

Hortonworks Data Platform

SmartSense Administration Guide

(December 16, 2015)

Hortonworks Data Platform: SmartSense Administration Guide

Copyright © 2012-2015 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.



Except where otherwise noted, this document is licensed under
Creative Commons Attribution ShareAlike 4.0 License.
<http://creativecommons.org/licenses/by-sa/4.0/legalcode>

Table of Contents

1. SmartSense Administration Guide	1
1.1. Document Navigation	1
1.2. SmartSense Installation	1
1.2.1. Meet Minimum System Requirements	1
1.2.2. SmartSense Architecture Overview	3
1.2.3. What's Included In A Bundle	4
1.2.4. Downloading SmartSense Binaries	5
1.2.5. Using SmartSense With Ambari	5
1.2.6. Using SmartSense in a Non-Ambari Environment	10
1.2.7. Manually Uploading Bundles to Hortonworks	14
1.3. SmartSense Gateway Installation	15
1.3.1. SmartSense Gateway Placement	15
1.3.2. Installing SmartSense Gateway	15
1.3.3. Integrating with Ambari Managed SmartSense	16
1.3.4. Integrating with Non-Ambari Managed SmartSense	17
1.3.5. Using the SmartSense Gateway to Automatically Upload Bundles	17
1.3.6. Uninstalling SmartSense Gateway	18
1.4. SmartSense Upgrade Scenarios	18
1.4.1. In-Place Upgrade	18
1.5. Appendix	21
1.5.1. Changing Capture Levels	21
1.5.2. Configure Data Anonymization Rules	21
1.5.3. Change Server/Agent Configurations in a Non-Ambari Environment	23
1.5.4. What's New in this Release	23
1.5.5. Known Issues	23
1.5.6. SmartSense Ports & Traffic Flow	25

List of Tables

- 1.1. 16
- 1.2. 18

1. SmartSense Administration Guide

This document provides you with the latest information about the SmartSense 1.2.0 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 \[4\]](#)
- [Meet Minimum System Requirements \[1\]](#)
- [Using SmartSense With Ambari \[5\]](#)
- [Using SmartSense in a Non-Ambari Environment \[10\]](#)
- [What's New in this Release \[23\]](#)
- [Known Issues \[23\]](#)
- [Manually Uploading Bundles to Hortonworks \[14\]](#)
- [SmartSense Gateway Installation \[15\]](#)
- [SmartSense Ports & Traffic Flow \[25\]](#)

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 \[2\]](#)
- [Software Requirements \[2\]](#)

- [JDK Requirements \[2\]](#)
- [Browser Requirements \[3\]](#)
- [Ambari Requirements \[3\]](#)

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
- Debian 7
- 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
- Ubuntu 14

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
- Python2 (2.6+)

1.2.1.3. 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

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

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. Ambari integration is certified with the following versions:

- Apache Ambari 1.7
- Apache Ambari 2.x

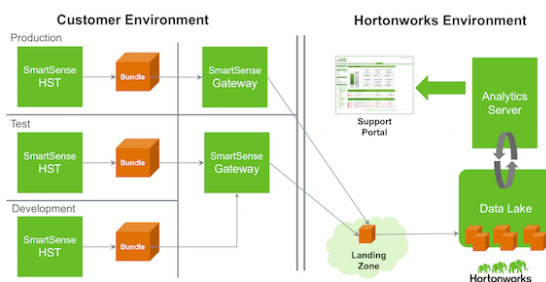
1.2.2. SmartSense Architecture Overview

The Hortonworks SmartSense Tool (HST) is used to collect cluster diagnostic data, both for assisting with support case troubleshooting, and for SmartSense analysis. HST 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.

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. In order to provide the most complete picture of cluster utilization, HST

agents must be installed on every node in the cluster. 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.

The process of uploading bundles depends on how HST has been deployed. For many customers, outbound internet access from the HST server is not readily available. For those users, the HST Gateway has been created to simplify the process of uploading bundles to Hortonworks. The Gateway allows customers with complex network environments to deploy a single Gateway that supports multiple internal HST Server deployments. With this deployment, direct outbound internet access from the HST server is not required for uploading bundles, only access from the HST Server to the Gateway. The Gateway is the only component that requires outbound internet access and will take on the responsibility of uploading all bundles to Hortonworks Support, or the SmartSense SFTP environment for SmartSense Analysis.



