DSS Getting Started

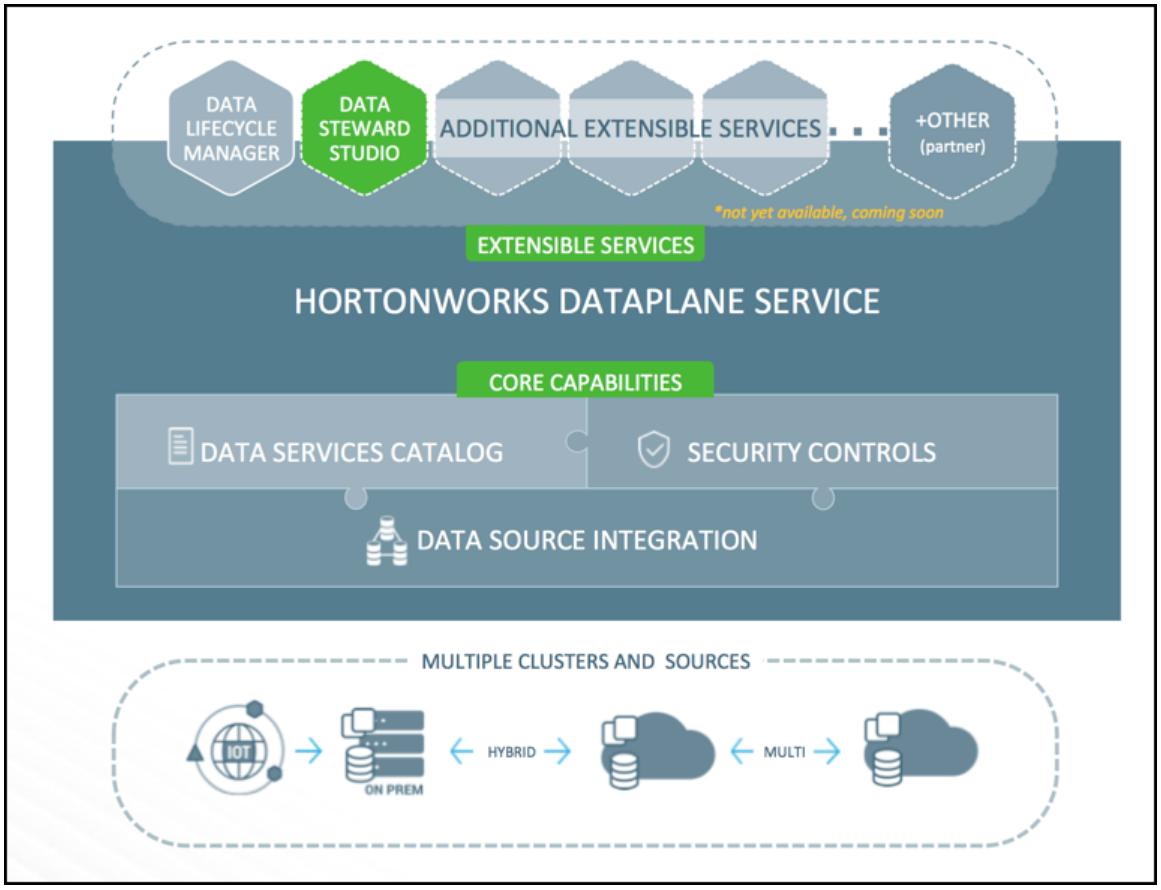
Date of Publish: 2018-05-16

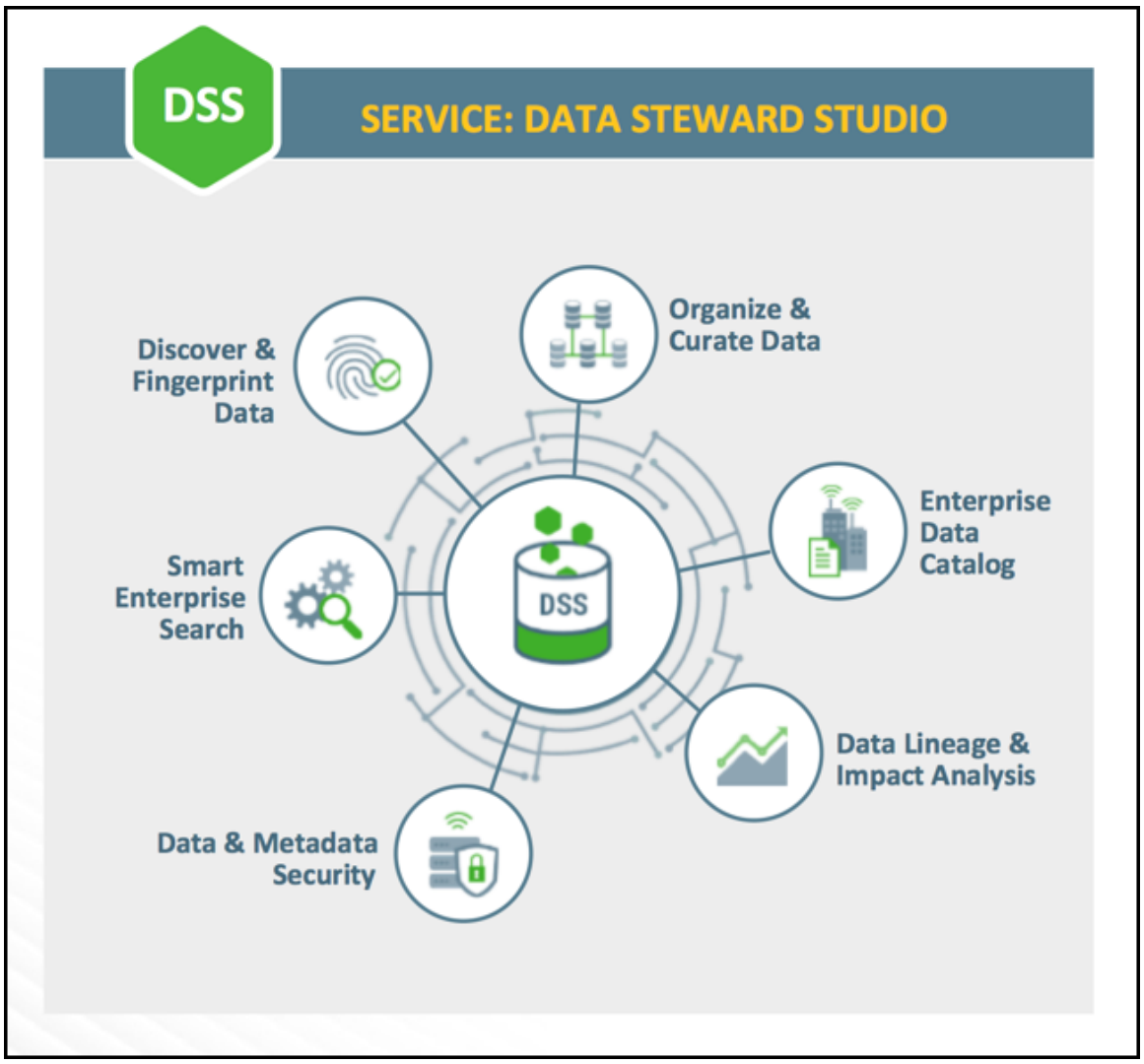
http://docs.hortonworks.com

Contents

Data Steward Studio Overview

Data Steward Studio (DSS) is a service within Hortonworks DataPlane Service (DPS) platform that enables you to understand, manage, secure, and govern data assets across enterprise data lakes. DSS helps you understand data across multiple clusters and across multiple environments (on-premises, cloud, and IOT).





