SmartSense Configuration Guidelines

Date of Publish: 2018-11-15

http://docs.hortonworks.com

Contents

SmartSense configuration guidelines

The following sections describe configuration properties related to SmartSense components and provide tuning guidelines.

The "Default Value" of a parameter is listed as "(no value)" if by default the parameter is set to an empty value. In order to set the parameter, find it in the Ambari configuration tab listed and set it to a desired value.

The "Default Value" of a parameter is listed as "(unspecified)" if the parameter is unset by default. In order to set the parameter, you must add it as a custom configuration.

HST server

The following configuration properties are available for HST server:

Table 1: HST Server configuration properties

Property Name	Description	Where to Configure	Guidelines
customer.smartsense.id	Your SmartSense ID uniquely identifies your account. You can obtain it from Hortonworks support. This is a mandatory field during SmartSense installation. Type: string Default Value: (unspecified)	Ambari Config: Basic Config File: /etc/hst/conf/hst-server.ini	You can obtain your existing SmartSense ID from the Hortonworks support portal.
customer.account.name	The name of your organization as it is registered with Hortonworks support. This is a mandatory field during SmartSense setup and it is one of the important identifiers for clusters belonging to the same customer.. Type: string Default Value: (unspecified)	Ambari Config: Basic Config File: /etc/hst/conf/hst-server.ini	You must enter the organization name exactly as it is registered in the Hortonworks support portal.
customer.notification.email	Email address used to send bundle upload and recommendation availability notifications. This is a mandatory field during SmartSense setup. Type: string Default Value: (unspecified)	Ambari Config: Basic Config File: /etc/hst/conf/hst-server.ini	Check your junk mailbox in case you do not receive notifications.
customer.enable.flex.subscription	Enables flex subscription for the cluster. Type: boolean Default Value: false	Ambari Config: Basic Config File: /etc/hst/conf/hst-server.ini	Enable only if you have a valid flex subscription ID obtained from Hortonworks support.
customer.flex.subscription.id	Your flex subscription ID obtained from Hortonworks support. Flex subscription offers flexible support subscription. Type: string Default Value: (unspecified)	Ambari Config: Basic Config File: /etc/hst/conf/hst-server.ini	Contact Hortonworks support to obtain a flex subscription ID. When passing the ID, you must also enable flex subscription.

Property Name	Description	Where to Configure	Guidelines
server.storage.dir	Directory used by HST server for storing bundles. Type: string Default Value: /var/lib/smartsense/hst-server/data	Ambari Config: Basic Config File: /etc/hst/conf/hst-server.ini	Use a non-root partition for hosting this directory. For reliable operations, we recommend that you have at least 10GB of free space on that partition.
server.tmp.dir	Directory used by HST server for temporary operations. Type: string Default Value: /var/lib/smartsense/hst-server/tmp	Ambari Config: Basic Config File: /etc/hst/conf/hst-server.ini	Use a non-root partition for hosting this directory. For reliable operations, we recommend that you have at least 10GB of free space on that partition.
server.port	Port to access the HST server web interface and API. Type: int Default Value: 9000	Ambari Config: Operations Config File: /etc/hst/conf/hst-server.ini	This port is internally used for HST operations. Change only if port 9000 is already in use or cannot be unblocked, or if SSL needs a different port. This has no impact on SmartSense Ambari View.
server.max.heap	Maximum heap size (in MB) allocated for the HST server process. Type: int Default Value: 2048	Ambari Config: Advanced > Advanced hst-server-conf Config File: /etc/hst/conf/hst-server.ini	Usually 2048 MB is sufficient for clusters up to 500 nodes. Tuning might help if cluster has more than 500 nodes or if you encounter OOM errors on the server side.
agent.request.processing.timeout	Agent request processing timeout (in seconds). This usually indicates the total time for agent capture to finish. Type: int Default Value: 7200	Ambari Config: Operations Config File: /etc/hst/conf/hst-server.ini	Increase the capture timeout to more than 120 minutes if you are capturing more than 4-5 services or have huge logs for support bundle captures. You may also want to increase this if captures are timing out.
agent.request.syncup.interval	Interval (in seconds) after submitting data collection request in which all the data collections requests from various agents are treated as part of same bundle. In other words, this determines the maximum time for any agent to sync back with server on capture request. If multiple agents join data collection process within this interval, they will be treated as part of same bundle. If any agent joins data collection after this interval, it will be treated as another bundle. Type: int Default Value: 180	Ambari Config: Operations Config File: /etc/hst/conf/hst-server.ini	Default value is suitable for most clusters. In cases where Ambari server and agent requests are slow and SmartSense bundle collection shows unreported agents in every bundle collection, increasing this interval may help.
client.threadpool.size.max	Server thread pool size to handle bundle requests. Type: int Default Value: 40	Ambari Config: Operations Config File: /etc/hst/conf/hst-server.ini	Default value is suitable for most clusters Consider increasing this property if you see multiple agent upload requests timing out on a large cluster with more than 500 nodes.

