Signiant FlightUpload Component Guide

Nov 2015

Table of Contents

Introduction

The FlightUpload component performs a secure, accelerated upload of files and/or folders from a Signiant Agent to cloud-based object storage. Currently supported object storage vendors are Amazon S3 and Microsoft Azure.

Functional Description

The FlightUpload component will perform a secure & accelerated upload of all specified content to the specified cloud storage container and will skip any files that are already at the target location. Naturally, the component must be executed on a Signiant Agent which has access to the source data, whether on local storage, NAS or SAN. The Flight Upload component utilizes the Signiant Flight command line interface (CLI).

Flight uploads utilize Signiant's WAN acceleration technology which maximizes transfer throughput to the cloud object storage from the source Signiant agent. Optionally TCP or HTTP based protocols may be used as failover in the event that a connection cannot be made using Signiant's WAN acceleration technology (UDP).

Flight uploads, as with all Signiant transfers, perform checkpoint restarts. So if a transfer is interrupted due to a network disruption or other cause, the transfer will be restarted automatically and will pick up exactly where it left off rather than re-transferring portions of the file(s) that have already been uploaded.

The primary input to the FlightUpload component is a list of files and/or folders to be uploaded to cloud object storage. This list can be in the form of a set of absolute paths of files and/or folders or it can be in the form of a manifest file containing just a list of files to be uploaded.

The FlightUpload component is capable of transferring content to either Amazon S3 or Microsoft Azure. Credentials for accessing the cloud object store can either be specified directly to the component, or can be referenced via a Flight Config ID. Flight Config IDs are generated and managed via the Flight Management Console available at https://manage.signiant.com.

The component has three modes for Folder Structure Preservation to address workflow requirements. The transfers can either preserve the structure of all uploaded folders (including the base folder), can preserve the structure of subfolders only (and not the base folder) or can flatten all content so that no folder structure is transferred to the object container. Note that if the Source Data is specified as a manifest file, the "flattening" mode must be used.

Flight Uploads can be configured to use both a Primary and Backup "Flight Service Entry Point". These define the specific locations or regions that the Flight CLI connects to when performing the upload. It is generally recommended that the Primary Entry Point be at the location that is geographically closest to where the object storage container is, not necessarily at the point that is closest to the source Signiant Agent. This is due to the Signiant acceleration protocol being used between the source agent and the Flight Service – making it virtually irrelevant where in the world the source agent is. However, once the data is received into the Flight Service, the Flight servers must perform a multi-part HTTP transfer to the object storage. This makes it important that the Flight servers chosen for the transfer be as close as possible to the actual object storage container. The Flight Service Backup Entry Point is provided for high-availability, so if in the unlikely event that the primary entry point is unavailable, the backup will be used. Note that the Backup Entry Point can be specified as a geographic region spanning multiple Signiant Flight clusters, providing even further high-availability and fault-tolerance.

While all Flight transfers are already encrypted in transit, the FlightUpload component is also optionally able to encrypt all content uploaded to the cloud. This provides an extra measure of security for sensitive material. Without the knowledge of the encryption key used for a given upload, it is impossible to decrypt the content.

As with standard Signiant transfers, Flight Uploads are able to be throttled so that a given upload job does not consume more than the set level of bandwidth.

Flight Uploads are also able to make use of Signiant Relay Agents, which are sometimes required due to restrictive firewall rules in highly secure enterprise networks.

Prerequisites

The following are prerequisites for performing uploads using the FlightUpload component:

- The Signiant Flight command line interface (CLI) utility must be located in the Signiant Agent's "bin" folder (e.g. /usr/signiant/dds/bin or C:\Program Files\Signiant\Mobilize\bin).
- The necessary UDP and/or TCP network ports must be open between the Signiant agent on which this component is being executed and the cloud-based Signiant Flight Service.

Agent Platforms

The FlightUpload component is supported on all platforms on which the Signiant Flight CLI is supported, which are currently Windows, Linux and Mac.

For more information please see the *Signiant Flight Command Line Interface Guide* available at http://flight.support.signiant.com.

Component Inputs

This section describes the input properties of the FlightUpload component.

