ThoughtSpot

ThoughtSpot Deployment Guide for VMware

Release 6.0

February, 2020

Table of Contents

VMware configuration overview

Summary: You can host ThoughtSpot on VMware.

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

About ThoughtSpot in 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:
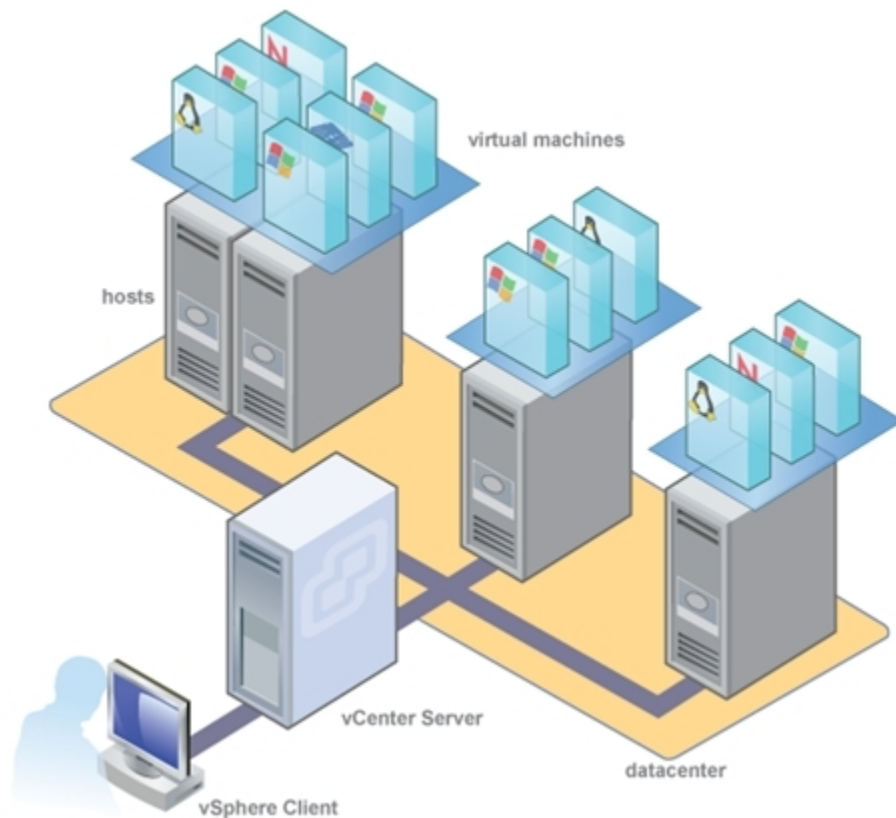
> **ⓘ Note:** This is a generic representation; Only CentOS-based virtual machines are supported with

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

Per VM user data capacity	CPU/RAM	Data disk
20 GB	16/128 GB	800 GB
100 GB	32/256 GB	800 GB
256 GB	72/512 GB	6 TB

Note: All cores must be hyperthreaded. 200GB SSD boot disk required for all configurations.

Locally attached storage provides the best performance.

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

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

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

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

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

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

Questions or comments?

We hope your experience with ThoughtSpot is excellent. Please let us know how it goes, and what we can do to make it better. You can contact ThoughtSpot [See page 0] by email, phone, or by filing a support ticket.

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.

> ❶ **Note:** Older versions of ESXi aren't supported due to hardware/driver incompatibility issues.

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.

3. Download [See page 0] and fill out the ThoughtSpot site survey to have a quick reference for any networking information you may need to fill out as you install ThoughtSpot. Ask your network administrator if you need help filling out the site survey.

Use the OVF to Create a VM

1. **Download [See page 0]** the `ThoughtSpot OVF` to a location on an accessible disk.