Property Name	Description	Where to Configure	Guidelines
gateway.host	Fully qualified domain name of the host where the SmartSense gateway process has been deployed and is running. Type: string Default Value: embedded	Ambari Config: Gateway Config File: /etc/hst/conf/hst-server.ini	Keep the default if your HST server has outbound internet access to reach smartsense.hortonworks.com. Otherwise, set up a separate standalone gateway which has outbound access.
gateway.port	Port on which the SmartSense gateway is listening and through which data is transferred. It is set up with two-way SSL. This port is not applicable for embedded gateway. Type: int Default Value: (no value)	Ambari Config: Gateway Config File: /etc/hst/conf/hst-server.ini	This port is used for internal communication between the gateway and HST server. Change this only if this port is already in use or cannot be unblocked. Note that if you change this port, you must update a similar property in the gateway.
gateway.registration.port	Port which is used by clients to register with the gateway. Data is not transferred through this port. It is set up with one-way SSL. This port is not applicable for embedded gateway. Type: int Default Value: (no value)	Ambari Config: Gateway Config File: /etc/hst/conf/hst-server.ini	This port is used for internal communication between the gateway and HST server. Change only if this port is already in use or cannot be unblocked. Note that if you change this port, you must update a similar property in the Gateway.
hst_log_dir	Directory where SmartSense log files are created. Type: string Default Value: /var/log/hst	Ambari Config: Advanced > Advanced hst-log4j Config File: /etc/hst/conf/log4j.properties	Changing this setting is usually not recommended. If you change it, you must provide read/write/create permissions for this directory to Ambari Agent user.
hst_max_file_size	Maximum size of SmartSense HST log files. Type: int Default Value: 30	Ambari Config: Advanced > Advanced hst-log4j Config File: /etc/hst/conf/log4j.properties	Default value is suitable for most clusters. Check available storage capacity before updating this property.
hst_max_backup_index	Maximum number of HST log files. Type: int Default Value: 10	Ambari Config: Advanced > Advanced hst-log4j Config File: /etc/hst/conf/log4j.properties	Increase this number to keep the record of older logs. Check available storage capacity before updating this property.
java.home	Path to the JAVA home for HST server. Type: string Default Value: (no value)	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	This setting is automatically configured from Ambari env settings and usually there is no reason to change it. We recommend that you use the latest 1.7/1.8 JAVA versions with up-to-date security updates. For more security we recommend that you have unlimited JCE policy installed.

Property Name	Description	Where to Configure	Guidelines
derby.system.home	Home directory path for Apache Derby database used internally by HST server. Type: string Default Value: /var/lib/smartsense/hst-server/hstDB	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	Default value is suitable for most clusters. This property should only be changed during the HST server setup. If you change thus after HST server is already set up, remember to make a backup and move existing data to the new location.
bundle.monitor.interval	Interval (in seconds) determining how often a bundle is checked for completeness. After every interval, data uploaded from agents will be collated into a single bundle. When data from all agents is collected into a bundle, the bundle is marked as completed. Type: int Default Value: 20	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	Default value is suitable for most clusters. On very large clusters (with more than 1000 nodes) if bundle collection causes performance issues with the default configuration, this interval can be increased to one minute to minimize file compressions/decompressions.
bundle.alert.progress.timeout.percent	This percentage of bundle processing for which the bundle is failing will raise an alert. Type: float Default Value: 0.6	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	Default value is suitable for most clusters. If the bundle is failing during capture or processing, you can adjust the percentage of processing for which alert will be issued.
server.cleanup.task.interval	Time in hours to execute server cleanup tasks (clean up stale/cancelled bundle temp data). Type: int Default Value: 1	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	Default value is suitable for all clusters.
security.server.two_way_ssl.port	Port for two-way SSL communication between HST server and HST agents. This port is used internally for HST operations. Type: int Default Value: 9441	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	Change only if port 9441 is already in use or cannot be unblocked. This has no impact on SmartSense Ambari View.
security.server.one_way_ssl.port	Port for one-way SSL communication between HST server and HST agents. This port is usually required during two-way SSL setup. This port is used internally for HST operations. Type: int Default Value: 9440	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	Change only if port 9442 is already in use or cannot be unblocked. This has no impact on SmartSense Ambari View.
security.openssl.digest.algorithms	Permitted algorithms for SSL encryption. Type: string Default Value: sha256,sha384,sha512,sha,sha1,md5	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	Not required to modify as sha256,sha512 are available and provide strong encryption. Change only if there are very specific security requirements that can not be met by sha256/sha512.

Property Name	Description	Where to Configure	Guidelines
server.connection.max.idle.millis	The maximum period in milliseconds that a connection may be idle before it is closed. Type: int Default Value: 900000	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	Update this if you see too many open threads in idle state on the HST server.
security.server.disabled.ciphers	A comma-separated list of disabled ciphers for SSL. Type: string Default Value: (no value)	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	Weaker ciphers are already disabled. Change only if you have very specific security requirements.
security.server.disabled.protocols	A comma-separated list of disabled protocols for SSL. Type: string Default Value: (no value)	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	Weaker ciphers are already disabled. Change only if you have very specific security requirements.
upload.permits	Agents capture data and upload it to the HST server which assembles it together into a single bundle. This property defines the number of concurrent uploads allowed from agent to server. Type: int Default Value: 10	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	This property may need to be increased if agent upload requests are timing out on a cluster with more than 500 nodes.
upload.initiate.timeout	Agents capture data and upload it to the HST server which assembles it together into a single bundle. Upload will fail if not initiated within the timeout window (in seconds) defined in this property. Type: int Default Value: 20	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	This property may need to be increased if agent upload requests are timing out on a cluster with more than 500 nodes.
bundle.keepuploaded	This tells the HST server whether to keep bundles received from agents even after merging. If set to false, the agent bundles are deleted after merging. Type: boolean Default Value: false	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	Set this to TRUE if you have to inspect the agent bundles for debugging purposes. Note that this will require plenty of available disk space.
bundle.purge.enabled	Enables a daemon process to purge old bundles. By default, the daemon process cleans up old bundles to efficiently use the disk space. Type: boolean Default Value: true	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	We recommend not to disable this process as it will require a large amount of additional disk space.
bundle.min.retention.days	Number of days to keep the bundle before soft purging. Bundles will be soft purged after the defined number of retention days: the bundle file will be deleted and the DB entries will be soft deleted. Type: int Default Value: 30	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	Update this if you want to keep bundles for longer time (to keep records) or for shorter time (to reduce storage utilization).

