

Kylo

Manual Deployment Guide

December 12, 2016

**Table of contents**

[Preface](#_gjdgxs)

[Audience](#_30j0zll)

[System Requirements](#_1fob9te)

[Kylo Dependencies](#_ej4ypcix9vmt)

[Prerequisites for Installation](#_2r5ztmx56j)

[Obtaining Documentation and Submitting a Service Request](#_3znysh7)

[Installation Locations](#_2et92p0)

[Installation](#_5nrxn2z5zuc5)

[Step 1: Setup Directory](#_3dy6vkm)

[Optional - Offline Mode](#_1t3h5sf)

[Step 2: Create Linux Users and Groups](#_4d34og8)

[Step 3: Install Think Big Services](#_2s8eyo1)

[Step 4: Run the database scripts](#_17dp8vu)

[Step 5: Install and Configure Elasticsearch](#_3rdcrjn)

[Step 6: Install ActiveMQ](#_26in1rg)

[Step 7: Install Java 8](#_m1od709op966)

[Step 8: Install Java Cryptographic Extension](#_1ksv4uv)

[Step 9: Install NiFi](#_44sinio)

[Step 10: Set Permissions for HDFS](#_ft6pxeao1fxq)

[Step 11: Create a dropzone folder on the edge node for file ingest, for example:](#_z337ya)

[Complete this step for Cloudera installations ONLY](#_3j2qqm3)

[Step 12: Edit the Properties Files](#_1y810tw)

[Step 13: Final Step: Start the 3 Think Big services](#_4i7ojhp)

[Configuration](#_2xcytpi)

[Database Changes](#_1ci93xb)

[MySQL](#_3whwml4)

[Postgres](#_2bn6wsx)

[Appendix: Cloudera Configuration File Changes](#_qsh70q)

# Preface

This document explains how to install each component of the framework manually. This is useful for when you are installing across multiple edge nodes. Use this link to the install wizard if you would prefer not to do the installation manually.

|  |  |
| --- | --- |
| **Tip** | Many of the steps below are similar to running the wizard-based install. If you want to take advantage of the same scripts as the wizard you can tar up the /opt/thinkbig/setup folder and untar it to a temp directory on each node. |

Information on the operation and maintenance of the Kylo solution is available at <https://www.thinkbiganalytics.com/kylo/>.

# Audience

This guide provides step-by-step instruction for installing the Kylo application on your computer. The reader is assumed to be an IT Administrator knowledgeable of IT terms and trained with the appropriate skillset.

Refer to the Kylo Operations and Maintenance (O&M) guide for detailed instruction on how to effectively manage:

• Production processing

• Ongoing maintenance

• Performance monitoring

Guides on the Continuous Integration and Deployment of Kylo, including instructions for maintaining, supporting, and using the solution in day-to-day operational basis, are available at at <https://www.thinkbiganalytics.com/kylo/>.

# Installation Components

Installing Kylo installs the following software:

* Thinkbig Applications: Kylo provides services to produce Hive tables, generate a schema based on data brought into Hadoop, perform Spark-based transformations, track metadata, monitor feeds and SLA policies, and to publish to target systems.
* Java 8: Kylo uses the Java 8 development platform.
* NiFi: Kylo uses Apache NiFi for orchestrating data pipelines.
* ActiveMQ: Kylo uses Apache ActiveMQ to manage communications with clients and servers.
* Elasticsearch: Kylo uses Elasticsearch, a distributed, multitenant-capable full-text search engine.

# Installation Locations

Installing Kylo installs the following software at these Linux file system locations:

* Thinkbig Applications - /opt/thinkbig
* Java 8 - /opt/java/current
* NiFi - /opt/nifi/current
* ActiveMQ - /opt/activemq

# Elasticsearch - RPM installation default location

# Installation

For each step below you will need to login to your network with root access permissions. Installation commands will be executed from the command-line interface (CLI).

# Step 1: Setup Directory

Kylo is most often installed on one edge node. If you are deploying everything to one node, the setup directory would typically be:

SETUP\_DIR=/opt/thinkbig/setup

Sometimes administrators install NiFi on a second edge node to communicate with a Hortonworks or Cloudera cluster. In this case, copy the setup folder to nodes that do not have the Think Big applications installed. In that case, use this SETUP\_DIR command:

SETUP\_DIR=/tmp/thinkbig-install

# Optional - Offline Mode

If an edge node has no internet access, you can generate a TAR file that contains everything in the /opt/thinkbig/setup folder, including the downloaded application binaries.