2. Log in to the ESXi web portal.
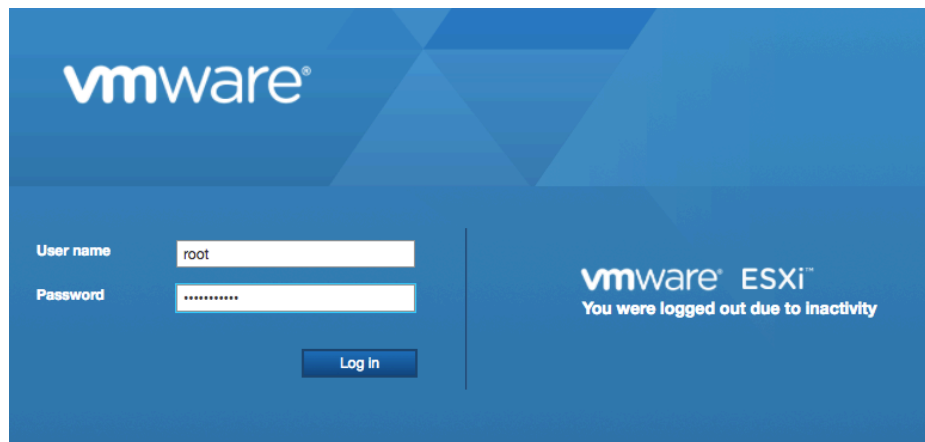
3. Select **Virtual Machines > Create/Register VM**.

The system displays the dialog for selecting an OVF template.
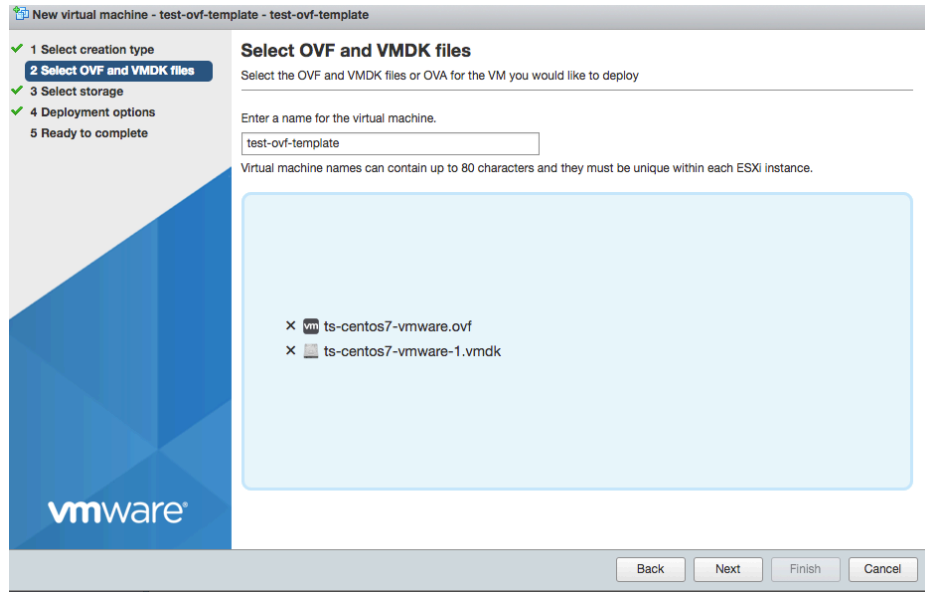
4. Choose the OVF template and click **Next**.

