Hortonworks SmartSense

Installation

docs.cloudera.com

Hortonworks SmartSense: Installation

Copyright © 2012-2016 Hortonworks, Inc. Some rights reserved.

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.

The *Hortonworks SmartSense Installation* provides you with the latest SmartSense 1.3 installation and upgrade information. To better navigate this document, read the SmartSense Architecture, System Requirements, and Downloading SmartSense Binaries sections, then select your scenario from the list below:

- Bundle Content [3]

- System Requirements [7]

- Installing SmartSense in an Ambari Environment [11]

- Installing SmartSense in a Non-Ambari Environment [16]

- Where do I upload the bundle once I've downloaded it?

- Installing SmartSense Gateway [12]

- SmartSense Ports and Traffic Flow [25]

- Reporting Issues [29]

2. SmartSense Architecture

The Hortonworks SmartSense Tool (HST):

1. Collects cluster diagnostic information to help you troubleshoot support cases.

2. Automatically captures and uploads bundles that are used to produce customized recommendations for your cluster on areas of improvement, such as performance, operational stability, and security.

3. (New in SmartSense 1.3.0) Reports, analyzes, and visualizes cluster activity.

You can install HST as a standalone component, manually installing it on all nodes of the cluster, or use Apache Ambari to automate installation and service management.

2.1. Cluster Diagnostic Collection

The HST agents capture, anonymize, and encrypt cluster diagnostic data, and then send it to the central HST server to coalesce into a single downloadable file called a *bundle*. The HST agent processes are short-lived services that are started only for specific data capture tasks. To provide the most complete picture of cluster utilization, HST agents must be installed on every node in the cluster. After an HST agent has captured the requested data from the host it is installed on, the process exits.

The following image illustrates the communication between HST agents and the HST server:
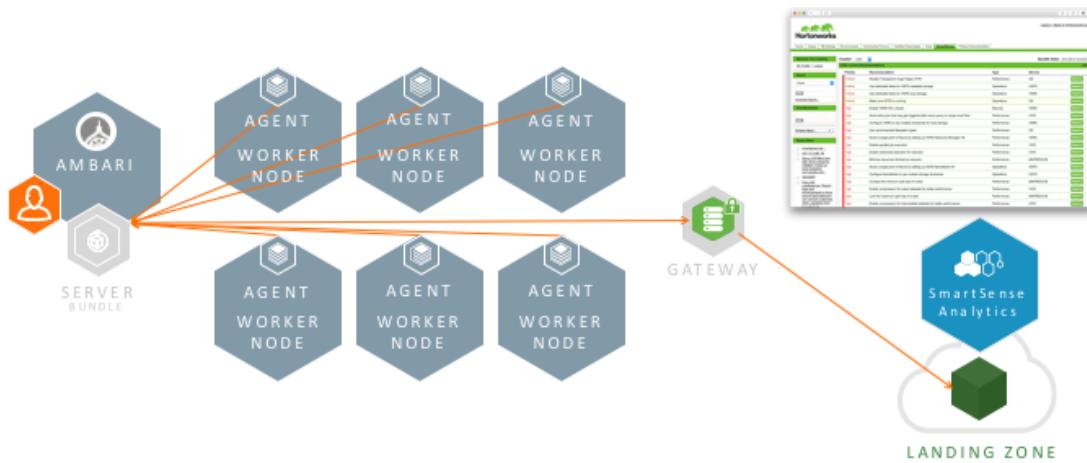
SmartSense anonymizes and encrypts the diagnostic information captured in the bundle. For more information about extending the anonymization process with site-specific rules, see Configure Anonymization Rules with Ambari or Configure Anonymization Rules in a Non-Ambari Environment.

There are two types of bundles: one for ad-hoc troubleshooting of support cases, and the other for analysis and recommendations.

SmartSense Troubleshooting Bundles

Bundles captured for troubleshooting contain configuration and metrics for each node in the cluster, and logs for only the subset of services and hosts that you chose before initiating the capture process. The purpose of these bundles is to provide support engineers with basic diagnostic information that can help them understand the state of your cluster so that they can troubleshoot and quickly resolve issues.

SmartSense Analysis Bundles

Bundles captured for analysis contain configuration and metrics for each node in the cluster, but do not contain any logs. Their purpose is to produce recommendations for changing your cluster configuration to ensure better security, performance, and operations. These recommendations are available in the SmartSense tab on the Hortonworks Support Portal.

For more information about bundles, see Bundle Content and Bundle Security

2.1.1. Bundle Content

SmartSense collects the following types of data:

- Operating system:

  - Configuration (partition layouts, file system mount options, key service status, network configurations, and so on)

  - Metrics (CPU, memory, I/O statistics, network statistics, and so on)

  - Logs (system messages and driver messages)

- Hortonworks Data Platform (HDP) service:

  - Configuration

  - Metrics (JMX reports and installed packages)

  - Logs (only for support case troubleshooting: not for SmartSense analysis)

  - Summary of cluster activity

To see data and files that are captured in your specific environment, perform a capture and then download the unencrypted bundle. To see a step-by-step example of how to do this, refer to the How to inspect SmartSense bundle contents Hortonworks Community Connection post. If any files contain information that you would like to remove, replace, or anonymize, refer to Configuring Data Anonymization Rules.

2.1.2. Bundle Security

Hortonworks takes security seriously. Multiple levels of provisions ensure that sensitive data is protected:

- Anonymization and exclusions:

  - IP addresses and host names are *always* anonymized.

- Passwords are not collected.

- Encryption:

  - SmartSense analysis bundles are optionally encrypted using AES + RSA encryption.

- Further customizations:

  - You can configure custom anonymization rules to include environment-specific patterns.

  - By default, all IP addresses, the domain component of fully qualified domain names, and S3 and WASB access keys are anonymized.

  - You can add custom configuration to 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 Apache Ambari and HDP stack version, node count, and amount of storage available and used, are stored for trending rules analysis. Recommendations generated for each bundle are available through the Hortonworks Support Portal and are stored for feedback purposes and used to improve future recommendations.

