags**.

10. Set a name for tagging your instances and click **Next: Configure Security Group**.

11. Select an existing security group to attach new security groups to so that it meets the security requirements for ThoughtSpot.

    ☑ **Tip: Security setting for ThoughtSpot**

    - The VMs need intragroup security, i.e. every VM in a cluster must be accessible from one another. For easier configuration, ThoughtSpot recommends that you enable full access between VMs in a cluster.
    - Additionally, more ports must be opened on the VM to provide data staging capabilities to your network. Check Network policies to determine the minimum required ports that must be opened for your ThoughtSpot appliance.

12. Click **Review and Launch**. After you have reviewed your instance launch details, click **Launch**.

13. Choose a key pair. A key pair consists of a public and private key used to encrypt and decrypt login information. If you don't have a key pair, you must create one, otherwise you won't be able to SSH into the AWS instance later on.

14. Click **Launch Instances**. Wait a few minutes for it to fully start up. After it starts, it appears on the EC2 console.

Prepare the VMs (ThoughtSpot Systems Reliability Team)

⚠ **Important:** This procedure is typically done by a ThoughtSpot Systems Reliability Engineer (SRE). Please consult with your ThoughtSpot Customer Service or Support Engineer on these steps.

Before we can install a ThoughtSpot cluster, an administrator must log into each VM through SSH as user "admin", and complete the following preparation steps:

1. Run `sudo /usr/local/scaligent/bin/prepare_disks.sh` on every machine.
2. Configure each VM based on the site-survey.

When complete, your storage is mounted and ready for use with your cluster.

Launch the cluster

Upload the TS tarball to one of the VMs, and proceed with the normal cluster creation process, using tscli cluster create.

If you are going to use S3 as your persistent storage, you must enable it when running this command, using the **enable_cloud_storage** flag. Example: `tscli cluster create 6.0-167.tar.gz --enable_cloud_storage=s3a`

When the setup is complete, you can load data into ThoughtSpot for search analytics.

Open the required network ports

To determine which network ports to open for a functional ThoughtSpot cluster, see Network policies.

Related information

EC2 Best Practices

Loading data from an AWS S3 bucket

Azure configuration options

ThoughtSpot can be deployed in your Azure environment by deploying compute (VM) instances in your VNET as well as an underlying persistent storage infrastructure. Currently we support Premium SSD Managed Disks for persistent storage. For more information, see Managed Disks pricing in Microsoft's Azure documentation.

All Azure VMs (nodes) in a ThoughtSpot cluster must be in the same availability zone (and, therefore, also in the same region). ThoughtSpot does not support deploying VMs(nodes) of the same cluster across availability zones. For more information, see What are Availability Zones in Azure? in Microsoft's Azure documentation.

ThoughtSpot Azure instance types

Per VM user data capacity	Instance type	CPU/ RAM	Recommended per-VM Premium SSD Managed Disk volume
200 GB	E64sv3	64/432	2x1 TB
100 GB	E32sv3	32/256	2X 400 GB
20 GB	E16sv3	16/128	2X 400 GB
120 GB	D64v3	64/256	2X 1 TB

Set up ThoughtSpot in Azure

Summary: After you determine your configuration options, you must set up your virtual machines using a ThoughtSpot image for Azure.

About the ThoughtSpot image

To provision ThoughtSpot in the Azure portal, you'll need to access the ThoughtSpot Virtual Machine in the Azure Marketplace.

The ThoughtSpot Virtual Machine comes provisioned with the custom ThoughtSpot image to make hosting simple. A virtual machine is a preconfigured template that provides the information required to launch an instance of ThoughtSpot. It includes the following:

- A template for the root volume for the instance (for example, an operating system, an appliance server, and applications).

The ThoughtSpot Virtual Machine has the ThoughtSpot software installed and configured, on an CentOS base image. Check with your ThoughtSpot contact to learn about the latest version of the ThoughtSpot Virtual Machine.

Due to security restrictions, the ThoughtSpot Virtual Machine does not have default passwords for the administrator users. When you are ready to obtain the password, contact ThoughtSpot Support.

Set up ThoughtSpot in Azure

Follow these steps to provision and set up the VMs and launch ThoughtSpot.

Prerequisites

Complete these steps before launching your ThoughtSpot Virtual Machine:

1. Obtain an Azure login account.
2. Set up usage payment details with Microsoft Azure.
3. Set up a Resource Group.

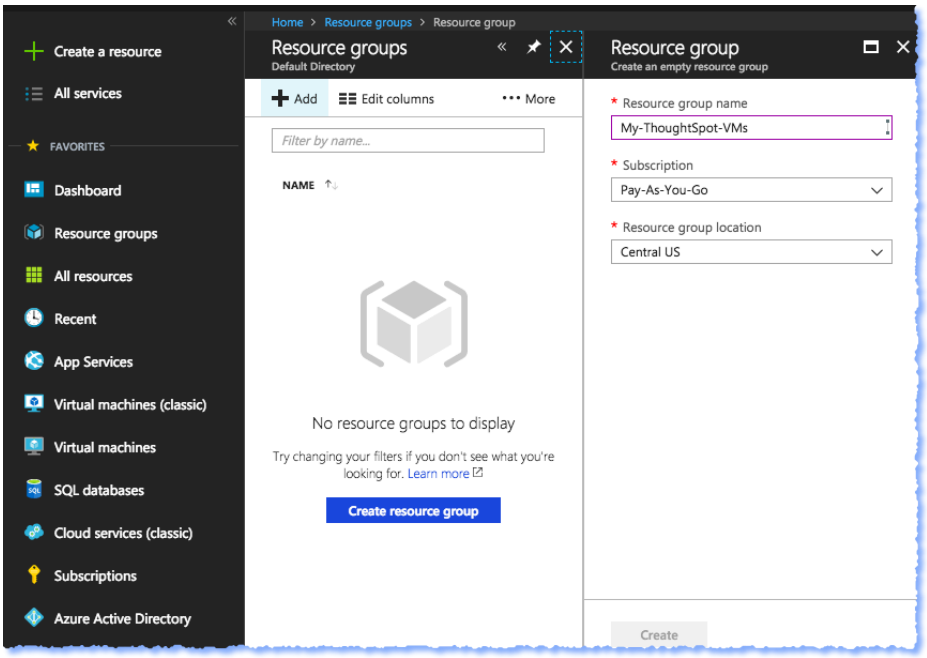
Create an instance

To get started, you need to log into the Azure portal, create a resource group, get the ThoughtSpot Virtual Machine on the Azure Marketplace, create a resource based on the VM, and complete initial setup. You can either start at the Marketplace or from within the resource group you just created, as described here.

1. Log in to the Azure portal.

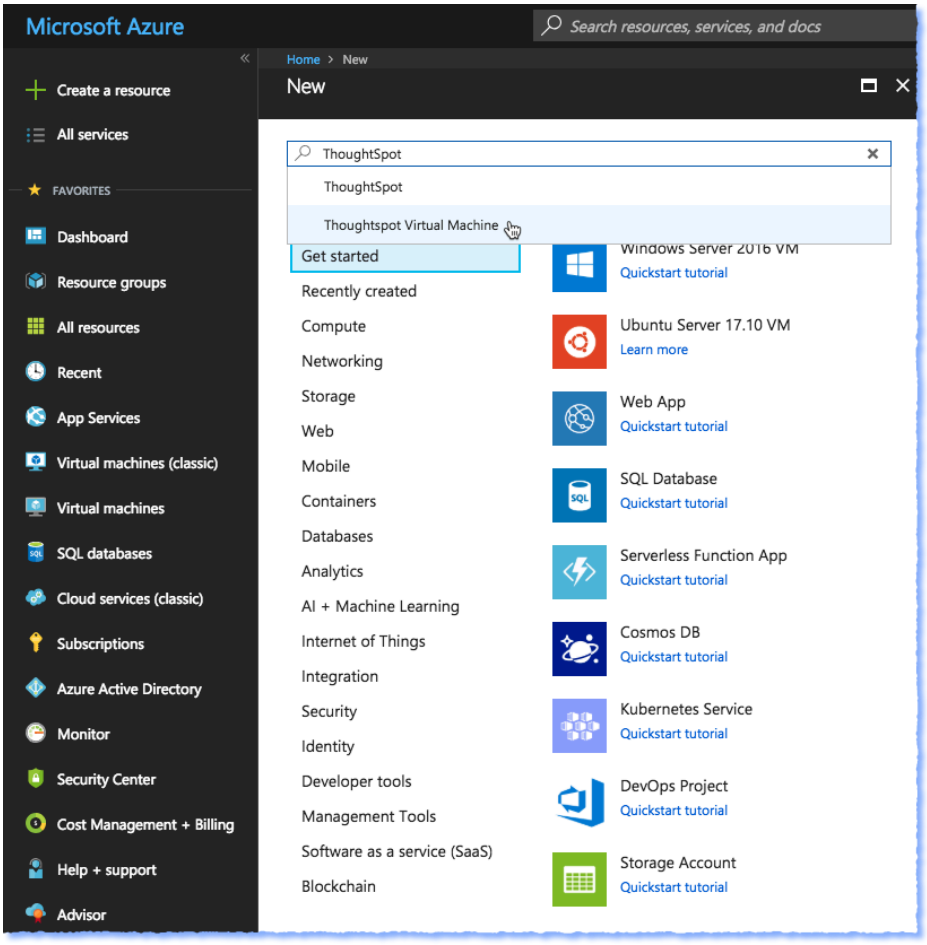
   In a browser, go to http://azure.microsoft.com, and log in to your Azure account.

2. Create a Resource Group.

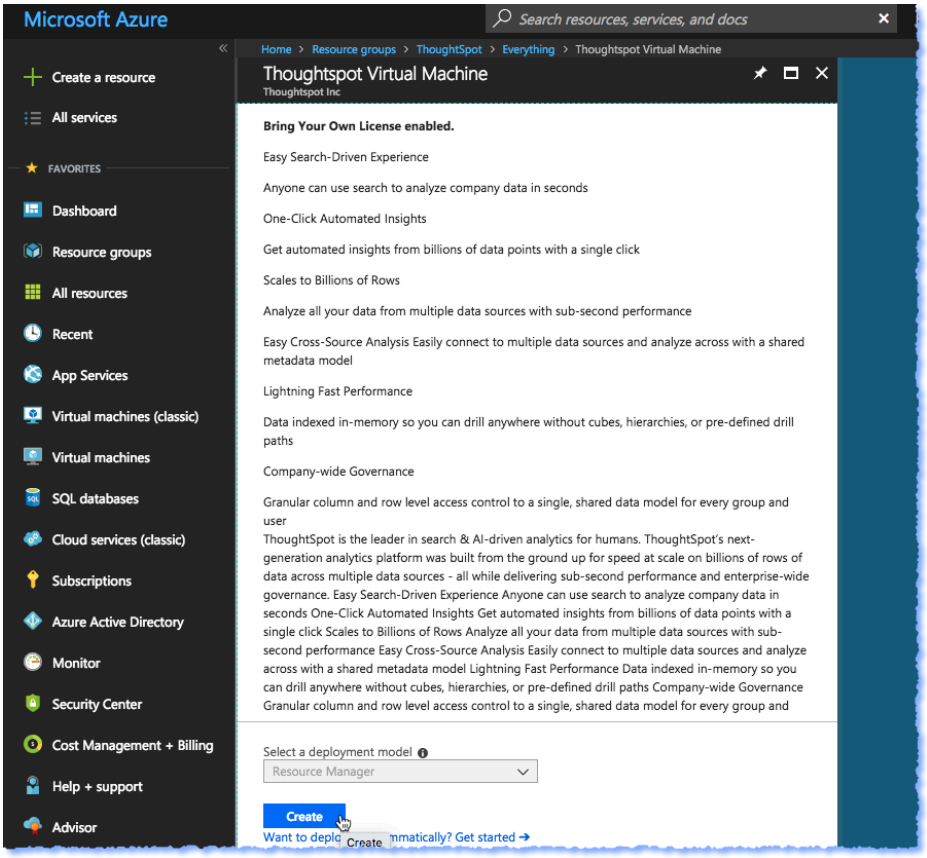


3. Next, create a resource based on the ThoughtSpot Virtual Machine.

   a. Click **Create a resource**, search the Marketplace for the ThoughtSpot Virtual Machine, and select it.



b. On the ThoughtSpot Virtual Machine page, click **Create**.



Configure basic settings

1. Provide a name and password for your new virtual machine.

2. Choose a disk type.

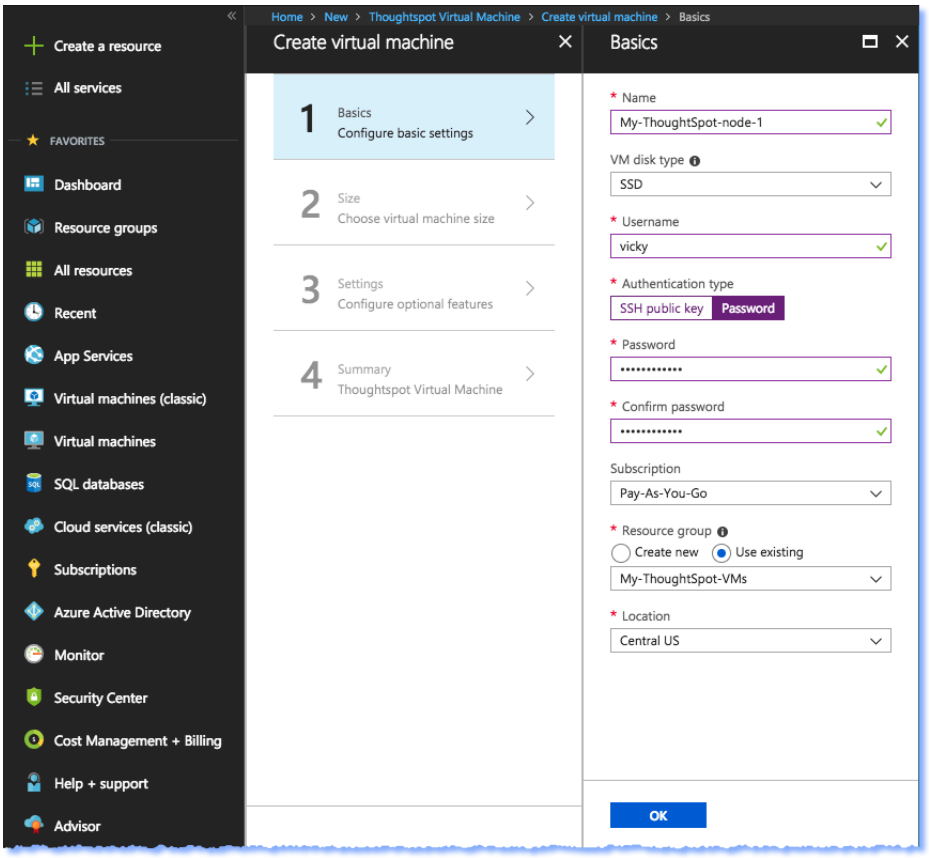
   > ☑ **Tip:** the new SSD disk types are currently available for only particular regions, so if you choose this disk type, make sure it's supported on the region you chose for your VM.

   See Standard SSD Disks for Virtual Machine workloads for more on SSD disks.

3. Provide a Resource Group, by clicking `existing` and selecting one.

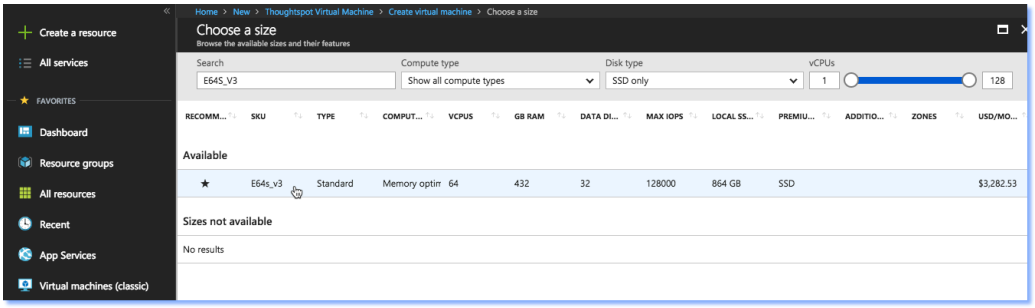
4. Select a location.

5. Click **OK** to save the Basics, which should look similar to the following example.



Choose a machine size

For **Choose a size**, select `E64S_V3 standard`.



Configure network settings, storage, and other options

Prerequisite: Get the details needed for setting up the Virtual Network, Subnet, and Network Security Group from your Azure support team.

1. For storage, select **Yes** to **use managed disks**.

2. Under **Network**, select **Virtual network**, then **Subnet**, then **Public IP addresses**, and set those names, addresses, and ranges approriately for your network.

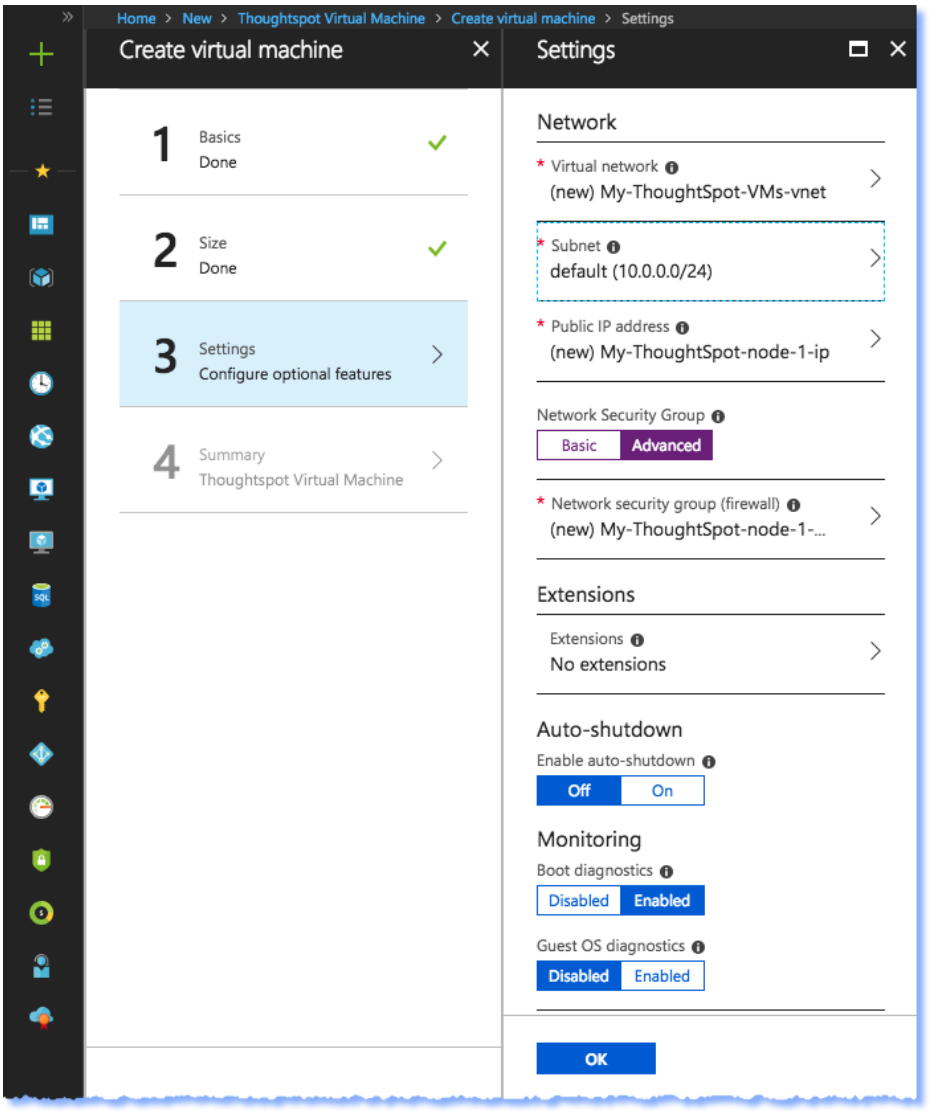
3. Open the necessary Inbound and Outbound ports to ensure that the ThoughtSpot processes do not get blocked.

   The minimum ports needed are:

Port	Protocol	Service
22	SSH	Secure Shell access
80	HTTP	Web access
443	HTTPS	Secure Web access
12345	TCP	ODBC and JDBC drivers access
2201	HTTP	Cluster Debugging
2101	HTTP	Node daemon Debugging
4001	HTTP	Data Cache Debugging

   ℹ **Note:** ThoughtSpot requires that nodes purchased from Azure must be reachable to each other so that they can communicate and form a distributed environment. ThoughtSpot only requires that those ports be accessible between nodes within a cluster. Use your discretion about whether to restrict public access or not for all nodes/ all ports

4. Leave other configurations such as `auto shutdown` and `monitoring` on their default settings.



5. Click **OK**.

   Azure will do the final validation check.

Review the Summary

Verify that the validation check succeeded and that summary of information shown is correct. If you find errors, reconfigure as needed.

When you are satisfied with the virtual machine setup, click **Create**.

Prepare for starting up ThoughtSpot

Prerequisite: To log in to the VM, you need the private key that is available in the image. You can obtain this from your ThoughtSpot contact.

1. Obtain the VM's public and private IP addresses.
    - To see the public IP, click the VM name link. This will show the public IP of the VM.
    - To see the private IP click Networking (under SETTINGS on the left side of the screen).
2. Connect to the VM through SSH, using the private key provided for the admin user.
    - You must file a support ticket to obtain this private key; it is necessary for the first login.
    - This key is different from the credentials, or the private keys supplied in earlier steps, which do not work in this context.
3. Update the password for both the `admin` and the `thoughtspot` users.

```
$ sudo passwd admin Changing password for user admin
$ sudo passwd thoughtspot Changing password for user tho
ughtspot
```

4. Update the file `/etc/hosts` with all the node IP addresses for the other VMs that will be part of the ThoughtSpot cluster.

Add Storage Disks

1. Go back to the VM and click it.
2. Add 2 SSD disks of 1TB each.
3. Click **Add data disk** and choose **Create disk from the menu**.
4. Create one mode data disk (demo-disk2) and save them both.
5. Click **Save** to add the disks to the VM.
6. Verify that the disks were added by issuing this command in the shell on the VM:

```
$ lsblk
```

Which returns results like:

```
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
fd0       2:0    1    4K  0 disk
sda       8:0    0  200G  0 disk
├─sda1    8:1    0    1G  0 part /mntboot
├─sda2    8:2    0   20G  0 part /
├─sda3    8:3    0   20G  0 part /update
└─sda4    8:4    0  159G  0 part /export
sdb       8:16   0    1T  0 disk
└─sb1     8:17   0    1T  0 part /mnt/resource
sdc       8:32   0    1T  0 disk
sdd       8:48   0    1T  0 disk
sr0      11:0    1  628K  0 rom
```

7. Unmount the temporary disk by issuing:

```
$ sudo umount /mnt/resource
```

8. Prepare the disks /dev/sdc and /dev/sdd for ThoughtSpot by issuing the command:

```
$ sudo /usr/local/scaligent/bin/prepare_disks.sh /dev/sd
c /dev/sdd
```

Warning: Do not use the disk /dev/sdb. This is reserved for ThoughtSpot use.

9. Check the disks status by issuing:

```
$ df –h
```

10. Repeat the steps in this section for each node in your cluster.

Make network support settings

☑ **Tip:** All changes in this section must be re-applied each time after a cluster is created or updated. If these changes are not present, a reboot of the VMs will not have network access. So when updating these files, keep a backup to copy after any subsequent cluster creation or update.

1. Update hostnames for all the nodes by issuing:

```
$ sudo hostnamectl set-hostname <HOSTNAME>
```

   If you are using a static name, you can issue:

```
sudo hostnamectl set-hostname <HOSTNAME> --static
```

2. Update `/etc/hosts` with the IP and hostname:

```
$ sudo vi /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE=eth0 ONBOOT=yes BOOTPROTO=dhcp HWADDR=<Add eth0 M
AC> TYPE=Ethernet USERCTL=no PEERDNS=yes IPV6INIT=no
```

3. Do not reboot any of the nodes, until these changes are made to each node:

   a. Open the grub file /update/etc/default/grub in an editor:

```
   $ sudo vi /update/etc/default/grub
```

   b. Change the line:

```
   GRUB_CMDLINE_LINUX="console=tty0 console=ttyS1,115200
n8"
```

to:

```
   GRUB_CMDLINE_LINUX="console=tty0 console=ttyS1,115200
n8 net.ifnames=0"
```

c. Save your changes.

4. Issue these commands:

```
$ sudo cp /update/etc/default/grub /etc/default/
$ rm /usr/local/scaligent/bin/setup-net-devices.sh
```

5. Reboot the nodes.

GCP configuration options

ThoughtSpot can be deployed in your GCP environment by deploying compute (VM) instances in your VPC as well as an underlying persistent storage infrastructure. Currently two configuration modes are supported by ThoughtSpot:

- Mode 1: Compute VMs + SSD Persistent Disk storage-only
- Mode 2: Compute VMs + SSD Persistent Disk and Google Cloud Storage (GCS).

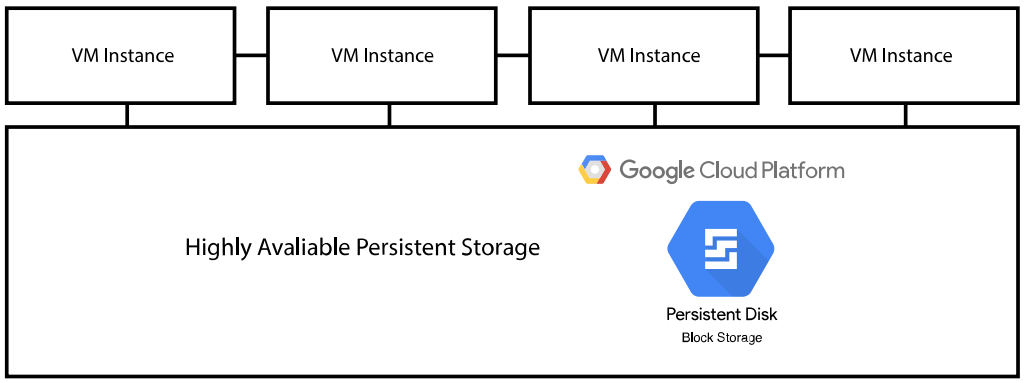
For more information about Persistant Storage, see Zonal Persistent SSD disks in Google's Cloud documentation.

For more information about Google Cloud Storage, see Cloud Storage Buckets in Google's Cloud documentation.

All GCP VMs (nodes) in a ThoughtSpot cluster must be in the same zone (and, therefore, also in the same region). ThoughtSpot does not support deploying VMs (nodes) of the same cluster across zones. For more information, see Regions and Zones in Google's Cloud documentation.

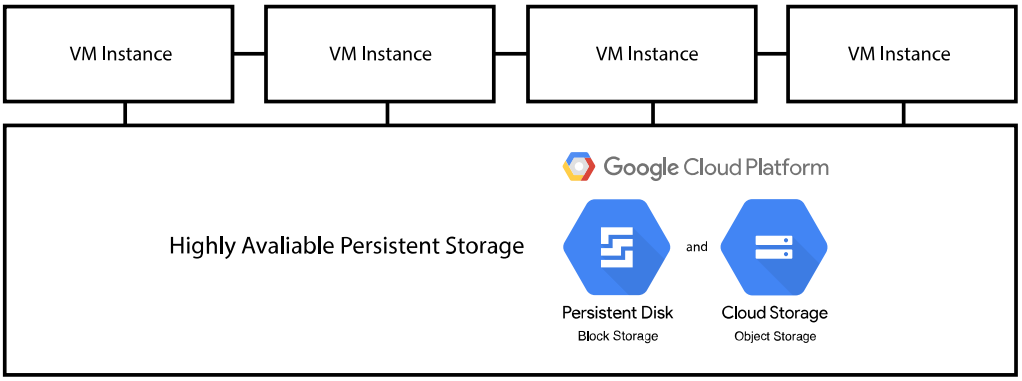
ThoughtSpot GCP instance types

VMs with Persistent Disk-only storage



Per VM user data capacity	Instance type	CPU/RAM	Recommended per-VM Zonal Persistent SSD Disk volume
208 GB	n1-highmem-64	64/416	2x 1 TB
312 GB	n1-highmem-96	96/624	2x 1.5 TB
100 GB	n1-highmem-32	32/208	2X 400 GB
20 GB	n1-highmem-16	16/122	2X 400 GB
180 GB	n1-standard-96	96/330	2X 1 TB

VMs with Persistent Disk and Google Cloud storage



Per VM user data capacity	Instance type	CPU/RAM	Recommended per-VM Zonal Persistent SSD Disk volume
208 GB	n1-highmem-64	64/416	1X 500 GB
312 GB	n1-highmem-96	96/624	1X 500 GB
100 GB	n1-highmem-32	32/208	1X 500 GB
20 GB	n1-highmem-16	16/122	1X 500 GB
180 GB	n1-standard-96	96/330	1X 500 GB

Set up ThoughtSpot in GCP

After you've determined your configuration options, set up your virtual machines (VMs). The ThoughtSpot base image for booting the VMs and some other aspects of system setup will be shared with you on GCP by ThoughtSpot.

About the ThoughtSpot and Google Cloud Platform

ThoughtSpot uses a custom image to populate VMs on GCP. The base image is a Centos derived image, which will be available to you in your Google Compute Engine project for Boot disk options under Custom Images.

Ask your ThoughtSpot liaison for access to this image. We need the Google account/email ID of the individual who will be signed into your organization's GCP console. We will share ThoughtSpot's GCP project with them so they can use the contained boot disk image for creating ThoughtSpot VMs.

Overview

Before you can create a ThoughtSpot cluster, you must provision VMs. You use the Google Compute Engine (GCP) platform for creating and running VMs.

The following topics walk you through this process.

Prerequisites

1. Ensure that **Network Service Tier** is set to **Premium** for all VMs to be used in your ThoughtSpot cluster.
2. A ThoughtSpot cluster requires 10 Gb/s bandwidth (or better) between any two nodes. This must be established before creating a new cluster.

Setting up your Google Cloud Storage (GCS) bucket

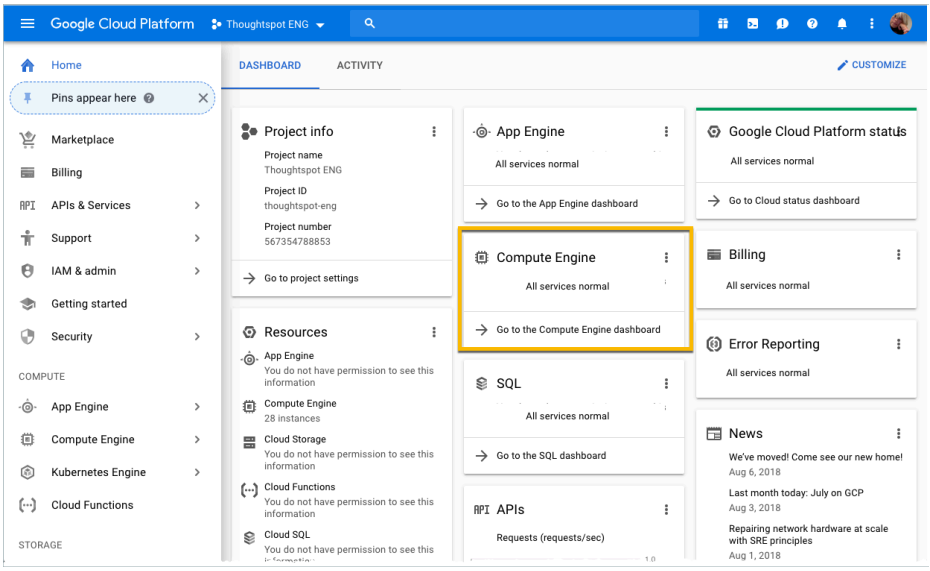
If you are going to deploy your cluster using the GCS-storage option, you must set up that bucket before you set up your cluster. Contact ThoughtSpot Support to find out if your specific cluster size will benefit from the GCS storage option.

