Hortonworks Data Platform

SmartSense Administration Guide

(November 1, 2015)

docs.cloudera.com

Hortonworks Data Platform: SmartSense Administration 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. SmartSense Administration Guide

This document provides you with the latest information about the SmartSense 1.1.0 release and its product documentation.

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, allowing customers to quickly gather configuration, metrics, and logs that are used for both SmartSense analysis, and troubleshooting Support Cases.

1.1. Document Navigation

To help you better navigate this document, please read the Meet Minimum System Requirements, SmartSense Architecture Overview, and Downloading SmartSense Binaries sections, then select your scenario from the list below:

- What's Included In A Bundle [ ]
- Meet Minimum System Requirements [ ]
- Using SmartSense With Ambari [ ]
- Using SmartSense in a Non-Ambari Environment [ ]
- What's New In This Release [ ]
- Known Issues [ ]
- Uploading Bundles to Hortonworks [ ]

1.2. SmartSense Installation

This section describes the information and materials required to install HST on a Hortonworks Data Platform (HDP) cluster.

1.2.1. Meet Minimum System Requirements

To run HST, your system must meet requirements in the following areas:

- Operating Systems Requirements [ ]
- Software Requirements [ ]
- JDK Requirements [ ]
- Browser Requirements [ ]
- Ambari Requirements [ ]

1.2.1.1. Operating Systems Requirements

The following, 64-bit operating systems are supported:

- CentOS v5.x (deprecated)
- CentOS v6.x
- CentOS v7.x (recommended to have net-tools installed)
- Debian 6
- Red Hat Enterprise Linux (RHEL) v5.x (deprecated)
- Red Hat Enterprise Linux (RHEL) v6.x
- Red Hat Enterprise Linux (RHEL) v7.x (recommended to have net-tools installed)
- SUSE Linux Enterprise Server (SLES) v11 SP3
- Ubuntu 12.04

1.2.1.2. Software Requirements

The following packages need to be installed on each of the hosts in your cluster. These packages are used to gain a more complete diagnostic profile of the cluster.

- wget
- sysstat
- dstat
- lsof
- net-tools
- Python 2.6+

1.2.1.3. JDK Requirements

The following Java runtime environments are supported:

- OpenJDK 7 64-bit
- OpenJDK 8 64-bit
- Oracle JDK 1.6 (deprecated)
- Oracle JDK 1.7 64-bit
- Oracle JDK 1.8 64-bit

1.2.1.4. Browser Requirements

The HST Server runs a browser-based Web application. You must have a machine capable of running a graphical browser to use this tool.

The minimum required browser versions per Operating System are:

- Windows (Vista, 7)
  - Firefox 39
  - Google Chrome 43
  - Internet Explorer 11.0
- Mac OS X (10.6 or later)
  - Firefox 39
  - Google Chrome 43
  - Safari 8

On any platform, we recommend updating your browser to the latest, stable version.

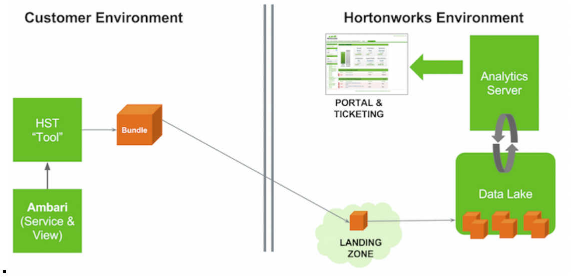
1.2.1.5. Ambari Requirements

SmartSense can be integrated with and deployed via Apache Ambari. The Ambari integration is certified with the following versions:

- Apache Ambari 1.7
- Apache Ambari 2.0
- Apache Ambari 2.1

1.2.2. SmartSense Architecture Overview

The Hortonworks SmartSense Tool (HST) is used to collect cluster diagnostic data, both for assisting with support case troubleshooting, and for SmartSense analysis. HST uses a central server daemon and a distributed set of passive HST agents. The HST Agent processes are not long-lived services, and are only started when specific data capture tasks are required. Once the HST Agent has captured the requested data, the process stops. All data captured by HST Agents are sent to the central HST server to coalesce into a single downloadable bundle file. These bundles can then be uploaded to Support for troubleshooting, or to the SmartSense SFTP server for SmartSense analysis.