Category: Source Options

Name	Required?	Default Value	Description
Target Agents	Yes		The Signiant agent on which the component will be run. This can either be a single agent or a load-balanced agent group.
Source Data	Yes		List of files and/or folders to be uploaded to cloud object storage. This list is in the form of a set of absolute paths of files and/or folders. UNC paths are supported on Windows.
File Manifest	No		A manifest file that is read by the component to determine the actual list of files to be transferred. Absolute paths must be specified. UNC paths are supported. Only one of ***Source Data*** and ***File Manifest*** can be used per workflow.
Manifest Delete On Success	Yes	Yes	Specifies whether the manifest file (if used) should be deleted after a successful transfer.
Delete Source Files After Upload	Yes	No	Specifies whether to delete the files successfully transferred from the source agent following the successful transfer.
Delete Empty Source Folders After Upload	Yes	No	In conjunction with ***Delete Source Files After Upload***, specifies whether to delete empty source folders that remain after source files have been deleted.
Target Working Directory	Yes	%DdsSourceTempDir%	The folder where the component will execute. Generally set to %DdsSourceTempDir%
Target User	Yes	%dds_default_user%	The userid that the component code is running as. Generally set to %dds_default_user%

Category: Target Options

Name	Required?	Default Value	Description
Cloud Vendor	Yes		The Cloud Vendor where the object storage container is located. Current choices are **Amazon S3** and **Microsoft Azure**.
Storage Specification Mechanism	Yes	Flight Storage Config ID	Defines whether the details defining object storage and access credentials are being done via a **Flight Storage Config ID** (managed using https://manage.signiant.com) or via credentials specified in the job itself. If specified as **Cloud Vendor Credentials** then the input properties for ***Amazon S3*** or ***Microsoft Azure*** must be specified, depending on which ***Cloud Vendor*** has been selected.
Flight Storage Config ID	No		The **Flight Storage Config ID**, if this mechanism has been selected as ***Storage Specification Mechanism***.
Folder Structure Preservation	Yes	Preserve Top-Level Folders and Subfolders	Three modes are available for Folder Structure Preservation to address various workflow requirements. The transfer can either preserve the structure of all uploaded folders (including the base folder), it can preserve the structure of subfolders only (and not the base folder) or it can flatten all content so that no folder structure is transferred to the object container. Note that if the Source Data is specified as a manifest file, the "flattening" mode must be used.
Optional Target Subfolder	No		By default, all uploaded content is placed relative to the root of the object storage. If an Optional Target Subfolder is specified, then all content will be placed relative to this subfolder rather than the root. This subfolder will be created if it does not already exist.

Category: Flight Service Options

Name	Required?	Default Value	Description
Flight Service Primary Entry Point	Yes		Flight Uploads can be configured to use both a Primary and Backup "Flight Service Entry Point". These define the specific locations or regions that the Flight CLI connects to when performing the upload. It is generally recommended that the Primary Entry Point be at the location that is geographically closest to where the object storage container is, not necessarily at the point that is closest to the source Signiant Agent. This is due to the Signiant acceleration protocol being used between the source agent and the Flight Service – making it virtually irrelevant where in the world the source agent is. However, once the data is received into the Flight Service, the Flight servers must perform a multi-part HTTP transfer to the object storage. This makes it important that the Flight servers chosen for the transfer be as close as possible to the actual object storage container.
Flight Service Backup Entry Point	Yes		The Flight Service Backup Entry Point is provided for high-availability, so if in the unlikely event that the primary entry point is unavailable, the backup will be used. Note that the Backup Entry Point can be specified as a geographic region spanning multiple Signiant Flight clusters, providing even further high-availability and fault-tolerance.
Flight API Key	Yes		The **Flight API Key** to be used for the job. Flight API keys are managed in the Flight Management Console at https://manage.signiant.com.
Load Balanced	Yes	Yes	Determines if transfers should make use of load balancing when initially connecting to the Flight servers. When enabled, the Flight client will attempt to determine and use the least heavily used server in the Flight cluster for the specified Flight Service Entry Point.

