Data Access 3

Accessing data using Apache Druid

Date of Publish: 2018-07-12

http://docs.hortonworks.com

Contents

Apache Druid introduction

HDP 3.x includes Apache Druid (incubating). Druid is an open-source, column-oriented data store for online analytical processing (OLAP) queries on event data. Druid is optimized for time-series data analysis and supports the following data analytics features:

- Real-time streaming data ingestion
- Automatic data summarization
- Scalability to trillions of events and petabytes of data
- Sub-second query latency
- Approximate algorithms, such as hyperLogLog and theta

Druid is designed for enterprise-scale business intelligence (BI) applications in environments that require minimal latency and high availability. Applications running interactive queries can "slice and dice" data in motion.

You can use Druid as a data store to return BI about streaming data from user activity on a website or multidevice entertainment platform, from consumer events sent over by a data aggregator, or from a large set of transactions or Internet events.

HDP includes Druid 0.12.1, which is licensed under the Apache License, version 2.0.

Related Information

The druid.io documentation

Apache Druid architectural overview

Druid Architecture

Apache Druid (incubating) supports streaming ingestion and batch ingestion data analytics modes. A Druid cluster consists of several Druid node types and components. Each Druid node is optimized to serve particular functions. The following list is an overview of Druid node types:

- Realtime nodes

  These nodes ingest and index streaming data that is generated by system events. The nodes construct segments from the data and store the segments until these segments are sent to historical nodes. The realtime nodes do not store segments after the segments are transferred.
- Historical nodes

  These nodes are designed to serve queries over immutable, historical data. Historical nodes download immutable, read-optimized Druid segments from deep storage and use memory-mapped files to load them into available memory. Each historical node tracks the segments it has loaded in ZooKeeper and transmits this information to other nodes of the Druid cluster when needed.
- Broker nodes

  These nodes form a gateway between external clients and various historical and realtime nodes. External clients send queries to broker nodes. The nodes then break each query into smaller queries based on the location of segments for the queried interval and forwards them to the appropriate historical or realtime nodes. Broker nodes merge query results and send them back to the client. These nodes can also be configured to use a local or distributed cache for caching query results for individual segments.
- Coordinator nodes

  These nodes mainly serve to assign segments to historical nodes, handle data replication, and to ensure that segments are distributed evenly across the historical nodes. They also provide a UI to manage different datasources and configure rules to load data and drop data for individual datas sources. The UI can be accessed via Ambari Quick Links.

- Middle manager nodes

  These nodes are responsible for running various tasks related to data ingestion, realtime indexing, and segment archives. Each Druid task is run as a separate JVM.
- Overlord nodes

  These nodes handle task management and maintain a task queue that consists of user-submitted tasks. The queue is processed by assigning tasks in order to the middle manager nodes, which actually run the tasks.The overlord nodes also support a UI that provides a view of the current task queue and access to task logs. The UI can be accessed via Ambari Quick Links for Druid.

To use Druid in a real-world environment, the cluster must have access to the following resources to make Druid operational in HDP:

- ZooKeeper:

  A Druid instance requires that you select Apache ZooKeeper as a Service when you add Druid to the cluster; otherwise, Ambari does not add Druid to the cluster. ZooKeeper coordinates Druid nodes and manages elections among coordinator and overlord nodes.
- Deep storage:

  HDFS or Amazon S3 can be used as the deep storage layer for Druid in HDP. In Ambari, you can select HDFS as a Service for this storage layer. Alternatively, you can set up Druid to use Amazon S3 as the deep storage layer by setting the druid.storage.type property to s3. The cluster relies on the distributed file system to store Druid segments for permanent backup of the data.
- Metadata storage:

  The metadata store is used to persist information about Druid segments and tasks. MySQL and Postgres are supported metadata stores. You can select the metadata database when you install and configure Druid with Ambari.
- Batch execution engine:

  Select YARN + MapReduce2 for the execution resource manager and execution engine, respectively. Druid Hadoop index tasks use MapReduce jobs for distributed ingestion of large amounts of data.
- (Optional) Druid metrics reporting:

  If you plan to monitor Druid performance metrics using Grafana dashboards in Ambari, select Ambari Metrics System as a Service.

If you plan to deploy high availability (HA) on a Druid cluster, you need to know which components to install and how to configure the installation so that the Druid instance is primed for an HA environment.

Related Information

The druid.io documentation

Apache Druid content roadmap

The content roadmap provides links to the available content resources for Apache Druid (incubating). The hyperlinks in the table and many others throughout this documentation jump to content published on the Druid site. Do not download any Druid code from this site for installation in an HDP cluster. Instead, install Druit by selecting Druid as a Service in an Ambari-asssisted HDP installation.

Table 1: Apache Hive Content roadmap

Type of Information	Resources	Description
Introductions	About Druid	Introduces the feature highlights of Druid, and explains in which environments the data store is best suited. The page also links to comparisons of Druid against other common data stores.
	Druid Concepts	This page is the portal to the druid.io technical documentation. While the body of this page describes some of the main technical concepts and components, the right-side navigation pane outlines and links to the topics in the druid.io documentation.
	Druid: A Real-time Analytical Data Store	This white paper describes the Druid architecture in detail, performance benchmarks, and an overview of Druid issues in a production environment. The extensive References section at the end of the document point to a wide range of information sources.
	Druid Architecture Paper	
Tutorial	Druid Quickstart	A getting started tutorial that walks you through a Druid package, installation, and loading and querying data. The installation of this tutorial is for instructional purposes only and not intended for use in a Hortonworks Hadoop cluster.
Integration with Hive	Druid and Hive Integration	Explains how to index data from Hive into Druid and query Druid datasources from Hive.
Developing on Druid	Developing on Druid	Provides an overview of major Druid components to help developers who want to code applications that use Druid-ingested data. The web page links to another about segments, which is an essential entity to understand when writing applications for Druid.
Data Ingestion	Batch Data Ingestion Loading Streams	These two pages introduce how Druid can ingest data from both static files and real-time streams.
Queries of Druid Data	Querying	Describes the method for constructing queries, supported query types, and query error messages.
Best Practices	Recommendations	A list of tips and FAQs.

Related Information

The druid.io documentation

Setting up and using Apache Druid

After learning hardware recommendations and software requirements, you add the Apache Druid (incubating) service to an HDP 3.x cluster.

Before you begin

Recommendations:

- Assign the Overload, Coordinator, and Router to one or more master nodes of size AWS m3.xlarge or equivalent: 4 vCPUs, 15 GB RAM, 80 GB SSD storage
- Co-locate the Historical and MiddleManager on different nodes from the Overload, Coordinator, Router, and Broker, and on nodes of size AWS r3.2Xlarge or equivalent: 8 vCPUs, 61 GB RAM, 160 GB SSD storage.
- Do not co-locate LLAP daemons and Historical components.

Software Requirements:

- A MySQL or Postgres database for storing metadata in a cluster for a production

  You can use the default Derby database installed and configured by Ambari if you are using a single-node cluster for development.
- Ambari 2.7.0 or later
- Database connector set up in Ambari
- HDP 3.0 or later
- ZooKeeper
- HDFS or Amazon S3
- YARN and MapReduce2

Related Information

The druid.io documentation

Set up a database

You choose and set up a database to use as the metastore for Apache Druid (incubating).

About this task

MySQL or Postgres databases that you install on the cluster are supported for production use. The MySQL database installed by Ambari does not meet the UTF-8 requirement for Druid, and is not suitable for production clusters. You can also use the default Derby database for development work. Database setup includes configuring the database connector in Ambari. If you use the default Derby database, you must assign all Druid components to the same node during installation.

Procedure

1. As root, install a database, MySQL for example, on any node in your cluster.

   On Centos 7.4, for example:

   ```
   yum install https://dev.mysql.com/get/mysql57-community-release-
   el7-8.noarch.rpm

   yum install mysql-community-server
   ```
2. Start the MySQL service.

