# **Azure Event Hubs events into Snowflake with Snowpipe Streaming:**

To Ingest data from Event hubs to Snowflake using snow pipe streaming we required to step up the connection between Event hubs to snowflake tables using Kafka connector .

Steps to Ingest data from Azure Event Hubs using Snowflake Connector for Kafka

## **Requirements:**

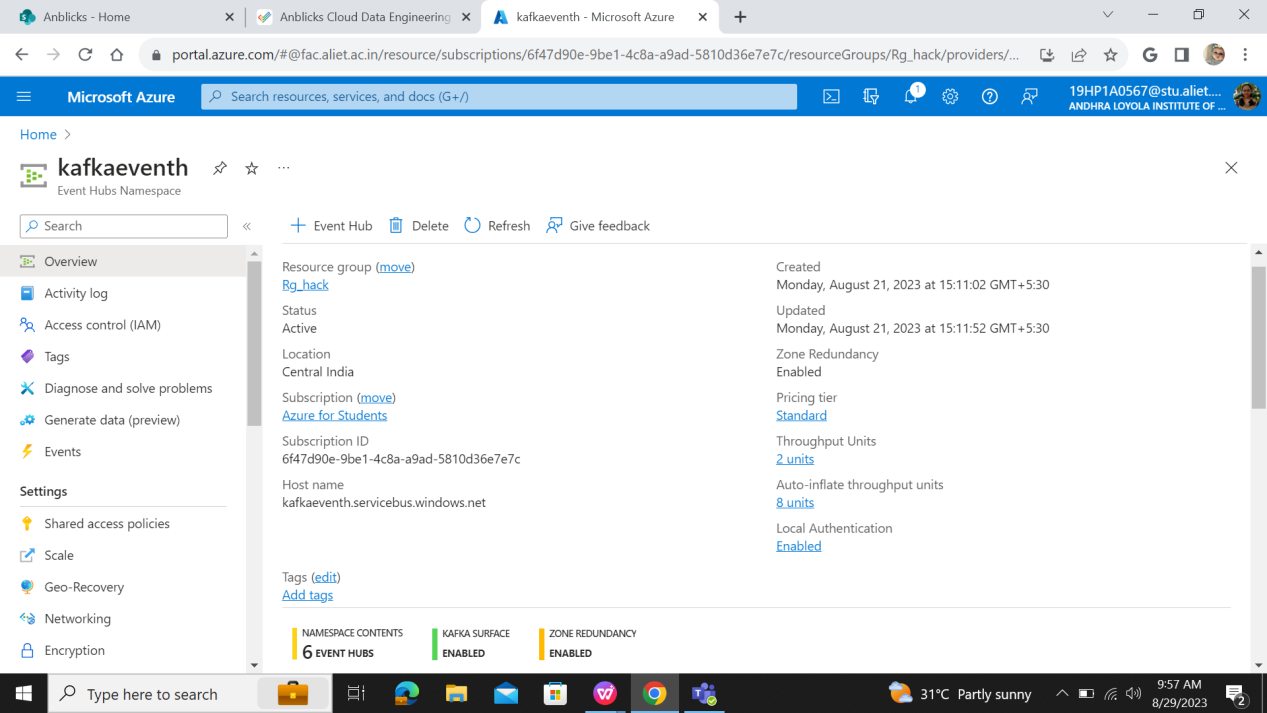
1. Azure Event Hubs ( Standard/Premium )
2. Azure VM to run the Kafka Connect on the same region as Azure Event Hub. You can use a docker or any VM.
3. VM/Docker to push data to Event Hubs.
4. Snowflake Connector for Kafka 2.0.0
5. Snowflake Account

