# TIBCO Flogo® Connector for Apache Pulsar

## User Guide

Version 1.3.4 | January 2025

Copyright © 2020-2021. Cloud Software Group, Inc. All Rights Reserved.

# Contents

Contents 2

Overview 3

Creating an Apache Pulsar Connection 4

Apache Pulsar Connection Details 4

Connecting to TIBCO Cloud Messaging - Apache Pulsar 7

Apache Pulsar Consumer Trigger 7

Apache Pulsar Function Trigger 13

Apache Pulsar Producer Activity 14

Apache Pulsar Message Ack Activity 19

Apache Pulsar Message NoAck Activity 19

TIBCO Documentation and Support Services 20

Legal and Third-Party Notices 22

# Overview

TIBCO Flogo® Connector for Apache Pulsar provides out-of-the-box activities and triggers to send and receive messages. You can connect to your Apache Pulsar broker by using transport layer security (TLS), JSON Web Token (JWT), or Open Authorization (OAuth2). Flogo® Connector for Apache Pulsar supports 2.6.x, 2.7.x, 2.8.x, 2.9.x, 2.10.x, and 3.0.x versions of Apache Pulsar.

Configure triggers that subscribe to messages published to a topic, including Exclusive, Shared, Failover, or KeyShared subscription types. Use activities to send messages to a topic by mapping string or JSON content, with optional compression.

Flogo Connector for Apache Pulsar supports OpenTelemetry. For more information, see the *Tracing Apps by Using OpenTelemetry Collector* topic of TIBCO Flogo® Enterprise TIBCO Flogo® app documentation.

Flogo Connector for Apache Pulsar supports Flow Limit. For more information, see the *Environment Variables* topic of TIBCO Flogo® EnterpriseTIBCO Flogo® app documentation.

Flogo Connector for Apache Pulsar is popularly used for the following purposes:

- To subscribe to messages produced by edge computing devices and perform transformation or analysis on them.
- To publish messages on any broker and topic in your network based on the state of the flow. You can send the results of the flow to a topic.

**Note:** Do not use the Linux/x86 option to build the app in Flogo® app as Apache Pulsar works only on a 64-bit machine. Install the GNU compiler collection (GCC) compiler on the Windows machine. Generate a Windows executable pulsar app from the builder-windows-amd64.exe tool on Flogo® Enterprise that may take more time than other platforms. For information about how to use Apache Pulsar, see the Apache Pulsar documentation.

# Creating an Apache Pulsar Connection

To use this connector, you must configure an Apache Pulsar client connection. The Apache Pulsar connection is used by all the activities in the Apache Pulsar category. The Apache Pulsar connection contains the parameters required to connect to Apache Pulsar.

### Before you begin

Before you create a connection, familiarize yourself with Apache Pulsar. For details about how to use Apache Pulsar, see the Apache Pulsar documentation.

### Procedure

1. On the global navigation bar, click the **Connections** tab and perform one of the following actions:
    - To add a connection for the first time, click the **Create connection** link and then select the Apache Pulsar card. You can search for a connector card by typing the connector name in the search field.
    - To create subsequent connections, click the **Create** button.
2. In the Apache Pulsar dialog, enter the connection details. For field descriptions, see the Apache Pulsar Connection Details topic.
3. Click **Save Connection**.

# Apache Pulsar Connection Details

The Apache Pulsar connection is a container for all the client connection parameters used by the Apache Pulsar Publisher activity and the Apache Pulsar Subscriber trigger. To establish the connection successfully, specify the following connection details in the Apache Pulsar dialog:

| Field | Required | Description |
|---|---|---|
| **Name** | Yes | Specify a unique name for the connection that you are creating. This name is displayed in the **Connection Name** drop-down list for each activity. |

