Hortonworks SmartSense

User Guide

(August 29, 2016)

docs.cloudera.com

Hortonworks SmartSense: User Guide

The Hortonworks Data Platform, powered by Apache Hadoop, is a massively scalable and 100% open source platform for storing, processing and analyzing large volumes of data. It is designed to deal with data from many sources and formats in a very quick, easy and cost-effective manner. The Hortonworks Data Platform consists of the essential set of Apache Hadoop projects including MapReduce, Hadoop Distributed File System (HDFS), HCatalog, Pig, Hive, HBase, ZooKeeper and Ambari. Hortonworks is the major contributor of code and patches to many of these projects. These projects have been integrated and tested as part of the Hortonworks Data Platform release process and installation and configuration tools have also been included.

Unlike other providers of platforms built using Apache Hadoop, Hortonworks contributes 100% of our code back to the Apache Software Foundation. The Hortonworks Data Platform is Apache-licensed and completely open source. We sell only expert technical support, training and partner-enablement services. All of our technology is, and will remain free and open source. Please visit the Hortonworks Data Platform page for more information on Hortonworks technology. For more information on Hortonworks services, please visit either the Support or Training page. Feel free to Contact Us directly to discuss your specific needs.

Table of Contents

1. Document Navigation

Hortonworks SmartSense gives all support subscription customers access to a unique service that analyzes HDP cluster diagnostic data, identifies potential issues, and recommends specific solutions and actions. These analytics proactively identify unseen issues and notify customers of potential problems before they occur.

The Hortonworks SmartSense Tool (HST) provides cluster diagnostic data collection capabilities, enabling customers to quickly gather configuration, metrics, and logs that they can use to analyze and troubleshoot SmartSense support cases.

Hortonworks SmartSense User Guide provides you with the latest information about using SmartSense 1.3.0. For SmartSense installation and upgrade instructions, see the Hortonworks SmartSense Installation. After installing SmartSense, refer to the *Hortonworks SmartSense User Guide* for information about using SmartSense in an Ambari or non-Ambari environment and performing additional configuration.

If you have already installed SmartSense and you are ready to use it, choose your scenario:

- Using SmartSense with Ambari [2]
- Using SmartSense in a Non-Ambari Environment [9]

2. Using SmartSense with Ambari

SmartSense is automatically included in Ambari 2.2.0 and later. If you have Ambari 2.0 or 2.1, you can easily install SmartSense. The integration between Ambari and SmartSense is facilitated by the Ambari stack and views extension mechanisms. These extensions enable you to add SmartSense as a native Ambari service, and they automatically deploy an Ambari view, enabling you to quickly capture data using the Ambari web UI.

2.1. Capturing Bundles

After you install the SmartSense service and view, data collection can begin. To trigger a single capture, follow these steps:

1. Access **SmartSense View** by clicking the ▦ icon and selecting **SmartSense View**.
2. Click **Capture for Analysis** or **Capture for Troubleshooting**.

   If you select **Capture for Troubleshooting**, you must enter your case number.
3. Click the **Capture** button.

   This triggers Ambari agents on each node to invoke the HST agent to capture specific data.

   After HST agents complete their captures and report data to the HST server, the completed bundle is available in the bundles list for download, or it is automatically uploaded to the SmartSense Gateway, if configured.

2.1.1. Automatically Capturing Bundles Using the SmartSense Gateway

When enabled, the gateway automatically uploads completed bundles to Hortonworks when a capture is completed. This includes SmartSense Analysis as well as support case troubleshooting bundles. You can also schedule SmartSense Analysis bundles for capture and automatic upload. You can easily set the capture schedule using the HST server UI or the SmartSense Ambari view.

2.1.2. Updating the Capture Schedule

The SmartSense view provides a way to easily create, update, pause, resume, and remove the schedules used for automated bundle capture and upload. When you deploy it, SmartSense creates a default capture schedule. To view this default capture schedule and update it, follow these steps:

1. Access **SmartSense View** by clicking ▦ and selecting **SmartSense View**.
2. Click the **Schedule** link in the top right corner to access the scheduler settings.
3. Remove, pause, or resume existing schedules.

You can also update the capture schedule by selecting a new scheduling period (weekly or monthly) or changing the day of the week and time of day that you want the capture to take place.

> **Note**
>
> Scheduler changes take up to 1 hour to take effect.

2.1.3. Creating a New Capture Schedule

If you have deleted the default capture schedule, you can create a new one:

1. Access **SmartSense View** by clicking ▦ and selecting **SmartSense View**.
2. Click the **Schedule** link in the top right corner to access the scheduler settings.
3. Select the scheduling period (weekly or monthly) and the day of the week and time of day that you want the capture to take place.
4. Click **Set Capture Schedule**.

