



---

# ThoughtSpot Deployment Guide for Google Cloud Platform

*Release 6.0*

*February, 2020*

© COPYRIGHT 2015, 2020 THOUGHTSPOT, INC. ALL RIGHTS RESERVED.

910 Hermosa Court, Sunnyvale, California 94085

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

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

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

# Table of Contents

Configuration options .....	2
Set up ThoughtSpot in GCP .....	4
Configure ThoughtSpot nodes in GCP .....	16
Install ThoughtSpot clusters in GCP .....	23
The nodes.config file.....	31
Parameters of the nodes.config file .....	34
Using the cluster create command .....	37
Parameters of the cluster create command.....	45
Network policies.....	48

# GCP configuration options

**Summary:** ThoughtSpot can be deployed using several different GCP instance types.

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).

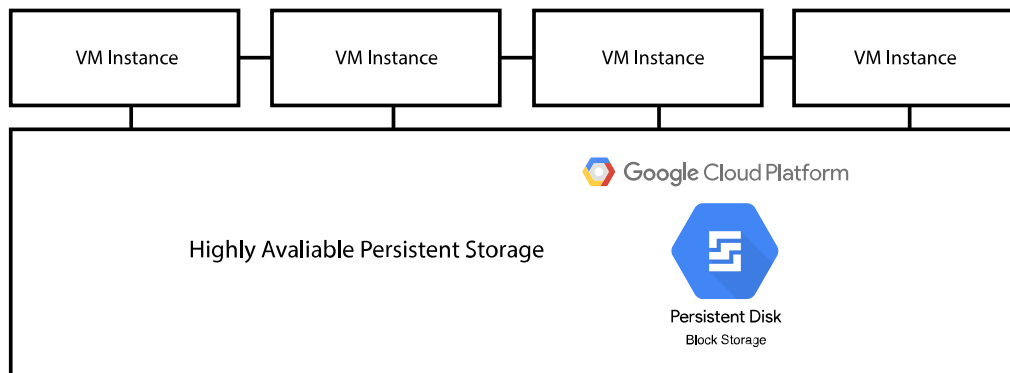
For more information about Persistent Storage, see [Zonal Persistent SSD disks \[See page 0\]](#) in Google's Cloud documentation.

For more information about Google Cloud Storage, see [Cloud Storage Buckets \[See page 0\]](#) in Google's Cloud documentation.

All GCP VMs (nodes) in a ThoughtSpot cluster must be in the same zone (and, therefore, also in the same region). ThoughtSpot does not support deploying VMs (nodes) of the same cluster across different zones. For more information, see [Regions and Zones \[See page 0\]](#) in Google's Cloud documentation.

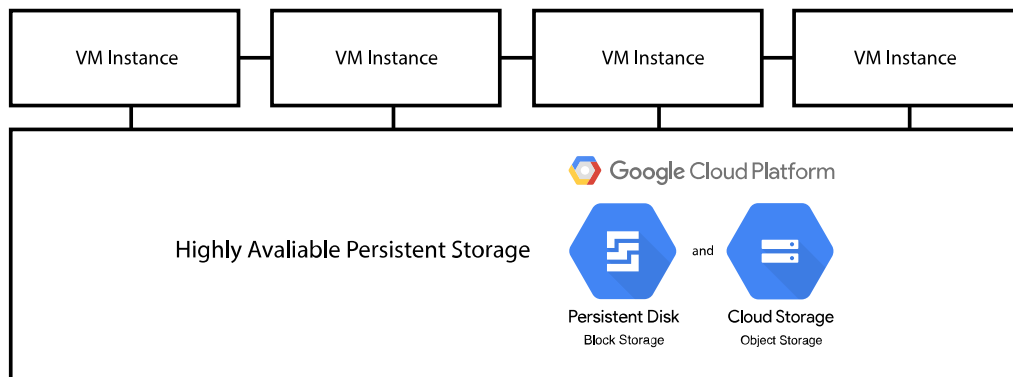
## ThoughtSpot GCP instance types

### VMs with Persistent Disk-only storage



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

### VMs with Persistent Disk and Google Cloud storage



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

# Set up ThoughtSpot in GCP

**Summary:** Set up your GCP virtual machines.

After you determine your configuration options, set up your virtual machines (VMs). ThoughtSpot will share the ThoughtSpot base image for booting the VMs and some other aspects of system setup with you on [the GCP platform](#) [See page 0].

## About the ThoughtSpot and Google Cloud Platform

ThoughtSpot uses a custom image to populate VMs in GCP. To find the ThoughtSpot custom image, refer to step 13 under [create an instance](#) [See page 6].

Ask your ThoughtSpot contact 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 to create ThoughtSpot VMs.

## Overview

Before you can create a ThoughtSpot cluster, you must set up your VMs. Use the Google Compute Engine (GCP) platform to create and run VMs.

The following topics walk you through this process.

## Prerequisites

1. Ensure that your **Network Service Tier** on the [Google Cloud Console](#) [See page 0] is set to **Premium** for the best performance of all your VMs.
2. A ThoughtSpot cluster requires 10 Gb/s bandwidth (or better) between any two nodes. You must ensure this *before* creating a new cluster.
3. [Download](#) [See page 0] and fill out the ThoughtSpot site survey to have a quick reference for your networking information. Ask your network administrator if you need help filling out the site survey.

## Setting up your Google Cloud Storage (GCS) bucket

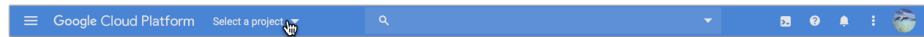
If you are going to deploy your cluster using the GCS-storage option, you must set up that bucket before you set up your cluster. Contact [ThoughtSpot Support \[See page 0\]](#) to find out if your specific cluster size will benefit from the GCS storage option. If you are not using GCS, skip this step and [create an instance \[See page 6\]](#).

1. Sign in to the [Google Cloud Console \[See page 0\]](#).
2. Go to the **Storage** dashboard from the navigation bar on the side of your screen.
3. Click **CREATE BUCKET** on the top menu bar.
4. Enter a name for your bucket, and click **CONTINUE**.
5. For location type, select **Region**.
6. Use the Location drop-down menu to select the region where you are going to set up your instance.
7. Click **CONTINUE**.
8. For default storage class, select **Standard**.
9. Click **CONTINUE**.
10. Under **Access Control**, select **Uniform** to ensure uniform access to all objects in the storage bucket.
11. Click **CONTINUE**.
12. Do not edit the advanced settings.  
Leave Encryption set to **Google-managed key** and do not set a retention policy.
13. Click **CREATE**.

