IBSNOW: Snowflake IoT Bridge properties configuration

New IoT Bridge for Snowflake Available

This document covers v2.0.0 and older versions of IoT Bridge for Snowflake. It now ships as part of Chariot. To see the latest IoT Bridge for Snowflake documentation, go to IoT Bridge for Snowflake in Chariot.

The IoT Bridge Properties (ibsnow.properties) contains configuration specific to the bridge and allows for granular control over MQTT server connections, Sparkplug and non-Sparkplug MQTT subscriptions, Sparkplug host behavior, schema create and data sync rates, etc.

Configuration Options

IoT Bridge for Snowflake is configured with a configuration file on the filesystem of the AWS or Azure instance.

If you are unfamiliar with how to access the instance, see these pages for access instructions:

AWS AMI Access Instructions

Azure VM Access Instructions

Also note after modifying the configuration, the application must be restarted. This can be done with the following command.

```
sudo systemctl restart ibsnow
```

The path to the configuration file is:

`/opt/ibsnow/conf/ibsnow.properties`

Once you open the file, you will see the following options.

ibsnow.properties

```
# The IBSNOW instance friendly name. If omitted, it will become 'IBSNOW-vm-instance-id'
#ibsnow_instance_name =

# The Cloud region the IoT Bridge for Snowflake instance is in
# ibsnow_cloud_region = us-east-1

# MQTT Server definitions. IoT Bridge for Snowflake supports multiple MQTT Servers. Each definition must
include and 'index' as shown
# below represented by 'X'. The first should begin with 1 and each additional server definition should have an
index of 1 greater
# than the previous.
# mqtt_server_url.X                    # The MQTT Server URL
# mqtt_server_name.X                   # The MQTT Server name
# mqtt_username.X                      # The MQTT username (if required by the MQTT Server)
# mqtt_password.X                      # The MQTT password (if required by the MQTT Server)
# mqtt_keepalive_timeout.X             # The MQTT keep-alive timeout in seconds
# mqtt_ca_cert_chain_path.X            # The path to the TLS Certificate Authority certificate chain
# mqtt_client_cert_path.X              # The path to the TLS certificate
# mqtt_client_private_key_path.X       # The path to the TLS private key
# mqtt_client_private_key_password.X   # The TLS private key password
# mqtt_verify_hostname.X               # Whether or not to verify the hostname against the server certificate
# mqtt_client_id.X                     # The Client ID of the MQTT Client
# mqtt_sparkplug_subscriptions.X       # The Sparkplug subscriptions to issue when connecting to the MQTT
Server.
                                       # By default this is spBv1.0/# but can be scoped more narrowly (e.g.
spBv1.0/Group1/#)
                                       # It can also be a comma separated list (e.g. spBv1.0/Group1/#,spBv1.0
/Group2/#)
```

```
mqtt_server_url.1 = ssl://REPLACE_WITH_MQTT_SERVER_ENDPOINT:8883
mqtt_server_name.1 = My MQTT Server

# Enable Snowflake Sparkplug MQTT Application
snowflake_application_enabled = true

# Enable Snowflake Raw MQTT Application
snowflake_mqtt_application_enabled = true

# Comma separated list of Sparkplug subscriptions
mqtt_sparkplug_subscriptions.1 = spBv1.0/#

# Comma separated list of MQTT subscription topic:QoS pairs (e.g. a/#:0,b/#:0)
# mqtt_subscriptions.1 =

#mqtt_keepalive_timeout.1 = 30
#mqtt_verify_hostname.1 = true
#mqtt_username.1 =
#mqtt_password.1 =
#mqtt_ca_cert_chain_path.1 = /opt/ibsnow/conf/certs/
#mqtt_client_cert_path.1 = /opt/ibsnow/conf/certs/
#mqtt_client_private_key_path.1 = /opt/ibsnow/conf/certs/
#mqtt_client_private_key_password.1 =
#mqtt_client_id.1 =

# The Sparkplug sequence reordering timeout in milliseconds
sequence_reordering_timeout = 5000

# Whether or not to block auto-rebirth requests
#block_auto_rebirth = false

# The primary host ID if this is the acting primary host
#primary_host_id =

# Snowflake streaming connection properties - A custom client name for the connection (e.g. MyClient)
#snowflake_streaming_client_name = IBSNOWClient

