IBM BigInsights BigIntegrate and BigQuality: IBM InfoSphere Information Server on Hadoop Deployment and Configuration Guide

IBM® BigInsights® BigIntegrate and BigQuality allow for IBM InfoSphere® Information Server to be deployed on Apache Hadoop, leveraging resources in the Hadoop cluster through the Hadoop resource manager known as Yet Another Resource Negotiator (YARN). This offering introduces data locality, allowing for logic in existing and new IBM InfoSphere DataStage® jobs to run on the Hadoop data nodes where the Hadoop Distributed File System (HDFS) blocks exist. This IBM Redbooks® Analytics Support Web Doc is intended to jumpstart deployment and configuration of the IBM BigInsights BigIntegrate and BigQuality solution. InfoSphere Information Server on Hadoop is available starting at version 11.5.

This document covers the following topics:

- Overview
- InfoSphere Information Server on Hadoop installation
- InfoSphere Information Server on Hadoop configuration
- The APT_CONFIG_FILE environment variable
- Container resource requirements
- Log files
- Kerberos
- IBM JDK recommendations (This web doc refers to IBM SDK Java Technology as IBM JDK.)

Overview

Hadoop 2.0 moves the resource management and scheduling of jobs across the Hadoop cluster to a new resource management layer called YARN. InfoSphere Information Server on Hadoop communicates with YARN to request the containers and resources it requires to run an IBM InfoSphere DataStage job. InfoSphere Information Server on Hadoop is available for Linux platforms and supports the major Hadoop distributions.

For a full list of distributions and versions, see “Preparing Hadoop” in the IBM Knowledge Center: https://ibm.biz/Bd4Gwy

Figure 1 shows where InfoSphere Information Server on Hadoop fits into the broader Hadoop architecture.

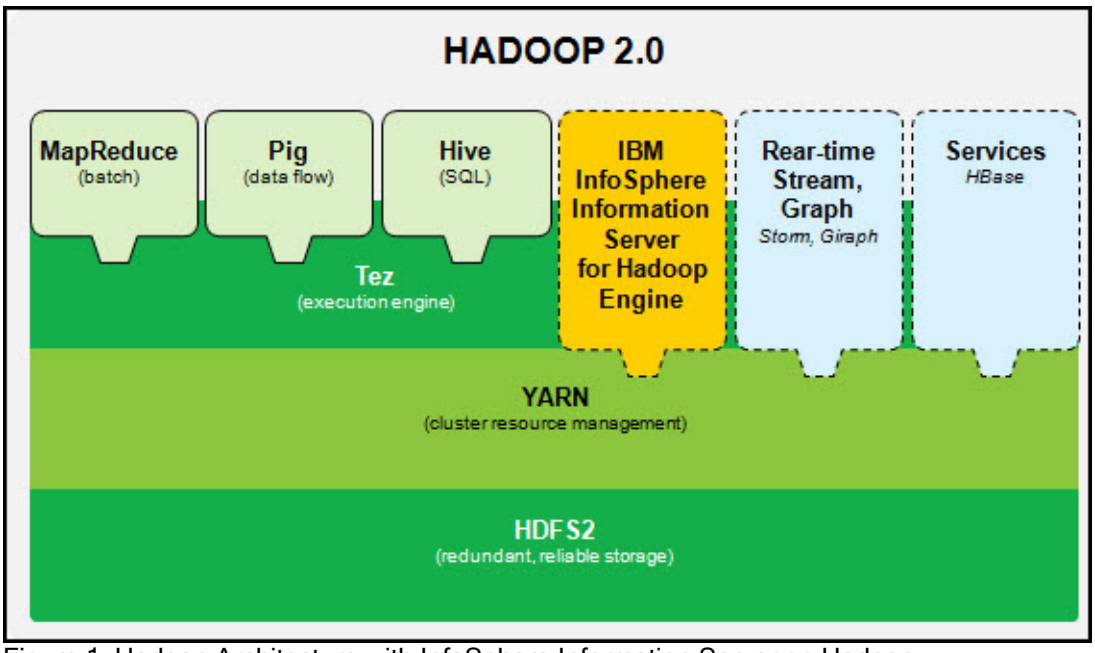

Figure 1 Hadoop Architecture with InfoSphere Information Server on Hadoop

YARN terminology to become familiar with is described in the following list:

- **Client**: A user-developed tool that submits an application to run on a YARN cluster.
- **ResourceManager (RM)**: The master process of a YARN cluster. Handles scheduling and allocation of resources to applications on the cluster.
- **NodeManager (NM)**: The worker processes on each data node. Handles launching and monitoring of container processes.
- **ApplicationMaster (AM)**: User-developed application launched in the cluster to manage the lifecycle of the application in the cluster. Can request additional containers within the cluster to run the user job.
- **Container**: A logical set of resources a process tree is given to run within the YARN cluster.

YARN configuration parameters to become familiar with are as follows:

- **yarn.nodemanager.pmem-check-enabled**: Enforce physical memory limit on containers.
- **yarn.nodemanager.vmem-check-enabled**: Enforce virtual memory limit on containers.
- **yarn.scheduler.minimum-allocation-mb**: Minimum size in megabytes (MB) for containers.
- **yarn.scheduler.maximum-allocation-mb**: Maximum size in MB for containers.

Other parameters are listed on the following Apache web page:
https://ibm.biz/Bd4nYL

Values are maintained in the `yarn-site.xml` file. See your respective Hadoop distribution for instructions to modify and update the YARN configuration.

YARN commands that can be run from any YARN client are as follows:

- **`yarn application -list`**: This command lists all running applications in YARN.
- **`yarn node -list`**: This command lists all running data nodes available to YARN.

If YARN log aggregation is enabled, the following command can be used to collect the container logs belonging to an Application Master:

```
yarn logs -applicationId [ApplicationID]
```

Where `[ApplicationID]` is the Application Master for the job run.

Most Hadoop clusters are Kerberos enabled. Kerberos is a network authentication protocol that provides strong authentication using the concept of *tickets* to allow cluster members to verify each other. Kerberos has strict time requirements, so keeping all members of the cluster's operating system (OS) clocks in sync is important. Many encryption types are supported and can be used with Kerberos. If you are using Advanced Encryption Standard 256-bit encryption (AES-256), you will need to download the IBM Java Cryptography Encryption (JCE) unrestricted policy files and copy them into the Java Development Kit (JDK) that is included with IBM InfoSphere Information Server. For additional information and configuration steps, see the IBM Knowledge Center:
https://ibm.biz/Bd4eSp

Kerberos terminology to become familiar with is described in the following list:

- **Principal**: A Kerberos principal is a service or a user. This is similar to the concept of a user name on UNIX, but it is a unique identifier for the realm.
- **Realm**: A Kerberos realm is a set of managed nodes that share the same Kerberos database.
- **Ticket**: A Kerberos ticket is what is acquired and stored from the Kerberos Key Distribution Center (KDC). Tickets are stored in what is referred to as a ticket cache. Tickets have a limited lifetime before they expire and must be renewed.
- **kinit**: UNIX command that acquires a ticket.
- **klist**: UNIX command that lists Kerberos credentials and displays the validity of ticket caches.
- **Keytab**: Stores long-term keys for one or more principals. Keytabs allow for easier authentication through kinit to establish and obtain a valid ticket.

Understanding the tier definitions for IBM InfoSphere Information Server is important before you pick a method of installation for InfoSphere Information Server on Hadoop. IBM InfoSphere Information Server consists of three tiers: *services*, *repository*, and *engine*. These tiers can be collocated or separated depending on business requirements.

For InfoSphere Information Server on Hadoop, the engine tier can be installed in two ways:

- Hadoop edge node

  In this topology, the IBM InfoSphere Information Server installation (either engine tier or all tiers) is installed on a Hadoop edge node within the cluster. An edge node is a node within the Hadoop cluster that does not contain any HDFS data, but has Hadoop client software configured and installed. This option provides the best performance and is the most common and preferred topology.

- Hadoop data node

  In this topology, the IBM InfoSphere Information Server installation (either engine tier or all tiers) is installed on a Hadoop data node within the cluster. This option is typically used for smaller clusters or single machine deployments.