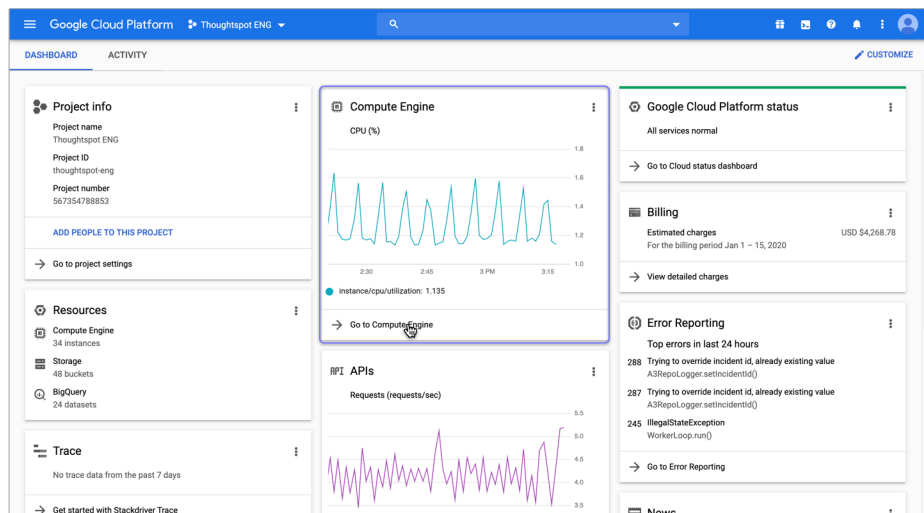
When you create your instance, make sure you set Storage to **Read Write** access.

## Create an instance

1. Sign in to the [Google Cloud Console](#) [See page 0].
2. Click **Select a Project** from the top bar.



3. Under **Select From**, pick your company's project.
4. Go to the Compute Engine dashboard.



5. Select **VM instances** on the left panel.
6. Click **CREATE INSTANCE** from the top menu bar.
7. Provide a name for the instance.
8. Select the region you are creating the instance in.
9. Select the zone you are creating your region in.
10. Under **Machine type**, select **custom**.



11. Select the number of CPUs you need.

Refer to [ThoughtSpot GCP instance types \[See page 2\]](#) to determine the number of CPUs your cluster needs.

12. Specify your memory requirements and CPU platform. Refer to [ThoughtSpot GCP instance types \[See page 2\]](#) to determine the memory your cluster needs.

Your configuration may look something like the following, but with your specific information.

Setting	Value
Cores	64 vCPU
Memory	416 GB
CPU platform	Automatic (or select either one of the preferred CPU platforms, Intel Skylake or Intel Broadwell, if available.)

**Machine family**

General-purpose
Memory-optimized

Machine types for common workloads, optimized for cost and flexibility

**Series**

N1

Powered by Intel Skylake CPU platform or one of its predecessors

**Machine type**

Custom

**Cores**

64 vCPU

1 - 96

**Memory**

416 GB

57.75 - 416

☐ Extend memory ?



13. Configure the Boot disk.

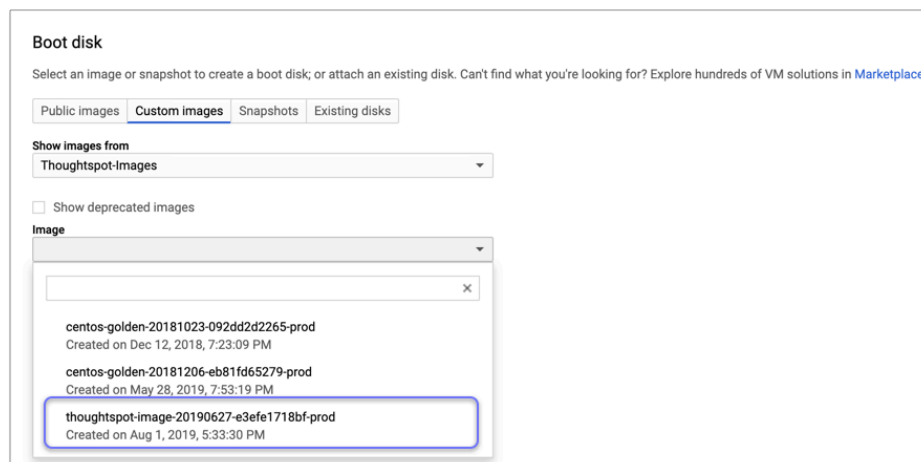
a. Scroll down to the **Boot disk** section and click **Change**.



b. Click **Custom Images** from the options under **Boot disk**.

c. Select **ThoughtSpot-images** under **Show images from**.

d. Select one of the ThoughtSpot base images. Under the name of the image, you can see when it was created. Select the latest image. ThoughtSpot may have directly sent you an image to use through the console. If so, use that image.



**Note:** ThoughtSpot updates these base images with patches and enhancements. If more than one image is available, select the latest one by looking at the dates of creation. Each image will work, but we recommend using the latest image because it typically contains the latest security and maintenance patches. Contact ThoughtSpot Support if you are unsure which image to use.

e. Configure the boot disk as follows:

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

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

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

**Boot disk** ?

New 250 GB standard persistent disk  
Image  
thoughtspot-1536332609 Change

**Identity and API access** ?

You don't have permission to view the service accounts in this project

**Service account** ?  
Compute Engine default service account

**Access scopes** ?

☒ Allow default access  
☐ Allow full access to all Cloud APIs  
☐ Set access for each API

**Firewall** ?  
Add tags and firewall rules to allow specific network traffic from the Internet

☐ Allow HTTP traffic  
☐ Allow HTTPS traffic

Management, security, disks, networking, sole tenancy

15. Attach two 1 TB SSD drives for data storage. Refer to [SSD-only persistent storage](#) [See page 2]. If you are using GCS, attach only 1 SSD drive, with 500 GB instead of 1 TB. Refer to [GCS and SSD persistent storage](#) [See page 3].

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

Management Security **Disks** Networking Sole Tenancy

**Boot disk**

**Deletion rule**

☒ Delete boot disk when instance is deleted

**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

**Additional disks** ⓘ (Optional)

**+ Add new disk** **+ Attach existing disk**

⤴ Less

You can select or unselect the **Deletion rule**, depending on your preferences.

b. Configure the following settings for each disk. Refer to [ThoughtSpot GCP instance types](#) [See page 3] to determine the size in GB when you have GCS. Ensure the disks have read/write access.

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

Under **Deletion rule**, select either **keep disk** or **delete disk**, depending on your preference.

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

16. (For use with GCS only) In the Identity and API access section, make sure Service account is set to **Compute Engine default service account**. Under Access scopes, select **Set access for each API**.

17. (For use with GCS only) After you click **Set access for each API**, scroll down to the **Storage** dropdown menu in the Identity and API access section. 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**.
18. Under **Networking**, customize the network settings as needed. Use your default VPC settings, if you know them. Ask your network administrator if you do not know your default VPC settings.

Update the network interface with your specific information or create a new one.

- 1 Add an existing VPC network, or create a new one by clicking **VPC network** from the main menu. Ensure that this network has a **firewall rule** attached, with the minimum ports required for ThoughtSpot operation open. Refer to the [minimum port requirements](#). [See page 14] See Google's [using firewalls](#) [See page 0] and [using VPCs](#) [See page 0] documentation for assistance creating a firewall rule and a VPC network.
- 2 Set the external IP as either ephemeral or static, depending on your preference.
- 3 Ensure that **network service tier** is set to **premium**.