This architecture is optional, and customers can continue to upload bundles manually if required.

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 are 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 is available for download from the "Tools" tab of the Hortonworks support portal (<https://support.hortonworks.com>) for non-Ambari and Ambari 1.7-2.1.2 installations. For Ambari 2.2.0 and above installations, SmartSense is automatically included and requires no additional download.



Note

During installation you will be asked for your SmartSense ID and Account Name. These fields are used to associate the cluster diagnostic information with your account in the Hortonworks support portal. This information is available in the support portal's **Tools** tab.

1.2.5. Using SmartSense With Ambari

SmartSense can be easily "plugged in" to Ambari 1.7, 2.0.x, and 2.1.0-2.1.2 making the installation and management more streamlined and familiar to administrators. SmartSense is automatically included in Ambari 2.2.0 and above. 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 will 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.

This information is available in support portal under the **Tools** tab.



Note

It's important to mention that SmartSense will **never** make any configuration changes to your cluster. Any configuration changes that are noticed are caused by the Ambari Stack Advisor. After the Add Service process has completed, review any configuration changes and revert them if necessary before restarting any HDP services.

1. Install the SmartSense Package on the Ambari Server host if using Ambari 1.7, 2.0, or 2.1.0-2.1.2.

For Ambari 2.2.0 and above, proceed to step 3.

- **RHEL/ CentOS / SLES:**

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

- **Debian/Ubuntu:**

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

If using the non-root agent capabilities of Ambari 2.0-2.1.2, please ensure 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 *}}`.

2. The SmartSense service needs to be added to Ambari, and to do so, run the command below. It will ask for the full path of the previously downloaded RPM or DEB package e.g.: `/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'
```

Make sure the Ambari Server has been restarted before continuing with the next step.

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 the **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 use 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 1GB 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 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. This directory should have at least 5GB of free space.

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

At this point, the Ambari Stack Advisor will assess your cluster configuration and may pop-up if any configuration warnings are found. **Please note that this is not related to SmartSense and is simply what Ambari does upon adding any service. SmartSense will never make any configuration changes to your cluster. Any configuration changes that are noticed should be reverted.** If you have a kerberized cluster, you may be prompted for the KDC admin credentials during this step as well. No additional kerberos principals or keytabs are required to use SmartSense.



Note


With Ambari 1.7, an issue with the Ambari Stack Advisor will cause changes be made to your HDP service configuration during the Add Service process. It is critically important to revert any configuration changes made to your

services after the Add Service process has completed before they are restarted.

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** or **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, or will be automatically uploaded to the SmartSense Gateway if configured.

SmartSense Analysis Bundles: These bundles include configuration, and metrics only for all services deployed and all hosts in the cluster. These bundles are anonymized, and encrypted by default and used to produce recommendations to improve cluster performance, operations, and security.

Support Case Troubleshooting Bundles: These bundles include configuration, metrics, and logs for selected services and hosts and used to aid in troubleshooting support cases with Hortonworks Support Engineers.

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 captured and their status. If data is still being captured, the UI will automatically update itself with the capture progress until completed. Once the bundle is in a completed state, it is ready for download, or if a SmartSense Gateway is configured, the bundle will be automatically uploaded to Hortonworks.

Completed bundles can either be manually downloaded and uploaded to Support for Support Case Troubleshooting, or to the SmartSense Environment for SmartSense Analysis, or this process can be automated and scheduled by using the SmartSense Gateway. When using the SmartSense Gateway, all bundles are uploaded to Hortonworks. Support Case Troubleshooting bundles, once received, will trigger a case notification once the bundle has been successfully received. This case notification uses the case number provided during the capture initiation process.

- [Manually Uploading Bundles to Hortonworks \[14\]](#)
- [Using the SmartSense Gateway to Automatically Upload Bundles \[17\]](#)

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. Based on your environment:
 - For Ambari 2.1, navigate to the **Data Capture** section.
 - For Ambari 1.7 and 2.0, expand **Advanced anonymization-rules**.
4. Add the new anonymization rule (or change existing) by following the 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. If the SmartSense Gateway is deployed on any existing HST Server or HST Agent node, ensure that it is stopped. (Optionally, you can run the command below to stop all Ambari-managed SmartSense components).