1. Install the Kylo RPM on a node that has internet access.
2. Run "/opt/thinkbig/setup/generate-offline-install.sh.
3. Copy the /opt/thinkbig/setup/thinkbig-install.tar file to the node you install the RPM on. This can be copied to a temp directory. It doesn’t have to be put in the /opt/thinkbig/setup folder.
4. Run "tar -xvf thinkbig-install.tar".
5. Note the directory name where you untar’d the files. This will be referred to in the rest of the doc by OFFLINE\_SETUP\_DIR.

The script will download all application binaries and puts them in their respective directory in the setup folder. Last it will TAR up the setup folder.

# 

# 

# Step 2: Create Linux Users and Groups

Creation of users and groups is done manually because many organizations have their own user and group management system. Therefore we cannot script it as part of the RPM install.

|  |  |
| --- | --- |
| **Note:** | Each of these should be run on the node on which the software will be installed. |

$ useradd -r -m -s /bin/bash nifi

$ useradd -r -m -s /bin/bash thinkbig

$ useradd -r -m -s /bin/bash activemq

Confirm that the above commands created groups as intended by looking at /etc/group level in the directory. Some operating systems may not create them by default.

$ cat /etc/group

If the groups are missing, then run the following:

$ groupadd thinkbig

$ groupadd nifi

$ groupadd activemq

# Step 3: Install Think Big Services

1. Find and download the RPM file from the artifactory and place it on the host linux machine that you want to install Kylo services on.

|  |  |
| --- | --- |
| **Note:** | To use wget instead, right-click the download link and copy the url. |

http://52.203.91.75:8080/artifactory/webapp/search/artifact/?7&q=thinkbig-datalake-accelerator (requires VPN)

1. Run the Kylo RPM install.

$ rpm -ivh thinkbig-datalake-accelerator-<version>.noarch.rpm

|  |  |
| --- | --- |
| **Note:** | The RPM is hard coded at this time to install to /opt/thinkbig. |

# Step 4: Run the database scripts

The database scripts will create one schema called "thinkbig" and install to that schema. Run the following script:

$ <SETUP\_DIR>/sql/mysql/setup-mysql.sh [db\_host\_or\_ip] [db\_user] [db\_password]

|  |  |
| --- | --- |
| **Note:** | The HDP sandbox doesn't have a password set for the root user so you would run "<SETUP\_DIR>/sql/mysql/setup-mysql.sh localhost root". |

# Step 5: Install and Configure Elasticsearch

To get Kylo installed and up and running quickly, a script is provided to stand up a single node Elasticsearch instance. You can also leverage an existing Elasticsearch instance. For example, if you stand up an ELK stack you will likely want to leverage the same instance.

1. Option 1: Install Elasticsearch from our script.

|  |  |
| --- | --- |
| **Note:** | The included Elasticsearch script was meant to speed up installation in a sandbox or DEV environment. |

..

* 1. Online Mode

$ <SETUP\_DIR>/elasticsearch/install-elasticsearch.sh

* 1. Offline Mode

$ <SETUP\_DIR>/elasticsearch/install-elasticsearch.sh -o <SETUP\_DIR>

ex. /tmp/thinkbig-install/setup/elasticsearch/install-elasticsearch.sh -o /tmp/thinkbig-install/setup

1. Option 2: Use an existing Elasticsearch
2. To leverage an existing Elasticsearch instance, you must update all feed templates that you created with the correct Elasticsearch URL.You can do this by going to the "Additional Properties" tab for that feed. If you added any re-usable flow templates you will need to modify the Elasticsearch processors in NiFI.

|  |  |
| --- | --- |
| **Tip:** | To test that Elasticsearch is running type "curl localhost:9200". You should see a JSON response. |

# Step 6: Install ActiveMQ

Another script has been provided to stand up a single node ActiveMQ instance. You can also leverage an existing ActiveMQ instance.

