**SnowIOT**

**Stream Azure Event Hubs events into Snowflake with Snowpipe Streaming**

**Requirements**

Below are the services used for the setup.

1. Azure Event Hubs (Standard/Premium)
2. Azure VM to run the Kafka Connect in the same region as Azure EH.
3. Snowflake Connector for Kafka 2.0.0
4. Snowflake Account

**Create Azure Event Hubs**

1. Create an Azure Event Hub Namespace with Standard Pricing tier. You can change the Throughput Units and Auto-Inflate Maximum Throughput Units as per your requirement.

A screenshot of a computer

Description automatically generated

1. Go to the Event Hub Namespace created and in the left pane go to Entities and click on Event Hubs. Create an Event Hub instance. You can set the partition count and retention time based on your requirement.

A screenshot of a computer

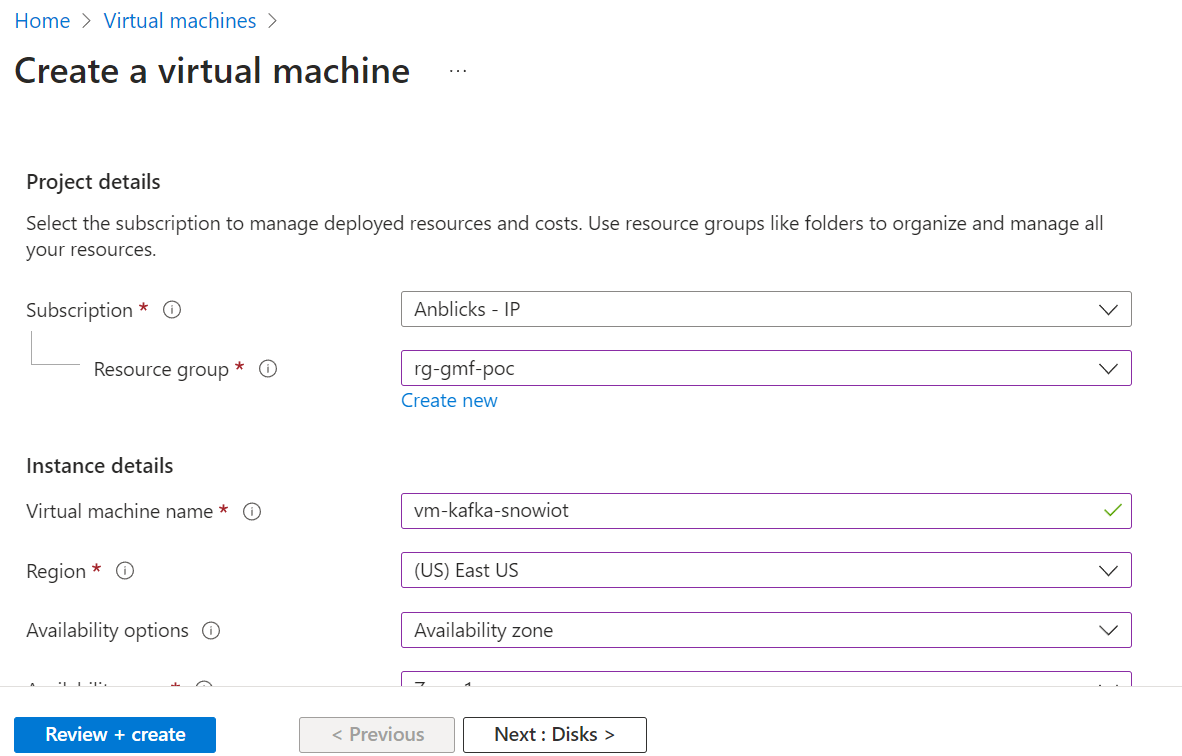
Description automatically generated

1. In the Event Hub namespace left panel click on the “Shared access policies” and save that “Connection string-primary key” for further use in the Producer Logic.
2. Take note of the event hub namespace which “evhns-snowiot” in this case. Also take note of the host name which in this case would be “evhns-snowiot.servicebus.windows.net”. This will be used to specify the value for the bootstrap servers in the Kafka Connect configurations.

**Setup Kafka Connect**

You can run the Kafka Connect on an Azure VM or on a Docker container as well.

1. Create an Azure VM and configure it as per your requirement.



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Make a note of that Public IP Address. You can connect to VM using Putty.
2. Create a downloads folder on your home directory.