2.2. Bundle Transport

After a bundle has been captured, there are four ways to upload that bundle to Hortonworks:

- Automated Bundle Upload:

  - HST server

  - SmartSense Gateway

- Manual Bundle Upload:

  - HTTPS

  - SFTP

2.2.1. Automated Bundle Upload

Depending on the availability of outbound internet access, you have two choices for automated bundle upload. If the HST server host has outbound internet access, you can configure it to automatically upload captured bundles to Hortonworks. In this case, bundles are uploaded automatically over HTTPS from the HST server to the externally hosted Hortonworks SmartSense environment. If the HST server does not have outbound internet access, you can deploy a standalone SmartSense Gateway to forward bundles to the hosted Hortonworks SmartSense environment.

HST Server

After a bundle has been captured, the HST server attempts to upload bundles to the Hortonworks hosted environment over HTTPS by default. This upload succeeds if your HST server host has outbound internet access. If your HST server host does not have outbound internet access, you have two options. If the HST server host can use a corporate HTTP proxy to upload bundles, you can configure your HST server host to do so using Configuring Bundle Upload, or you can use the SmartSense Gateway.

The following image illustrates bundle upload using the HST server:

SmartSense Gateway

For those whose HST server hosts do not have outbound internet access, Hortonworks created the SmartSense Gateway, which simplifies uploading bundles to Hortonworks. You can deploy a single gateway that supports multiple internal HST server deployments. In this deployment scenario, you do not need direct outbound internet access from the HST server to upload bundles. You need access only from the HST server to the gateway, and the gateway uploads all bundles to Hortonworks Support or to the SmartSense environment for SmartSense analysis.

The following image illustrates bundle upload using the SmartSense Gateway:
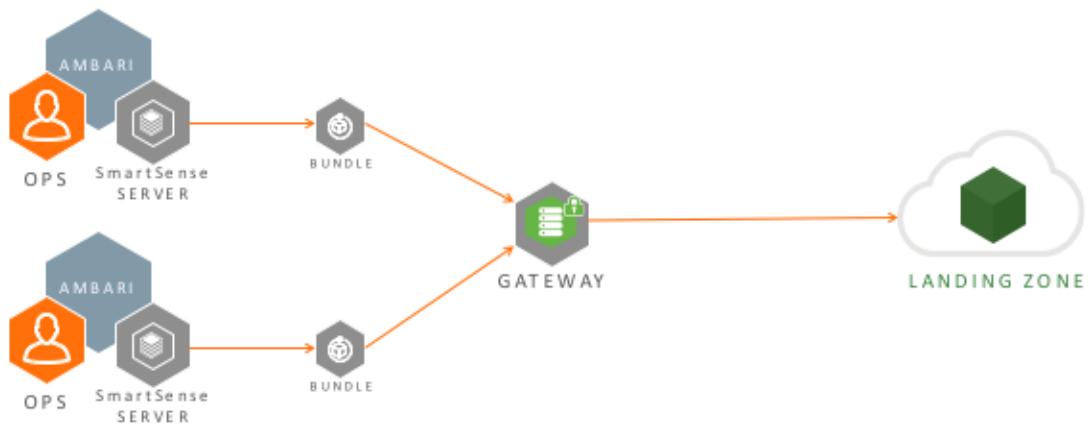
2.2.2. Manual Bundle Upload

If you are just getting started with SmartSense, you might still be waiting on your security or network operations resources to provide the necessary access for the HST server or the SmartSense Gateway to send bundles. If you are in this situation, you have one manual upload option: a browser-based HTTPS upload.

After a bundle has been captured, you can go to SmartSense view in Ambari and download the bundle onto your desktop. You can then navigate to https://smartsense.hortonworks.com/ and log in using the credentials and steps specified in the following 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).

> **Note**
>
> Using an SFTP-based gateway is deprecated, effective end of April 2018. If you are using SFTP-based gateway you should upgrade to HTTPS-based gateway.

2.3. Activity Analysis

There are two new components in SmartSense 1.3.0: Activity Analyzer and Activity Explorer. These two components provide job utilization metric aggregation, reporting, and visualization for YARN-based workloads.

Activity Analyzer

Activity Analyzer communicates with YARN Application Timeline Server v1.5 and later, and with Hadoop Distributed File System (HDFS) to consume MapReduce history data. It aggregates and transforms this data, and stores it in the Ambari Metrics Collector.

Activity Explorer

Activity Explorer includes an embedded instance of Apache Zeppelin, which hosts prebuilt notebooks that visualize cluster utilization data for YARN, Apache Hive or Apache Tez, and MapReduce workloads. Specifically, Activity Explorer includes data related to user, queue, job duration, and job resource consumption.

The following image illustrates how Activity Analyzer sends aggregated job history data to Ambari Metrics Collector, which makes that data available to Activity Explorer:
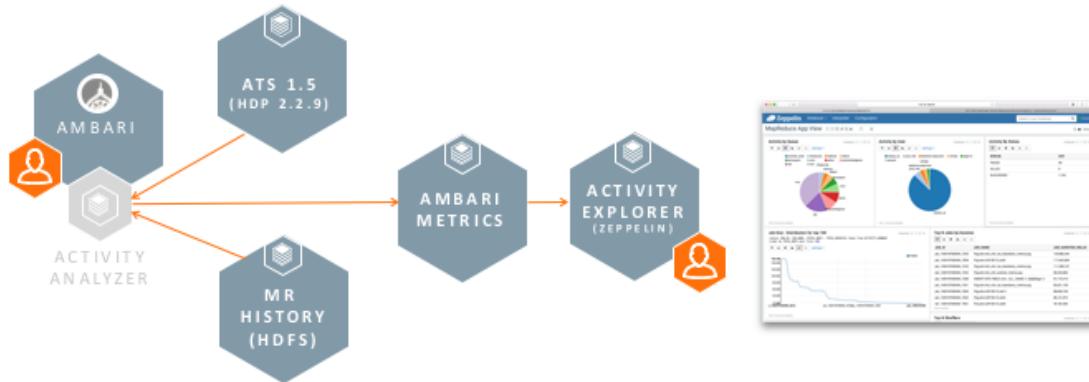
3. SmartSense Installation

Installing Hortonworks SmartSense Tool (HST) on a Hortonworks Data Platform (HDP) cluster involves meeting minimum system requirements, installing SmartSense, and choosing how bundles will be uploaded to Hortonworks.