19. Repeat these steps to create the necessary number of VMs for your cluster.

## Minimum required ports

Open the following ports between the User/ETL server and ThoughtSpot nodes. This ensures that the ThoughtSpot processes do not get blocked. Refer to [network policies \[See page 0\]](#) for more information on what ports to open for intracluster operation, so that your clusters can communicate.

The minimum ports needed are:

Port	Protocol	Service
22	SSH	Secure Shell access
443	HTTPS	Secure Web access
12345	TCP	ODBC and JDBC drivers access

## Prepare the VMs

Before you can install your ThoughtSpot cluster, an administrator must log in to each VM through SSH as user “admin”, and complete the following preparation steps:

1. Open a terminal application on your machine and ssh into one of your VMs.

```
ssh admin@<VM-IP>
```

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

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

3. Configure the VM based on the site-survey.
4. Repeat this process for each of your VMs.



## Install cluster

To install your ThoughtSpot cluster, complete the installation process outlined in [Installing ThoughtSpot in GCP](#) [See page 16].

## Related information

[Connecting to Google Cloud Storage buckets](#) [See page 0]

[Loading data from a GCP GCS bucket](#) [See page 0]

# Configure ThoughtSpot nodes in GCP

**Summary:** Prepare to install your ThoughtSpot cluster by configuring nodes.

Before you can install a ThoughtSpot cluster in GCP, you must configure your nodes.

## Installation Prerequisites

Ensure the successful creation of the virtual machines (VMs) before you install the ThoughtSpot cluster in GCP.

1. **Review configuration options** Refer to [GCP configuration options \[See page 2\]](#) for detailed instance specs.
2. **Create the instance** Refer to [Set up GCP for ThoughtSpot \[See page 4\]](#) to create and launch your instance.
3. **Review required ports** Refer to [Network Policies \[See page 48\]](#) to view the required ports for successful operation of ThoughtSpot.

## Configure Nodes

After creating the instance, you must configure the nodes. Follow the steps in this checklist.

- ☐ Step 1: Log in to your cluster [\[See page 0\]](#)
- ☐ Step 2: Get a template for network configuration [\[See page 0\]](#)
- ☐ Step 3: Prepare node configuration [\[See page 0\]](#)
- ☐ Step 4: Configure the nodes [\[See page 0\]](#)
- ☐ Step 5: Confirm node configuration [\[See page 0\]](#)

### Step 1: Log in to your cluster

Log in to your cluster with admin credentials from Terminal on a Mac or a terminal emulator on Windows.

Ask your network administrator if you do not know the admin credentials.

1. Run `ssh admin@<nodeIP> .`

Replace `nodeIP` with your specific network information.

```
$ ssh admin@<nodeIP>
```

2. Enter your admin password at the prompt.

Ask your network administrator if you don't know the password.

**Note:** The password does not appear on the screen as you type it.

## Step 2: Get a template for network configuration

Run the `tscli cluster get-config` command to get a template for network configuration for the new cluster. Redirect it to the file `nodes.config`.

You can find more information on this process in the [nodes.config file reference \[See page 31\]](#).

```
$ tscli cluster get-config |& tee nodes.config
```

## Step 3: Prepare node configuration

1. Add your specific network information for the nodes in the `nodes.config` file, as demonstrated in the [autodiscovery of one node example \[See page 33\]](#). Run `vim nodes.config` to edit the file.

```
$ vim nodes.config
```

**Note:** Some of the information in the `nodes.config` file may be pre-populated from earlier steps. For example, if you specified an IP address while creating VMs, that IP address might already be present in your `nodes.config` file.

2. Fill in the areas specified in [Parameters of the nodes.config file \[See page 34\]](#) with your specific network information.

If you have additional nodes, complete each node within the `nodes.config` file in the same

way.

Do not edit any part of the `nodes.config` file except the sections described in [Parameters of the nodes.config file \[See page 34\]](#). If you delete quotation marks, commas, or other parts of the code, it may cause setup to fail.

## Step 4: Configure the nodes

Configure the nodes in the `nodes.config` file using the `set-config` command.

1. Disable the `firewalld` service by running `sudo systemctl stop firewalld` in your terminal. The `firewalld` service is a Linux firewall that must be off for ThoughtSpot installation. After the cluster installer reboots the nodes, `firewalld` automatically turns back on.

```
$ sudo systemctl stop firewalld
```

2. To make sure you temporarily disabled `firewalld`, run `sudo systemctl status firewalld`. Your output should specify that `firewalld` is inactive. It may look something like the following:

```
$ sudo systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
   Active: inactive (dead)
```

3. Run the configuration command: `$ cat nodes.config | tscli cluster set-config`.  
If the command returns an error, refer to [set-config error recovery \[See page 20\]](#).  
After you run the node configuration command, your output appears similar to the following:

```
$ cat nodes.config | tscli cluster set-config

Connecting to local node-scout
Setting up hostnames for all nodes
Setting up networking interfaces on all nodes
Setting up hosts file on all nodes
Setting up NTP Servers
Setting up Timezone
Done setting up ThoughtSpot
```

### Step 5: Confirm node configuration

Use the `get-config` command to confirm node configuration.

Your output may look similar to the following:

```
$ tscli cluster get-config

{
  "ClusterId": "",
  "ClusterName": "",
  "DataNetmask": "255.255.252.0",
  "DataGateway": "192.168.4.1",
  "IPMINetmask": "255.255.252.0",
  "IPMIGateway": "192.168.4.1",
  "Timezone": "America/Los_Angeles",
  "NTPServers": "0.centos.pool.ntp.org,1.centos.pool.ntp.org,2.centos.pool.ntp.org,3.centos.pool.ntp.org",
  "DNS": "192.168.2.200,8.8.8.8",
  "SearchDomains": "example.company.com",
  "Nodes": {
    "ac:1f:6b:8a:77:f6": {
      "NodeId": "ac:1f:6b:8a:77:f6",
      "Hostname": "Thoughtspot-server1",
      "DataIface": {
        "Name": "eth2",
        "IPv4": "192.168.7.70"
      },
      "IPMI": {
        "IPv4": "192.168.5.70"
      }
    }
  }
}
```

## Install ThoughtSpot software

Next, [install your ThoughtSpot clusters](#) [See page 23].

## Error recovery

### Set-config error recovery

If you get a warning about node detection when you run the `set-config` command, restart the node-scout service.