Data Steward Studio enables data stewards across the enterprise to work with data assets in the following ways:

- Organize and curate data globally
  - Organize data based on business classifications, purpose, protections needed, etc.
  - Promote responsible collaboration across enterprise data workers
- Understand where relevant data is located
  - Catalog and search to locate relevant data of interest (sensitive data, commonly used, high risk data, etc.)
  - Understand what types of sensitive personal data exists and where it is located
- Understand how data is interpreted for use
  - View basic descriptions: schema, classifications (business cataloging), and encodings
  - View statistical models and parameters
  - View user annotations, wrangling scripts, view definitions etc.
- Understand how data is created and modified
  - Visualize upstream lineage and downstream impact
  - Understand how schema or data evolve
  - View and understand data supply chain (pipelines, versioning, and evolution)
- Understand how data access is secured and protected, and audit use

- Understand who can see which data and metadata (for example, based on business classifications) and under what conditions (security policies, data protection, anonymization)
- View who has accessed what data from a forensic audit or compliance perspective
- Visualize access patterns and identify anomalies

Related reference

DP Platform support requirements

Understanding Asset Collections

An asset collection is a group of assets that fit search criteria so that you can manage and administer them collectively.

Asset collections enable you to perform the following tasks when working with your data:

- Organize

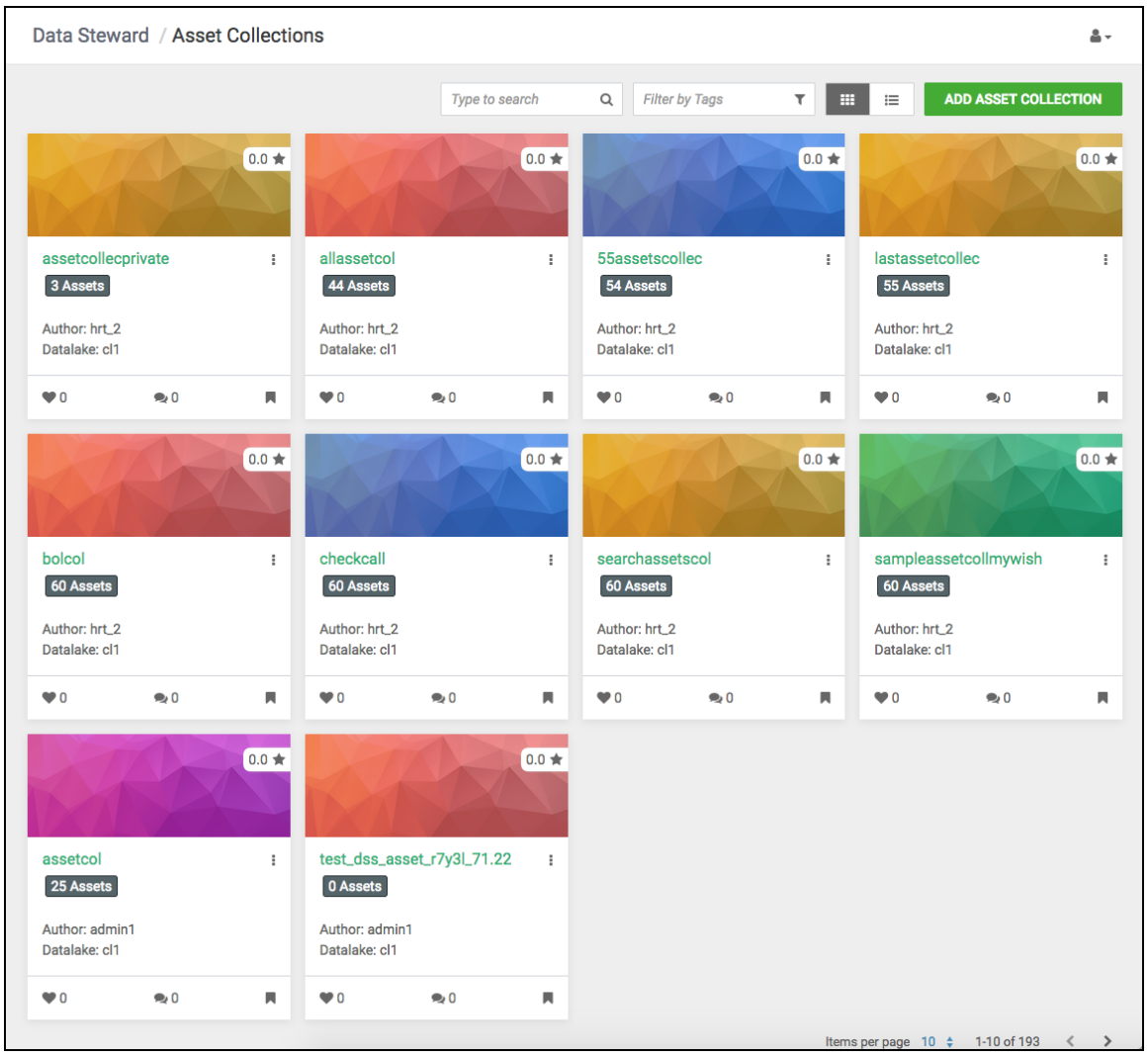
  Group data assets into asset collections based on business classifications, purpose, protections, relevance, etc.
- Search