The services and repository tiers can be installed on the same node that is chosen for the InfoSphere Information Server on Hadoop engine tier or they can be installed outside of the Hadoop cluster on separate servers as long as they are accessible to the engine tier through a local area network (LAN).

The Hadoop client libraries must be provisioned onto either a Hadoop edge node or a Hadoop data node depending on which type of node was chosen for installation. Client libraries can be provisioned by using Apache Ambari or whatever other cluster management tools is available for the Hadoop distribution.

InfoSphere Information Server on Hadoop installation

Complete the following steps:

1. Before installing IBM InfoSphere Information Server, ensure that the **`hostname`** and **`hostname -f`** commands both return the fully qualified domain name (FQDN) of your environment. Hadoop expects host names to be in the FQDN format.
2. After you pick a topology for installation, install IBM InfoSphere Information Server 11.5.0.1 or later on an edge or data node:
   https://ibm.biz/BdsPkx
3. Verify that your installation was successful by compiling and running a simple IBM InfoSphere DataStage job that contains a transformer stage to ensure that the compiler has been set up correctly.
4. Install the following patches:
   - Ensure that Fix Pack 1 or later is installed by installing either the suite image or Fix Pack if the environment is currently 11.5.0.0:
     https://ibm.biz/BdsPtD
   - The latest JDK. See the "IBM JDK recommendation" section later in this document.

     11.5.0.1 (Fix Pack 1) and later releases (11.5.x.x) contain all of the critical fixes listed above provided that the 11.5.0.1 suite installation is used. If you apply Fix Pack 1 to an existing 11.5.0.0 installation, you must manually apply the latest JDK. Access the InfoSphere Information Server on Hadoop download document to ensure that no further patches have been released on top of new fix packs or releases.

5. For IBM InfoSphere DataStage jobs that have connectivity to databases requiring client libraries, similar to a massively parallel process (MPP) environment, each data node will need database clients installed for all databases that are planned to be used as a source or target. If you want to limit the number of database client installations on the data nodes, a node map constraint can be used for jobs that have connectivity to databases. The node map forces those jobs to use only the nodes that have the client (or clients) installed.

After confirming that your installation was a success, you are ready to begin configuring the environment for InfoSphere Information Server on Hadoop.

InfoSphere Information Server on Hadoop configuration

This section includes details about binary transfer, permissions for job processes and directories, environment variables, and configuration parameters.

Binary transfer to Hadoop data nodes

Four options for binary transfer and distribution are available:

- HDFS
- Passwordless Secure Shell (SSH)
- Copy-orchdist utility
- Network File System (NFS)

This document focuses on the default and preferred option for binary transfer, HDFS. InfoSphere Information Server on Hadoop uses HDFS to manage and copy the necessary InfoSphere Information Server binaries when you run jobs.

A check is performed when the PX YARN client starts to compare the `Version.xml` file checksum on the engine tier to the version that is stored in HDFS. If a difference exists, the updated binaries from the engine tier are transferred into HDFS. The `/tmp/copy-orchdist.log` file can be checked for progress. When new patches are installed to the engine tier, the `Version.xml` file will be updated, triggering binary localization into HDFS for the next time the PX YARN client is restarted. Binary localization from HDFS to the data node (or nodes) is performed after a job is invoked on that particular data node. When the binaries on the data node match the copy that is in HDFS, no binary localization will occur for subsequent job runs until the version in HDFS is updated.

Note that configuration file changes on the engine tier to files, such as `.odbc.ini, ishdfs.config, isjdbc.config`, and others, will not trigger binary localization.

To force binary localization for configuration files, such as those listed above, run the following commands:

```
cd $DSHOME/../..
echo '<!-- Forced PX Yarn Client Binary Localization on' "`date` -->" >>
Version.xml
cd $APT_ORCHHOME/etc/yarn_conf
./stop-pxyarn.sh
./start-pxyarn.sh
```

The binary localization generates tar commands beginning at specific directory structures and includes everything beneath the starting directory. Additional files and directories (custom configuration files, and so on) are localized only if they exist in the following directories:

```
/opt/IBM/InformationServer/Server/DSEngine
/opt/IBM/InformationServer/Server/PXEngine
/opt/IBM/InformationServer/Server/DSComponents
/opt/IBM/InformationServer/Server/DSParallel
/opt/IBM/InformationServer/ASBNode
/opt/IBM/InformationServer/jdk
/opt/IBM/InformationServer/Server/StagingArea
/opt/IBM/InformationServer/Server/branded_odbc
```

To avoid startup time delay for the job, the preferred technique is to create a simple IBM InfoSphere DataStage job that contains a static configuration file of all data nodes in the Hadoop cluster to force binary localization. This technique can be used after patch installations to ensure no further delays during job run time.