In order to provide the most complete picture of cluster utilization, HST agents must be installed on every node in the cluster. HST can be installed as a standalone component, and manually installed on all nodes of the cluster, or can be integrated with Apache Ambari for automatic installation and service management.

1.2.3. What's Included In A Bundle

SmartSense collects following types of data:

- Operating System
  - Configuration (Partition layouts, file system mount options, key service status, network configurations etc.)
  - Metrics (CPU, memory, IO statistics, network statistics, etc.)
  - Logs (System messages, driver messages)
- HDP Service
  - Configuration
  - Metrics (JMX reports, installed packages)
  - Logs (Only for support case troubleshooting and not for SmartSense Analysis)

To see specific data and files that are captured, perform a capture in your environment, then download the unencrypted bundle to see what data is collected.

Bundle Security

We take security seriously. There are multiple levels of provisions made to make sure that any sensitive data is protected to the extent possible.

- Anonymization and Exclusions
  - All IP addresses and host names are ALWAYS anonymized
  - Passwords such as hive, oozie, sqoop, ambari password are excluded from collection
- Encryption
  - SmartSense Analysis bundles are optionally encrypted using AES + RSA encryption
- Further customizations
  - Custom anonymization rules can be configured to include environment-specific patterns
  - Custom configuration can be added to additionally exclude files, and Hadoop properties from collection.

Bundles sent to the Hortonworks SmartSense Analysis environment are stored in their original anonymized and encrypted form for 90 days before being removed. Specific metadata such as Ambari and HDP stack version, node count, and amount of storage available/used is stored for trending rules analysis. Recommendations generated for each bundle are made available through the Hortonworks Support Portal and are stored for feedback purposes and used to improve future recommendations.

1.2.4. Downloading SmartSense Binaries

SmartSense HST is available for download from the “Tools” tab of Hortonworks support portal (https://support.hortonworks.com).

Note

During Installation you will be asked for your SmartSense ID and Account Name. These fields are used to associate the cluster diagnostic data to your account in Hortonworks support portal. This information is available in support portal under both the “Tools” and “SmartSense” tabs.

1.2.5. Using SmartSense With Ambari

SmartSense can be easily "plugged in" to Ambari 1.7, 2.0.x, and 2.1.x making the installation and management more streamlined and familiar to administrators. The integration between Ambari and SmartSense is facilitated by the Ambari stack and views extension mechanisms. These extensions allow SmartSense to be added as a native Ambari Service, and automatically deploy an Ambari View for users to quickly capture data using the Ambari Web UI.

1.2.5.1. Installing SmartSense with Ambari

SmartSense is optionally integrated with Ambari through the Ambari Service, and Ambari View extension framework. Both the SmartSense Service, and the SmartSense View are included in the downloaded binary, and must be installed as follows:

Note

You should *not* proceed with the installation unless you have your SmartSense ID and Account Name. These are required to complete the installation. Also, please note that your SmartSense ID must match this format: A-00000000-C-00000000. Invalid SmartSense ID’s will cause the installation to fail.

1. Install SmartSense Package on the Ambari Server host

    - RHEL / CentOS:

      ```
      # rpm -ivh smartsense-hst-1.1.0-67.x86_64.rpm
      ```

    - Ubuntu / Debian:

      ```
      # dpkg -i smartsense-hst_1.1.0-67.deb
      ```

2. Add the SmartSense Ambari integrations to the existing Ambari Installation:

    ```
    # hst add-to-ambari
    Enter SmartSense distributable path: Full path to package from Step 1
    It is required to restart Ambari server for changes to reflect. Do you want
     to restart Ambari server? [y/n] (default: n): y
    Restarting Ambari server.
    ```

3. From Ambari Web UI, select **Add Service** from the **Actions** drop-down menu.

4. From the list of installable services, select **SmartSense**.

5. In the **Assign Masters** step, select a cluster node for the HST Server.

   For a list of criteria to determine the best node to select see the HST Server Placement section.