  Find tags or assets in your data lake using Hive assets, attribute facets, or free text.

  Advanced asset search uses facets of technical and business metadata about the assets, such as those captured in Apache Atlas, to help users define and build collections of interest. Advanced search conditions are a subset of attributes for the Apache Atlas type hive_table.
- Summarize

  View dashboards with an overview of data assets within an asset collection.
- Understand

  Audit data asset security and use for anomaly detection, forensic audit and compliance, and proper control mechanisms.



You can edit Asset Collections after you create them and the assets contained within the collection will be updated. CRUD (Create, Read, Update, Delete) is supported for Asset Collections.

Note: Asset Collections must have less than 130 assets.

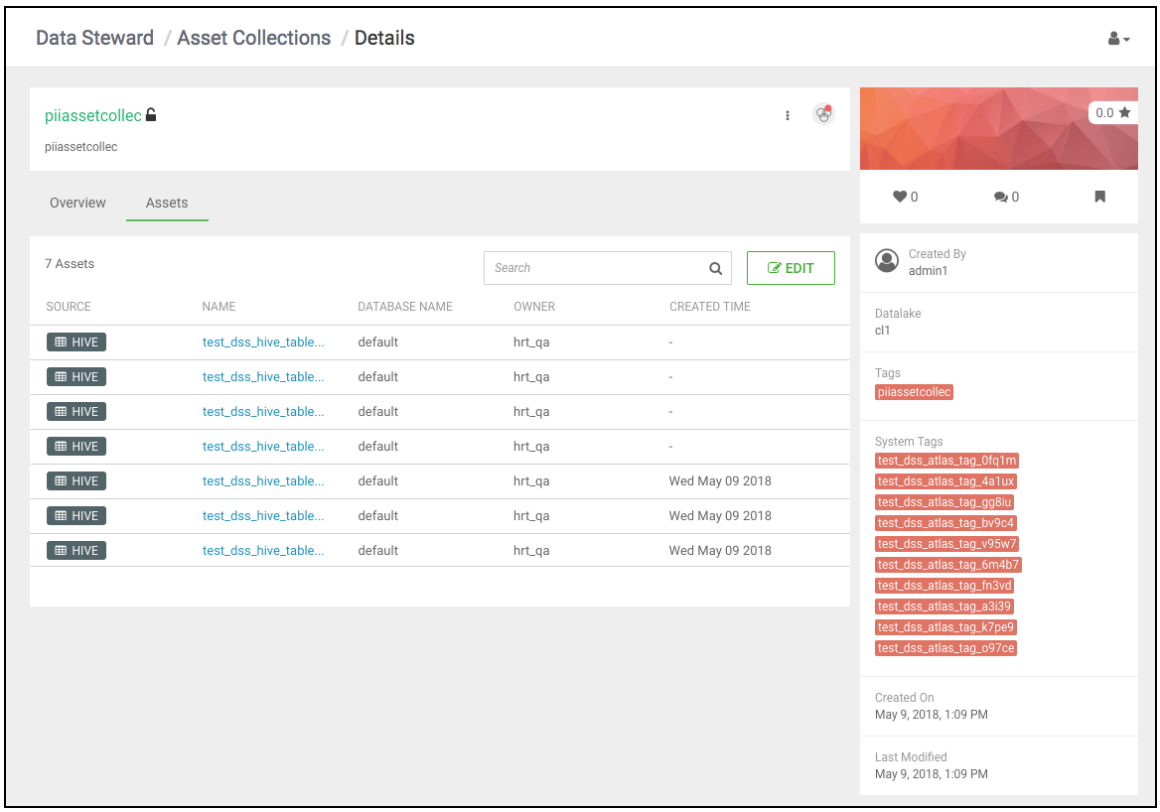
Related Information

Atlas API Reference

Understanding Data Assets

A data asset is a specific instance of a data type, including the related attributes and metadata. A data asset is a physical asset located in the Hadoop ecosystem, such as a Hive table, that contains business or technical data.

Data assets are also known as “entities” in Apache Atlas.



Understanding the DSS Profiler

Data Steward Studio (DSS) includes a profiler engine that can run data profiling operations as a pipeline on data located in multiple data lakes. You can install the profiler agent in a data lake and set up a specific schedule to generate various types of data profiles. Data profilers generate metadata annotations on the assets for various purposes.

For example, data profilers can create summarized information about contents of an asset and also provide annotations that indicate its shape (such as distribution of values in a box plot or histogram).