3.1. System Requirements

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

- Operating System Requirements [7]

- Software Requirements [7]

- JDK Requirements [8]

- Browser Requirements [8]

- Ambari Requirements [8]

- Requirements for Activity Analysis [9]

3.1.1. Operating System Requirements

The following 64-bit operating systems are supported:

- CentOS 5 (deprecated)

- CentOS 6

- CentOS 7 (it's recommended to have the net-tools package installed)

- Debian 6

- Debian 7

- Red Hat Enterprise Linux (RHEL) 5 (deprecated)

- Red Hat Enterprise Linux (RHEL) 6

- Red Hat Enterprise Linux (RHEL) 7 (it's recommended to have net-tools package installed)

- SUSE Linux Enterprise Server (SLES) 11 SP3

- Ubuntu 12.04

- Ubuntu 14

3.1.2. Software Requirements

You must install the following packages 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

- Python2, version 2.6 or later

3.1.3. JDK Requirements

The following Java runtime environments are supported:

- Oracle JDK 1.8 64-bit

- Oracle JDK 1.7 64-bit

- OpenJDK 8 64-bit

- OpenJDK 7 64-bit

3.1.4. Browser Requirements

The HST server runs a browser-based web application. To use this application, you must have a machine capable of running a graphical browser. Although you should update your browser to the latest stable version, the minimum required browser versions per operating system are as follows:

- Windows (Vista, 7 or later)

  - Internet Explorer 11.0

  - Firefox 39

  - Google Chrome 43

- Mac OS X (10.6 or later)

  - Safari 8

  - Firefox 39

  - Google Chrome 43

3.1.5. Ambari Requirements

You can integrate SmartSense with and deploy it through Apache Ambari. Ambari integration is certified with the following versions:

- Apache Ambari 2.x

3.1.6. Requirements for Activity Analysis

To use SmartSense Activity Analysis, you need the following component versions:

- Apache Tez 0.5.2 or later

- HDP 2.3 - 2.3.4.7+

- HDP 2.4 - 2.4.x

- HDP 2.5 - 2.5.x

> **Note**
>
> Ensure that your YARN App Timeline Server (ATS) is at least version 1.5. For information on how to upgrade to 1.5, refer to the Timeline Server 1.5 Overview documentation.

3.2. Downloading and Installing SmartSense Binary

SmartSense 1.3.0 is automatically included with Ambari 2.4. Earlier versions of SmartSense are included with Ambari 2.2, and Ambari 2.1 and 2.0 do not include SmartSense.

If you want to use SmartSense 1.3.0 with Ambari 2.0 or in a non-Ambari installation, you can download it from the **Tools** tab of the Hortonworks support portal (https://support.hortonworks.com).

To install SmartSense, follow these steps:

1. Install the SmartSense package on the Ambari server host:

    - **RHEL, CentOS, or SLES**:

      ```
      # rpm -ivh smartsense-hst-$HST_VERSION.x86_64.rpm
      ```

    - **Debian or Ubuntu**:

      ```
      # dpkg -i smartsense-hst_$HST_VERSION.deb
      ```

2. If using the non-root agent capabilities of Ambari 2.0 or 2.1, ensure that the following commands are added to the '# Ambari Commands' section of the `/etc/sudoers` file on each node in the cluster: `/usr/bin/dpkg *`, `/bin/rpm *`, and `/usr/sbin/hst *`.

3. Add SmartSense service to Ambari by running `hst add-to-ambari`.

    If you are using Ambari 2.1.2 or earlier, you will be asked for the full path of the previously downloaded RPM or DEB package; for example, `/root/smartsense-hst-$HST_VERSION.x86_64.rpm`.

```
# hst add-to-ambari
Enter SmartSense distributable path: /root/smartsense-hst-$HST_VERSION.
x86_64.rpm
Added SmartSense service definition to Ambari

NOTE: It is required to restart Ambari Server for changes to reflect. Please
 restart ambari using 'ambari-server restart'
```

4. Restart Ambari server by running `ambari-server restart`.

After you complete this task, you should read HST Server Placement, Activity Analyzer Placement, and follow the steps in Installing SmartSense with Amabri

3.3. HST Server Placement

You should designate one node in the HDP cluster as the HST server, so that this component can efficiently consolidate the data collected by all HST agents into a single downloadable file (referred to as a *bundle*).

Although it is not a requirement, when you install HST without using Ambari, you should provide the HST server node with passwordless root access, through SSH, to all HST agent nodes in the cluster. This access enables single-click capture through the HST server web UI in a non-Ambari environment. If the web UI is not available, you can still perform captures 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. For a full list of ports and a data flow diagram, refer to SmartSense Ports & Traffic Flow.

3.4. Activity Analyzer Placement

The Activity Analyzer component has the ability to extract, aggregate, and store utilization data for all three supported analyzers: HDFS, YARN, and MapReduce & Tez. Before installing SmartSense, you should understand how and where to deploy and place these analyzers. You must install multiple Activity Analyzer instances, the exact number depending on which analyzers that you are planning to use and if HDFS is configured for NameNode HA.

> **Note**
>
> Activity Analyzers need HDFS, YARN, MR, and Tez clients installed on the same host as the analyzer.

HDFS Analyzer

For HDFS analysis, an Activity Analyzer needs to be deployed to **each** NameNode in the cluster. These instances will automatically begin processing the fsimage on startup and will reprocess the latest fsimage data once every 24 hours. By default, when deployed on a NameNode, these Activity Analyzers do not process YARN, or MapReduce & Tez utilization

data; This is to reduce the amount of processing done on servers hosting critical services like the NameNode.

Resource requirements: HDFS Analyzer typically runs for a very short period of time, its resource consumption depending on fsImage size. For example, analyzing a 200-million-object fsImage is anticipated to take less than 15 minutes; HDFS Analyzer is mostly a single-threaded process and consumes up to one core during this execution time.

YARN, MapReduce & Tez Analyzer

Activity Analyzers deployed to the NameNodes in the cluster do not process any utilization data besides HDFS. Therefore, to process YARN, MapReduce, and Tez utilization data, another instance of the Activity Analyzer needs to be deployed to another node in the cluster, preferably on a **non-master node**. On startup, the Activity Analyzer will check to ensure that it's not deployed to a NameNode, and then will begin to process YARN, MapReduce, and Tez utilization data. This Activity Analyzer individually starts and schedules analysis for YARN applications, MapReduce and Tez jobs. Both the YARN, and MapReduce and Tez analysis constantly polls for completed applications or jobs. Upon completion, each is analyzed and the utilization data is stored in the Ambari Metrics System

3.5. Installing SmartSense in an Ambari Environment

Before you start the installation:

• You should know your SmartSense ID and account name (both are available in the Hortonworks support portal in the **Tools** tab).