Your error may look something like the following:

```
Connecting to local node-scout WARNING: Detected 0 nodes, but found configuration for only 1 nodes.  
Continuing anyway. Error in cluster config validation: [] is not a valid link-local  
IPv6 address for node: 0e:86:e2:23:8f:76 Configuration failed.  
Please retry or contact support.
```

Restart the node-scout service with the following set of commands.

```
$ sudo vim /etc/systemd/system/node-scout.service  
$ sudo systemctl daemon-reload  
$ sudo systemctl restart node-scout
```

Ensure that you restarted the node-scout by running `sudo systemctl status node-scout`. Your output should specify that the node-scout service is active. It may look something like the following:

```
$ sudo systemctl status node-scout  
● node-scout.service - Setup Node Scout service  
   Loaded: loaded (/etc/systemd/system/node-scout.service; enabled; vendor preset: disabled)  
   Active: active (running) since Fri 2019-12-06 13:56:29 PST; 4s ago
```

Next, retry the `set-config` command.

```
$ cat nodes.config | tscli cluster set-config
```

The command output should no longer have a warning.

## Related information

Use these references for successful installation and administration of ThoughtSpot.

- [The nodes.config file \[See page 0\]](#)

- [Parameters of the nodes.config file \[See page 34\]](#)
- [Using the tscli cluster create command \[See page 37\]](#)
- [Parameters of the `cluster create` command \[See page 45\]](#)
- [ThoughtSpot Documentation \[See page 0\]](#)
- [Contact Support \[See page 0\]](#)



# Install ThoughtSpot clusters in GCP

**Summary:** Learn how to install ThoughtSpot clusters in GCP.

## Prerequisites

Before you can install your ThoughtSpot clusters in GCP, complete these prerequisites.

1. **Review configuration options** Refer to [GCP configuration options \[See page 2\]](#) for detailed instance specs.
2. **Create the instance** Refer to [Set up ThoughtSpot in GCP \[See page 4\]](#) to create and launch your instance.
3. **Review required ports** Refer to [Network Policies \[See page 48\]](#) to view the required ports for successful operation of ThoughtSpot.
4. **Configure nodes** Refer to [Configure ThoughtSpot nodes in GCP \[See page 16\]](#) to configure your nodes.

## Install ThoughtSpot Software

Install the cluster using the ThoughtSpot software release bundle. The estimated installation time is one hour. Follow the steps in this checklist.

- ☐ [Step 1: Run the installer \[See page 0\]](#)
- ☐ [Step 2: Check cluster health \[See page 0\]](#)
- ☐ [Step 3: Finalize installation \[See page 0\]](#)

Refer to your welcome letter from ThoughtSpot to find the link to download the release bundle. If you do not have a link, open a support ticket at [ThoughtSpot Support \[See page 0\]](#) to request access to the release bundle.

### Step 1: Run the installer

1. Copy the downloaded release bundle to `/export/sdc1/TS_TASKS/install` using the following command:

```
$ scp <release-number>.tar.gz admin@<hostname>:/export/
sdcl/TS_TASKS/install/<file-name>
```

Note the following parameters:

- `release-number` is the release number of your ThoughtSpot instance, such as 5.3, 6.0, and so on.
- `hostname` is your specific hostname.
- `file-name` is the name of the tarball file on your local computer.

**Note:** You can use another secure copy method, if you prefer a method other than the `scp` command.

- Alternatively, use `tscli fileserver download-release` to download the release bundle. You must [configure the fileserver](#) [See page 0] by running `tscli fileserver configure` before you can download the release.

```
$ tscli fileserver download-release <release-number> --
user <username> --out <release-location>
```

Note the following parameters:

- `release-number` is the release number of your ThoughtSpot instance, such as 5.3, 5.3.1, 6.0, and so on.
- `username` is the username for the fileserver that you set up earlier, when configuring the fileserver.
- `release-location` is the location path of the release bundle on your local machine. For example, `/export/sdcl/TS_TASKS/install/6.0.tar.gz`.

- Verify the checksum to ensure you have the correct release.

Run `md5sum -c <release-number>.tar.gz.MD5checksum`.

```
$ md5sum -c <release-number>.tar.gz.MD5checksum
```

Your output says `ok` if you have the correct release.

4. Launch a [screen](#) [See page 0] session. Use screen to ensure that your installation does not stop if you lose network connectivity.

```
$ screen -S DEPLOYMENT
```

5. Create the cluster.

Run `tscli cluster create` to create the cluster.

If you are using a gcs bucket for object storage, include the flag `--enable_cloud_storage=gcs`.

```
$ tscli cluster create <release-number>.tar.gz --enable_cloud_storage=gcs
```

6. Edit the output with your specific cluster information.

For more information on this process, refer to [Using the tscli cluster create command](#) [See page 37] and [Parameters of the cluster create command](#) [See page 45].

The cluster installer automatically reboots all the nodes after a successful install. The `firewalld` service automatically turns on. At this time, the system is rebooting, which may take approximately 15 minutes.

Log in to any node to check the current cluster status:

```
$ tscli cluster status
```

## Step 2: Check cluster health

After the cluster installs, check its status using the `tscli cluster status` command.

Your output may look similar to the following:

```
$ tscli cluster status
Cluster: RUNNING
Cluster name      : thoughtspot
Cluster id       : 1234X11111
Number of nodes  : 3
Release          : 6.0
Last update      = Wed Oct 16 02:24:18 2019
Heterogeneous Cluster : False
Storage Type     : HDFS

Database: READY
Number of tables in READY state: 2185
Number of tables in OFFLINE state: 0
Number of tables in INPROGRESS state: 0
Number of tables in STALE state: 0
Number of tables in ERROR state: 0

Search Engine: READY
Has pending tables. Pending time = 1601679ms
Number of tables in KNOWN_TABLES state: 1934
Number of tables in READY state: 1928
Number of tables in WILL_REMOVE state: 0
Number of tables in BUILDING_AND_NOT_SERVING state: 0
Number of tables in BUILDING_AND_SERVING state: 128
Number of tables in WILL_NOT_INDEX state: 0
```

Ensure that the cluster is **RUNNING** and that the Database and Search Engine are **READY** .

```

$ tscli cluster check
Connecting to hosts...
[Wed Jan  8 23:15:47 2020] START Diagnosing ssh
[Wed Jan  8 23:15:47 2020] SUCCESS
#####
#####
[Wed Jan  8 23:15:47 2020] START Diagnosing connection
[Wed Jan  8 23:15:47 2020] SUCCESS
#####
#####
[Wed Jan  8 23:15:47 2020] START Diagnosing zookeeper
[Wed Jan  8 23:15:47 2020] SUCCESS
#####
#####
[Wed Jan  8 23:15:47 2020] START Diagnosing sage
[Wed Jan  8 23:15:48 2020] SUCCESS
#####
#####
[Wed Jan  8 23:15:48 2020] START Diagnosing timezone
[Wed Jan  8 23:15:48 2020] SUCCESS
#####
#####
[Wed Jan  8 23:15:48 2020] START Diagnosing disk
[Wed Jan  8 23:15:48 2020] SUCCESS
#####
#####
[Wed Jan  8 23:15:48 2020] START Diagnosing cassandra
[Wed Jan  8 23:15:48 2020] SUCCESS
#####
#####
[Wed Jan  8 23:15:48 2020] START Diagnosing hdfs
[Wed Jan  8 23:16:02 2020] SUCCESS
#####
#####
[Wed Jan  8 23:16:02 2020] START Diagnosing orion-oreo
[Wed Jan  8 23:16:02 2020] SUCCESS
#####
#####
[Wed Jan  8 23:16:02 2020] START Diagnosing memcheck
[Wed Jan  8 23:16:02 2020] SUCCESS
#####
#####
[Wed Jan  8 23:16:02 2020] START Diagnosing ntp
[Wed Jan  8 23:16:08 2020] SUCCESS
#####