   > **Note**
   >
   > Scheduler changes take up to one hour to take effect.

2.2. Viewing and Downloading Bundles by Using Ambari

Completed bundles can be manually downloaded and uploaded, either to Hortonworks Support for support case troubleshooting, or to the SmartSense environment for SmartSense Analysis. You can also automate and schedule this process by using the SmartSense Gateway. When using the SmartSense Gateway, all bundles are uploaded to Hortonworks. When support case troubleshooting bundles are received, they trigger a case notification. This case notification uses the case number provided during the capture initiation process.

To view and download bundles, follow these steps:

1. Access the **SmartSense View** by clicking ▦ and selecting **SmartSense View**.
2. Click the **Bundles** link in the top right corner.

   This page shows all bundles that have been captured and their status. If data is still being captured, the UI automatically updates itself with the capture progress until completed.
3. After the bundle is in a completed state, download it manually by clicking **Download** and selecting either **Download Encrypted Bundle** or **Download Unencrypted Bundle**.

   Alternatively, if a SmartSense Gateway is configured, the bundle is automatically uploaded to Hortonworks.

2.3. Configuring Anonymization Rules by Using Ambari

As data is captured, specific types of data are automatically anonymized. By default, IP addresses and the domain component of host names are anonymized. To customize these anonymization rules, follow these steps:

1. Navigate to the Ambari **Dashboard** and click the **SmartSense** service.
2. Click the **Config** tab.
3. Based on your environment, choose one of the following actions:
    - For Ambari 2.1, navigate to the **Data Capture** section.
    - For Ambari 2.0, expand **Advanced anonymization-rules**.
4. Add the new anonymization rule (or change the existing rule) by following the details provided in Configure Data Anonymization Rules.

2.4. Accessing the Activity Explorer

The Activity Explorer includes an embedded instance of Apache Zeppelin, which hosts prebuilt notebooks that visualize cluster utilization data related to user, queue, job duration, and job resource consumption. To access the Activity Explorer:

Note

The quick link to the Activity Explorer is available only in Ambari 2.4 and later. If you are using **Ambari version earlier than 2.4**, you must access the Activity Explorer using the following URL: *http://<activity_explorer_host>:9060/*.

1. Navigate to the Ambari **Dashboard** and click the **SmartSense** service.
2. In the **Summary** tab, click **Quick Links** > **Activity Explorer**.

    This launches the Activity Explorer in a new browser tab.
3. Log in with your Activity Explorer admin credentials.
4. From the **Notebook** dropdown in the top toolbar, select the name of the notebook that you want to view.

    The following preconfigured notebooks are available:
    - Chargeback Dashboard [5]
    - HDFS Dashboard [6]
    - MapReduce & Tez Dashboard [7]
    - YARN Dashboard [7]

Zeppelin organizes data in notebooks, where each notebook contains rows of paragraphs. Each paragraph visualizes the results of a single SQL statement using either a table, bar chart, pie chart, area chart, line chart, or scatter plot.

Once you opened a notebook, be aware of these three operations:

1. Since the notebooks represent a view of SmartSense utilization data at a specific point in time, they need to be refreshed. In order to refresh all of the data shown in all paragraph of a notebook, you need to:

   a. Hover over the row containing the notebook title, and a set of controls will appear.

   b. Click on the ▷ button to “Run all paragraphs”. The data for each paragraph in the notebook will be refreshed.

2. Top N paragraphs show the top 10 entries by default, but you can change this number by entering a new number in the **Top** input field and then typing enter.

3. Charts have interactive filters that let you select and deselect specific resources by clicking on the circle in the chart legend. For example, if there are four resources being displayed in a chart, and you only want to see four, you can click on a colored circle in the legend to filter it out:

   ● HIVE ● MR ● PIG ● TEMPLETON

   Once clicked, the inside of the circle will change to white, and the entry will not be displayed in the chart. For example, if you deselect "Hive", the legend will look like this:

   ○ HIVE ● MR ● PIG ● TEMPLETON

2.4.1. Chargeback Dashboard

The Chargeback Dashboard helps operators understand which resources are being consumed and what costs are associated with these resources. This dashboard exposes five types of resources:

- **CPU Hours** (in hours) - The amount of CPU used by MapReduce and Tez jobs
- **Memory Hours** (in gigabytes) - The amount of memory consumed by MapReduce and Tez jobs, and length of consumption
- **Storage** (in gigabytes) - The amount of HDFS space being consumed
- **Data IO** (in gigabytes) - The amount of data read and written to HDFS
- **Network IO** (in gigabytes) - The amount of data sent and received over the cluster’s network

Paragraph	Description
Chargeback Report	This paragraph lets you associate a financial cost with each of the five resources presented in the previous paragraph. Based on these per unit financial costs, you can see how much should be charged for each resource type.