• You must also ensure that an Ambari agent is running on the same host as the Ambari server.

To begin the installation, follow these steps:

1. If you are using Ambari 2.0 or a non-Ambari installation and want to use SmartSense 1.3.0, you must first download and install it (Downloading SmartSense 1.3.0 Binary).

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

3. From the list of installable services, select **SmartSense**, and then click **Next**.

4. On the **Assign Masters** page, select cluster nodes for the HST server, Activity Analyzer, and Activity Explorer, and then click **Next**.

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

   • For a list of criteria to determine the best nodes to select for Activity Analyzers see the Activity Analyzer Placement section.

5. On the **Customize Services** page, validate the values in the following fields, as appropriate to your environment:

Ambari 2.1+	Ambari 2.0	Note
Configuration Tab: Basic **Property**: Customer account name	**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	**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	**Configuration Section**: Advanced hst-server-conf **Property**: customer.notification.email	The email address notified when SmartSense bundles have been received and recommendations are ready for your review
Configuration Tab: Basic **Property**: Bundle Storage Directory	**Configuration Section**: Advanced hst-server-conf **Property**: server.storage.dir	The directory on the HST server that will be used to store completed bundles Because bundles can be large, this directory should have at least 1GB of free space.
Configuration Tab: Basic **Property**: Server Temporary Data Directory	**Configuration Section**: Advanced hst-server-conf **Property**: server.tmp.dir	The directory on the HST server that is 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: at least 5 GB of free space.
Configuration Tab: Activity Analysis **Property**: Password for user 'admin'	**SmartSense Configuration Section**: Advanced activity-zeppelin-shiro **Property**: users.admin	Password for the Activity Explorer admin user.

Click **Next**.

The Ambari Stack Advisor assesses your cluster configuration and might alert you to configuration issues. Note that this is not related to SmartSense, and is simply what Ambari does upon adding any service. SmartSense never makes configuration changes to your cluster. No cluster services need to be restarted after installing SmartSense, and any configuration changes that are noticed should be reverted.

If you have a kerberized cluster, you will be prompted for the KDC admin credentials during this step.

6. On the **Review** page, click **Deploy** to complete your SmartSense service installation.

7. Restart the Ambari server:

```
# ambari-server restart
```

3.6. Installing SmartSense Gateway

If your HST server host does not have outbound internet access, you can connect it to a single SmartSense Gateway that does. A single gateway can support multiple internal HST server deployments, uploading all of their bundles to Hortonworks support or to the SmartSense SFTP environment for SmartSense analysis:
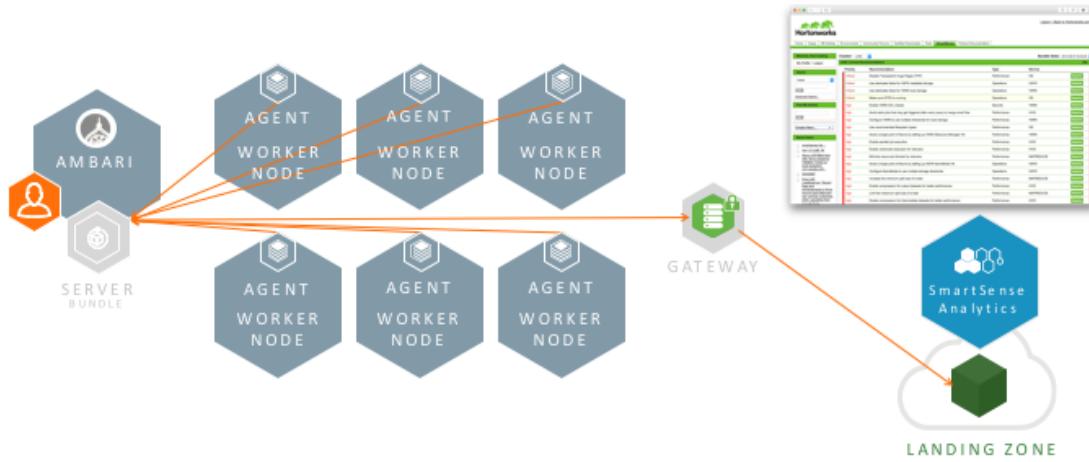
Using the SmartSense Gateway involves knowing where to place it, installing it, integrating it with Ambari, and configuring bundle upload.

3.6.1. SmartSense Gateway Placement

You must deploy the SmartSense Gateway on a host in a network zone that has both outbound internet access and inbound access from all HST server instances. The connectivity between the HST server and the gateway is secured using mutually authenticated SSL.

By default, TCP port 9450 is used to register HST server instances with the gateway. After initial registration, TCP port 9451 is used for the authenticated API communication between the HST server and the gateway. Both the registration and API communication ports are configurable in the gateway `hst-gateway.ini` file.

Outbound connections from the SmartSense Gateway to the external Hortonworks SmartSense analysis environment use SFTP to transmit bundles to Hortonworks. Specific connectivity details for the SmartSense environment are outlined in SmartSense Ports & Traffic Flow.

3.6.2. Installing and Starting SmartSense Gateway

The SmartSense Gateway is not managed by Ambari and must be installed manually. It is included in the same `smartsense-hst-$HST_VERSION` package used for the HST server and HST agent. You access this package in the public Ambari 2.2.0 repository, if you are using Ambari 2.2.x, or on the **Tools** tab of the Hortonworks support portal, for non-Ambari and Ambari 2.1.x installations.

1. Install the SmartSense package on the chosen gateway host:

   • **RHEL, CentOS, or SLES**:

   ```
   # rpm -ivh smartsense-hst-$HST_VERSION.rpm
   ```

   • **Ubuntu or Debian:**

   ```
   # dpkg -i smartsense-hst_$HST_VERSION.deb
   ```

2. Configure the gateway by editing the `/etc/hst/conf/hst-gateway.ini` file:

- Specify the path to the JDK to be used by the gateway in the [java] section:

```
[java]
home={$path_to_your_JAVA_HOME}
```

- To configure SFTP connectivity from the server between , refer to this Knowledge Base article for the SFTP user name, password, host, and port details: https://support.hortonworks.com/s/article/SmartSense-Gateway-setup.

3. Start the gateway:

```
# hst gateway start
```

On startup, the gateway attempts to connect to the configured SFTP host, and if the connection fails, the gateway does not start and logs the reason for connectivity failure to the `/var/log/hst/hst-gateway.log` file.