| Field | Required | Description |
|---|---|---|
| **Description** | No | A brief description of the connection. |
| **Broker URL** | Yes | The URL of the broker. It follows the format- `pulsar+ssl://<host_name>:<port>` where `pulsar` is the protocol and `+ssl` is an optional parameter that shows that the connection is secured by TLS. The content following the `://` is the host name and the port on which the broker is running.<br>If the URL of the broker is SSL then the **CaCert** field is enabled. This field must be provided unless the **Allow Insecure Connection** option is enabled.<br>If the authorization is enabled, then the connector would insist that you use SSL to protect the credentials when you are on the network. The field is app property enabled. |
| **Authorization Type** | Yes | You can choose the authorization type from the following options:<br>• None<br>• TLS: The client must present a certificate in which the **Common Name** field matches the role for resources requested on the broker<br>• JWT: The client must present a JSON Web Token composed on the connected broker for the specific role authorized by this connection<br>• OAuth2: Access the token-based client authentication by using an OAuth 2.0 authentication service |
| **Allow Insecure Connection** | Yes | The **Allow Insecure Connection** field is available if `+ssl` is present in the **Broker URL** field so that the self-signed development mode certificates can be used to connect to the broker. If this field is enabled, then the **Broker CA** field is removed as it is not needed anymore. The field is app property enabled. |

| Field | Required | Description |
|---|---|---|
| **Broker CA** | Yes | The **Broker CA** field points to the certificate authority used to sign the broker's certificate for secure connections. The Flogo client now trusts the broker. The field is app property enabled. |
| **Client Cert** | Yes | If the **Authorization Type** field is TLS, then the **Client Cert** field must point to the client's certificate. The broker must trust this certificate. The field is app property enabled. |
| **Client Key** | Yes | If the **Authorization Type** is TLS, then the **Client Key** field must point to the client's key. The field is app property enabled. |
| **JSON Web Token** | Yes | If the **Authorization Type** is JWT, then the **JSON Web Token** field must contain the text contents of the JWT obtained for this connection. |
| **Connection Timeout** | No | Connection timeout in seconds. The timeout for establishing a TCP connection. The default value is 30 seconds. The field is app property enabled. |
| **Operation Timeout** | No | Operation timeout in seconds. The operations like Producer-create and Subscribe would be retried until this interval. The default value is 30 seconds. The field is app property enabled. |
| **Private Key** | Yes | If the **Authorization Type** is OAuth2, then select the JSON credential file as the private key. The field is app property enabled. |
| **Issuer URL** | No | If the **Authorization Type** is OAuth2, provide the URL of the authentication provider that allows the pulsar client to obtain an access token. The field is app property enabled. |
| **Audience** | No | If the **Authorization Type** is OAuth2, provide an OAuth 2.0 resource server identifier for the pulsar cluster. The field is app property enabled. |

| Field | Required | Description |
|---|---|---|
| **Scope** | No | If the **Authorization Type** is OAuth2, mention the scope of an access request separated by spaces. The field is app property enabled. |

# Connecting to TIBCO Cloud Messaging - Apache Pulsar

You can configure Apache Pulsar connection using *tcm-config.yaml* as follows:

| Field | Description |
|---|---|
| **Broker URL** | Configure the Pulsar URL by using pulsar_url from the tcm-config.yaml file. |
| **Authorization Type** | JWT |
| **JSON Web Token** | Configure JWT authentication by using tcm_authentication_key from the tcm-config.yaml file as the JSON web token. |

You can obtain pulsar_namespace to be used in the pulsar topic from the *tcm-config.yaml* file. For more information, see Apache Pulsar on TIBCO Cloud Messaging documentation.

# Apache Pulsar Consumer Trigger

Apache Pulsar consumer is a Flogo trigger that receives messages published to the configured topic. Each message triggers a new flow. The message gets acknowledged as soon as the Message Ack activity is encountered in the flow or the flow is successful.