Paragraph	Description
	The report also sums up the charge per resource to a per user total, so it's easy to see how much should be charged back to that specific user for their total resource consumption. The goal is to show how much money each user should be charged for the cluster resources that they have consumed.

2.4.2. HDFS Dashboard

The HDFS Dashboard helps operators better understand how HDFS is being used and which users and jobs are consuming the most resources within the file system.

This dashboard includes the following paragraphs:

- File Size Distribution
- Top N Users with Small Files
- Top N Largest HDFS Users
- Average File Size
- HDFS File Size Distribution Trend
- HDFS Utilization Trend
- HDFS File Size Distribution Trend by User
- HDFS File Size Distribution Trend by User
- Jobs With High Number of HDFS Operations
- HDP 2.5: Jobs Creating Many HDFS Files
- Jobs With Large Amount of Data Written

Most of these paragraphs have titles that are self-explanatory. A few of them are described below to provide more context:

Paragraph	Description
File Size Distribution	For any large multi-tenant cluster, it's important to identify and keep the proliferation of small files in check. The paragraph displays a pie chart showing the relative distribution of files by file size categorized by Tiny (0-10K), Mini (10K-1M), Medium (30M-128M), and Large (128M+) files. The goal is to show how dominant specific file size categories are within HDFS. If there are many small files, you can easily identify (in the next paragraph) who is contributing to those small files.
Top N Users with Small Files	Understanding how prevalent files of specific sizes are is helpful, but the next step is understanding who is responsible for creating those files. The goal of this paragraph is to show who is responsible for creating the majority of small files within HDFS.
Top N Largest HDFS Users	This paragraph helps you understand where all of the HDFS capacity is being consumed, and who is consuming it. The goal is to help you quickly understand which user or users are storing the most data in HDFS.
HDFS File Size Distribution Trend by User	Each "by User" paragraph allows you to see how an individual user's file sizes are trending.

Paragraph	Description
	This paragraph helps answer questions related to points in time where large or small files start becoming more or less prevalent for specific users, and can help measure the success of coaching users on Hadoop best practices.
HDP 2.5: Jobs Creating Many HDFS Files	When troubleshooting issues related to HDFS NameNode performance, it's helpful to understand which jobs are creating the most files, and potentially putting the largest amount of load on the NameNode. In HDP 2.5, new counters have been added to track how many files are created by each YARN application. This is helpful in troubleshooting erroneous jobs that are unintentionally creating hundreds of thousands, or even millions of files within HDFS.

2.4.3. MapReduce & Tez Dashboard

The MapReduce & Tez Dashboard was created to provide key information for workloads that use MapReduce or Tez for execution.

This dashboard includes the following paragraphs:

- Top N Longest Running Jobs
- Top N Resource Intensive Jobs
- Top N Resource Wasting Jobs
- Job Distribution By Type
- Top N Data IO Users
- CPU Usage By Queue
- Job Submission Trend By Day.Hour

Most of these paragraphs have titles that are self-explanatory. A few of them are described below to provide more context:

Paragraph	Description
Top N Resource Wasting Jobs	Resource wasting is calculated by calculating the difference between the memory asked for and the memory that was actually used. For example, if a job asks for 100 8GB containers but only uses 5GB per container, 3GB per container is considered wasted. This is calculated per job, and the top 10 are listed.
Job Submission Trend By Day.Hour	This paragraph shows the number of jobs submitted by day and hour with the notation being <day>.<hour>. For example: • Monday.1 - 1am on Monday • Monday.20 - 8pm on Monday The goal of this dashboard is to identify specific job submission hotspots during the week and day. You can use this information to identify the best time to schedule resource intensive jobs to execute.

2.4.4. YARN Dashboard

The YARN Dashboard provides key information for queue, application, container, and NodeManager host metrics.

This dashboard includes the following paragraphs:

- Application Runtime Duration by Queue
- Top N Applications by Number of Containers Requested
- Top N Applications by Number of Containers Failed
- Top N Hosts by Number of Containers Executed
- Top N Hosts by Number of Application Failures
- Top N Hosts by Localization Time
- Top N Hosts by Container Launch Delay

Most of these paragraphs have titles that are self-explanatory. One of them is described below to provide more context:

Paragraph	Description
Top N Applications by Number of Containers Failed	This paragraph shows the top jobs with the highest number of failed containers and the reason for each failure, so that you can quickly identify which containers failed and why.