   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.
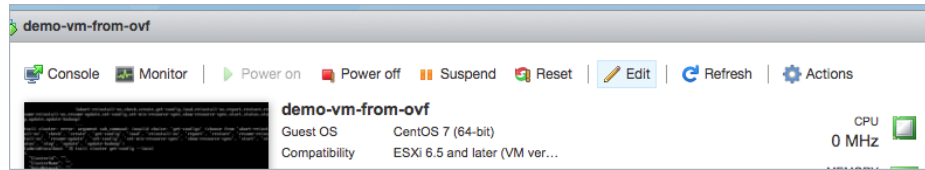
2. Select **Add hard disk > New hard disk**.
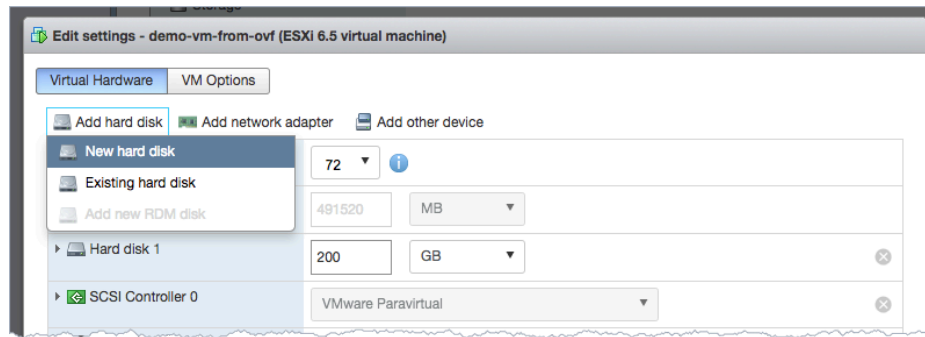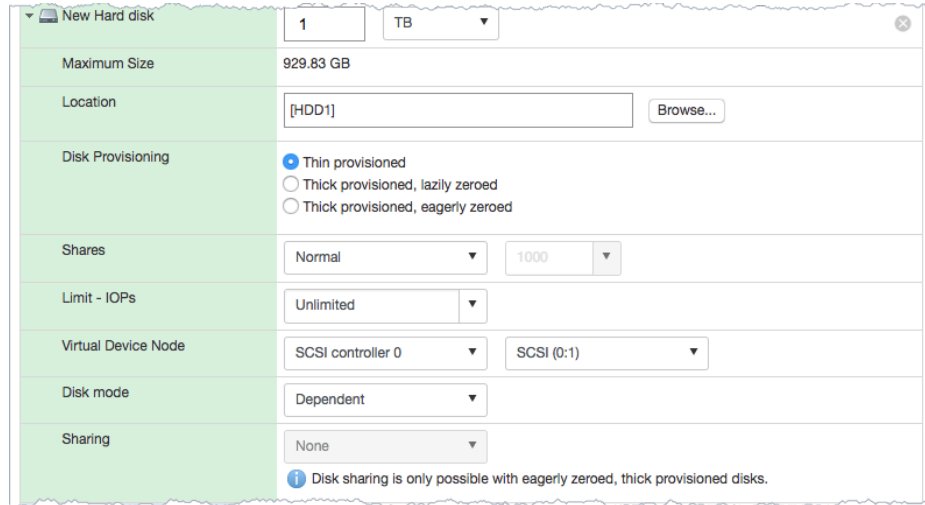
The new disk appears as a new row under the only existing SSD row.

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.

Configure ThoughtSpot nodes in VMware

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

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

Installation Prerequisites

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

1. **Download the OVF** Download the Open Virtualization Format [See page 0] (OVF) file.
2. **Review configuration overview** Refer to VMware configuration overview [See page 2] for detailed instance specs.
3. **Create the instance** Refer to Set up VMware for ThoughtSpot [See page 5] to create and launch your instance.
4. **Review required ports** Refer to Network Policies [See page 43] 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.

   > ❶ **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 26].

```
$ 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 28]. 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 29] 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 29]. 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 da
emon
    Loaded: loaded (/usr/lib/systemd/system/firewalld.s
ervice; 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 15].

   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.or
g,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 18].

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 f
ound configuration for
only 1 nodes.
Continuing anyway. Error in cluster config validation: [] is no
t 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; ena
bled; vendor preset: disabled)
   Active: active (running) since Fri 2019-12-06 13:56:29 PS
T; 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 29]

- Using the tscli cluster create command [See page 32]

- Parameters of the `cluster create` command [See page 40]

- ThoughtSpot Documentation [See page 0]

- Contact Support [See page 0]

Install ThoughtSpot clusters in VMware

> **Summary:** Learn how to install ThoughtSpot clusters in VMware.

Prerequisites

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

1. **Download the OVF** Download the Open Virtualization Format [See page 0] (OVF) file.
2. **Review configuration overview** Refer to VMware configuration overview [See page 2] for detailed instance specs.
3. **Create the instance** Refer to Set up VMware for ThoughtSpot [See page 5] to create and launch your instance.
4. **Review required ports** Refer to Network Policies [See page 43] to view the required ports for successful operation of ThoughtSpot.
5. **Configure nodes** Refer to Configure ThoughtSpot nodes in VMware [See page 11] 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/sdb1/TS_TASKS/install` using the

following command:

```
$ scp <release-number>.tar.gz admin@<hostname>:/export/
sdb1/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.