## Trigger Settings

The **Trigger Settings** tab has the following fields:

| Field | Required | Description |
|---|---|---|
| **Pulsar Connection** | Yes | Name of the connection. |
| **Topic** | No | The name of the topic from which the message is consumed.<br>The **Topic** field has the following format : `persistent://public/default/foo`<br>The first segment is the type of topic. The topic can be persistent or nonpersistent. With persistent topics, all the messages are persisted on the disk. The second segment is the name of the domain or a tenant. The third segment is the namespace within the domain and the fourth segment is the name of the individual topic. |
| **Topics Pattern** | No | A pattern of topics you can subscribe to, for consuming messages from multiple topics. You can specify the group using a regular expression pattern.<br>Example: `mytopic*` |
| **Subscription Name** | Yes | The subscription name is used by the broker to combine subscribers belonging to a single app into a logical group for message delivery. Subscriptions can also be used to deliver backlog messages to a consumer that goes offline for some time. |
| **Subscription Type** | Yes | The following options are available:<br>• **Exclusive**: In the **Exclusive** mode, only a single consumer is allowed to attach to the subscription. An error occurs if multiple consumers subscribe to a topic by using the same subscription.<br>• **Shared**: In the **Shared** mode, multiple consumers can attach to the same subscription. Messages are delivered in a round-robin distribution across consumers, and any given message is delivered to only one consumer. |

| Field | Required | Description |
|---|---|---|
| | | • **Failover**: In the **Failover** mode, multiple consumers can attach to the same subscription. Messages are delivered to the master consumer only. In case of the failover of a master consumer, the next master is selected by the broker and further messages are redirected to the next master.<br>• **KeyShared**: In the **KeyShared** mode, multiple consumers can attach to the same subscription. Messages are delivered in distribution across consumers and messages with the same key or the same ordering key are delivered to only one consumer. |
| **Processing Mode** | Yes | The trigger processes the messages in one of the following modes:<br>• **Sync**: The **Sync** mode is the default mode. The handler receives and processes only one message at a time. This mode guarantees the processing order of the messages.<br>• **Async**: The handler receives messages concurrently. This mode does not guarantee the processing order of the messages. The total concurrent messages handled by the trigger are based on the type of the engine runner. You can set the engine runner type by using the *FLOGO_ RUNNER_TYPE* variable. The following values are supported:<br>◦ *POOLED*: The total concurrent messages handled by the trigger handler. To achieve higher concurrency, set *FLOGO_RUNNER_ WORKERS* to a higher value.<br>For an app, the total messages processed concurrently is equal to the number of |

| Field | Required | Description |
|---|---|---|
| | | trigger handlers configured with async processing multiplied by the FLOGO_RUNNER_WORKERS. For example, if two Pulsar trigger handlers are configured with async mode and FLOGO_RUNNER_WORKERS is set to 10, 20 messages would be processed concurrently at a given time.<br>○ *DIRECT*: Currently, 200 concurrent messages are handled by the trigger handler.<br>For an app, the total messages processed concurrently is equal to the number of trigger handlers configured with async processing multiplied by 200. For example, if two Pulsar trigger handlers are configured with async mode then 400 messages would be processed concurrently at a given time. |
| **Initial Position** | Yes | On the **Initial Position** field, select **Latest** to receive the messages that have been published after the subscriber has been connected. Select **Earliest** to receive all the stored and new messages. |
| **Seek By** | No | Reset the subscription associated with the consumer to a specific message id or publish timestamp. The following options are available:<br>• **Seek By MessageID**: Reset the subscription associated with the consumer to the specified message id. Seeking by Message ID in Pulsar allows you to start consuming messages from a specific message with the Ledger ID and Entry ID.<br>• **Ledger ID**: The Ledger ID identifies a specific ledger in the bookkeeper storage system used by Pulsar. Pulsar stores messages in ledgers, and |