Property Name	Description	Where to Configure	Guidelines
bundle.min.force.purge.retention.days	Number of days to keep the bundle before hard purging. Bundles will be hard purged after the defined number of retention days: the DB entries of bundle data along with associated recommendations will be cleaned up. Type: int Default Value: 90	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	Default value is suitable for all clusters.
bundle.purge.threadpool.size	Thread pool used for purging hundreds of bundles. Type: int Default Value: 1	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	Default value is suitable for all clusters.
bundle.purge.interval	The frequency (in hours) with which to run the purge process. Type: int Default Value: 24	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	The default setting (once per day) is sufficient unless you have tens of bundles created daily.
bundle.validity.days	Bundle validity days for retrieving recommendations. After this number of days, a bundle will no longer be considered for retrieving recommendations. Type: int Default Value: 15	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	We recommend not to increase this beyond default because older bundles might not provide the latest status of the cluster.
recommendation.expiry	Recommendation actions such as "Apply" are not permitted on bundles which are older than this number of days. Type: int Default Value: 30	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	We recommend that you capture a new bundle and get new recommendations instead of referring to older recommendations.
recommendation.history.expiry	Recommendation history actions are not permitted on bundles older than this number of days. Type: int Default Value: 90	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	We recommend that you capture a new bundle and get new recommendations instead of referring to older recommendations. Update this value if you have to refer to earlier actions.
recommendation.auto.download.bundle.expiry	If recommendations are not received, HST server will stop trying to retrieve recommendations after this number of days. Type: int Default Value: 7	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	Change this only if you have a very specific requirement and want to stop requesting for recommendations earlier than after 7 days.
recommendation.auto.download.interval	Interval (in seconds) for retrieving recommendations. By default, recommendations are retrieved every 300 seconds. Type: int Default Value: 300	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	Change this only if you have a very specific requirement. Increase this if you want to reduce the frequency of retry attempts.

Property Name	Description	Where to Configure	Guidelines
recommendation.feedback.push.interval	Interval (in seconds) for submitting customer feedback for recommendations. By default, HST server will submit feedback to Hortonworks every 30 minutes if new feedback is available. Type: int Default Value: 1800	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	Default value is suitable for most clusters. Change this only if you have a very specific requirement.
recommendation.feedback.push.max.entries	The number of feedback entries submitted in one request. By default, HST server submits a batch of 50 feedback entries in one request. Type: int Default Value: 50	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	Default value is suitable for most clusters. Change this only if you have a very specific requirement. Requires tuning only if you submit more than 100 feedback entries on a daily basis.
gateway.enabled	Enables auto upload of bundles after capture. Type: boolean Default Value: true	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	Disable this if you are capturing the bundles for internal review purposes only. We recommend to keep it enabled to receive valuable insights and recommendations for your cluster.
gateway.retry.attempts	Defines how many attempts HST server makes to connect to the SmartSense Gateway. Type: int Default Value: 10	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	Default value is suitable for all clusters.
gateway.retry.interval.increment	The amount of time (in milliseconds) to wait before making a subsequent SmartSense Gateway connection attempt. In other words, this is the wait time between subsequent connection attempts. Type: int Default Value: 5000	Ambari Config: Advanced > Custom hst-server-conf Config File: /etc/hst/conf/hst-server.ini	Default value is suitable for all clusters.

HST agent

The following configuration properties are available for HST agent:

Table 2: HST Agent Configuration Properties

Property Name	Description	Where to Configure	Guidelines
agent.tmp_dir	Temporary directory used by agents to keep local bundles during bundle preparation. Type: string Default Value: /var/lib/smartsense/hst-agent/data/tmp	Ambari Config: Basic Config File: /etc/hst/conf/hst-agent.ini	You must have at least 10GB of free space in this directory. This should be set to a different location than the server tmp directory.