The `/tmp` location or a path defined by APT_YARN_BINARIES_PATH must have at least 5 GB of free space on all nodes (engine tier and data nodes) to localize the IBM InfoSphere Information Server binaries.

Determining required permissions for job processes

Depending on the configuration of the Hadoop cluster, the user running the job processes on the data nodes can do one of the following two options:

1. The owner of the PX YARN client process on the edge node. This is typically the IBM InfoSphere DataStage administrator, dsadm, which is the default OS user or if you have the APT_YARN_MULTIPLE_USERS environment variable set the engine credential user is used.
2. The YARN administrative user, typically the OS user, yarn.

Option 1 is used when the Hadoop cluster is Kerberos enabled or if in the `yarn-site.xml` configuration file, the following parameters are set:

- `yarn.nodemanager.container-executor.class=org.apache.hadoop.yarn.server.nodemanager.LinuxContainerExecutor`
- `yarn.nodemanager.linux-container-executor.nonsecure-mode.limit-users=false`

Otherwise option 2 is used. If you use option 2, note that the YARN user should be added to the primary group of the IBM InfoSphere DataStage administrator user, typically dstage.

Depending on the option you selected, the next step is to configure the user (or users), to have valid Hadoop user permissions:
Note: These commands might require elevated HDFS permissions.

- Create a home directory in Hadoop for the user (or users):
  ```
  hdfs dfs -mkdir /user/<username>
  ```
- Change ownership on the home directory to the user:
  ```
  hdfs dfs –chown <username>:<group> /user/<username>
  ```

Typically, the <group> will match the primary group for the IBM InfoSphere DataStage administrative user, dsadm, and will be dstage.

Create directories and verify permissions

When using HDFS or passwordless SSH for binary transfer, the following commands should be run on each data node in the Hadoop cluster:

- `mkdir -p /pathTo/IBM/InformationServer`
- `mkdir -p /pathTo/IBM/InformationServer/Server/Projects/<ProjectName>`

  The product installation paths should match the installation path for the IBM InfoSphere Information Server engine tier.

  Users must have read, write, and execute permissions to access these directories. These directories must be created on all data nodes under the above directory structure.
- `mkdir -p /pathTo/scratch`

  The scratch disk location, as defined in APT_CONFIG_FILE, should be local to the data nodes for optimal performance. The scratch disk location must be pre-created. The amount of scratch space required on each data node depends on the size of data and the degree of parallelism that is being used in the parallel job. For instance, sorts of large data sets can require a large amount of scratch space (local disk), which might not be readily apparent to Hadoop administrators because most disk space on data nodes are dedicated to HDFS or the OS itself. Ensure that adequate disk space is available in the location selected for scratch space.
- `hdfs dfs -mkdir -p /pathTo/ResourceDisk`

  The resource disk location, as defined in APT_CONFIG_FILE, must be in HDFS (by default) and must be pre-created in HDFS.

Environment variables and configuration parameters

The `yarnconfig.cfg` or `yarnconfig.cfg.default` file in `IBM/InformationServer/Server/PXEngine/etc/yarn_conf` contains environment variables and configuration parameters for InfoSphere Information Server on Hadoop.

APT_YARN_CONFIG should point to the location of the `yarnconfig.cfg` file.

A preferred practice is to store `yarnconfig.cfg` in `/IBM/InformationServer/Server/DSEngine` ($DSHOME). This will protect your configuration file from being overwritten when a patch is installed. For example, installing Fix Pack 1 will replace the `PXEngine/etc/yarn_conf` directory. This means any customization to the `yarnconfig.cfg` configuration file will be overwritten.