| Field | Required | Description |
|---|---|---|
| | | each ledger contains multiple entries. Set the ledger ID to the desired ledger.<br>• **Entry ID**: The Entry ID identifies a specific entry (or message) within a ledger. Set the entry ID to the desired entry.<br>• **Seek By Timestamp**: Reset the subscription associated with the consumer to the specified publish timestamp. Seeking by timestamp in Pulsar allows you to start consuming messages from a specific point in time.<br>Timestamp: Set the timestamp of the desired message in RFC 3339 format. |
| **DLQ Topic** | No | The **DLQ Topic** is available only if the **Subscription Type** is **Shared**. If the flow started by the trigger does not complete successfully, the message is negatively acknowledged. If this happens repeatedly, the message effectively blocks processing by the trigger. This field allows you to relocate messages that cannot be processed to another topic. If the **DLQ Topic** field is not provided, then the dead letter queue (DLQ) processing is not performed.<br>**Note:** A delay exists between a negative acknowledgment and reposting the message. So, it can take several minutes for a message to arrive on the DLQ. |
| **DLQ Max Deliveries** | No | The number of times a message is negatively acknowledged before being rerouted to the **DLQ Topic** field. |
| **Nack Redelivery Delay** | Yes | The delay in seconds after which the message is redelivered when the message is not acknowledged. Default: 60 seconds. |

| Field | Required | Description |
|---|---|---|
| **Enable Batch Index Acknowledgment** | No | This is a checkbox to enable the batch index acknowledgments. Ensure that the batch index acknowledgment is also enabled by setting the acknowledgmentAtBatchIndexLevelEnabled parameter to true at the broker side. |
| **Max Pending Chunked Message** | No | Specifies the maximum size of the queue that can hold pending message chunks for a consumer.<br>Default: 100 |
| **Expire time of incomplete Chunk** | No | Specifies the total time interval to expire the incomplete chunks, if the consumer fails to receive all the chunks of a message in the given time period.<br>Default: 60 sec |
| **Auto Ack Incomplete Chunk** | No | This automatically acknowledges any pending message chunks when the Chunk Max Pending Messages threshold value is reached. |
| **Replicate Subscription State** | No | Select the Geo Replication State checkbox to enable the geo replication feature. |
| **Message Format** | Yes | The **Message Format** field controls the format of the output schema. The following options are available:<br>• **String**<br>• **JSON**: If **JSON** is selected, a text editor is provided on the **Output Settings** tab to accept a JSON document |

The consumer created has a name in the following format that can be seen by using the Pulsar Admin REST API: `<ApplicationName>-<AppVersion>-<FlowName>-<HostName>`

## Output Settings

The **Output Settings** tab has the following fields:

| Field | Required | Description |
|---|---|---|
| **Message Properties** | No | You can add a property value to the properties field presented on the input schema. Each additional property is presented as a named string to be mapped. |
| **Schema for JSON value** | No | The **Schema for JSON value** field is only available when the **Message Format** field on the **Triggers Settings** tab is JSON. This is a free form text editor that accepts any valid JSON document, which is then presented on the output schema. |

### Map to Flow Inputs

The **Map to Flow Inputs** tab has the following fields:

| Field | Description |
|---|---|
| **payload** | Either a simple string or a representation of the JSON document provided on the **Output Settings** tab. |
| **properties** | An object with a string value for each of the named properties from the **Output Settings** tab. |
| **topic** | If the subscriber subscribes to multiple topics, the **topic** field provides the actual topic on which the message has arrived. |
| **msgid** | A string representing the message id. |
| **redelivery Count** | An integer representing the redelivery count of the message. |

# Apache Pulsar Function Trigger

The Apache Pulsar Function Trigger receives the input message in byte format from the source (input) topic. The activities inside the flow perform operations on the input message and return the output as a message object to the sink (output) topic. The output message can be coerced with the schema.

When an Apache Pulsar Function Trigger is added in a flow, no other triggers can be added to the app. Download the app, generate the executable binary, and deploy it to the Apache Pulsar cluster. Configure the app as an Apache Pulsar function executable binary by using the Apache Pulsar functions CLI (Command Line Interface) or REST API. Pass the runtime argument as Go language argument. The app fails if you push the Apache Pulsar Function Trigger to TIBCO® Cloud Integration.