```

```
#####
[Wed Jan  8 23:16:08 2020] START Diagnosing trace_vault
[Wed Jan  8 23:16:09 2020] SUCCESS
#####
#####
[Wed Jan  8 23:16:09 2020] START Diagnosing postgres
[Wed Jan  8 23:16:11 2020] SUCCESS
#####
#####
[Wed Jan  8 23:16:11 2020] START Diagnosing disk-health
[Wed Jan  8 23:16:11 2020] SUCCESS
#####
#####
[Wed Jan  8 23:16:11 2020] START Diagnosing falcon
[Wed Jan  8 23:16:12 2020] SUCCESS
#####
#####
[Wed Jan  8 23:16:12 2020] START Diagnosing orion-cgroups
[Wed Jan  8 23:16:12 2020] SUCCESS
#####
#####
[Wed Jan  8 23:16:12 2020] START Diagnosing callosum
/usr/lib/python2.7/site-packages/urllib3/connectionpool.py:85
2: InsecureRequestWarning: Unverified HTTPS request is being made. Adding certificate verification is strongly advised. See: https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings
InsecureRequestWarning)
[Wed Jan  8 23:16:12 2020] SUCCESS
#####
#####
```

Your output may look something like the above. Ensure that all diagnostics show **SUCCESS**.

**⚠ Warning:** If `tscli cluster check` returns an error, it may suggest you run `tscli storage gc` to resolve the issue. If you run `tscli storage gc`, note that it restarts your cluster.

### Step 3: Finalize installation

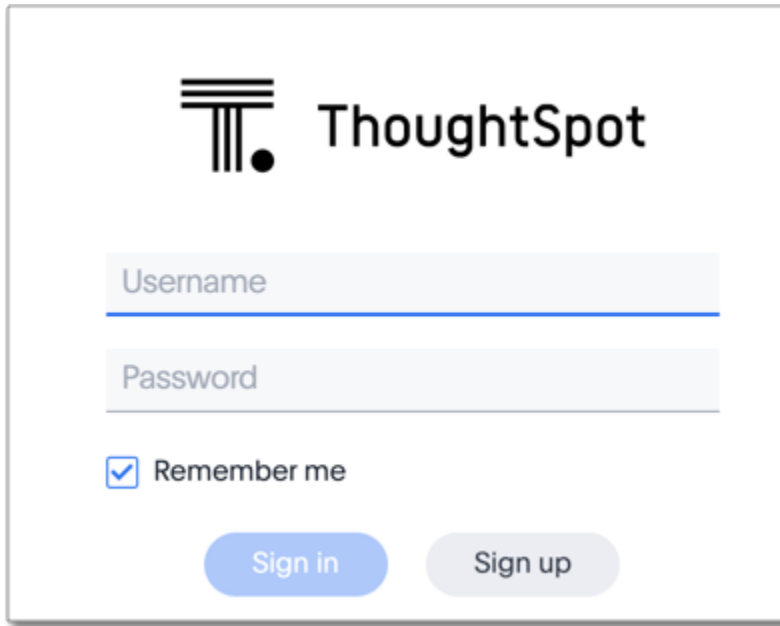
After the cluster status changes to **READY**, sign in to ThoughtSpot on your browser. Follow these steps:

1. Start a browser from your computer.

2. Enter your secure IP information on the address line.

`https://<IP-address>`

3. If you don't have a security certificate for ThoughtSpot, you must bypass the security warning:
  - Click **Advanced**
  - Click **Proceed**
4. The ThoughtSpot sign-in page appears.
5. In the [ThoughtSpot sign-in window](#) [See page 29], enter admin credentials, and click **Sign in**.  
ThoughtSpot recommends changing the default admin password.

The image shows the ThoughtSpot sign-in page. At the top is the ThoughtSpot logo, which consists of a stylized icon of three vertical bars of increasing height with horizontal lines at the top, followed by the word "ThoughtSpot". Below the logo are two input fields: "Username" and "Password". The "Username" field has a blue underline. Below the "Password" field is a checkbox labeled "Remember me" which is checked. At the bottom are two buttons: "Sign in" (blue) and "Sign up" (grey).

## Lean configuration

**(For use with thin provisioning only)** If you have a [small or medium instance type](#) [See page 0], with less than 100GB of data, advanced lean configuration is required before loading any data into ThoughtSpot. After installing the cluster, contact [ThoughtSpot Support](#) [See page 0] for assistance with this configuration.

## Related information

Use these references for successful installation and administration of ThoughtSpot:

- [the nodes.config file \[See page 0\]](#)
- [Parameters of the nodes.config file \[See page 34\]](#)
- [Using the tscli cluster create command \[See page 37\]](#)
- [Parameters of the `cluster create` command \[See page 45\]](#)
- [Deployment Overview \[See page 0\]](#)
- [Contact Support \[See page 0\]](#)



# The nodes.config file

**Summary:** Learn how to use the `get.config` command and the `nodes.config` file to install your hardware or cloud appliance.

## Using the nodes.config file

As you install your appliance, you must configure the nodes.

1. Run the configuration command in your terminal.

```
$ tscli cluster get-config |& tee nodes.config
```

2. Fill in the areas specified in [Parameters of the nodes.config file \[See page 34\]](#) with your specific network information, as shown in [Autodiscovery of one node example \[See page 0\]](#).

**Note:** Some of the information in the `nodes.config` file may be pre-populated from earlier steps. For example, if you specified an IP address while creating VMs, that IP address might already be present in your `nodes.config` file.

3. If you have additional nodes, complete each node within the `nodes.config` file as shown in the [Autodiscovery of one node example \[See page 0\]](#). [Autodiscovery of one node \[See page 0\]](#) shows the `nodes.config` file before you fill in your specific information.

Do not edit any part of the `nodes.config` file except the sections explained in [Parameters of the nodes.config file \[See page 34\]](#). If you delete quotation marks, commas, or other parts of the code, setup may fail.

See [Parameters of the nodes.config file \[See page 34\]](#) to understand the parameters in the file.