When you create an Asset Collection, all data assets in that collection are added to a scheduler in the profiler backend agent. You cannot manually trigger the profiler to run; you can set the global refresh rate in **Ambari** > **Dataplane Profiler** > **Configs** > **Advanced** > **Refresh table cron**.

Understanding the Sensitive Data Profiler (SDP)

The Sensitive Data Profiler automatically performs context and content inspection to detect various types of sensitive data and suggest suitable classifications or tags based on the type of sensitive content detected or discovered.

Auto-detected data types

- Bank account
- Credit card
- Driver number (UK)
- Email
- IBAN number
    - Austria (AUT)
    - Belgium (BEL)

    - Bulgaria (BGR)
    - Switzerland (CHE)
    - Cyprus (CYP)
    - Czech Republic (CZE)
    - Germany (DEU)
    - Denmark (DNK)
    - Spain (ESP)
    - Estonia (EST)
    - Finland (FIN)
    - France (FRA)
    - United Kingdom (GBR)
    - Greece (GRC)
    - Croatia (HRV)
    - Hungary (HUN)
    - Ireland (IRL)
    - Iceland (ISL)
    - Italy (ITA)
    - Liechtenstein (LIE)
    - Lithuania (LTU)
    - Latvia (LVA)
    - Luxembourg (LUX)
    - Malta (MLT)
    - Netherlands (NLD)
    - Norway (NOR)
    - Poland (POL)
    - Portugal (PRT)
    - Romania (ROU)
    - Slovakia (SVK)
    - Slovenia (SVN)
    - Sweden (SWE)
- IP address
- NPI
- Name
- National ID number
    - Bulgaria (BGR)
    - Switzerland (CHE)
    - Czech Republic (CZE)
    - Denmark (DNK)
    - Spain (ESP)
    - Estonia (EST)
    - Finland (FIN)
    - Greece (GRC)
    - Ireland (IRL)
    - Iceland (ISL)
    - Italy (ITA)
    - Lithuania (LTU)
    - Latvia (LVA)
    - Norway (NOR)
    - Poland (POL)