6. In the **Customize Services** step, there are three required fields, and three fields that should be reviewed before proceeding:

Note

For Ambari 1.7, make sure that HST Agent is selected on *all cluster hosts* during **Select Slaves and Customize Services** step.

Ambari 2.1	Ambari 1.7, 2.0	Note
Configuration Tab: Basic **Property**: Customer account name	**Expand Configuration Section**: Advanced hst-server-conf **Property**: customer.account.name	Your account name, available from the “Tools” tab in Hortonworks support portal.
Configuration Tab: Basic **Property**: SmartSense ID	**Expand Configuration Section**: Advanced hst-server-conf **Property**: customer.smartsense.id	Your SmartSense ID, available from the “Tools” tab in Hortonworks support portal.
Configuration Tab: Basic **Property**: Notification Email	**Expand Configuration Section**: Advanced hst-server-conf **Property**: customer.notification.email	The email we will used to notify you when SmartSense bundles have been received and recommendations are ready for your review.
Configuration Tab: Basic **Property**: Bundle Storage Directory	**Expand Configuration Section**: Advanced hst-server-conf **Property**: server.storage.dir	The directory on the HST Server that will be used to store completed bundles. As bundles can be large, this directory should have at least 10GB of free space..
Configuration Tab: Basic **Property**: Server Temporary Data Directory	**Expand Configuration Section**: Advanced hst-server-conf **Property**: server.tmp.dir	The directory on the HST server used to assemble results from HST Agents into completed bundles. This directory must be large enough to handle the intermediate results of HST agent collection data. This directory should have at least 5GB of free space.
Configuration Tab: Basic **Property**: Minimum size required for storage directory (in GB)	**Expand Configuration Section**: Advanced hst-server-conf **Property**: server.min_required_storage	This property determines the minimum amount of storage space required when setting up the HST server. If your Bundle Storage Directory has less than this specified requirement, the HST setup process will fail. If the 10GB recommendation for Bundle Storage Directory cannot be adhered to, this must be set to a value below the current amount of free space on the Bundle Storage Directory you plan on using to prevent HST setup failures.

7. Once the fields above have been validated, click **Next**.

8. In the **Review** step, click **Deploy** to complete your SmartSense Service installation.

9. After the SmartSense service has been successfully added to Ambari, the Ambari server **must** be restarted to load the SmartSense view. Restart the Ambari Server by executing the following command:

```
# ambari-server restart
```

1.2.5.2. Capturing Bundles with Ambari

Once the SmartSense service and view have been installed, data collection can begin. To capture bundles both for support case troubleshooting and SmartSense analysis, the SmartSense View will be used. To access the **SmartSense View**, click on the ▦ icon and choose **SmartSense View**. The **SmartSense View** will display capture options based on intent: **SmartSense Analysis**, **Support Case Troubleshooting**. In either case, clicking the **Capture** button will trigger Ambari Agents on each node to invoke the HST Agent to capture specific data. Once HST Agents complete their capture and report data back to the HST Server, the completed bundle will be available in the bundles list for download.

1.2.5.3. Viewing and Downloading Bundles with Ambari

Login to ambari. To access the **SmartSense View**, click on the ▦ icon and choose **SmartSense View**. In the view , the list of bundles is available from the **Bundles** link. This page shows all bundles that have been collected and their status. If data is still being collected, the UI will automatically update itself with the progress of the capture process until completed. Once the bundle is in a completed state, it is ready for download.