   ```
   systemctl start mysqld.service
   ```
3. Obtain the temporary password and change the password for the root user. Do not use special characters in the password.

   Ambari is compatible with most, but not all, special characters.

   ```
   grep 'A temporary password is generated for root@localhost' \
   /var/log/mysqld.log |tail -1

   /usr/bin/mysql_secure_installation
   ```
4. On the database host, launch Mysql and create a user named druid with superuser permissions:

   ```
   CREATE USER 'druid'@'%' IDENTIFIED BY '<DRUIDPASSWORD>';
   GRANT ALL PRIVILEGES ON *.* TO 'druid'@'%’;
   GRANT ALL PRIVILEGES ON *.* TO 'druid'@'%' WITH GRANT OPTION;
   FLUSH PRIVILEGES;
   ```

5. Create a UTF8 database named druid.

```
CREATE DATABASE druid DEFAULT CHARACTER SET utf8;
```

Related Information

The druid.io documentation

Add Apache Druid to the cluster

You use Apache Ambari to add Apache Druid (incubating) to your cluster.

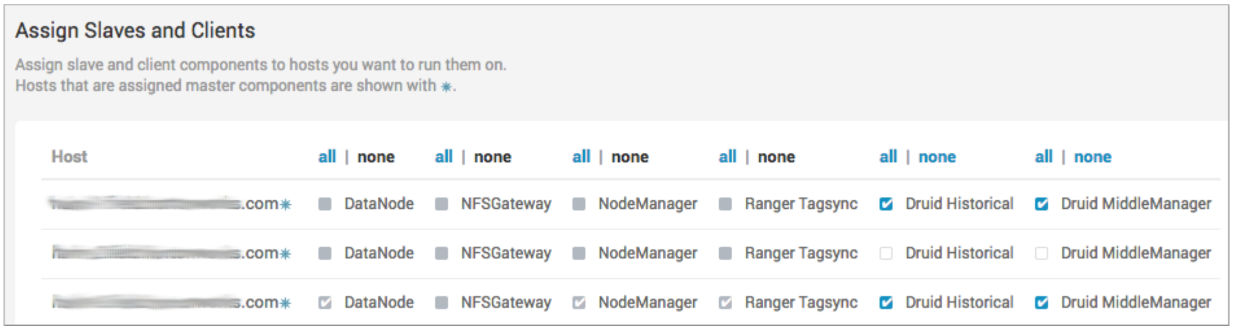
About this task

Procedure

1. On the Ambari host, download the database connector that matches your database.
   For MySQL, on Centos 7.4 for example:

   ```
   yum install mysql-connector-java*
   ```

   The database is installed in /usr/share/java.
2. On the Ambari server node, run the command to set up the connector, which includes the path to the downloaded JAR. For MySQL, on Centos 7.4 for example:
   ambari-server setup --jdbc-db=mysql --jdbc-driver=/usr/share/java/mysql-connector-java.jar
3. Start Ambari, and in Ambari select Services > Add Service.
4. Select Druid, and click Next.
5. In Assign Masters, typically you can accept the defaults to put the Broker, Coordinator, Overload, and Router components on the same master node, and click Next.
6. In Assign Slaves and Clients, putting Druid Historical and Druid MiddleManager on the same nodes and running these co-located components on additional nodes is recommended.



Assign Slaves and Clients

Assign slave and client components to hosts you want to run them on.
Hosts that are assigned master components are shown with *.

Host	all | none	all | none	all | none	all | none	all | none	all | none
[redacted].com*	■ DataNode	■ NFSGateway	■ NodeManager	■ Ranger Tagsync	☑ Druid Historical	☑ Druid MiddleManager
[redacted].com*	■ DataNode	■ NFSGateway	■ NodeManager	■ Ranger Tagsync	□ Druid Historical	□ Druid MiddleManager
[redacted].com*	☑ DataNode	■ NFSGateway	☑ NodeManager	☑ Ranger Tagsync	☑ Druid Historical	☑ Druid MiddleManager

Related Information

The druid.io documentation

Configure and deploy Apache Druid

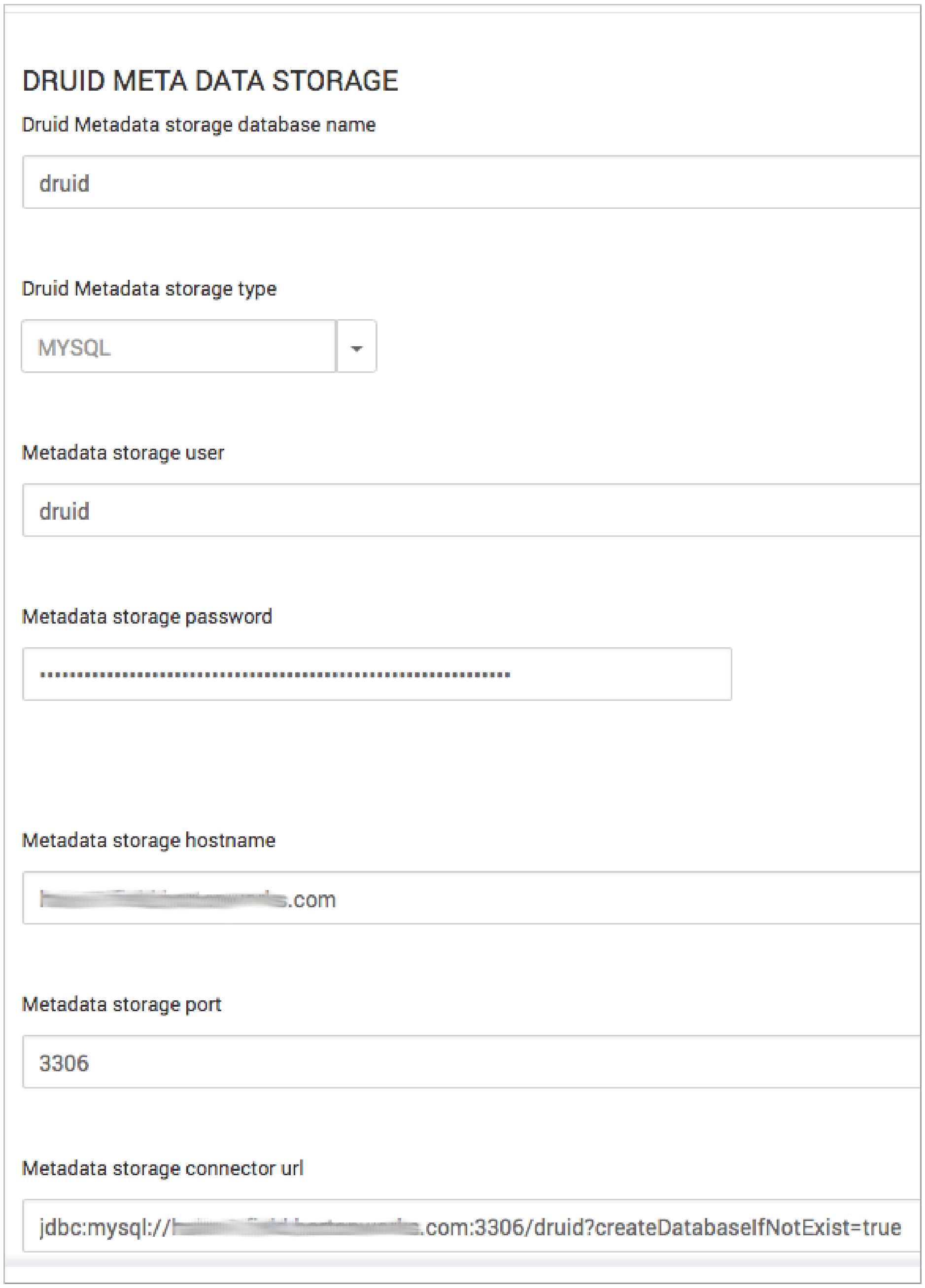
Using Ambari, you configure Apache Druid (incubating) by setting parameters in a graphical user interface.