# Maximum number of rows to insert into the snowflake_streaming_table at once. Set to 0 for no limit.
snowflake_max_streaming_insert_batch_size = 0

# Maximum number of rows to insert into the snowflake_mqtt_streaming_table at once. Set to 0 for no limit.
snowflake_max_mqtt_streaming_insert_batch_size = 0

# Do raw MQTT inserts one topic at a time
snowflake_topic_based_mqtt_streaming_insert = false

# Ingest task execution period (in seconds)
snowflake_streaming_ingest_task_period = 1

# Raw MQTT ingest task execution period (in seconds)
snowflake_mqtt_streaming_ingest_task_period = 1


# Snowflake streaming connection properties - The scheme to use for channels and their names
# This MUST be one of the following: STATIC, GROUP_ID, EDGE_ID
# STATIC - means to use a single channel. If using this mode, the snowflake_streaming_channel_name
# GROUP_ID - means to use the Sparkplug Group ID for the channel name on incoming data
# EDGE_ID - means to use the Sparkplug Group ID and the Edge Node ID for the channel name on incoming data
# DEVICE_ID - means to use the Sparkplug Group ID, Edge Node ID, and Device ID for the channel name on incoming
data
snowflake_streaming_channel_scheme = EDGE_ID

# Snowflake streaming connection properties - A custom channel name for the connection (e.g. MyChannel)
# If this is left blank/empty, Channel names of the Sparkplug Group ID will be used instead of a single channel
# snowflake_streaming_channel_name =

# Snowflake streaming connection properties - The Table name for Sparkplug data associated with the Database
and Schema already provisioned in the Snowflake account (e.g. MyTable)
snowflake_streaming_table_name = SPARKPLUG_RAW
```

```
# Snowflake streaming connection properties - The Table name for RAW MQTT data.
snowflake_mqtt_streaming_table_name = MQTT_RAW

# Maximum number of streaming channels for RAW MQTT messages
# snowflake_max_mqtt_streaming_channels =

# Snowflake notify connection properties - The Database name associated with the connection that is already
provisioned in the Snowflake account (e.g. MyDb)
snowflake_notify_db_name = cl_bridge_node_db

# Snowflake notify connection properties - The Schema name associated with the Database already provisioned in
the Snowflake account (e.g. PUBLIC)
snowflake_notify_schema_name = stage_db

# Snowflake notify connection properties - The Warehouse name associated with the notifications already
provisioned in the Snowflake account (e.g. PUBLIC)
snowflake_notify_warehouse_name = cl_bridge_ingest_wh

# Whether or not to create and update IBSNOW informational tracking metrics
# ibsnow_metrics_enabled = true

# The Sparkplug Group ID to use for IBSNOW asset names
ibsnow_metrics_sparkplug_group_id = IBSNOW

# The 'Bridge Info' Sparkplug Edge Node ID to use for IBSNOW assets
ibsnow_metrics_bridge_info_sparkplug_edge_node_id = Bridge Info

# The 'Edge Node Info' Sparkplug Edge Node ID to use for IBSNOW assets
ibsnow_metrics_edge_node_info_sparkplug_edge_node_id = Edge Node Info

# The 'MQTT Client Info' Sparkplug Edge Node ID to use for IBSNOW assets
ibsnow_metrics_mqtt_client_info_sparkplug_edge_node_id = MQTT Client Info

# Whether or not to send notification tasks to Snowflake based on incoming Sparkplug events
snowflake_notify_task_enabled = true

# The number of threads to use for BIRTH handling in Snowflake
# snowflake_notify_task_birth_thread_count = 100

# The number of times to retry the NotifyIngest task on failure
snowflake_notify_nbirth_retries = 10

# The number of milliseconds to delay after receiving an NBIRTH before notifying Snowflake over the event
(requires snowflake_notify_task_enabled is true)
snowflake_notify_nbirth_task_delay = 15000

# The number of milliseconds to delay after receiving a DBIRTH or DATA message before notifying Snowflake over
the event (requires snowflake_notify_task_enabled is true)
snowflake_notify_data_task_delay = 5000

# NotifyTimerTask keep alive in seconds. This setting allows to kill inactive threads after keep-alive expires.
snowflake_notify_task_keep_alive = 60

# Enable performance metrics such as 'NotifyIngest thread pool info' and 'Message Inflow Rates' per EdgeNode or
topic.
snowflake_enable_performance_metrics = true
```

