DSS Installation

Date of Publish: 2017-12-22

http://docs.hortonworks.com

Contents

Installation Overview 3

Installing the Data Steward Studio App 3

Set Up a Local Repository 3
Create the Repository Configuration File 4
Install the Data Steward Studio Service App 5
Enable the Data Steward Studio in the Data Plane Service 6
Add Users and Assign Roles for DSS App 6
Add Data Lakes to the Data Plane Service 7

Installing Profiler Agent on Clusters 7

Supported Configurations for DSS Installation 8
Pre-installation tasks for Data Plane Profiler Agent 9
Configure External Database 10
Configure MySQL external database 10
Configure Postgres external database 10
Grant Permissions in Ranger 10
Install the Data Plane Profiler Agent 10
Ambari Dataplane Profiler Configs 12
Set up Knox topologies 18

Installation Overview

You must install the Data Plane Service, Data Steward Studio app, and Data Plane Profiler Agent in the same order.

To install Data Steward Studio, you must install the following components:

1. Data Plane Service
2. Data Steward Studio app
3. Data Plane Profiler Agent on HDP Clusters

You are strongly encouraged to read completely through this entire document before starting the installation process, to that you understand the interdependencies and order of the steps.

Installing the Data Steward Studio App

After installing the DPS Platform, you must install the Data Steward Studio app.

About this task

Data Steward Studio app must be installed on the same host as DPS Platform. You can install one DPS service or any combination of DPS services with the DPS Platform.

Procedure

1. Set up a local repository.
2. Create the repository configuration file.
3. Install the Data Steward Studio app.
4. Enable the Data Steward Studio in the Data Plane Service.
5. Add users and assign roles for DSS app.
6. Add data lakes to the Data Plane service.

Set Up a Local Repository

Setting up a local repository involves moving the tarball to the selected mirror server and extracting the tarball to create the repository.

Before you begin

Ensure that you have downloaded the required tarballs from the customer portal, following the instructions provided as part of the product procurement process.

You must have completed the preparatory tasks before setting up a repository.

Procedure

1. Copy the repository tarballs to the web server directory and expand (uncompress) the archive file:
   a) Navigate to the web server directory you previously created.
   cd /var/www/html/

   All content in this directory is served by the web server.
   b) Move the tarballs to the current directory and expand each of the repository tarballs that you downloaded.

   Replace <file-name> with the actual name of the RPM tarball that you are expanding.

   tar zxvf <file-name>.tar.gz

During expansion of the tarball, subdirectories are created in /var/www/html/, such as DSS/centos7. These directories contain the repositories.

Expanding the DSS app tarball takes several seconds.

2. Confirm that you can browse to the newly created local repositories by using the base URLs: http://<webserver-host-name>/<repo-name>/<OS>/<service-version-X>

   - <webserver-host-name>

     This is the FQDN of the web server host.
   - <repo-name>

     This is composed of the abbreviated name of the repository, such as DSS.
   - <OS>

     This is the operating system version.
   - <service-version-X>

     This is the version number of the downloaded repository with an appended unique number.

   Base URL Examples

   DSS Base URL:

   ```
   http://webserver.com:port/DSS/centos7/1.1.0.0-X
   ```

   Be sure to record these Base URLs, because you need them when installing DSS app on the host, and installing the associated agent on the clusters.
3. If you have multiple repositories configured in your environment, deploy the following plugin on all the nodes in your cluster.

   ```
   yum install yum-plugin-priorities
   ```
4. Edit the /etc/yum/pluginconf.d/priorities.conf file to add the following values:

   ```
   [main]
   enabled=1
   gpgcheck=0
   ```

Results

The repositories for DSS are now prepared for installation.

What to do next

Create the configuration file for the DSS repository.

Create the Repository Configuration File

A repository configuration file must be created for the DSS Service on the DPS host. The file is required to identify the path to the repository data, establish whether a GPG signature check should be performed on the repository packages, etc. Only one repository configuration file is needed.