About this task

Ambari populates many configuration settings based on the your previously entered settings and your environment. You might want to manually tune the configuration parameter settings. After you configure basic settings, you can access other settings on the Advanced tab.

Procedure

1. In Customize Services, change the Druid Metadata storage database name to druid if necessary.
2. Select a metadata storage type.
   - Choose a Druid Metadata storage type: MySQL or Postgres.
   - Accept the default Derby database if you have a single-node cluster or are working in a development environment.
3. Accept the Metadata storage user name druid, and enter the password for Metadata storage.
4. Configure a Metadata storage host name and storage port.
   - Enter the FQDN of the node that hosts the MySQL or Postgres database, and then enter a port number: 3306 for the MySql or 5432 for the Postgres port.
   - Accept the default configuration values if you use the default Derby database.



DRUID META DATA STORAGE

Druid Metadata storage database name

druid

Druid Metadata storage type

MYSQL

Metadata storage user

druid

Metadata storage password

••

Metadata storage hostname

[redacted].com

Metadata storage port

3306

Metadata storage connector url

jdbc:mysql://[redacted].com:3306/druid?createDatabaseIfNotExist=true

5. Address any prompts for additional configuration, click Next, and then deploy Druid.

6. Restart any components if prompted by Ambari.

   Success results unless you use an Ambari-installed MySQL database that uses a Latin character set. You must change the character set to UTF8: alter database druid character set utf8 collate utf8_general_ci;

Related Information

The druid.io documentation

Ingest data into Apache Druid

When Apache Druid (incubating) ingests data, Druid indexes the data. You can use one of several methods to ingest and index the data.

About this task

Druid has multiple ways to ingest data:

- Through Hadoop. Data for ingestion is on HDFS.
- From a file.
- From a Kafka stream. Data must be in a topic.

For more information, see about ingestion methods, see documentation on the druid.io web site (see link below).

In this task, you ingest data from a file. You create an index task specification in JSON and use HTTP to ingest the data.

Procedure

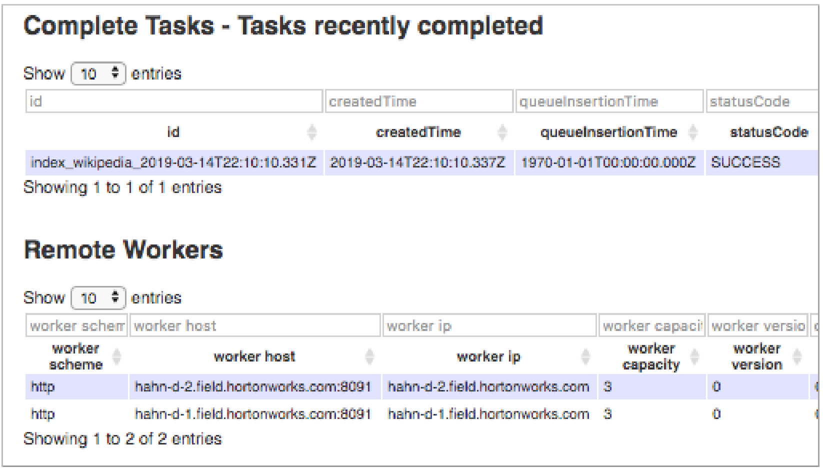
1. Create the following index task specification and name it index_local.json.

```
{
  "type" : "index",
  "spec" : {
    "dataSchema" : {
      "dataSource" : "wikipedia",
      "parser" : {
        "type" : "string",
        "parseSpec" : {
          "format" : "json",
          "dimensionsSpec" : {
            "dimensions" : [
              "channel",
              "cityName",
              "comment",
              "countryIsoCode",
              "countryName",
              "isAnonymous",
              "isMinor",
              "isNew",
              "isRobot",
              "isUnpatrolled",
              "metroCode",
              "namespace",
              "page",
              "regionIsoCode",
              "regionName",
              "user",
              { "name": "added", "type": "long" },
              { "name": "deleted", "type": "long" },
              { "name": "delta", "type": "long" }
            ]
          },
          "timestampSpec": {
            "column": "time",
            "format": "iso"
          }
        }
      },
      "metricsSpec" : [],
      "granularitySpec" : {
        "type" : "uniform",
        "segmentGranularity" : "day",
        "queryGranularity" : "none",
        "intervals" : ["2015-09-12/2015-09-13"],
        "rollup" : false
      }
    },
    "ioConfig" : {
      "type" : "index",
      "firehose" : {
        "type" : "local",
        "baseDir" : "/usr/hdp/current/druid-overlord/quickstart/",
        "filter" : "wikiticker-2015-09-12-sampled.json.gz"
      },
      "appendToExisting" : false
    },
    "tuningConfig" : {
      "type" : "index",
      "maxRowsPerSegment" : 5000000,
      "maxRowsInMemory" : 25000,
      "forceExtendableShardSpecs" : true
    }
  }
}
```

2. Ingest the data using the file you created.

```
curl -X 'POST' -H 'Content-Type:application/json' -d index_local.json
 http://localhost:8090/druid/indexer/v1/task
```

3. In Ambari, in Quick Links, click the Druid Overload Console.
   You see your job running, and then successfully complete.



Complete Tasks - Tasks recently completed

Show 10 entries

id	createdTime	queueInsertionTime	statusCode
index_wikipedia_2019-03-14T22:10:10.331Z	2019-03-14T22:10:10.337Z	1970-01-01T00:00:00.000Z	SUCCESS

Showing 1 to 1 of 1 entries

Remote Workers

Show 10 entries

worker scheme	worker host	worker ip	worker capacity	worker version
http	hahn-d-2.field.hortonworks.com:8091	hahn-d-2.field.hortonworks.com	3	0
http	hahn-d-1.field.hortonworks.com:8091	hahn-d-1.field.hortonworks.com	3	0

Showing 1 to 2 of 2 entries

Related Information

The druid.io documentation

Ingestion Overview and tutorials on the druid.io web site

Query Apache Druid

You use the Druid native query format in JSON to query Druid for the top 10 Wikipedia articles from the data Druid ingested earlier.

Procedure

1. Create the following file and save it as wickiticker-top.json.

```
{
  "queryType" : "topN",
  "dataSource" : "wikipedia",
  "intervals" : ["2015-09-12/2015-09-13"],
  "granularity" : "all",
  "dimension" : "page",
  "metric" : "count",
  "threshold" : 10,
  "aggregations" : [
     {
       "type" : "count",
       "name" : "count"
      }
   ]
}
```

2. Call HTTP to query Druid for the top 10 most-edited Wikipedia articles.

```
curl -X 'POST' -H 'Content-Type:application/json' -d @wikiticker-top.json
 http://localhost:8082/druid/v2?pretty
```

Related Information

The druid.io documentation

Configure Apache Druid for high availability

To make Apache Druid (incubating) highly available, you need to configure a sophisticated, multinode structure.

About this task

The architecture of a Druid cluster includes a number of nodes, two of which are designated Overlord and Coordinator. Within each Overlord and Coordinator domain, ZooKeeper determines which node is the Active node. The other nodes supporting each process are in Standby state until an Active node stops running and the Standby nodes receive the failover. Multiple Historical and Realtime nodes also serve to support a failover mechanism. But for Broker and Realtime processes, there are no designated Active and Standby nodes. Muliple instances of Druid Broker processes are required for HA. Recommendations: Use an external, virtual IP address or load balancer to direct user queries to multiple Druid Broker instances. A Druid Router can also serve as a mechanism to route queries to multiple broker nodes.