1. Option 1: Install ActiveMQ from the script

The included ActiveMQ script was meant to speed up installation in a sandbox or DEV environment. It is not a production ready configuration.

* 1. Online Mode

$ /opt/thinkbig/setup/activemq/install-activemq.sh

* 1. Offline Mode

$ <SETUP\_DIR>/activemq/install-activemq.sh -o <SETUP\_DIR>

ex. /opt/thinkbig/setup/activemq/install-activemq.sh -o /opt/thinkbig/setup

|  |  |
| --- | --- |
| **Note:** | If installing on a different node than NiFi and thinkbig-services you will need to update the following properties: |

* + 1. /opt/nifi/ext-config/config.properties

spring.activemq.broker-url

* + 1. /opt/thinkbig/thinkbig-services/conf/application.properties

jms.activemq.broker.url

1. Option 2: Leverage an existing ActiveMQ instance

Update the below properties so that NiFI and thinkbig-services can communicate with the existing server.

* + 1. /opt/nifi/ext-config/config.properties

spring.activemq.broker-url

* + 1. /opt/thinkbig/thinkbig-services/conf/application.properties

jms.activemq.broker.url

**Installing on SUSE**

The deployment guide currently addresses installation in a Redhat based environment. There are a couple of issues installing Elasticsearch and ActiveMQ on SUSE. Below are some instructions on how to install these two on SUSE.

* **ActiveMQ**

When installing ActiveMQ, you might see the following error:

ERROR: Configuration variable JAVA\_HOME or JAVACMD is not defined correctly.

(JAVA\_HOME='', JAVACMD='java')

This indicates that ActiveMQ isn’t properly using Java as it is set in the system. To fix this issue, use the following steps to set the JAVA\_HOME directly.

1. Edit /etc/default/activemq and set JAVA\_HOME at the bottom
2. Restart ActiveMQ (service activemq restart)

* **Elasticsearch**

RPM installation isn’t supported on SUSE. To work around this issue, we created a custom init.d service script and wrote up a manual procedure to install Elasticsearch on a single node.

<https://www.elastic.co/support/matrix>

We have created a service script to make it easy to start and stop Elasticsearch, as well as leverage chkconfig to automatically start Elasticsearch when booting up the machine. Below are the instructions on how we installed Elasticsearch on a SUSE box.

1. Make sure Elasticsearch service user/group exists
2. mkdir /opt/elasticsearch
3. cd /opt/elasticsearch
4. mv /tmp/elasticsearch-2.3.5.tar.gz
5. tar -xvf elasticsearch-2.3.5.tar.gz
6. rm elasticsearch-2.3.5.tar.gz
7. ln -s elasticsearch-2.3.5 current
8. cp elasticsearch.yml elasticsearch.yml.orig
9. Modify elasticsearch.yml if you want to change the cluster name. Our copy that is installed the wizard scripts is located in /opt/thinkbig/setup/elasticsearch
10. chown -R elasticsearch:elasticsearch /opt/elasticsearch/
11. vi /etc/init.d/elasticsearch - paste in the values from /opt/thinkbig/setup/elasticsearch/init.d/sles/elasticsearch
12. Uncomment and set the java home on line 44 of the init.d file in step #10
13. chmod 755 /etc/init.d/elasticsearch
14. chkconfig elasticsearch on
15. service elasticsearch start

# Step 7: Install Java 8

|  |  |
| --- | --- |
| **Note:** | If you are installing NiFI and the thinkbig services on two separate nodes , you may need to perform this step on each node. |

There are 3 scenarios for configuring the applications with Java 8.

1. Scenario 1: Java 8 is installed on the system and is already in the classpath.

In this case you need to remove the default JAVA\_HOME used as part of the install. Run the following script:

For thinkbig-ui and thinkbig-services

$ <SETUP\_DIR>/java/remove-default-thinkbig-java-home.sh

To test this you can look at each file referenced in the scripts for thinkbig-ui and thinkbig-services to validate the 2 lines setting and exporting the JAVA\_HOME are gone.