Procedure

1. Navigate to the repository directory.
   cd /etc/yum.repos.d/
2. Create a repository file.
   vi dss-app.repo

Alternatively, you can copy an existing repository file to edit.

3. Add the following content in the repository file:
#VERSION_NUMBER=<downloaded-version#> [<service-name-abbreviation>]

   This is composed of the service name abbreviation and version number (includes the build number). Example: DSS-APP-1.0.0.0-59

```
name=<service-name-abbreviation> Version - <service-name-abbreviation>
```

```
baseurl=http://<webserver-host-name>/<directory-containing-repo>
```

   <webserver-host-name> is the FQDN of the web server host that contains the repository. This is the same base URL that you used in the task to prepare the repositories.

   <directory-containing-repo> is the path expanded from the tarball.

```
gpgcheck=1
```

```
gpgkey=http://<webserver-host-name>/<directory-containing-repo>/RPM-GPG-
KEY/RPM-GPG-KEY-Jenkins
```

```
enabled=1
```

```
priority=1
```

   Example Repository File

```
#VERSION_NUMBER=1.0.0.0-59
[DSS-APP-1.0.0.0-59]
name=DSS-APP Version - DSS-APP-1.0.0.0-59
baseurl=http://<your_webserver>:port/DSS-APP/centos7/1.0.0.0
gpgcheck=1
gpgkey=http://<your_webserver>:port/DSS-APP/centos7/1.0.0.0/RPM-GPG-KEY/
RPM-GPG-KEY-Jenkins
enabled=1
priority=1
```

Install the Data Steward Studio Service App

Follow the instructions to install the Data Steward Studio Service app.

Before you begin

You must have successfully installed DPS Platform and DPS is running.

Procedure

1. Log in as root to the host on which you set up the DPS repositories.

```
sudo su
```

2. Install the RPMs for the DSS service application.

```
yum install dss-app
```

A folder is created that contains the Docker image tarball files and a configuration script.

If the yum command fails, then the local repository was not set up correctly. Check the repository file /etc/yum.repos.d/dss-app.repo on the host.

3. Navigate to the directory containing the installation scripts for the DSS service, for example:
cd /usr/dss-app/current/apps/dss/bin
4. Load the DSS Docker images and initialize the environment.
./dssdeploy.sh init

   Loading the images might take a while.

   Note:

   If you run into errors while deploying, you must destroy the deployment using ./dssdeploy.sh destroy command and re-install the app. To check the logs of the dss-app container, you can use the command ./dssdeploy.sh logs.
5. Verify that the container you installed is running.
./dssdeploy.sh ps

   Make sure that the container with the name dss-app is running.

Enable the Data Steward Studio in the Data Plane Service

After installing the Data Steward Studio service app, you must enable in the Data Plane Service Platform.

Procedure

1. Log in to the DPS Platform as a DataPlane Admin user.
2. Click the Services icon in the DPS Platform navigation pane.

   The Services page displays. Services identified by a tile icon are available to be enabled.
3. Move the cursor over the Data Steward Studio service and click the Enable button that appears.

   A verification page is displayed.

   Note:

   If the Data Steward Studio Service button displays Not Installed, it indicates that the DSS app is not started as mentioned in the previous section. Follow the steps in Install the Data Steward Studio app section and retry enabling.
4. Enter the SmartSense ID and click **Verify**.

   The ID is case-sensitive. You can retrieve the SmartSense ID from the Hortonworks Support Portal under the Tools tab.
5. Click **Next**.

   The DSS service displays in the Enabled list on the Services page.

Add Users and Assign Roles for DSS App

After you set up the LDAP configuration for DPS Platform, you need to add users for the DSS app. During LDAP configuration, you added users and groups that can log in as DPS Admin. You must now assign roles to users and groups, which allow users to access the services that plug into DPS.

About this task

You must select the Data Steward role for accessing the Data Steward Studio Service. Users and groups should be assigned this role to access Data Steward Studio service. To enable the Data Steward Studio role, see Role Management section of the *Data Plane Service Administration Guide.*

Before you begin

User accounts must already exist within your corporate LDAP prior to adding the user to DPS Platform.