4. Integrate the gateway with the HST server:

Integrating with Ambari Managed SmartSense or Integrating with Non-Ambari Managed SmartSense.

If you deploy the gateway on a server that is also hosting an HST agent, and that HST agent has been deployed through Ambari, *and* Ambari is configured for nonroot operation, you should run the SmartSense Gateway as the same user that the Ambari agent is configured to run as.

Additionally in this specific scenario, you must modify the following permissions:

```
chown -R ambari:hadoop /var/lib/smartsense/hst-gateway
chown -R ambari:hadoop /var/log/hst
chown -R ambari:hadoop /var/run/hst
```

In this example, "ambari" is the user the Ambari agent has been configured to run as.

3.6.3. Integrating Gateway with Ambari-Managed SmartSense

When you use Ambari to manage SmartSense, integrating with the gateway requires only a few configuration changes within Ambari. Integrating the gateway depends on the version of Ambari you use:

Ambari 2.1	Ambari 2.0	Note
Configuration Tab: Gateway **Property**: Auto-upload bundles	**Expand Configuration Section**: Advanced hst-server-conf **Property**: gateway.enabled	Set this to `true` to enable the gateway to automatically upload bundles after they are created.
Configuration Tab: Gateway **Property**: Gateway host	**Expand Configuration Section**: Advanced hst-server-conf **Property**: gateway.host	The fully qualified domain name of the host running the gateway
Configuration Tab: Gateway **Property**: Gateway port	**Expand Configuration Section**: Advanced hst-server-conf **Property**: gateway.port	The port on the gateway host that the gateway is monitoring (9451 by default)

When enabled, the gateway automatically uploads completed bundles to Hortonworks when a capture is completed.

3.6.4. Configuring Bundle Upload

You can configure SmartSense Gateway to upload bundles by using either SFTP or HTTPS, depending on your corporate network firewall policies:

- Configuring the Gateway to Use SFTP [15]

- Configuring the Gateway to Use HTTPS [15]

3.6.4.1. Configuring the Gateway to Use SFTP

You can configure the gateway to use SFTP to upload bundles to Hortonworks support using the connectivity and configuration details 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).

3.6.4.2. Configuring the Gateway to Use HTTPS

You can configure the gateway to use HTTPS to upload bundles to Hortonworks by using the connectivity and configuration details 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).

To use an authenticated proxy to upload bundles to Hortonworks, follow these steps:

1. On the SmartSense Gateway host, edit the `/etc/hst/conf/gateway/hst-gateway.ini` file and supply the appropriate values for your environment:

```
; All proxy configurations are applicable only for HTTPS provider type
; flag to setup a proxy between gateway and SmartSense environment
;default:false
provider.https.proxy.enabled=true
; fully qualified proxy hostname
provider.https.proxy.hostname=your.proxy.host
; proxy port that will be used by gateway for outbound access
provider.https.proxy.port=3128
; Supported types : HTTP / HTTPS [default:HTTP]
provider.https.proxy.type=HTTP
; Supported types : NONE / BASIC / DIGEST [default:NONE]
provider.https.proxy.auth.type=BASIC
; proxy username for identified auth.type
provider.https.proxy.auth.username=proxyuser
; proxy password for identified auth.type
provider.https.proxy.auth.password=proxypassword
; [optional] any additional proxy setup parameters
; "|" separated for multiple parameters
; for eg: Digest requires some parameters like
; realm=default|nonce=12GHtqeZA!7Ke43
provider.https.proxy.auth.parameters=
```

2. After you update the configuration file, restart the SmartSense Gateway:

```
hst gateway restart
```

3.7. Installing SmartSense in a Non-Ambari Environment

Without Ambari, you must manually install SmartSense HST on every node in the cluster, which requires that you already know your SmartSense ID and account name (available in the support portal **Tools** tab).

- **RHEL, CentOS, or SLES**:

```
# rpm -ivh smartsense-hst-$HST_VERSION.x86_64.rpm
```

- **Ubuntu or Debian**:

```
# dpkg -i smartsense-hst-$HST_VERSION.deb
```

3.7.1. HST Server Setup

For information about HST server placement, see HST Server Placement.

To set up the HST server instance, initiate the HST setup wizard with the following command; the values that you must supply are shown in italics:

```
# 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: 1.00 GB, default: /var/lib/smartsense/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
# hst start
```

You can run this command any number of times. Each invocation automatically restarts the server, implementing your changes.

The HST server uses the default storage directory `/var/lib/smartsense/hst-server/data` to store all collected bundles. The minimum recommended size of that directory is 1 GB, but you can change that value by using the configuration property `min_required_storage` in the `/etc/hst/conf/hst-server.ini` file.

3.7.2. HST Agent Setup

You should set up HST agents on all HDP cluster nodes, including the HST server, and configure them with the HST server host:

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

3.7.3. Enable Capture Through the UI

To enable this feature, you must have passwordless root SSH access from the HST server to all agents in the cluster. (Otherwise, you must use the HST Agent CLI Capture).

To configure HST for web UI capture, follow these steps:

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=
   ```

   • Set the `password_less_ssh.enabled` property 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.

   • You can use the `sshkey.path` property in an SSH environment in which 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 be used. For example: `/root/.ssh/id_rsa`.

2. After you have updated the appropriate properties, restart the HST server:

   ```
   # hst restart
   ```

3.7.4. Integrating the Gateway with Non-Ambari Managed SmartSense

To configure a SmartSense deployment that is not using Ambari for management to communicate with an installed gateway, you must first update the HST server configuration:

1. On the HST server, edit the `/etc/hst/conf/hst-server.ini` file **[gateway]** section:

   a. Enable the gateway's automatic upload capability, so that after a bundle successfully captured, the HST server automatically uses the gateway to upload the bundle:

   ```
   enabled=true
   ```

   b. Specify the fully qualified domain name of the server hosting the gateway:

   ```
   host=gateway-host.yourdomain.com
   ```

   c. Specify the port on the gateway host from which the gateway is configured to monitor:

   ```
   port=9451
   ```

2. Restart the HST server:

```
#hst restart
```

3.8. Uninstalling SmartSense 1.3.0 in a Non-Ambari Environment

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

1. Remove packages:

   • **RHEL, CentOS, or SLES**:

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

   • **Ubuntu or 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
   # rm -rf /var/lib/smartsense
   ```

3.9. Uninstalling SmartSense Gateway

To remove the SmartSense Gateway, follow these steps:

1. Ensure that the SmartSense Gateway is stopped:

   ```
   # hst gateway stop
   ```

2. Remove the smartsense-hst package:

   • **RHEL, CentOS, r SLES:**

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

   • **Ubuntu or Debian:**

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

3. Remove logs produced by the gateway:

   ```
   # rm /var/log/hst/hst-gateway.*
   ```

4. SmartSense Upgrade Scenarios

Depending on your current and target SmartSense versions, you can upgrade to SmartSense 1.3 in place (without unistallation), or you can uninstall your current version and replace it with SmartSense 1.3.

