ThoughtSpot

ThoughtSpot Deployment Guide for Microsoft Azure

Release 6.0

February, 2020

Table of Contents

Azure configuration options

> **Summary:** ThoughtSpot offers several Microsoft Azure instance types.

ThoughtSpot can be deployed in your Azure environment by deploying compute (VM) instances in your VNET as well as an underlying persistent storage infrastructure. Currently we support Premium SSD Managed Disks for persistent storage. For more information, see Managed Disks pricing [See page 0] in Microsoft's Azure documentation.

All Azure VMs (nodes) in a ThoughtSpot cluster must be in the same availability zone (and, therefore, also in the same region). ThoughtSpot does not support deploying VMs(nodes) of the same cluster across availability zones. For more information, see What are Availability Zones in Azure? [See page 0] in Microsoft's Azure documentation.

ThoughtSpot Azure instance types

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

Set up ThoughtSpot in Azure

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

About the ThoughtSpot image

To provision ThoughtSpot in the Azure portal, access the ThoughtSpot Virtual Machine in the Azure Marketplace.

The ThoughtSpot Virtual Machine comes provisioned with the custom ThoughtSpot image to make hosting simple. A virtual machine is a preconfigured template that provides the information required to launch an instance of ThoughtSpot. It includes a root disk for the instance, which contains an operating system, application server, and other necessary software.

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

Due to security restrictions, the ThoughtSpot Virtual Machine does not have default passwords for the administrator users. When you are ready to obtain the password, contact ThoughtSpot Support [See page 0].

Set up ThoughtSpot in Azure

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

Prerequisites

Complete these steps before launching your ThoughtSpot Virtual Machine:

1. Obtain an Azure login account.
2. Set up usage payment details with Microsoft Azure.
3. Find your company's Resource Group [See page 0]. (optional–you can also create one while creating your virtual machines.)
4. 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. Ask your network administrator if you need help filling out the site survey.

Create an instance

Create your virtual machines based on the ThoughtSpot Virtual Machine [See page 0].

1. Log in to the Azure portal.

   In a browser, go to https://portal.azure.com/#home [See page 0], and log in to your Azure account.

2. On the Azure portal homepage, hover over **Virtual machines**, and click **Create**.
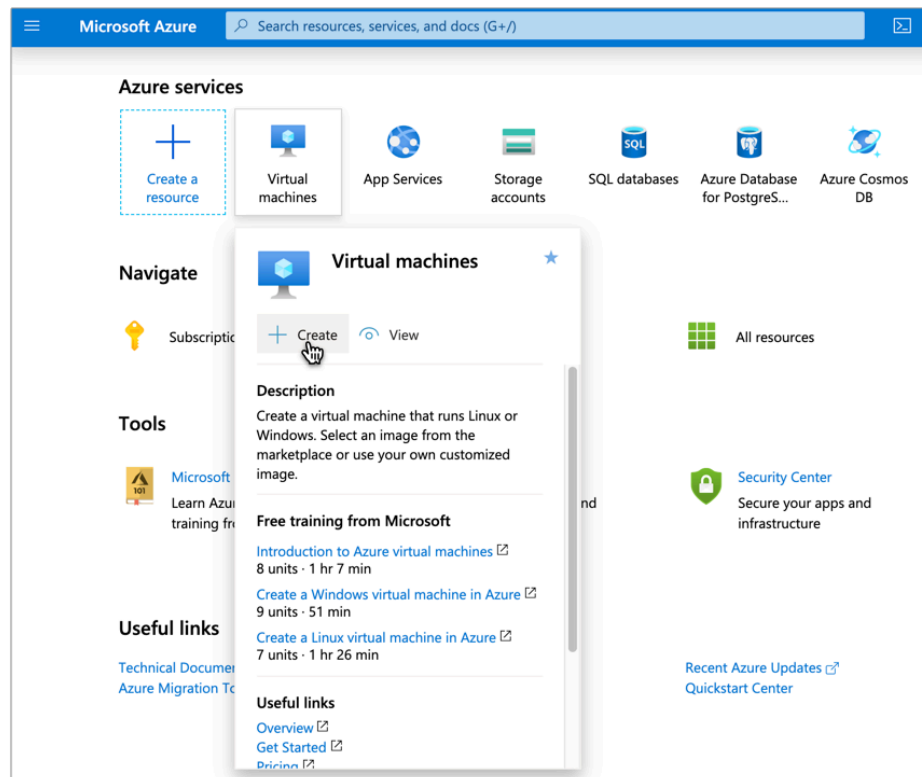
3. Specify information under **Basics**.

1	Choose a subscription type from the dropdown menu.
2	If your company already has a resource group, *select existing*. If not, *create new*.
3	Specify a name for your virtual machine.