### Autodiscovery of one node

```
$ tscli cluster get-config |& tee nodes.config
{
  "ClusterId": "",
  "ClusterName": "",
  "DataNetmask": "",
  "DataGateway": "",
  "IPMINetmask": "",
  "IPMIGateway": "",
  "Timezone": "",
  "NTPServers": "",
  "DNS": "",
  "SearchDomains": "",
  "Nodes": {
    "06:83:1f:f8:99:9e": {
      "NodeId": "06:83:1f:f8:99:9e",
      "Hostname": "",
      "DataIface": {
        "Name": "eth0",
        "IPv4": ""
      },
      "IPMI": {
        "IPv4": ""
      }
    }
  }
}
```

## Autodiscovery of one node example

```
$ vim nodes.config
{
  "ClusterId": "",
  "ClusterName": "",
  "DataNetmask": "255.255.252.0",
  "DataGateway": "192.168.4.1",
  "IPMINetmask": "255.255.252.0",
  "IPMIGateway": "192.168.4.1",
  "Timezone": "America/Los_Angeles",
  "NTPServers": "0.centos.pool.ntp.org,1.centos.pool.ntp.org,2.centos.pool.ntp.org,3.centos.pool.ntp.org",
  "DNS": "192.168.2.200,8.8.8.8",
  "SearchDomains": "example.company.com",
  "Nodes": {
    "ac:1f:6b:8a:77:f6": {
      "NodeId": "ac:1f:6b:8a:77:f6",
      "Hostname": "Thoughtspot-server1",
      "DataIface": {
        "Name": "eth2",
        "IPv4": "192.168.7.70"
      },
      "IPMI": {
        "IPv4": "192.168.5.70"
      }
    }
  }
}
```

## Related information

- [Deploying on the SMC appliance \[See page 0\]](#)
- [Configure ThoughtSpot nodes in AWS \[See page 0\]](#)
- [Configure ThoughtSpot nodes in GCP \[See page 16\]](#)
- [Configure ThoughtSpot nodes in VMware \[See page 0\]](#)
- [Configure ThoughtSpot nodes in Azure \[See page 0\]](#)
- [Deploying on the Dell Appliance \[See page 0\]](#)
- [Parameters of the nodes.config file \[See page 34\]](#)

# Parameters of the nodes.config file

**Summary:** Learn the parameters of the nodes.config file to install your cloud or hardware appliance.

## Using the nodes.config file

As you install your appliance, you must configure the nodes.

1. Run the configuration command in your terminal.

```
$ tscli cluster get-config |& tee nodes.config
```

2. Add network information for your nodes in the nodes.config file output, as demonstrated in the nodes.config file [See page 31] example.

Run `vim nodes.config` to edit the file.

```
$ vim nodes.config
```

Fill in the areas specified in Parameters of nodes.config [See page 35] with your specific network information.

**Note:** Some of the information in the nodes.config file may be pre-populated from earlier steps. For example, if you specified an IP address while creating VMs, that IP address might already be present in your nodes.config file.

3. If you have additional nodes, complete this process for each node.

Do not edit any part of the nodes.config file except the sections explained in Parameters of nodes.config [See page 35]. If you delete quotation marks, commas, or other parts of the code, setup may fail.

See [Parameters of nodes.config](#) [\[See page 35\]](#) to understand the parameters in the file. Different hardware and cloud installations have different parameters. Your installation may not require all the listed parameters.

## Parameters of the nodes.config file

**ClusterId** and **Cluster Name**: Leave these two parameters blank. You fill them out later, when running `tscli cluster create`.

**DataNetmask** The IP of the data netmask, in the form 000.000.000.000. For example, 255.255.252.0.

**DataGateway** The IP of the data gateway, in the form 000.000.000.000. For example, 192.168.4.1.

**IPMINetmask** The IP of the Intelligent Platform Management Interface (IPMI) netmask, in the form 000.000.000.000. For example, 255.255.252.0.

**IPMIGateway** The IP of the Intelligent Platform Management Interface (IPMI) gateway, in the form 000.000.000.000. For example, 192.168.4.1.

**Timezone** The timezone the majority of your ThoughtSpot users are in, in the form Country/City. For example, America/Los\_Angeles. To find your timezone and a city you can use to identify it, use [this timezone list](#) [\[See page 0\]](#).

**NTPServers** The address of your company's Network Time Protocol (NTP) server. If your company does not have an NTP server, you can use one of ThoughtSpot's, as listed in [the nodes.config](#) [\[See page 33\]](#) example under NTPServers. For example, 0.centos.pool.ntp.org.

**DNS** The address of your company's Domain Name Server (DNS). For example, 192.168.2.200,8.8.8.8.

**⚠ Warning:** Configure only two DNS servers. ThoughtSpot does not support configuration of three DNS servers.

**ⓘ Note:** You can only edit DNS settings with this command if you are deploying ThoughtSpot on hardware. ThoughtSpot does not support using `set-config` to edit your DNS settings for cloud deployment.

**SearchDomains** The domain of your company or organization, in the form `example.company.com`.

**Hostname** The name of the host server. For example, `Thoughtspot-server1`.

**IPv4** The main IP address, associated with `DataNetmask` and `DataGateway`, in the form `000.000.000.000`. For example, `192.168.7.70`.

**IPMI IPv4** A secondary IP address, associated with the `IPMINetmask` and `IPMIGateway`, in the form `000.000.000.000`. For example, `192.168.5.70`.

## Related information

- [Deploying on the SMC appliance \[See page 0\]](#)
- [Configure ThoughtSpot nodes in AWS \[See page 0\]](#)
- [Configure ThoughtSpot nodes in GCP \[See page 16\]](#)
- [Configure ThoughtSpot nodes in VMware \[See page 0\]](#)
- [Configure ThoughtSpot nodes in Azure \[See page 0\]](#)
- [Deploying on the Dell Appliance \[See page 0\]](#)
- [The `nodes.config` file \[See page 31\]](#)

# Using the tscli cluster create command

**Summary:** Learn how to use the cluster create command to install your appliance.

## Using the tscli cluster create command

To install your appliance, you must install the cluster using the release tarball (estimated time 1 hour).

1. **Download the release tarball** Download the release tarball from the download link sent by ThoughtSpot Support.  
Refer to your welcome letter to find a link to download the release tarball. If you do not have a download link, open a support ticket at [ThoughtSpot Support \[See page 0\]](#) to access the release tarball.
2. **Copy the release tarball** In your Terminal (Apple) or using [Winscp \[See page 0\]](#) (Windows application), copy the downloaded release tarball to `/home/admin/` and run the cluster installer as shown below.

Run the secure copy command: `scp <release-number> admin@<hostname>:/home/admin/<file-name>` . Note the following parameters:

- `release-number` is the version of ThoughtSpot you have on your cluster, such as `6.0` or `5.3` . `release-number` is of the form `0.0.tar.gz` .
- `hostname` is your network hostname. Ask your network administrator if you do not know your hostname.
- `file-name` is the name of the tarball file on your local machine. For example:

```
$ scp 6.0.tar.gz admin@ThoughtSpot:/home/admin/T  
S-tarball.
```

ThoughtSpot recommends that you start the session using [screen \[See page 0\]](#), so you do not lose your progress if you lose your connection at any point.

```
$ scp <release-number> admin@<hostname>:/home/adm  
in/<file-name>
```

3. **Run the `tscli cluster create` command** Run `tscli cluster create <release-number>` in your terminal.