The DataPlane Admin role is required to perform this task.

Procedure

1. Log in to the DPS Platform.
2. Click the (Users) icon in the DPS Platform navigation pane.
3. On the Users and Groups page, click **Add User**.
4. Enter the name of the user.

   With your own LDAP server, the user must already exist within your corporate LDAP. If you are using the packaged LDAP, enter one of the predefined users (guest, sam, tom). The name auto-populates as you type.

   Tip:

   You must click the name of the user when it displays and ensure it appears in the Username field on a dark background.

   If the name appears on a white background, it means the name is not recognized and the action fails.
5. Select the Data Steward role to assign to the user:

   Data Steward - Can perform all actions in the Data Steward Studio service UI, and can manage DSS-enabled clusters in DPS Platform.
6. Click **Save**.

   You can log in and see Data Steward Studio service inside the DPS Platform. If Data Steward role is the only role assigned, you will be directed to the Data Steward Studio Service. If you have more roles assigned, you can select the Data Steward Studio Service in the navigation menu in the top left corner.

   Note: If you assign the Data Steward role to yourself or to the group that you belong to, you must log out and log in again to verify that Data Steward Studio Service is available.

   The new user displays in the list on the Users page.

Add Data Lakes to the Data Plane Service

Make sure to add data lakes to the Data Plane Service to access them in the Data Steward Studio Service.

Procedure

Register a cluster in the DPS Platform. For more information, see the *Data Plane Service Administration Guide*.

Data Steward Studio can only work with clusters that are identified as datalakes. Clusters that have Atlas and Ranger installed can be identified as datalakes. For more information, see the *Add Clusters* section in the *DPS Administration Guide*. Users with DPS Admin role can only add clusters on the DPS platform.

Installing Profiler Agent on Clusters

After installing the Data Steward Studio app, you must install the DP Profiler Agent on clusters to complete the installation of Data Steward Studio service.

Procedure

1. Make sure all the prerequisites are met.
2. Complete the pre-installation tasks for Data Plane Profiler.
3. Configure an external database.

4. Install the Data Plane Profiler Agent.
5. Configure the Ambari Data Plane Profiler Properties.
6. Set up Knox topologies.

Supported Configurations for DSS Installation

Requirements for the DSS host

The DSS application is installed on the same host at DPS Platform and has no requirements beyond what is required by DPS Platform.

Cross-version support for DSS application and engine

Table 1: Support by engine or app version

Engine or App Version	Supports...
DP Profiler Agent 1.0	HDP 2.6.5
DPS 1.1 and DSS 1.0 UI application	DP Profiler Agent 1.0

Requirements for clusters used with Dataplane Profiler Agent

The clusters on which you install the Dataplane Profiler Agent must meet the requirements identified in the following sections.

Table 2: Version requirements for clusters used with DSS

Item	Versions
HDP versions	2.6.5
Ambari versions	2.6.2.0

You can find the most current information about your product's interoperability for this release on the Support Matrix. The Support Matrix tool provides information about:

- Operating Systems
- Databases
- Browsers
- JDKs

To access the tool, go to: https://supportmatrix.hortonworks.com

HDP Apache Component Requirements for DSS

The following additional Apache components are required on your clusters for DSS support:

Component	Purpose	Comments
Atlas	Required for metadata search and discovery	
Hive	Only Hive assets are currently supported in DSS for management	
Spark2	Required for running profiler jobs	
Livy2	Required for submitting profiler jobs to the cluster	
Knox	Required for authentication, federation, and proxying	Knox must be enabled on clusters before you can register the clusters with DPS.

Component	Purpose	Comments
Ranger	Required for looking at security policies and mining audit information	

Port and Network Requirements for clusters

Have the following ports available and open on each cluster:

Default Port Number	Purpose	Comments	Required to be open?
21900	Port for Dataplane Profiler service on hosts.	Accessibility is required from all clusters.	Yes
8080	Ambari server host		Yes
6080	Ranger Port		Yes
8443	Knox Port		Yes
21000	Atlas Port		Yes
8999	Livy2 Port		Yes

Pre-installation tasks for Data Plane Profiler Agent

Perform these tasks before you try to install the Data Profiler agent on the cluster.