4	Specify the region in which you are creating the VM.
5	Click **Browse all public and private images**, and search for the ThoughtSpot image. Click on it.
6	Refer to Azure configuration options [See page 2] to choose a size for your VM that works for your cluster needs.
7	Select **SSH public key** and specify a username.
8	Enter your SSH public key. Contact ThoughtSpot support [See page 0] to obtain a public key, if you do not have one. Note that this SSH public key is different from the SSH private key you use later, to ssh into your VM from the command line.
9	Choose **allow selected ports**.
10	Open the necessary Inbound and Outbound ports to ensure that the ThoughtSpot processes do not get blocked. See the minimum port requirements. [See page 8]

4. Specify information under **Disks**.

1	Choose a disk type from the dropdown menu. ThoughtSpot recommends the Premium SSD disks.

> **2** Click **Create and attach a new disk**. Add two data disks. Refer to Azure configura-
> tion options [See page 0] to see what size they should be.

> **3** Under **Advanced**, click **yes** to **use managed disks**.

> ☑ **Tip:** The new Standard SSD disk types are only available for particular regions. Make
>
> sure this disk type is supported in the region you chose for your VM before selecting it.

See Standard SSD Disks for Virtual Machine workloads [See page 0] for more on SSD disks.

ThoughtSpot recommends the Premium SSD disks.

5. Specify information under **Networking**.

> **1** Find your company's virtual network and select it, or **create new**.

> **2** Find your company's public IP, or **create new**.

> **3** Select **Advanced** for *NIC network security group*.
>
> **4** After you select **Advanced**, the **Configure network security group** option appears. Find your company's security group, or **create new**. When creating your security group, ensure that the required ports are open. Refer to the minimum port requirements. [See page 8]

6. Under **Management**, configure your monitoring and management preferences. If you have no preferences, you can leave them at their default settings.

7. Under **Advanced**, configure your advanced settings preferences. If you have no preferences, you can leave them at their default settings.

8. Under **Tags**, tag your virtual machine with a human-readable string to help you identify it.

9. Click **Review + create** in the bottom left corner of your screen.

10. Review your changes, and click **create**. Azure does the final validation check.

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.

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

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

Refer to network policies [See page 46] for more information.

Prepare for starting up ThoughtSpot

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

1. Obtain the VM's public and private IP addresses.

   - To see the public IP, click the VM name link. This will show the public IP of the VM.
   - To see the private IP, select **more services** from the Microsoft Azure homepage. Select **Networking** from the list on the left side of the screen.

2. In a terminal application, connect to the VM through SSH. When prompted, enter the private key provided for the admin user.

   - You must file a support ticket to obtain this private key; it is necessary for the first login.
   - This key is different from the credentials, or the public keys supplied in earlier steps, which do not work in this context.

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

3. Update the password for both the `admin` and the `thoughtspot` users.

   The command prompts you to type in a new password, and then to confirm the password.

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

   > ❶ **Warning:** If you do not change the password, you cannot log back into your Azure VMs. Your private key does not work after initial installation.

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

Verify storage disks

Verify the existence of your data disks, created in Step 4 of create an instance [See page 4], by issuing `lsblk` in your terminal application:

```
$ lsblk
```

Your result may look something like the following:

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

1. Unmount the temporary disk by issuing the following command:

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

> ❶ **Warning:** The `/mnt/resource` disk, which is mounted on the `/dev/sdb` disk, is temporary. Any data on it will be wiped if the VM is shut down. You must unmount the `/mnt/resource` disk.

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

> ❶ **Warning:** Do not use the disk `/dev/sdb`. Any data on it will be wiped if the VM is
>
> shut down.

```
$ sudo /usr/local/scaligent/bin/prepare_disks.sh /dev/s
dc /dev/sdd
```

3. Check the disks' status by issuing the following command:

```
$ df -h
```

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

Create network support settings

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

1. SSH into one of your VMs.

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

2. Update the VM's hostname:

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

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

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

3. Update `/etc/sysconfig/network-scripts/ifcfg-eth0` with the IP and hostname:

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

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

4. Modify permissions for `/etc/sysconfig/network-scripts/ifcfg-eth0` . This command allows the root user to retain read/write permissions, and grants read-only permissions to other users.

```
 $ sudo chmod 644 /etc/sysconfig/network-scripts/ifcfg-e
th0
```

5. Repeat this process (steps 1 through 4) for each node.

Configure ThoughtSpot nodes in Azure

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

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

Installation Prerequisites

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

1. **Review configuration options** Refer to Azure configuration options [See page 2] for detailed instance specs.
2. **Create the instance** Refer to Set up Azure for ThoughtSpot [See page 3] to create and launch your instance.
3. **Review required ports** Refer to Network Policies [See page 46] 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

Use Terminal on a Mac or a terminal emulator on Windows to log in to your cluster. Log in using the ssh private key provided by ThoughtSpot.

If you do not have a private key, contact ThoughtSpot Support [See page 0] by email or through the support portal.

To log in to your cluster, run `ssh -i <private-key> admin@<public-vm-ip>`.

```
$ ssh -i <private_key> admin@<public-vm-ip>
```

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 29].

```
$ 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 31]. Run `vim nodes.config` to edit the file.