Before you begin

- You ensured that no local storage is used.
- You installed MySQL or Postgres as the metadata storage layer. You cannot use Derby because it does not support a multinode cluster with HA.
- You configured your MySQL or Postgres metadata storage for HA mode to avoid outages that impact cluster operations. See your database documentation for more information.
- You planned to dedicate at least three ZooKeeper nodes to HA mode.

Procedure

1. In Ambari, enable Namenode High Availability using the Ambari wizard. See Ambari documentation for more information.
2. Install the Druid Overlord, Coordinator, Broker, Realtime, and Historical processes on multiple nodes that are distributed among different hardware servers.
3. Ensure that the replication factor for each data source is greater than 1 in the Coordinator process rules. If you did not change data source rule configurations, no action is required because the default replication factor is 2.

Related Information

The druid.io documentation

Visualizing Druid data in Superset

You can add the Superset service to Ambari, define how to slice Druid data, create visualizations of the data, and build dashboards. You can visualize data in graphs, such as histograms, box plots, heatmaps, or line charts.

Before you begin

- You installed MySQL or Postgres for Druid metadata storage, or you intend to use SQLite.
- On the Ambari host, you downloaded the MySQL or Postgres driver and set up the driver:

  ```
  ambari-server setup --jdbc-db=mysql --jdbc-driver=<path to the driver>
  ```

  You do not have to set up a driver if you use SQLite.

Add Superset to the cluster

You can configure Superset to use the SQLite database in the Python library. Alternatively, you can configure Superset to use the existing MySQL or Postgres database that you installed. MySQL or Postgres are suitable for production work. SQLite is suitable for development work only.

Procedure

1. In Ambari, click Service > Add Service and select Superset. Accept the default node assignment, or select another node for the service. Click Next.
2. In Customize Services - Superset Metadata Storage, accept the default Superset Database Name, superset.
3. Select a database type.
   - SQLite and skip the next step. A superset user and database are already set up.
   - MySQL or Postgres
4. If you chose MySql or Postgres in the last step, open a terminal window, log into your database, create a superset user, and grant the superset user all privileges with the GRANT option. Create a superset database using the UTF8 character set.
   For example, in MySQL:

   ```
   CREATE USER 'superset'@'%' IDENTIFIED BY '<password>';
   GRANT ALL PRIVILEGES ON *.* TO 'superset'@'%';
   GRANT ALL PRIVILEGES ON *.* TO 'superset'@'%' WITH GRANT OPTION;
   FLUSH PRIVILEGES;

   CREATE DATABASE superset DEFAULT CHARACTER SET utf8;
   ```
5. In Customize Services - Superset Metadata Storage continue configuring the Superset service: In Superset Database Password, enter the password.

6. In Database Host Name:
   - SQLite: Accept the default value localhost.
   - MySQL or Postgres: Change the value to the FQDN of the node where you installed the database.
7. In Superset Database port:
   - SQLite: 8088
   - MySQL: 3306
   - Postgres: 5432
8. In SECRET_KEY, enter a random number, (recommended 20-digits long) preceded by at least one alphabetic character.
   For example: ab19336102290664791884
9. In Advanced, in Advanced superset-env, enter a Superset Admin password.

   Do not change the Superset Admin Password. This password is encrypted and embedded in Ambari.
10. Accept the Superset Admin user name admin and other defaults. Click Next, click Deploy, and Complete. Restart services as required, starting from the top of the Ambari Services list.

Related Information

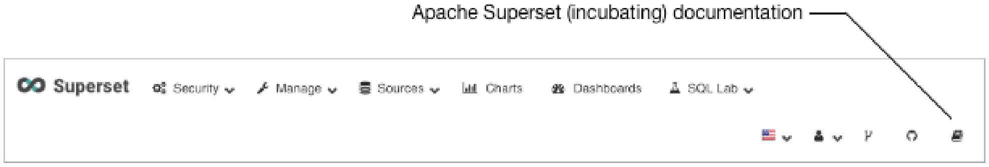
Apache Superset (incubating) Documentation about Druid

Visualize data using Superset

In the Superset UI, you connect to Druid data by filling out a dialog containing the fully qualified domain names (FQDN) of nodes that run Druid components. You specify a slice of data to visualize and query Druid. The visualization appears in the Superset UI.

About this task

This task introduces you to the Superset Web UI, which appears after you sign in. From the UI, you can navigate to the Apache documentation to obtain information not covered in this documentation, such as defining a list of users who can access views, opening Superset functions to certain groups of users, setting up permissions, and viewing user statistics. For more information about authentication in Superset, see the Flask documentation (link below).


Apache Superset (incubating) documentation

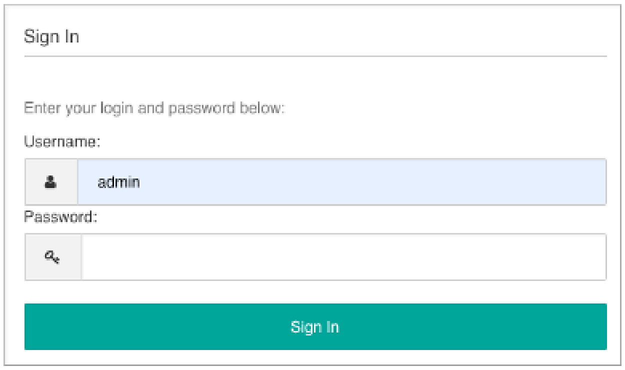
Before you begin

- You are running the Druid and Superset services in Ambari.
- You ingested data, such as the Wikipedia data from the Wikiticker example, into Druid.

  The data is records of edits to Wikipedia data.

Procedure

1. In Ambari, in Services > Superset > Summary > Quick Links, click Superset.
2. In Superset Sign In, enter the Superset Admin name admin and enter the Superset Admin password that you set up.



Sign In

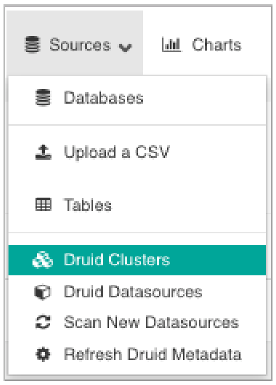
Enter your login and password below:

Username: admin

Password:

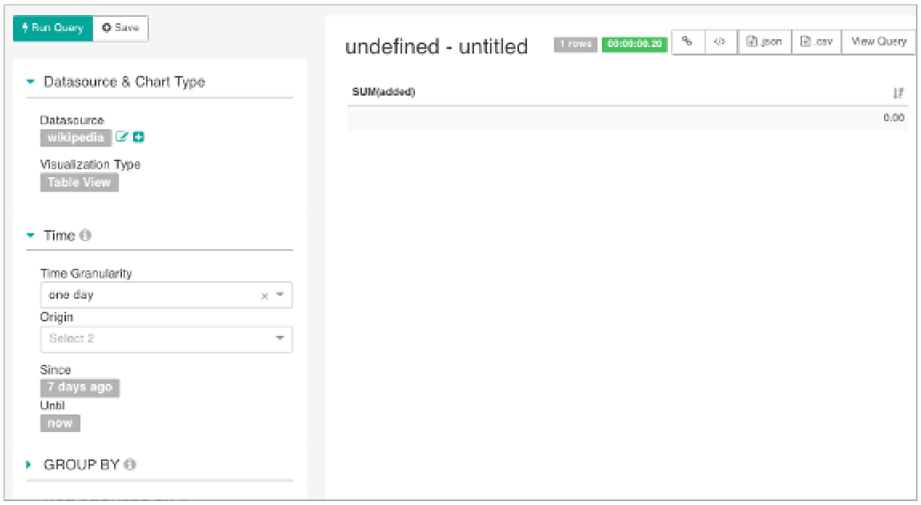
The Superset Web UI appears.

3. Select Sources > Druid Clusters.



4. Select Sources > Refresh Druid Metadata
   In List Druid Data Source, the wikipedia data source appears.

5. Click the data source wikipedia.
   The Data Source & Chart Type pane appears on the left. The canvas for query results appears on the right.