Configuration Examples

Cirrus Link's Chariot MQTT Server using a real signed TLS certification

If you are using Cirrus Link's Chariot MQTT Server using a real signed TLS certification then your configuration file should look similar to the one below.

Chariot MQTT Server using a real signed TLS certificate

```
# The IBSNOW instance friendly name. If omitted, it will become 'IBSNOW-vm-instance-id'
#ibsnow_instance_name =
```

```
# The Cloud region the IoT Bridge for Snowflake instance is in
# ibsnow_cloud_region = us-east-1

# MQTT Server definitions. IoT Bridge for Snowflake supports multiple MQTT Servers. Each definition must
include and 'index' as shown
# below represented by 'X'. The first should begin with 1 and each additional server definition should have an
index of 1 greater
# than the previous.
# mqtt_server_url.X                    # The MQTT Server URL
# mqtt_server_name.X                   # The MQTT Server name
# mqtt_username.X                      # The MQTT username (if required by the MQTT Server)
# mqtt_password.X                      # The MQTT password (if required by the MQTT Server)
# mqtt_keepalive_timeout.X             # The MQTT keep-alive timeout in seconds
# mqtt_ca_cert_chain_path.X            # The path to the TLS Certificate Authority certificate chain
# mqtt_client_cert_path.X              # The path to the TLS certificate
# mqtt_client_private_key_path.X       # The path to the TLS private key
# mqtt_client_private_key_password.X   # The TLS private key password
# mqtt_verify_hostname.X               # Whether or not to verify the hostname against the server certificate
# mqtt_client_id.X                     # The Client ID of the MQTT Client
# mqtt_sparkplug_subscriptions.X       # The Sparkplug subscriptions to issue when connecting to the MQTT
Server.
                                       # By default this is spBv1.0/# but can be scoped more narrowly (e.g.
spBv1.0/Group1/#)
                                       # It can also be a comma separated list (e.g. spBv1.0/Group1/#,spBv1.0
/Group2/#)

mqtt_server_url.1 = ssl://chariot.mycompany.com:8883
mqtt_server_name.1 = Chariot MQTT Server

# Enable Snowflake Sparkplug MQTT Application
snowflake_application_enabled = true

# Enable Snowflake Raw MQTT Application
snowflake_mqtt_application_enabled = true

# Comma separated list of Sparkplug subscriptions
mqtt_sparkplug_subscriptions.1 = spBv1.0/#

# Comma separated list of MQTT subscription topic:QoS pairs (e.g. a/#:0,b/#:0)
# mqtt_subscriptions.1 =

#mqtt_keepalive_timeout.1 = 30
#mqtt_verify_hostname.1 = true
mqtt_username.1 = admin
mqtt_password.1 = changeme
#mqtt_ca_cert_chain_path.1 = /opt/ibsnow/conf/certs/
#mqtt_client_cert_path.1 = /opt/ibsnow/conf/certs/
#mqtt_client_private_key_path.1 = /opt/ibsnow/conf/certs/
#mqtt_client_private_key_password.1 =
#mqtt_client_id.1 =

# The Sparkplug sequence reordering timeout in milliseconds
sequence_reordering_timeout = 5000

# Whether or not to block auto-rebirth requests
#block_auto_rebirth = false

# The primary host ID if this is the acting primary host
#primary_host_id =

# Snowflake streaming connection properties - A custom client name for the connection (e.g. MyClient)
#snowflake_streaming_client_name = IBSNOWClient

# Maximum number of rows to insert into the snowflake_streaming_table at once. Set to 0 for no limit.
snowflake_max_streaming_insert_batch_size = 0

# Maximum number of rows to insert into the snowflake_mqtt_streaming_table at once. Set to 0 for no limit.
snowflake_max_mqtt_streaming_insert_batch_size = 0

# Do raw MQTT inserts one topic at a time
```

```
snowflake_topic_based_mqtt_streaming_insert = false

# Ingest task execution period (in seconds)
snowflake_streaming_ingest_task_period = 1