1. Scenario 2: Install Java in the default /opt/java/current location.
   1. Install Java 8 - You can modify and use the following script if you want:
      1. Online Mode

$ <SETUP\_DIR>/java/install-java8.sh

* + 1. Offline Mode

$ <SETUP\_DIR>/java/install-java8.sh -o <SETUP\_DIR>

ex. /opt/thinkbig/setup/java/install-java8.sh -o /opt/thinkbig/setup

1. Scenario 3: Java 8 is installed on the node, but it’s not in the default JAVA\_HOME path.

If you already have Java 8 installed and want to reference that one one there is a script to remove the existing path and another script to set the new path for the thinkbig apps.

For thinkbig-ui and thinkbig-services

$ /opt/thinkbig/setup/java/remove-default-thinkbig-java-home.sh

$ /opt/thinkbig/setup/java/change-thinkbig-java-home.sh <PATH\_TO\_JAVA\_HOME>

# Step 8: Install Java Cryptographic Extension

The Java 8 install script above will automatically download and install the [Java Cryptographic Extension](http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html). This extension is required to allow encrypted property values in the Kylo configuration files. If you already have a Java 8 installed on the system, you can install the Java Cryptographic Extension by running the following script:

$ <SETUP\_DIR>/java/install-java-crypt-ext.sh <PATH\_TO\_JAVA\_HOME>

This script downloads the extension zip file and extracts the replacement jar files into the JRE security directory ($JAVA\_HOME/jre/lib/security). It will first make backup copies of the original jars it is replacing.

# Step 9: Install NiFi

You can leverage an existing NiFi installation or follow the steps in the setup directory that are used by the wizard. Note that Java 8 is required to run NiFi with our customizations. Make sure Java 8 is installed on the node.

1. Option 1: Install NiFi from our scripts

This method downloads and installs NiFi, and also installs and configures the Think Big specific libraries. This instance of NiFi is configured to store persistent data outside of the NiFi installation folder in /opt/nifi/data. This makes it easy to upgrade since you can change the version of NiFi without migrating data out of the old version.

* 1. Install NiFi
     1. Online Mode

$ <SETUP\_DIR>/nifi/install-nifi.sh

* + 1. Offline Mode

$ <SETUP\_DIR>/nifi/install-nifi.sh -o <SETUP\_DIR>

* 1. Update JAVA\_HOME (default is /opt/java/current).

$ <SETUP\_DIR>/java/change-nifi-java-home.sh <path to JAVA\_HOME>

* 1. Install Think Big specific components.

$ <SETUP\_DIR>/nifi/install-thinkbig-components.sh

1. Option 2: Leverage an existing NiFi instance

In some cases you may have a separate instance of NiFi or Hortonworks Data Flow you want to leverage. Follow the steps below to include the Think Big resources.

|  |  |
| --- | --- |
| **Note:** | If Java 8 isn't being used for the existing instance then you will be required to change it. |

* 1. Copy the <SETUP\_DIR>/nifi/thinkbig- \*.nar and thinkbig-spark- \*.jar files to the node NiFi is running on. If it’s on the same node you can skip this step.
  2. Shutdown the NiFi instance.
  3. Create folders for the jar files. You may choose to store the jars in another location if you want.

$ mkdir -p <NIFI\_HOME>/thinkbig/lib/app

* 1. Copy the thinkbig-\*.nar files to the <NIFI\_HOME>/thinkbig/lib directory.
  2. Create a directory called "app" in the <NIFI\_HOME>/lib directory.

$ mkdir <NIFI\_HOME>/lib/app

* 1. Copy the thinkbig-spark-\*.jar files to the <NIFI\_HOME>/thinkbig/lib/app directory.
  2. Create symbolic links for all of the jars. Below is an example of how to create it for one NAR file and one JAR file. At the time of this writing there are 8 NAR files and 3 spark JAR files.