>$ mkdir Downloads

1. Run the following command which downloads the Kafka version 3.5.0.

> $curl "https://downloads.apache.org/kafka/3.5.0/kafka\_2.13-3.5.0.tgz" -o ~/Downloads/kafka.tgz

1. Create another folder kafka.

> $ mkdir kafka

1. Change the path to that folder.

> cd /home/{USERNAME}/kafka

1. Extract the tar file by running the below command.

> $ tar -xvzf ~/Downloads/kafka.tgz --strip 1

1. Once Kafka is installed, download the snowflake kafka connect libraries and move them to /home/azureuser/kafka/libs folder. You can run the below commands to download the required libraries. Ensure you are downloading the latest version and at the time of writing this document the latest version was 2.0.0.

> sudo wget https://repo1.maven.org/maven2/com/snowflake/snowflake-kafka-connector/2.0.0/snowflake-kafka-connector-2.0.0.jar -P /home/{USERNAME}/kafka/libs/

> $ sudo wget https://repo1.maven.org/maven2/org/bouncycastle/bc-fips/1.0.1/bc-fips-1.0.1.jar -P /home/{USERNAME}/kafka/libs/

> sudo wget https://repo1.maven.org/maven2/org/bouncycastle/bcpkix-fips/1.0.3/bcpkix-fips-1.0.3.jar -P /home/{USERNAME}/kafka/libs/

1. Ensure java sdk is installed and you can check that by running $ java –version. If Java is not installed, then you can run the below command to install openjdk.

> $ sudo apt-get install openjdk-11-jdk

> $ sudo apt-get update

> $ sudo apt-get install openjdk-11-jdk

1. In the folder /home/{USERNAME}/kafka/config, update the connect-distributed.properties file.



1. In the bootstrap.servers give the Event Hub Namespace name . In the password give the connection string-primary key. At the end change the plugin path.
2. Once you have updated the connect-distribution file you would need to create another file which will have details about the snowflake account details and the Kafka topics name. Create a file with the name SF\_Connect.json. For that we need private key and public key.
3. To generate keys , go to the below path and run the following commands.

> cd /home/{USERNAME}/

> $ openssl genrsa -out rsa\_key.pem 2048

> openssl rsa -in rsa\_key.pem -pubout -out rsa\_key.pub

> cat rsa\_key.pub

> cat rsa\_key.pem

1. Copy the keys for future reference.
2. Set that public key to the Snowflake User.
3. Login to the Snowflake Account and create a Database and a schema. Set the public key to the User in Snowflake using the below query.

alter user {USERNAME} set rsa\_public\_key = '';

1. . Create a file with the name SF\_Connect.json with the following details.



1. topics is your Event Hub instance name. In the topic2table.map, give your table name to which you want to load Streaming data from Event Hub. Give other required details as well.
2. Some additional packages need to be installed for running jq files.

> $ sudo apt install jq

1. Run the below commands as well to install curl packages.

> $ sudo apt-get install curl

> $ sudo apt-get update

1. Start the Kafka connect in distributed mode. Go to the Kafka folder and run the below command.

> $ bin/connect-distributed.sh config/connect-distributed.properties

1. After starting Kafka Connect, for the first time proceed to open a new terminal and execute the following CURL commands. Navigate to the config folder within the Kafka home directory and run the following commands: (no need to run from next time)

> $ curl -X POST -H "Content-Type: application/json" -d @SF\_Connect.json <http://localhost:8083/connectors>

1. If you want to delete the existing sink, run this command.

> $ curl -s -X DELETE "http://localhost:8083/connectors/snowflakesink"

By running through the above steps, you should be able to successfully ingest streaming data from Azure Event Hubs to Snowflake tables.

**References**:

<https://medium.com/snowflake/simplifying-real-time-data-ingestion-stream-azure-event-hubs-events-into-snowflake-with-snowpipe-62010d8f479a>

<https://learn.microsoft.com/en-us/azure/event-hubs/event-hubs-create>

<https://learn.microsoft.com/en-us/azure/event-hubs/event-hubs-kafka-connect-tutorial>

<https://docs.snowflake.com/en/user-guide/kafka-connector-install#distributed-mode>

<https://docs.snowflake.com/en/user-guide/kafka-connector-install#using-key-pair-authentication-key-rotation>