To implement this practice, follow these steps:

1. Move `yarnconfig.cfg` to $DSHOME (`/IBM/InformationServer/Server/DSEngine`).
2. Change the value of APT_YARN_CONFIG to `/IBM/InformationServer/Server/DSEngine/yarnconfig.cfg`.

Subsequent updates (new variables, and so on) to `yarnconfig.cfg` are shown in the example configuration file, which, after the patch installs, will be renamed `yarnconfig.cfg.default` in the `PXEngine/etc/yarn_conf` directory. Review the `yarnconfig.cfg.default` file after patch installation (or installations) so any new variables can be added to the `yarnconfig.cfg` file in $DSHOME.

Similar to the IBM InfoSphere DataStage environment file, `dsenv`, all parameters set in the `yarnconfig.cfg` file will be used by default for a job run, provided APT_YARN_CONFIG is set at the job, project, or dsenv level.
Most environment variables can be overwritten at the job level by setting them to a custom value within the job properties.

Key environment variables are as follows:

- APT_YARN_MODE: If set to the default value of true or 1, jobs will be run through the YARN resource manager, otherwise jobs will be run in the normal MPP manner.
- APT_YARN_CONTAINER_SIZE: This parameter defines the size in MBs that will be requested from YARN for the container. The IBM InfoSphere DataStage parallel engine section leader and player processes will run within the container. Note that if this value is set to the default, 64 MB, it is overwritten by the lower bound that is defined in YARN by the `yarn.scheduler.minimum-allocation-mb` parameter. Therefore, if `yarn.scheduler.minimum-allocation-mb` exceeds the APT_YARN_CONTAINER_SIZE value, the APT_YARN_CONTAINER_SIZE value will be ignored. An important concept to realize is that if `yarn.scheduler.minimum-allocation-mb` is set too high and a large number of jobs are executed concurrently on the Hadoop cluster, resource usage can spike. Conversely, if APT_YARN_CONTAINER_SIZE is set higher than the value of `yarn.scheduler.maximum-allocation-mb`, the value of APT_YARN_CONTAINER_SIZE will be ignored and instead set to the value of `yarn.scheduler.maximum-allocation-mb`.
- APT_YARN_AM_POOL_SIZE: This parameter defines the number of Application Masters (AMs) that will remain running in YARN at all times. The purpose of the pool is to reduce job startup time by having AMs started and waiting for jobs to run. The default for this parameter is 2.
- APT_YARN_MULTIPLE_USERS: If set to true, this parameter will allow for multiple PX YARN clients running on the engine tier, one for each engine credential user running an IBM InfoSphere DataStage job. Remember that if a PX YARN client is not running for a user that submits a job, an instance will be automatically started. The PX YARN client for that user will continue running on the IBM InfoSphere DataStage engine unless it is manually shut down. The APT_YARN_AM_POOL_SIZE applies to each running PX YARN client.
- APT_YARN_CONNECTOR_USER_CACHE_CRED_PATH: This parameter, if set, allows for the localization of the Kerberos credential cache from the engine tier when using the ODBC Connector. This is useful if the data nodes do not contain the credential cache.
- APT_YARN_FC_DEFAULT_KEYTAB_PATH: This parameter if set, allows for the localization of a Kerberos keytab file from the engine tier when using the File Connector. This is useful if the data nodes do not contain the keytab file. Remember that a keytab file can contain multiple principals if needed.

Issues with path performance and possible warning messages, such as the following example, might occur in the log:

```
WARN util.NativeCodeLoader: Unable to load native-hadoop library for your
platform... using builtin-java classes where applicable.
WARN shortcircuit.DomainSocketFactory: The short-circuit local reads feature
cannot be used because libhadoop cannot be loaded.
```

To avoid the issues, set LD_LIBRARY_PATH to contain the Hadoop native libraries; add the following line to the `dsenv` file:

```
export LD_LIBRARY_PATH=/pathToHadoop/hadoop/lib/native/:$LD_LIBRARY_PATH
```

The location should be set to where `libhadoop.so` exists:
https://ibm.biz/Bd4eeS

APT_CONFIG_FILE environment variable

This variable points to a configuration file that is used by the IBM InfoSphere DataStage parallel engine to determine the nodes and resources that will be used for the IBM InfoSphere DataStage job execution. When using InfoSphere Information Server on Hadoop, new options are available to the standard configuration file:

- Ability to define a dynamic node using “$host”
- Ability to define node duplication using “instances x” where x is the number of identical nodes you want to define

The following example shows a static, 30-node configuration file with 30 logical nodes on `mymachine.domain.com`; it uses the new `instances x` option:

```
{
   node "node1"
   {
      fastname "mymachine.domain.com"
      pools ""
      resource disk "/mydisk/ " {pools ""}
      resource scratchdisk "/myscratch/ " {pools ""}
      instances 30
   }
}
```

The next example shows a dynamic, 30-node configuration file with 30 logical or physical nodes; the `$host` option will be populated by the Hadoop YARN resource manager based on resource availability in the Hadoop cluster:

```
{
   node "node0"
 {
  fastname "the-engine-tier-machine.domain.com"
  pools "conductor"
  resource disk "/mydisk/" {pools ""}
  resource scratchdisk "/myscratch/" {}
  }
   node "node1"
  {
   fastname "$host"
   pools ""
   resource disk "/mydisk/" {pool ""}
   resource scratchdisk "/myscratch" {pools ""}
   instances 30
  }
}
```

Container resource requirements

The Hadoop YARN resource manager assigns a set of resources for a particular container to use. In the case of IBM InfoSphere DataStage jobs, sometimes these resource definitions are too low for what the job needs to run successfully. For example, for larger jobs that contain lookups or other high memory usage stages, the value of APT_YARN_CONTAINER_SIZE might need to be increased at the job level. Although the IBM InfoSphere DataStage job log will indicate whether the APT_YARN_CONTAINER_SIZE needs to be increased, the easiest way to determine what value is needed is by looking at the YARN NodeManager logs on the Hadoop data node where the container was running. If you are using a dynamic configuration file, set APT_DUMP_SCORE and APT_PM_SHOW_PIDS to see messages in the log that print out where each process is running. The message in the NodeManager log will be similar to the following example:

```
WARN  monitor.ContainersMonitorImpl (ContainersMonitorImpl.java:run(508)) -
Container [pid=2652,containerID=container_e06_1459973705125_0002_01_000003] is
running beyond physical memory limits. Current usage: 192.7 MB of 64 MB
physical memory used; 1.8 GB of 2.0 GB virtual memory used. Killing container.
```

In this example, for both APT_YARN_CONTAINER_SIZE=64 and `yarn.scheduler.minimum-allocation-mb=64`, an appropriate value for APT_YARN_CONTAINER_SIZE would be 256 MB.

Also possible is to exceed the maximum virtual memory upper bound for the container. The message will be similar in the NodeManager logs.

The message in the IBM InfoSphere DataStage job log will be similar to the following example:

```
main_program: If this is the only error seen it may indicate that
APT_YARN_CONTAINER_SIZE with value 64 is set too low for this job resulting in
YARN killing the container. This can be confirmed by looking in the YARN
Resource Manager log.
```

A couple of ways are available to address this issue, in addition to trying to set the APT_YARN_CONTAINER_SIZE environment variable:

- Set YARN parameter `yarn.nodemanager.pmem-check-enabled` (for physical memory) or `yarn.nodemanager.vmem-check-enabled` (for virtual memory) to false. This way allows the container to request and use whatever resources it needs and trust that the data node has enough resources to handle the request. This approach is most similar to how normal IBM InfoSphere DataStage Parallel engine processes are run, however the preferred way is to work with your Hadoop team and IBM InfoSphere DataStage developers to find a good set of boundaries for the container's memory size.
- Tune lower and upper bound within the YARN configuration. Make the changes through your respective Hadoop distribution's preferred method. The YARN parameters that impact the lower and upper bounds are `yarn.scheduler.minimum-allocation-mb` and `yarn.scheduler.maximum-allocation-mb`.

Log files

Several log files are available to review for job failures or issues with the PX YARN client:

- `/tmp/yarn_client.[USERID].out`

  This log shows errors and logging from the PX YARN client startup.
- `/tmp/copy-orchdist.log`

  This log shows binary localization details from the IBM InfoSphere Information Server engine to HDFS.
- `/IBM/InformationServer/Server/PXEngine/logs/yarn_logs`

  This log shows the PX YARN client logging after the PX YARN client has started.