Datasource & Chart Type

Datasource: wikipedia

Visualization Type: Table View

Time

Time Granularity: one day

Origin: Select 2

Since: 7 days ago

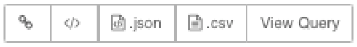
Until: now

GROUP BY

undefined - untitled

SUM(added)
0.00

At the top of the canvas, the UI includes controls for viewing the query in JSON and downloading the query in JSON or CSV format:



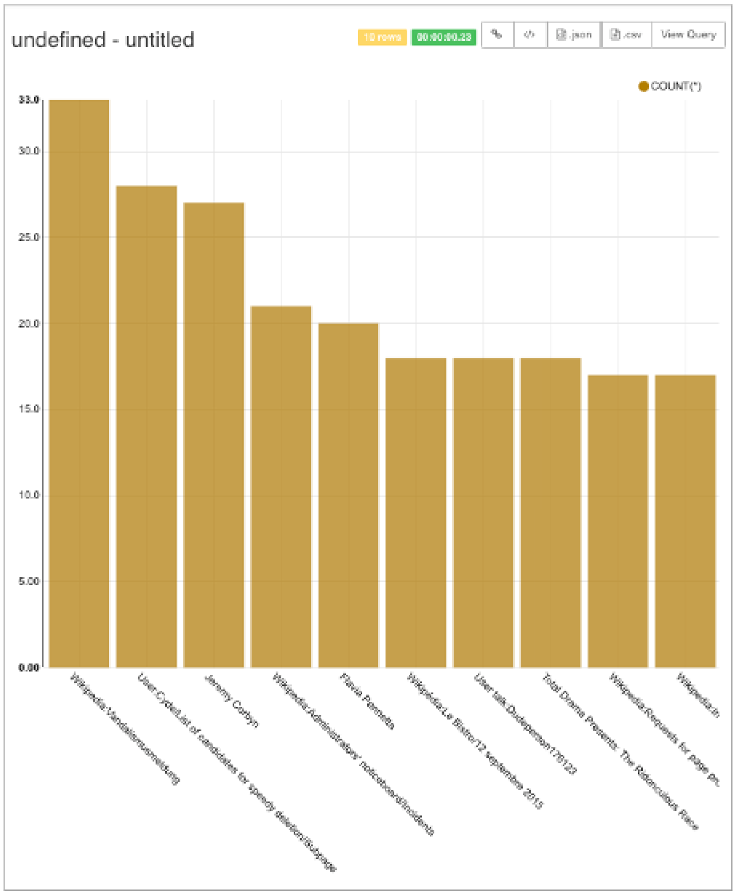
6. In Data Source & Chart Type, build a query that slices your Wikipedia data. For example, get the top 10 most-edited articles between September 12 and 13, 2015 by setting the following values.

Option	
Visualization Type	Distribution - Bar Chart

Option	
Time Granularity	All
Time - Since	9/12/2015
Time - Until	9/13/2015
Query - Metrics	COUNT(*)
Query - Series	page
Query - Row limit	10

In Since and Until, click Free form and enter a date in the format shown above.

7. Click Run Query.
A bar chart appears showing the top 10 articles for the time frame you specified.



8. On the canvas, change the default title of the visualization from undefined - untitled to Most Edits by Page Name, for example.

9. Click Save a Slice

,

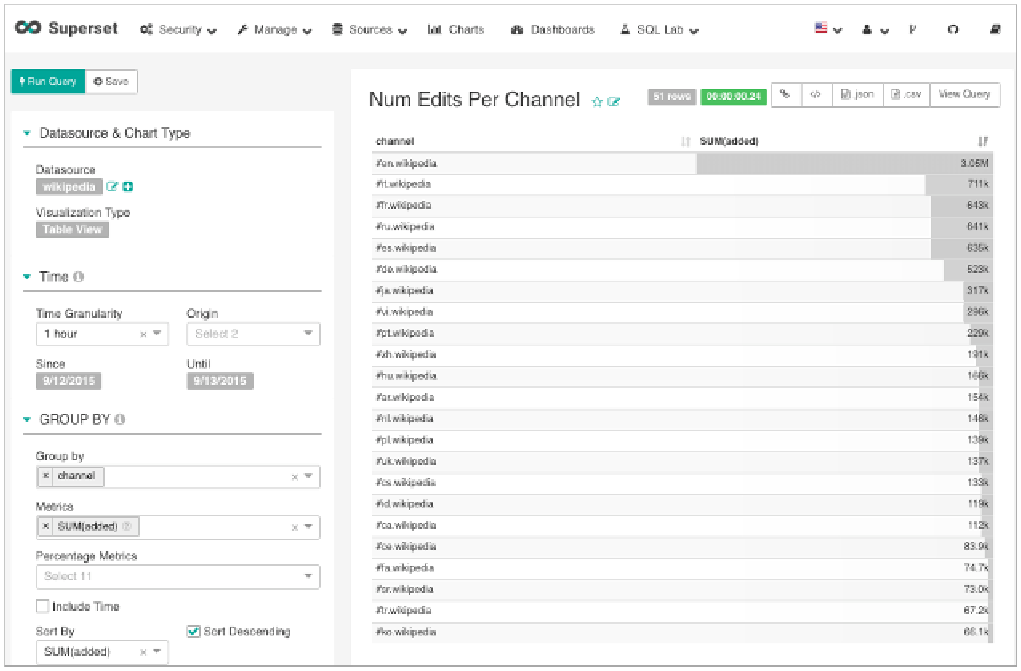
specify a file name, and click OK.

10. In Data Source & Chart Type, create a table view that aggregates edits per channel by changing the following values, run the query, and save the slice:

Option	
Visualization Type	Distribution - Bar Chart
Time Granularity	1 hour

Option	
Time - Since	9/12/2015
Time - Until	9/13/2015
Group by	channel
Metrics	SUM(added)
Sort By	SUM(added)

The resulting table shows the number of edits per channel:



▾ Datasource & Chart Type

Datasource: wikipedia

Visualization Type: Table View

▾ Time

Time Granularity: 1 hour

Origin: Select 2

Since: 9/12/2015

Until: 9/13/2015

▾ GROUP BY

Group by: channel

Metrics: SUM(added)

Percentage Metrics: Select 11

☐ Include Time

Sort By: SUM(added)

☑ Sort Descending

Num Edits Per Channel

51 rows 00:00:00.24

channel	SUM(added)
#en.wikipedia	3.05M
#it.wikipedia	711k
#fr.wikipedia	643k
#ru.wikipedia	641k
#es.wikipedia	635k
#de.wikipedia	523k
#ja.wikipedia	317k
#vi.wikipedia	296k
#pt.wikipedia	220k
#zh.wikipedia	191k
#hu.wikipedia	166k
#ar.wikipedia	154k
#nl.wikipedia	148k
#pl.wikipedia	139k
#uk.wikipedia	137k
#cs.wikipedia	133k
#id.wikipedia	119k
#ca.wikipedia	112k
#ce.wikipedia	83.9k
#fa.wikipedia	74.7k
#sr.wikipedia	73.0k
#tr.wikipedia	67.2k
#ko.wikipedia	66.1k

Related Information

Apache Superset (incubating) Documentation about Druid

Authentication: LDAP in Flask documentation

Building a dashboard

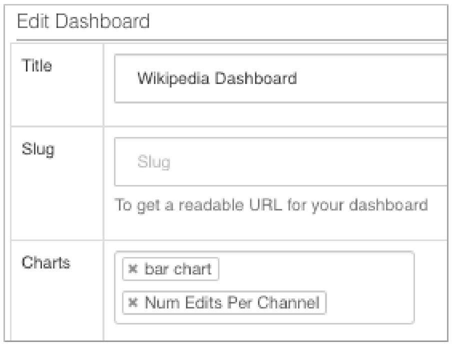
In Superset, you can create a Druid data dashboard using Superset visualizations that you save.