- Portugal (PRT)
- Romania (ROU)
- Slovakia (SVK)
- Sweden (SWE)

- National insurance number (UK)
- Passport number
    - Austria (AUT)
    - Belgium (BEL)
    - Switzerland (CHE)
    - Germany (DEU)
    - Spain (ESP)
    - Finland (FIN)
    - France (FRA)
    - Greece (GRC)
    - Ireland (IRL)
    - Italy (ITA)
    - Poland (POL)
    - United Kingdom (UK)

    - Bank Routing Number
    - US Social Security Number
    - Society for Worldwide Interbank Financial Telecommunication (SWIFT)
    - Telephone

Related Information

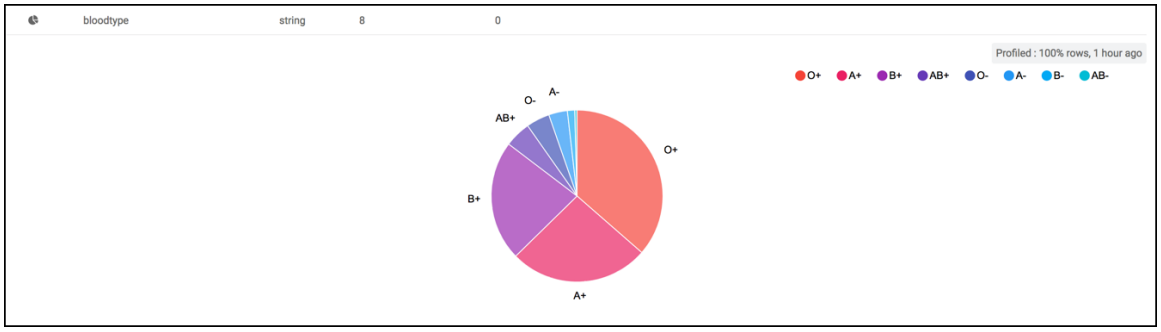
GDPR Compliance

Understanding the Hive Column Statistical Profile

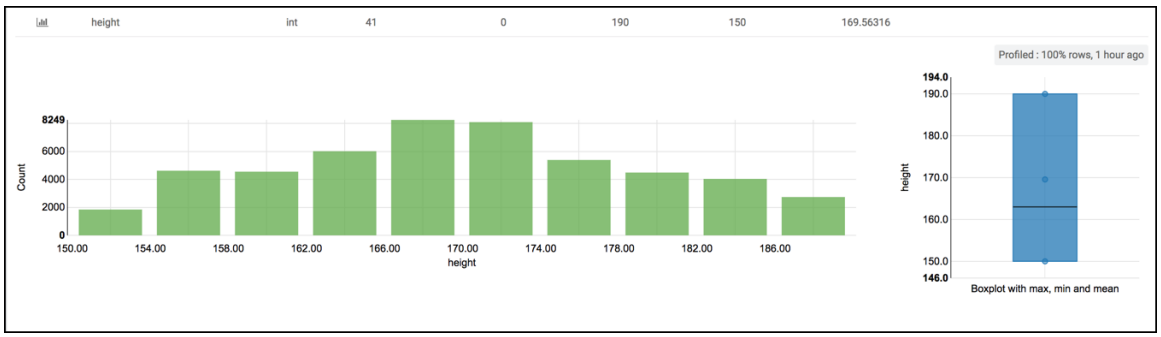
You can view the shape or distribution characteristics of the columnar data within a Hive table based on the Hive column profiler.

There are different charts available to help visualize the shape and distribution of the data within the column as well as summary statistics (such as means, null count, and cardinality of the data) for a column. The profiler computes column univariate statistics that are displayed using an appropriate chart in the **Schema** tab.

Pie charts are presented for categorical data with limited number of categories or classes. Examples include data such as blood group, gender, and titles that only have a fixed list of values (categories or labels).