1. Sign in to the Google Cloud Console.
2. Go to the Storage dashboard.
3. Click **CREATE BUCKET**.
4. Enter a name for your bucket, and click **CONTINUE**.
5. For location type, select **Region** and use the Location drop-down menu to select the region where you are going to set up your instance, and click **CONTINUE**.
6. For default storage class, make sure **Standard** is selected, and click **CONTINUE**.
7. For access control model, make sure **Set permissions uniformly at bucket-level** is selected, and click **CONTINUE**.
8. For advanced settings, leave Encryption set to **Google-managed key**, do not set a retention policy, and click **CREATE**.

When you create your instance, make sure you set Storage to **Read Write** access.

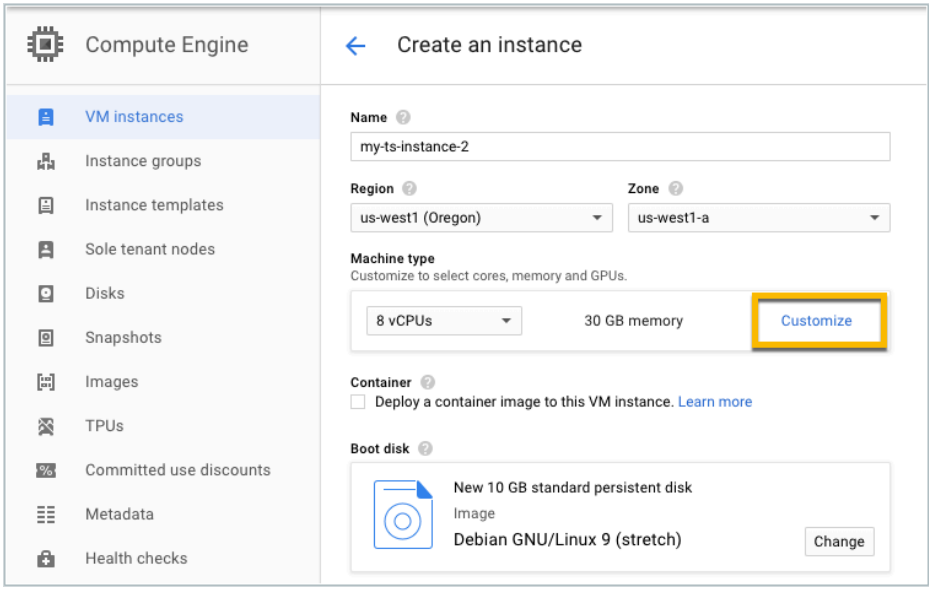
Create an instance

1. Sign in to the Google Cloud Console.
2. Go to the Compute Engine dashboard, and select the associated ThoughtSpot project.



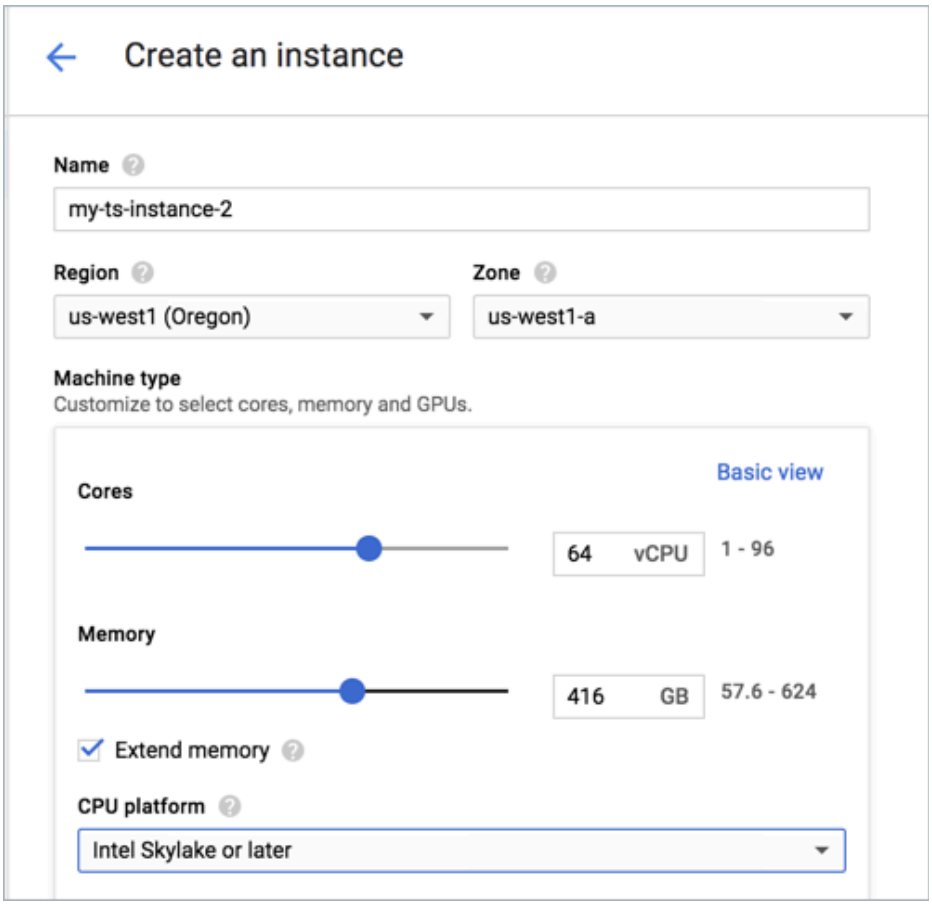
3. Select **VM instances** on the left panel and click **CREATE INSTANCE**.

4. Provide a name for the instance, choose a region, choose number of CPUs (e.g., 8 vCPUs for a cluster), and click **Customize** to further configure CPUs and memory.



5. For **Machine type** set the following configuration:

Setting	Value
Cores	`64 vCPU`
Memory	`416 GB`
Extend memory	Enabled (checkmark)
CPU platform	`Automatic` (or select one of the preferred CPU platforms, `Intel Skylake` or `Intel Broadwell`, if available)

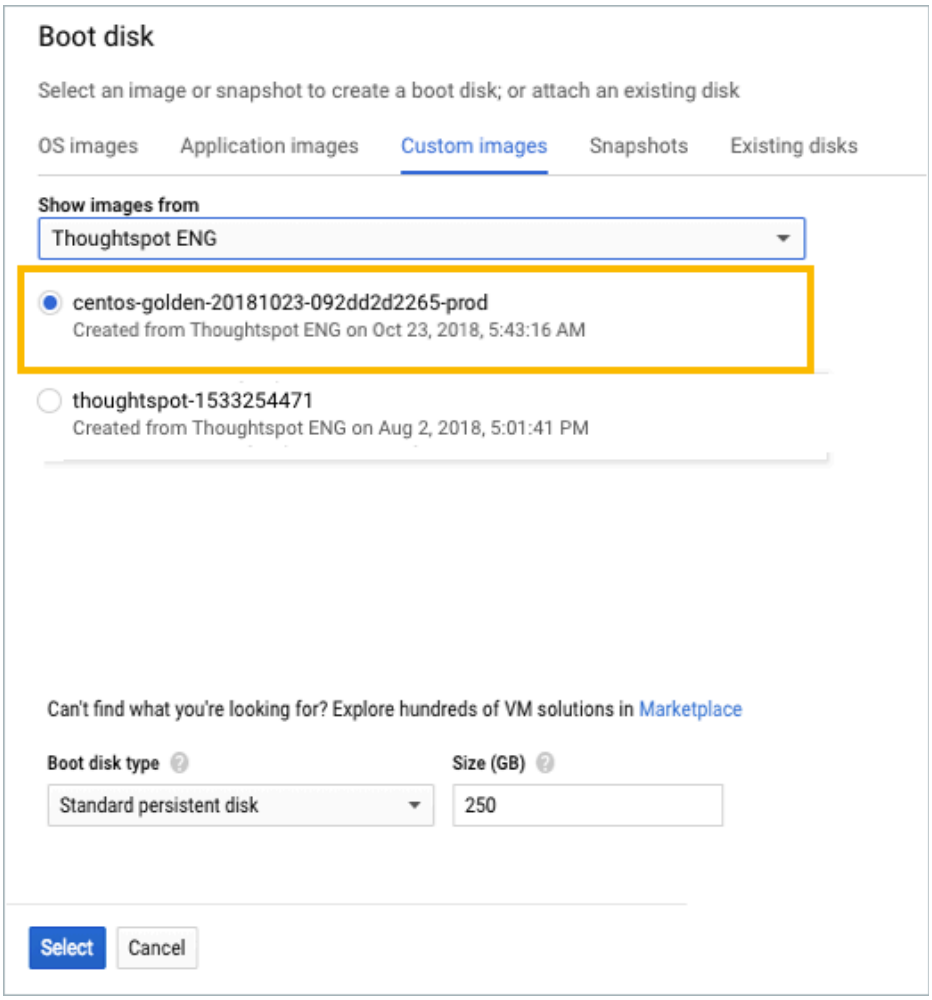


6. Configure the Boot disk.

a. Scroll down to the find the **Boot disk** section and click **Change**.

b. Click **Custom Images** on the tabs at the top, select a ThoughtSpot base image and configure the boot disk as follows:

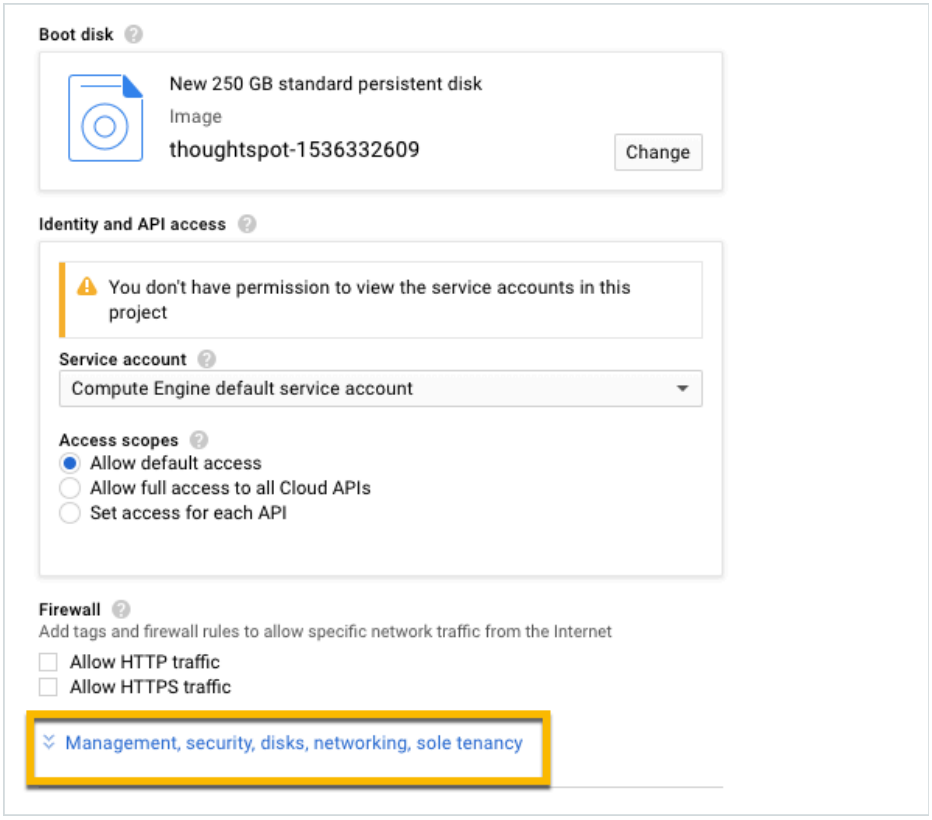
Setting	Value
Image	`ThoughtSpot`
Boot disk type	`Standard persistent disk`
Size (GB)	`250`



> ℹ **Note:** ThoughtSpot updates these base images with patches and enhancements. If more than one image is available, the latest one is always at the top of the list. Both will work, but we recommend using the latest image because it typically contains the latest security and maintenance patches.

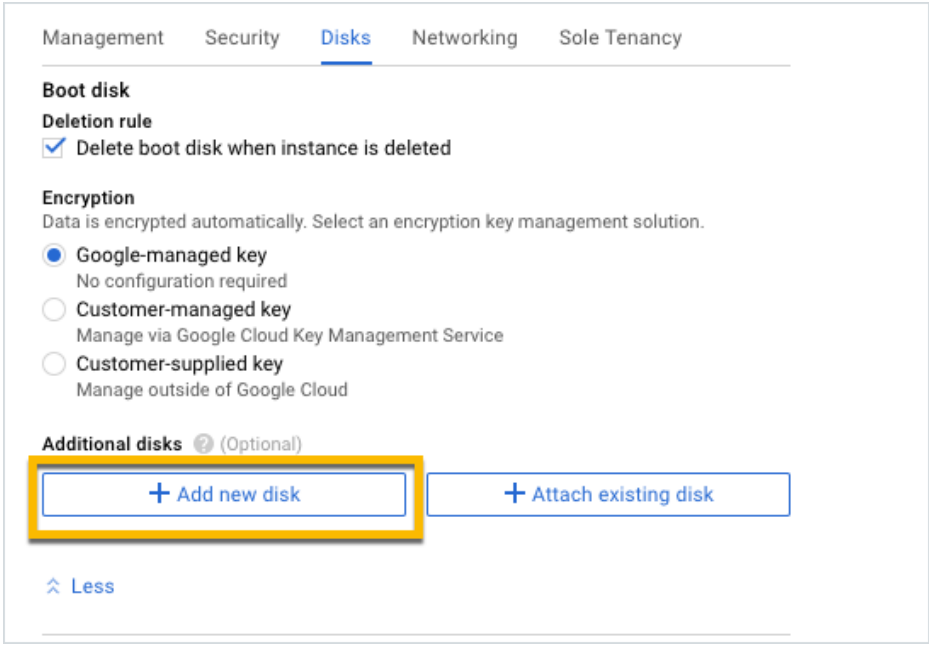
c. Click **Select** to save the boot disk configuration.

7. Back on the main configuration page, click to expand the advanced configuration options (**Management, security, disks, networking, sole tenancy**).



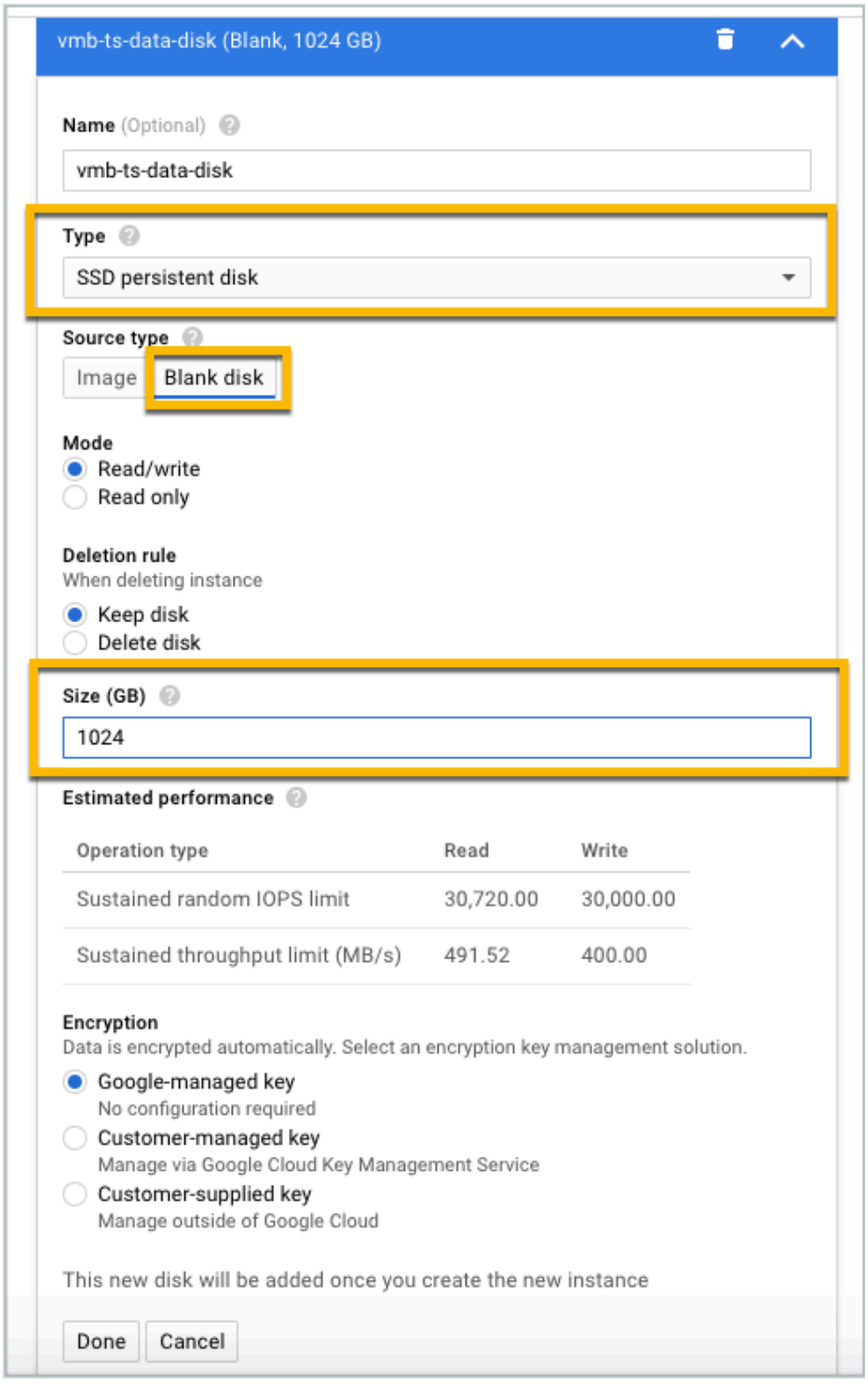
8. Attach two 1 TB SSD drives. These drives will be used for the data storage.

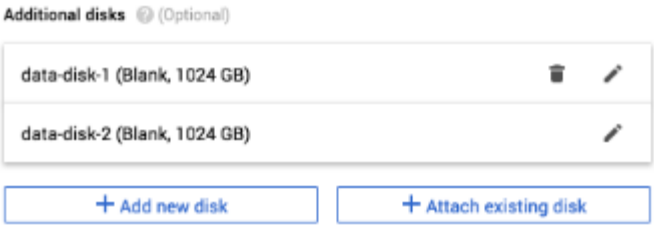
   a. Click the **Disks** tab, and click **Add new disk**.



b. Configure the following settings for each disk.

Setting	Value
Type	`SSD persistent disk`
Source type	`Blank disk`
Size (GB)	`1024`


vmb-ts-data-disk (Blank, 1024 GB)
Name (Optional)
vmb-ts-data-disk
Type
SSD persistent disk
Source type
Image
Blank disk
Mode
Read/write
Read only
Deletion rule
When deleting instance
Keep disk
Delete disk
Size (GB)
1024
Estimated performance
Operation type
Read
Write
Sustained random IOPS limit
30,720.00
30,000.00
Sustained throughput limit (MB/s)
491.52
400.00
Encryption
Data is encrypted automatically. Select an encryption key management solution.
Google-managed key
No configuration required
Customer-managed key
Manage via Google Cloud Key Management Service
Customer-supplied key
Manage outside of Google Cloud
This new disk will be added once you create the new instance
Done
Cancel



9. (For use with GCS only) In the Identity and API access section, make sure Service account is set to **Compute Engine default service account**, and under Access scopes, select **Set access for each API**.
10. (For use with GCS only) Scroll down to the Storage setting, and set it to one of the following options:
    - To use Google Cloud Storage (GCS) as persistent storage for your instance, select **Read Write**.
    - To only use GCS to load data into ThoughtSpot, select **Read Only**.
11. Customize the network settings as needed, preferably use your default VPC settings.
12. Repeat these steps to create the necessary number of such VMs.

Prepare the VMs (ThoughtSpot Systems Reliability Team)

> ⚠ **Important:** This procedure is typically done by a ThoughtSpot Systems Reliability Engineer (SRE). Please consult with your ThoughtSpot Customer Service or Support Engineer on these steps.

Before we can install a ThoughtSpot cluster, an administrator must log into each VM through SSH as user “admin”, and complete the following preparation steps:

1. Run `sudo /usr/local/scaligent/bin/prepare_disks.sh` on every machine.
2. Configure each VM based on the site-survey.

Launch the cluster

Upload the TS tarball to one of the VMs and proceed with the normal cluster creation process, using tscli cluster create.

If you are going to use GCS as your persistent storage, you must enable it when running this command, using the **enable_cloud_storage** flag. Example: `tscli cluster create 6.0-167.tar.gz --enable_cloud_storage=gcs`

Open the required network ports

To determine which network ports to open for a functional ThoughtSpot cluster, see Network policies.

Related information

Connecting to Google Cloud Storage buckets

Loading data from a GCP GCS bucket

VMware configuration overview

Congratulations on purchasing the ThoughtSpot instance. This section is an overview of the ThoughtSpot AI-Driven analytics platform hosted on the VMware vSphere Hypervisor (ESXi) 6.5 environment.

About ThoughtSpot on VMware

The VMware virtualization platform provides highly scalable and efficient memory and CPU resources management that can be used by ThoughtSpot instances. Additionally, the VMware virtualization environment is an easy transition between development and production environments. The following diagram shows the components of a VMware and ThoughtSpot architecture:

Note: This is a generic representation; Only CentOS-based virtual machines are supported with ThoughtSpot.

Your database capacity will determine the number of ThoughtSpot instances and the instance network/storage requirements. In addition, you can scale your ThoughtSpot VMs as your dataset size grows.

Supported configurations

ThoughtSpot Engineering has performed extensive testing of the ThoughtSpot platform on VMware for the best performance, load balancing, scalability, and reliability. Based on this testing, ThoughtSpot recommends the following *minimum specifications* for an individual VMware ESXi host machine:

Per VM user data capacity	CPU/RAM	Data disk
20 GB	16/128 GB	800 GB
100 GB	32/256 GB	800 GB
256 GB	72/512 GB	6 TB
Note: All cores must be hyperthreaded. 200GB SSD boot disk required for all configurations.		

Locally attached storage provides the best performance.

SAN can be used, but must comply with the following requirements:

- 136 MBps minimum random read bandwidth
- 240 random IOPS (~4ms seek latency)

NAS/NFS is not supported since its latency is so high that it tends to be unreliable.

All virtualization hosts should have VMware vSphere Hypervisor (ESXi) 6.5 installed.

ThoughtSpot provides a VMware template (OVF) together with a VMDK (Virtual Machine Disk) file for configuring a VM. VMDK is a file format that describes containers for virtual hard disk drives to be used in virtual machines like VMware Workstation or VirtualBox. OVF is a platform-independent, efficient, extensible, and open packaging distribution format for virtual machines.

The ThoughtSpot VM configuration uses thin provisioning and sets the recommended reserved memory, among other important specifications. You can obtain these files from your ThoughtSpot Customer Success Engineer.

Questions or comments?

We hope your experience with ThoughtSpot is excellent. Please let us know how it goes, and what we can do to make it better. You can contact ThoughtSpot by filing a support ticket, email or phone.

Set up ThoughtSpot in VMware

Summary: Learn how to install a ThoughtSpot cluster in a VMware environment.

This page explains how to install a ThoughtSpot cluster in a VMware VSphere Hypervisor (ESXi) 6.5 environment. For each hardware node, you must:

- Complete the prerequisites
- Use the ThoughtSpot Open Virtualization Format (OVF) file to create a virtual machine (VM)
- Add hard disks to the VM

Prerequisites

This installation process assumes you have already acquired your host machines. You can install on a cluster with any number of nodes. A one node cluster is suitable for a sandbox environment but is insufficient for a production environment. You need at least three nodes for high availability (HA), but there is no limit on the number of nodes.

1. Make sure you have installed the Hypervisor on each of your nodes.

   The VM template, by default, captures a 72-core configuration. If your physical host has more than 72 cores, you may want to edit VM to have (`n-2`) cores (for a physical host with n cores) to fully take advantage of computing power of the physical host. Extra cores help performance.

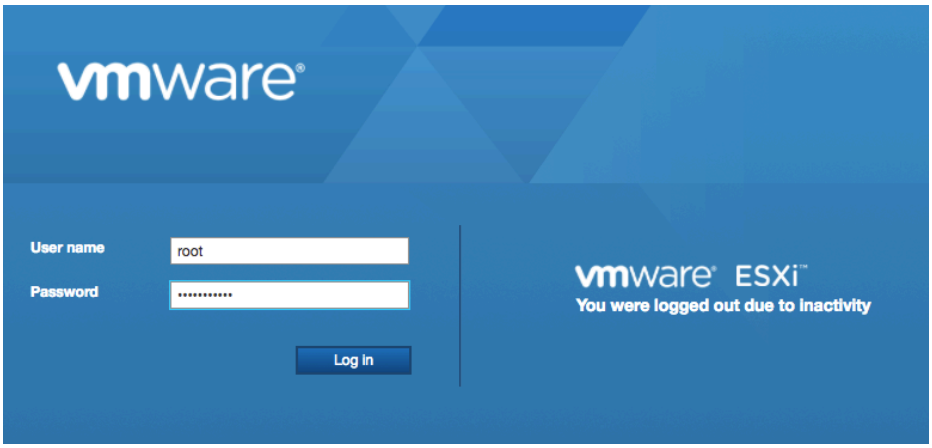
   You should aim to allocate 490 GB or more RAM.

2. Create datastores for all solid-state drive (SSD) and hard drive devices.

Use the OVF to Create a VM

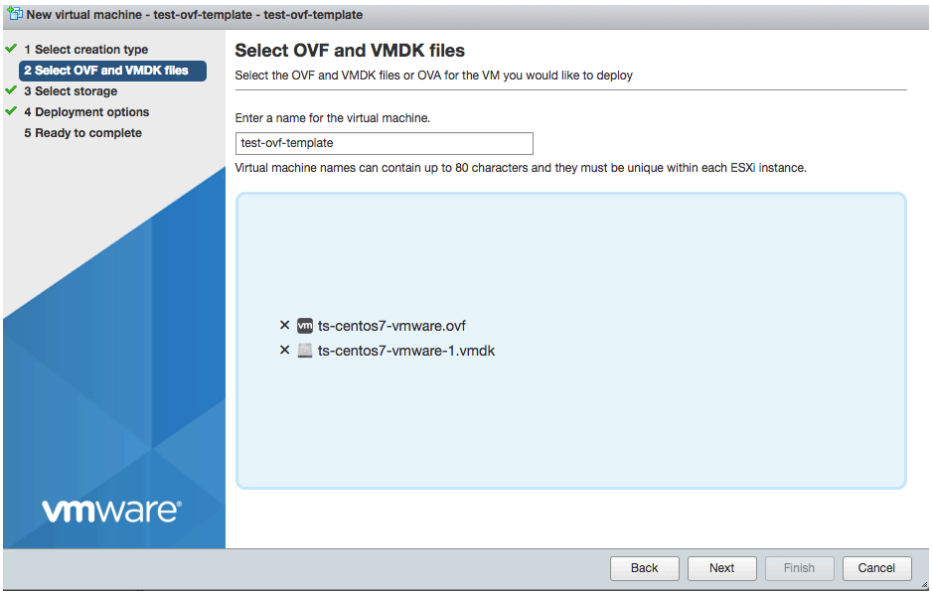
1. **Download** the `ThoughtSpot OVF` to a location on an accessible disk.

2. Log into the ESXi web portal.



3. Select **Virtual Machines > Create/Register VM**.

   The system displays the dialog for selecting an OVF template.



4. Choose the OVF template and click **Next**.

   The system prompts you to select a storage.

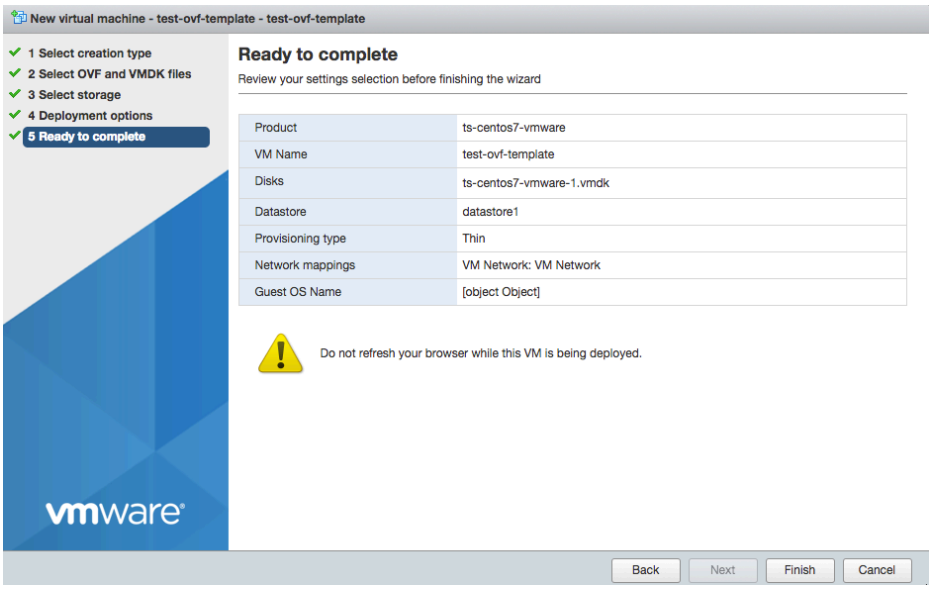
5. Choose the SSD as the destination and click **Next**.

   The system displays the **Deployment Options** dialog.

6. Enter the options and click **Next**.

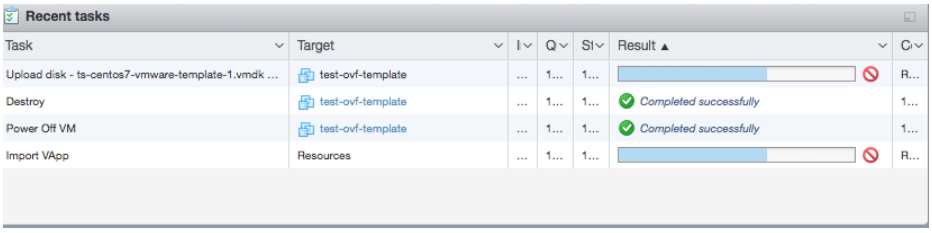
Setting	Value
Network mappings	Select the correct network for your installation.
Disk provisioning	Choose Thin.
Power on automatically	Check this box.

7. Review your selection and click **Finish**.



8. Wait for the template to be loaded.

   Depending on your network speed, loading can take several minutes.



9. Make sure that VM is powered off.

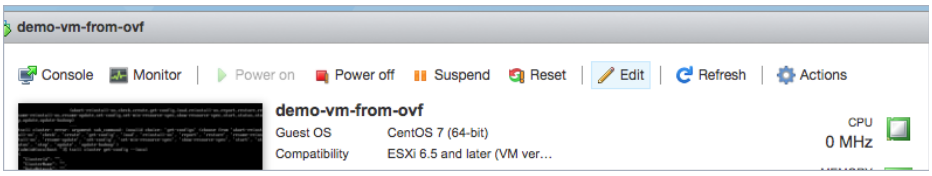
Add hard disks to the VM

Use Case	HDFS Disk Requirements
POC	2 x 1 TB on HDD
Production	3 x 2 TB on HDD

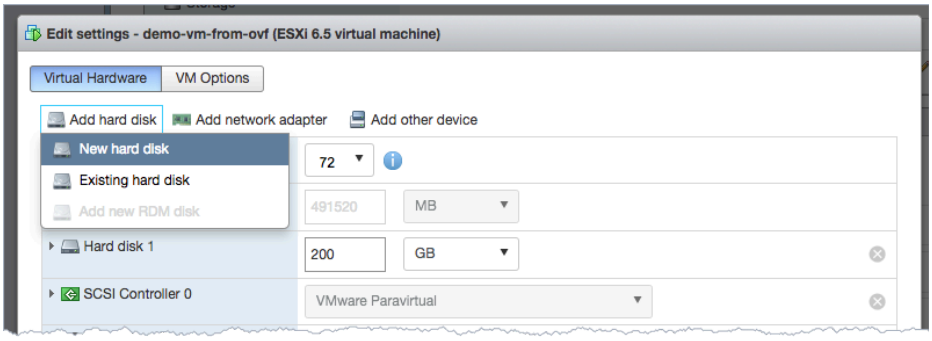
For a proof of concept (POC), follow these steps to create two 1TB HDFS disks on HDD storage, as shown here (2 x 1TB).