Once downloaded, the bundle can be uploaded to Support for Support Case Troubleshooting, or to the SmartSense Environment for SmartSense Analysis by following the steps outlined in the following sections:

- Uploading SmartSense Analysis Bundles to Hortonworks SmartSense [ ]
- Uploading Bundles to Hortonworks [ ]

1.2.5.4. Configure Anonymization Rules with Ambari

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

1. Navigate to the Ambari **Dashboard**, and click on the **SmartSense** service.

2. Click on the **Config** tab.

3. For Ambari 2.1, navigate to the **Data Capture** section.

   a. For Ambari 1.7 and 2.0, expand **Advanced anonymization-rules**.

4. Add new anonymization rule (or change existing) by following details provided in Configure Data Anonymization Rules.

1.2.5.5. Uninstall SmartSense from Ambari

To remove the SmartSense service and view, follow the steps below.

1. First make sure all of the SmartSense services are stopped from the Ambari UI. (Optionally you can run below command). *Please ensure that all the SmartSense services are completely stopped before continuing. If they are not stopped, the uninstallation process will not be successful, and the service will not be able to be removed.*

   ```
   curl -u admin:$PASSWORD -i -H 'X-Requested-By: ambari' -
   X PUT -d '{"RequestInfo": {"context" :"Stop SmartSense via
   REST"}, "Body": {"ServiceInfo": {"state": "INSTALLED"}}}'
   http://AMBARI_SERVER_HOST:8080/api/v1/clusters/CLUSTER_NAME/
   services/SMARTSENSE
   ```

2. Uninstall all SmartSense components:

   ```
   curl -u admin:$PASSWORD -i -H 'X-Requested-By: ambari'
   -X POST -d '{"RequestInfo": {"context" :"Uninstall
   SmartSense via REST", "command":"Uninstall"}, "Requests/
   resource_filters":[{"hosts":"comma separated host names",
   "service_name":"SMARTSENSE", "component_name":"HST_AGENT"}]}'
   http://AMBARI_SERVER_HOST:8080/api/v1/clusters/CLUSTER_NAME/
   requests
   ```

   Note

   After issuing this command wait for Ambari operations to successfully complete. You can check this through Ambari UI

   . Make sure that there are no operations in progress.

3. Run the following command to remove SmartSense service from Ambari:

   ```
   curl -u admin:$PASSWORD -H 'X-Requested-By: ambari' -X DELETE
   http://AMBARI_SERVER_HOST:8080/api/v1/clusters/CLUSTER_NAME/
   services/SMARTSENSE
   ```

4. Restart ambari server for all changes to take effect:

   ```
   # ambari-server restart
   ```

1.2.5.6. Upgrading from HST 1.0.x to SmartSense 1.1.0 with Ambari

To upgrade from HST 1.0.x to SmartSense 1.1.0, please do the following:

1. On **every** node in the cluster, execute the following four commands to remove the HST 1.0.x software and all data and configuration associated with it:

   ```
   # rpm -e hst
   # rm -rf /usr/hdp/share/hst
   # rm -rf /etc/hst
   # rm -rf /var/log/hst
   ```

   Once, the previous version of HST has been removed, follow the installation guidelines for Installing SmartSense with Ambari.

1.2.6. Using SmartSense in a Non-Ambari Environment

Deploying HST on a cluster that is not managed by Ambari requires manual installation and configuration. The following sections will outline the specific steps required to successfully deploy HST in this type of environment.

1.2.6.1. Installing SmartSense

Without Ambari, SmartSense HST must be manually installed on every node in the cluster.

Note

You should *not* proceed with the installation unless you have your SmartSense ID and Account Name. These are required to complete the installation. Also, please note that your SmartSense ID must match this format: A-00000000-C-00000000. Invalid SmartSense ID’s will cause the installation to fail.