Current SmartSense Version	Target SmartSense Version	Upgrade Path
1.0	1.3.x - Non-Ambari Managed	Uninstall, Install
1.0	1.3.x - Ambari Managed	Uninstall, Install with Ambari
1.1 - Non-Ambari Managed	1.3.x - Ambari Managed	Uninstall, Install with Ambari
1.1 - Non-Ambari Managed	1.3.x - Non-Ambari Managed	In-Place Upgrade
1.1 - Ambari Managed	1.3.x - Ambari Managed	In-Place Upgrade
1.2 - Non-Ambari Managed	1.3.x - Non-Ambari Managed	In-Place Upgrade
1.2 - Ambari Managed	1.3.x - Ambari Managed	In-Place Upgrade

4.1. In-Place Upgrade

Your in-place upgrade method depends on whether you are working in an Ambari environment and, if so, what version of Ambari you are currently using:

- In-Place Upgrade with Ambari 2.4 [19]

- In-Place Upgrade with Ambari 2.0-2.2 [21]

- In-Place Upgrade without Ambari [22]

4.1.1. In-Place Upgrade with Ambari 2.4

When using SmartSense 1.1 or 1.2, you can upgrade to SmartSense 1.3 without uninstalling SmartSense:

1. Log in to Ambari web UI and stop the SmartSense service.

2. Upgrade binaries on the HST server and all HST agents on *every node* in the cluster, assuming that the Ambari 2.4 repository is configured on all nodes in the cluster:

   - RHEL or CentOS

     ```
     yum clean all
     yum info smartsense-hst
     ```

     In the `info` output, visually validate that there is an available version containing "1.3.0":

     ```
     yum upgrade smartsense-hst
     ```

   - SLES

     ```
     zypper clean
     zypper info smartsense-hst
     ```

     In the `info` output, visually validate that there is an available version containing "1.3.0":

```
zypper up smartsense-hst
```

- Ubuntu or Debian

```
apt-get clean all
apt-get update
apt-cache show smartsense-hst | grep Version
```

In the info output, visually validate that there is an available version containing "1.3.0":

```
apt-get install smartsense-hst
```

3. Upgrade Ambari service and Ambari view by running the following command as the root user from the machine running the Ambari server:

```
# hst upgrade-ambari-service
Please enter Ambari Server hostname (ambari-server.hortonworks.local):
Please enter Ambari Server port (8080):
Please enter Ambari admin user id (admin):
Please enter password for admin:

Un-installing old view ...
Installing new view ...
Removing deprecated alerts ...
Updating SmartSense configurations in Ambari ...

SmartSense service upgrade completed!
NOTE: It is required to restart Ambari Server for changes to reflect. Please
 restart ambari using 'ambari-server restart'
```

4. Restart the Ambari server:

```
# ambari-server restart
```

5. After the packages are upgraded and the HST upgrade is complete, log in to Ambari web UI and start all SmartSense services.

6. Since SmartSense 1.3.0 includes two new components, Activity Analyzer and Activity Explorer, you need to add these components to Ambari. To do this:

   a. In Ambari web UI, click on **Hosts** and then navigate to the host on which you want to install the components.

   > **Note**
   >
   > Refer to the Activity Analyzer Placement section for advice on which hosts to choose when placing the Activity Analyzer.

   b. Click on **Add** and add Activity Analyzer and then Activity Explorer component.

   c. Start Activity Analyzer and Activity Explorer components by selecting **Start** next to the component name.

7. Ensure that all components are upgraded by triggering a SmartSense analysis capture, and ensure that the capture finishes successfully.

8. If you are upgrading from SmartSense 1.2.x to 1.3, you need to also upgrade your HST Gateway:

   - If the HST Gateway is installed on the same node as HST Server or HST Agent, then the HST Gateway will get upgraded along with them.

   - If the HST Gateway is a standalone node outside of the cluster, perform upgrade steps described in Upgrading SmartSense Gateway.

4.1.2. In-Place Upgrade with Ambari 2.0-2.2

When using SmartSense 1.1 or 1.2, you can upgrade to SmartSense 1.3 without uninstalling SmartSense:

1. Log in to Ambari web UI and stop the SmartSense service.

2. Download the SmartSense 1.3 binaries from the **Tools**tab of the Hortonworks support portal (https://support.hortonworks.com).

3. The SmartSense 1.3 binaries need to be manually copied to *every node* in the cluster.

4. Once the binaries have been manually copied, they need to be used to upgrade the HST server and HST agents on *every node* in the cluster. To do so, follow the steps below:

   - RHEL, CentOS, or SLES

     ```
     rpm -Uvh smartsense-hst-HST_VERSION.rpm
     ```

   - Ubuntu or Debian

     ```
     dpkg -i smartsense-hst-HST_VERSION.deb
     ```

5. Upgrade Ambari service and Ambari view by running the following command as the root user from the machine running the Ambari server:

   When using Ambari 2.0-2.1.3:

   ```
   # hst upgrade-ambari-service
   Enter SmartSense distributable path: /tmp/smartsense-hst-HST_VERSION.rpm
   Please enter Ambari Server hostname (ambari-server.hortonworks.local):
   Please enter Ambari Server port (8080):
   Please enter Ambari admin user id (admin):
   Please enter password for admin:

   Un-installing old view ...
   Installing new view ...
   Removing deprecated alerts ...
   Updating SmartSense configurations in Ambari ...

   SmartSense service upgrade completed!
   NOTE: It is required to restart Ambari Server for changes to reflect. Please
    restart ambari using 'ambari-server restart'
   ```

   When using Ambari 2.1.3-2.2:

```
# hst upgrade-ambari-service
Please enter Ambari Server hostname (ambari-server.hortonworks.local):
Please enter Ambari Server port (8080):
Please enter Ambari admin user id (admin):
Please enter password for admin:

Un-installing old view ...
Installing new view ...
Removing deprecated alerts ...
Updating SmartSense configurations in Ambari ...

SmartSense service upgrade completed!
NOTE: It is required to restart Ambari Server for changes to reflect. Please
 restart ambari using 'ambari-server restart'
```

6. Restart the Ambari server:

```
# ambari-server restart
```

7. After the packages are upgraded and the Ambari Service upgrade is complete, log in to Ambari web UI and start SmartSense service.