Procedure

1. Ensure that the clusters are running the latest version of HDP.
2. Ensure that the following HDP components are installed and configured:
   - Atlas
   - Ranger
   - Knox
   - Spark2 and Livy Server2
3. If you plan to sync users from LDAP into Ranger, ensure a dpprofiler user is created in LDAP and synced into Ranger.
4. Make sure that HDFS Audit logging for Ranger is enabled.
5. Add the following proxy users details in the custom core-site.xml file as follows:

   hadoop.proxyuser.livy.groups=* hadoop.proxyuser.livy.hosts=* hadoop.proxyuser.knox.groups=* hadoop.proxyuser.knox.hosts=*
6. If the cluster is kerberos-enabled, go to the Kerberos configuration section in Ambari and look up the value of the global property called principal suffix. Go to the Spark2 service and access the Custom livy2-conf section and add this property.

   livy.superusers=dpprofiler${principalsuffix}
7. Ensure that the following configuration is set up in Spark2 for cleaning up history files without filling up HDFS space over time.
   a) Log in to Ambari on the cluster.
   b) Select Spark2 > Configs > Custom spark2-defaults.
   c) Add the following lines:

      spark.history.fs.cleaner.enabled=true spark.history.fs.cleaner.interval=1d spark.history.fs.cleaner.maxAge=7d
      This ensures that Spark history from jobs older than seven days will be cleaned up once per day. Modify the values as needed.
8. Restart the services as required.

Configure External Database

You must configure an external database and add the profileragent database user to the database.

Configure MySQL external database

You must configure an MySQL database and add the profileragent database user to the database.

About this task

MySQL is supported only if Ambari is installed to use on MySQL.

Procedure

1. Log in to your MySQL database client.
2. Create a database user. The default value is profileragent.
3. Create a database name. The default value is profileragent.
4. Grant the user profileragent all rights on the database profileragent.

Configure Postgres external database

You must configure a Postgres database and add the profileragent database user to the database.

Procedure

1. Log in to postgres shell using admin user like postgres:
   psql -U postgres
2. Create profileragent database and user and grant all previleges:

   CREATE DATABASE $profileragentdb;

   CREATE USER $profileragentuser WITH PASSWORD '$password';

   GRANT ALL PRIVILEGES ON DATABASE $profileragentdb TO $profileragentuser;

   The default value for $profileragentdb and $profileragentuser is profileragent.
3. Add $profileragentuser to pg_hba.conf to have access from profiler agent host.

Grant Permissions in Ranger

The DP Profiler user needs permissions to read entities for the profilers to function. Such a policy should be created via Ranger.

About this task

The dpprofiler user needs access to the following:

- Read and list tables from the Hive metastore
- Read and write types, entities, and classifications in Atlas
- Run jobs in YARN against a configured queue

Install the Data Plane Profiler Agent

DSS requires that the DP Profiler Agent be installed on all custers. The Profiler is installed on the Ambari host, using an Ambari management pack (MPack). An MPack bundles service definitions, stack definitions, and stack add-on service definitions.

About this task

This task must be completed on all clusters to be used with DSS.

Before you begin

You must have root access to the Ambari Server host node to perform this task.

> **Important:** Prior to starting installation, you must have downloaded the required repository tarballs from the Hortonworks customer portal, following the instructions provided as part of the product procurement process. The repository tarballs for the Data Plane Profiler agent are different from the DSS app repository tarballs.