## Map to Flow Inputs

The **Map to Flow Inputs** tab has the following field:

| Field | Description |
|---|---|
| message | Receives the input message in byte format. |

## Map from Flow Outputs

The **Map from Flow Outputs** tab has the following field:

| Field | Description |
|---|---|
| out | The mapped output is published to the sink (output) topic. |

# Apache Pulsar Producer Activity

Apache Pulsar Producer activity can be used to map a string or a JSON object to a message. You can use the Producer activity to map the properties such as `string:stringpairs` and an optional key. You can send messages to a single topic and compress them by all the supported algorithms.

## Settings

On the **Settings** tab, configure the following settings:

| Field | Required | Description |
|---|---|---|
| Pulsar Connection | Yes | Name of the connection. |
| Topic Name | Yes | The name of the topic to which the message is published. The **Topic Name** field has the following format :`persistent://public/default/foo`<br>The first segment is the type of topic. The topic can be persistent or nonpersistent. With persistent topics, all the messages are persisted to the disk. The second segment is the name of the domain or tenant. The third segment is the namespace within the domain and the fourth segment is the name of the individual topic.<br>An abbreviated form of the topic name can be used only when the rightmost segment is provided. If the other segments of the topic name are needed then the complete topic name must be used. |
| Send Mode | Yes | The message can be sent to the brokers using the Sync and Async mode. |
| Send Timeout | No | Set the SendTimeout (in milliseconds) to specify how long to wait for the server to acknowledge the message. If the server doesn't respond within this time, the operation will fail with a SendTimeout error. To disable this feature, use -1. The default value is -1. |
| Compression Type | No | The following options are available for the compression type:<br>• None<br>• LZ4<br>• ZLIB<br>• ZSTD |
| Chunking | No | Select the Chunking checkbox to enable the chunking feature. |

| Field | Required | Description |
|---|---|---|
| | | **Note:** If you set Chunking to True, Batching automatically gets disabled. |
| ChunkMaxMessage Size | Yes | The maximum chunk size of a message that can be sent in bytes.<br>Default: 5242880. |
| Batching | No | Select the Batching checkbox to enable batching. The batching at the<br>producer happens only when the Send Mode is Async on the General tab.<br>**Note:** If you set Batching to True, Chunking automatically gets disabled. |
| Batching Max Messages | Yes | Set the maximum number of messages that can be sent in a particular batch.<br>Default: 1000 |
| Batching Max Size | Yes | This specifies maximum number of bytes permitted in a batch.<br>Default: 128000 |
| Batching Max Publish Delay | Yes | Specifies the maximum time in milliseconds within which the messages sent can be batched. Default value: 10 ms |
| Enable Replication | No | Enables the replication for this message. |
| Replication Clusters | No | Provide the comma separated list of cluster names. |

The producer created has a name in the following format that can be seen by using the Pulsar Admin REST API: `<ApplicationName>-<AppVersion>-<FlowName>-<ActivityName>-<HostName>`

## Input Settings

The **Input Settings** tab has the following fields:

| Field | Required | Description |
|---|---|---|
| Message Format | Yes | The following options are available for the message format:<br>• String: A simple string can be mapped on the Input schema.<br>• JSON: The Input schema is updated to reflect the structure of the JSON document and flow data can be mapped to the fields. |
| Message Properties | No | You can use the **Message Properties** field to add property value to the properties field presented on the Input schema. Each additional property is presented as a named string that is mapped. |
| Schema for JSON Value | No | You can see the **Schema for JSON Value** field only if the **Message Format** on the **Input Settings** tab is JSON. This is a free form text editor, which accepts any valid JSON document that is presented on the Input schema. This JSON document must also be used by the receiving app to decode the message. |