For production deployments, ThoughtSpot requires you to have three 2TB HDFS disks on HDD (3 x 2TB). For this use case, follow these same steps to create the additional, larger capacity disks.

1. Edit the VM you just created.



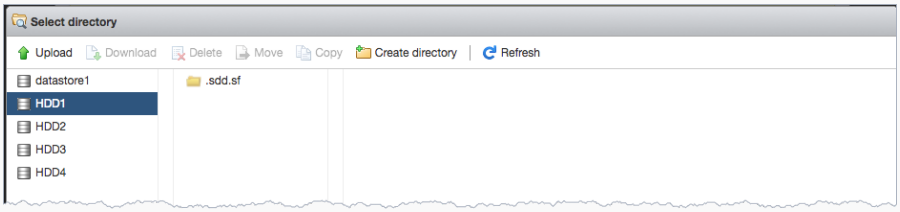
2. Select **Add hard disk > New hard disk**.



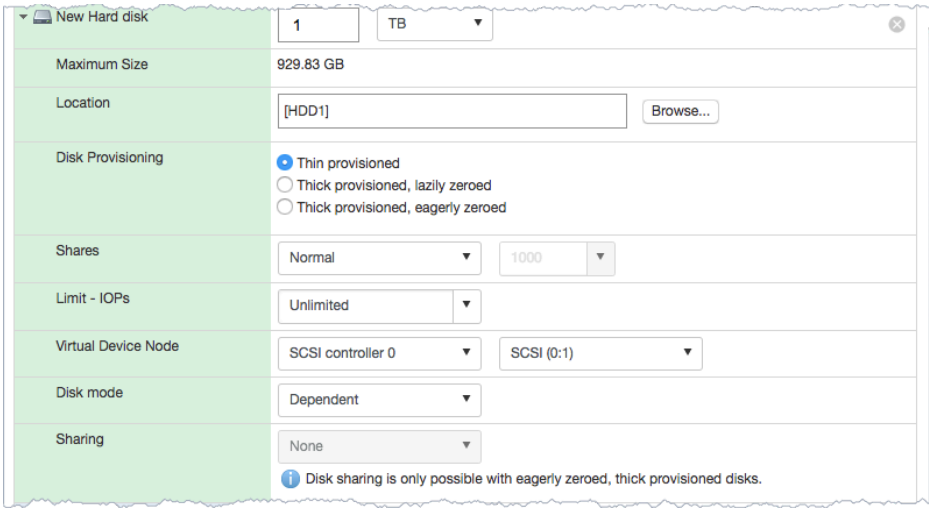
   The new disk appears as a new row under the only existing SSD row.

3. Click the **New Hard disk** to expand the detailed configuration options.

4. For a proof of concept, set the options as follows. (For production deployments, set the size to 2TB.)

Setting	Value
size	1 TB
Location	Use the **Browse** button to select the hard disk store. 
Thin provisioned	Check this box.

You should see something similar to the following:



5. Save your changes.
6. Repeat steps 1-5 to create more hard disks.
7. Power on the VM
8. After the VM is online, run the following command to prepare the HDFS disks:

```
sudo /usr/local/scaligent/bin/prepare_disks.sh
```

Next steps

There is no network at this point on your VMs. As a prerequisite:

1. Verify that Network Adapter type is set to VMware vmxnet3 (Recommended).
2. Verify that all ESXi hosts in your VMware farm for ThoughtSpot have been trunked to the VLAN assigned to your ThoughtSpot VMs.
3. Verify that the console of all ThoughtSpot VMs is accessible in VMware vCenter Server.

After you finish, go to the ThoughtSpot Support website and use the support ticket for installation tasks. If necessary, create a new ticket.

Network policies

Summary: Lists the required and optional ports for an installation.

For regular operations and for debugging, there are some ports you must keep open to network traffic from end users. Another, larger list of ports must be kept open for network traffic between the nodes in the cluster.

Required ports for operations and debugging

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

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

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

Network Ports

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

Required ports for intracluster operation

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

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

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

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

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

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

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

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

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

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

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

Required ports for inbound and outbound cluster access

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

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

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

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

Required ports for IPMI (Intelligent Platform Management Interface)

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

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

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

Contact support

You can contact ThoughtSpot by phone, mail, email, or by filing a support ticket.

File a support ticket

If you encounter a technical issue, file a support ticket using the Support Portal ticket filing system at:

http://support.thoughtspot.com/

Please provide as much detail as possible about your issue, to help us resolve it quickly.

You need a Support Portal login to file a ticket. Please contact ThoughtSpot to get an account, if necessary.

Address

ThoughtSpot, Inc. 910 Hermosa Ct Sunnyvale, CA 94085

Phone numbers

Phone Number	Description
1-800-508-7008 ext 1	ThoughtSpot Support
1-800-508-7008	Toll-free number for ThoughtSpot headquarters.

Email

Reason for contacting us	Email
Sales inquiries	sales@thoughtspot.com
Customer support and software update inquiries	support@thoughtspot.com

Reason for contacting us	Email
Other inquiries	hello@thoughtspot.com

Introduction to Data Integration

This guide explains how to integrate ThoughtSpot with other data sources for loading data. It also includes information on installing and using the ThoughtSpot clients (ODBC, JDBC, and more).

ThoughtSpot Clients

ThoughtSpot provides certified clients to help you load data easily from your ETL tool or another database. These include ODBC and JDBC drivers.

You can obtain the ThoughtSpot client downloads from the Help Center. Always use the version of the ThoughtSpot clients that corresponds with the version of ThoughtSpot that you are running. When upgrading, make sure to upgrade your clients as well.

> ⚠ **Important:** The ETL tool must add a data transformation step if the source column data type does not exactly match the target's, ThoughtSpot's, column data type. The driver does not do any implicit conversions.

Methods for loading data

There are several ways to load data into ThoughtSpot, depending on your goals and where the data is located. Always consider your requirements for recurring loads when planning how best to bring the data into ThoughtSpot.

Here are the options, with information on where to find the documentation for each method:

Method	Description
ThoughtSpot Loader (tsload)	ThoughtSpot Loader is a command line tool to load CSV files into an existing database schema in ThoughtSpot. This is the fastest way to load extremely large amounts of data, and it can be run in parallel. You can also use this method to script recurring loads. See the ThoughtSpot Administrator Guide for details.

Method	Description
User Data Import	Users can upload a spreadsheet through the web interface with User Data Import. This is useful for giving everyone easy access to loading small amounts of their own data. See the ThoughtSpot Administrator Guide for details.
ODBC	ThoughtSpot provides an ODBC (Open Database Connectivity) driver to enable transferring data from your ETL tool into ThoughtSpot.
JDBC	ThoughtSpot provides a JDBC (Java Database Connectivity) driver to enable transferring data from your ETL tool into ThoughtSpot.
Microsoft SSIS (SQL Server Integration Services)	You can use the ODBC driver to connect to SSIS and import data into ThoughtSpot. Basic instructions are included in this guide.
Connect to Pentaho	You can use the JDBC driver to connect to Pentaho and import data into ThoughtSpot. Basic instructions are included in this guide.

Where to go next

- **Server-side prerequisites for using JDBC/ODBC to import data**

  You must follow setup prerequisites for importing data using JDBC/ODBC.
- **About the ODBC Driver**

  You can use the ThoughtSpot ODBC driver to bring data into ThoughtSpot from your ETL tool or database.
- **About the JDBC Driver**

  Java Database Connectivity (JDBC) is a Java standard API that allows applications to interact with databases in a standard manner. ThoughtSpot has JDBC support through a JDBC driver that we provide.

Embrace overview

Summary: Using Embrace, you can perform live query on external databases.

If your company stores source data externally in data warehouses, you can use ThoughtSpot Embrace to directly query that data and use ThoughtSpot's analysis and visualization features, without moving the data into ThoughtSpot. If you decide later you want to copy your data into ThoughtSpot, you can also do that with Embrace.

Embrace supports the following external databases:

- Snowflake
- Amazon Redshift (*in beta*)

To enable Embrace, contact ThoughtSpot support.

How it works

You create a connection to the external database, choosing the columns from each table that you want to explore in your live query. Primary key and foreign key relationships are imported along with the primary and foreign key tables. If there are any joins in the tables of your connection, they are also imported. After your connection is complete, it becomes a **linked** data source in ThoughtSpot that allows you to query the external database directly. It's easy to apply transformations and filter the data also.

Key benefits

- Set up and deploy ThoughtSpot faster by connecting directly to the external database.
- Eliminate the need to move data into ThoughtSpot for analysis.
- Centralize data management and governance in the external database.
- Save significant time and money by avoiding ETL pipelines.
- Set up and schedule sync of data into ThoughtSpot.
- Connect to multiple external databases.

Embrace modes

Embrace has two operating modes:

- **Linked**: ThoughtSpot queries your data in the external database.
- **Synced**: ThoughtSpot queries a copy of your data stored in ThoughtSpot.

When you create your connection to an external database, by default, it is a **Linked** connection. If you want to copy the external data into ThoughtSpot, you must sync the data. The features available with Linked and Synced tables are slightly different.

Features in Embrace modes

Feature	Linked Tables	Synced Tables
Simple Search	Yes	Yes
Complex searches like Versus, Inline Subquerying, Growth	Yes	Yes
Search Suggestions for column names	Yes	Yes
Search Suggestions for column values	Yes	Yes
Headlines at the bottom that summarize tables	Yes	Yes
All Chart Types & Configurations	Yes	Yes
SpotIQ Instant Insights	No	Yes
SpotIQ pre-computed insights	No	Yes
Table and Column Remapping	Yes	N/A
Custom Calendar	No	Yes
Materialized Views	No	Yes
Indexing of table columns	Yes	Yes

Next steps

- **Add a connection**

Create the connection between ThoughtSpot and tables in an external database.

- **Sync** Set your connection to copy tables from the external database into ThoughtSpot.
- **Modify a connection**

  Edit, remap or delete a connection to tables in an external database.
- **Connectors reference**

  Source cloud data connectors, and their connection credentials, supported by Embrace.

JDBC and ODBC setup prerequisites

Before you can use JDBC or ODBC to import data into ThoughtSpot, you must do the following server-side configuration:

1. Open up the ThoughtSpot firewall to allow incoming requests to Simba server.

   ```
   tscli firewall open-ports --ports 12345
   ```

2. Confirm that the `simba_server` process is up. Output of the command below should contain exactly one line, as shown below.

   ```
   ps -ef | grep simba_server | grep -v grep
   admin    26679 25672  0 Jul13 ?        00:01:49 simba_se
   rver_main --logbufsecs=0
   ```

   For assistance, contact ThoughtSpot Support.

Overview of the ODBC Driver

Summary: Use the ODBC driver to bring data in from your ETL tool or database.

ThoughtSpot comes packaged with an ODBC (Open Database Connectivity) driver, so that you can transfer data between ThoughtSpot and other databases. Basic knowledge of ODBC data source administration is helpful when setting up ODBC.

Supported operating systems for the ODBC driver are:

- Microsoft Windows 32-bit
- Microsoft Windows 64-bit
- Linux 32-bit
- Linux 64-bit

Version compatibility and connection parameters

To ensure compatibility, always use the ODBC driver with the same version number as the ThoughtSpot instance to which you are connecting. You can make a secure ODBC connection to the ThoughtSpot database by configuring a user and password combination with the driver. For detailed information about connection parameters, see the ODBC and JDBC configuration properties

Supported Data Types

The ODBC driver supports these data types:

- `INT`
- `BIGINT`
- `BOOLEAN`
- `DOUBLE`
- `FLOAT`
- `DATE`
- `TIME`
- `TIMESTAMP`
- `DATETIME`

- `CHAR`
- `VARCHAR`

Source and target data compatibility

By default, ThoughtSpot takes a permissive approach to data type compatibility between source and target data in ODBC. In this mode, ThoughtSpot *assumes* that the incoming data matches exactly with the target data types and loads the table as is.

Alternatively, you can explicitly require that ThoughtSpot match the source data types exactly and, if it can't find a match, it returns an error and the data load fails. In this mode, for example, if the target ThoughtSpot data type for a column is INT, the source data type for that column must be INT in order for the data load to succeed.

By toggling ***strict*** and ***permissive*** `true` and `false` options, you can configure settings along a scale of behavior between the permissive, automatic approach and the strictness of the "must match" approach.

		Strictness	
		true	false
Permissiveness	true	Data types are inferred and automatically converted. ThoughtSpot returns an error in cases where the data conversion is not possible. Data load fails in its entirety if any data contains mismatches. You must correct the problem in the source data and try the load again.	Data types are inferred and automatically converted. No error is thrown even if source and target data types don't match. Data load continues even when the source and target data types don't match. This means your data load may contain data types that you do not intend or that are not helpful. You are responsible for checking and validating the data in this case.

false	The source and target data types must match. If any data contains mismatches, ThoughtSpot returns an error to the client a data load fails in its entirety. You must correct the problem in the source data and try the load again. This is the strictest configuration.	No data types are inferred and conversion does not check for matches. This is the most permissive configuration.

Your customer support engineer can assist you in configuring custom ODBC behavior. Regardless of the configuration you choose, you must validate that the results of data loading *as they appear* in ThoughtSpot are what you require.

Data type conversion matrix

The following table describes the conversion matrix between SQL data types and ThoughtSpot data types.

Source SQL Data Types	BOOL	INT32	INT64	DOUBLE	FLOAT	CHAR	DATE	TIME	DATETIME
SQL_BIT	Y	Y	Y	Y	Y	Y	–	–	–
SQL_TINYINT	Y	Y	Y	Y	Y	Y	–	–	–
SQL_SMALLINT	Y	Y	Y	Y	Y	Y	–	–	–
SQL_INTEGER	Y	Y	Y	Y	Y	Y	–	–	–
SQL_BIGINT	Y	Y	Y	Y	Y	Y	–	–	–
SQL_CHAR	Y	Y	Y	Y	Y	Y	Y	Y	Y
SQL_VARCHAR	Y	Y	Y	Y	Y	Y	Y	Y	Y
SQL_LONGVARCHAR	Y	Y	Y	Y	Y	Y	Y	Y	Y
SQL_BINARY	–	–	–	–	–	Y	–	–	–
SQL_VARBINARY	–	–	–	–	–	Y	–	–	–
SQL_LONGVARBINARY	–	–	–	–	–	Y	–	–	–

Source SQL Data Types	BOOL	INT32	INT64	DOUBLE	FLOAT	CHAR	DATE	TIME	DATETIME
SQL_DOUBLE	Y	Y	Y	Y	Y	Y	–	–	–
SQL_REAL	Y	Y	Y	Y	Y	Y	–	–	–
SQL_FLOAT	Y	Y	Y	Y	Y	Y	–	–	–
SQL_NUMERIC	Y	Y	Y	Y	Y	Y	–	–	–
SQL_GUID	–	–	–	–	–	Y	–	–	–
SQL_INTERVAL_MINUTE_TO_SECOND	–	–	–	–	–	Y	–	–	–
SQL_INTERVAL_HOUR_TO_SECOND	–	–	–	–	–	Y	–	–	–
SQL_INTERVAL_HOUR_TO_MINUTE	–	–	–	–	–	Y	–	–	–
SQL_INTERVAL_DAY_TO_SECOND	–	–	–	–	–	Y	–	–	–
SQL_INTERVAL_DAY_TO_MINUTE	–	–	–	–	–	Y	–	–	–
SQL_INTERVAL_DAY_TO_HOUR	–	–	–	–	–	Y	–	–	–
SQL_INTERVAL_YEAR	–	Y	Y	–	–	Y	–	–	–
SQL_INTERVAL_MONTH	–	Y	Y	–	–	Y	–	–	–
SQL_INTERVAL_DAY	–	Y	Y	–	–	Y	–	–	–
SQL_INTERVAL_HOUR	–	Y	Y	–	–	Y	–	–	–
SQL_INTERVAL_MINUTE	–	Y	Y	–	–	Y	–	–	–
SQL_INTERVAL_SECOND	–	Y	Y	–	–	Y	–	–	–
SQL_TYPE_TIME	–	–	–	–	–	Y	–	Y	Y
SQL_TYPE_DATE	–	–	–	–	–	Y	Y	–	Y
SQL_TYPE_TIMESTAMP	–	–	–	–	–	Y	Y	Y	Y

If a conversion is not possible, an error is returned to the client to indicate conversion failure. The ETL tool must add a data transformation step if the source column data type does not exactly match the target's ThoughtSpot column data type. The driver does not do any implicit conversions.

Install the ODBC driver on Windows

Summary: Use this procedure to obtain the Microsoft Windows ODBC driver and install it.

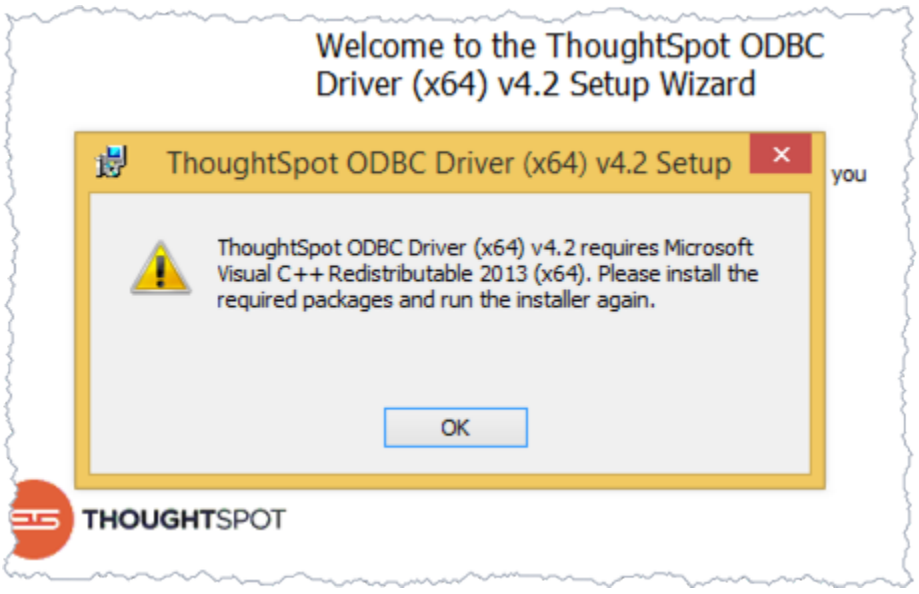
ThoughtSpot's ODBC connection relies on the SimbaEngine X SDK to connect through ODBC or JDBC to ThoughtSpot's remote data stores. The instructions on this page explain how to configure the Simba ODBC driver on a Windows workstation.

Make sure you have read the overview material in the ODBC driver overview. This workstation is the same machine where you plan to run your ETL activities.

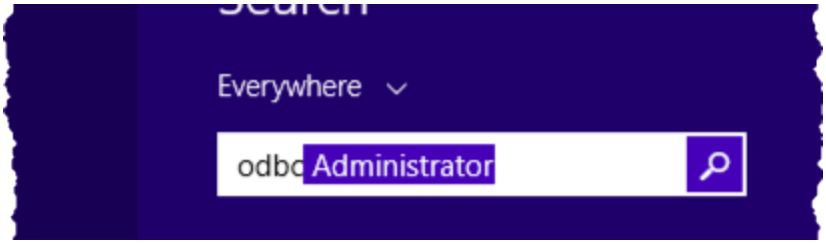
Prerequisites

These instructions include directions to use the `ssh` command. Make sure your Windows workstation is equipped with a tool such as Putty for making `ssh` connections to your ThoughtSpot server.

The ODBC driver for Windows requires Visual C++ Redistributable for Visual Studio 2013. You are prompted to install it during installation of the driver if it isn't already installed.



To check if this Microsoft tool is already installed, search for it on your workstation.



If it isn't installed, make sure you download and install it before continuing.

Check the ThoughtSpot IP and the simba_server status

Before you begin, you need to know the IP address or DNS name of the server you intend to connect your server to.

1. SSH as `admin` or the `thoughtspot` user to your ThoughtSpot node.
2. Verify the node IP(s).

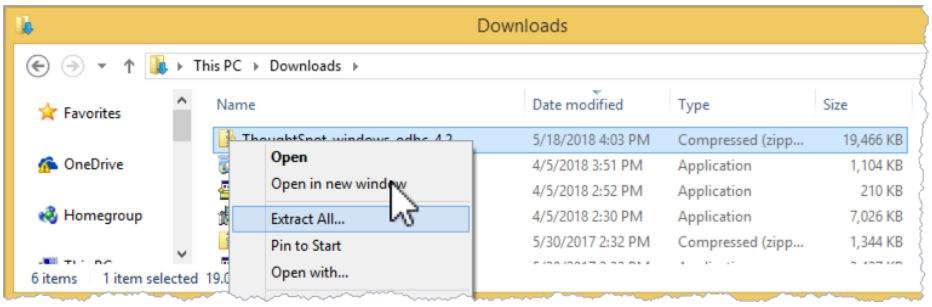
```
$ tscli node ls
172.18.231.17
172.18.231.18
```

3. Make a note of each IP; there may be more than one.
4. Configure the ThoughtSpot firewall to allow connections from your ETL client, by running the following command on any ThoughtSpot node: `tscli firewall open-ports --ports 12345`
5. Exit or close the shell.

Download the driver

On the workstation where you want to connect from, do the following:

1. Navigate to the **Downloads** page.
2. Download the **ODBC Driver for Windows**.
3. Unzip the file you downloaded at a convenient location on your workstation.



4. Take a moment to examine the contents of the new directory.

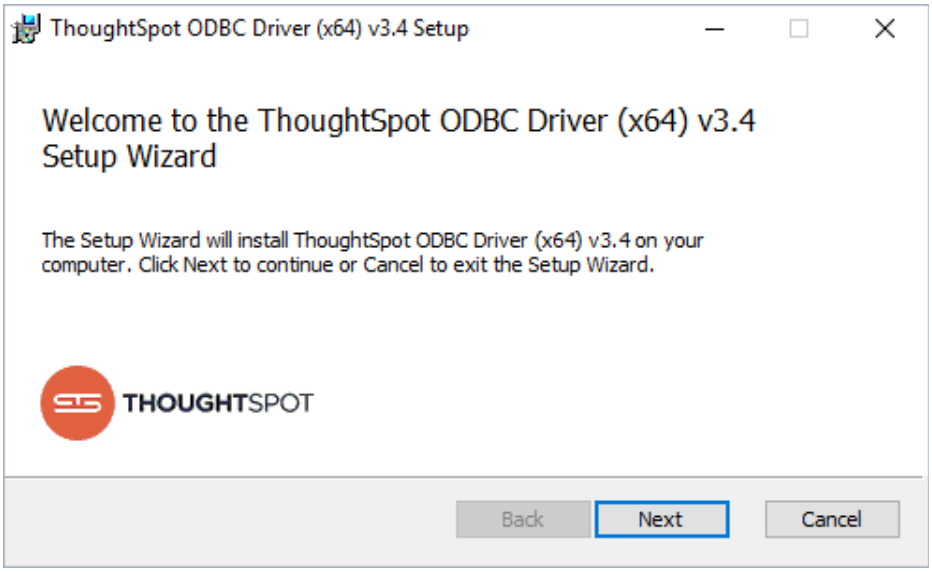
   There are two different Windows ODBC installers included in the file you downloaded.

   - ThoughtSpotODBC (x86).msi for Windows 32-bit
   - ThoughtSpotODBC (x64).msi for Windows 64-bit

Install the driver and supporting software

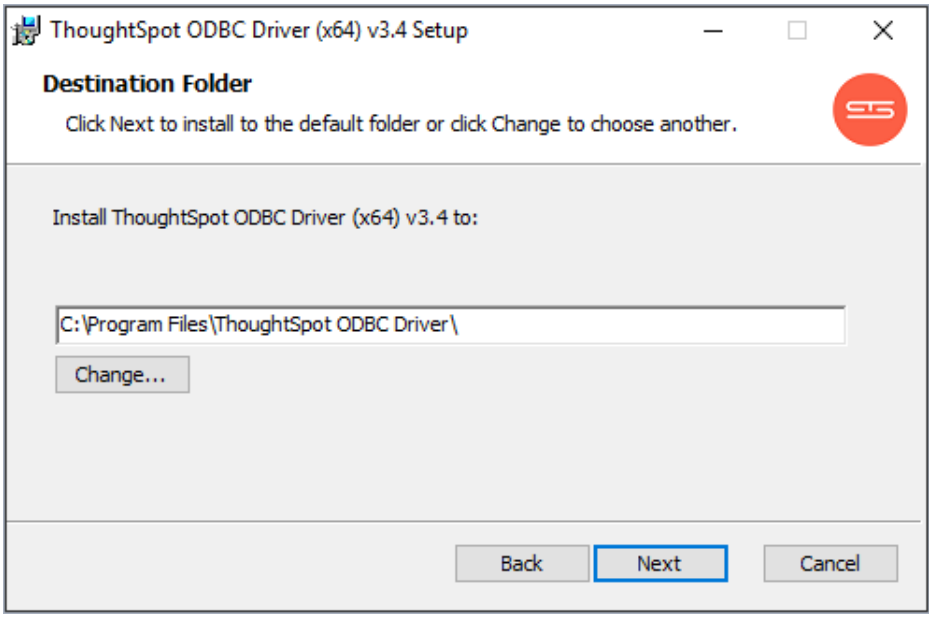
The installation process installs the Simba driver and adds the ODBC Administrator software to your workstation. You use this software to configure the driver.

1. Launch the installer for your version of Windows.
2. Click **Next** to continue.

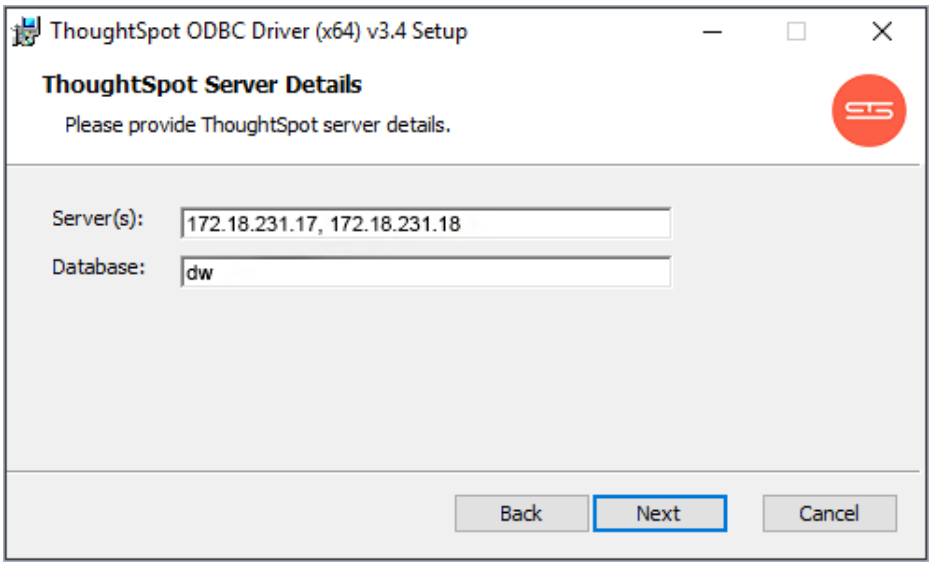


3. Accept the End User License Agreement (EULA), and click **Next**.

4. Specify the destination folder where the driver will be installed.



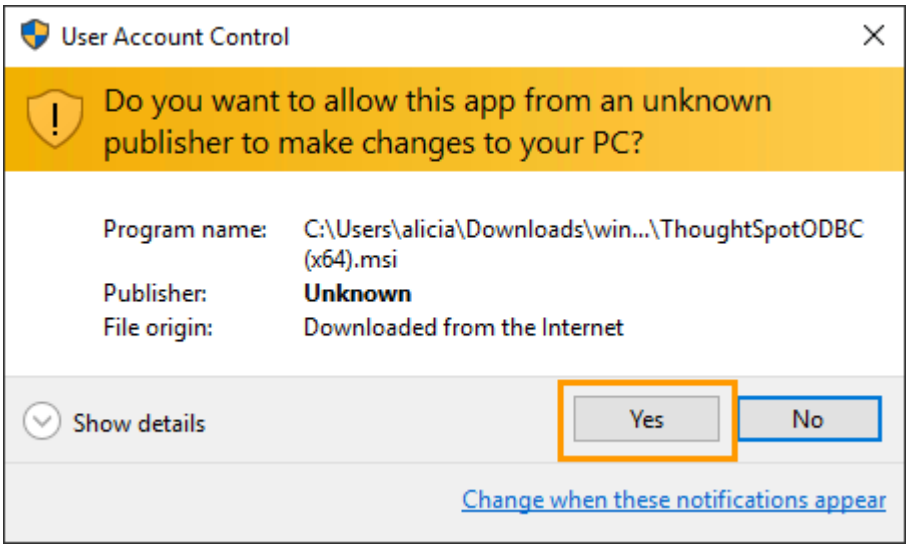
5. Enter the ThoughtSpot server details, and click **Next**.



- For **Server(s)**, provide a comma separated list of the IP addresses of each node on the ThoughtSpot instance.
- For **Database**, optionally specify the database to use. If you skip this entry, you must provide the database each time you connect using ODBC.

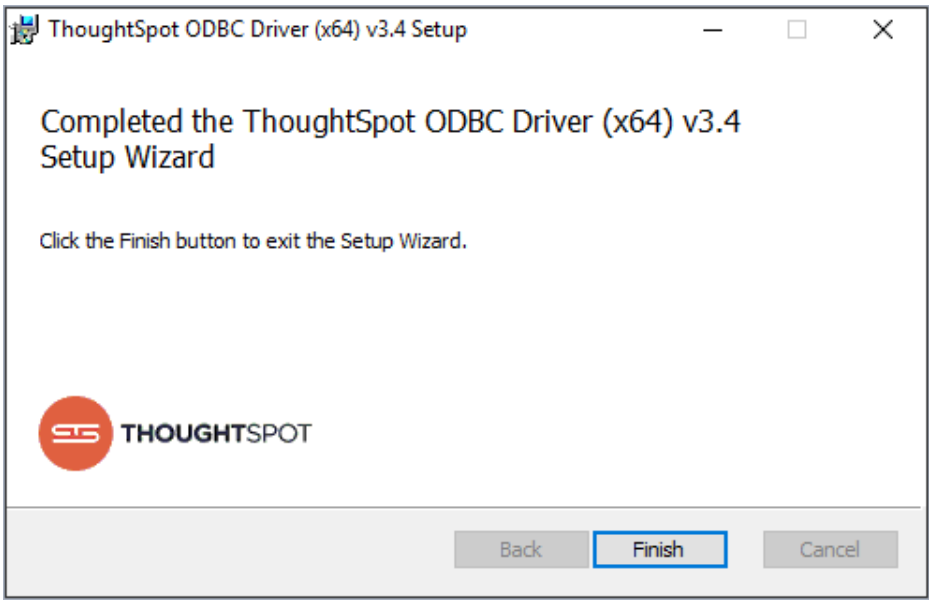
6. Confirm that the install can begin by clicking **Install**.

7. You may see a security warning.



8. Click **Yes** to continue.

   A confirmation message appears when the installation is complete.