$ ln -s <NIFI\_HOME>/thinkbig/lib/thinkbig-nifi-spark-nar-\*.nar <NIFI\_HOME>/lib/thinkbig-nifi-spark-nar.nar

$ ln -s <NIFI\_HOME>/thinkbig/lib/app/thinkbig-spark-interpreter-\*-jar-with-dependencies.jar <NIFI\_HOME>/lib/app/thinkbig-spark-interpreter-jar-with-dependencies.jar

* 1. Modify <NIFI\_HOME>/conf/nifi.properties and update the following property. This modifies NiFI to use our custom provenance repository to send data to the thinkbig-services application.

nifi.provenance.repository.implementation=com.thinkbiganalytics.nifi.provenance.v2.ThinkbigProvenanceEventRepository

nifi.web.http.port=8079

|  |  |
| --- | --- |
| **Note:** | If you decide to leave the port number set to the current value you must update the "nifi.rest.port" property in the thinkbig-services application.properties file. |

* 1. There is a controller service that requires a MySQL database connection. You will need to copy the MySQL connector jar to a location on the NiFI node. The pre-defined templates have the default location set to /opt/nifi/mysql.
     1. Create a folder to store the MySQL jar in.
     2. SCP the /opt/thinkbig/thinkbig-services/lib/mysql-connector-java-<version>.jar to the folder in step #1
     3. If you created a folder name other than the /opt/nifi/mysql default folder you will need to update the "MySQL" controller service and set the new location. You can do this by logging into NiFi and going to the Controller Services section on the top right.
  2. Create H2 folder for fault tolerance. If the JMS queue goes down for some reason our custom Provenance library will startup a local H2 database and store provenance events until JMS comes back up. Once back up, it will send all of the events stored in the database then shut down the local H2 instance. Below are steps to configure the H2 folder.

|  |  |
| --- | --- |
| **Note:** | Right now the plugin is hard coded to use the /opt/nifi/ext-config directory to load the properties file. There is a Jira to address this PC-261. |

* + 1. Create the folders.

$ mkdir /opt/nifi/h2

$ mkdir /opt/nifi/ext-config

* + 1. SCP the /opt/thinkbig/setup/nifi/config.properties file to the /opt/nifi/ext-config folder.
    2. Change the ownership of the above folders to the same owner that nifi runs under. For example, if nifi runs as the "nifi" user:

$ chown -R nifi:users /opt/nifi

OPTIONAL: The /opt/thinkbig/setup/nifi/install-thinkbig-components.sh contains steps to install NiFi as a service so that NiFi can startup automatically if you restart the node. This might be useful to add if it doesn't already exist for the NiFi instance.

# Step 10: Set Permissions for HDFS

This step is required on the node that NiFi is installed on to set the correct permissions for the "nifi" user to access HDFS.

1. NiFi Node - Add nifi user to the HDFS supergroup or the group defined in hdfs-site.xml, for example:

Hortonworks

$ usermod -a -G hdfs nifi

Cloudera

$ groupadd supergroup

# Add nifi and hdfs to that group:

$ usermod -a -G supergroup nifi

$ usermod -a -G supergroup hdfs

|  |  |
| --- | --- |
| **Note:** | If you want to perform actions as a root user in a development environment, run the below command. |

$ usermod -a -G supergroup root

1. thinkbig-services node - Add thinkbig user to the HDFS supergroup or the group defined in hdfs-site.xml, for example:

Hortonworks

$ usermod -a -G hdfs thinkbig

Cloudera

$ groupadd supergroup

# Add nifi and hdfs to that group:

$ usermod -a -G supergroup hdfs

|  |  |
| --- | --- |
| **Note:** | If you want to perform actions as a root user in a development environment run the below command. |

$ usermod -a -G supergroup root

1. For Clusters:

In addition to adding the nifi/thinkbig user to the supergroup on the edge node you also need to add the users/groups to the name nodes on a cluster.

Hortonworks

$ useradd thinkbig

$ useradd nifi

$ usermod -G hdfs nifi

$ usermod -G hdfs thinkbig

Cloudera - <Fill me in after testing >

# Step 11: Create a dropzone folder on the edge node for file ingest, for example:

Perform the following step on the node on which NiFI is installed:

$ mkdir -p /var/dropzone

$ chown nifi /var/dropzone

|  |  |
| --- | --- |
| **Note:** | Files should be copied into the dropzone such that user nifi can read and remove. Do not copy files with permissions set as root. |

# Complete this step for Cloudera installations ONLY

See the appendix section in the deployment guide "Cloudera Configuration File Changes" link:deployment-guide{outfilesuffix}[Deployment Guide],

# Step 12: Edit the Properties Files

# Step 13: Final Step: Start the 3 Think Big services

$ /opt/thinkbig/start-thinkbig-apps.sh

At this point all services should be running.