## Input

The **Input** tab has the following fields:

| Field | Description |
|---|---|
| payload | The message to be published to a topic. Either a simple string or a representation of the JSON document provided on the **Input Settings** tab. |
| properties | An object with a string value for each of the named properties from the **Input Settings** tab. |
| key | A string value used by the topic compaction function of the broker. In the shared subscriber mode subscribers are bound to specific keys so that the repeat keys are always processed by the same consumer. |

## Output Settings

The **Output Settings** tab has the following field:

| Field | Description |
|---|---|
| msgid | A string representing the message id. |

## Output

The **Output** tab displays the output schema of the activity as a tree structure. The output is read-only. The information in the schema varies based on the fields selected on the **Settings** tab. The properties that are displayed in the schema correspond to the output of this activity and can be used as input by subsequent activities in the flow.

## Output Settings

The **Output Settings** tab has the following field:

| Field | Description |
|---|---|
| msgid | A string representing the message id. |

## Using the Loop feature in an activity

If you want this activity to iterate multiple times within the flow, enter an expression that evaluates the iteration details. Select a type of iteration from the Type menu. The default type is None, which means the activity does not iterate. For more information, refer to the "Using the Loop Feature in an Activity" topic".

## Using Retry on error feature in an activity

Using the Retry on Error tab, you can set the number of times the flow tries to run the activity on encountering an error that can be fixed on retrial. The errors such as waiting for a server to start, intermittent connection failures, or connection timeout can be fixed on retrial.

You can set the count and the interval in one of the following ways:

- Manually type the value in the mapper.
- Map the value from the previous Activity.
- Select a function from the list of functions
- Map app property to override the values

| Field | Description |
|---|---|
| Count | The number of times the flow should attempt to run the activity. This value must be an integer. |
| Interval (in millisecond) | The time to wait in between each attempt to run the activity. This value must be an integer. |

**Note:** The Count and Interval fields are mandatory. By default, the values are set to 0.

# Apache Pulsar Message Ack Activity

Apache Pulsar Message Ack activity acknowledges the Pulsar message received by the Apache Pulsar Consumer trigger. This activity helps you to acknowledge the message as soon as it is processed in the flow. By default, messages are acknowledged only when the flow is completed. This activity does not have any settings and does not need any configuration.

# Apache Pulsar Message NoAck Activity

Apache Pulsar Message NoAck activity sends the no acknowledgment response to the Apache Pulsar Consumer trigger as soon as the flow reaches the activity. This activity does not have any settings and does not need any configuration.

# TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join TIBCO Community.

## How to Access TIBCO Documentation

Documentation for TIBCO products is available on the Product Documentation website, mainly in HTML and PDF formats.

The Product Documentation website is updated frequently and is more current than any other documentation included with the product.

## Product-Specific Documentation

The documentation for this product is available on the TIBCO Flogo® Connector for Apache Pulsar Product Documentation page.

## How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our product Support website.
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the product Support website. If you do not have a username, you can request one by clicking **Register** on the website.

## How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the TIBCO Ideas Portal. For a free registration, go to TIBCO Community.

# Legal and Third-Party Notices

SOME CLOUD SOFTWARE GROUP, INC. (“CLOUD SG”) SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, “INCLUDED SOFTWARE”). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

USE OF CLOUD SG SOFTWARE AND CLOUD SERVICES IS SUBJECT TO THE TERMS AND CONDITIONS OF AN AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER AGREEMENT WHICH IS DISPLAYED WHEN ACCESSING, DOWNLOADING, OR INSTALLING THE SOFTWARE OR CLOUD SERVICES (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH LICENSE AGREEMENT OR CLICKWRAP END USER AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE SAME TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of Cloud Software Group, Inc.

TIBCO, the TIBCO logo, the TIBCO O logo, and Flogo are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG’s Third Party Trademark Notices (https://www.cloud.com/legal) for more information.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: https://scripts.sil.org/OFL

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

Cloud SG software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the “readme” file for the availability of a specific version of Cloud SG software on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "README" FILES.

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at https://www.cloud.com/legal.