9. Click **Finish**.

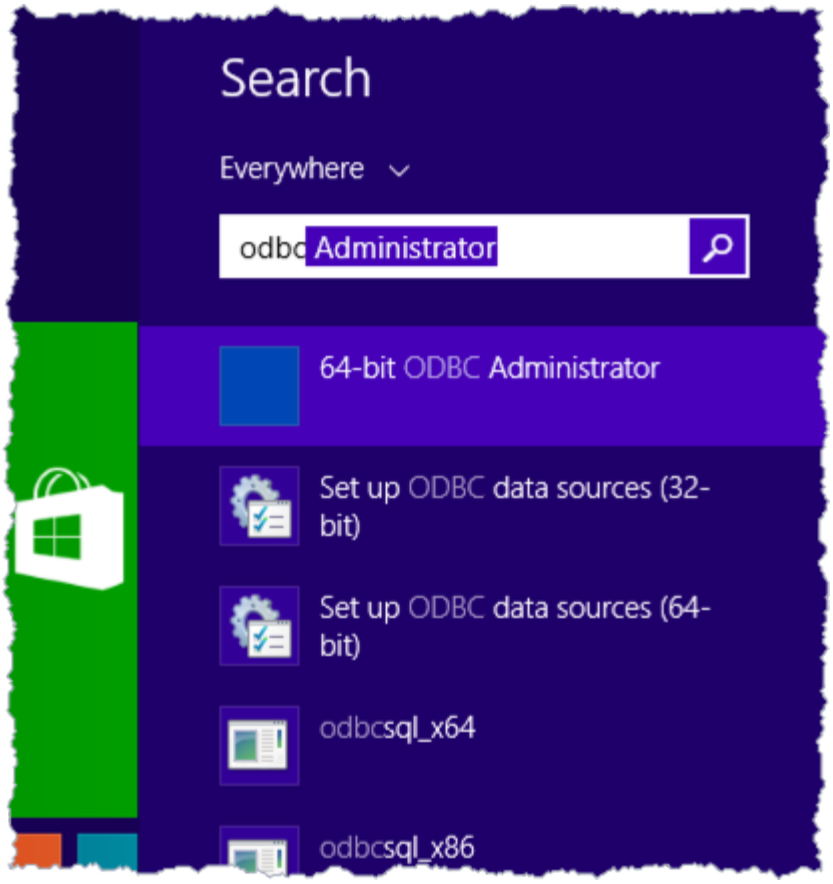
Configure the driver and test your connection

After installation completes, use the ODBC Administrator to configure the ODBC connection on your Windows workstation. For example, you may want to add a default schema or change the server IP address or the default database.

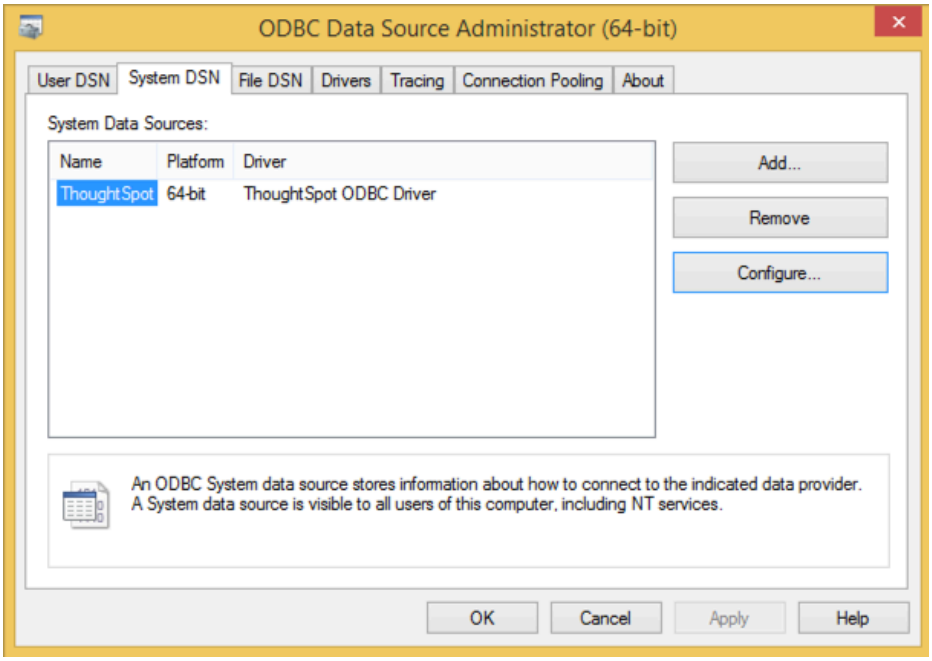
It is recommended to add a default schema. If you don't specify a default schema, you must supply it every time you use the ODBC driver.

At this point, you can test your ODBC connection to ThoughtSpot. It is important to recall that the username/password you use belongs to a ThoughtSpot application user. Typically, this user is a user with data management or administrative privileges on the application.

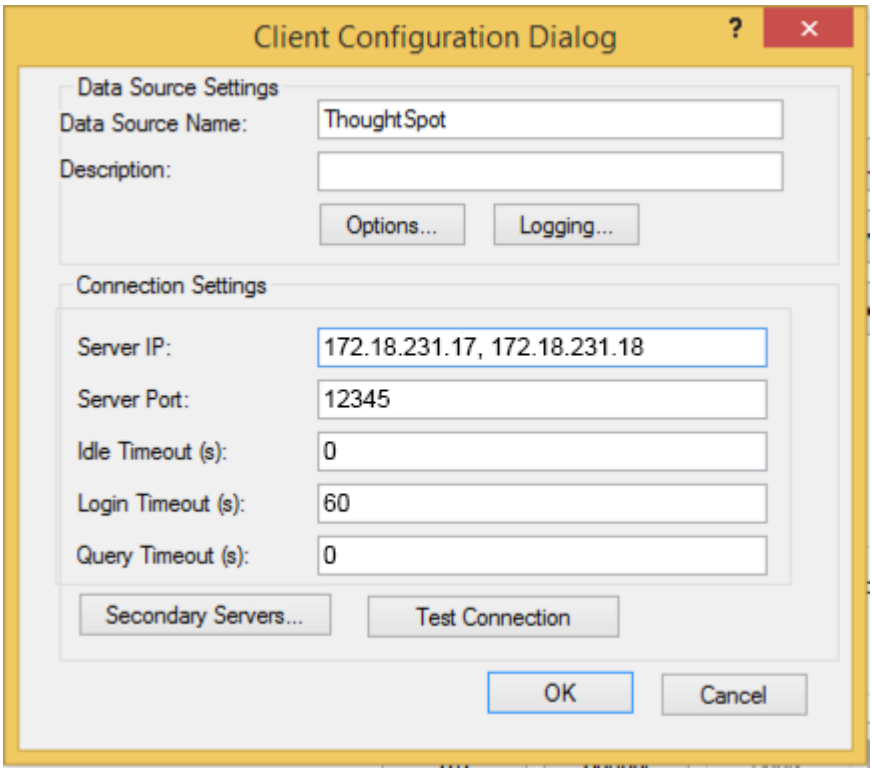
1. Before trying the ODBC connection, confirm a username/password that can log into the ThoughtSpot applications.
2. Click the **Data** tab, and confirm the user's privileges.
3. Return to your workstation.
4. Locate and open the **ODBC Data Source Administrator (64-bit)** application.



5. Click the **System DSN** tab.



6. Select **ThoughtSpot** and click **Configure…**



7. Click **Options…**

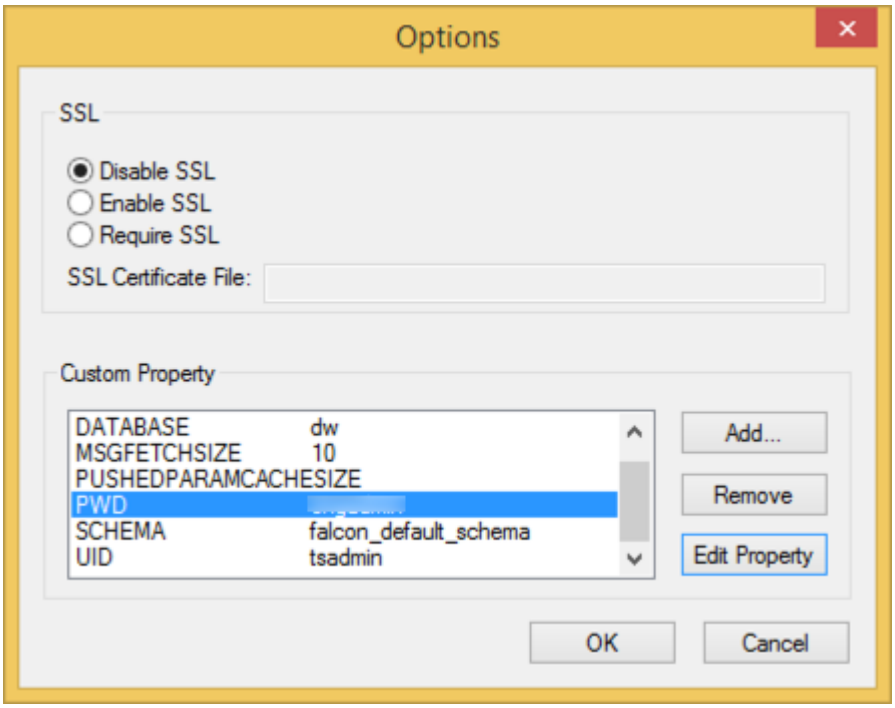
8. Ensure you have the following **Custom Property** values set:

Custom Property	Value
`SCHEMA`	`falcon_default_schema` is the default
`UID`	The username of a user with data management privilege.
`PWD`	The password for the username you specify.

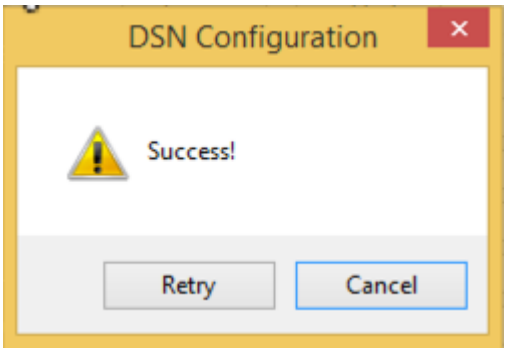
You don't have to use the ThoughtSpot default schema. You can specify your own. We recommend that you define a default schema. Otherwise, you must supply a schema every time you use the ODBC driver. Moreover, without a schema (or if the schema is not present), the ODBC driver returns an error that states that the schema could not be found.

Similarly, adding the `UID` and `PWD` properties are not required. If you don't add them, you are prompted to supply them each time you connect.

When you are done, your options should look similar to the following:



9. When you are done, click **OK** to save your new properties.
10. Click **Test Connection** to test your database connection.



11. Click **Cancel** to close the **DSN Configuration** dialog.
12. Click **OK** to close the **Client Configuration Dialog** the dialog.
13. Click **OK** to close the **ODBC Data Source Administrator (64-bit)** application.

Now, you are ready to begin using the connection you've configured.

Related information

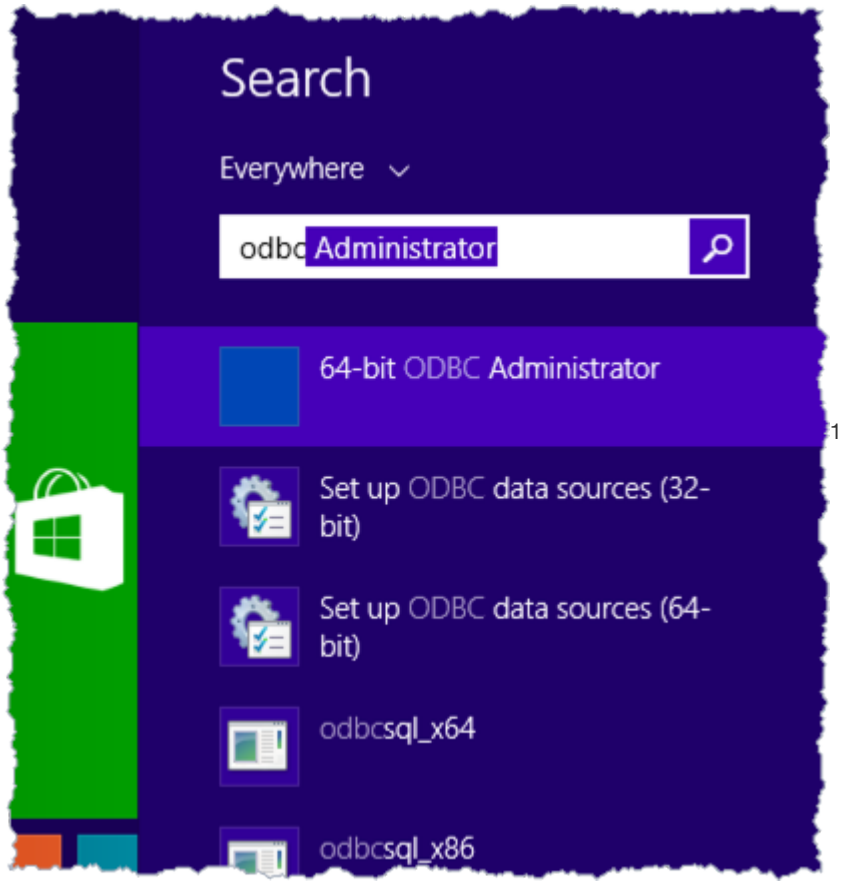
- Enable ODBC logs.
- Configure multiple connections on Windows.

Configure multiple connections on Windows

Summary: You can add multiple ODBC data sources.

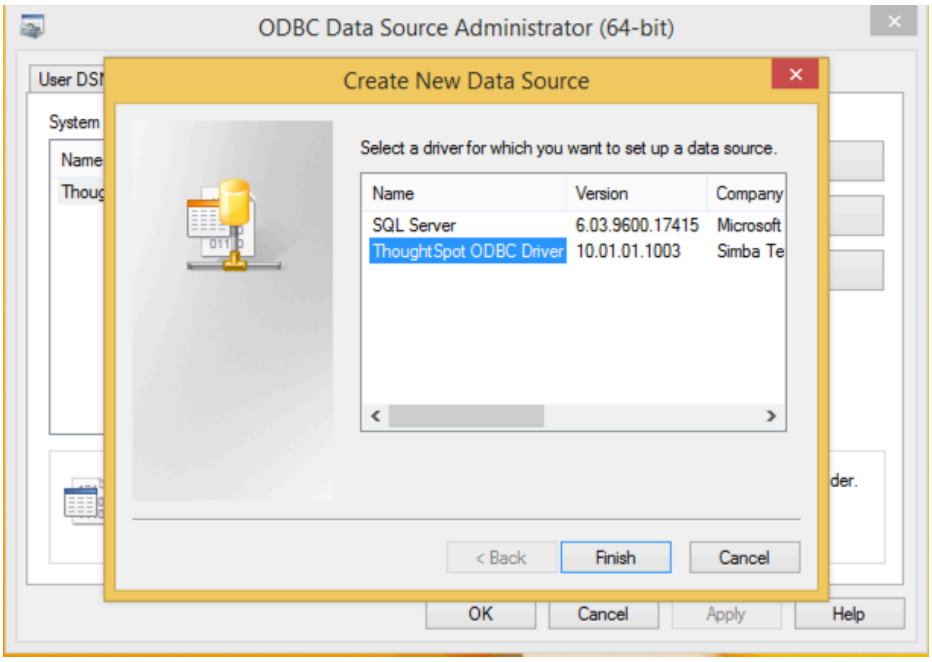
Use this procedure if you want to add an additional data source after creating a single source succeeds. The main reason for needing to set up multiple ThoughtSpot ODBC data sources is that you have a production cluster and a test or development cluster.

1. Locate and open the **ODBC Data Source Administrator (64-bit)** application.



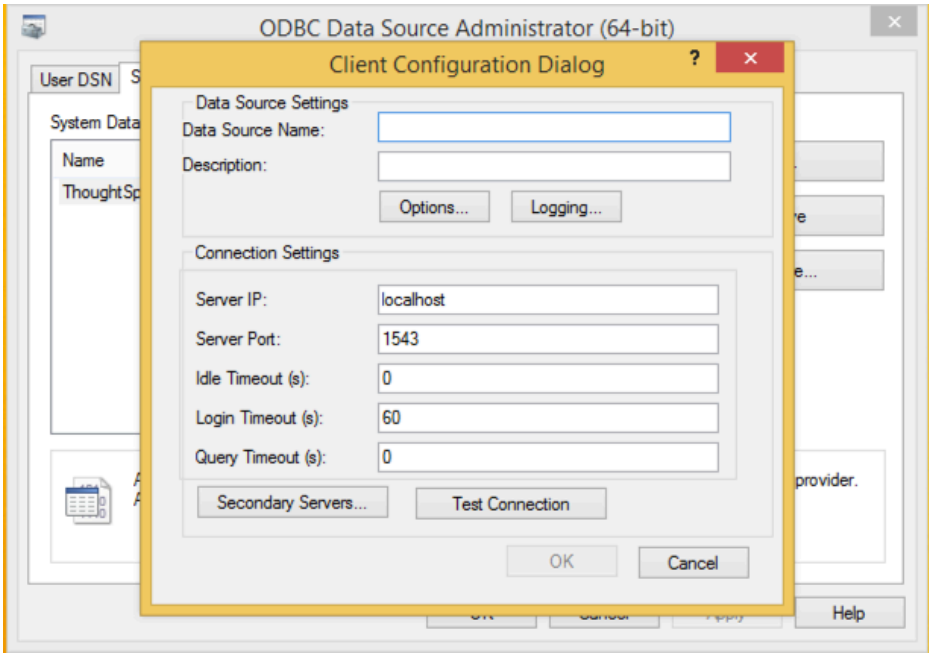
2. Click the **System DSN** tab.
3. Select **Add**.

   The system lists the available drivers.



4. Choose the **ThoughtSpot ODBC Driver** and click **Finish**.

   The system displays the **Client Configuration Dialog** dialog.

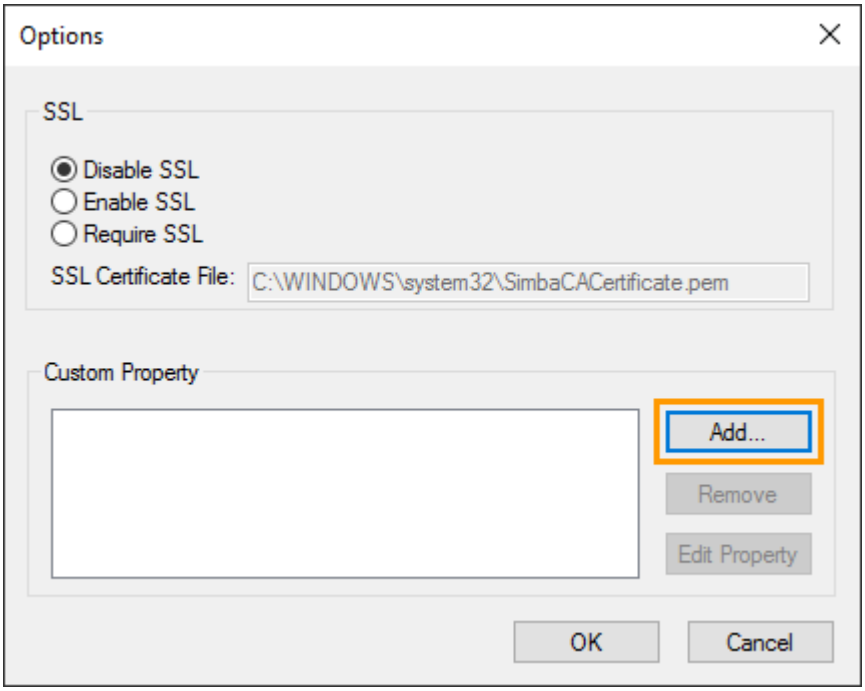


5. Enter the your data source configuration.

Configuration Property	Value
Data Source Name	The name you want to call the data source.
Description	A description of the data source.
Server IP	A list of the IP addresses for each node, separated by commas.
Server Port	12345
Idle Timeout	Time in seconds after which an idle ODBC connection times out.
Login Timeout	Time in seconds after which a login request times out.
Query Timeout	Time in seconds after which a query times out.

6. Configure custom properties by clicking **Options**.

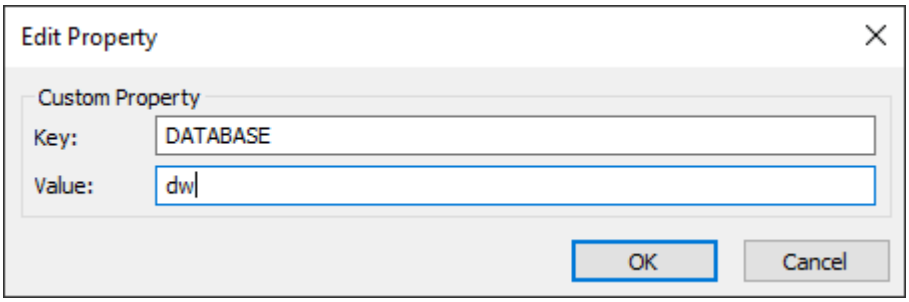
   The system displays the **Options** dialog.



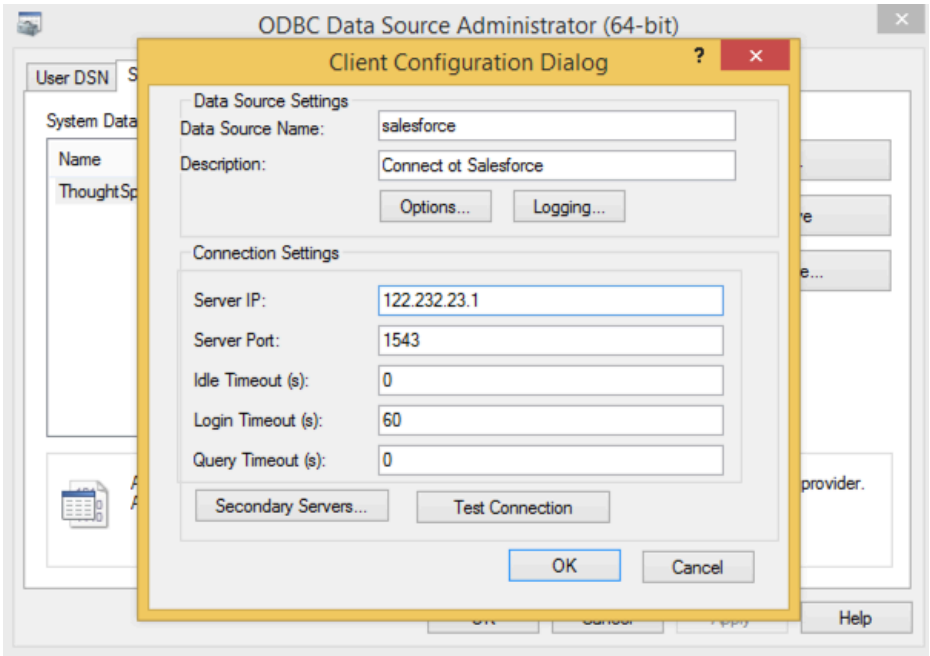
7. Add these properties using the **Add** to enter an option and click **OK** after to save an option.

Option	Value
DATABASE	The default database to connect to.
SCHEMA	The default schema to connect to. Use `falcon_default_schema` if you aren't sure.
CONNECTIONTIMEOUT	Optional. Seconds before an idle connection times out.

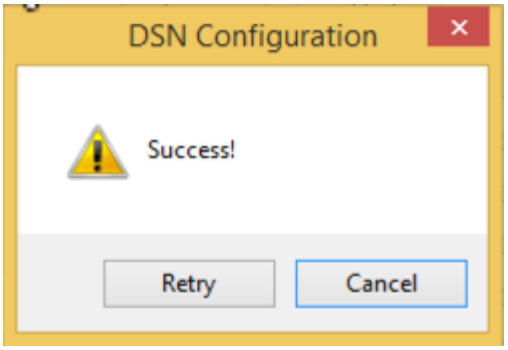
The key must be defined exactly as it appear here, using all capital letters. You can find other supported properties in ODBC and JDBC configuration properties.



8. When you are done, click **OK** to save your new configuration.



9. Click **Test Connection** to test your database connection.



If your test connection fails, enable ODBC logging to troubleshoot.

10. Click **Cancel** to close the **DSN Configuration** dialog.
11. Click **OK** to close the **Client Configuration Dialog** the dialog.
12. Click **OK** to close the **ODBC Data Source Administrator (64-bit)** application

Deploy SSL with ODBC on Windows

You can configure a secure ODBC connection between your ThoughtSpot cluster and a remote Windows Machine. This article explains the SSL resources and ODBC configuration options you need to enable SSL for an ODBC connection.

Prerequisites

Before configuring SSL over the ThoughtSpot ODBC connection, make sure that your system administrator has created and configured your network's Certificate Authority. Additionally, the system administrator should have available both the proper Private Key and Server Certificate.

Configure the ThoughtSpot cluster nodes

> ⚠ **Important:** Portions of this procedure require that you work with your ThoughtSpot Customer Service or Support Engineer.

The SimbaServer Configuration Properties reference includes full details on SSL Configuration Properties.

Before you change your ODBC configuration, decide on a path where you will store the Private Key and Server Certificate, for example, you could decide to use `/home/admin/Simba_SSL/` as the path.

Then, do the following on *every ThoughtSpot node* in your cluster.

1. Create the path on the node.
2. Copy the SSL certificate and private key to this path.
3. Edit the node's `/etc/thoughtspot/simba.ini` file (Simba server configuration) with your favorite editor.
4. Add the following lines:

```
 SslCertfile=/home/admin/Simba_SSL/Server-Certificate.pe
m
 SslKeyfile=/home/admin/Simba_SSL/Private-Key.pem
 UseSsl=Required
```

5. Restart the Simba service.

   You must work with your ThoughtSpot Customer Success or Support Engineer to do this.

Deploy the certificate on your windows workstation

Please note that the SSL settings on the server and client are interdependent.

The SimbaClient for ODBC Configuration Properties reference describes how to set parameters on the client to use SSL (scroll down to useSsl section at the end). The Simba documentation also provides a chart showing configuration properties for SSL where you can see how different combinations of SSL settings on client and server will behave. For example:

- Setting both server and client to `UseSsl=Enabled` provides the ability for clients to connect with or without SSL.
- Setting both server and client to `UseSsl=Required` requires that all clients use SSL.

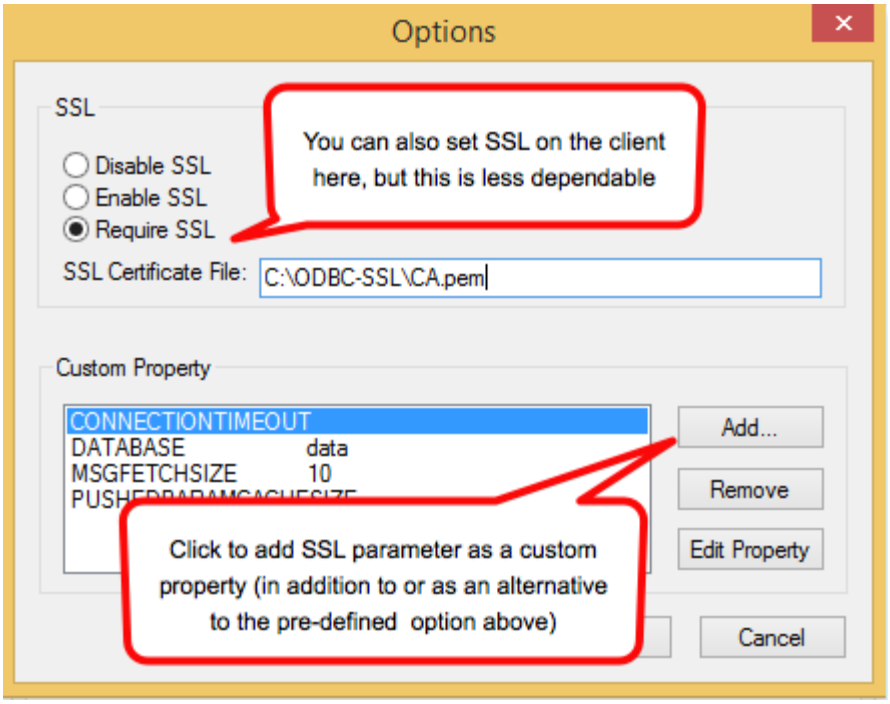
Note: Note that the SSL and certificate parameters can be set through the pre-defined options on the options dialog, but customers have reported that these are not always reliable. In the following procedure, we recommend using custom properties to define these settings (either preemptively, or as a solution if the ODBC connection over SSL does not work with the pre-defined options). There is no harm in setting both. Example settings are: `UseSSL = Required` and `SslCACertfile = C:\ODBC-SSL\CA.pem`

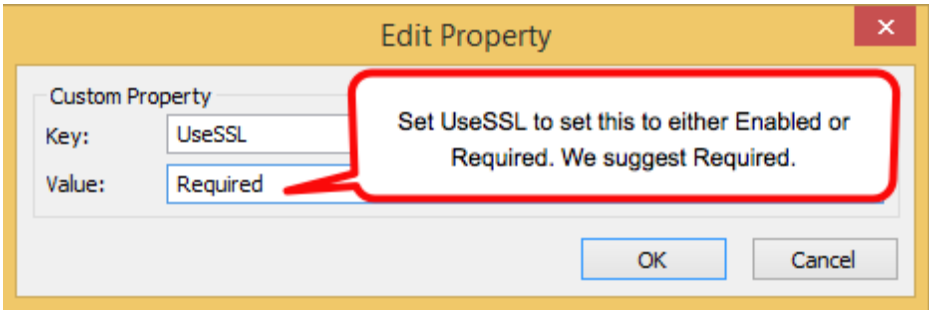
On the workstation you want to use for your ODBC connection, specify the level of SSL you want to use on the client along with the path to the CA certificate, and then test the connection.

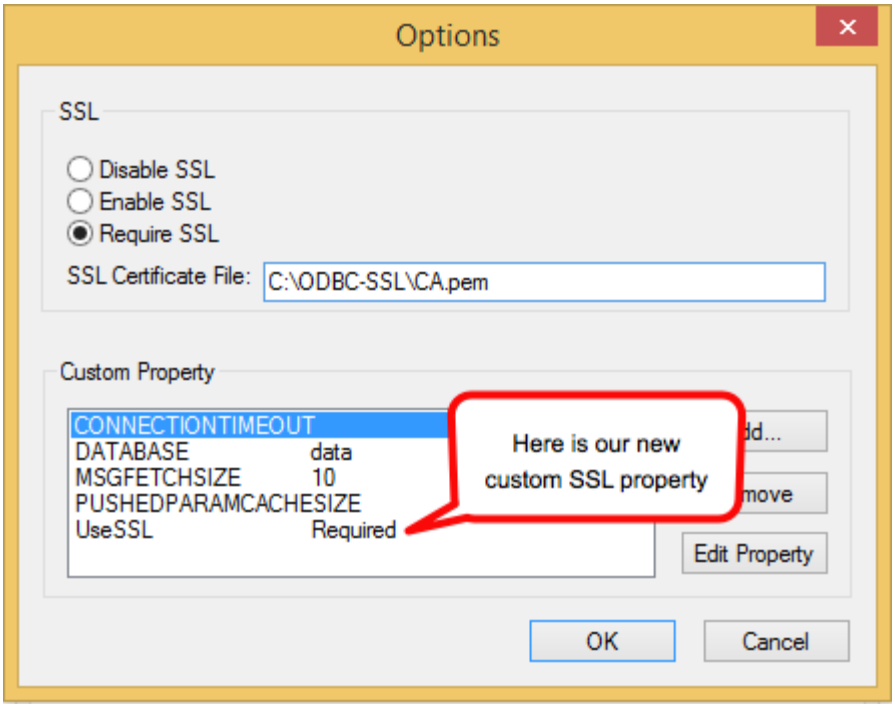
1. Save the CA certificate to a secure location on the workstation disk.

Choose a location where the certificate is unlikely to be deleted by mistake, for example, `C:\ODBC-SSL\CA.pem` is an example of a full path to such a location.