- **RHEL / CentOS**:

  ```
  # rpm -ivh smartsense-hst-1.1.0-67.x86_64.rpm
  ```

- **Ubuntu / Debian**:

  ```
  # dpkg -i smartsense-hst-1.1.0-67.deb
  ```

1.2.6.1.1. SmartSense HST Server Placement

One node in the cluster must be designated as the HST server, as this component consolidates all of the HST Agent's collected data together into a single downloadable file, referred to as a "bundle". For this functionality, the HST Server must be setup on a node in the HDP cluster where inbound access to the preferred HTTP(S) server port is accessible. The HST Server will listen on the configured HTTP(S) port chosen during setup.

It is preferred, however not required, that when installed outside of Ambari the HST Server node have passwordless root access via SSH to all HST Agent nodes in the cluster. This access will allow for single-click capture through the HST Server Web UI however if not available, captures can be performed from the CLI.

Administrators, and each HST Agent in the cluster must have network access to the HST Server. This connectivity is required for agents to consolidate their data, and for Hadoop administrators to download completed bundles. To set up the HST Server instance, run the following command and follow the instructions below. The entries that need to be supplied based on the specific environment are referenced in bold {}’s. Initiate the HST setup wizard with the following command:

```
# hst setup

Welcome to Hortonworks SmartSense Tool
Enter Account Name: Enter your Account Name from Hortonworks Support Portal
Enter SmartSense ID: Enter Your SmartSense ID from Support Portal
Enter notification email
: Email address where the SmartSense notifications should to be sent
Enter storage directory (minimum: 10 GB, default: /usr/hdp/share/hst/hst-
server/data): Preferred local storage directory to store collected bundles
```

```
Web UI Port (default:9000): Preferred port number
Enable Web UI SSL (Enabling requires a certificate)? [y/n] (default: n): n
Looking for available JDKs at /usr/jdk64
Enter java home directory: Path to java home
Enter Cluster Name: HDP Cluster Name
Is "{Cluster Name}" cluster secured? [y/n] (default: n): Enter y if Kerberos
 is enabled
```

Note

This setup command can be run any number of times. Each invocation will automatically restart the HST Server to allow the new changes to take effect.

HST server uses a storage directory (default: `/usr/hdp/share/hst/hst-server/data`) to store all collected bundles. The minimum recommendation size of that directory is 10GB. In order to change that restriction, change the configuration `property min_required_storage` in the `/etc/hst/conf/hst-server.ini` file.

1.2.6.1.2. SmartSense HST Agent Setup

The HST Agents should be set up on all HDP cluster nodes (including HST server) and configured with the HST Server host. To do so, run the following command:

```
# hst setup-agent --server=HST server’s fully qualified domain name
```

1.2.6.1.3. Enable Capture Through UI

To enable this feature, passwordless root SSH access is required from the HST Server to all agents in the cluster. If passwordless root SSH access is not available, the HST Agent CLI Capture process must be used. To configure HST for Web UI capture, follow the steps below:

1. Edit the `/etc/hst/conf/hst-server.ini` file. The **[client]** section contains two properties that need to be updated to enable remote capture:

   ```
   [client]
   ; thread pool maximums
   threadpool.size.max=50
   ; Password-less SSH enabled or not
   password_less_ssh.enabled=
   ; SSH key for data capture
   sshkey.path=
   ```

   The `password_less_ssh.enabled` property should be set to `true` in an SSH environment that uses a single private key on the HST Server, with the corresponding public key distributed in the authorized_keys file on all HST Agents.

   The `sshkey.path` is used in an SSH environment where there is a distributed private key on all HST Agents. For this scenario, the property should be set to the path of the private key to use. For example: `/root/.ssh/id_rsa`.

2. Once the appropriate properties have been updated, restart the HST Server using the following command:

   ```
   # hst restart
   ```

1.2.6.2. Capturing Bundles in a Non-Ambari Environment

There are two options available for data capture when HST is deployed outside of Ambari: `<CLI>`, and *Web UI* capture. The following sections will outline setting up both options.

1.2.6.2.1. HST Server Web UI Capture

For this option the HST Web UI for Capture needs to be enabled. Follow steps outlined in the Enable Capture Through UI section to enable. The Web UI capture method allows users to capture data by simply clicking the desired services to capture, entering the case number, and clicking **Capture**.