Category: Amazon S3

Name	Required?	Default Value	Description
AWS Access Key	No		If **Amazon S3** has been chosen as ***Cloud Vendor*** and **Cloud Vendor Credentials** is specified as the ***Storage Specification Mechanism***, then the **AWS Access Key** must be specified.
AWS Secret Key	No		If **Amazon S3** has been chosen as ***Cloud Vendor*** and **Cloud Vendor Credentials** is specified as the ***Storage Specification Mechanism***, then the **AWS Secret Key** must be specified. The Secret Key will be masked for security.
S3 Bucket	No		If **Amazon S3** has been chosen as ***Cloud Vendor*** and **Cloud Vendor Credentials** is specified as the ***Storage Specification Mechanism***, then the **S3 Bucket** must be specified.

Category: Microsoft Azure

Name	Required?	Default Value	Description
Azure Storage Account Name	No		If **Microsoft Azure** has been chosen as ***Cloud Vendor*** and **Cloud Vendor Credentials** is specified as the ***Storage Specification Mechanism***, then the **Azure Storage Account Name** must be specified.
Azure Access Key	No		If **Microsoft Azure** has been chosen as ***Cloud Vendor*** and **Cloud Vendor Credentials** is specified as the ***Storage Specification Mechanism***, then the **Azure Access Key** must be specified. The Access Key will be masked for security.
Azure Container	No		If **Microsoft Azure** has been chosen as ***Cloud Vendor*** and **Cloud Vendor Credentials** is specified as the ***Storage Specification Mechanism***, then the **Azure Container** must be specified.

Category: Encryption

Name	Required?	Default Value	Description
Encrypt Uploaded Files	Yes	No	While all Flight transfers are already encrypted in transit, the FlightUpload component is also optionally able to encrypt all content uploaded to the cloud. This provides an extra measure of security for sensitive material. Without the knowledge of the encryption key used for a given upload, it is impossible to decrypt the content.
Encryption Key	No		The encryption key to use when encrypting the uploaded content. The Encryption Key must be a **64-character hex string** (0..9,A-F).
Initialization Vector	No		The initialization vector to use when encrypting the uploaded content. The Initialization Vector must be **a 32-character hex string** (0..9,A-F).

Category: Transport

Name	Required?	Default Value	Description
Protocol	Yes	All	Specifies either a specific protocol (**UDP**, **TCP**, **HTTP**) for the Flight upload, or **All. All** means to try UDP first, then TCP, and finally HTTP.
Bandwidth Throttle	Yes	0	The maximum transfer rate in bytes per second. If zero, the transfer rate is unlimited.
Transport Restart Attempts	Yes	3	The number of restarts to attempt on a transfer failure before the component exits in error. If zero, then no restarts will be attempted. The component will not retry on invalid storage credentials or other conditions under which a retry has no chance of succeeding.
Transport Port Usage	Yes	Range	Set to **Single** or **Range**. Use **Single** to use a single port (49221) for firewall efficiency. For enhanced throughput, set to **Range**. If outbound firewall rules are set, then a port range of (49221-49420) must be added to the firewall.
Relay Agents	No		If the network connection to the Flight Server must be routed via a Signiant relay agent (typically for DMZ navigation), then select one or more relay agents through which the network connection is to be routed. If multiple relay agents are selected, all routes to the Flight Server will be tried, and the quickest to respond will be used.
HTTP Proxy	No		If you have an outbound HTTP proxy, specify the proxy server URL and port. This is required for the component key check. This proxy will not be used for HTTP failover transfers. The specified value should be in the format: ***http://<host>:<port>***
HTTP Port	No	80	Allows specifying the port to be used for HTTP failover transfers, if the default port (80) is not being used by the Flight service.
Create File Delivery Log	Yes	Yes	Specifies whether to create a File Delivery Log for all files successfully transferred. If specified, this log will be viewable via the Signiant Manager under *Administration / Manager / Transfer Logs / log / delivery_logs / <date>*
Transport Log Delete On Success	No	Yes	The transport log contains detailed information regarding the Flight transfer. It can be rather verbose and is of little value if the transfer completes successfully. If the job is being run in DEBUG mode, then this option will be forced to “No”.

Component Outputs

Category: Component

Name	Description
Return Code	Set to zero if all files were transferred correctly. Set to non-zero if any errors were occurred.
Return Message	A short message indicating the overall result of the component.