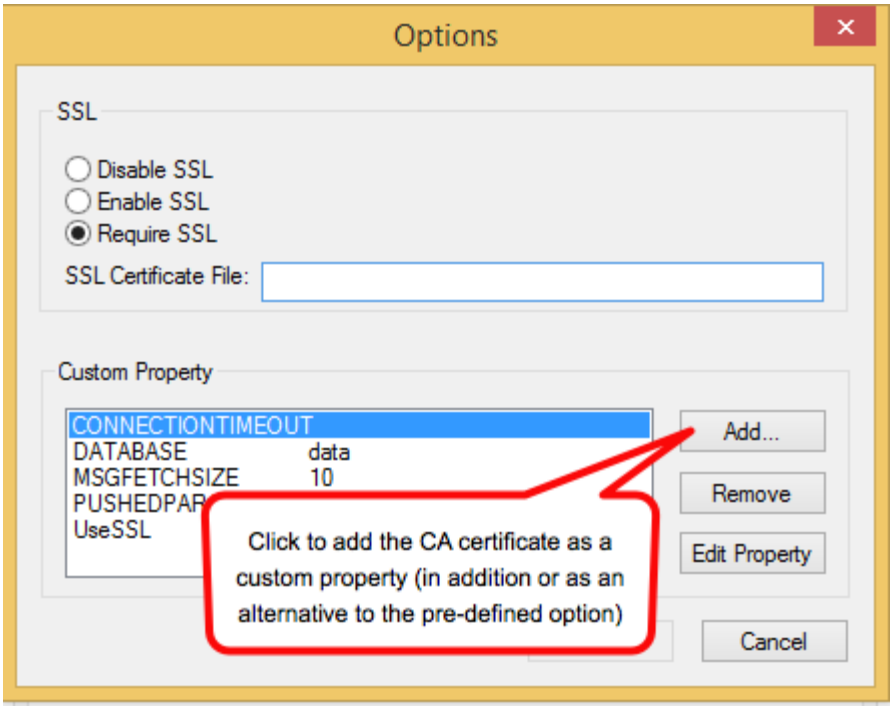
2. Open your ThoughtSpot ODBC connection configuration dialog.
3. Click **Options**.
4. Check the **Require SSL** option and/or add SSL as a custom property.

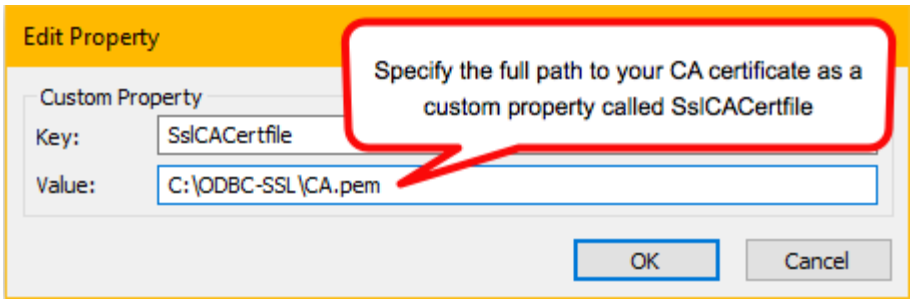

Options
SSL
Disable SSL
Enable SSL
Require SSL
You can also set SSL on the client here, but this is less dependable
SSL Certificate File:
C:\ODBC-SSL\CA.pem
Custom Property
CONNECTIONTIMEOUT
DATABASE data
MSGFETCHSIZE 10
Add...
Remove
Edit Property
Click to add SSL parameter as a custom property (in addition to or as an alternative to the pre-defined option above)
Cancel

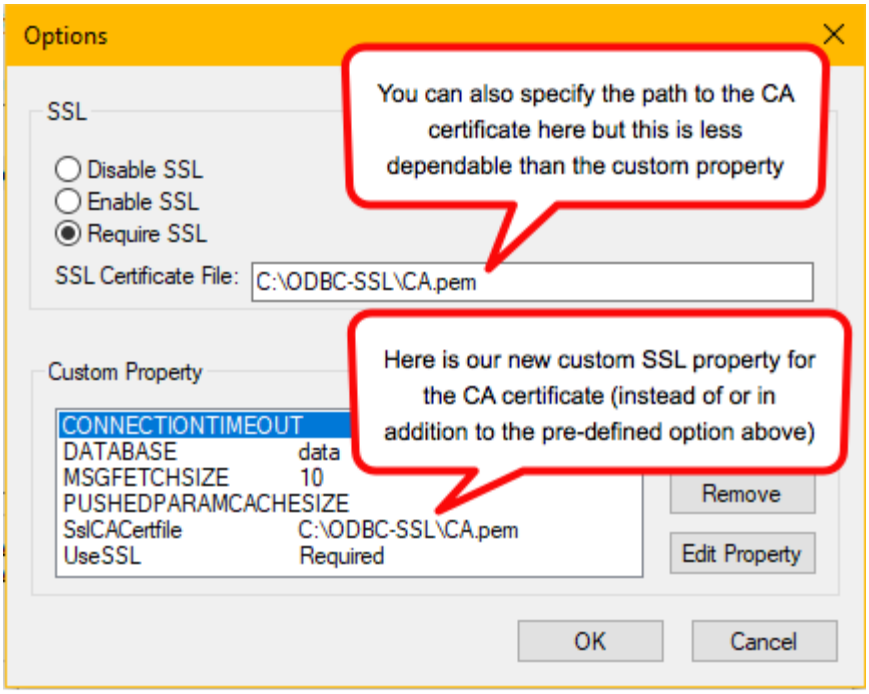

Edit Property
Custom Property
Key:
UseSSL
Value:
Required
Set UseSSL to set this to either Enabled or Required. We suggest Required.
OK
Cancel



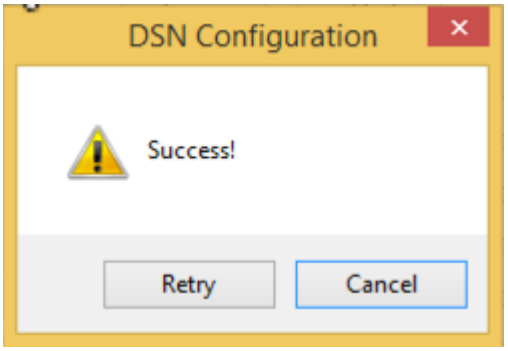
5. Enter the location of the CA certificate in the **SSL Certificate File** field and/or add the CA certificate as a custom property. Be sure to provide the full path to the certificate (`{certificate_directory}\{CA_certificate}.pem`).


Options
SSL
Disable SSL
Enable SSL
Require SSL
SSL Certificate File:
Custom Property
CONNECTIONTIMEOUT
DATABASE data
MSGFETCHSIZE 10
PUSHEDPAR
UseSSL
Add...
Remove
Edit Property
Cancel
Click to add the CA certificate as a custom property (in addition or as an alternative to the pre-defined option)


Edit Property
Custom Property
Key: SslCACertfile
Value: C:\ODBC-SSL\CA.pem
OK
Cancel
Specify the full path to your CA certificate as a custom property called SslCACertfile



6. When you are done, click **OK** to save your new properties.
7. Click **Test Connection** to test your database connection.



8. Click **Cancel** to close the configuration dialog.
9. Click **OK** to close the **Client Configuration Dialog** the dialog.
10. Click **OK** to close the **ODBC Data Source Administrator (64-bit)** application.

Set up the ODBC Driver for SSIS

Summary: Use SSIS to set up the ODBC Driver.

Microsoft SSIS (SQL Server Integration Services) is a data integration and workflow applications platform you can use to connect to ThoughtSpot. The platform is a component of the Microsoft SQL Server database software.

You can use a SSIS connection to perform data migration tasks. Its data warehousing tool is useful for data ETL (extraction, transformation, and loading). The SSIS Import/Export Wizard creates packages that transfers data with no transformations. It can move data from a variety of source types to a variety of destination types, including text files and other SQL Server instances.

Use SSIS to set up the ODBC Driver by creating a connection manager. This manager connects an OLE DB Source and the ODBC Destination.

Prerequisites

On Windows 64-bit, you have to install both the 32-bit and 64-bit ThoughtSpot ODBC drivers. In addition, they must be named the same, such as ThoughtSpot. By default they are named ThoughtSpot-32 and ThoughtSpot-64. This is required because the 64-bit SSIS shows a list of 32-bit ODBC drivers when you configure an ODBC target. However, it executes the 64-bit driver. If the drivers aren't named the same, then you can get an error stating the driver doesn't exist.

Set up the driver

To set up the ODBC driver using SSIS:

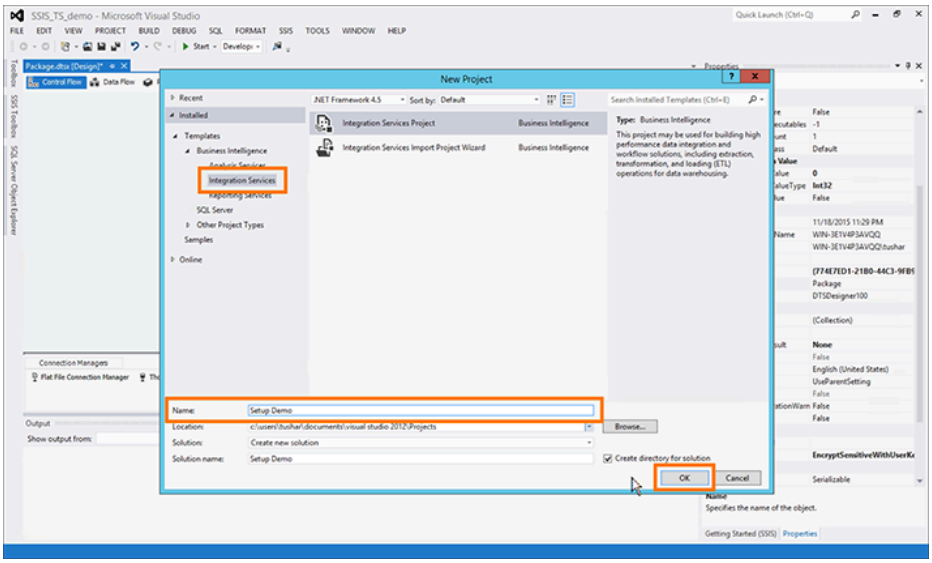
1. Open your SQL Server visual development tool that is based on Microsoft Visual Studio.
2. Select **OLE DB Source**, and click **New**.

   Where ODBC provides access only to relational databases, OLE DB provides access to data regardless of its format or location.
3. Add the server by name from the machine accessible list.

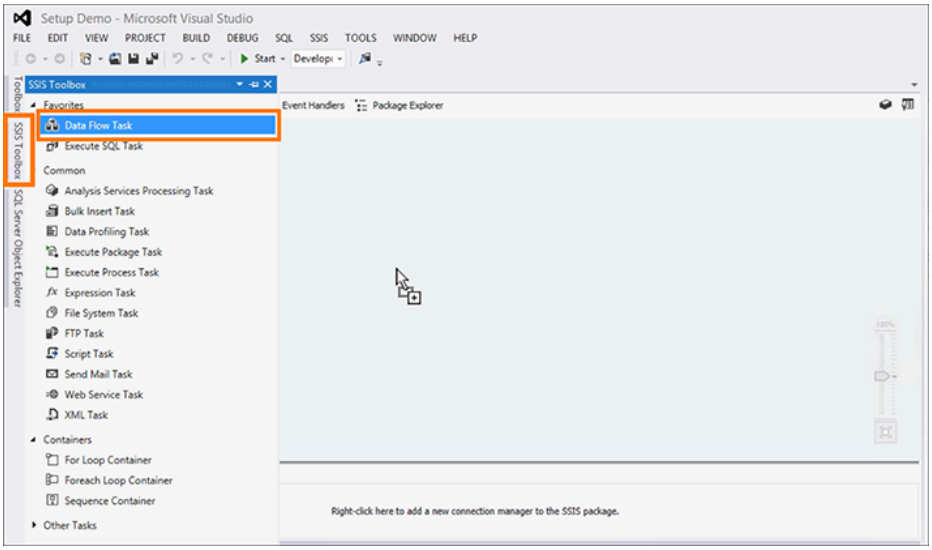
4. Enter the authentication information: db name, user name, password, and test connection.

   You can add the UID and password by clicking on **Options**.

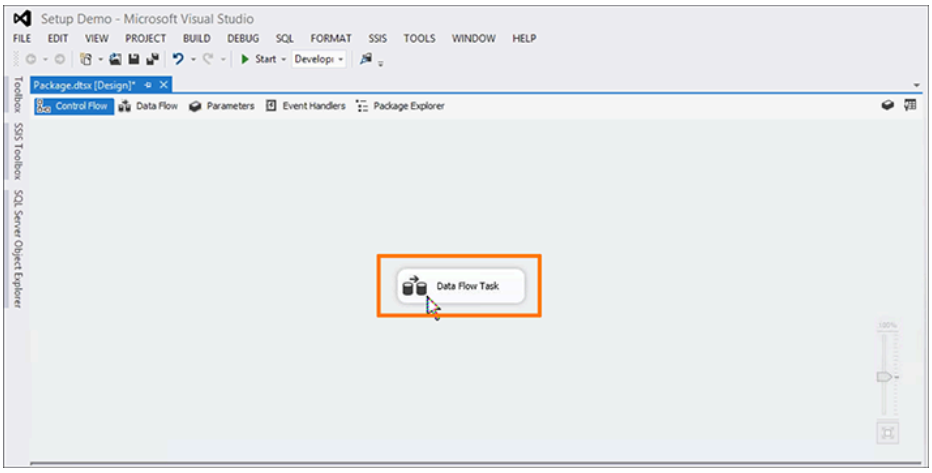
5. Click **File** and select **New**, then **Project**.
6. Select the **Integration Services** tab under **Installed > Templates > Business Intelligence**.
7. Enter a name in the **Name** field and click **OK**.



8. Select the **SSIS Toolbox** tab on the left hand side of the platform, and drag and drop **Data Flow Task** to the main window.



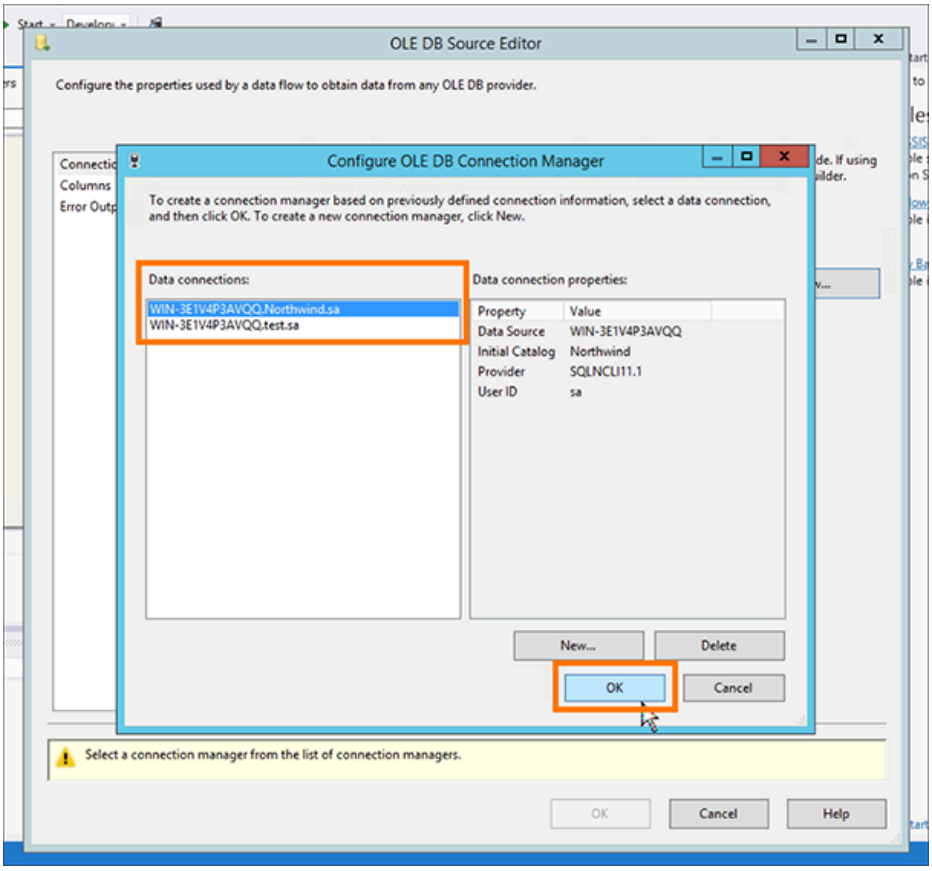
9. Double click the **Data Flow Task** icon when it appears in the center of the page.



10. Navigate back to the **SSIS Toolbox** tab. You now want to create sources and destinations.

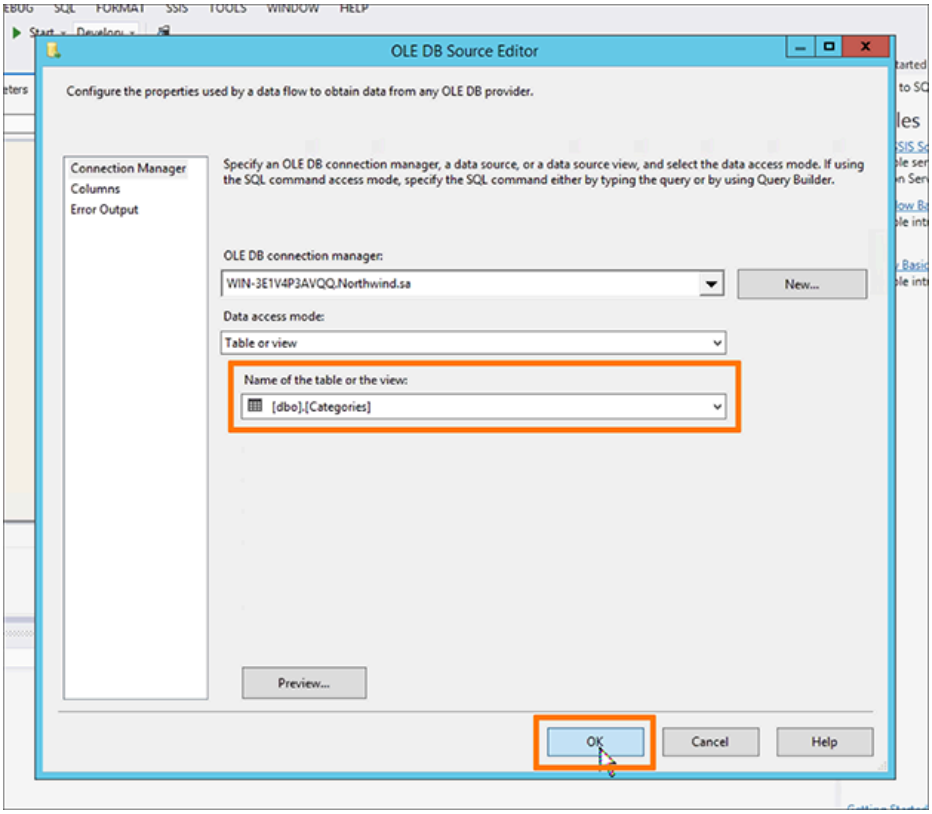
Create sources and destinations

1. Under **Other Sources**, find **OLE DB Source** and drag and drop it to the main window.
2. Double click the **OLE DB Source** icon when it appears in the center of the page to open the OLE DB Source Editor.
3. Select a new OLE DB connection manager by clicking **New**.
4. In the Configure OLE DB Connection Manager window, select your **Data connection** and click **OK**.



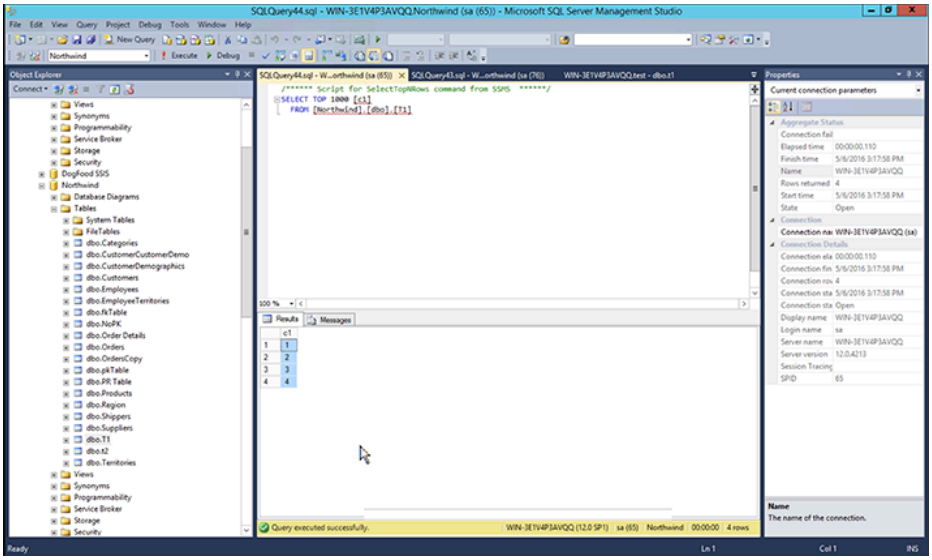
If you do not see your data connection, you will have to create a new one in the Connection Manager by clicking **New**.

5. Back in the OLE DB Source Editor, select the **Name of the table or the view**, and click **OK**.



6. Select the table, and see what columns are in it.

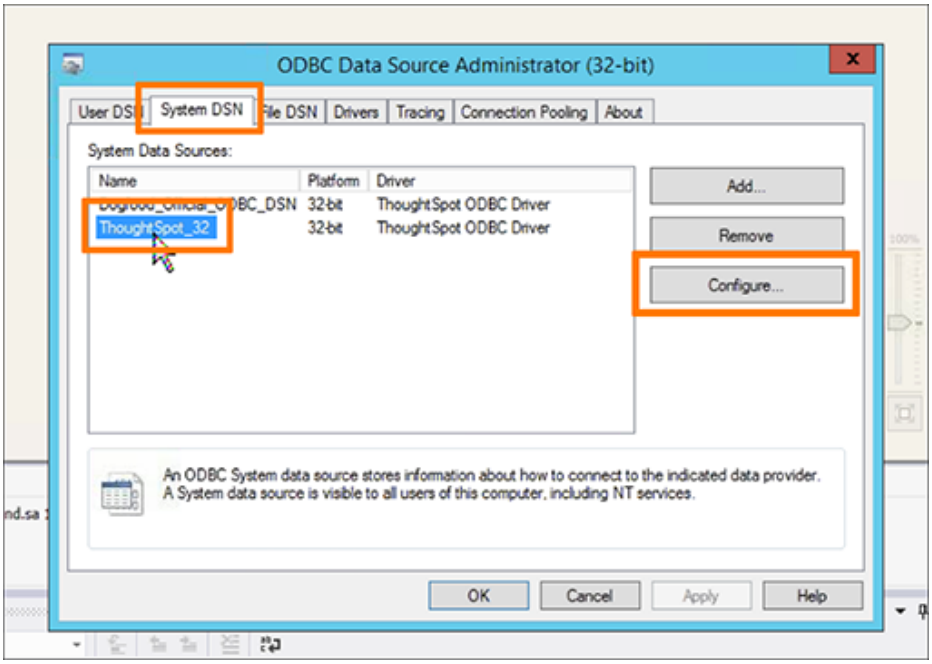
   In this example, a single column, `c1`, is selected.



Configure the ODBC Data Source Administrator

The ODBC Data Source Administrator has to be configured to connect to ThoughtSpot and bring the table in.

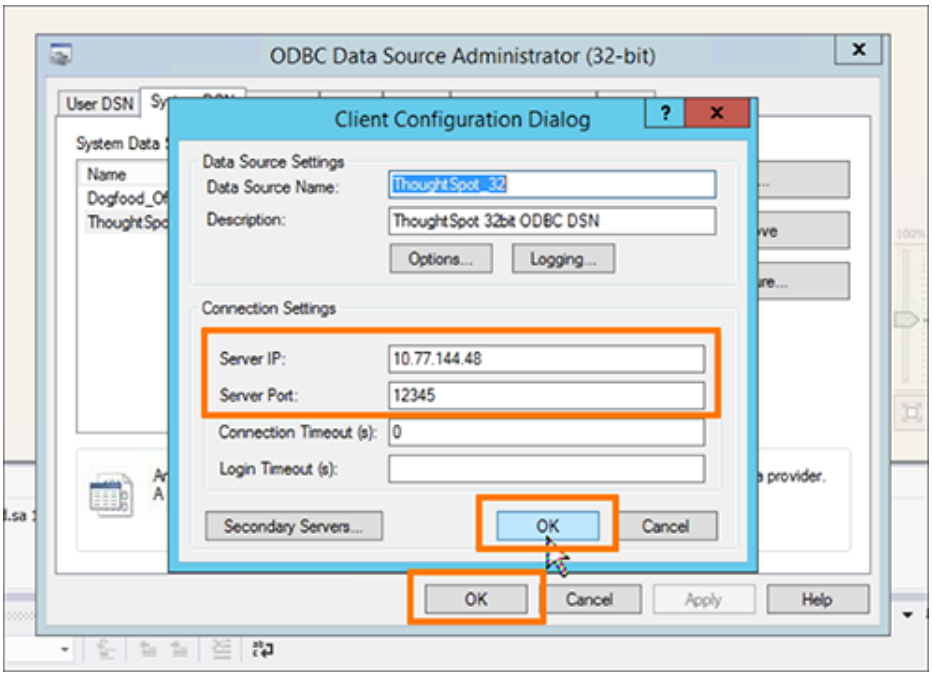
1. Search for and open your **ODBC Data Sources (32-bit)** program.
2. Click the **System DSN** tab and select **ThoughtSpot_32**.
3. Click **Configure**.



4. In the Client Configuration Dialog, enter the **Server IP** and **Server Port**.

   Enter any node IP that has Simba server running on it. In **Secondary Servers**, you must specify all node IPs, because ThoughtSpot must resolve to the server Simba runs on, and that server can change after an upgrade. Enter one server IP per line. The line return serves as a separator. Comma separated values are not supported.

5. Click **OK** twice to close the Client Configuration Dialog and the ODBC Data Source Administrator.

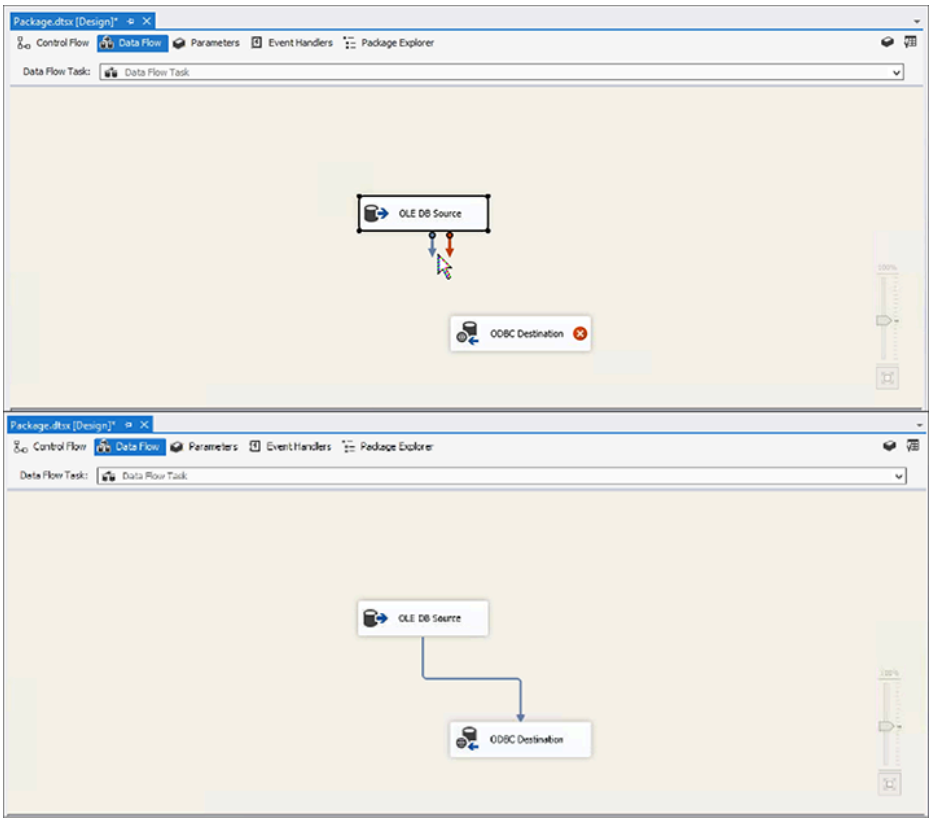


Create a file to take the feed

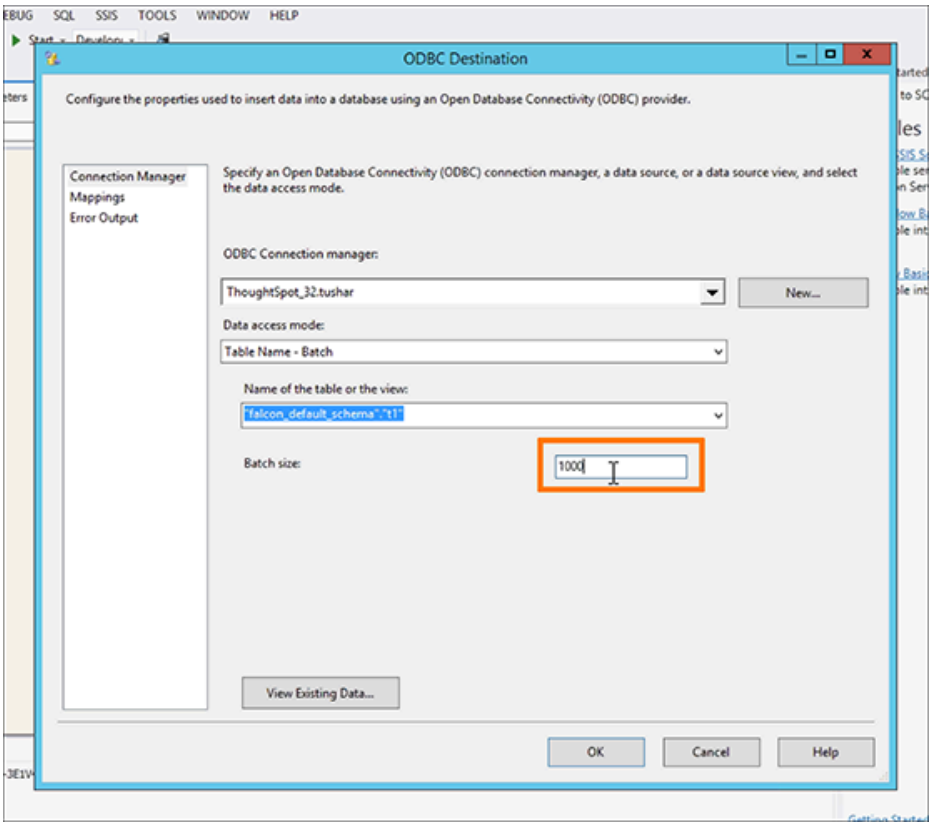
Now that you have set up your source, create the empty table in ThoughtSpot to take this feed. SSIS does not allow you to create the table in ThoughtSpot. You have to do this first in TQL. In Pentaho, it will create the table in ThoughtSpot, but not in SSIS.

Create the ODBC Destination. Use the one you created and named in the ODBC Data Source Administrator.

1. In the **SSIS Toolbox** tab, under **Other Destinations**, drag and drop **ODBC Destination** to the main window.
2. Drag the **blue arrow** to connect the OLE DB Source icon to the ODBC Destination icon.
3. Double click the **ODBC Destination** icon.



4. Use ODBC Destination to set the **Batch size** for the connection in the Connection Manager tab. You can set the size to be up to 10,000.



If the load fails, the entire batch will be lost, and you will have to start that load over again.