# Raw MQTT ingest task execution period (in seconds)
snowflake_mqtt_streaming_ingest_task_period = 1


# Snowflake streaming connection properties - The scheme to use for channels and their names
# This MUST be one of the following: STATIC, GROUP_ID, EDGE_ID
# STATIC - means to use a single channel. If using this mode, the snowflake_streaming_channel_name
# GROUP_ID - means to use the Sparkplug Group ID for the channel name on incoming data
# EDGE_ID - means to use the Sparkplug Group ID and the Edge Node ID for the channel name on incoming data
# DEVICE_ID - means to use the Sparkplug Group ID, Edge Node ID, and Device ID for the channel name on incoming
data
snowflake_streaming_channel_scheme = EDGE_ID

# Snowflake streaming connection properties - A custom channel name for the connection (e.g. MyChannel)
# If this is left blank/empty, Channel names of the Sparkplug Group ID will be used instead of a single channel
# snowflake_streaming_channel_name =

# Snowflake streaming connection properties - The Table name for Sparkplug data associated with the Database
and Schema already provisioned in the Snowflake account (e.g. MyTable)
snowflake_streaming_table_name = SPARKPLUG_RAW

# Snowflake streaming connection properties - The Table name for RAW MQTT data.
snowflake_mqtt_streaming_table_name = MQTT_RAW

# Maximum number of streaming channels for RAW MQTT messages
# snowflake_max_mqtt_streaming_channels =

# Snowflake notify connection properties - The Database name associated with the connection that is already
provisioned in the Snowflake account (e.g. MyDb)
snowflake_notify_db_name = cl_bridge_node_db

# Snowflake notify connection properties - The Schema name associated with the Database already provisioned in
the Snowflake account (e.g. PUBLIC)
snowflake_notify_schema_name = stage_db

# Snowflake notify connection properties - The Warehouse name associated with the notifications already
provisioned in the Snowflake account (e.g. PUBLIC)
snowflake_notify_warehouse_name = cl_bridge_ingest_wh

# Whether or not to create and update IBSNOW informational tracking metrics
# ibsnow_metrics_enabled = true

# The Sparkplug Group ID to use for IBSNOW asset names
ibsnow_metrics_sparkplug_group_id = IBSNOW

# The 'Bridge Info' Sparkplug Edge Node ID to use for IBSNOW assets
ibsnow_metrics_bridge_info_sparkplug_edge_node_id = Bridge Info

# The 'Edge Node Info' Sparkplug Edge Node ID to use for IBSNOW assets
ibsnow_metrics_edge_node_info_sparkplug_edge_node_id = Edge Node Info

# The 'MQTT Client Info' Sparkplug Edge Node ID to use for IBSNOW assets
ibsnow_metrics_mqtt_client_info_sparkplug_edge_node_id = MQTT Client Info

# Whether or not to send notification tasks to Snowflake based on incoming Sparkplug events
snowflake_notify_task_enabled = true

# The number of threads to use for BIRTH handling in Snowflake
# snowflake_notify_task_birth_thread_count = 100

# The number of times to retry the NotifyIngest task on failure
snowflake_notify_nbirth_retries = 10

# The number of milliseconds to delay after receiving an NBIRTH before notifying Snowflake over the event
(requires snowflake_notify_task_enabled is true)
snowflake_notify_nbirth_task_delay = 15000
```

```
# The number of milliseconds to delay after receiving a DBIRTH or DATA message before notifying Snowflake over
the event (requires snowflake_notify_task_enabled is true)
snowflake_notify_data_task_delay = 5000

# NotifyTimerTask keep alive in seconds. This setting allows to kill inactive threads after keep-alive expires.
snowflake_notify_task_keep_alive = 60

# Enable performance metrics such as 'NotifyIngest thread pool info' and 'Message Inflow Rates' per EdgeNode or
topic.
snowflake_enable_performance_metrics = true
```

AWS IoT Core

If you are using AWS IoT Core then your configuration file should look similar to the one below.