Property Name	Description	Where to Configure	Guidelines
security.anonymization.max.heap	The maximum heap allocated (in MB) on every agent for anonymization. Type: int Default Value: 2048	Ambari Config: Advanced > Custom hst-agent-conf Config File: /etc/hst/conf/hst-agent.ini	If you experience out of memory exceptions during the anonymization process, increase the heap size gradually depending on availability.
agent.loglevel	Provides ability to change the hst-agent logging level. Possible values are: INFO, DEBUG, WARNING, ERROR, CRITICAL. Type: string Default Value: INFO	Ambari Config: Advanced > Custom hst-agent-conf Config File: /etc/hst/conf/hst-agent.ini	To debug issues on the agent, set this to DEBUG.
bundle.logs_to_capture	Patterns of log files to be captured. Type: string Default Value: (.*).log$,(.*).out$	Ambari Config: Data Capture Config File: /etc/hst/conf/hst-agent.ini	Be careful when capturing more log files as they may turn out to be large and require extra space on the HST server.
server.url_port	Port for one-way SSL communication between HST server and HST agents. This port is usually required during two-way SSL setup. Type: Default Value: 9440	Ambari Config: Advanced > Custom hst-agent-conf Config File: /etc/hst/conf/hst-agent.ini	This should be modified in sync with similar property in HST server configurations.
server.secured_url_port	Port for two-way SSL communication between HST server and HST agents. Type: Default Value: 9441	Ambari Config: Advanced > Custom hst-agent-conf Config File: /etc/hst/conf/hst-agent.ini	This should be modified in sync with similar property in HST server configurations.
server.two_way_ssl	Enables two-way SSL for communication between HST server and HST agents. Type: boolean Default Value: true	Ambari Config: Advanced > Custom hst-agent-conf Config File: /etc/hst/conf/hst-agent.ini	We recommend not to change this unless you have a very specific requirement.
server.connection_retry_count	Number of times to retry to connect to server in case of connection failures and timeouts. Type: int Default Value: 100	Ambari Config: Operations Config File: /etc/hst/conf/hst-agent.ini	In many cases, the default value (100 retry attempts) is often more than needed. Reduce it if retry connection attempts are keeping the system busy.
server.connection_retry_interval	Defines the interval (in seconds) between retries. Type: int Default Value: 10	Ambari Config: Operations Config File: /etc/hst/conf/hst-agent.ini	Default value is suitable for all clusters.

Property Name	Description	Where to Configure	Guidelines
java.home	Path to the JAVA home for HST agents. Type: string Default Value: (no value)	Ambari Config: Advanced > Custom hst-agent-conf Config File: /etc/hst/conf/hst-agent.ini	This setting is automatically configured from Ambari env settings and usually there is no reason to change it. We recommend that you use the latest 1.7/1.8 JAVA versions with up-to-date security updates. For more security we also recommend that you have unlimited JCE policy installed.
command.heartbeat_interval	The heartbeat interval (in seconds). During agent capture, this heartbeat helps ensure connectivity with HST server and executes certain commands such as cancel capture. Type: int Default Value: 30	Ambari Config: Advanced > Custom hst-agent-conf Config File: /etc/hst/conf/hst-agent.ini	Do not change this unless you experience performance issues.
command.check_commands_retry_count	Number of times to retry check commands. Type: int Default Value: 10	Ambari Config: Advanced > Custom hst-agent-conf Config File: /etc/hst/conf/hst-agent.ini	Default value is suitable for all clusters.
command.check_commands_retry_interval	Interval (in seconds) between retries for check commands. Type: int Default Value: 10	Ambari Config: Advanced > Custom hst-agent-conf Config File: /etc/hst/conf/hst-agent.ini	Default value is suitable for all clusters.
management.updates.dir	Directory to store updates received from HST server. Type: string Default Value: /var/lib/smartsense/hst-agent/updates	Ambari Config: Advanced > Custom hst-agent-conf Config File: /etc/hst/conf/hst-agent.ini	We recommend not to change this unless you have a very specific requirement. If changing this, verify that permissions are set accordingly.
management.patch.auto.apply.enabled	Enable automatically downloading and applying updates received from HST server. Type: boolean Default Value: true	Ambari Config: Advanced > Custom hst-agent-conf Config File: /etc/hst/conf/hst-agent.ini	Disable only if you do not want the HST server to propagate the agent configuration changes to all agents.
bundle.compress_captured_log_locally	By default this is set to false; i.e. the log files are included as they are, without applying compression. Type: boolean Default Value: false	Ambari Config: Advanced > Custom hst-agent-conf Config File: /etc/hst/conf/hst-agent.ini	If capture requests are timing out, set this to true to activate log compression. Note that the compressed files will not be anonymized.
upload.retry_count	Number of times the agent will retry to submit its local bundle to server. Note that this is different from uploading the final bundle to Hortonworks. Type: int Default Value: 100	Ambari Config: Operations Config File: /etc/hst/conf/hst-agent.ini	Default value is suitable for all clusters.

Property Name	Description	Where to Configure	Guidelines
upload.min_retry_interval	Minimum interval (in seconds) between bundle upload retries made by agents. Random value between min_retry_interval and max_retry_interval will be used. For constant value, use retry_interval=x. Type: int Default Value: 15	Ambari Config: Operations Config File: /etc/hst/conf/hst-agent.ini	Default value is suitable for all clusters.
upload.max_retry_interval	Maximum interval (in seconds) between bundle upload retries made by agents. Random value between min_retry_interval and max_retry_interval will be used. For constant value, use retry_interval=x. Type: int Default Value: 120	Ambari Config: Operations Config File: /etc/hst/conf/hst-agent.ini	Default value is suitable for all clusters.
handler.largefiles.size	Minimum file size (in MB) for a file to qualify as a large file. Large files are handled based on action defined in the `handler.largefiles.action` property. Type: int Default Value: 512	Ambari Config: Advanced > Custom hst-agent-conf Config File: /etc/hst/conf/hst-agent.ini	Update this if you have a specific requirement which includes capturing files larger than 512 MB.
handler.largefiles.action	Action to handle a large file. Supported actions are 'allow', 'ignore', 'truncate', 'fail'. Type: string Default Value: truncate	Ambari Config: Advanced > Custom hst-agent-conf Config File: /etc/hst/conf/hst-agent.ini	Configure action to handle large files based on your requirements.