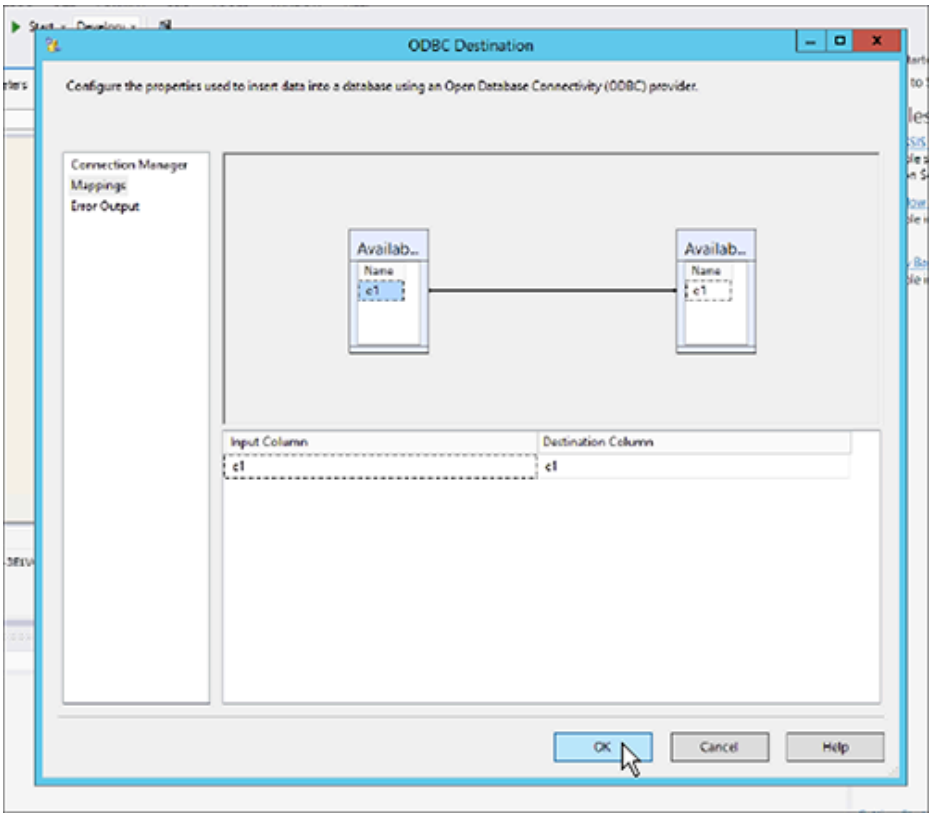
5. Set the **Transaction Size** to match the total number of rows that are expected to be loaded in the load cycle.

   Your transaction size can be quite large—even spanning a million rows. However, too many small batches can leave the cluster in a rough state. This is because each batch acts as a separate transaction and creates a separate commit. Too many of these will slow down our system since each transaction creates a “data version” in our system. In Pentaho, the transaction size setting is called Commit Size.

6. Set the **Transaction Option** attribute of the Data Flow Task to **Supported**.

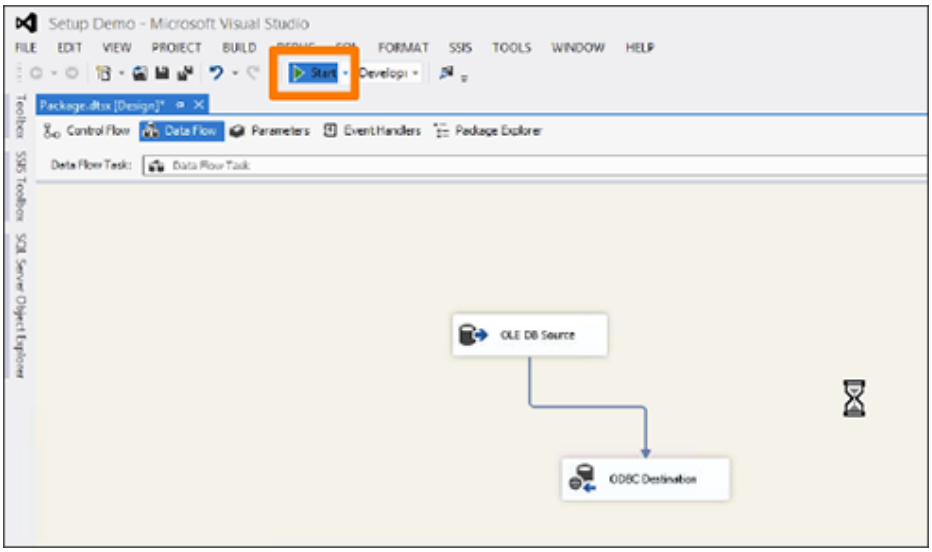
7. In the **Mappings** tab, validate the mapping or change it.

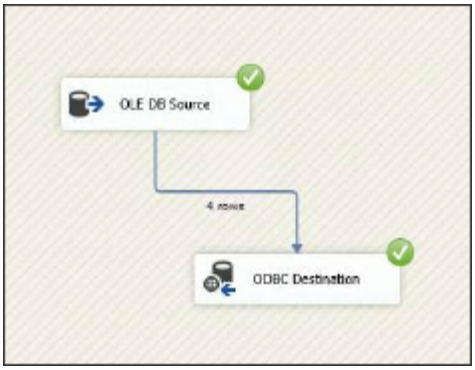
   You can have different column names in each database if you map them. Of course, they must be of the same or compatible datatype.



8. Start the import job by clicking the **Start** button.

   You should see an animation indicating that the data is transferring over. When the import is complete, the number of successfully transferred rows is displayed.





You can validate the import using TQL or from the **Data** screen.

Install the ODBC Driver on Linux

Summary: Use this procedure to obtain the Linux ODBC driver and install it.

ThoughtSpot's ODBC connection relies on the SimbaEngine X SDK to connect through ODBC or JDBC to ThoughtSpot's remote data stores. The instructions on this page explain how to configure the Simba ODBC driver on a Linux workstation.

Make sure you have read the overview material in the ODBC driver overview. This workstation is the same machine where you plan to run your ETL activities.

Check the ThoughtSpot IP and the simba_server status

Before you begin, you need to know the IP address or DNS name of the server you intend to connect your server to.

1. SSH as `admin` or the `thoughtspot` user to your ThoughtSpot node.
2. Verify the node IP(s).

```
$ tscli node ls
172.18.231.17
172.18.231.18
```

3. Make a note of each IP; there may be more than one.
4. Configure the ThoughtSpot firewall to allow connections from your ETL client, by running the following command on any ThoughtSpot node: `tscli firewall open-ports --ports 12345`
5. Exit or close the shell.

Install the Simba client

On your workstation, where you want to connect from, do the following to get the ODBC driver:

1. Open a browser on your workstation.

2. Navigate to the **Downloads** page.
3. Click **ODBC Driver for Linux** to download the driver.
4. Open a terminal on your workstation.
5. Change directory to the location where you downloaded the file.
6. Optionally, move the file to a permanent location on your machine.

   When you expand the downloaded file it will create a directory in the location.

7. Unzip the zip file:

   ```
   gunzip ThoughtSpot_linux_odbc_<version>.tar.gz
   ```

8. Extract the contents of the `tar` file.

   ```
   tar –xvf ThoughtSpot_linux_odbc_<version>.tar
   ```

   This extracts a subdirectory called `linux` into the current directory.

9. Take a moment to examine the contents of the new directory.

   The structure contains a Simba client library, supporting libraries and setup files for two different architectures. It also continues error messages for multiple languages.

```
├── Bin
│   ├── Linux_x86
│   │   └── libSimbaClient.so
│   └── Linux_x8664
│       └── libSimbaClient.so
├── ErrorMessages
│   ├── de-DE
│   │   ├── ...
│   ├── en-US
│   │   ├── CSCommonMessages.xml
│   │   ├── ClientMessages.xml
│   │   ├── JNIDSIMessages.xml
│   │   ├── ODBCMessages.xml
│   │   ├── SQLEngineMessages.xml
│   │   └── ServerMessages.xml
│   ├── es-ES
│   ├── fr-FR
│   ├── it-IT
│   └── ja-JP
├── Lib
│   ├── Linux_x86
│   │   ├── libcrypto.so
│   │   ├── libcrypto.so.1.0.0
│   │   ├── ...
│   └── Linux_x8664
│       ├── ...
└── Setup
    ├── .simba.javaquickstart.ini
    ├── .simba.javaultralight.ini
    ├── .simba.quickstart.ini
    ├── .simba.ultralight.ini
    ├── .simbaclient.ini
    ├── .simbaserver.javaquickstart.ini
    ├── .simbaserver.javaultralight.ini
    ├── .simbaserver.quickstart.ini
    ├── .simbaserver.ultralight.ini
    ├── odbc.ini
    └── odbcinst.ini

14 directories, 92 files
```

The `linux/Setup` directory contains the key ODBC configuration files and sample Simba client configurations you can use later in this procedure.

10. You must know your workstation architecture to continue, confirm your workstation's architecture.

    You can use the `arch` or the `uname` command or both.

    ```
     $ arch
       x86_64
     $ uname -a
       Linux nebula-docs-production-4vfnv 4.4.108-1.el7.elre
    po.x86_64 #1 SMP Mon Dec 25 09:55:39 EST 2017 x86_64 x8
    6_64 x86_64 GNU/Linux
    ```

    In previous examples, the workstation is a 64 bit workstation. Your workstation may be 32-bit. You can use this architecture information in the procedures that follow.

(Optional) Install unixODBC tools for testing

The procedures on this page rely on the unixODBC tools to test your configuration and connection. If you are experienced with ODBC and want to skip this, you can. Simply substitute your preferred mechanism in the subsequent procedures where references are made to the unixODBC tools.

> ❗ **Warning:** Your ThoughtSpot installation contains a version of the unixODBC tools. These tools are incompatible with CentOS. Do not use these tools if you are performing this procedure on your ThoughtSpot server.

1. Search for the unixODBC tools on your system.

   The `yum` package manager searches for software already installed or available on your system or from the configured repositories. Depending on your workstation configuration, you may need to use the `sudo` command with your workstation.

```
$ yum search unixODBC
...
 * updates: repos-lax.psychz.net
================ N/S matched: unixODB
C============================
opensips-unixodbc.x86_64 : OpenSIPS unixODBC Storage sup
port
unixODBC-devel.i686 : Development files for programs whi
ch will use the unixODBC library
unixODBC-devel.x86_64 : Development files for programs w
hich will use the unixODBC library
erlang-odbc.x86_64 : A library for unixODBC support in E
rlang
freeradius-unixODBC.x86_64 : Unix ODBC support for freer
adius
unixODBC.i686 : A complete ODBC driver manager for Linux
unixODBC.x86_64 : A complete ODBC driver manager for Lin
ux
```

Make note of the correct package to install for your architecture.

2. Install the appropriate package for your architecture.

   In this case the command installs the tools for a 64-bit architecture. A 32-bit package needs the `unixODBC.i686` package.

```
 [admin@nebula-docs-odbc-test-cxmrn ~]$ yum install unix
ODBC.x86_64
 Loaded plugins: fastestmirror, ovl
 Loading mirror speeds from cached hostfile
  * base: mirror.linuxfix.com
  * elrepo: repos.lax-noc.com
  * epel: mirror.hmc.edu
  * extras: centos-distro.cavecreek.net
  * rpmforge: mirror.lstn.net
  * updates: repos-lax.psychz.net
 Resolving Dependencies
 --> Running transaction check
 ---> Package unixODBC.x86_64 0:2.3.1-11.el7 will be ins
talled
 ...
 Complete!
```

3. Verify the files were installed.

```
$ ls /usr/bin/isql
/usr/bin/isql
$ ls /usr/bin/odbcinst
/usr/bin/odbcinst
```

Set up your environment

In this section, you set parameters in your workstation to support your ODBC connection.

1. Copy the library for your architecture from the `Lib` directory on your Linux machine.

Library	Architecture
`/linux/Lib/Linux_x86`	32-bit
`/linux/Lib/Linux_x8664`	64-bit

2. Add the location's path to the `LD_LIBRARY_PATH` environment variable.

   For example if your architecture is 64-bit and you keep the library in your `home` directory:

```
export LD_LIBRARY_PATH=~/linux/Lib/Linux_x8664/
```

3. Use the `echo` command to verify the path was added correctly.

```
echo $LD_LIBRARY_PATH
```

4. Copy the `odbc.ini` file to the `/etc` directory.

```
$ cp ~/linux/Setup/odbc.ini  /etc
```

If you have trouble making the copy, use the `sudo` command to make the move.

5. Copy the `odbcinst.ini` file to the `/etc` directory.

```
$ cp ~/linux/Setup/odbcinst.ini  /etc
```

6. Copy the hidden `.simba.quickstart.ini` file to the `/etc` directory, renaming it in the process to `simbaclient.ini`.

```
$ cp ~/linux/Setup/.simba.quickstart.ini  /etc/simbacli
ent.ini
```

7. Update your environment with the `ODBCSYSINI` and `ODBCINI` variables.

```
$ export ODBCSYSINI=/etc/
$ export ODBCINI=/etc/odbc.ini
```

8. Use the `/usr/bin/odbcinst` command to confirm your settings:

```
$ /usr/bin/odbcinst -j
unixODBC 2.3.1
DRIVERS............: /etc/odbcinst.ini
SYSTEM DATA SOURCES: /etc/odbc.ini
FILE DATA SOURCES..: /etc/ODBCDataSources
USER DATA SOURCES..: /etc/odbc.ini
SQLULEN Size.......: 8
SQLLEN Size........: 8
SQLSETPOSIROW Size.: 8
```

Edit the /etc/simbaclient.ini file

When you are ready, follow this procedure to configure the driver.

1. Edit the `/etc/simbaclient.ini` file with your favorite editor. 2.

2. Change the `ErrorMessagesPath` property to point to the location where you unzipped the client.

```
[Driver]
ErrorMessagesPath=<path_to_error_messages_directory>
```

3. Comment out the `# Generic ODBCInstLib` value.
4. Uncomment the `ODBCInstLib` property.

   When you are done, your file looks like the following:

```
# Generic ODBCInstLib
#   iODBC
#ODBCInstLib=libiodbcinst.so

#   SimbaDM / unixODBC
ODBCInstLib=libodbcinst.so
```

5. Save and close the `/etc/simbaclient.ini` file.

Edit the odbcinst.ini file

The `odbcinst.ini` file is a registry and configuration file for ODBC drivers. Depending on your workstation architecture, you configure the 32-bit or 64-bit driver.

1. Open the file `/etc/odbcinst.ini` in your favorite editor.
2. Comment out the driver that you don't need.

   For example, if you are using 64-bit, comment out 32-bit.

3. Edit the `Driver` line so that it contains the path to the file `libSimbaClient.so`

   Use the path where you copied the library files. For example, for the 64-bit ODBC driver:

```
 [ThoughtSpot(x64)]
 APILevel            = 1
 ConnectFunctions    = YYY
 Description         = ThoughtSpot 64bit ODBC driver
 Driver              = /home/admin/linux/Bin/Linux_x866
4/libSimbaClient.so
 DriverODBCVer       = 03.52
 SQLLevel            = 1
```

4. Make sure the remaining driver is named `ThoughtSpot` without any special characters.

   When you are done, you should see something similar to the following:

```
 #[ThoughtSpot]
 #APILevel            = 1
 #ConnectFunctions    = YYY
 #Description         = ThoughtSpot 32bit ODBC driver
 #Driver              = /usr/local/scaligent/toolchain/l
ocal/simba/odbc/linux/Bin/Linux_x86/libSimbaClient.so
 #DriverODBCVer       = 03.80
 #SQLLevel            = 1

 [ThoughtSpot]
 APILevel            = 1
 ConnectFunctions    = YYY
 Description         = ThoughtSpot 64bit ODBC driver
 Driver              = /home/admin/linux/Bin/Linux_x866
4/libSimbaClient.so
 DriverODBCVer       = 03.80
 SQLLevel            = 1
```

5. Save and close the `/etc/odbcinst.ini` file.

Edit the odbc.ini file

The `odbc.ini` file is a registry and configuration file for ODBC DSNs (Data Source Names). This file relies on the drivers registered in the `/etc/odbcinst.ini` file. Depending on your workstation architecture, you configure the 32-bit or 64-bit driver.

1. Open the file `/etc/odbc.ini` in the editor of your choice.

2. Comment out the configuration that you don't need.

   For example, if you are using 64-bit, comment out 32-bit.

3. Locate the `Description` section for the type of Linux you are using (32-bit or 64-bit).
4. Locate the line that begins with `ServerList`.
5. Replace `127.0.0.1` with a comma separated list of the IP addresses of each node on the ThoughtSpot instance.

   The syntax for the `ServerList` is:

   ```
    ServerList = <node1_IP> 12345, <node2_IP> 12345 [, <nod
   e3_IP> 12345, ...]
   ```

   If you need to obtain the IP addresses of the ThoughtSpot cluster nodes, run the command `tscli node ls` from a Linux shell on a ThoughtSpot appliance.

6. Do not edit the port number, leave it as `12345`.

   When you are done, your entry will look similar to the following (this example is for the 64-bit ODBC driver):

   ```
    [ThoughtSpot]
    Description = ThoughtSpot 64-bit ODBC Driver
    Driver = ThoughtSpot
    ServerList =  172.18.231.17 12345
    Locale = en-US
    ErrorMessagesPath = /home/admin/linux/ErrorMessages
    UseSsl = 0
    #SSLCertFile = # Set the SSL certificate file path. Th
   e certificate file can be obtained by extracting the SD
   K tarball
    #LogLevel = 0 # Set log level to enable debug logging
    #LogPath = # Set the debug log files path
    DATABASE = # Set the default database to connect to
    SCHEMA = # Set the default schema to connect to
   ```

7. Save and close the `odbc.ini` file.

Test your ODBC connection

At this point, you can test your ODBC connection to ThoughtSpot. It is important to recall that the username/password you use belongs to a ThoughtSpot application user. Typically, this user is a user with data management or administrative privileges on the application.

1. Before trying the ODBC connection, make sure you can use this username/password to login into the ThoughtSpot application.
2. Confirm the user's privileges by going to the **Data** tab.
3. Go back to your workstation's terminal shell.
4. Use the `/usr/bin/isql` and confirm you can connect.

   Specify the `ThoughtSpot` DSN:

```
/usr/bin/isql –v ThoughtSpot tsadmin adminpwd
+---------------------------------------+
| Connected!                            |
|                                       |
| sql–statement                         |
| help [tablename]                      |
| quit                                  |
|                                       |
+---------------------------------------+
SQL>
```

Now, you are ready to begin using the connection you've configured.

Best Practices for Using ODBC

Summary: To successfully use ODBC, following these best practices is recommended.

When developing tools that use the ODBC driver, use these best practices:

- When setting up ODBC for the first time, begin by using the ThoughtSpot `tsload` for the initial data loads. This allows you to do more in-depth troubleshooting on any initial loading issues. After initial loads work properly, switch to ODBC to perform incremental loads.

- You should create the parameterized SQL statement outside of ODBC. Using this method, the SQL statement can be sent to ThoughtSpot in batches by the ODBC driver, so you only have to update the memory itself. ETL tools have this implemented already (end users shouldn't have to actually write the `INSERT` statement). But as a developer, you may be writing code that leverages the ODBC driver, so this tip can help you write your SQL for the best performance with the driver.

- Data can be loaded into a table through multiple parallel connections. You can achieve this by splitting the input data into multiple parts. Then, load those individual parts through multiple parallel connections. You can use parallel loading even while loading to a single table or multiple tables at the same time.

- When doing an incremental data load, note that the same `UPSERT` behavior that occurs in TQL also occurs. This means that if you import a row whose primary key matches an existing row, the existing row will be updated with the new values.

Related information

- Enable ODBC logs
- Introduction to loading and managing data
- Loading and constraints

JDBC Driver Overview

Summary: Use JDBC to interact with databases in a standard manner.

Java Database Connectivity (JDBC) is a Java standard API that allows applications to interact with databases in a standard manner. ThoughtSpot has JDBC support through a JDBC driver that we provide.

Connector type

There are different types of JDBC connectors. Driver types categorize the technology used to connect to the database. The ThoughtSpot JDBC drive is a type 4 connector. It uses Java to implement a networking protocol for communicating with ThoughtSpot.

This driver is Java driver. There is no client installation or configuration.

When to use JDBC

JDBC can be used whenever you want to connect to ThoughtSpot to insert data programmatically from a Java program or application. You should begin by using the ThoughtSpot Loader for initial data loads and then use JDBC for incremental loads. This is because the ThoughtSpot Loader is generally faster than JDBC. Information on using the ThoughtSpot Loader is available in the ThoughtSpot Administrator Guide.

Version Compatibility

To ensure compatibility, always use the JDBC driver with the same version number as the ThoughtSpot instance to which you are connecting.

Performance Considerations

These are some general recommendations for maximizing the performance of JDBC:

- Insert in batches rather than doing single inserts at a time using the

`PreparedStatement::addBatch()` and `PreparedStatement::executeBatch` commands.

- If you need to upload a lot of data, consider running multiple connections with batch inserts in parallel.

Note: The ETL tool must add a data transformation step if the source column data type does not exactly match the target's, ThoughtSpot's, column data type. The driver does not do any implicit conversions.

Use the JDBC Driver

Summary: How to configure the JDBC driver.

ThoughtSpot's ODBC connection relies on the SimbaEngine X SDK to connect through ODBC or JDBC to ThoughtSpot's remote data stores. The instructions on this page explain how to configure the JDBC driver.

The ThoughtSpot JDBC driver is supplied by a `.jar` file you install on a workstation. This workstation is the same machine where you plan to run your ETL activities.

JDBC configuration parameters

Information	Description
Driver name	`com.simba.client.core.jdbc4.SCJDBC4Driver`
Server IP address	The ThoughtSpot appliance URL or IP address.
Simba port	The simba port, which is `12345` by default.
Database name	This is not the machine login username. The ThoughtSpot Database name to connect to.
username	The name of a ThoughtSpot user with administrator permissions.
password	The password of a ThoughtSpot application user. This is not the machine or SSH userpassword.

For more JDBC configuration options, see also:

- JDBC properties reference in this ThoughtSpot documentation
- SimbaClient for JDBC Configuration Properties reference

Check the ThoughtSpot IP and the simba_server status

Before you begin, you need to know the IP address or DNS name of the server you intend to connect your server to.

1. SSH as `admin` or the `thoughtspot` user to your ThoughtSpot node.
2. Verify the node IP(s).

```
$ tscli node ls
172.18.231.17
172.18.231.18
```

3. Make a note of each IP; there may be more than one.
4. Configure the ThoughtSpot firewall to allow connections from your ETL client, by running the following command on any ThoughtSpot node: `tscli firewall open-ports --ports 12345`
5. Exit or close the shell.

Install the driver

The JDBC driver is a `.jar` packaged application. To use the package, you download it, install it

1. Log in to the local machine where you want to install the JDBC driver.
2. Click **Here** to download the JDBC driver.
3. Click **JDBC Driver** to download the file `thoughtspot_jdbc<version>.jar`.
4. Move the driver to the desired directory on your local machine.
5. Add the downloaded JDBC driver to your Java class path on the local machine.

Write your application

Using JDBC with ThoughtSpot is the same as using any other JDBC driver with any other database. You must provide the connection information, create a connection, execute statements, and close the connection.

Specify each of the nodes in the cluster in the connection string, as shown. This enables high availability for JDBC connections. To find out the nodes in the cluster, you can run the command `tscli node ls` from the Linux shell on the ThoughtSpot instance.

The format for the connection is:

```
jdbc:simba://<node1>:12345,<node2>:12345,<node3>:12345[,…];
          LoginTimeout=<seconds>;DATABASE=<db>;SCHEMA=<schema>
```

For example:

```
jdbc:simba://192.168.2.248:12345,192.168.2.249:12345,192.16
8.2.247:12345;
          LoginTimeout=5;DATABASE=test;SCHEMA=falcon_default_s
chema
```

As shown, the `DATABASE` and `SCHEMA` parameters need to be in all caps. For the `simba` JDBC driver to work with Spark, the `DATABASE` and `SCHEMA` must be specified in the URL. They cannot be specified as a name/value pair as a map or property. For example:

```
val tssqldf1 = sparkSession.read.format("jdbc").options(Map("ur
l" ->
"jdbc:simba://10.84.78.181:12345;DATABASE=movieratings;SCHEMA=f
alcon_default_schema", "driver" ->
"com.simba.client.core.jdbc4.SCJDBC4Driver", "dbtable" -> "Movi
es", "user" ->
"tsadmin", "password" -> "admin")).load()
```

This `InsertData.java` example shows how to use ThoughtSpot with JDBC. This is an example of a reference JDBC application:

```
import java.sql.DriverManager;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class InsertData {

  // JDBC class to use.
  private static final String DB_DRIVER = "com.simba.client.cor
e.jdbc4.SCJDBC4Driver";
  // jdbc_example should be an existing database.

  private static final String DB_CONNECTION = "jdbc:simba://19
2.168.2.129:12345;
     192.168.2.249:12345,192.168.2.247:12345;
     LoginTimeout=5;DATABASE=jdbc_example;SCHEMA=falcon_defaul
t_schema";

  private static final String TABLE_NAME = "jdbc_example";
  private static final String DB_USER = "<username>";
  private static final String DB_PASSWORD = "<password>";

  // Assuming everything in local directory use:
  //   javac InsertData.java
  //   java -cp .:thoughtspot_jdbc4.jar InsertData
  public static void main(String[] argv) {

    try {
      insertRecordsIntoTable();
    }
    catch (SQLException e) {
      System.out.println(e.getMessage());
    }
  }

  /**
   * Insert some records using batch updates.
   *  Assumes a table exists:  CREATE TABLE "jdbc_example" ( "t
ext" varchar(10) );
   */
  private static void insertRecordsIntoTable() throws SQLExcept
ion {

    System.out.println("Inserting records.");
    Connection dbConnection = getDBConnection();
```

```
    PreparedStatement preparedStatement = null;
    String insertTableSQL = "INSERT INTO falcon_default_schem
a.jdbc_example (text) VALUES (?)";

    try {
      preparedStatement = dbConnection.prepareStatement(insertT
ableSQL);

      // Create multiple statements and add to a batch update.
      for (int cnt = 1; cnt <= 10; cnt++) {
        preparedStatement.setString(1, "some string " + cnt);
        preparedStatement.addBatch();
        System.out.println("Record " + cnt + " was added to th
e batch!");
      }
      preparedStatement.executeBatch();  // For large numbers o
f records, recommend doing sets of executeBatch commands.
      System.out.println("Records committed");

    }
    catch (SQLException sqle) {
      sqle.printStackTrace();
    }
    finally {

      if (preparedStatement != null) {
        preparedStatement.close();
      }
      if (dbConnection != null) {
        dbConnection.close();
      }
    }
  }

  /** Create a connection to the database. */
  private static Connection getDBConnection() {
    Connection dbConnection = null;
    try {
      Class.forName(DB_DRIVER);
    }
    catch (ClassNotFoundException e) {
      System.out.println(e.getMessage());
    }
    try {
      dbConnection = DriverManager.getConnection(DB_CONNECTIO
```

```
N, DB_USER,DB_PASSWORD);
      return dbConnection;
    }
    catch (SQLException sqle) {
      System.out.println(sqle.getMessage());
    }

    return dbConnection;
  }

}
```

Related Information

- Enable JDBC logs
- Connection configuration
- Supported SQL commands

Set up the JDBC driver for Pentaho

Summary: JDBC to connect to the ThoughtSpot Simba server from Pentaho.

You can use the Pentaho Data Integration (PDI) to create a JDBC connection. The Pentaho Data Integration (PDI) suite is a comprehensive data integration and business analytics platform. You can use it to create a JDBC connection to ThoughtSpot.

PDI consists of a core data integration (ETL) engine and GUI applications that allow you to define data integration jobs and transformations. Through Pentaho, we primarily use the JDBC driver to set up a connection. The process is not as complicated as with SSIS, and is much more lenient.

Community and enterprise editions of PDI are available. Using the community edition is sufficient, though you may use the enterprise edition, which is subscription based, and therefore contains extra features and provides technical support.

Use JDBC to connect to the ThoughtSpot Simba server from Pentaho. The connection will be made between a new ThoughtSpot Table Input and Output objects.

Check the ThoughtSpot IP and the simba_server status

Before you begin, you need to know the IP address or DNS name of the server you intend to connect your server to.

1. SSH as `admin` or the `thoughtspot` user to your ThoughtSpot node.
2. Verify the node IP(s).

```
$ tscli node ls
172.18.231.17
172.18.231.18
```

3. Make a note of each IP; there may be more than one.
4. Configure the ThoughtSpot firewall to allow connections from your ETL client, by running the following command on any ThoughtSpot node: `tscli firewall open-ports --ports`

`12345`

5. Exit or close the shell.

Install the Simba drivers in the Pentaho directories

Before starting the Pentaho Data Integration (PDI) client and creating the connection, ensure that the Simba JDBC client libraries are present in the Pentaho client/server machines. This will ensure that the drivers picked up at runtime.

1. Log in to the local machine where you have alredy installed the Pentaho Data Integration (PDI) client.
2. Click **Here** to download the JDBC driver.
3. Click **JDBC Driver** to download the file `thoughtspot_jdbc<version>.jar`.
4. Copy the `thoughtspot_jdbc<version>.jar` file to the following directories:
    - `<Pentaho_install_dir>/server/data-integration-server/tomcat/webapps/pentaho-di/WED-INF/lib/`
    - `<Pentaho_install_dir>/design-tools/data-integration/lib/`
    - `<Pentaho_install_dir>/server/data-integration-server/tomcat/lib/`
    - `<Pentaho_install_dir>/design-tools/data-integration/plugins/spoon/agile-bi/lib/`

Set up the driver

This section explains how to set up the JDBC driver using Pentaho. These instructions use Spoon, the graphical transformation and job designer associated with the PDI suite. It is also known as the Kettle project.