YARN Logs

The following parameters in the `yarn-site.xml` file are relevant to determining the location to key YARN logs and localized files:

- **yarn.nodemanager.log-dirs**: Contains the storage location of the container logs.
- **yarn.nodemanager.delete.debug-delay-sec**: Number of seconds before the NodeManager cleans up logs and files that are related to a container's execution. Set the value to 600 to allow for time to debug container startup issues.
- **yarn.log-aggregation-enable**: This parameter determines whether YARN log aggregation is enabled.

Parameter if YARN Log aggregation is enabled:

- **yarn.nodemanager.remote-app-log-dir**: This parameter controls the location where logs are aggregated, typically in HDFS.

Environment variable:

- YARN_LOG_DIR: This environment variable is typically in `yarn-env.sh`. It defines where YARN NodeManager logs are stored, which in most cases is `/var/log/hadoop-yarn`. These logs will contain helpful messages concerning container resource sizes, and so on.

Kerberos

When working with InfoSphere Information Server on Hadoop that is deployed to a Kerberos enabled cluster, remember to always have a valid, non-expired ticket. One of the easiest ways to ensure you have this valid ticket is to configure a crontab entry to automatically renew or request the ticket. An example crontab entry is as follows:

```
30 1 * * * kinit USER/edgenode.ibm.com@IBM.COM -A -k -t ~/USER.keytab
```

Another method is to configure Pluggable Authentication Module (PAM) for Linux to authenticate against the Kerberos libraries which can be configured to obtain a valid ticket cache.

Many implementations of Kerberos client libraries exist, such as these examples:

- MIT Kerberos (`/usr/bin/klist`)
- IBM JDK (`/IBM/InformationServer/jdk/jre/bin/klist`)
- Another vendor's JDK, which is typically included with the Hadoop distribution

Be sure you understand which client libraries that the various components of InfoSphere Information Server on Hadoop are built against.

The components that have hard requirements for the Kerberos client libraries are listed in Table 1.

Table 1 IBM InfoSphere Information Server components and Kerberos client libraries used

Component	Kerberos client
PX Engine (DataSets in HDFS, and others)	IBM JDK
File Connector Stage	IBM JDK
PX YARN client	MIT or Hadoop JDK

Kerberos recommendation

Use IBM JDK's kinit to generate a ticket cache and then set environment variable KRB5CCNAME to the value of that ticket cache (`~/krb5cc_$user`). Doing so allows for MIT's kinit, and other Kerberos client libraries to work with the IBM JDK ticket cache.

Other vendor or Kerberos clients can be used to generate the ticket cache, but the IBM JDK must be upgraded to at least version 1.7 SR3 FP30 in order for the IBM JDK to work with the generated ticket cache using the KRB5CCNAME environment variable.

IBM JDK recommendation

Upgrade JDK to the latest release, which can resolve some known issues with Kerberos, including but not limited to IV74633:
https://ibm.biz/Bd4nfP

To upgrade the IBM JDK that is used with IBM InfoSphere Information Server, see one of the security bulletins at the following web page:
https://ibm.biz/Bd4nfb

Note that IBM Fix Central will display the latest IBM JDK version that is certified by IBM InfoSphere Information Server, which might supersede the fix mentioned in the security bulletin.

Related publications and more information

See the following resources:

- InfoSphere Information Server on Hadoop in the IBM Knowledge Center:
  https://ibm.biz/Bd4ecD
- Kerberos:
  https://ibm.biz/Bd4nW7
- Apache Hadoop:
  https://ibm.biz/BdFZyM

For questions or comments, contact Scott Brokaw at slbrokaw@us.ibm.com.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided “AS IS”, without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

This document was created or updated on December 6, 2016.

Send us your comments in one of the following ways:

- Use the online **Contact us** review form found at:
  ibm.com/redbooks
- Send your comments in an e-mail to:
  redbooks@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400 U.S.A.

This document is available online at http://www.ibm.com/redbooks/abstracts/tips1339.html .

Trademarks