If you are using an s3 or GCS bucket for object storage, include the flag `--enable_cloud_storage=s3` or `--enable_cloud_storage=gcs`. GCS is GCP's object storage, and s3 is AWS's object storage.

```
$ tscli cluster create 6.0.tar.gz --enable_cloud_storage=s3
```

```
$ tscli cluster create 6.0.tar.gz --enable_cloud_storage=gcs
```

4. **Specify your installation information** Fill out the cluster name, cluster ID, email alert preferences and the IP's of the nodes at the prompts specified in [Parameters of the `tscli cluster create` command \[See page 45\]](#).
5. **Wait for output** You may need to wait about 15 seconds before you see any output. The installer is unpacking files and copying them over to the nodes, which can take a few seconds.

Do not edit any part of the installer file except the sections specified in [Parameters of the `tscli cluster create` command \[See page 45\]](#). If you delete colons, commas, or other parts of the code, setup may fail.

Refer to [Parameters of the `tscli cluster create` command \[See page 45\]](#) for further information.

Your `tscli cluster create` output may look something like the following:

## Run the Installer

The output for a hardware installation:



```

[admin@jessi-gcs-test ~]$ tscli cluster create 6.0-145.tar.gz
Unpacking 6.0-145.tar.gz to /export/release_cache/e695feeec6275
91dc644635c0d8ea03d
#####
#
#
#
#           Welcome to ThoughtSpot installer
#
#
#####
#
Enter a name for the cluster (alphanumeric characters only): je
ssi-ts-gcs-test
Enter cluster ID: 0x0000
Enter IP addresses of all hosts in the cluster (space separate
d): 10.116.0.66
Enter email addresses for alerts (space separated, "later" to s
kip): later
2019-11-03 21:26:47,959 Pushing /usr/local/scaligent/toolchain/
jolokia to localhost
2019-11-03 21:26:48,195 Rsync finished on localhost
2019-11-03 21:26:48,195 Rsync finished to all hosts
2019-11-03 21:26:48,195 Pushing /usr/local/scaligent/toolchain/
jvm to localhost
2019-11-03 21:26:48,461 Rsync finished on localhost
2019-11-03 21:26:48,461 Rsync finished to all hosts
2019-11-03 21:26:48,462 Pushing /usr/local/scaligent/toolchain/
hadoop to localhost
2019-11-03 21:26:48,738 Rsync finished on localhost
2019-11-03 21:26:48,738 Rsync finished to all hosts
2019-11-03 21:26:48,738 Pushing /usr/local/scaligent/toolchain/
zookeeper to localhost
2019-11-03 21:26:49,004 Rsync finished on localhost
2019-11-03 21:26:49,004 Rsync finished to all hosts
#####
#
#           Setup Hadoop
#####
#
Installing Zookeeper
First deleting existing Zookeeper service
Deleting Zookeeper
Zookeeper Deleted

```

```
Starting Zookeeper servers
Zookeeper Ready
Installing HDFS
First deleting existing HDFS service
Deleting HDFS
HDFS Deleted
Deploying configs
Formatting and starting Primary Name Node
Starting Data Nodes
Setting NFS configs
Waiting for HDFS
HDFS Ready
2019-11-03 21:27:06,212 Starting a secondary namenode for check
point...
2019-11-03 21:27:06,551 Cannot disable unknown service: hdfs_se
condary_namenode
2019-11-03 21:27:06,552 Failed to disable new service: hdfs_sec
ondary_namenode
2019-11-03 21:27:07,198 HDFS secondary namenode started success
fully.
Successfully Created zookeeper znodes
Successfully initialized HDFS files
Successfully saved hadoop layout proto
Successfully moved the release dir to export partition
Successfully installed orion on localhost
Successfully connected to Orion Master
Successfully enabled data protection
Successfully enabled firewall
Successfully added machines to cluster
Couldn't install application services
Successfully connected to Orion Master
Successfully enabled data protection
Successfully enabled firewall
Successfully added machines to cluster
Couldn't install application services
Successfully connected to Orion Master
Successfully enabled data protection
Successfully enabled firewall
Successfully added machines to cluster
Successfully installed application services
All nodes are now healthy
Restarted all services
Successfully pushed statsdb
Successfully refreshed alert metadata
Successfully enabled alerts
```

```
Successfully configured new ssh keys on cluster  
Successfully recorded install event  
Successfully deleted deployer service  
2019-11-03 21:40:02,917 Cannot check for enabled status of unknown service: deploy
```

The output for a cloud installation, with GCS object storage:

```

admin@jessi-gcs-test ~]$ tscli cluster create ./6.0-145.tar.gz
z --enable_cloud_storage gcs
Unpacking ./6.0-145.tar.gz to /export/release_cache/e695feec62
7591dc644635c0d8ea03d
#####
#
#
#
#           Welcome to ThoughtSpot installer
#
#
#####
#
Enter a name for the cluster (alphanumeric characters only): je
ssi-ts-gcs-test
Enter cluster ID: 0x0000
Enter IP addresses of all hosts in the cluster (space separate
d): 10.116.0.66
Enter email addresses for alerts (space separated, "later" to s
kip): later
Enter the GCS bucket to be configured for the cluster: jessi-gc
s-test-bucket
Bucket successfully validated.
2019-11-03 22:00:29,175 Pushing /usr/local/scaligent/toolchain/
jolokia to localhost
2019-11-03 22:00:29,411 Rsync finished on localhost
2019-11-03 22:00:29,411 Rsync finished to all hosts
2019-11-03 22:00:29,412 Pushing /usr/local/scaligent/toolchain/
jvm to localhost
2019-11-03 22:00:29,678 Rsync finished on localhost
2019-11-03 22:00:29,678 Rsync finished to all hosts
2019-11-03 22:00:29,678 Pushing /usr/local/scaligent/toolchain/
hadoop to localhost
2019-11-03 22:00:29,945 Rsync finished on localhost
2019-11-03 22:00:29,945 Rsync finished to all hosts
2019-11-03 22:00:29,945 Pushing /usr/local/scaligent/toolchain/
zookeeper to localhost
2019-11-03 22:00:30,211 Rsync finished on localhost
2019-11-03 22:00:30,211 Rsync finished to all hosts
#####
#
#           Setup Hadoop
#####
#