```
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. *Please ensure that all 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 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 the 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
```

1.2.6. 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. This information is available in support portal under the **Tools** tab.

- **RHEL / CentOS / SLES:**

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

- **Ubuntu / Debian:**

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

1.2.6.1.1. HST Server Placement

One node in the cluster must be designated as the HST server, as this component consolidates all of the HST agents' collected data together into a single downloadable file, referred to as a "bundle". For this functionality, the HST Server must be set up 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. For a full list of ports and a data flow diagram, refer to this section: [SmartSense Ports & Traffic Flow](#).

It is preferred, but not required, that when installed outside of Ambari the HST Server node has 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 italics. 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: 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
```



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.

The HST server uses a storage directory (default: `/var/lib/smartsense/hst-server/data`) to store all collected bundles. The minimum recommended size of that directory is 1GB. 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. 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 be used. 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 the 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} {optional: 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:
AMS           : Collect data for Ambari metrics issue
Ambari        : Collect data for Ambari issue
Falcon        : Collect data for Falcon issue
Ganglia       : Collect data for Ganglia issue
HBase         : Collect data for HBase issue
HCatalog      : Collect data for HCatalog issue
HDFS          : Collect data for HDFS issue
Hive          : Collect data for Hive issue
Kafka         : Collect data for Kafka issue
```



```
Knox      : Collect data for Knox issue
MR        : Collect data for MapReduce issue
Nagios    : Collect data for Nagios issue
Oozie     : Collect data for Oozie issue
Pig       : Collect data for Pig issue
Ranger    : Collect data for Ranger issue
Spark     : Collect data for Spark issue
Sqoop     : Collect data for Sqoop issue
Storm     : Collect data for Storm issue
Tez       : Collect data for Tez issue
YARN      : Collect data for YARN issue
ZK        : Collect data for ZooKeeper issue
```

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

Support Case Troubleshooting Capture

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
```

SmartSense Analysis Capture

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

```
# hst capture all 0
```

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 a bundle has been captured, it can be downloaded, or if using the SmartSense Gateway, can be automatically uploaded. For more information on the Gateway see

the [SmartSense Gateway Installation](#). In the event that a Gateway is not configured, the bundle must be manually uploaded to Hortonworks using SFTP. The connectivity details for the SmartSense SFTP environment are available in this Knowledge Base Article: https://hortonworks.my.salesforce.com/articles/en_US/How_To/Uploading-SmartSense-Bundles

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 1.2.0 in a Non-Ambari Environment

To upgrade from HST 1.0.x to SmartSense 1.2.0, 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 1.2.0 in a Non-Ambari Environment

To uninstall SmartSense, run following commands on all hosts:

1. Remove packages:

- **RHEL / CentOS / SLES:**

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

- **Ubuntu / Debian:**

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

2. Remove data generated by SmartSense:

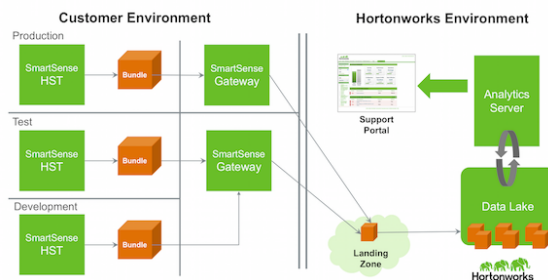
```
# rm -rf /usr/hdp/share/hst
# rm -rf /etc/hst
# rm -rf /var/log/hst
# rm -rf /var/lib/smartsense
```

1.2.7. Manually Uploading Bundles to Hortonworks

Once a bundle has been captured, it can be downloaded, or if using the SmartSense Gateway, can be automatically uploaded. For more information on the Gateway see the [SmartSense Gateway Installation](#). In the event that a Gateway is not configured, the bundle must be manually uploaded to Hortonworks using SFTP. The connectivity details for the SmartSense SFTP environment are available in this Knowledge Base Article: https://hortonworks.my.salesforce.com/articles/en_US/How_To/Uploading-SmartSense-Bundles.