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

2. 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/sdb1/TS_TASKS/install/6.0.tar.gz`.

3. 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.

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

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 32] and Parameters of the `cluster create` command [See page 40].

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 ma
de. Adding certificate verification is strongly advised. See: h
ttps://urllib3.readthedocs.io/en/latest/advanced-usage.html#ss
l-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
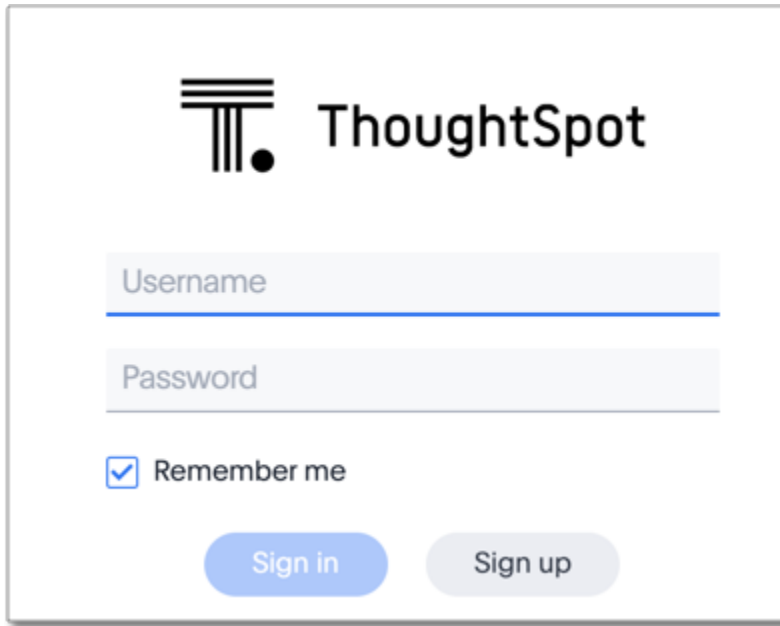
After the cluster status changes to `READY`, sign in to ThoughtSpot on your browser. Follow these steps:

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 24], enter admin credentials, and click **Sign in**.
    ThoughtSpot recommends changing the default admin password.

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 29]

- Using the tscli cluster create command [See page 32]

- Parameters of the `cluster create` command [See page 40]

- 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 29] with your specific network information, as shown in Autodiscovery of one node example [See page 0].

   > ❶ **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 29]. If you delete quotation marks, commas, or other parts of the code, setup may fail.

See Parameters of the nodes.config file [See page 29] 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.or
g,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

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 26] example.
Run `vim nodes.config` to edit the file.

```
$ vim nodes.config
```

Fill in the areas specified in Parameters of `nodes.config` [See page 30] 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 30]. If you delete quotation marks, commas, or other parts of the code, setup may fail.

See Parameters of `nodes.config` [See page 30] 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 28] 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.

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

> ❶ **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 0]
- Configure ThoughtSpot nodes in VMware [See page 11]
- Configure ThoughtSpot nodes in Azure [See page 0]
- Deploying on the Dell Appliance [See page 0]
- The nodes.config file [See page 26]

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_storag
    e=s3
    ```

    ```
    $ tscli cluster create 6.0.tar.gz --enable_cloud_storag
    e=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 40].

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 40]. 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 40] 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 unkn
own service: deploy
```

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

```
admin@jessi-gcs-test ~]$ tscli cluster create ./6.0-145.tar.g
z  --enable_cloud_storage gcs
Unpacking ./6.0-145.tar.gz to /export/release_cache/e695feeec62
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 unkn
own 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 0]

- Configure ThoughtSpot nodes in VMware [See page 11]

- 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 40]

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/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/ad
     min/<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_storag
e=s3
```

```
 $ tscli cluster create 6.0.tar.gz ——enable_cloud_storag
e=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 41] below.