Procedure

1. Log in as root to an Ambari host on a cluster.

```
ssh root@<ambari-ip-address>
```

2. Install the Data Plane Profiler MPack by running the following command, replacing <mpack-file-name> with the name of the MPack.

```
ambari-server install-mpack --mpack <mpack-file-name> --verbose
```

3. Restart the Ambari server.

```
ambari-server restart
```

4. Launch Ambari in a browser and log in.
   http://<ambari-server-host>:8080

   Default credentials are:

   Username: admin
   Password: admin

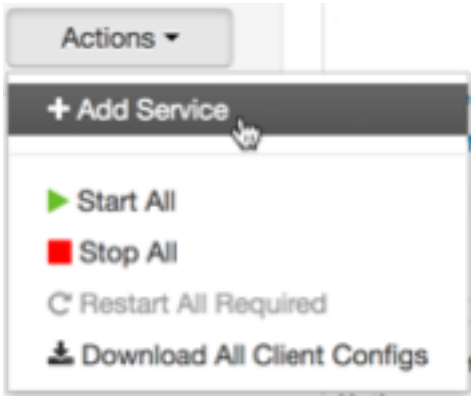
5. Click **Admin**>**Manage Ambari**.

6. Click **Versions**, and then do the following on the Versions page:
   a) Click the HDP version in the **Name** column.
   b) Change the **Base URL** path for the DSS service to point to the local repository, for example:

```
http://webserver.com/DSS/centos7/1.1.0.0-X
```

   URLs shown are for example purposes only. Actual URLs might be different.

7. Click the Ambari logo to return to the main Ambari page.
8. In the Ambari Services navigation pane, click **Actions**>**Add Service**.



The Add Service Wizard displays.

9. On the **Choose Services** page of the Wizard, select the Dataplane Profiler service to install in Ambari, and then follow the on-screen instructions.

   Other required services are automatically selected.

10. When prompted to confirm addition of dependent services, give a positive confirmation to all.

    This adds other required services.

11. On the **Assign Masters** page, you can choose the default settings.
12. On the **Customize Services** page, fill out the database details and other required fields that are highlighted.

    Make sure to enter the credentials that you set while configuring the external database. Change the username profileragent to the values set in the external database.

    > **Note:** Make sure to add the database driver to the machine based on the external database that you configured.

13. If you are using Postgres as your external database, during installation, the mpack suggests an incorrect database url after selecting the database type, host, and user. Change the database url as follows:

    Suggested URL: jdbc:postgres://hostname:5432/profileragent

    Correct URL: jdbc:postgresql://hostname:5432/profileragent

14. Complete the remaining installation wizard steps and exit the wizard.
15. Ensure that all components required for your DPS service have started successfully.
16. Open the quick link of the profiler for service verification.
17. Add /profilers to the quick link URL.

    If the quick link is xyz:21900, change it to xyz:21900/profilers.

    > **Note:** For non-Kerberized clusters, this request returns the list of all registered profilers. For kerberos-enabled clusters, you will see an HTTP-401 response which is expected.

18. After installing the profiler agent using **Add Service Wizard** in Ambari, the NodeManager hosts do not have the dpprofiler user. For Ambari to automatically create these users, restart all NodeManagers by going to **Services->YARN->Restart NodeManagers** (NodeManagers can be restarted in a rolling fashion - Ambari UI shows restart batching options)

Ambari Dataplane Profiler Configs

From **Ambari** > **Dataplane Profiler** > **Configs**, you can view or update your database or advanced configurations.

Dataplane Profiler Database Configs

From **Ambari** > **Dataplane Profiler** > **Configs** > **Database**, you can view or update the DataPlane Profiler Database configurations.

Table 3: Database configs

Value	Description	Example
DP Profiler Database	Database type or flavor used for DSS profiler.	H2 MySQL POSTGRES
Slick JDBC Driver Class	System driver that is used to connect to the database. **Important:** Do not modify.	H2: slick.driver.H2Driver$ MySQL: slick.driver.MySQLDriver$ POSTGRES: slick.driver.PostgresDriver$
Database Username	A Database user needs to be created in the MySQL or Postgres DB that the profiler service would use to connect to the DB. This is name of that database user.	profileragent
Database Name	Name must be “profileragent”. **Important:** Do not modify.	profileragent
Database URL	The URL of DP profiler database.	H2: jdbc:h2:/var/lib/profiler_agent/h2/profileragent;DATABASE_TO_UPPER=false;DB_CLOSE_DEL MySQL: jdbc:mysql://hostname:3306/profileragent?autoreconnect=true POSTGRES: jdbc:postgresql://hostname:5432/profileragent
Database Host	Database host name for Profiler Agent server	<hostname>
JDBC Driver Class	Driver name for your profiler database. **Important:** Do not modify.	H2: org.h2.Driver MySQL: com.mysql.jdbc.Driver POSTGRES: org.postgresql.Driver
Database password	The password for your DP database.	<your_password>