1.3. SmartSense Gateway Installation

For many SmartSense users, outbound internet access from the HST server is not readily available. For those users, the HST Gateway has been created to simplify the process of uploading bundles to Hortonworks. The Gateway allows customers with complex network environments to deploy a single Gateway that supports multiple internal HST Server deployments. With this deployment, direct outbound internet access from the HST server is not required for uploading bundles, only access from the HST Server to the Gateway. The Gateway is the only component that requires outbound internet access and will take on the responsibility of uploading all bundles to Hortonworks Support, or the SmartSense SFTP environment for SmartSense Analysis.



The following section will outline how the Gateway is deployed and used with HST.

1.3.1. SmartSense Gateway Placement

The Gateway is typically deployed on a host in a network zone that has outbound internet access and inbound access from the 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's `hst-gateway.ini` file.

Outbound connections from the 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 the following section: [SmartSense Ports & Traffic Flow](#).

1.3.2. Installing SmartSense Gateway

The SmartSense Gateway is not managed by Ambari and must be installed manually. The SmartSense Gateway is included in the same `smartsense-hst- $\$HST_VERSION$` package used for the HST Server and HST Agent.

1. The SmartSense package is located in the public Ambari 2.2.0 repository for those using Ambari 2.2.x, or on the **Tools** tab of the Hortonworks Support Portal for non-Ambari, and Ambari 1.7 and 2.1.x installations. Install the SmartSense package on the Gateway host using the example invocations below.

- **RHEL / CentOS / SLES:**

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

- **Ubuntu / 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 by setting:

```
[java]
home={path to your JAVA_HOME}
```

- The gateway uses SFTP to upload bundles from internal HST Server environments to the externally hosted SmartSense Analysis environment. To configure the appropriate connectivity details, refer to this Knowledge Base article for the SFTP username, password, host, and port details: <https://support.hortonworks.com/s/article/SmartSense-Gateway-setup>. On startup, the Gateway will attempt to connect to the configured SFTP host, and if the connection fails, the Gateway will not start and log the reason for connectivity failure to the `/var/log/hst/hst-gateway.log` file.

3. Start the Gateway:

```
# hst gateway start
```

4. Integrate Gateway with HST Server:

In order for your SmartSense deployment to use the SmartSense Gateway, you must update the HST server's configuration with the Gateway's FQDN and port. See the following Sections for more details: [Integrating with Ambari Managed SmartSense](#), or [Integrating with Non-Ambari Managed SmartSense](#).

If the Gateway is being deployed on a server that is also hosting an HST Agent, and that HST Agent has been deployed through Ambari, and Ambari is configured for non-root operation, the SmartSense Gateway should be run as the same user that the Ambari Agent is configured to run as. Additionally, the following permissions need to be modified, only if this specific scenario applies. In this case the 'ambari' user has been chosen as an example of the user the Ambari Agent has been configured to run as:

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

1.3.3. Integrating with Ambari Managed SmartSense

When Ambari is used to manage SmartSense, integrating with a Gateway only requires a few configuration changes within Ambari. The steps below outline how to complete the Gateway integration depending on the version of Ambari in use:

Table 1.1.

Ambari 2.1	Ambari 1.7, 2.0	Note
Configuration Tab: Gateway Property: Auto-upload bundles	Expand Configuration Section: Advanced hst-server-conf Property: gateway.enabled	Set this to <code>true</code> to enable the Gateway to be used for automatic upload of bundles once they have been completed.

Ambari 2.1	Ambari 1.7, 2.0	Note
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 on which the Gateway is listening. This is 9451 by default and is configured in the Gateway configuration.

1.3.4. Integrating with Non-Ambari Managed SmartSense

In order to configure a SmartSense deployment that is not using Ambari for management to communicate with an installed Gateway, the HST Server's configuration needs to be updated. The following properties in the HST Server's `/etc/hst/conf/hst-server.ini` file need to be updated to enable communication with the Gateway:

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

- a. Enable the Gateway's automatic upload capability, which means once a bundle has been successfully captured, the HST Server will automatically use 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 on which the Gateway is configured to listen:

```
port=9451
```


2. Restart the HST Server:

```
#hst restart
```

1.3.5. Using the SmartSense Gateway to Automatically Upload Bundles

When enabled, the Gateway will automatically upload completed bundles to Hortonworks when a capture is completed. This includes SmartSense Analysis as well as Support Case Troubleshooting bundles. It's also possible to schedule SmartSense Analysis bundles for capture and auto-upload. The capture schedule can be easily set using the HST Server UI or the SmartSense Ambari View.