SmartSense gateway

The following configuration properties are available for the SmartSense gateway:

Table 3: SmartSense Gateway Configuration Properties

Property Name	Description	Where to Configure	Guidelines
security.openssl.digest.algorithms	A comma separated list of supported algorithms for SSL. Type: string Default Value: sha256,sha384,sha512,sha,sha1,md5	Ambari Config: N/A Config File: /etc/hst/conf/hst-gateway.ini	Typically it there is no need to modify this as sha256,sha512 are available and provide strong encryption. Change this only if you have a very specific requirement.
security.gateway.cert.name	Use this property if you have to use a custom root certificate authority for SmartSense gateway operations. This file must exist at /var/lib/hst-gateway/keys before gateway is started. Type: string Default Value: ca.crt	Ambari Config: N/A Config File: /etc/hst/conf/hst-gateway.ini	You can set up your own CA to sign certificates for two-way SSL communication between HST server and SmartSense gateway. Modify this property to customize the root CA.

Property Name	Description	Where to Configure	Guidelines
gateway.thread.pool.size	Thread pool for the gateway server's API endpoint. Default is automatically calculated based on CPU cores. Type: int Default Value: (Automatically calculated)	Ambari Config: N/A Config File: /etc/hst/conf/hst-gateway.ini	Since the count is already dynamic, it is usually not required to change it.
gateway.start.validation.enabled	Enables the outbound connectivity check that SmartSense gateway performs during startup. By default, the connectivity check in enabled. Type: boolean Default Value: true	Ambari Config: N/A Config File: /etc/hst/conf/hst-gateway.ini	Set to false if you use an HTTP proxy for gateway and gateway start command fails with "unable to connect" error. There is bug in SmartSense gateway versions earlier than 1.3.2 where the socket connectivity test does not use the intended proxy.
gateway.cache.expiry.hours	The frequency with which the SmartSense gateway cache is refreshed. Gateway caches the outbound connectivity status to report to HST server. By default, this cache is refreshed every two hours and upon gateway startup. Type: int Default Value: 2	Ambari Config: N/A Config File: /etc/hst/conf/hst-gateway.ini	Tweak this property to refresh the outbound connectivity status.
gateway.data.transfer.buffer.size	Buffer size for data transfer via SmartSense gateway. Gateway uses chunked buffers to transfer encrypted data between SmartSense and Hortonworks Datalake. Type: int Default Value: 4096	Ambari Config: N/A Config File: /etc/hst/conf/hst-gateway.ini	Tune this property to effectively use the network bandwidth for communication between HST server and Hortonworks.

Activity analyzer

The following configuration properties are available for activity analyzer:

Table 4: Activity Analyzer Configuration Properties

Property Name	Description	Where to Configure	Guidelines
phoenix.sink.batch.size	Activities are batched for better storage performance. A batch is persisted when either the batch size becomes equal to phoenix.sink.batch.size or activity.status.update.interval.seconds has elapsed. Type: int Default Value: 100	Ambari Config: Activity Analysis Config File: /etc/smartsense-activity/conf/activity.ini	Increasing batch size can lower the load on storage and improve storage performance; however, it can delay the availability of data and increase memory pressure. Reducing batch size can make data available sooner but has negative performance impact on storage layer.

Property Name	Description	Where to Configure	Guidelines
global.activity.processing.parallelism	Number of parallel threads that process each activity type. Controls the threads used for Tez, YARN, MR, and HDFS activity data collection. Type: int Default Value: 8	Ambari Config: Activity Analysis Config File: /etc/smartsense-activity/conf/activity.ini	Reduce the number of threads if you encounter out of memory exceptions.
phoenix.sink.flush.interval.seconds	Time after which data will be flushed to Phoenix. A batch is persisted when either the batch size becomes equal to phoenix.sink.batch.size or activity.status.update.interval.seconds has elapsed. Type: int Default Value: 30	Ambari Config: Activity Analysis Config File: /etc/smartsense-activity/conf/activity.ini	Increase the time to reduce the number of persist operations to Phoenix only if number of records to be batched together is much less than 100.
mr_job.activity.watcher.enabled	Enables automatic activity analysis for MapReduce jobs. Type: boolean Default Value: true	Ambari Config: Activity Analysis Config File: /etc/smartsense-activity/conf/activity.ini	Disable only if you do not want to analyze MapReduce jobs.
mr_job.max.job.size.mb.for.parallel.execution	Maximum size (in bytes) that a MapReduce job can have in order to be executed in parallel. Some large MapReduce jobs may contain thousands of tasks. Such jobs require a lot of memory and they put memory pressure on JVM, especially in multi-threaded execution. Any job with history size larger than specified in this parameter will be executed in synchronized fashion. This may slow the performance down, but will avoid OOM errors. Any job with history file size smaller than specified in this parameter will be executed in parallel. Type: int Default Value: 500	Ambari Config: Activity Analysis Config File: /etc/smartsense-activity/conf/activity.ini	Reduce the parallel execution job size if you encounter OOM errors.
tez_job.activity.watcher.enabled	Enables automatic activity analysis for Tez jobs. Type: boolean Default Value: true	Ambari Config: Activity Analysis Config File: /etc/smartsense-activity/conf/activity.ini	Disable only if you do not want to analyze Tez jobs.
tez_job.tmp.dir	Temporary location where Tez job information is downloaded. Type: string Default Value: /var/lib/smartsense/activity-analyzer/tez/tmp/	Ambari Config: Activity Analysis Config File: /etc/smartsense-activity/conf/activity.ini	You can symlink it to a non-root partition or change it to use a directory in a non-root partition.