Dataplane Profiler Advanced Configs

From **Ambari** > **Dataplane Profiler** > **Configs** > **Advanced**, you can view or update the DataPlane Profiler advanced configurations.

Table 4: Advanced dpprofiler-config

Value	Description	Example
Cluster Configs	Provides various cluster configurations, including: atlasUrl rangerAuditDir metastoreUrl metastoreKeytab metastorePrincipal	atlasUrl=application-properties/atlas.rest.address;rangerAuditDir=ranger-env/xasecure.audit.destination.hdfs.dir;metastoreUrl=hive-site/hive.metastore.uris;metastoreKeytab=hive-site/hive.metastore.kerberos.keytab.file;metastorePrincipal=hive-site/hive.metastore.kerberos.principal
Job Status Refresh in seconds	How often the profiler job status should refresh, in seconds.	15
Autoregister profilers	Looks for the profilers in {Profilers local Dir} directory and install them (if not installed) at the time of startup.	true
Profilers local Dir	Local directory for the profilers.	/usr/dss/current/profilers

Value	Description	Example
Profilers DWH Dir	The HDFS directory where DSS Profilers will store their metrics output. Ensure the dpprofiler user has full access to this directory.	/user/dpprofiler/dwh
Profilers Hdfs Dir	HDFS directory for the profilers.	/apps/dpprofiler/profilers
Refresh table cron	The format is a standard CRON expression. This will periodically refresh the metrics cache.	0 0/30 * * * ?
Refresh table retry	Number of time profiler agent will retry to clear cache in case of error.	3
Partitioned table location for sensitive tags	Metric name where Hive sensitive information is stored in partitioned format. **Important:** Do not modify.	hivesensitivitypartitioned
Partitioned table location for all sensitive tags	Metric name where Hive sensitive information is stored. **Important:** Do not modify.	hivesensitivity
SPNEGO Cookie Name	Cookie name that is returned to the client after successful SPNEGO authentication.	dpprofiler.spnego.cookie
SPNEGO Signature Secret	Secret for verifying and signing the generated cookie after successful authentication	***some***secret**
Submitter Batch Size	Max number of assets to be submitted in one profiler job.	50
Submitter Max Jobs	Number of profiler jobs active at a point in time. This is per profiler.	2
Submitter Job Scan Time	Time in seconds after which the profiler looks for an asset in the queue and schedules the job if the queue is not empty.	30

Value	Description	Example
Submitter Queue Size	Max size of the profiler queue. After which it rejects any new asset submission request.	500
Livy Session Config	Specifies the configuration required for interactive Livy sessions the profiler creates. These sessions will be swapped with new ones based on their lifetime. Lifetime of session is decided by the configurations below. session.lifetime.minutes - Session lifetime in minutes after its creation before it will be swapped. session.lifetime.requests - Maximum number of requests a session can process before it will swapped. session.max.errors - Number of adjacent errors after which session will be swapped There are two separate session.config sections describing interactive session's Spark configurations. Both read and write has same schema and following Livy session properties can be specified here. name,heartbeatTimeoutInSecond,driverMemory,…mExecutors,queue For more on above properties refer to Livy documentation. The properties session.config.read.timeoutInSeconds and session.config.write.timeoutInSeconds specifies timeouts for requests using interactive session. Notes: session.starting.message and session.dead.message are for internal use. **Important:** Do not modify. It is advisable to have a separate YARN queue for sessions created by the profiler.	session { lifetime { minutes = 2880 requests = 500 } max.errors = 20 starting.message = "java.lang.IllegalStateException: Session is in state starting" dead.message = "java.lang.IllegalStateException: Session is in state dead" config { read { name = "dpprofiler-read" heartbeatTimeoutInSecond = 172800 timeoutInSeconds = 90 driverMemory = "5G" driverCores = 4 executorMemory = "4G" executorCores = 2 numExecutors = 25 queue = "profilerqueue" } write { name = "dpprofiler-write" heartbeatTimeoutInSecond = 172800