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

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

1.2.6.2.2. HST Agent CLI Capture

HST Agents collect data for the specific node they are installed on. In order to capture data for all nodes in the cluster, which is the most common use case, the hst capture command must be run 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*, as it allows for bundles to be captured in the least amount of time. In order for the all agents to consolidate data in the same bundle, it is important that all agents initiate capture *within 3 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 [level]
```

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

```
# hst list-services

Supported services:
  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
```

Services can be specified individually, combined using commas as a delimiter, or specified all using the ‘all’ keyword.

For example, to capture data just for HDFS and for case number 0001, run hst as follows:

```
# hst capture HDFS 0001
```

To capture data for HDFS, Hive, and 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
```

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

Once a bundle has been initiated, the HST Server Web UI can be used to check bundle status and download the bundle once it is complete.

1.2.6.3.1. HST Server Login

To access the HST Server Web UI, navigate to

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

The default username and password is:

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

1.2.6.3.2. View And Download Bundles

Once logged in, the list of bundles is available from the “Bundles” link. This page shows all bundles that have been collected and their status. If data is still being collected, the UI will automatically update itself with the progress of the capture process until completed. Once a bundle is in a completed state, it is ready for download.

Once downloaded, the bundle can be uploaded to Support for Support Case Troubleshooting, or to the SmartSense Environment for SmartSense Analysis by following the steps outlined in the following sections:

- Uploading SmartSense Analysis Bundles to Hortonworks SmartSense [ ]
- Uploading Bundles to Hortonworks [ ]

1.2.6.4. Configure Anonymization Rules in a Non-Ambari Environment

1. SSH to 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 the Configure Data Anonymization Rules section.

1.2.6.5. Upgrading from HST 1.0.x to SmartSense 1.1.0 in a Non-Ambari Environment

To upgrade from HST 1.0.x to SmartSense 1.1.0, please first remove HST 1.0.x from **all hosts**:

```
# rpm -e hst
# rm -rf /usr/hdp/share/hst
# rm -rf /etc/hst
# rm -rf /var/log/hst
```

Once this is complete, see the Installing SmartSense without Ambari section.

1.2.6.6. Uninstalling SmartSense 1.1.0 in a Non-Ambari Environment

To uninstall SmartSense, run following commands on **all hosts**:

1. Remove packages:

   - **RHEL / CentOS**:

     ```
     # rpm -e smartsense-hst
     ```

   - **Ubuntu / Debian**:

     ```
     # dpkg -r smartsense-hst
     ```

2. Remove data generated by SmartSense:

   ```
   # rm -rf /usr/hdp/share/hst
   # rm -rf /etc/hst
   # rm -rf /var/log/hst
   ```

1.2.7. Uploading Bundles to Hortonworks

Once a bundle has been captured, it needs to be uploaded to Hortonworks depending on the intent chosen during capture.

SmartSense Analysis

If a bundle has been captured for SmartSense Analysis and you're looking for recommendations on cluster performance, operations, and security, your encrypted SmartSense Analysis bundle needs to be uploaded to the hosted SmartSense Analysis environment. Please see the Uploading SmartSense Analysis Bundles to Hortonworks SmartSense section for details about connecting to and uploading bundles to this environment.

Support Case Troubleshooting

If you are working with Hortonworks Support to troubleshoot an issue, your Support Case Troubleshooting bundle needs to be either attached to your support case, if it's below 65MB, or uploaded to Exavault if it's larger. Please see the Uploading Support Case Bundles to Hortonworks Support section for details about connecting to and uploading bundles to Exavault.