AWS IoT Core

```
# The IBSNOW instance friendly name. If omitted, it will become 'IBSNOW-vm-instance-id'
#ibsnow_instance_name =

# The Cloud region the IoT Bridge for Snowflake instance is in
# ibsnow_cloud_region = us-east-1

# MQTT Server definitions. IoT Bridge for Snowflake supports multiple MQTT Servers. Each definition must
include and 'index' as shown
# below represented by 'X'. The first should begin with 1 and each additional server definition should have an
index of 1 greater
# than the previous.
# mqtt_server_url.X                    # The MQTT Server URL
# mqtt_server_name.X                   # The MQTT Server name
# mqtt_username.X                      # The MQTT username (if required by the MQTT Server)
# mqtt_password.X                      # The MQTT password (if required by the MQTT Server)
# mqtt_keepalive_timeout.X             # The MQTT keep-alive timeout in seconds
# mqtt_ca_cert_chain_path.X            # The path to the TLS Certificate Authority certificate chain
# mqtt_client_cert_path.X              # The path to the TLS certificate
# mqtt_client_private_key_path.X       # The path to the TLS private key
# mqtt_client_private_key_password.X   # The TLS private key password
# mqtt_verify_hostname.X               # Whether or not to verify the hostname against the server certificate
# mqtt_client_id.X                     # The Client ID of the MQTT Client
# mqtt_sparkplug_subscriptions.X       # The Sparkplug subscriptions to issue when connecting to the MQTT
Server.
                                       # By default this is spBv1.0/# but can be scoped more narrowly (e.g.
spBv1.0/Group1/#)
                                       # It can also be a comma separated list (e.g. spBv1.0/Group1/#,spBv1.0
/Group2/#)

mqtt_server_url.1 = ssl://b9ffnzzzzzzzz-ats.iot.us-east-1.amazonaws.com:8883
mqtt_server_name.1 = AWS IoT Core MQTT Server

# Enable Snowflake Sparkplug MQTT Application
snowflake_application_enabled = true

# Enable Snowflake Raw MQTT Application
snowflake_mqtt_application_enabled = true

# Comma separated list of Sparkplug subscriptions
mqtt_sparkplug_subscriptions.1 = spBv1.0/#

# Comma separated list of MQTT subscription topic:QoS pairs (e.g. a/#:0,b/#:0)
# mqtt_subscriptions.1 =

#mqtt_keepalive_timeout.1 = 30
#mqtt_verify_hostname.1 = true
#mqtt_username.1 =
#mqtt_password.1 =
mqtt_ca_cert_chain_path.1 = /opt/ibsnow/conf/certs/AmazonRootCA1.pem
mqtt_client_cert_path.1 = /opt/ibsnow/conf/certs/72d382zzzz.cert.pem
```

```
mqtt_client_private_key_path.1 = /opt/ibsnow/conf/certs/72d382zzzz.private.key
#mqtt_client_private_key_password.1 =
#mqtt_client_id.1 =

# The Sparkplug sequence reordering timeout in milliseconds
sequence_reordering_timeout = 5000

# Whether or not to block auto-rebirth requests
#block_auto_rebirth = false

# The primary host ID if this is the acting primary host
#primary_host_id =

# Snowflake streaming connection properties - A custom client name for the connection (e.g. MyClient)
#snowflake_streaming_client_name = IBSNOWClient

# Maximum number of rows to insert into the snowflake_streaming_table at once. Set to 0 for no limit.
snowflake_max_streaming_insert_batch_size = 0

# Maximum number of rows to insert into the snowflake_mqtt_streaming_table at once. Set to 0 for no limit.
snowflake_max_mqtt_streaming_insert_batch_size = 0

# Do raw MQTT inserts one topic at a time
snowflake_topic_based_mqtt_streaming_insert = false

# Ingest task execution period (in seconds)
snowflake_streaming_ingest_task_period = 1

# Raw MQTT ingest task execution period (in seconds)
snowflake_mqtt_streaming_ingest_task_period = 1