1.3.5.1. Updating the Capture Schedule: Ambari View

When deployed with Ambari, the SmartSense View provides a way to easily create, update, pause, resume, and remove the schedules used for automate bundle capture and upload. To view the default capture schedule and update it, access the **SmartSense View**, by clicking on the  icon and choose **SmartSense View**.

The Scheduler settings are available by clicking on the **Schedule** link. Here you can Remove, Pause, Resume and Create new schedules based on the time that is most convenient for you. Four types of schedules are available: Daily, Weekly, Monthly, and Custom.



Note

Please note that scheduler changes take up to 1 hour to go into effect.

1.3.6. Uninstalling SmartSense Gateway

To remove the Gateway, follow the steps below:

1. Ensure the SmartSense Gateway is stopped:

```
# hst gateway stop
```

2. Remove the smartsense-hst package:

- **RHEL / CentOS / SLES:**

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

- **Ubuntu / Debian:**

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

3. Remove logs produced by the Gateway:

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

1.4. SmartSense Upgrade Scenarios

There are different SmartSense upgrade options based on your current SmartSense and target SmartSense versions. This section describes the different upgrade options and their prerequisites.

The following table describes the different upgrade options based on the current and target Stack combinations.

Table 1.2.

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

1.4.1. In-Place Upgrade

When using SmartSense 1.1, the upgrade to 1.2 takes advantage of upgradable packages allowing for an upgrade without uninstallation. The following steps should be addressed before proceeding with the upgrade:

1. • **Ambari Installation:** Log in to Ambari web UI and stop the SmartSense service.

- **Non-Ambari Installation:** On the host running the HST server, stop the process:

```
# hst stop
```

2. Backup the database by creating a backup directory, then copying the database data to it::

```
# mkdir ~/hst_backup  
# cp -r /usr/hdp/share/hst/hst-server/hstDB ~/hst_backup/hstDB
```

3. If using Ambari 2.2.0, HST 1.2 is included in the Ambari repository and does not have to be separately downloaded. If using prior versions of Ambari such as 1.7, 2.0, and 2.1.0-2.1.2, or if this is a non-Ambari installation, the HST 1.2 packages must be downloaded from the **Tools** tab of the Hortonworks support portal (<https://support.hortonworks.com>).
4. Upgrade binaries on the HST Server and all HST agents. This should be done on **every node** in the cluster.
5. Choose Ambari 2.2.0 Installations, or Ambari 1.7, 2.0, and 2.1.0-2.1.2 and non-Ambari Installations:

- **Ambari 2.2.0 Installations:**

The following steps assume that the Ambari 2.2.0 repository is configured on all nodes in the cluster.

- For RHEL/CentOS:

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

In the info output, visually validate that there is an available version containing "1.2.0":

```
yum upgrade smartsense-hst
```

- For SLES:

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

In the info output, visually validate that there is an available version containing "1.2.0":

```
zypper up smartsense-hst
```

- For Ubuntu/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 "1.2.0":

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

- **Ambari 1.7, 2.0, and 2.1.0-2.1.2 and non-Ambari Installations:**

The following steps assume that the HST 1.2 package has been downloaded from the **Tools** tab of the Hortonworks support portal and is available on **all** nodes in the cluster. The following commands must be run on **all** nodes in the cluster to ensure the entire cluster is upgraded:

- For RHEL/CentOS/SLES:

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

- For Ubuntu/Debian:

```
dpkg -i smartsense-hst- $\$$ HST_VERSION.rpm
```

6. Upgrade Ambari Service and View:

If using any version of Ambari, the SmartSense service and view need to be updated with the HST 1.2 service definitions and new view deployment. This can be done by running the command below as the root user from the machine running the Ambari Server. It will ask for the full path of the previously downloaded RPM or DEB package e.g.: `/root/smartsense-hst- $\$$ HST_VERSION.x86_64.rpm`.

```
# hst upgrade-ambari-service
Enter SmartSense distributable path: /root/smartsense-hst- $\$$ HST_VERSION.
x86_64.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'
```

If HTTPS has been enabled for the Ambari Server, a workaround is required before the `hst upgrade-ambari-service` command will run successfully. If, and only if HTTPS has been enabled for the Ambari Server, follow the steps below:

- a. On the Ambari Server host, edit the following file: `/usr/hdp/share/hst/hst-agent/lib/hst_agent/upgrade/UpgradeService120.py`.
- b. On line 306, replace `"http://"` with `"https://"`
- c. Save the file and then re-run the `hst upgrade-ambari-service` command

7. After the packages are upgraded and, if using Ambari, HST upgrade is complete, restart SmartSense:

- **Ambari Installation:** Log in to Ambari web UI and start SmartSense service.