1.2.7.1. Uploading SmartSense Analysis Bundles to Hortonworks SmartSense

Credentials and connectivity details for the SmartSense Analysis environment contain upload credentials that are only available to Hortonworks Support customers. Clicking on the following link will take you to the Hortonworks Support Portal Knowledge Base article that contains specific details on connecting to and uploading bundles to this environment. To view this article, you must have a valid Hortonworks Support Portal login credential: https://hortonworks.my.salesforce.com/articles/en_US/How_To/SmartSense-Analysis-Bundles.

Once Hortonworks has successfully received and processed your bundle, you will be notified via email.

1.2.7.2. Uploading Support Case Bundles to Hortonworks Support

Credentials and connectivity details for uploading SupportCase Troubleshooting bundles contain upload credentials that are only available to Hortonworks Support customers. Clicking on the following link will take you to the Hortonworks Support Portal Knowledge Base article that contains specific details on connecting to and uploading bundles to Hortonworks. To view this article, you must have a valid Hortonworks Support Portal login credential: https://hortonworks.my.salesforce.com/articles/en_US/How_To/SmartSense-Bundles-for-Support-Case-Troubleshooting.

Once the bundle has been uploaded, please update the support case with the filename of the bundle that was uploaded.

1.3. Appendix

1.3.1. Changing Capture Levels

There are three capture levels available in SmartSense HST: L1, L2, and L3. Each level collects a different type of diagnostic data. The levels and what is collected is listed below:

L1 - Configuration

L2 - Configuration and Metrics

L2 - Configuration, Metrics and Logs

The default capture level is L2. To change the capture level, edit the following line in the /etc/hst/conf/hst-agent.ini file on the node where HST server is setup.

```
capture.level=L2
```

1.3.2. Configure Data Anonymization Rules

Anonymization rules define regular expression to anonymize sensitive data (like IP addresses, Domain Names, etc.). 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 is `null` means all files)
   - `pattern` - Regular expression to defined the pattern to match within the file
   - `extract` - An optional regular expression to extract the data from the matched pattern. Each of the extracts will be marked as regular expression group.
   - `shared` - Flag to indicate which key to use for anonymization the (`shared` or `private`) key to use for masking. If shared key is used, Hortonworks support team would be able to unmask data if needed for diagnostic purposes. For example hostname and IP addresses for resolving issues on specific hosts or communication between hosts. Please note, 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 above. For example, if the pattern is '.*dfs.datanode.*', the value should not contain 'dfs.datanode'. Also, note that if the value is specified, `shared` flag will be 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` - It can be either DELETE or REPLACE. Default is REPLACE. If DELETE is specified, the property will be removed from the config file and if REPLACE is specified, the property value will be replace by either constant `value` or masked value.
- `value` - An optional value for REPLACE operation. If not specified, private or shared key is used to mask the data to replace.
- `enabled` - Flag to enable/disable rule definition, 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 may become a roadblock for support case analysis)

1.3.3. Change Server/Agent Configurations in a Non-Ambari Environment

All SmartSense HST configurations are stored in `/etc/hst/conf`. There is `hst-server.ini` and `hst-agent.ini` which has 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 restart of HST Server.

1.3.4. What’s New In This Release

- Ambari integration
  - Ability to install, operate and manage HST using Ambari
  - Ability to capture and download bundles through Ambari View
- Support for additional platforms
  - CentOS 7.x
  - Debian 6.x
  - RHEL 7.x
  - Ubuntu 12.x
- Ability to anonymize bundle data
- Ability to encrypt bundle data

- Support for additional HDP services
  - Falcon
  - Kafka
  - Knox
  - Spark
  - Sqoop
  - Storm
- Performance improvements for large clusters
- Recommendations view through Hortonworks support portal