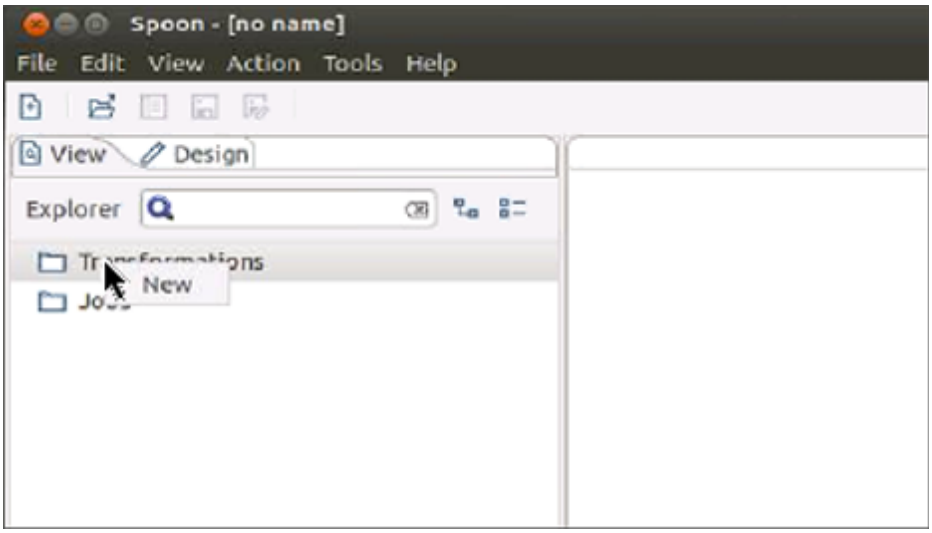
Create a transformation

Do the following on your ETL workstation with the Pentaho client:

1. Open the PDI client.

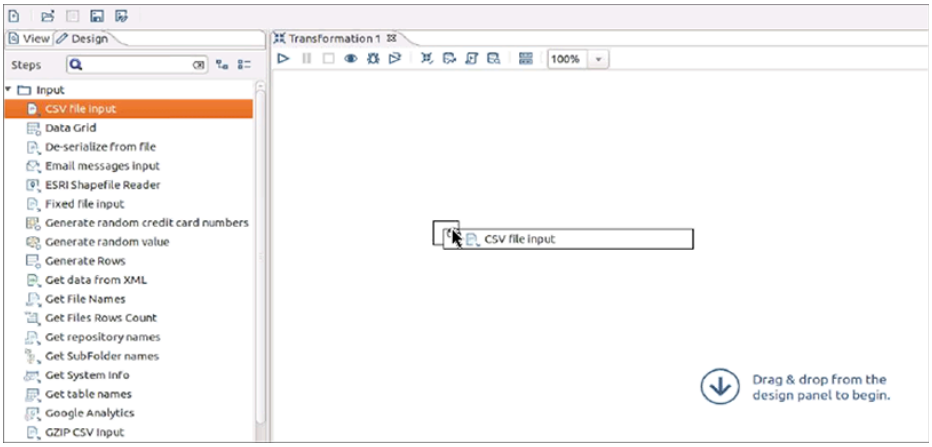
```
./spoon.sh &>/dev/null &
```

2. Right click **View > Transformations** tab.
3. Click **New** to create a new transformation.

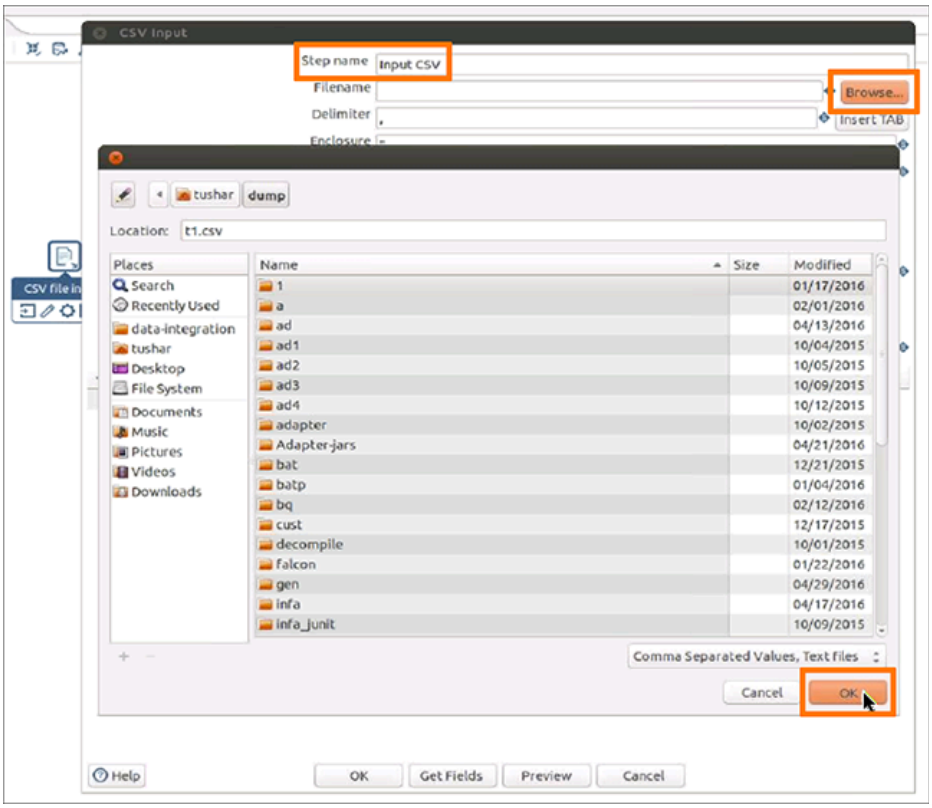


4. Click **Input** under the **Design** tab to expand it.
5. Drag and drop **CSV File Input** to the **Transformation** window.

   This opens a new CSV file.



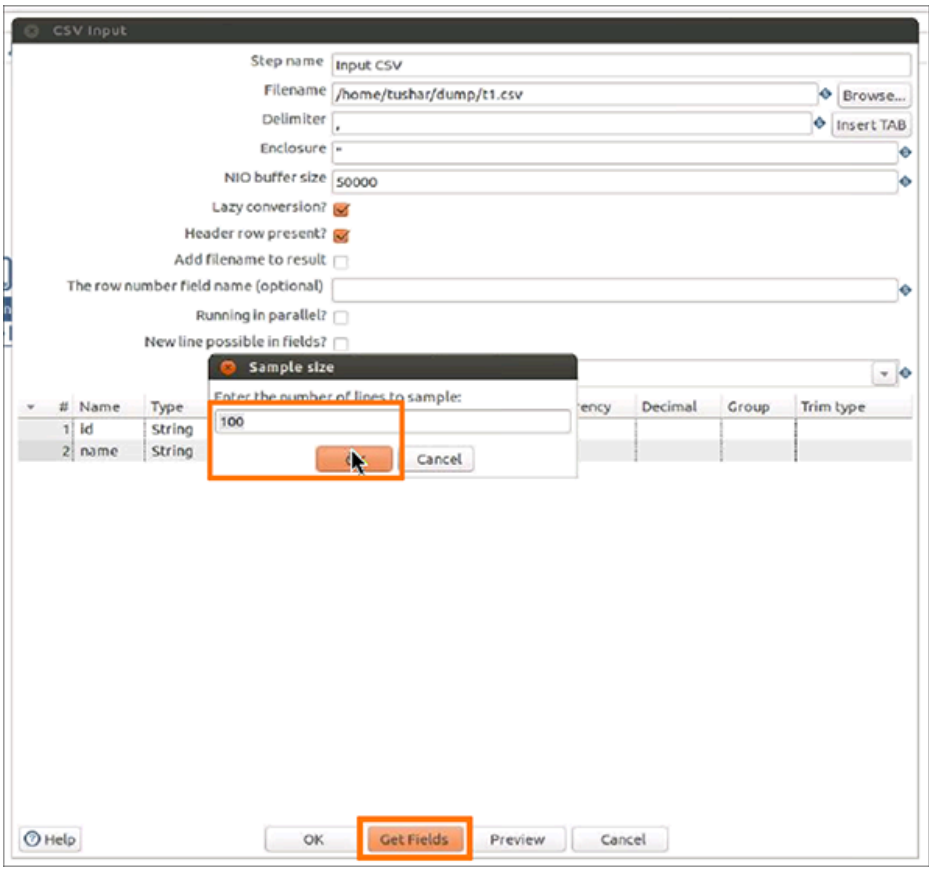
6. Double-click the **CSV File Input** icon to open the **CSV Input** dialog .
7. Name the **Step**.
8. Click **Browse** next to the **Filename** field and provide the file you want to read from.
9. Click **OK**.



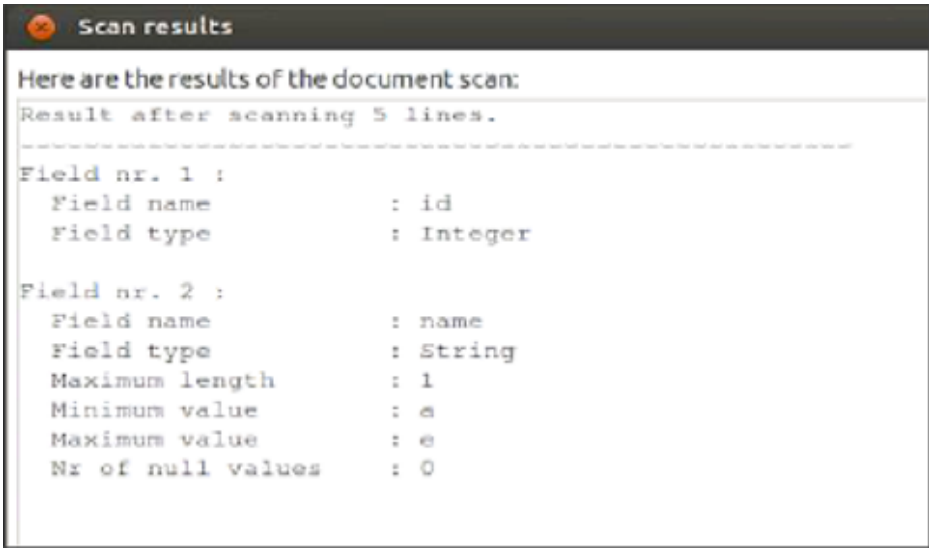
10. In the CSV Input dialog, click **Get Fields**.
11. Enter the number of lines you would like to sample in the Sample size dialog.

The default setting is 100.

1. Click **OK** when you are ready.



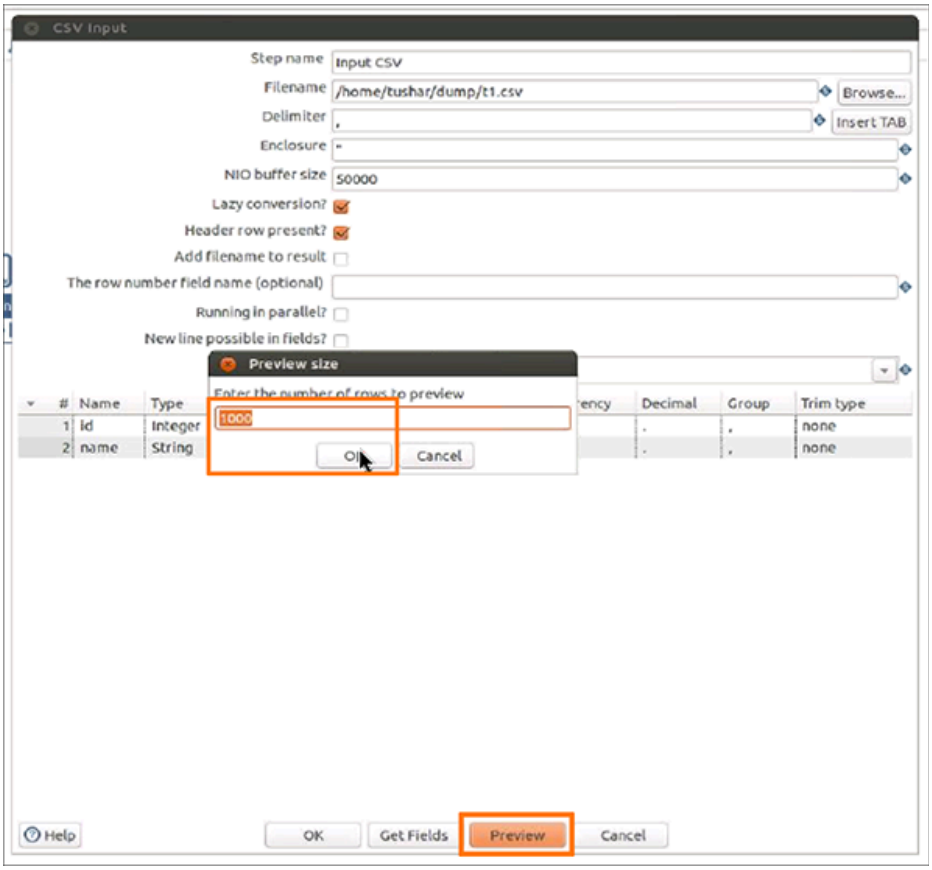
The tool reads the file and suggests the field name and type.



2. Click **Preview** to preview the data.

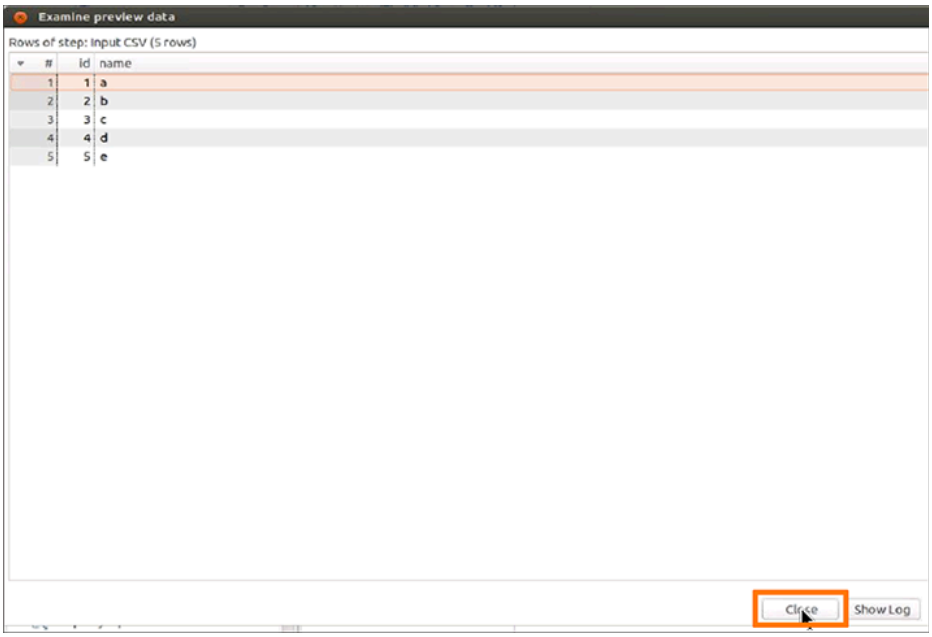
3. Enter the number of rows to preview in the **Preview size** dialog.

The default setting is 1000. Click **OK** to start the transformation in preview.



4. Examine the preview data, then click **Close**.

You may want to verify that you are able to read the data using the SQL query from ThoughtSpot.

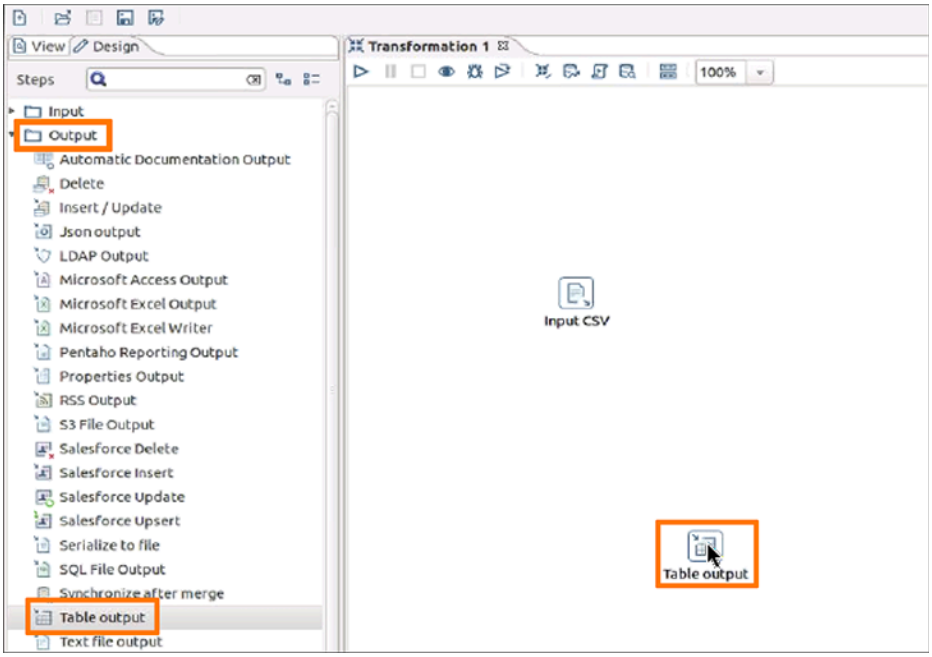


5. Click **OK** in the CSV Input dialog to confirm your CSV input settings.

Define the Output

1. Click **Design > Output**.

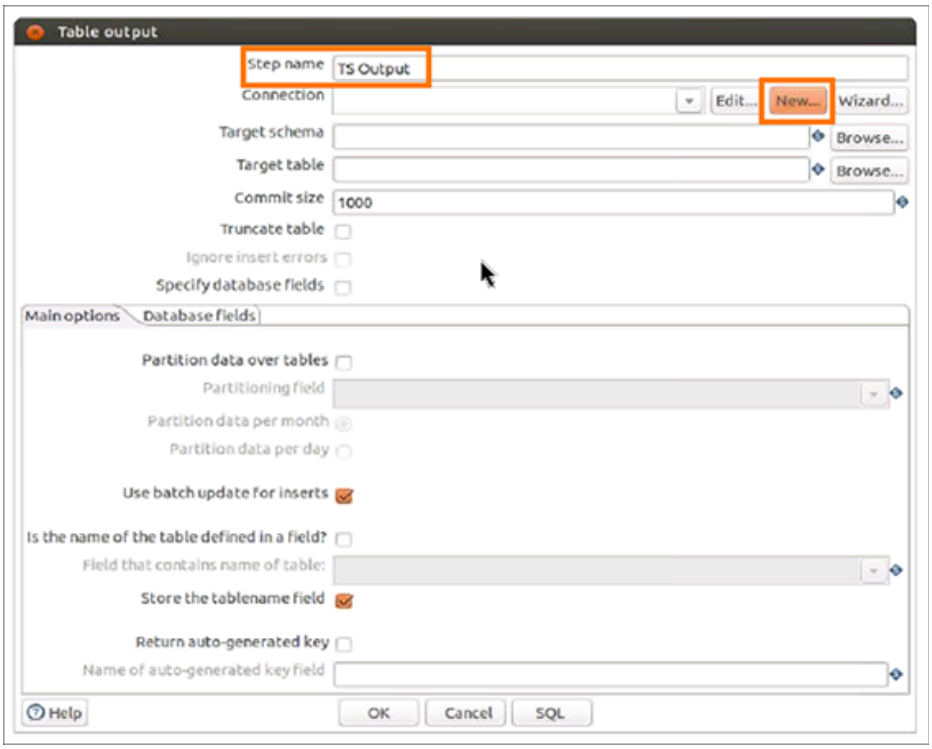
2. Drag and drop **Table output** to the **Transformation** window.



3. Double click the **Table output** icon to open the Table output dialog.

4. Enter a **Step name**.

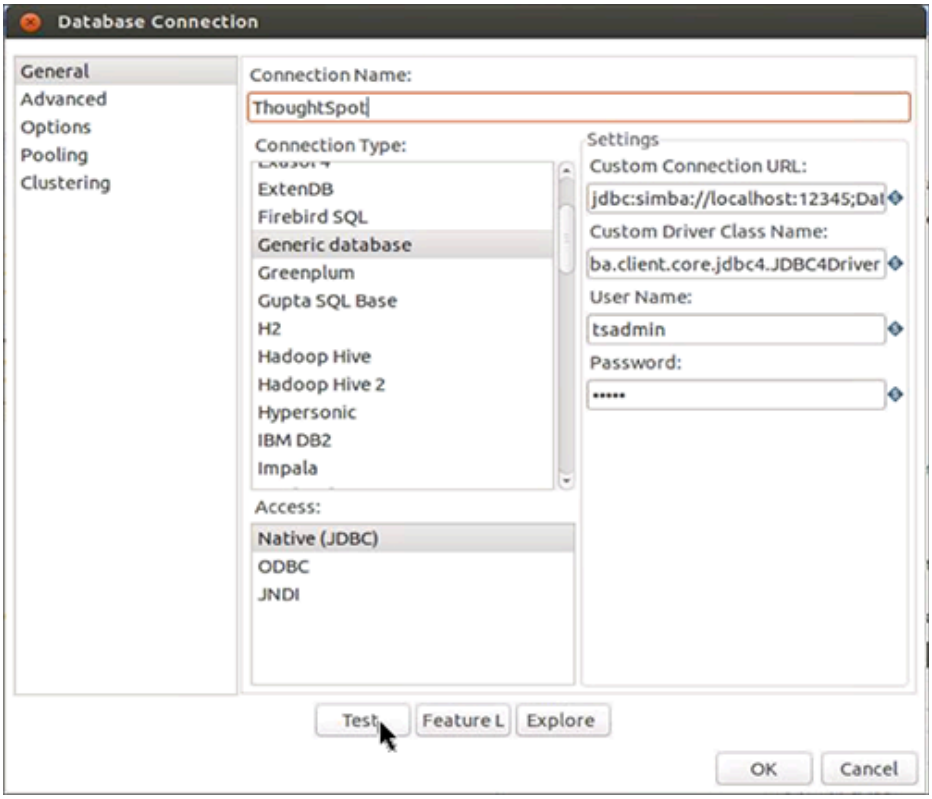
5. Click **New** to create a new connection.



6. Enter or select the following information in the Database Connection dialog:

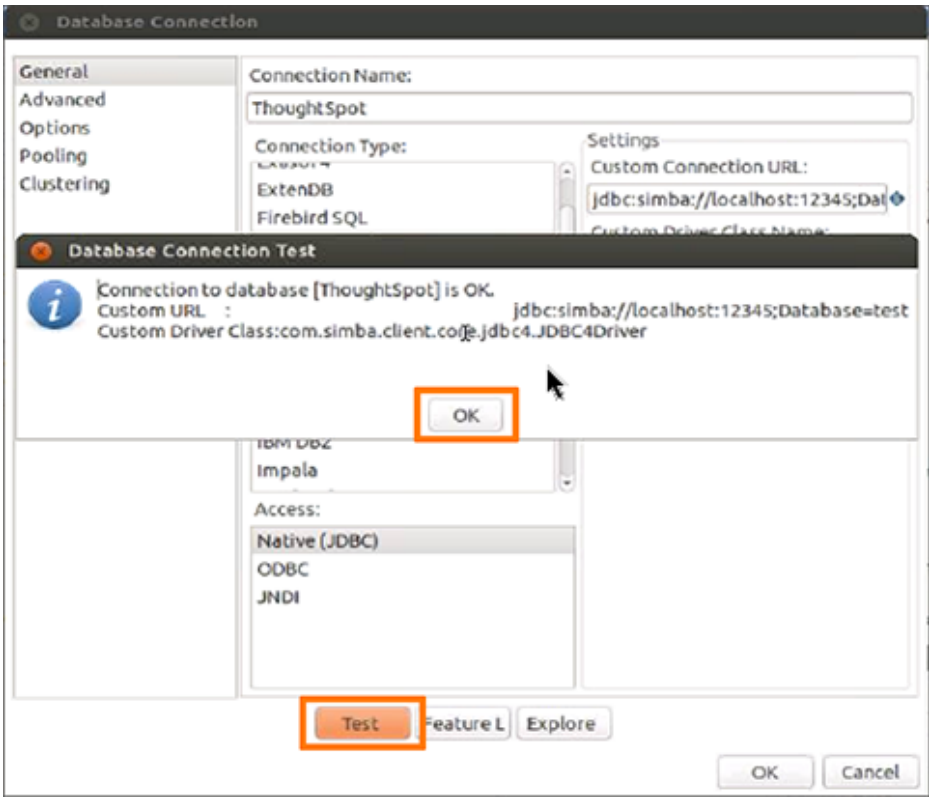
Field	Description
Connection Name	Any string.
Connection Type	Generic database
Access	Native (JDBC)
Custom Connection URL	`jdbc:simba://SERVER_IP:12345;Database=DATABASE_or_SCHEMA_NAME </code>` The IP is a node in your ThoughtSpot cluster. The name or schema of the database you want to connect to. Use TQL to create a database name if needed. Ensure that there are no leading or trailing spaces.

Custom Driver Class Name	`com.simba.client.core.jdbc4.JDBC4Driver` Ensure that there are no leading or trailing spaces.
User Name	A ThoughtSpot username. If you leave this empty, you are prompted for it at connection time. This user should have **Data Management** privileges on ThoughtSpot.
Password	The password for the **User Name**. If you leave this empty, you are prompted for it at connection time.



7. Click **Test** to test your database connection.

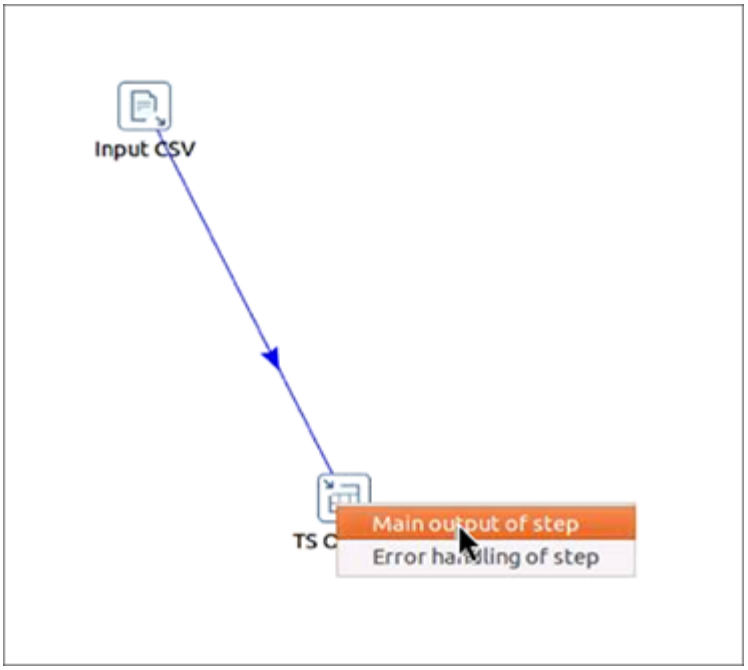
8. If you are able to make a successful connection to the ThoughtSpot Simba Server, click **OK**.



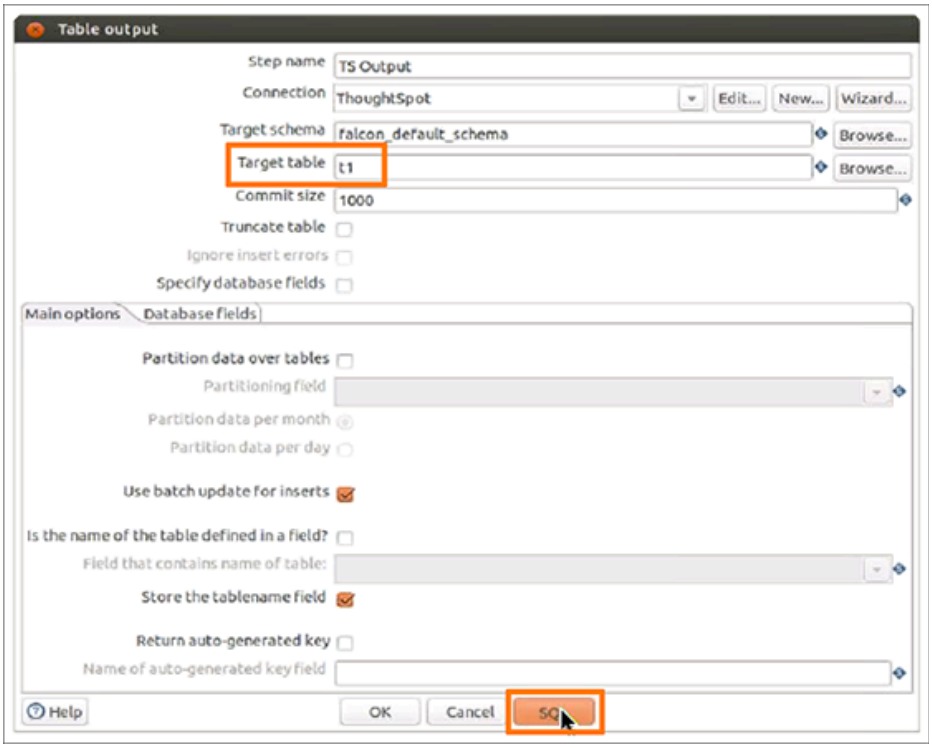
9. Click **OK** in the Database Connection dialog to create the new connection.

Import data

1. In the **Table output** dialog, select the connection you just created.
2. Click **Browse** next to the **Target schema** field and select your **Target schema**.
3. Click **OK** when you are done.
4. Connect the **Input CSV** icon to the **Table output** icon by clicking and dragging an arrow.
5. When prompted, choose **Main output of step**.

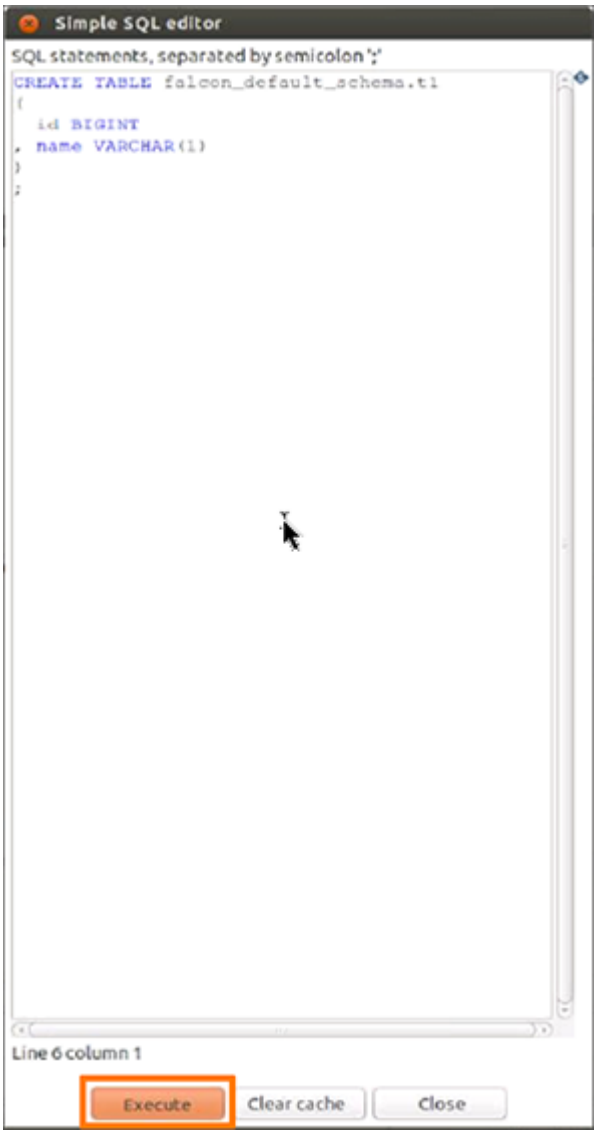


6. Double click the **Table output** icon to reopen the **Table output** dialog.
7. Enter a **Target table name**.
8. Click **SQL**.



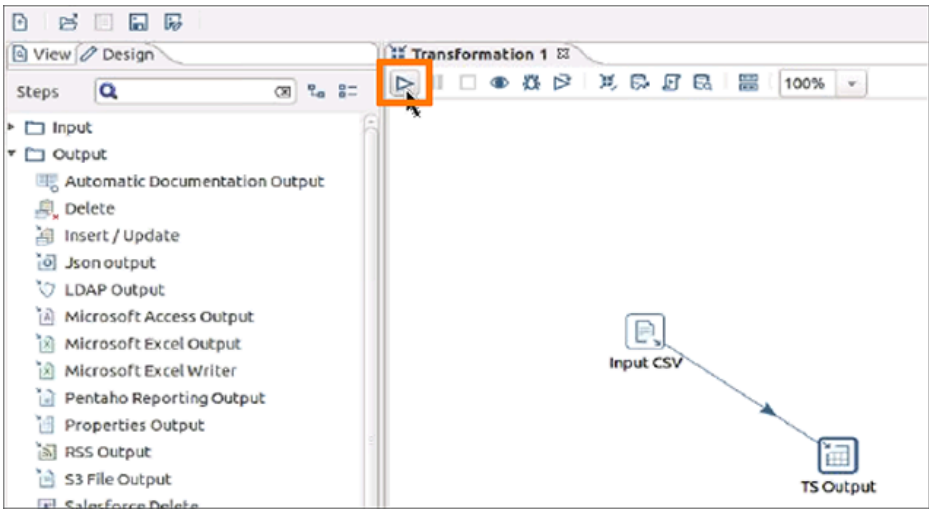
9. In the **Simple SQL editor** dialog, click **Execute**.

   The system processes and then displays the results of the SQL statements.