- **Non-Ambari Installation:** Start HST manually:

```
# hst start
```

8. To ensure that all components have been successfully upgraded, trigger a SmartSense Analysis capture. This capture will capture all services on all hosts in the cluster. Ensure that the capture successfully completes.

1.5. Appendix

1.5.1. 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

L3 - 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 set up.

```
capture.level=L2
```

1.5.2. Configure Data Anonymization Rules

Anonymization rules define regular expressions 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 will use for masking. If the `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 the `REPLACE` operation. If not specified, a `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.)

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

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

1.5.4. What's New in this Release

- Ambari Integration:
 - Built into Ambari 2.2 as a native service
 - Still available for Ambari 1.7-2.1.2 as a Plug-In Service downloadable from the **Tools** tab of the Hortonworks Support Portal
 - Support for Non-Root Ambari Agent Installation
- New components:
 - SmartSense Gateway - allows for scheduled bundle capture and auto-upload
- Additionally Supported Operating Systems:
 - Ubuntu 14
 - Debian 7
- New Capabilities:
 - Ability to cancel capture as well as force complete in progress captures
 - Upgradable packaging allowing for easy upgrade path from previous SmartSense installations
 - New alerts in Ambari for bundle captures failures, and Gateway connectivity

1.5.5. Known Issues

Issue	Workaround
After clicking on logout, sometimes doesn't redirect user to login page.	Refresh the browser (or reloading) fixes the issue.
During some captures the bundles status turns into "TIMED OUT".	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.
HST Agent process dies after initiating capture without any warning or entries in the <code>hst-agent.log</code> .	Ensure that the hosts <code>hostname -f</code> outputs the Fully Qualified Domain Name (FQDN) of the host and not the short name. For example <code>revol</code> (short name) instead of <code>revol.hortonworks.local</code> (FQDN). If the short name is used, please re-configure the operating system to ensure that <code>hostname -f</code> outputs the FQDN.
In some Ambari based installation of SmartSense, Ambari Agent operations may be blocked.	Identify the node that's task is not progressing in the Background Operations Ambari modal dialog window. On

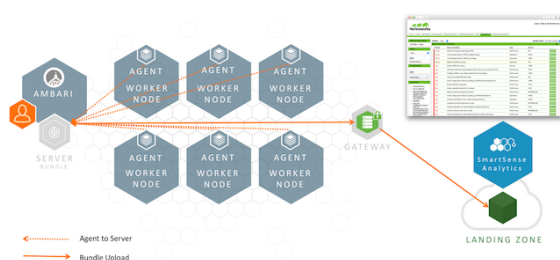
Issue	Workaround
	<p>that host, see if the <code>/var/log/hst/hst-agent.log</code> has any of following messages being continuously printed.</p> <pre>Found stale lock on /var/lib/smartsense/hst-agent/keys/.lock, check running processes for possible malfunction. Failed to acquire lock in x second(s)</pre> <p>If those messages are seen, remove the <code>/var/lib/smartsense/hst-agent/keys/.lock</code> file.</p>
<p>In some environments, SmartSense agent commands fail with an 'SSLError' in the agent logs</p>	<p>This is mainly associated with agent keys becoming out of sync. Deleting the certificates in the folder <code>/var/lib/smartsense/hst-agent/keys/</code> will remove this error.</p>
<p>In some rare situations HST install fails on certain nodes during Ambari's deploy stage.</p>	<p>Use "Retry" option available through the Ambari UI.</p>
<p>In some situations, the stack components for SmartSense are removed and cause the Ambari Server to fail to start with the following error:</p> <pre>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)</pre>	<p>Workaround: Re-run <code>hst add-to-ambari</code> to re-add the stack and view to the Ambari Server instance</p>
<p>SmartSense UI indicates its working but does not display content for a while.</p>	<p>This usually happens when browser UI can not communicate with the server or the session has been timed out. Refresh the browser.</p>
<p>SmartSense view does not appear on some Ambari clusters.</p>	<p>For SmartSense to work appropriately, all hosts in the cluster (including host running Ambari Server) should have Ambari agent running. Please make sure that host running Ambari Server is also part of hosts managed by Ambari. Follow Add host wizard to add Ambari Server host to cluster.</p>
<p>Some CentOS 7 installations will not capture all diagnostic data.</p>	<p>Some CentOS 7 environments (typically minimal CentOS 7) do not have <code>net-tools</code> package installed. Installing it (<code>yum install net-tools</code>) will address this issue. In general please make sure that all prerequisites are met.</p>
<p>Uninstalling or removing SmartSense 1.2.0 Agent might not remove the crontab job automatically.</p>	<p>If any node is getting decommissioned after SmartSense 1.2.0 agent installation, it is required to follow the below steps to manually remove the crontab jobs.</p> <ol style="list-style-type: none"> 1. Login (or ssh) into the node that is getting decommissioned. 2. Run the command <code>"crontab -e"</code> and it will automatically open the editor with list of cron jobs. 3. Remove all lines which have <code>"/usr/hdp/share/hst/bin/hst-scheduled-capture.sh"</code>.