Property Name	Description	Where to Configure	Guidelines
yarn_app.activity.watcher.enabled	Enables automatic activity analysis for YARN apps. Type: boolean Default Value: true	Ambari Config: Activity Analysis Config File: /etc/smartsense-activity/conf/activity.ini	Disable only if you do not want to analyze YARN jobs.
hdfs.activity.watcher.enabled	Enables automatic analysis for HDFS files. Type: boolean Default Value: true	Ambari Config: Activity Analysis Config File: /etc/smartsense-activity/conf/activity.ini	Disable only if you do not want to analyze HDFS fsImage.
global.activity.analyzer.user	Defines the user used to read activity data from HDFS and YARN. This user must have read access to all activity data from HDFS/YARN/ATS, and so on. Type: string Default Value: activity_explorer	Ambari Config: Advanced > Advanced activity-conf Config File: /etc/smartsense-activity/conf/activity.ini	Default value is suitable for all clusters.
activity.explorer.user	Defines the user used to read pre-analyzed data. This user does not need access to HDFS and YARN. Type: string Default Value: activity_explorer	Ambari Config: Advanced > Advanced activity-conf Config File: /etc/smartsense-activity/conf/activity.ini	Default value is suitable for all clusters.
analyzer_jvm_opts	Allows you to specify multiple jvm options separated by space. Type: string Default Value: -Xms128m	Ambari Config: Advanced > Advanced activity-env Config File: /etc/smartsense-activity/conf/activity-env.sh	This parameter allows you to add any additional jvm options for executing activity analyzers, for example for GC tuning.
analyzer_jvm_heap	Maximum heap space (in MB) allocated for Activity Analyzer process. Type: int Default Value: 8192	Ambari Config: Advanced > Advanced activity-env Config File: /etc/smartsense-activity/conf/activity-env.sh	Usually 8192 MB is sufficient, but it can be increased if you encounter OOM errors.
activity_log_dir	Directory where activity log files are created. Type: string Default Value: var/log/smartsense-activity	Ambari Config: Advanced > Advanced activity-log4j Config File: /etc/smartsense-activity/conf/log4j.properties	Default value is suitable for most clusters. If you change this directory, you must provide read/write/create permissions on the new directory to activity_analyzer user.
activity_max_file_size	Maximum size (in MB) for SmartSense activity log files. Type: int Default Value: 30	Ambari Config: Advanced > Advanced activity-log4j Config File: /etc/smartsense-activity/conf/log4j.properties	Default value is suitable for most clusters. Check available storage capacity before updating this property.

Property Name	Description	Where to Configure	Guidelines
activity_max_backup_index	Maximum number of SmartSense activity log files. Type: int Default Value: 10	Ambari Config: Advanced > Advanced activity-log4j Config File: /etc/smartsense-activity/conf/log4j.properties	You can increase this number to keep the record of older logs. Check available storage capacity before updating this property.
global.date.format	Format in which dates are converted to strings and sometimes persisted. Type: string Default Value: "YYYY-mm-DD"	Ambari Config: Advanced > Custom activity-analyzer-conf Config File: /etc/smartsense-activity/conf/activity.ini	Default value is suitable for all clusters.
global.activity.status.update.interval.seconds	Interval (in seconds) after which status of processed/failed/in process activities is updated in DB. Type: int Default Value: 30	Ambari Config: Advanced > Custom activity-analyzer-conf Config File: /etc/smartsense-activity/conf/activity.ini	Default value is suitable for all clusters.
activity.batch.interval.seconds	Interval for batching activities. Activities are batched for better storage performance. A batch is persisted when either the batch size becomes equal to phoenix.sink.batch.size or activity.status.update.interval.seconds is elapsed. Type: int Default Value: 60	Ambari Config: Advanced > Custom activity-analyzer-conf Config File: /etc/smartsense-activity/conf/activity.ini	Increasing the batch interval can lower the load on storage and improve storage performance; however, it can also delay the availability of data and increase memory pressure. Reducing the interval size can make data available sooner, but has negative performance impact on storage layer.
activity.watcher.enabled	Enables regular collection of job data for analysis. Type: boolean Default Value: true	Ambari Config: Advanced > Custom activity-analyzer-conf Config File: /etc/smartsense-activity/conf/activity.ini	Disable this only if you want to temporarily turn off data collection.
activity.history.max.back.track.days	The number of days of history to retrieve job information. Type: int Default Value: 7	Ambari Config: Advanced > Custom activity-analyzer-conf Config File: /etc/smartsense-activity/conf/activity.ini	Increase this number if you have to refer to older jobs. Note that older jobs should have data available in AMS. This is used only during first run after installation.
phoenix.setup.continue.on.error	During initial setup, errors in DB setup may occur. This parameter indicates whether to continue if any error occurs. Type: boolean Default Value: false	Ambari Config: Advanced > Custom activity-analyzer-conf Config File: /etc/smartsense-activity/conf/activity.ini	Default value is suitable for all clusters.