```
$ vim nodes.config
```

> 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.

2. Fill in the areas specified in Parameters of the nodes.config file [See page 32] 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 32]. 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 17].

    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 20].

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 32]

- Using the tscli cluster create command [See page 35]

- Parameters of the `cluster create` command [See page 43]

- ThoughtSpot Documentation [See page 0]

- Contact Support [See page 0]

Install ThoughtSpot clusters in Azure

Summary: Learn how to install ThoughtSpot clusters in Azure.

Prerequisites

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

1. **Review configuration options** Refer to Azure configuration options [See page 2] for detailed instance specs.
2. **Create the instance** Refer to Set up Azure for ThoughtSpot [See page 3] to create and launch your instance.
3. **Review required ports** Refer to Network Policies [See page 46] to view the required ports for successful operation of ThoughtSpot.
4. **Configure nodes** Refer to Configure ThoughtSpot nodes in Azure [See page 13] 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/
sdc1/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/sdc1/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 35] and Parameters of the `cluster create` command [See page 43].

   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.

7. Make this change to each node to ensure that you can log back in:

   a. Open the grub file `/update/etc/default/grub` :

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

   b. Change the line:

```
GRUB_CMDLINE_LINUX="console=tty0 console=ttyS1,115200n
8"
```

   to:

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

c. Save your changes.

8. Issue these commands:

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

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` .

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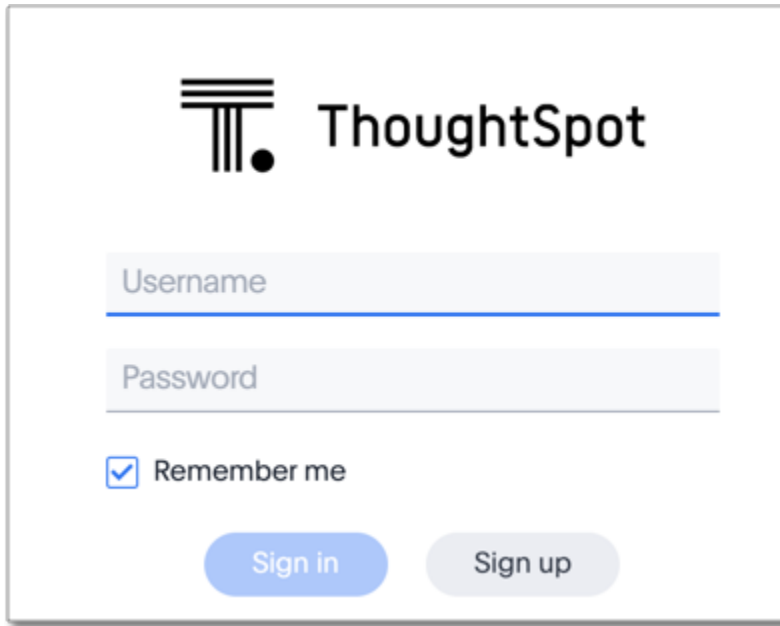
After the cluster status changes to `READY` , sign in to ThoughtSpot on your browser. Follow these steps:

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 27], 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 32]
- Using the tscli cluster create command [See page 35]
- Parameters of the `cluster create` command [See page 43]
- 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 32] 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 32]. If you delete quotation marks, commas, or other parts of the code, setup may fail.

See Parameters of the nodes.config file [See page 32] 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

- 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 0]

- Configure ThoughtSpot nodes in Azure [See page 13]

- Deploying on the Dell Appliance [See page 0]

- Parameters of the nodes.config file [See page 32]

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

   ```
   $ vim nodes.config
   ```

   Fill in the areas specified in Parameters of `nodes.config` [See page 33] 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 33]. If you delete quotation marks, commas, or other parts of the code, setup may fail.

See Parameters of `nodes.config` [See page 33] 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 31] 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 0]
- Configure ThoughtSpot nodes in Azure [See page 13]
- Deploying on the Dell Appliance [See page 0]
- The nodes.config file [See page 29]

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 43].

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 43]. 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 43] 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 0]

- Configure ThoughtSpot nodes in Azure [See page 13]

- Deploying on the Dell Appliance [See page 0]

- Parameters of the tscli cluster create command [See page 43]

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 44] below.

Refer to Parameters of ThoughtSpot Installer [See page 44] for further information. Refer to Using the tscli cluster create command [See page 35] 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 44]. 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 0]

- Configure ThoughtSpot nodes in Azure [See page 13]

- Deploying on the Dell Appliance [See page 0]

- Using the tscli cluster create command [See page 35]

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 (ng-inx)
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 service port (exported by Tomcat), Callo-sum services like meta-data services, medata-dependency service, scheduling service, session-less service, spotiq service	bidirectional	All nodes	All nodes	Port where search service contacts meta-data service for metadata

Port	Protocol	Service Name	Direction	Source	Dest.	Description
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 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 metadata 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 metadata 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 metadata 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 metadata service for metadata
8080	HTTP	Tomcat	bidirectional	All nodes	All nodes	BI engine communication with clients
8081	HTTP	Callosum/Tomcat status	bidirectional	All nodes	All nodes	BI engine communication with clients
8787	HTTP	Periscope (UI) service HTTP port	bidirectional	All nodes	All nodes	Administration UI back end

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.