Before you begin

- You are running the Druid and Superset services in Ambari.
- You ingested data, such as the Wikipedia data from the Wikiticker example, into Druid.
- You created a Superset bar chart view of some of the data and a table view of some other data.

Procedure

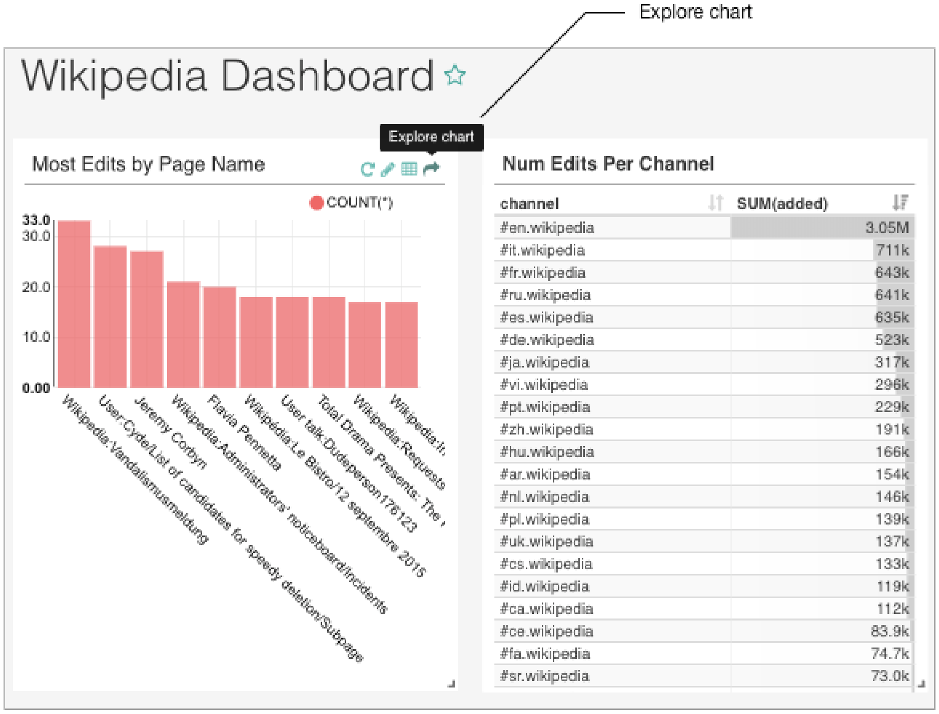
1. Click Dashboards > Add a New Record ➕ , enter a title: Wikipedia Dashboard, for example.
2. In Charts, browse to and select Superset results of queries you saved.
   For example, select the bar chart and the table view Num Edits Per Channel.



Edit Dashboard

Title	Wikipedia Dashboard
Slug	Slug To get a readable URL for your dashboard
Charts	× bar chart × Num Edits Per Channel

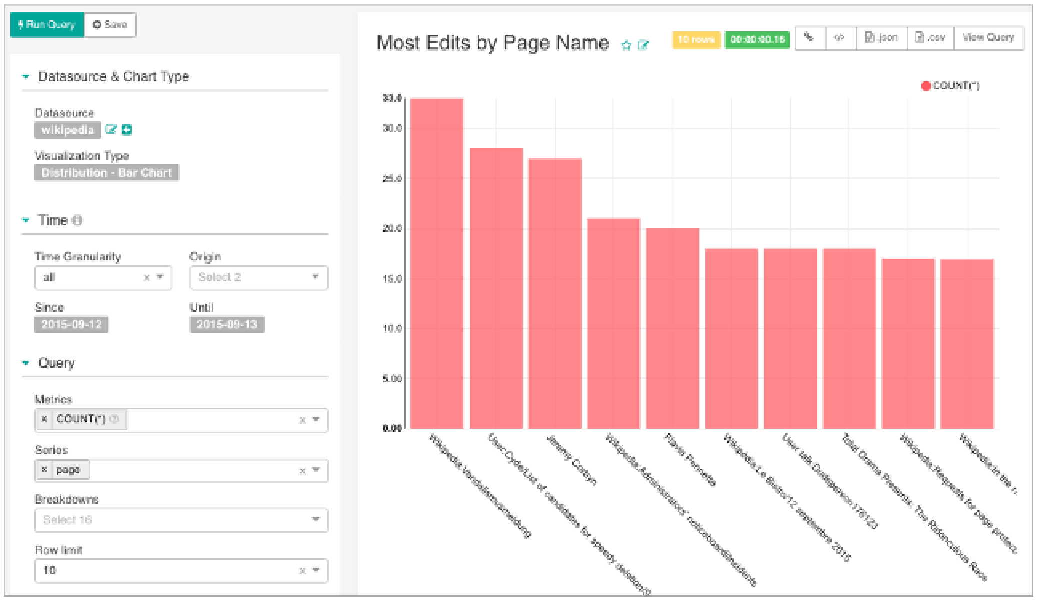
3. Click Save. In the Superset navigation bar, click Dashboards, and then select Wikipedia Dashboard. The Wikipedia Dashboard appears.
4. On the bar chart, click Explore the Chart.


Explore chart
Wikipedia Dashboard ☆
Explore chart
Most Edits by Page Name
COUNT(*)
33.0
30.0
20.0
10.0
0.00
Wikipedia:Vandalismusmeldung
User:Cyde/List of candidates for speedy deletion/Subpage
Jeremy Corbyn
Wikipedia:Administrators' noticeboard/Incidents
Flavia Pennetta
Wikipédia:Le Bistro/12 septembre 2015
User talk:Dudeperson176123
Total Drama Presents: The
Wikipedia:Requests
Wikipedia:I

Num Edits Per Channel

channel	SUM(added)
#en.wikipedia	3.05M
#it.wikipedia	711k
#fr.wikipedia	643k
#ru.wikipedia	641k
#es.wikipedia	635k
#de.wikipedia	523k
#ja.wikipedia	317k
#vi.wikipedia	296k
#pt.wikipedia	229k
#zh.wikipedia	191k
#hu.wikipedia	166k
#ar.wikipedia	154k
#nl.wikipedia	146k
#pl.wikipedia	139k
#uk.wikipedia	137k
#cs.wikipedia	133k
#id.wikipedia	119k
#ca.wikipedia	112k
#ce.wikipedia	83.9k
#fa.wikipedia	74.7k
#sr.wikipedia	73.0k

The Data Source & Chart Type dialog re-appears. You can view or edit the definition of the chart.



Run Query
Save
Datasource & Chart Type
Time
Query
Most Edits by Page Name
10 rows
00:00:00.16
.json
.csv
View Query
COUNT(*)
33.0
30.0
25.0
20.0
15.0
10.0
5.00
0.00
Wikipedia:Vandalismusmeldung
User:Cyde/List of candidates for speedy deletion/S
Jeremy Corbyn
Wikipedia:Administrators' noticeboard/Incidents
Flavia Pennetta
Wikipedia:Le Bistro/12 septembre 2015
User talk:Dudepenson176123
Total Drama Presents: The Ridonculous Race
Wikipedia:Requests for page protect
Wikipedia:In the n

Datasource: wikipedia

Visualization Type: Distribution - Bar Chart

Time Granularity: all

Origin: Select 2

Since: 2015-09-12

Until: 2015-09-13

Metrics: COUNT(*)

Series: page

Breakdowns: Select 16

Row limit: 10

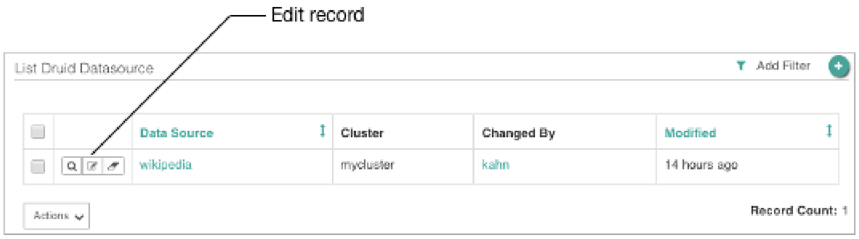
Related Information

Apache Superset (incubating) Documentation about Druid

Apache Superset aggregations

You can create new metrics or manage the visibility of metrics in Superset dashboards. Using JSON, you can operate on dimension values at query time.