8. Since SmartSense 1.3.0 includes two new components, Activity Analyzer and Activity Explorer, you need to add these components to Ambari. To do this:

   a. In Ambari web UI, click on **Hosts** and then navigate to the host on which you want to install the components.

   > **Note**
   >
   > Refer to the Activity Analyzer Placement section for advice on which hosts to choose when placing the Activity Analyzer.

   b. Click on **Add** and add Activity Analyzer and then Activity Explorer component.

   c. Start Activity Analyzer and Activity Explorer components by selecting **Start** next to the component name.

9. Ensure that all components are upgraded by triggering a SmartSense analysis capture, and ensure that the capture finishes successfully.

10. If you are upgrading from SmartSense 1.2.x to 1.3, upgrade your HST Gateway:

   • If the HST Gateway is installed on the same node as HST Server or HST Agent, then the HST Gateway will get upgraded along with them. This is true for Ambari managed and non-Ambari managed nodes.

   • If the HST Gateway is a standalone node, perform upgrade steps described in Upgrading SmartSense Gateway.

4.1.3. In-Place Upgrade without Ambari

When using SmartSense 1.1 or 1.2, you can upgrade to SmartSense 1.3 without uninstalling SmartSense:

1. Perform on of the following:

   • On the host running the HST server, stop the process:

   ```
   # hst stop
   ```

2. Download HST 1.3 packages from the **Tools** tab of the Hortonworks support portal (https://support.hortonworks.com).

3. Upgrade binaries on the HST Server and all HST agents. The following steps assume that the HST 1.3 package has been downloaded from the **Tools** tab of the Hortonworks support portal and is available on **all** nodes in the cluster.

   > **Note**
   >
   > The following commands must be run on **all** nodes in the cluster to ensure the entire cluster is upgraded:

   • RHEL, CentOS, or SLES

   ```
   rpm -Uvh smartsense-hst-HST_VERSION.rpm
   ```

   • Ubuntu or Debian

   ```
   dpkg -i smartsense-hst-$HST_VERSION.deb
   ```

4. After the packages are upgraded restart SmartSense:

   • Start HST manually:

   ```
   # hst start
   ```

5. Ensure that all components are upgraded by triggering a SmartSense analysis capture, and ensure that the capture finishes successfully.

6. If you are upgrading from SmartSense 1.2.x to 1.3, upgrade your HST Gateway:

   • If the HST Gateway is installed on the same node as HST Server or HST Agent, then the HST Gateway will get upgraded along with them. This is true for Ambari managed and non-Ambari managed nodes.

   • If the HST Gateway is a standalone node, perform upgrade steps described in Upgrading SmartSense Gateway.

4.1.4. Upgrading SmartSense Gateway

If you are upgrading from SmartSense 1.2 to 1.3 and the HST Gateway is a standalone node, perform the following upgrade steps:

> **Note**
>
> If you using earlier versions of SmartSense with a SmartSense 1.3.0 Gateway, in order to retain backwards compatibility with the older SmartSense versions using the gateway, ensure that only SFTP is used as the smartsense.upload.provider.type in the gateway configuration.