# Snowflake streaming connection properties - The scheme to use for channels and their names
# This MUST be one of the following: STATIC, GROUP_ID, EDGE_ID
# STATIC - means to use a single channel. If using this mode, the snowflake_streaming_channel_name
# GROUP_ID - means to use the Sparkplug Group ID for the channel name on incoming data
# EDGE_ID - means to use the Sparkplug Group ID and the Edge Node ID for the channel name on incoming data
# DEVICE_ID - means to use the Sparkplug Group ID, Edge Node ID, and Device ID for the channel name on incoming
data
snowflake_streaming_channel_scheme = EDGE_ID

# Snowflake streaming connection properties - A custom channel name for the connection (e.g. MyChannel)
# If this is left blank/empty, Channel names of the Sparkplug Group ID will be used instead of a single channel
# snowflake_streaming_channel_name =

# Snowflake streaming connection properties - The Table name for Sparkplug data associated with the Database
and Schema already provisioned in the Snowflake account (e.g. MyTable)
snowflake_streaming_table_name = SPARKPLUG_RAW

# Snowflake streaming connection properties - The Table name for RAW MQTT data.
snowflake_mqtt_streaming_table_name = MQTT_RAW

# Maximum number of streaming channels for RAW MQTT messages
# snowflake_max_mqtt_streaming_channels =

# Snowflake notify connection properties - The Database name associated with the connection that is already
provisioned in the Snowflake account (e.g. MyDb)
snowflake_notify_db_name = cl_bridge_node_db

# Snowflake notify connection properties - The Schema name associated with the Database already provisioned in
the Snowflake account (e.g. PUBLIC)
snowflake_notify_schema_name = stage_db

# Snowflake notify connection properties - The Warehouse name associated with the notifications already
provisioned in the Snowflake account (e.g. PUBLIC)
snowflake_notify_warehouse_name = cl_bridge_ingest_wh

# Whether or not to create and update IBSNOW informational tracking metrics
# ibsnow_metrics_enabled = true

# The Sparkplug Group ID to use for IBSNOW asset names
```

```
ibsnow_metrics_sparkplug_group_id = IBSNOW

# The 'Bridge Info' Sparkplug Edge Node ID to use for IBSNOW assets
ibsnow_metrics_bridge_info_sparkplug_edge_node_id = Bridge Info

# The 'Edge Node Info' Sparkplug Edge Node ID to use for IBSNOW assets
ibsnow_metrics_edge_node_info_sparkplug_edge_node_id = Edge Node Info

# The 'MQTT Client Info' Sparkplug Edge Node ID to use for IBSNOW assets
ibsnow_metrics_mqtt_client_info_sparkplug_edge_node_id = MQTT Client Info

# Whether or not to send notification tasks to Snowflake based on incoming Sparkplug events
snowflake_notify_task_enabled = true

# The number of threads to use for BIRTH handling in Snowflake
# snowflake_notify_task_birth_thread_count = 100

# The number of times to retry the NotifyIngest task on failure
snowflake_notify_nbirth_retries = 10

# The number of milliseconds to delay after receiving an NBIRTH before notifying Snowflake over the event
(requires snowflake_notify_task_enabled is true)
snowflake_notify_nbirth_task_delay = 15000

# The number of milliseconds to delay after receiving a DBIRTH or DATA message before notifying Snowflake over
the event (requires snowflake_notify_task_enabled is true)
snowflake_notify_data_task_delay = 5000

# NotifyTimerTask keep alive in seconds. This setting allows to kill inactive threads after keep-alive expires.
snowflake_notify_task_keep_alive = 60

# Enable performance metrics such as 'NotifyIngest thread pool info' and 'Message Inflow Rates' per EdgeNode or
topic.
snowflake_enable_performance_metrics = true
```