1.3.5. Known Issues

- **Issue**: When integrated with Ambari 2.1, SmartSense will prevent rolling upgrades from successfully finalizing.
- **Workaround**: Prior to attempting Rolling Upgrade, you must edit the SmartSense stack definition to correct the `<versionAdvertised>` value. On the Ambari Server host, remove the “`<versionAdvertised>true</versionAdvertised>`” tags from the `/var/lib/ambari-server/resources/stacks/HDP/2.0.6/services/SMARTSENSE/metainfo.xml` file and restart the Ambari Server. Once this has been done, proceed with your Rolling Upgrade process as normal.
- **Issue**: Some CentOS 7 installations will not capture all diagnostic data.
- **Workaround**: Some CentOS 7 environments (typically minimal CentOS 7) do not have net-tools package installed. Installing it (yum install net-tools) will address this issue. In general please make sure that all prerequisites are met.
- **Issue**: In some rare situations HST install fails on certain nodes during Ambari’s deploy stage.
- **Workaround**: Use “Retry” option available through the Ambari UI.
- **Issue**: Occasionally on some platforms only during first bundle capture attempt through HST UI, only Ambari data gets captured.
- **Workaround**: Initiate capture again after first capture was successfully completed.
- **Issue**: SmartSense install through Ambari fails if any of the following conditions occurs:
  - Bad SmartSense ID
  - Invalid notification email
  - Insufficient space available for bundle storage
- **Workaround**: Follow these steps:

1. Review the failure reason by clicking on the failed host/task link in Ambari Wizard
2. Cancel the SmartSense install
3. Delete the failed installation from Ambari: `curl -u admin:$PASSWORD -H 'X-Requested-By: ambari' -X DELETE http://AMBARI_SERVER_HOST:8080/api/v1/clusters/CLUSTER_NAME/services/SMARTSENSE`.
4. Follow the add service wizard again (this time, make sure that the settings that cause failure are addressed)

- **Issue**: During some captures the bundles status turns into “TIMED OUT”
- **Workaround**: Verify which agents have not completed processing and check their logs. Failure during agent capture can lead to the bundle timing out. Fix the problem in agents, or skip these agents for subsequent captures.
- **Issue**: Capture status stuck when multiple captures executed within the same capture window.
- **Workaround**: Initiating new capture while another capture is in progress is not supported at this time. Please let the ongoing capture finish before initiating new capture.
- **Issue**: SmartSense UI indicates its working but does not display content for a while.
- **Workaround**: This usually happens when browser UI can not communicate with the server or the session has been timed out. Refresh the browser.
- **Issue**: After clicking on logout, sometimes doesn't redirect user to login page.
- **Workaround**: Refresh the browser (or reloading) fixes the issue.
- **Issue**: When more than one capture is initiated at the same time produces bundles with some extra node files.
- **Workaround**: Please wait for the capture to finish before issuing another capture. Capture is based on CASE NUMBER and it will work if CASE NUMBER for simultaneous captures are different.
- **Issue**: In bundles page, order of display of the bundle gets messed up when DELETE operation is performed. Older bundles are displayed at the top.
- **Workaround**: Reload the page and clicking on Bundles link might fix the issue.
- **Issue**: hst setup fails on Debian 6
- **Workaround**: Install chkconfig using apt-get install chkconfig
- **Issue**: In some situations, the stack components for SmartSense are removed and cause the Ambari Server to fail to start with the following error:

```
17 Sep 2015 20:36:43,986 ERROR [main] AmbariServer:717 - Failed to run the
 Ambari Server
```

```
org.apache.ambari.server.StackAccessException: Stack data, stackName=HDP,
 stackVersion=2.3, serviceName=SMARTSENSE
        at org.apache.ambari.server.api.services.AmbariMetaInfo.
getService(AmbariMetaInfo.java:517)
        at org.apache.ambari.server.api.services.AmbariMetaInfo.
getComponent(AmbariMetaInfo.java:285)
        at org.apache.ambari.server.controller.utilities.DatabaseChecker.
checkDBConsistency(DatabaseChecker.java:95)
        at org.apache.ambari.server.controller.AmbariServer.
run(AmbariServer.java:228)
        at org.apache.ambari.server.controller.AmbariServer.
main(AmbariServer.java:715)
```

- **Workaround**: Re-run hst add-to-ambari to re-add the stack and view to the Ambari Server instance.