1.5.6. SmartSense Ports & Traffic Flow

When deploying SmartSense into an Enterprise environment, specific network architecture decisions need to be made. SmartSense relies on bundles, produced within the cluster, to be sent to the hosted Hortonworks environment for Analysis or to be used by Hortonworks Support to troubleshoot support cases.

To give more flexibility to customers, the SmartSense Gateway has been introduced. The Gateway can be deployed centrally, and be shared by multiple internal SmartSense clusters. This gateway is in charge of sending bundles, produced within each cluster by the SmartSense Server and Agents, to the hosted Hortonworks environment.

An architectural illustration has been created below to highlight the communication paths, and direction.



The following communication channels from the illustration above are outlined below:

- [User/Ambari View to HST Server \[25\]](#)
- [HST Agent to HST Server \[25\]](#)
- [HST Server to SmartSense Gateway \[26\]](#)
- [SmartSense Gateway to Hortonworks \[26\]](#)

1.5.6.1. User/Ambari View to HST Server

When using SmartSense without Ambari, users will access the Web UI directly, whereas when using Ambari, the Ambari View will communicate with the Server.

Source Component	Destination Component	Destination Port	Purpose
User/Ambari View	HST Server	tcp/9000	Web UI Communication

1.5.6.2. HST Agent to HST Server

HST Servers do not initiate communications to HST Agents, all communication is initiated by the HST Agent to the HST Server. For this interaction, the following ports are used:

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.

1.5.6.3. HST Server to SmartSense Gateway

SmartSense Gateways do not initiate communications to HST Servers, all communication is initiated by the HST Server to the SmartSense Gateway. For this interaction, the following ports are used:

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, and when bundle capture is complete, the Two-Way SSL Communication channel is used to securely stream the bundle file to the SmartSense Gateway.

1.5.6.4. 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:

Source Component	Destination Component	Destination Port	Purpose
Gateway	Hortonworks	tcp/2222	SFTP

Upon bundle capture completion, the HST Server will use the Two-Way SSL communication channel to securely stream the bundle file to the SmartSense Gateway. Once this process has started, the SmartSense Gateway opens up a secure communication to Hortonworks using the SFTP port to upload the bundle.

There are two options to use when configuring the communication between the SmartSense Gateway and Hortonworks:

- Allow firewall access from the Gateway to a CNAME using port 2222. The Hortonworks SFTP servers utilizes Elastic Load Balancing from Amazon Web Services. The CNAME is recommended as the number of instances, and IP's 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 IP's using port 2222. These IP's will not change, and use DNS Round Robin for load balancing. These is the least preferred option as instances availability is not quickly updated in DNS and using the CNAME will provide the greatest availability.

Details for both options are available here: <https://support.hortonworks.com/s/article/SmartSense-Gateway-setup>