3. Using SmartSense in a Non-Ambari Environment

Deploying Hortonworks SmartSense Tool (HST) on a cluster that is not managed by Apache Ambari requires manual installation and configuration.

3.1. Capturing Bundles in a Non-Ambari Environment

You have two options for data capture when HST is deployed outside of Ambari: using an HST agent CLI and using the Web UI.

3.1.1. HST Server Web UI Capture

To use this option, you must enable the HST web UI for capture: Enable Capture Through UI. The web UI capture method enables users to capture data by simply clicking the desired services to capture, entering their case number, and clicking **Capture**.

To access the HST server web UI, navigate to http(s)://*HST Server FQDN*:9000/. The default user name and password is:

- **Default Username**: admin
- **Default Password**: admin

3.1.2. HST Agent CLI Capture

HST agents collect data for the specific node on which they are installed. To capture data for all nodes in the cluster, which is the most common use case, you must run the `hst capture` command on all nodes. Typically this is done using pdsh or other parallel distributed shell utilities. **Running the `hst capture` command on all nodes in parallel is highly recommended**, because it allows bundles to be captured in the least amount of time. For the all agents to consolidate data in the same bundle, it is important that all agents initiate capture within three minutes after the first agent initiates.

To initiate capture of service data for a specific case number, use the following syntax:

```
# hst capture {service} {case number} {optional: level}
```

The HST agent can collect data for multiple services simultaneously. To obtain the list of supported services, run the following command:

```
# hst list-services

Supported services:
  AMS          : Collect data for Ambari metrics issue
  Ambari       : Collect data for Ambari issue
  Falcon       : Collect data for Falcon issue
  Ganglia      : Collect data for Ganglia issue
  HBase         : Collect data for HBase issue
  HCatalog      : Collect data for HCatalog issue
  HDFS          : Collect data for HDFS issue
  Hive          : Collect data for Hive issue
  Kafka         : Collect data for Kafka issue
  Knox          : Collect data for Knox issue
  MR            : Collect data for MapReduce issue
  Nagios        : Collect data for Nagios issue
  Oozie         : Collect data for Oozie issue
  Pig           : Collect data for Pig issue
  Ranger        : Collect data for Ranger issue
  Spark         : Collect data for Spark issue
  Sqoop         : Collect data for Sqoop issue
  Storm         : Collect data for Storm issue
  Tez           : Collect data for Tez issue
  YARN          : Collect data for YARN issue
  ZK            : Collect data for ZooKeeper issue
```

You can specify services individually, combine services using commas as delimiters, or specify all services by using the all keyword.

Support Case Troubleshooting Capture Example

For example, to capture data just for Hadoop Distributed File System (HDFS) and for case number 0001, run **hst** as follows:

```
# hst capture HDFS 0001
```

To capture data for HDFS, Apaceh Hive, and Apache Oozie for case number 0002, run hst as follows:

```
# hst capture HDFS,HIVE,OOZIE 0002
```

To capture L3 capture-level data for every service listed for case number 0003, run hst as follows:

```
# hst capture all 0003 L3
```

SmartSense Analysis Capture Example

To capture data for SmartSense Analysis, only configuration and metrics are required and 0 is used as the case number:

```
# hst capture all 0
```

3.2. Viewing and Downloading Bundles in a Non-Ambari Environment

After a bundle has been initiated, you can use the HST Server web UI to check bundle status and then download the bundle when it is complete.

3.2.1. HST Server Login

To access the HST Server web UI, navigate to

```
http(s)://HST Server FQDN:9000/
```

The default user name and password are both “admin”. :

3.2.2. Viewing and Downloading Bundles

After a bundle is captured, you can either download it or, if using the SmartSense Gateway, have it automatically uploaded. For more information about the gateway, see the Installing SmartSense Gateway.

If a gateway is not configured, you must manually upload the bundle to Hortonworks by using SFTP. The connectivity details for the SmartSense SFTP environment are available in this article: https://hortonworks.my.salesforce.com/articles/en_US/How_To/Uploading-SmartSense-Bundles (To view this article, you need a valid Hortonworks support account).

3.3. Configuring Anonymization Rules in a Non-Ambari Environment

1. Use SSH to access the HST server host.
2. Edit the `/etc/hst/conf/anonymization_rules.json` file to add or change existing anonymization rules by following details provided in Configure Data Anonymization Rules.