Property Name	Description	Where to Configure	Guidelines
phoenix.setup.drop.existing.tables	During initial setup matching tables may be found in the DB (typically from previous install attempts). This parameter determines whether they should be dropped and recreated. By default, the existing entries are kept. Type: boolean Default Value: false	Ambari Config: Advanced > Custom activity-analyzer-conf Config File: /etc/smartsense-activity/conf/activity.ini	Default value is suitable for all clusters.
phoenix.activity.analyzer.jdbc.url	JDBC URL used by activity analyzer to store its data. Type: string Default Value: (no value)	Ambari Config: Advanced > Custom activity-analyzer-conf Config File: /etc/smartsense-activity/conf/activity.ini	Do not change it. It is auto configured based on the cluster setup.
ams.jdbc.url	JDBC URL used by activity analyzer to fetch data from AMS. Type: string Default Value: (no value)	Ambari Config: Advanced > Custom activity-analyzer-conf Config File: /etc/smartsense-activity/conf/activity.ini	Do not change it. It is auto configured based on the cluster setup.
global.store.job.configs	Enables storing job-specific configs in AMS after analysis. Type: boolean Default Value: true	Ambari Config: Advanced > Custom activity-analyzer-conf Config File: /etc/smartsense-activity/conf/activity.ini	Do not disable it. Keeping it on helps in debugging.
global.store.tasks	Enables persisting task-level data in AMS after analysis. Type: boolean Default Value: false	Ambari Config: Advanced > Custom activity-analyzer-conf Config File: /etc/smartsense-activity/conf/activity.ini	Task-level data can be huge and may overwhelm AMS, so keep it disabled unless absolutely needed. If enabling, disable again later.
global.store.task.counters	Enables storing task counter data in the AMS after analysis. Type: boolean Default Value: false	Ambari Config: Advanced > Custom activity-analyzer-conf Config File: /etc/smartsense-activity/conf/activity.ini	All task counters can be huge and may overwhelm AMS, so keep it disabled unless absolutely needed. If enabling, disable again later.
global.activity.fetch.retry.interval.seconds	Interval (in seconds) between retry attempts to fetch the activity details. Type: int Default Value: 5	Ambari Config: Advanced > Custom activity-analyzer-conf Config File: /etc/smartsense-activity/conf/activity.ini	Default value is suitable for all clusters.

Property Name	Description	Where to Configure	Guidelines
global.activity.fetch.retry.attempts	Number of tries to fetch activities before giving up. Type: int Default Value: 5	Ambari Config: Advanced > Custom activity-analyzer-conf Config File: /etc/smartsense-activity/conf/activity.ini	Default value is suitable for all clusters.
global.tmp.dir	Temporary directory used by activity-analyzer for internal purposes. Type: string Default Value: /var/lib/smartsense/activity-analyzer/tmp/	Ambari Config: Advanced > Custom activity-analyzer-conf Config File: /etc/smartsense-activity/conf/activity.ini	We do not recommended to change this unless you have a very specific requirement. If using a different directory than the default, verify that permissions are set accordingly.

Activity explorer

The following configuration properties are available for activity explorer:

Table 5: Activity Explorer configuration properties

Property Name	Description	Where to Configure	Guidelines
users.admin	Password for activity explorer's admin user when using local authentication. Type: string Default Value: (no value)	Ambari Config: Activity Analysis Config File: /etc/zeppelin/conf/shiro.ini	This should be updated only during installation. It requires uninstall and re-install if you have to update the password for admin access.
main.sessionManager	The SessionManager, as its name might imply, manages sessions for all subjects in an application: session creation, deletion, inactivity, validation, and so on. Like other core architectural components in Apache Shiro, the SessionManager is a top-level component maintained by the SecurityManager. The default SecurityManager implementation uses a DefaultSessionManager out of the box. The DefaultSessionManager implementation provides enterprise-grade session management features (such as session validation and orphan cleanup) needed for an application. Type: string Default Value: org.apache.shiro.web.session.mgt.Default WebSessionManager	Ambari Config: Advanced > Advanced activity-zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Shiro documentation.
main.securityManager.sessionManager	The default value applies the value set in main.sessionManager to this property. If needed, you can set this to a value specific for security manager. Type: string Default Value: $sessionManager	Ambari Config: Advanced > Advanced activity-zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Shiro documentation.