Refer to Parameters of ThoughtSpot Installer [See page 41] for further information. Refer to Using the tscli cluster create command [See page 32] 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 41]. 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 0]

- Configure ThoughtSpot nodes in VMware [See page 11]

- Configure ThoughtSpot nodes in Azure [See page 0]

- Deploying on the Dell Appliance [See page 0]

- Using the tscli cluster create command [See page 32]

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 them-selves
4000	RPC	Falcon worker RPC port	bidirectional	All nodes	All nodes	Port used by data cache for communication between them-selves
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 them-selves
4004	RPC	Falcon worker RPC port	bidirectional	All nodes	All nodes	Port used by data cache for communication between them-selves
4021	RPC	Sage metadata ser-vice port (exported by Tomcat), Callo-sum services like meta-data services, medata-dependency service, scheduling service, session-less service, spotiq ser-vice	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 them-selves
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 com-munication
4233	RPC	Sage index server RPC port	bidirectional	All nodes	All nodes	Port used for search service internal com-munication
4241	HTTP	Sage auto complete server HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Port used to debug the search service
4242	RPC	Sage auto complete server RPC port	bidirectional	All nodes	All nodes	Port used for search service internal com-munication
4243	RPC	Sage auto complete server metadata sub-scriber port	bidirectional	All nodes	All nodes	Port used for search internal communication
4244	RPC	Sage auto complete server metadata sub-scriber port	bidirectional	All nodes	All nodes	Port used for search internal communication
4245	RPC	Sage auto complete server metadata sub-scriber port	bidirectional	All nodes	All nodes	Port used for search internal communication

Port	Protocol	Service Name	Direction	Source	Dest.	Description
4243	RPC	Sage auto complete server metadata sub-scriber port	bidirectional	All nodes	All nodes	Port used for search internal communication
4251	RPC	Sage master RPC port	bidirectional	All nodes	All nodes	Port used for search service internal com-munication
4405	RPC	Diamond (graphite) port	bidirectional	All nodes	All nodes	Port used for communication with monitoring service
4406	RPC	Diamond (graphite) port	bidirectional	All nodes	All nodes	Port used for communication with monitoring service
4500	RPC	Trace vault service RPC port	bidirectional	All nodes	All nodes	Trace collec-tion for ThoughtSpot services
4501	HTTP	Trace vault service HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Debug trace collection
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, 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
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 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 na-menode 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 da-ta
50010	HTTP	HDFS datanode server HTTP port	bidirectional	All nodes	All nodes	Debug DFS da-ta
50020	HTTP	HDFS datanode server HTTP port	bidirectional	All nodes	All nodes	Debug DFS da-ta
7000	TCP	Cassandra KV store database	bidirectional	All nodes	All nodes	Debug DFS da-ta
7001	TCP	Cassandra	bidirectional	All nodes	All nodes	Debug DFS da-ta
9042	HTTP	Munshi server im-pression service, Cassandra	bidirectional	All nodes	All nodes	Debug DFS da-ta
9160	TCP	Cassandra	bidirectional	All nodes	All nodes	Debug DFS da-ta
4010	HTTP	Falcon moderator	bidirectional	All nodes	All nodes	Debug DFS da-ta
4011	HTTP	Falcon moderator	bidirectional	All nodes	All nodes	Debug DFS da-ta

Port	Protocol	Service Name	Direction	Source	Dest.	Description
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
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 thoughtspot.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.amazon-aws.com s3.us-west-1.amazon-aws.com s3-us-west-1.amazon-aws.com s3.dualstack.us-west-1.amazonaws.com	For transferring monitoring data to InfluxCloud. (Given address will resolve to point to AWS instances).
25 or 587	SMTP	SMTP or Secure SMTP	outbound	All nodes and SMTP relay (provided by customer)	All nodes	Allow outbound access for the IP address of whichever email relay server is in use. This is for sending alerts to ThoughtSpot Support.

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.