4. Uploading Support Bundles

You can use SmartSense Gateway to automatically upload bundles, or you can upload bundles manually.

For more information about uploading support bundles, see Bundle Transport.

5. Configuring SmartSense

This chapter guides you through common configuration tasks such as changing capture levels, configuring data anonymization rules, and changing server and agent configurations.

5.1. Configuring Data Anonymization Rules

Anonymization rules define regular expressions to anonymize sensitive data (like IP addresses, domain names, and so on). Each rule uses JSON format to define what to match and the value to replace.

Note

Anonymization rule formats vary between different SmartSense versions. Make sure that you consult the documentation that matches your SmartSense version.

1. To define regular expression-based rules, refer to the following sample:

```
  {
    "name":"ip_address",
    "path":null,
    "pattern": "[ :\\/]?[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}[ :
\\/]?",
    "extract": "[ :\\/]?([0-9\\.]+)[ :\\/]?",
    "shared": true
  }
```

   Key reference:

   - `name` - The rule name
   - `path` - An optional regular expression path of files on which to apply this rule (default `null` means all files)
   - `pattern` - A regular expression that defines the pattern to match within the file
   - `extract` - An optional regular expression to extract the data from the matched pattern

     Each of the extracts is marked as a regular expression group.

   - `shared` - A flag that indicates which key to use for anonymization (the `shared` or `private`) key is used for masking)

     If the shared key is used, the Hortonworks support team can unmask data if needed for diagnostic purposes: for example, host names and IP addresses for resolving issues on specific hosts or communication between hosts. Note that unmasked data is not stored in Hortonworks repositories; it is discarded as soon as the analysis finishes.

   - `value` - An optional constant value to replace

Note that the value chosen should **not** be matchable by the *pattern* specified earlier. For example, if the pattern is .“*dfs.datanode.*”, the value should not contain “dfs.datanode”. Also, note that if the value is specified, the `shared` flag is ignored.

2. To use property-based rules, use the following example:

```
{
   "name":"delete_oozie_jdbc_password",
   "path":"oozie-site.xml",
   "property": "oozie.service.JPAService.jdbc.password",
   "operation":"DELETE"
   "shared": false
  }
```

- `name` - The rule name
- `path` - A regular expression path of files on which to apply this rule
- `property` - The name of a specific property within the matching files
- `operation` - Either DELETE or REPLACE (the default)

  If DELETE is specified, the property is removed from the configuration file, and if REPLACE is specified, the property value is replaced by either a constant value or a masked value.
- `value` - An optional value for the REPLACE operation.

  If a value is not specified, a private or shared key is used to mask the data to replace.
- `enabled` - A flag used to enable or disable rule definition, the default being **true**.
- `excludes` - A set of path patterns to be excluded by the rule: for example, **“excludes”: [“oozie-site.xml”, “core-site.xml”]**
- `shared` - Flag to allow anonymized data to be reversed by Hortonworks. If shared is true, anonymized data is reversible by Hortonworks, if false, that data cannot be reversed.

Note

Rules configured with `shared = false` cannot be unmasked by Hortonworks (and in some cases might become a roadblock for support case analysis.)

5.2. Change Server and Agent Configurations in a Non-Ambari Environment

All HST configurations are stored in `/etc/hst/conf`. Both `hst-server.ini` and `hst-agent.ini` have server and agent configurations. Changes performed on the HST server host are automatically propagated to all of the agents. Note that any change to the `hst-server.ini` file requires that you restart the HST server.

5.3. SmartSense Performance Tuning

To achieve optimal performance for your cluster size, you may need to increase the JVM memory settings.

The default setting is:

```
-Xms512m -Xmx2048m
```

This setting is appropriate for a cluster with up to 100 nodes. For each additional 100 nodes, increase this setting by 0.5 GB to improve performance. To adjust the setting, edit the `/etc/sbin/hst-server.py` file:

```
Line: 1408
 def get_server_start_cmd():
  return "{0} -server -XX:NewRatio=3 "\
          "-XX:+UseConcMarkSweepGC " +\
          "-XX:-UseGCOverheadLimit -XX:CMSInitiatingOccupancyFraction=60 " +\
          debug_options +\
          " -Dlog.file.name="+ SERVER_LOG_FILENAME +" -Xms512m -Xmx2048m -cp
 {1}" + os.pathsep + "{2}" +\
          " com.hortonworks.support.tools.server.SupportToolServer "\
          ">" + get_out_file() + " 2>&1 &"
```