```

```
Installing Zookeeper
First deleting existing Zookeeper service
Deleting Zookeeper
Zookeeper Deleted
Starting Zookeeper servers
Zookeeper Ready
Installing HDFS
First deleting existing HDFS service
Deleting HDFS
HDFS Deleted
Deploying configs
Formatting and starting Primary Name Node
Starting Data Nodes
Setting NFS configs
Waiting for HDFS
HDFS Ready
2019-11-03 22:00:47,287 Starting a secondary namenode for check
point...
2019-11-03 22:00:47,629 Cannot disable unknown service: hdfs_se
condary_namenode
2019-11-03 22:00:47,630 Failed to disable new service: hdfs_sec
ondary_namenode
2019-11-03 22:00:48,282 HDFS secondary namenode started success
fully.
Successfully Created zookeeper znodes
Successfully initialized HDFS files
Successfully saved hadoop layout proto
Successfully moved the release dir to export partition
Successfully installed orion on localhost
Successfully connected to Orion Master
Successfully enabled data protection
Successfully enabled firewall
Successfully added machines to cluster
Couldn't install application services
Successfully connected to Orion Master
Successfully enabled data protection
Successfully enabled firewall
Successfully added machines to cluster
Couldn't install application services
Successfully connected to Orion Master
Successfully enabled data protection
Successfully enabled firewall
Successfully added machines to cluster
Successfully installed application services
All nodes are now healthy
```

```
Restarted all services
Successfully pushed statsdb
Successfully refreshed alert metadata
Successfully enabled alerts
Successfully configured new ssh keys on cluster
Successfully recorded install event
Successfully deleted deployer service
2019-11-03 22:11:54,571 Cannot check for enabled status of unknown service: deploy
```

## Related information

- [Deploying on the SMC appliance \[See page 0\]](#)
- [Configure ThoughtSpot nodes in AWS \[See page 0\]](#)
- [Configure ThoughtSpot nodes in GCP \[See page 16\]](#)
- [Configure ThoughtSpot nodes in VMware \[See page 0\]](#)
- [Configure ThoughtSpot nodes in Azure \[See page 0\]](#)
- [Deploying on the Dell Appliance \[See page 0\]](#)
- [Parameters of the `tscli cluster create` command \[See page 45\]](#)

# Parameters of the tscli cluster create command

**Summary:** Learn the parameters of the tscli cluster create command.

## Using the tscli cluster create command

To install your appliance, you must install the cluster.

1. **Copy the release tarball** In your Terminal (Apple) or using [Winscp \[See page 0\]](#) (Windows application), copy the downloaded release tarball to /home/admin/ and run the cluster installer as shown below.

Run the secure copy command: `scp <release-number> admin@<hostname>:/home/admin/<file-name>` . Note the following parameters:

- `release-number` is the version of ThoughtSpot you have on your cluster, such as `6.0` or `5.3` . `release-number` is of the form `0.0.tar.gz` .
- `hostname` is your network hostname. Ask your network administrator if you do not know your hostname.
- `file-name` is the name of the tarball file on your local machine. For example:

```
$ scp 6.0.tar.gz admin@ThoughtSpot:/home/admin/TS-tarball.
```

ThoughtSpot recommends that you start the session using [screen \[See page 0\]](#), so you do not lose your progress if you lose your connection at any point.

```
$ scp <release-number> admin@<hostname>:/home/admin/<file-name>
```

2. **Run the cluster create command** Run `tscli cluster create <release-number>` in your terminal.

If you are using an s3 or GCS bucket for object storage, include the flag `--enable_cloud_storage=s3` or `--enable_cloud_storage=gcs` . GCS is GCP's object

storage, and s3 is AWS's object storage.

```
$ tscli cluster create 6.0.tar.gz --enable_cloud_storage=s3
```

```
$ tscli cluster create 6.0.tar.gz --enable_cloud_storage=gcs
```

3. **Fill out your specific information** Fill out the cluster name, cluster ID, email alert preferences and the IP's of the nodes at the prompts specified in [Parameters of ThoughtSpot Installer \[See page 46\]](#) below.

Refer to [Parameters of ThoughtSpot Installer \[See page 46\]](#) for further information. Refer to [Using the tscli cluster create command \[See page 37\]](#) to see the expected output of the install command, `tscli cluster create`.

Do not edit any part of the installer file except the sections specified in [Parameters of ThoughtSpot Installer \[See page 46\]](#). If you delete colons, commas, or other parts of the code, setup may fail.

## Parameters of ThoughtSpot Installer

**Cluster Name** Name your cluster based on the ThoughtSpot naming convention, in the form *company-clustertype-location-clusternumber*. For example, *ThoughtSpot-prod-Sunnyvale-12*.

**Cluster ID** Enter the ID of your cluster that [ThoughtSpot Support \[See page 0\]](#) provided for you. Open a ticket with ThoughtSpot Support if you do not have an ID.

**Host IPs** Enter the IP addresses of all cluster hosts, in the form 000.000.000.000. For example, 192.168.7.70. Use spaces instead of commas to separate multiple IP addresses.

**Email alerts** Enter the email addresses you would like to receive alerts about this cluster, in the form *company@example.com*. The address *prod-alerts@thoughtspot.com* appears automatically and should remain, so that ThoughtSpot can be aware of the status of your cluster. Separate email addresses using a space.



## Related information

- [Deploying on the SMC appliance \[See page 0\]](#)
- [Configure ThoughtSpot nodes in AWS \[See page 0\]](#)
- [Configure ThoughtSpot nodes in GCP \[See page 16\]](#)
- [Configure ThoughtSpot nodes in VMware \[See page 0\]](#)
- [Configure ThoughtSpot nodes in Azure \[See page 0\]](#)
- [Deploying on the Dell Appliance \[See page 0\]](#)
- [Using the `tscli cluster create` command \[See page 37\]](#)

# 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.

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).
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.

## 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
2181	TCP	Zookeeper servers listen on this RPC port for client connections	bidirectional	All nodes	All nodes	Zookeeper servers listen on this RPC port for client connections
3181	TCP	Zookeeper servers listen on this RPC port for client connections	bidirectional	All nodes	All nodes	Zookeeper servers listen on this RPC port for client connections
4181	TCP	Zookeeper servers listen on this RPC port for client connections	bidirectional	All nodes	All nodes	Zookeeper servers listen on this RPC 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

Port	Protocol	Service Name	Direction	Source	Dest.	Description
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
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

Port	Protocol	Service Name	Direction	Source	Dest.	Description
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
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, metadata-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
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
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

Port	Protocol	Service Name	Direction	Source	Dest.	Description
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
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

Port	Protocol	Service Name	Direction	Source	Dest.	Description
5021	RPC	Callosum services like meta-data services, metadata-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, metadata-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, metadata-dependency service, scheduling service, session-less service, spotiq service	bidirectional	All nodes	All nodes	Port where search service contacts meta-data service for metadata
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, metadata-dependency service, scheduling service, session-less service, spotiq service	bidirectional	All nodes	All nodes	Port where search service contacts meta-data 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



Port	Protocol	Service Name	Direction	Source	Dest.	Description
8888	HTTP	HTTP proxy server (tinyproxy)	bidirectional	All nodes	All nodes	Reverse SSH tunnel
11211	Mem-cached	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
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

Port	Protocol	Service Name	Direction	Source	Dest.	Description
20123 - 32768	TCP (dynamic)	Dynamic port in this range used for various services and ancillary services like atlas, caffeine, call-home, 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
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.

Port	Protocol	Service Name	Direction	Source	Dest.	Description
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 thoughtsport.egnyte.com.
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.

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