Property Name	Description	Where to Configure	Guidelines
securityManager.sessionManager.globalSessionTimeout	Sets the default timeout value (in milliseconds) for all newly created sessions. Changing this property will automatically apply the new value to all sessions. Type: long Default Value: 86400000	Ambari Config: Advanced > Advanced activity-zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Shiro documentation.
zeppelin.server.addr	Binding address for Zeppelin activity explorer. Type: string Default Value: 0.0.0.0	Ambari Config: Advanced > Advanced activity-zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Zeppelin 0.6.2 documentation.
zeppelin.server.port	Port on which Zeppelin UI is available. Type: int Default Value: 9060	Ambari Config: Advanced > Advanced activity-zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Zeppelin 0.6.2 documentation.
zeppelin.server.context.path	Context path of the web application. Type: string Default Value: /	Ambari Config: Advanced > Advanced activity-zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Zeppelin 0.6.2 documentation.
zeppelin.war.tempdir	Location of Jetty temporary directory. Type: string Default Value: /var/lib/smartsense/activity-explorer/webapp	Ambari Config: Advanced > Advanced activity-zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Zeppelin 0.6.2 documentation.
zeppelin.notebook.dir	Path or URI for notebook persist. Type: string Default Value: /var/lib/smartsense/activity-explorer/notebook	Ambari Config: Advanced > Advanced activity-zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Zeppelin 0.6.2 documentation.
zeppelin.notebook.homescreen.hide	When set to true, hides home screen notebook from list. Type: boolean Default Value: false	Ambari Config: Advanced > Advanced activity-zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Zeppelin 0.6.2 documentation.
zeppelin.notebook.storage	Notebook persistence layer implementation. Type: string Default Value: org.apache.zeppelin.notebook.repo.VFSNotebookRepo	Ambari Config: Advanced > Advanced activity-zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Zeppelin 0.6.2 documentation.

Property Name	Description	Where to Configure	Guidelines
zeppelin.interpreter.dir	Interpreter implementation base directory. Type: string Default Value: /usr/hdp/share/hst/activity-explorer/ interpreter	Ambari Config: Advanced > Advanced activity- zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Zeppelin 0.6.2 documentation.
zeppelin.interpreters	A comma separated list of interpreter configurations. First interpreter becomes default. Type: string Default Value: org.apache.zeppelin.phoenix.Phoenix Interpreter	Ambari Config: Advanced > Advanced activity- zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Zeppelin 0.6.2 documentation.
zeppelin.interpreter.connect.timeout	Interpreter process connect timeout in milliseconds. Type: int Default Value: 30000	Ambari Config: Advanced > Advanced activity- zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Zeppelin 0.6.2 documentation.
zeppelin.ssl	Enables using SSL for the servers. Type: boolean Default Value: false	Ambari Config: Advanced > Advanced activity- zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Zeppelin 0.6.2 documentation.
zeppelin.ssl.client.auth	Enables client authentication for SSL connections. Type: boolean Default Value: false	Ambari Config: Advanced > Advanced activity- zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Zeppelin 0.6.2 documentation.
zeppelin.ssl.keystore.path	Path to keystore relative to the activity explorer configuration directory. Type: stringzeppelin.ssl.truststore.type Default Value: /var/lib/smartsense/activity-explorer/ keystore	Ambari Config: Advanced > Advanced activity- zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Zeppelin 0.6.2 documentation.
zeppelin.ssl.keystore.type	The format of the given keystore (for example JKS or PKCS12). Type: string Default Value: JKS	Ambari Config: Advanced > Advanced activity- zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Zeppelin 0.6.2 documentation.
zeppelin.ssl.keystore.password	Keystore password. It can be obfuscated using the Jetty password tool. Type: string Default Value: admin	Ambari Config: Advanced > Advanced activity- zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Zeppelin 0.6.2 documentation.

Property Name	Description	Where to Configure	Guidelines
zeppelin.ssl.key.manager.password	Key Manager password. Defaults to keystore password. It can be obfuscated. Type: string Default Value: admin	Ambari Config: Advanced > Advanced activity-zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Zeppelin 0.6.2 documentation.
zeppelin.ssl.truststore.path	Path to truststore relative to activity explorer configuration directory. Defaults to the keystore path. Type: string Default Value: /var/lib/smartsense/activity-explorer/truststore	Ambari Config: Advanced > Advanced activity-zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Zeppelin 0.6.2 documentation.
zeppelin.ssl.truststore.type	The format of the given truststore (for example JKS or PKCS12). Defaults to the same type as the keystore type. Type: string Default Value: JKS	Ambari Config: Advanced > Advanced activity-zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Zeppelin 0.6.2 documentation.
zeppelin.ssl.truststore.password	Truststore password. Can be obfuscated using the Jetty password tool. Defaults to the keystore password. Type: string Default Value: admin	Ambari Config: Advanced > Advanced activity-zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Zeppelin 0.6.2 documentation.
zeppelin.server.allowed.origins	Allowed sources for REST and WebSocket requests (i.e. http://onehost:8080,http://otherhost.com). If you change from * you are vulnerable to the issue described in ZEPPELIN-173. Type: string Default Value: *	Ambari Config: Advanced > Advanced activity-zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Zeppelin 0.6.2 documentation.
zeppelin.anonymous.allowed	Enables access by anonymous user. Type: boolean Default Value: false	Ambari Config: Advanced > Advanced activity-zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Zeppelin 0.6.2 documentation.
zeppelin.websocket.max.text.message.size	Size in characters of the maximum text message to be received by WebSocket. Type: long Default Value: 1024000	Ambari Config: Advanced > Advanced activity-zeppelin-shiro Config File: /etc/zeppelin/conf/shiro.ini	Refer to Apache Zeppelin 0.6.2 documentation.

Related Information

Apache Shiro documentation

Apache Zeppelin documentation