Category: Transfer Statistics

Name	Description
Transfer Direction	The direction of the transfer – generally used to pass to a FlightReport component.
Transfer Server	The actual Flight server FQDN that was used for the transfer.
Transfer State	The final state of the transfer (e.g. COMPLETED/FAILED).
Transfer Protocol	The protocol used for the transfer. One of UDP, TCP, or HTTP.
Transfer Duration	The duration of the file transfer in seconds.
Transfer Restarts	The number of restarts required for the transfer (normally 0).
Transfer Rate Average	The average network transfer rate from the Signiant agent (Signiant Flight client application) to the Signiant Flight server in bits per second.
Transfer Rate Maximum	The maximum network transfer rate from the Signiant agent (Signiant Flight client application) to the Signiant Flight server in bits per second.
Filesize Average	The average filesize of all files actually transferred.
Filesize Maximum	The maximum filesize of all files actually transferred.
Bytes to Transfer	The combined size of all the source files in bytes.
Bytes Transferred	The actual number of bytes transferred.
Total Files to Transfer	The total number of source files to transfer.
Transferred File Count	The number of files actually transferred.
Skipped File Count	The number of files skipped because they were already at the target.
Failed File Count	The number of files that failed to transfer.

Category: Logs

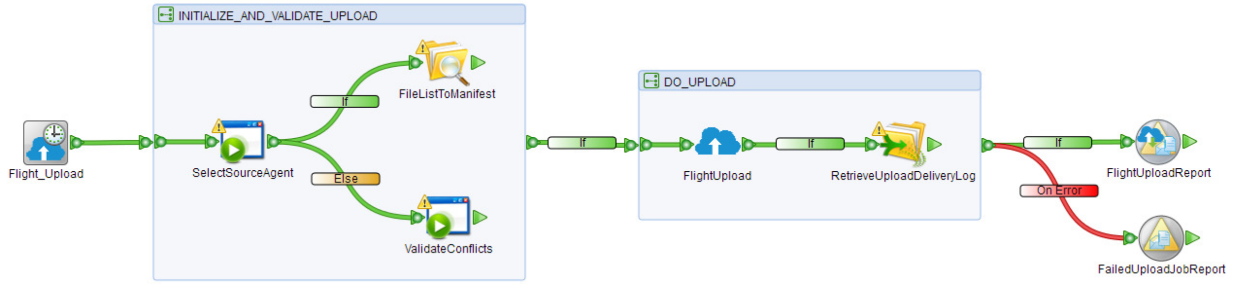
Name	Description
Transport Log Path	Path to Transport Log file. Optionally deleted after a successful transfer. This log is located on the Signiant Agent which performed the Flight transfer.
Delivery Log Path	Path to Delivery Log file, if created. This log is located on the Signiant Manager.

Category: File Lists

Name	Description
Transferred File List	The list of successfully transferred files. These are absolute paths. This is in SigListXML format. No separate entries are provided for the folders involved. For this file list to be populated, the Delivery Log Create option must be enabled.
Skipped File List	The list of files that were skipped because they were already present in the cloud storage. These are absolute paths. This is in SigListXML format. No separate entries are provided for the folders involved. For this file list to be populated, the Delivery Log Create option must be enabled.
Failed File List	The list of files that failed to transfer. These are absolute paths. This is in SigListXML format. No separate entries are provided for the folders involved. For this file list to be populated, the Delivery Log Create option must be enabled.
Aggregate File List	A combined list of files from the transferred and skipped file lists. These are absolute paths. This is in SigListXML format. No separate entries are provided for the folders involved. For this file list to be populated, the Delivery Log Create option must be enabled.

Reference Workflow

The following illustrates the reference workflow that utilizes the FlightUpload component.



Error Handling

The component exits in error if there is a configuration error. If an error is encountered during the file transfer and the component determines the transfer cannot proceed, the component exits in error. Otherwise in error conditions, the component retries all transfers until the retry counter is exhausted.

All errors occurring during a file transfer are written to the job log. Additionally, the last portion of the Signiant Flight transport log is also written to the job log.

If the job log level is set to debug, then any deletion of the manifest file and the Signiant Flight client configuration file is overridden to facilitate troubleshooting.

If a specified file to be transferred does not exist, the component will log the missing file as an error but will continue to transfer remaining files. The component will exit with a success.