When the data within columns is numeric, a histogram of the distribution of values organized into 10 groups (decile frequency histogram) and a box plot with a five-number summary (mean, median, quartiles, maximum, and minimum values) are shown for the column.



Understanding the Hive Metastore Profiler

The metadata profiler scans the Hive Metastore to retrieves information about the number of hive tables that have been added every day, computes the number of partitions, and finds values like time created, size, number of rows, input format, output format, etc. Information provided by this profiler is used in the data lake and data asset dashboards.

Understanding the Ranger Audit Profiler

You can view who has accessed which data from a forensic audit or compliance perspective, visualize access patterns, and identify anomalies in access patterns using the Ranger audit profiler.

The audit profiler uses the Apache Ranger audit logs to show the most recent raw audit event data as well as summarized views of audits by type of access and access outcomes (allowed/denied). Such summarized views are obtained by profiling audit records in the data lake with the audit profiler.

DSS Terminology

An overview of terminology used in Data Steward Studio.

Hortonworks DataPlane Service (HDS)	The family of components that include the Hortonworks DataPlane Service platform and all services that plug into it.
Profiler	Enables the data steward to gather and view information about different relevant characteristics of data such as shape, distribution, quality, and sensitivity which are important to understand and use the data effectively. For example, view the distribution between males and females in column “Gender”, or min/max/mean/null values in a column named “avg_income”. Profiled data is generated on a periodic basis from the profilers, which run at regularly scheduled intervals. Works with data sourced from Apache Ranger Audit Logs, Apache Atlas Metadata Store, and Hive.
Data Lake	A trusted and governed data repository that stores, processes, and access to many kinds of enterprise data to support data discovery, data preparation, analytics, insights, and predictive analytics. In the context of Hortonworks DPS, a data lake can be realized in practice with an Apache Ambari managed Hadoop cluster that runs Apache Atlas for metadata and governance services,

	and Apache Knox and Apache Ranger for security services.
Data Asset	A data asset is a physical asset located in the Hadoop ecosystem such as a Hive table which contains business or technical data. A data asset could include a specific instance of an Apache Hive database, table, or column. An asset can belong to multiple asset collections. Data assets are equivalent to “entities” in Apache Atlas.
Asset Collection	Asset collections allow users of DSS to manage and govern various kinds of data objects as a single unit through a unified interface. Asset collections help organize and curate information about many assets based on many facets including data content and metadata, such as size/schema/tags/alterations, lineage, and impact on processes and downstream objects in addition to the display of security and governance policies. The content of an asset collection is a static list that can only be modified by a user. So, adding new assets to (or removing from) a collection must be done manually.

Related Concepts

DPS Platform terminology

Data Lifecycle Manager terminology

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 1: 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

Value	Description	Example
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 2: 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
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

Value	Description	Example
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 3: 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.logb > <appender name="FILE" class="ch.qos.logback.core.FileApp <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 4: Custom dpprofiler-config

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

Security requirements for DSS-enabled clusters

You must configure a minimum set of security actions on each HDP cluster as part of configuring security for DLM-enabled clusters. You can perform any additional security-related tasks as appropriate for your environment and company policies.

Ensure that the following tasks were completed during cluster installation. You must configure Apache Atlas and Apache Knox SSO before you can use DSS.

Table 5: Minimum Security Requirements Checklist for DSS

Task	Instructions	Found in...	Comments
Enable Atlas in Ambari	Installing and Configuring Apache Atlas Using Ambari	HDP Data Governance guide	
Configure LDAP with Atlas	Customize Services	HDP Data Governance guide	Adapt the instructions for Ranger
Configure Ranger plugin for Atlas	Enabling Ranger Plugins: Atlas	HDP Security guide	
Configure Knox SSO for Atlas	Setting up Knox SSO for Atlas	HDP Security guide	
Configure Knox SSO for Ranger	Setting up Knox SSO for Ranger	HDP Security guide	