Table 5: Advanced dpprofiler-env

Value	Description	Example
dpprofiler.conf.dir	Configuration files directory.	/etc/profiler_agent/conf
dpprofiler.data.dir	Data directory. If using h2, data is stored here.	/var/lib/profiler_agent
dpprofiler.http.port	Port where profiler agent runs.	21900
dpprofiler.kerberos.enabled	True if Kerberos is enabled.	false
dpprofiler.kerberos.keytab	Profiler agent keytab location.	/etc/security/keytabs/ dpprofiler.kerberos.keytab
dpprofiler.kerberos.principal	Profiler agent kerberos principal.	dpprofiler${principalSuffix}@REALM.COM principalSuffix is a random string which is generated by Ambari for a cluster. This string is used to uniquely identify services on a cluster in case of multiple clusters being managed by single KDC
dpprofiler.log.dir	Log Directory	/var/log/profiler_agent
dpprofiler.pid.dir	Pid Directory	/var/run/profiler_agent
dpprofiler.spengo.kerberos.keytab	SPNEGO keytab location.	/etc/security/keytabs/spnego.service.keytab

Value	Description	Example
dpprofiler.spnego.kerberos.principal	SPNEGO Kerberos principal.	HTTP/${FQDN}@REALM.COM FQDN - fully qualified domain name of the machine
logback.content	Content for logback.xml.	<configuration> <conversionRule conversionWord="coloredLevel" converterClass="play.api.libs.log > <appender name="FILE" class="ch.qos.logback.core.FileAp <file>{{dpprofiler_log_dir}}/ application.log</file> <encoder> <pattern>%date [%level] from %logger in %thread - %message%n %xException</pattern> </encoder> </appender> <appender name="STDOUT" class="ch.qos.logback.core.Consol <encoder> <pattern>%coloredLevel %logger{15} - %message %n%xException{10}</ pattern> </encoder> </appender> <appender name="ASYNCFILE" class="ch.qos.logback.classic.Asy <appender-ref ref="FILE" /> </appender> <appender name="ASYNCSTDOUT" class="ch.qos.logback.classic.Asy <appender-ref ref="STDOUT" /> </appender> <logger name="play" level="INFO" /> <logger name="application" level="DEBUG" /> <!-- Off these ones as they are annoying, and anyway we manage configuration ourselves --> <logger name="com.avaje.ebean.config.Prop level="OFF" /> <logger name="com.avaje.ebeaninternal.ser level="OFF" /> <logger name="com.avaje.ebeaninternal.ser level="OFF" />

Table 6: Custom dpprofiler-config

Value	Description	Example
dpprofiler.user	User for Profiler Agent **Important:** Do not modify.	dpprofiler

Set up Knox topologies

After configuring the external database, you must set up the knox topologies.

About this task

To access the Profiler agent behind Knox gateway, set up the two Knox topologies - dp-proxy.xml and token.xml.

Procedure

1. Log in to the cluster as root user.
2. Create two topology files - dp-proxy.xml and token.xml in Knox on the HDP cluster.
   For more information about setting up these files, see *DPS Installation Guide*.
3. On each cluster, place the dp-proxy.xml and token.xml in the Knox topologies folder.
   a) Navigate to the topologies folder using this command:
      cd /etc/knox/conf/topologies
   b) Paste the dp-proxy.xml and token.xml files to this folder.
4. Add the Profiler Agent Server address in the the dp-proxy.xml file as follows:

```
<service>
<role>PROFILER-AGENT</role>
<url>URI of the server address</url>
</service>
```