1. On the host running the HST Gateway, stop the hst-gateway process:

```
hst gateway stop
```

2. Upgrade binaries on that node:

   • RHEL, CentOS, or SLES

   ```
   rpm -Uvh smartsense-hst-$HST_VERSION.x86_64.rpm
   ```

   • Ubuntu or Debian

   ```
   dpkg -i smartsense-hst-$HST_VERSION.deb
   ```

3. Start the hst-gateway process:

```
hst gateway start
```

4.2. Upgrading from HST 1.0.x to SmartSense 1.3 in a non-Ambari Environment

To upgrade from HST 1.0.x to SmartSense 1.3, 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
```

Proceed with the installation documented in the Installing SmartSense without Ambari section.

5. SmartSense Ports and Traffic Flow

When deploying SmartSense in an enterprise environment, you must carefully plan your network architecture. SmartSense functionality relies on how multiple internal clusters create bundles and then send them through a central SmartSense Gateway to the hosted Hortonworks environment for analysis or to Hortonworks Support to troubleshoot support cases.

The following graphic illustrates how data traffic flows along various communication channels between cluster nodes and SmartSense ports:
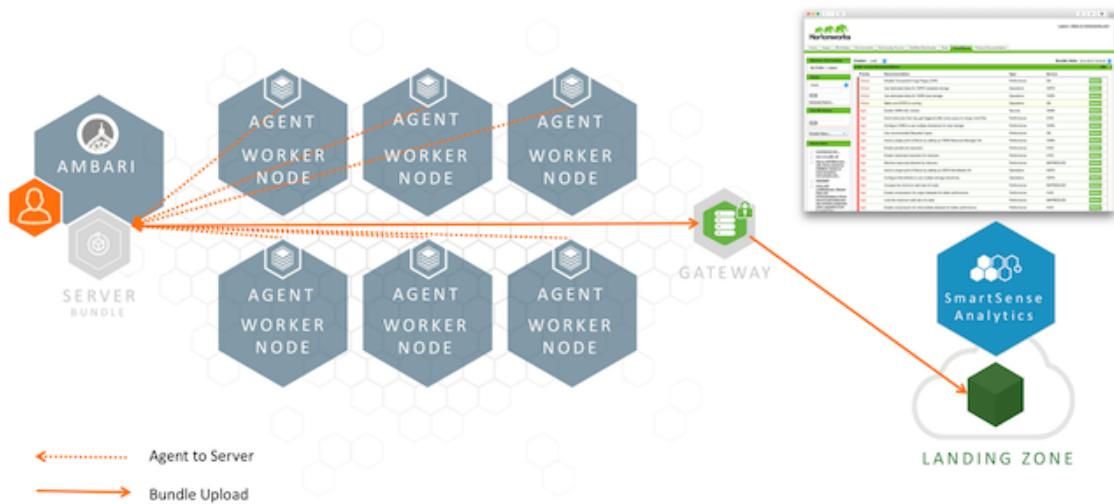
The communication channels illustrated here are described in the following sections:

- User Interface or Ambari View to HST Server [25]

- HST Agent to HST Server [25]

- HST Server to SmartSense Gateway [26]

- SmartSense Gateway to Hortonworks [26]

5.1. User Interface or Ambari View to HST Server

When using SmartSense without Ambari, users access the web UI directly, whereas when using Ambari, they use Ambari View to communicate with the server.

Source Component	Destination Component	Destination Port	Purpose
User interface or Ambari View	HST server	tcp/9000	Web UI communication
Zeppelin Activity Explorer UI	Zeppelin server	tcp/9060	Web UI communication

5.2. HST Agent to HST Server

All communication between the HST server and HST agents is initiated by the agent, using the following ports:

Source Component	Destination Component	Destination Port	Transport Security	Purpose
HST Agent	HST Server	tcp/9440	One-way SSL	Agent registration
HST Agent	HST Server	tcp/9441	Two-way SSL	Anonymized bundle transfer

HST agents register themselves with the HST server, and when invoked to capture data, use the same port to securely transmit captured data back to the HST server.

5.3. HST Server to SmartSense Gateway

All communication between the HST server and the SmartSense Gateway is initiated by the server, using the following ports:

Source Component	Destination Component	Destination Port	Transport Security	Purpose
HST Server	SmartSense Gateway	tcp/9450	One-way SSL	HST server Registration
HST Server	SmartSense Gateway	tcp/9451	Two-way SSL	Encrypted bundle transfer

HST servers register themselves with the SmartSense Gateway using the two-way SSL registration port (tcp/9451), and when bundle capture is complete, this port is used to securely stream the bundle file to the SmartSense Gateway.

5.4. SmartSense Gateway to Hortonworks

Hortonworks does not initiate communications to the SmartSense Gateway, all communication is initiated by the SmartSense Gateway to Hortonworks. For this interaction, the following ports are used:

> **Note**
>
> Using an SFTP-based gateway is deprecated, effective end of April 2018. If you are using SFTP-based gateway you should upgrade to HTTPS-based gateway.

Source Component	Destination Component	Destination Port	Purpose
Gateway	Hortonworks	tcp/2222	SFTP (Deprecated)
Gateway	Hortonworks	tcp/443	HTTPS bundle upload

As bundles are captured, the HST server uses the two-way SSL communication channel to securely stream the bundle file to the SmartSense Gateway. After this process starts, the SmartSense Gateway opens up a secure connection to Hortonworks, using the SFTP port to upload the bundle.

You have two options when configuring the communication between the SmartSense Gateway and Hortonworks:

• Allow firewall access from the Gateway to a CNAME using port 2222 or 443.

  The Hortonworks SFTP and HTTPS servers utilize Elastic Load Balancing from Amazon Web Services. The CNAME is recommended as the number of instances, and IPs of

instances used by the load balancer are fluid. Using the CNAME provides the greatest availability.

• Allow firewall access from the Gateway to a pair of static IPs using port 2222 or 443.

These IPs do not change, and they use "round-robin" DNS for load balancing. This is the least preferred option, because instance availability is not quickly updated in DNS.

Details about both options are available at the following location: https://support.hortonworks.com/s/article/SmartSense-Gateway-setup

6. SmartSense Installation Troubleshooting

This section includes steps for troubleshooting issues that you might encounter during SmartSense installation.

6.1. SmartSense SSL Troubleshooting

SmartSense components use SSL for protecting communications between the HST server and agents, and between the HST server and SmartSense Gateway. If installation issues arise, you can reset these SSL certificates.

HST Server

1. To reset the HST server SSL certificate database, which forces all HST agents to regenerate their certificates, use the **hst reset** command:

```
# hst reset
Resetting SmartSense Server will remove server and all registered agent
 certificates and reset the certificate database. Do you want to continue? [y/
n] (default: n): y
SmartSense Server is currently running and needs to be stopped in order to
 reset. Do you want to stop the SmartSense Server? [y/n] (default: n): y
SmartSense Server stopped
SmartSense Server reset completed.
Do you want to restart SmartSense Server? [y/n] (default: y): y
Server PID at: /var/run/hst/hst-server.pid
Server out at: /var/log/hst/hst-server.out
Server log at: /var/log/hst/hst-serer.log
Waiting for server start . . . . . . .
```

2. Next, you must *manually* reset each individual HST agent after running this command. For instructions on how to reset the agents, see the following **HST Agent** section.

HST Agent

Perform these steps in the following cases:

• An individual agent is having issues related to SSL when communicating with the HST server.

• You have just reset the HST server SSL certificate database (see the **HST Server** section above). In this case, you must perform these steps on each individual HST agent.

1. Use the **hst reset-agent** command to remove all certificates registered with the HST server for the specific agent.

2. Next, run **hst setup-agent -q** to register the agent with the server and download new certificates.

```
# hst reset-agent
Resetting SmartSense Agent will remove all certificates registered with
 SmartSense server.  Do you want to continue? [y/n] (default: n): y
SmartSense Agent reset completed.
# hst setup-agent -q
```

SmartSense Gateway

If HST servers are having issues related to SSL when communicating with the SmartSense Gateway, you can use the **hst gateway reset** to remove all HST server certificates registered with the specific gateway.

From the SmartSense Gateway, you can execute the following process:

```
# hst gateway reset
Resetting SmartSense Gateway will remove all certificates and reset the
 certificate database.  Do you want to continue? [y/n] (default: n): y
SmartSense Gateway stopped
SmartSense Gateway reset completed.
Gateway has to be started to create new certificates.  Do you want to start
 the Gateway? [y/n] (default: y): y
SmartSense Gateway PID at: /var/run/hst/hst-gateway.pid
SmartSense Gateway out at: /var/log/hst/hst-gateway.out
SmartSense Gateway log at: /var/log/hst/hst-gateway.log
Waiting for Gateway start . . . . . . . . . .
SmartSense Gateway started.
```

6.2. Reporting Issues

If you have encountered a functional issue or observed a security issue, you can raise a support ticket in the Hortonworks Support Portal (http://support.hortonworks.com). To open a new support case, navigate to the **Cases** tab and click **Create New Case**. During case creation choose **Product Component: SmartSense**.