You select Sources > Druid Data Sources > Edit record to view Druid columns.

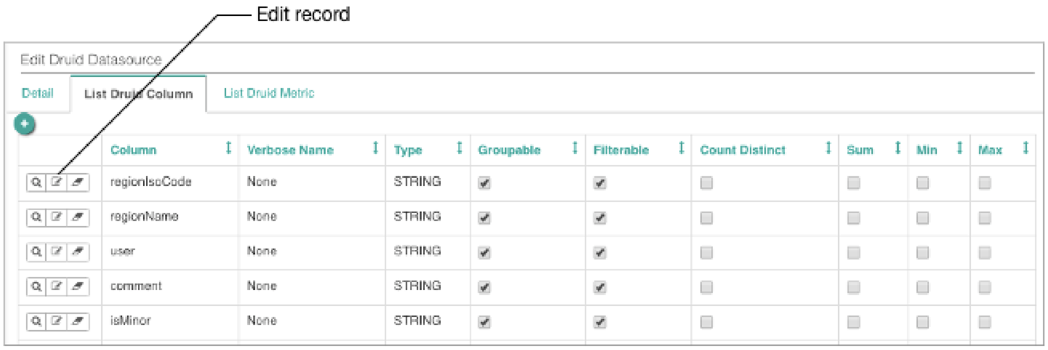


List Druid Datasource

□		Data Source	Cluster	Changed By	Modified
□		wikipedia	mycluster	kahn	14 hours ago

Record Count: 1

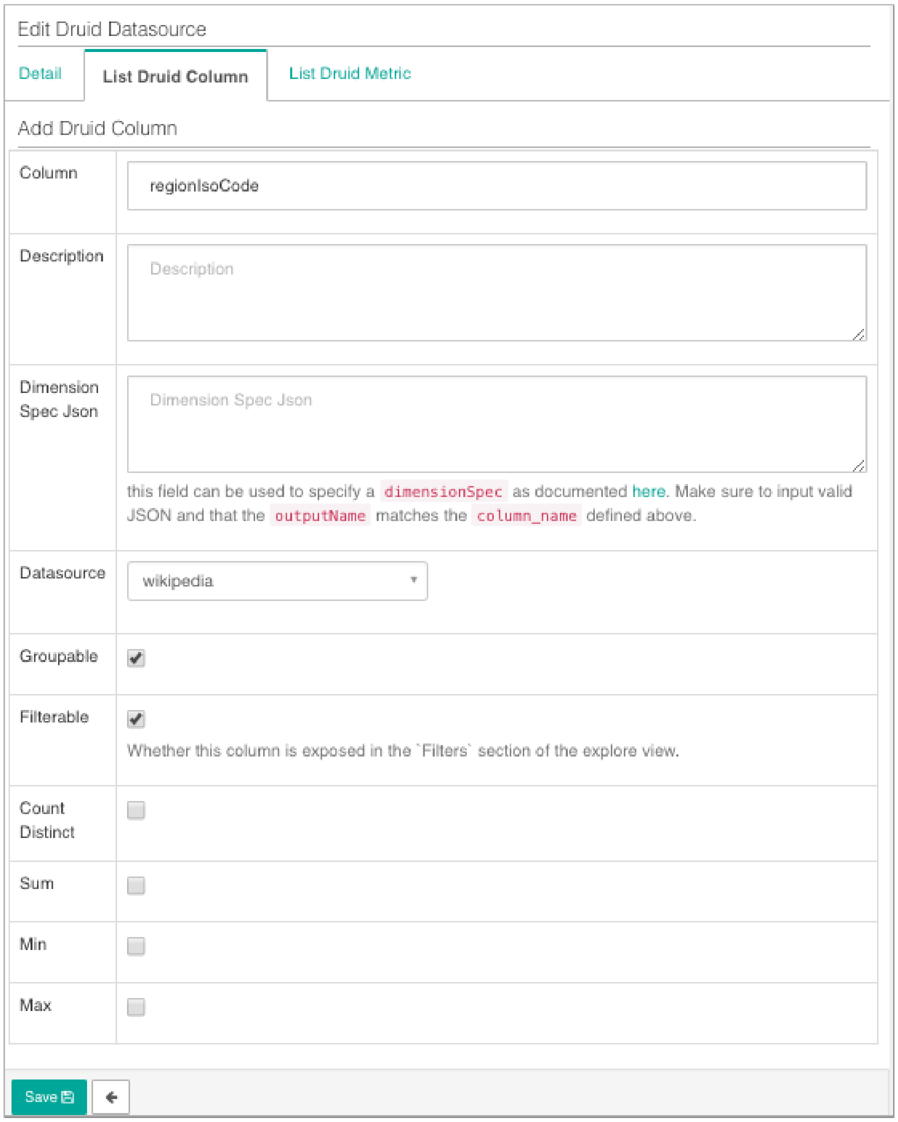
In List Druid Column, you can select Druid records for editing.



Edit Druid Datasource

	Column	Verbose Name	Type	Groupable	Filterable	Count Distinct	Sum	Min	Max
	regionIsoCode	None	STRING	☑	☑	□	□	□	□
	regionName	None	STRING	☑	☑	□	□	□	□
	user	None	STRING	☑	☑	□	□	□	□
	comment	None	STRING	☑	☑	□	□	□	□
	isMinor	None	STRING	☑	☑	□	□	□	□

In Edit Druid Data Source, the List Druid Column includes Groupable and Filterable controls.