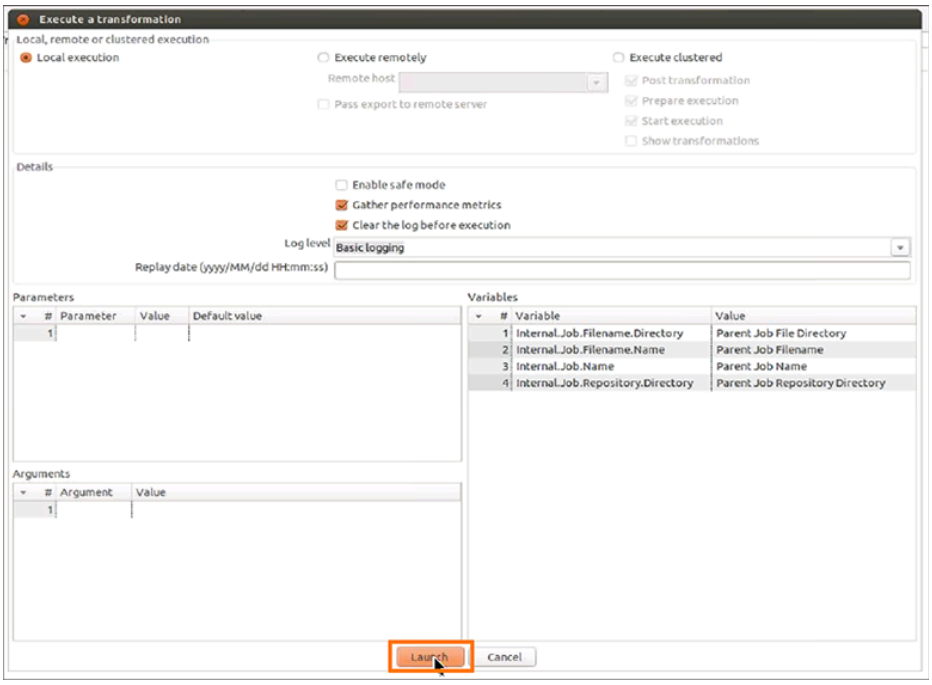


10. Close all open dialogs.

11. Click the **Play** button at the top of the **Transformation** window to execute the transformation.

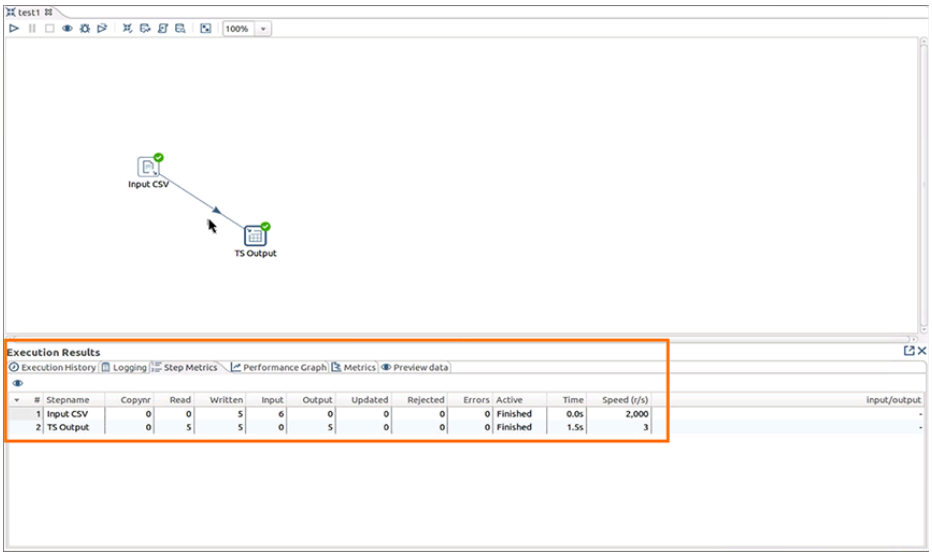


12. Click **Launch** in the **Execute a transformation** dialog.



The system prompts you to save it if you have not already.

13. View the **Execution Results**.


test1
100%
Input CSV
TS Output
Execution Results
Execution History
Logging
Step Metrics
Performance Graph
Metrics
Preview data
Stepname Copynr Read Written Input Output Updated Rejected Errors Active Time Speed (r/s) input/output
1 Input CSV 0 0 5 6 0 0 0 0 Finished 0.0s 2,000 -
2 TS Output 0 5 5 0 5 0 0 0 Finished 1.5s 3 -

Troubleshooting Data Integrations

Summary: Learn how to fix connection issues.

This section can help if you're having trouble creating a connection or need to find out more information about what is going on with ODBC or JDBC.

The information contained here is very basic, and mostly about how to enable logs on the client side. If you need more detailed troubleshooting information or help, please contact ThoughtSpot Support.

- **Enable ODBC Logs**

  If you need more information in order to troubleshoot ODBC connections, you can enable logging for ODBC.
- **Enable JDBC Logs**

  To enable logging for JDBC, add the logging parameters to the connect string. Logs are stored on ThoughtSpot.
- **Schema not found error with ODBC**

  When connecting with ODBC, you need to specify both the database and schema to connect to. If no schema is supplied, you will get an error indicating that the schema could not be found.
- **How to improve throughput of the load**

  The transaction/commit size value can improve the throughput of the load when setting up the ODBC Driver.
- **ODBC tracing on Windows**

  Using logs to aid in troubleshooting.

Enable ODBC Logs

Summary: Learn how to troubleshoot ODBC connections.

If you need more information in order to troubleshoot ODBC connections, you can enable logging for ODBC on the workstation you use for connecting to ThoughtSpot. There are two points where you can enable logging:

- the workstation where you run your ETL activities
- the server where the Simba service is running

On both workstation and servers, the verbosity of the log is controlled by the `LogLevel` property. This property can be one of the following:

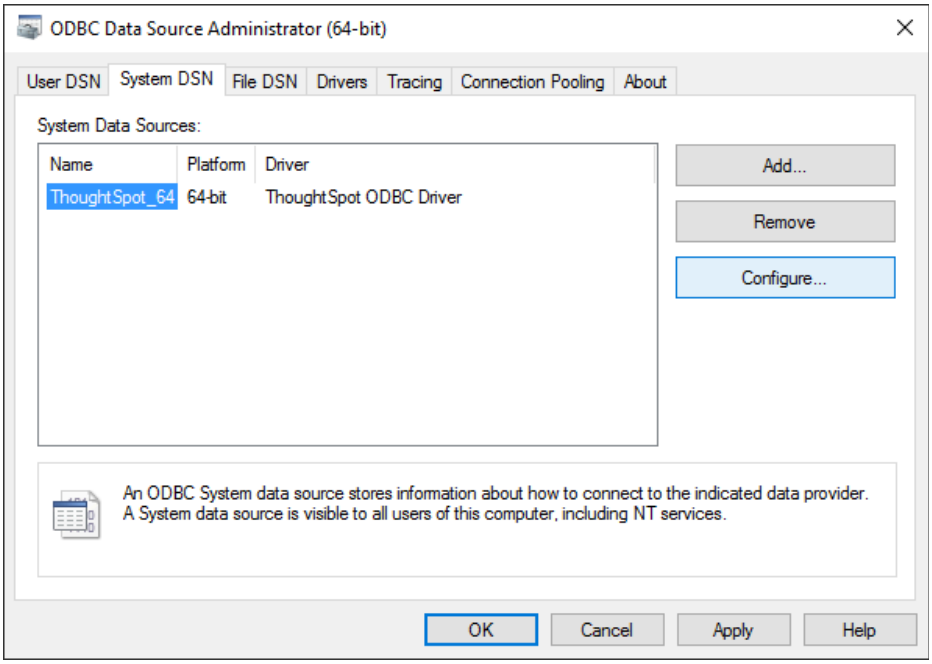
- `0` or `LOG_OFF`: no logging occurs
- `1` or `LOG_FATAL`: only log fatal errors
- `2` or `LOG_ERROR`: log all errors
- `3` or `LOG_WARNING`: log all errors and warnings
- `4` or `LOG_INFO`: log all errors, warnings, and informational messages
- `5` or `LOG_DEBUG`: log method entry and exit points and parameter values for debugging
- `6` or `LOG_TRACE`: log all method entry points

Larger values include the information from lessor values. For example, if you set `3` or `LOG_WARNING`, you log all warnings *and* all errors.

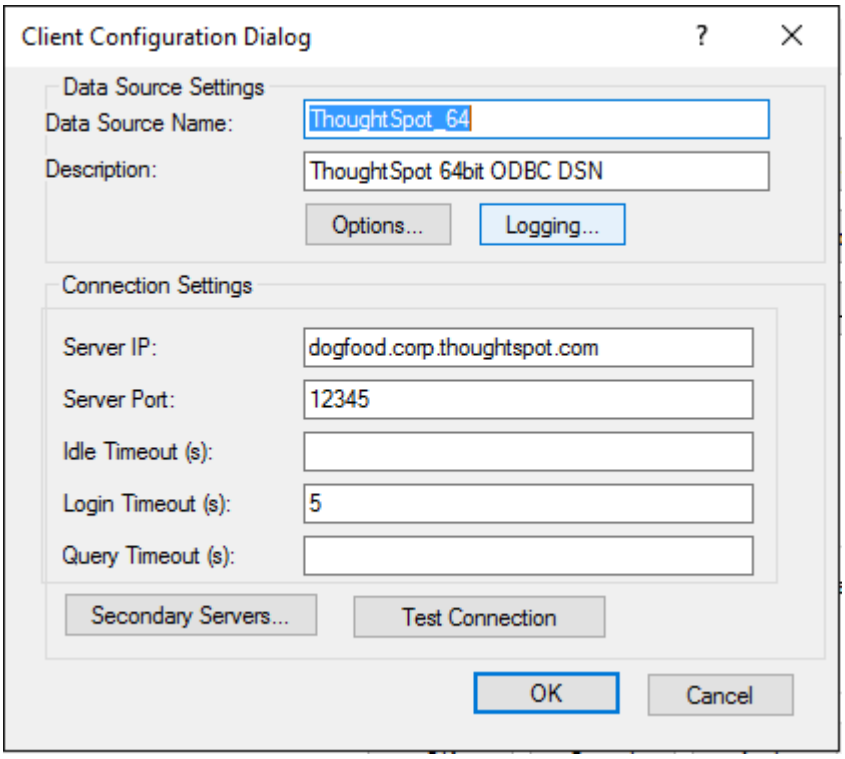
Enable ODBC logs on a Windows workstation

To enable ODBC logs on Windows:

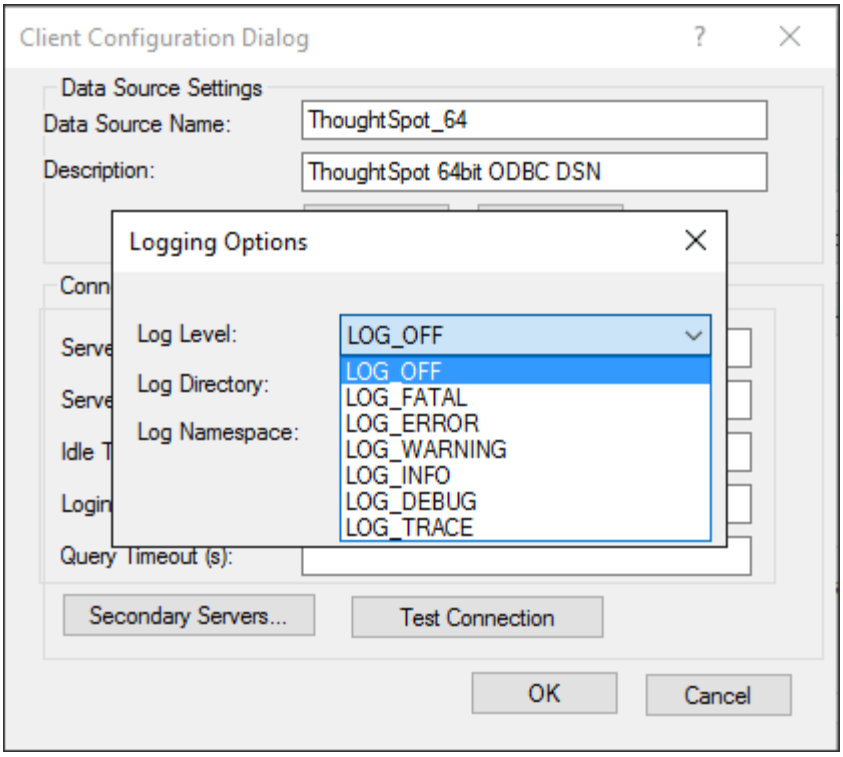
1. Open the **ODBC Data Source Administrator** and select the **System DSN** tab.
2. Select your ThoughtSpot data source and click **Configure**.



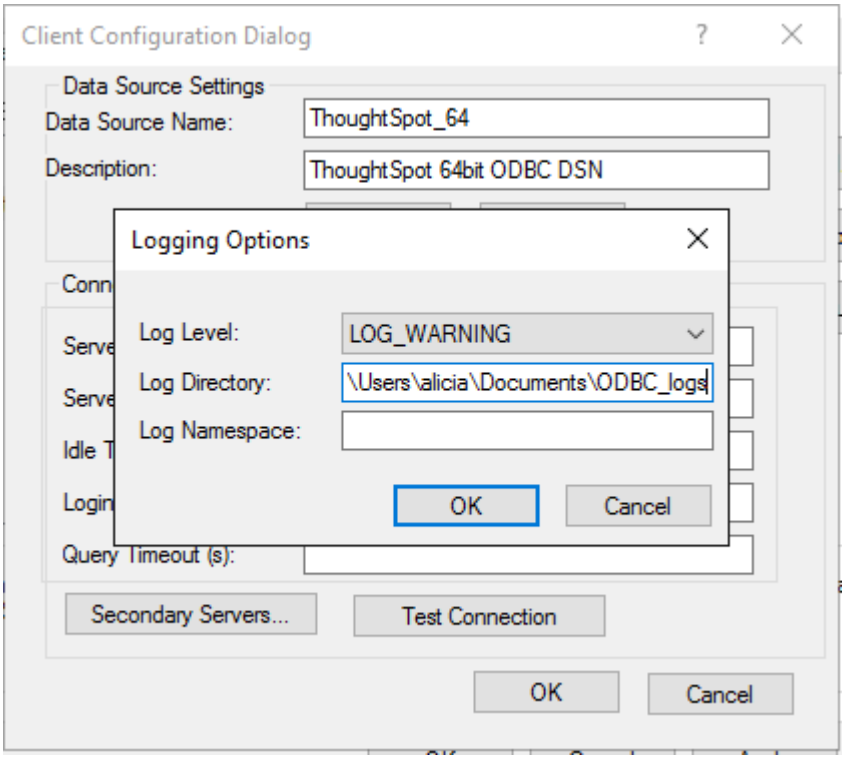
3. In the Client Configuration Dialog, click **Logging**.



4. Choose a **Log Level**, depending on what level of verbosity you want to show in the logs.



5. For **Log Directory:**, type in the fully qualified path where you want the logs to be saved.



6. Click **OK** to save your settings, and **OK** again, to dismiss the ODBC Data Source Administrator.
7. Run the ODBC load.
8. Locate the log file that was generated, and send it to ThoughtSpot Support with a description of the problem.

Enable ODBC logs on a Linux workstation

To enable logging on Linux, follow these instructions:

1. Navigate to the directory where you installed ODBC.
2. Open the `odbc.ini` file in a text editor.

   This file is the registry and configuration file for ODBC.
3. Locate the `LogLevel` and `LogPath` properties.
4. Uncomment the properties.
5. Enter a value for the `LogLevel` the.

   Acceptable values are from 1 to 6 with 6 being the most verbose.
6. Enter the fully qualified path for the `LogPath` values.

   The log will be written here. Your file will look similar to the following: Example for Linux 64-bit:

```
[ThoughtSpot]
Description = ThoughtSpot 64-bit ODBC Driver
Driver = ThoughtSpot
ServerList =  172.18.231.17 12345
Locale = en-US
ErrorMessagesPath = /home/admin/linux/ErrorMessages
UseSsl = 0
#SSLCertFile = # Set the SSL certificate file path. The
certificate file can be obtained by extracting the SDK t
arball
LogLevel = 3 # Set log level to enable debug logging
LogPath = /home/admin/odbc-logs # Set the debug log file
s path
DATABASE = # Set the default database to connect to
SCHEMA = # Set the default schema to connect to
```

7. Save and close the file.
8. To test the configuration, run the ODBC load and review the log files.

Control logs from the Simba server

You may want to collect logs from the Simba service. Do the following to procedure on every ThoughtSpot node running the Simba service.

1. SSH into the ThoughtSpot node.
2. Edit the `/etc/thoughtspot/linux.ini` file.

```
 ...
 [Driver]

 ## Note that this default DriverManagerEncoding of UT
F-32 is for iODBC. unixODBC uses UTF-16 by default.
 ## If unixODBC was compiled with -DSQL_WCHART_CONVERT,
then UTF-32 is the correct value.
 ## Execute 'odbc_config --cflags' to determine if you n
eed UTF-32 or UTF-16 on unixODBC
 DriverManagerEncoding=UTF-32
 DriverLocale=en-US
 ErrorMessagesPath=/usr/home/linux/ErrorMessages/
 LogLevel=0
 LogNamespace=
 LogPath=

 ....
```

3. Uncomment the `LogLevel` setting.

   The `LogLevel` is the level of logging to capture (0-6).

4. Set `LogPath` to a directory to save the logs.

   The `LogPath` is the fully qualified path where ThoughtSpot should write the logs.

5. Work with ThoughtSpot Support to restart the Simba serivce.

   The node IP may change because of the restart. If this happens, repeat the entire procedure.

Enable JDBC Logs

Summary: Configure logging parameter strings.

To enable logging for JDBC, add the logging parameters to the connect string. Logs are stored on ThoughtSpot. Before enabling JDBC logging, you need:

- The level of logging you want to capture.
- The path on the ThoughtSpot server where the logs will be written. Make sure the directory has the correct permissions so that the “admin” Linux user can write logs to it.

To enable JDBC logging:

1. When forming the connect string for JDBC, add these two parameter, separated by “&”:

   For example:

   ```
    jdbc:simba://192.168.2.248:12345;SERVERS=192.168.2.24
   9:12345,
    192.168.2.247:12345;Database=test;Schema=falcon_defaul
   t_schema;**LogLevel=3;LogPath=/usr/local/scaligent/log
   s**
   ```

   The `LogLevel` is the level of logging to capture (0-6). The `LogPath` is the fully qualified path where logs will be written on ThoughtSpot.

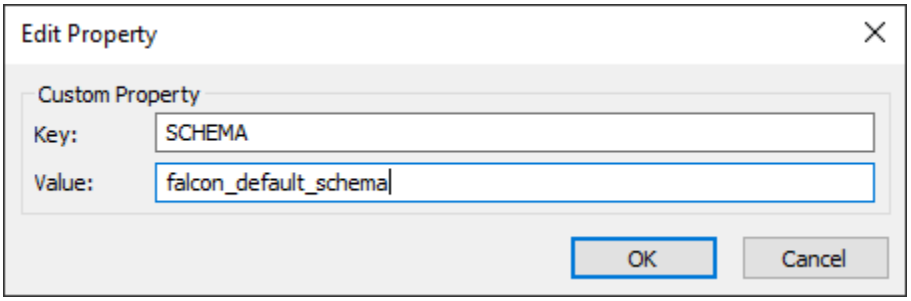
2. Run the JDBC code that uses the connection you modified.
3. Check the `LogPath` directory for logs generated by JDBC.

Schema not found error with ODBC

> **Summary:** Correct schema not found errors.

When connecting with ODBC, you need to specify both the `DATABASE` and `SCHEMA` parameters. This is true even if you do not use schema names in ThoughtSpot. If you don't supply a `SCHEMA`, you get an error indicating that the schema could not be found.

The default schema name in ThoughtSpot is `falcon_default_schema`. To set the `SCHEMA` on Windows, adding a custom property with the key `SCHEMA` and the value `falcon_default_schema`.



On Linux, you can edit the properties in the `odbc.ini` file for the driver you are using:

```
[ThoughtSpot]
Description = ThoughtSpot 64-bit ODBC Driver
Driver = ThoughtSpot
ServerList =  172.18.231.17 12345
Locale = en-US
ErrorMessagesPath = /home/admin/linux/ErrorMessages
UseSsl = 0
#SSLCertFile = # Set the SSL certificate file path. The certifi
cate file can be obtained by extracting the SDK tarball
#LogLevel = 0 # Set log level to enable debug logging
#LogPath = # Set the debug log files path
DATABASE = # Set the default database to connect to
SCHEMA = # Set the default schema to connect to
```

Related information

- Configuring ODBC on Windows
- Configuring ODBC on LINUX
- ODBC and JDBC configuration properties

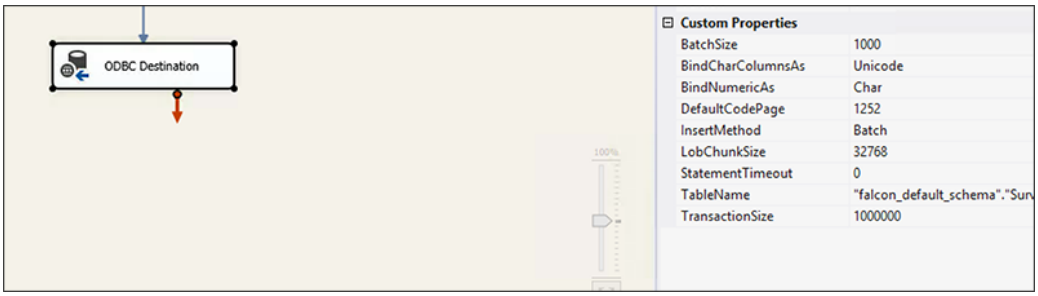
How to improve throughput

> **Summary:** Adjusting the transaction size may correct poor performance and low throughput.

The transaction/commit size value can improve the throughput of the load when setting up the ODBC Driver.

Adjusting the transaction size may correct poor performance and low throughput issues. The transaction size should be set to match the total number of rows that are expected to be loaded in the load cycle. However, increasing this value even higher should help improve throughput of the load.

> ❗ **Warning:** A high transaction size may slow down the ThoughtSpot system.



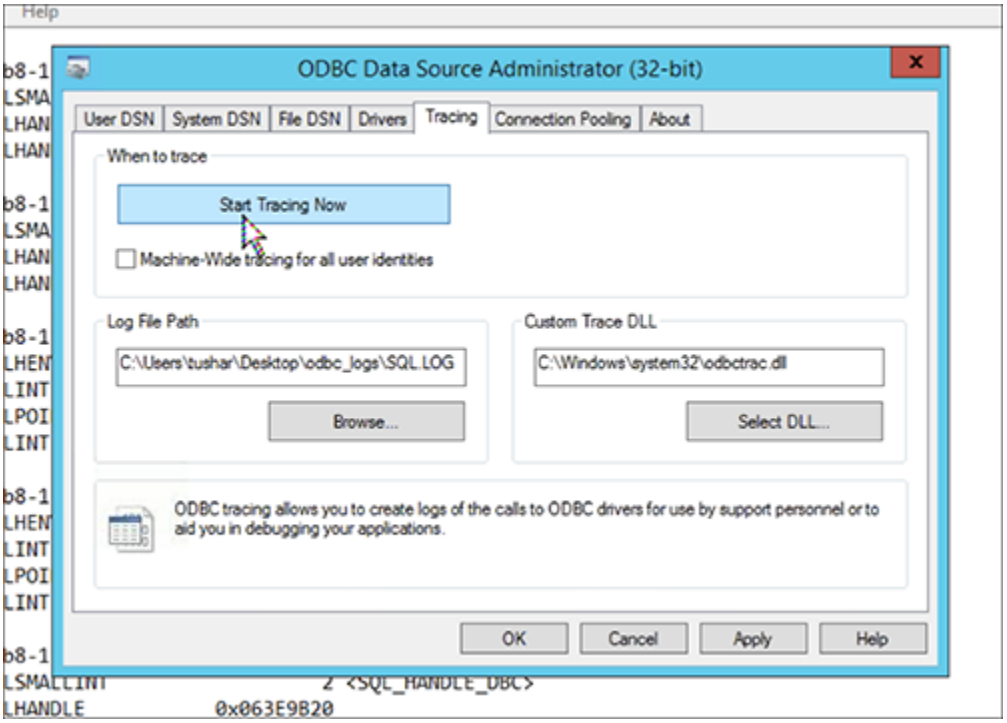
This is where the transaction size field exists for SSIS. Clicking on the ODBC destination reveals the properties on the right hand side, where the **Transaction Size** can be found.

See Set up the ODBC Driver for SSIS for more details on setting the transaction size.

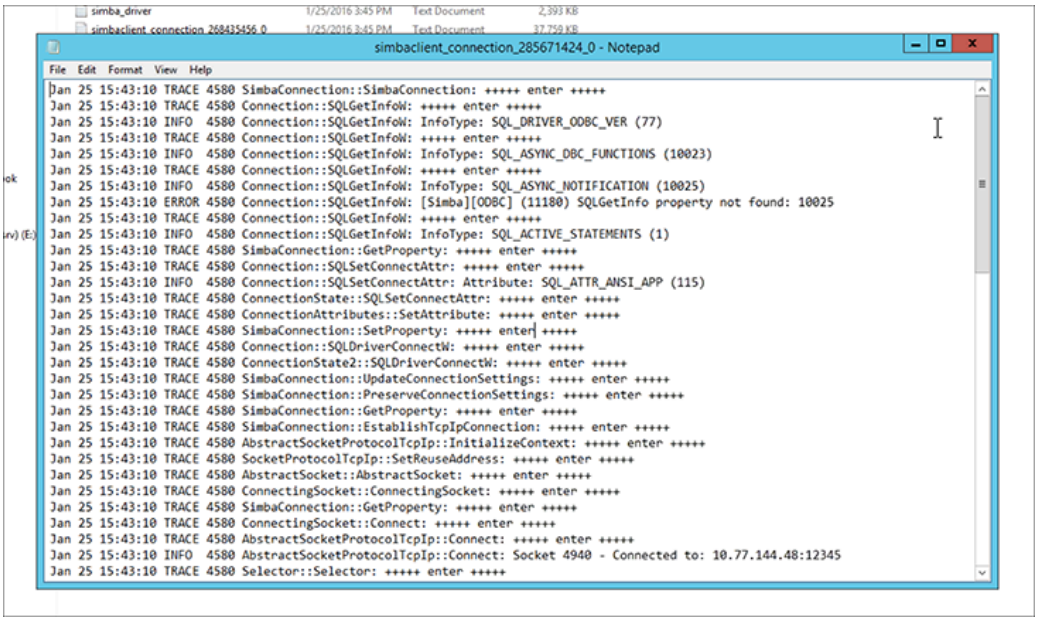
ODBC tracing on Windows

Summary: Using logs to aid in troubleshooting.

Windows shows ODBC specific tracing in the ODBC Data Source Administrator Tracing tab. You can start tracing there by clicking **Start Tracing Now**. This logs every ODBC call from this system, and prints the input and output for the call.



Although this is lower level information, it can still be helpful in troubleshooting. When you are not sure if it is our driver or the tool causing an issue, doing this trace will help narrow the inquiry.



If you start or stop tracing, make sure you do not have the SSIS client open. Close it, change the trace, and reopen.

Supported SQL commands

> **Summary:** The ThoughtSpot connection drivers support a limited set of SQL commands.

The ODBC and JDBC drivers support a limited set of SQL commands. When developing software that uses a ThoughtSpot ODBC driver, use this reference of supported commands. This reference is intended for developers using other tools (ETL, etc.) to connect to ThoughtSpot through the ODBC or JDBC driver.

> **Note:** ThoughtSpot displays `VARCHAR` fields using lower case, regardless of what the original casing of your loaded data is.

ODBC

These SQL commands are supported for ODBC:

- `CREATE TABLE`

  Creates a table with the specified column definitions and constraints. The table is replicated on each node.

  ```
  CREATE TABLE country_dim (id_number int, country varch
  ar, CONSTRAINT PRIMARY KEY (id_number));
  ```

- `INSERT`

  Creates placeholders in the table to receive the data.

  ```
  INSERT INTO TABLE country_dim (?, ?);
  ```

- `DELETE FROM <table>`

Deletes `ALL` rows from the specified table. Use the `WHERE` clause to specify only certain rows to be deleted. Example: You could remove all data for sales before a certain date to free up space in ThoughtSpot.

```
  DELETE FROM country_dim;
```

- `SELECT <cols_or_expression> FROM <table_list> [WHERE <predicates>] [GROUP BY <expressions>] [ORDER BY <expressions>]`

  Fetches the specified set of table data.

```
  SELECT id_number, country FROM country_dim WHERE id_nu
mber > 200;
```

JDBC

`TRUNCATE` is not supported. Instead, use `DELETE FROM TABLE` which is functionally equivalent to “truncate table” in terms of table compression and so forth.

Connection configuration

> **Summary:** Lists the properties you can set for ODBC or JDBC connections

This section lists the properties you can set for ODBC or JDBC connections.

Setting Properties for ODBC

Not all the parameters Simba accepts are supported by the ThoughtSpot ODBC clients, and ThoughtSpot has added some properties, which are listed separately here. All configuration properties use the type String (text).

You can set these properties on Windows by using the ODBC Administrator client. For Linux, the properties are located in three files, depending on the property type:

Property Type	Location
DSN	`odbc.ini` file
Driver	`odbsinst.ini` file
SimbaSetting Reader	`simbaclient.ini` file

Setting Properties for JDBC

For JDBC, these properties are passed as key value pairs in the connect string. For more information, see Use the JDBC Driver.

Properties Reference

The following tables summarize the configuration properties.

Property	Type	Description
DATABASE	DSN or Driver	The default database to connect to.
SCHEMA	DSN or Driver	The default schema to connect to.
Description	DSN	A brief, human-readable description of the DSN. This describes the DSN to users who are deciding which DSN to use.
Driver	DSN or Driver	In the driver configuration location, Driver should contain the path to the driver binary. In the DSN configuration location, Driver could contain the path to the driver binary, or it could contain the driver entry in the registry.
IdleTimeout	DSN	The time to wait for a response from the server, in seconds. This property is optional, but SimbaClient will wait indefinitely for SimbaServer to respond to a request made to the server unless you specify a timeout period. IdleTimeout specifies how many seconds that SimbaClient will wait before aborting the attempt and returning to the application with an error. This timeout corresponds to ODBC's CONNECTION_TIMEOUT property and is only used when more specific timeouts, such as QUERY_TIMEOUT or LOGIN_TIMEOUT aren't applicable.
Locale	DSN	The connection locale. If this value is set, it overrides the driver-wide locale. For example, the driver-wide locale could be en-US. If the client would prefer fr-CA, it can set the connection locale to fr-CA. Values are composed of a 2-letter language code (in lower case), and an optional 2-letter country code (in upper case). If the country code is specified, it must be separated from the language code by a hyphen (-).
LoginTimeout	DSN	The timeout, in seconds, to wait for a response from the server when attempting to log in. A value of 0 means no timeout. The default value is 60.
QueryTimeout	DSN	The timeout, in seconds, to wait for a response from the server during Prepare, Execute, and ExecuteDirect. A value of 0 means no timeout. The default value is 60.
ServerList	DSN	A comma separated list of all servers (IP address and port number) to connect to. SimbaClient must be able to find SimbaServer on the network. This property enables server discovery. SimbaClient will try to make a network connection to the servers in the order specified until a connection is made.
LogLevel	SimbaSettingReader	Controls the granularity of the messages and events that are logged. With this keyword, you can control the amount of log output by controlling the kinds of events that are logged. Possible values (case sensitive): • 0 or LOG_OFF : no logging occurs • 1 or LOG_FATAL : only log fatal errors • 2 or LOG_ERROR : log all errors • 3 or LOG_WARNING : log all errors and warnings • 4 or LOG_INFO : log all errors, warnings, and informational messages • 5 or LOG_DEBUG : log method entry and exit points and parameter values for debugging • 6 or LOG_TRACE : log all method entry points

Property	Type	Description
`LogPath`	SimbaSetting Reader	Specifies the directory where the log files are created. For example: `LogPath=C:\Simba Technologies\Temp` If this value is not set, the log files are written to the current working directory of the SimbaClient.
`LogFileSize`	SimbaSetting Reader	The size of each log file, in bytes. The default values is 20971520 bytes. When the maximum size of the file is reached, a new file is created.
`LogFileCount`	SimbaSetting Reader	The number of log files to create. When the maximum number of log files has been created, the oldest file will be deleted and a new one created. The default value is 50.
`username`	UID	Part of a user username/password combination. This combination should correspond to a ThoughtSpot application user with permissions appropriate to your ETL requirements. Typically, this user is a user with data management or administrative privileges on the application.
`password`	Password	Part of a user username/password combination. This combination should correspond to a ThoughtSpot application user with permissions appropriate to your ETL requirements. Typically, this user is a user with data management or administrative privileges on the application.