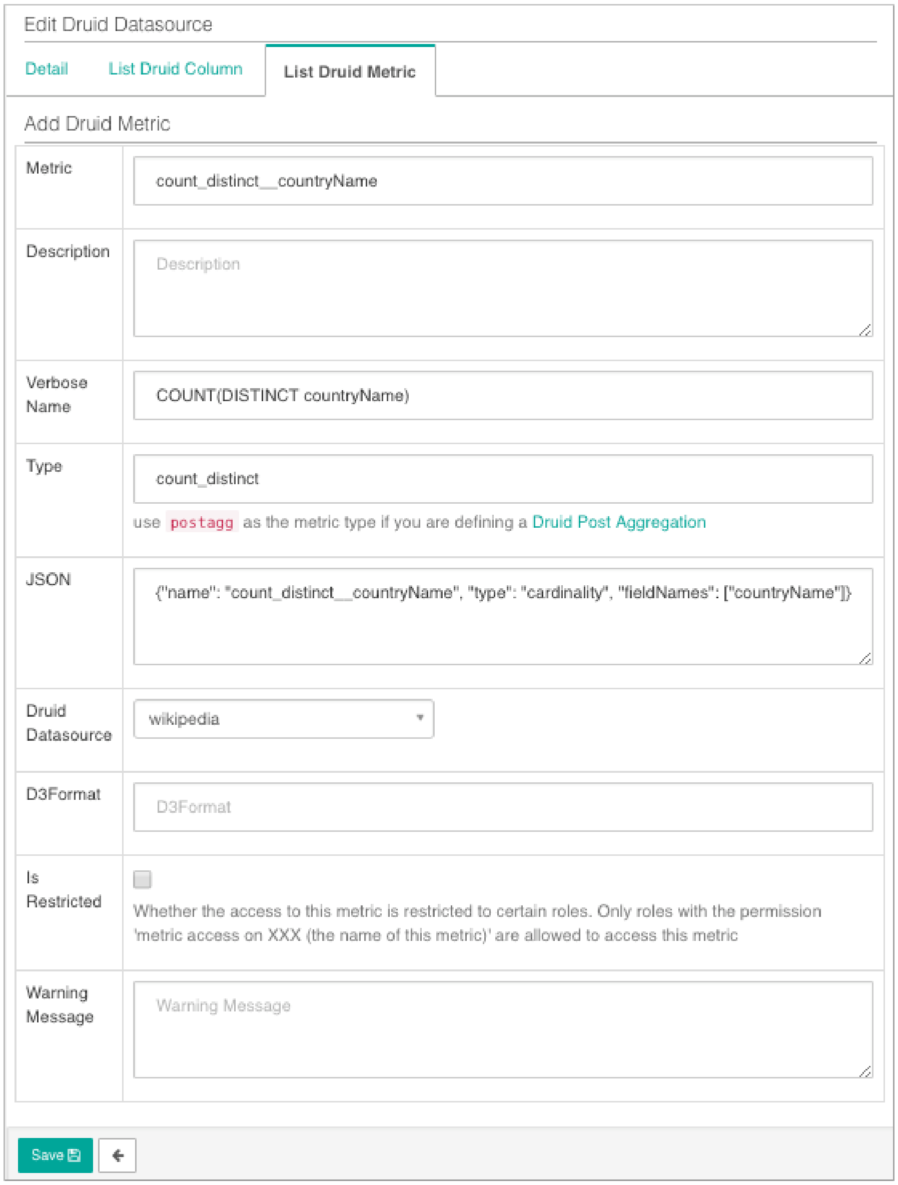
Edit Druid Datasource

Add Druid Column

Column	regionIsoCode
Description	Description
Dimension Spec Json	Dimension Spec Json this field can be used to specify a `dimensionSpec` as documented here. Make sure to input valid JSON and that the `outputName` matches the `column_name` defined above.
Datasource	wikipedia ▾
Groupable	☑
Filterable	☑ Whether this column is exposed in the `Filters` section of the explore view.
Count Distinct	☐
Sum	☐
Min	☐
Max	☐

Filterable columns appear in dropdowns in the Explore view of a dashboard. You create new metrics by checking Count Distinct, Min, Max, or Sum. For more information about transforming dimension values using a query, click the link to Druid.io documentation below or in the Superset UI.

In List Druid Metric, you can define a new metric or view the definition of an existing one.



Add Druid Metric

Metric	count_distinct__countryName
Description	Description
Verbose Name	COUNT(DISTINCT countryName)
Type	count_distinct use `postagg` as the metric type if you are defining a Druid Post Aggregation
JSON	{"name": "count_distinct__countryName", "type": "cardinality", "fieldNames": ["countryName"]}
Druid Datasource	wikipedia
D3Format	D3Format
Is Restricted	□ Whether the access to this metric is restricted to certain roles. Only roles with the permission 'metric access on XXX (the name of this metric)' are allowed to access this metric
Warning Message	Warning Message

A post-aggregation occurs when you specify postagg as the metric type and provide the post-aggregation definition in JSON. For more information about post aggregations, click the link to Druid.io documentation below or in the Superset UI.

Related Information

Transforming Dimension Values

Post-Aggregations

Securing Apache Druid using Kerberos

It is important to place the Apache Druid (incubating) endpoints behind a firewall and configure authentication.

You can configure Druid nodes to integrate a Kerberos-secured Hadoop cluster. Such an integration provides authentication between Druid and other HDP Services. If Kerberos is enabled on your cluster, to query Druid data sources that are imported from Hive, you must set up LLAP.

You can enable the authentication with Ambari 2.5.0 and later to secure the HTTP endpoints by including a SPNEGO-based Druid extension. After enabling authentication in this way, you can connect Druid Web UIs to the core Hadoop cluster while maintaining Kerberos protection. The main Web UIs to use with HDP are the Coordinator Console and the Overlord Console.

To use Hive and Druid together under Kerberos, you must run Hive Interactive Server to configure Hive in low-latency, analytical processing (LLAP) mode.

Related Information

Set up LLAP

Enable Kerberos authentication in Apache Druid

As Administrator, you can set up authentication of users who submit queries through Apache Druid (incubating) to the rest of the Hadoop cluster. If Kerberos is enabled on your cluster, to query Druid data sources that are imported from Hive, you must set up LLAP (low-latency, analytical processing).

Before you begin

- You enabled SPENGO-based Kerberos security on the cluster using the Ambari Server and Services.
- You planned for temporary down-time that is associated with this task.

  The entire HDP cluster must shut down after you configure the Kerberos settings and initialize the Kerberos wizard.
- You set up and enabled LLAP if you want to use Hive and Druid, and Hive Server Interactive is running.

About this task

To enable SPNEGO-based Kerberos authentication between the Druid HTTP endpoints and the rest of the Hadoop cluster, you run the Ambari Kerberos Wizard and manually connect to Druid HTTP endpoints in a command line. In this wizard, you configure the following Advanced Druid Identity Properties.

Property	Default Value Setting	Description
druid.hadoop.security.spnego.excludedPaths	['status'] To set more than one path, enter values in the following format:['/status','/condition']	Specify here HTTP paths that do not need to be secured with authentication. A possible use case for providing paths here are to test scripts outside of a production environment.
druid.hadoop.security.spnego.keytab	keytab_dir/spnego.service.keytab	The SPNEGO service keytab that is used for authentication.
druid.hadoop.security.spnego. principal	HTTP/_HOST@realm	The SPNEGO service principal that is used for authentication.
druid.security.extensions.loadlist	[druid-kerberos]	Indicates the Druid security extension to load for Kerberos.

Initializing the Kerberos Wizard might require a significant amount of time to complete, depending on the cluster size. Refer to the GUI messaging on the screen for progress status.

Procedure

1. In Ambari, launch the Kerberos wizard automated setup.
2. In Configure Identities, adjust advanced Druid configuration settings: Review the principal names, particularly the Ambari Principals on the General tab and either leave the default appended names or adjust them by removing the -cluster-name from the principal name string.
   If your cluster is named druid and your realm is EXAMPLE.COM, the Druid principal that is created is druid@EXAMPLE.COM

   These principal names, by default, append the name of the cluster to each of the Ambari principals.

3. Select the **Advanced** > **Druid drop-down**.
4. Determine for which Advanced Druid Identity properties, if any, you need to change the default settings.

   Generally, you do not need to change the default values.
5. Confirm your configuration, and, ptionally, download a CSV file of the principals and keytabs that Ambari can automatically create.
6. Click Next.
   Kerberos configuration settings are applied to various components, and keytabs and principals are generated. When the Kerberos process finishes, all Services are restarted and checked. After authenticating successfully to Druid, users can submit queries through the endpoints.

Related Information

Set up LLAP

Access Kerberos-protected HTTP endpoints

As a user, before accessing any Druid HTTP endpoints, you need to authenticate yourself using Kerberos and get a valid Kerberos ticket.

Procedure

1. Log in through the Key Distribution Center (KDC).
   kinit -k -t <keytab_file_path> <user@REALM.COM>
2. Verify that you received a valid Kerberos authentication ticket by running the klist command.
   The console prints a Credentials cache message if authentication was successful. An error message indicates that the credentials are not cached.
3. Access Druid with a curl command, including the SPNEGO protocol --negotiate argument.

   ```
   curl --negotiate -u:<anyUser> -b ~/<cookies.txt> -c ~/<cookies.txt> -X
    POST -H'Content-Type: application/json' http://_<endpoint>
   ```

   The negotiate argument is prefaced by two hyphens.
4. Submit a query to Druid:

   ```
   curl --negotiate -u:<anyUser> -b ~/<cookies.txt> -c ~/<cookies.txt> -X
    POST -H'Content-Type: application/json'  http://broker-host:port/druid/
   v2/?pretty -d @query.json
   ```