**These are the services used for the setup.**

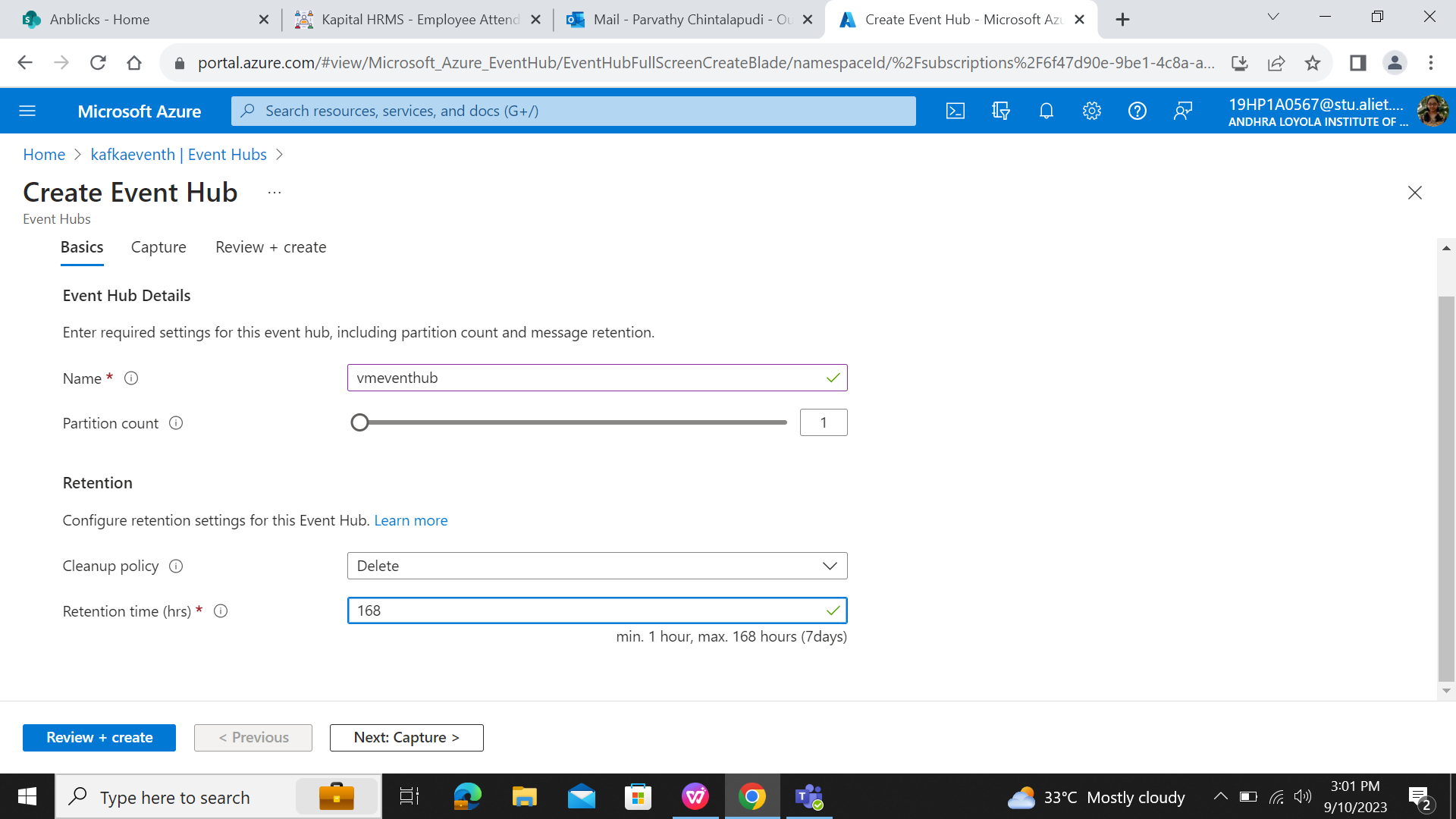
Azure Event Hubs ( Standard/Premium )

--> create Azure Event hub Name Space in Standard Price tier

--> create Azure Event hub in the above name space (go to event hub name space search for ‘entities’ in the left side panel. Under the entities there is “event hub”



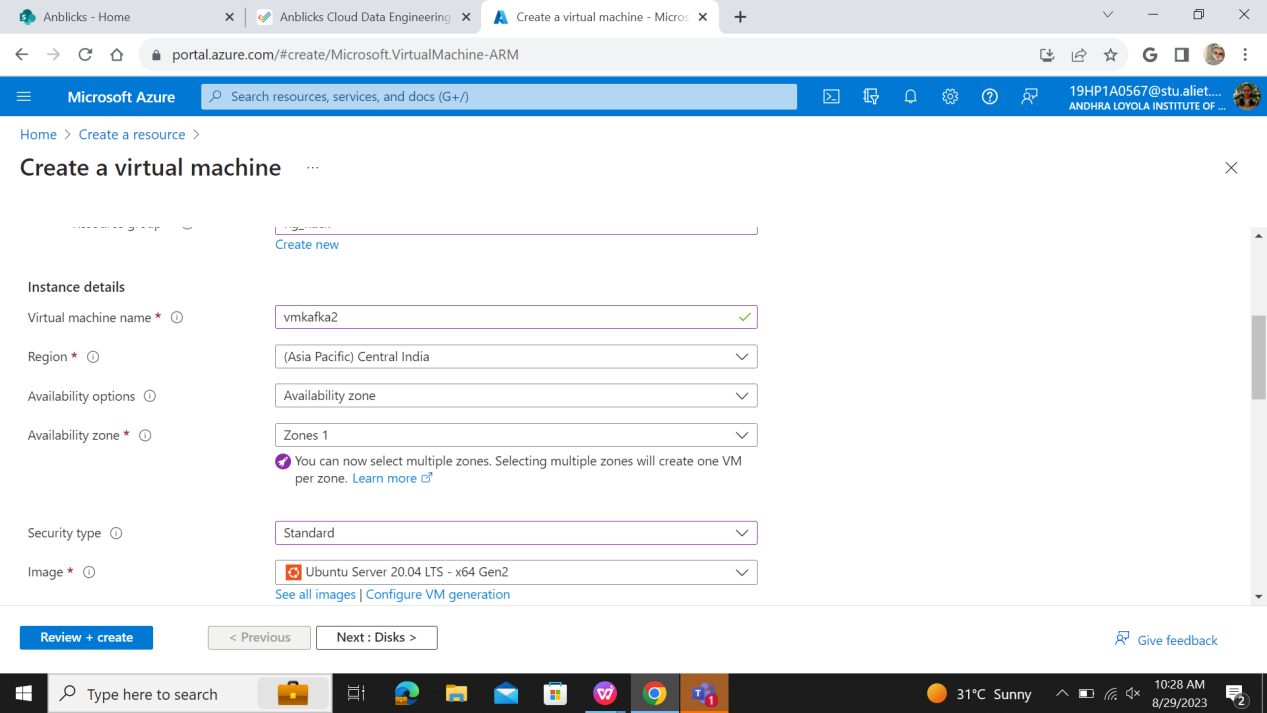
Creating an event hub which stores data as events from iot generator

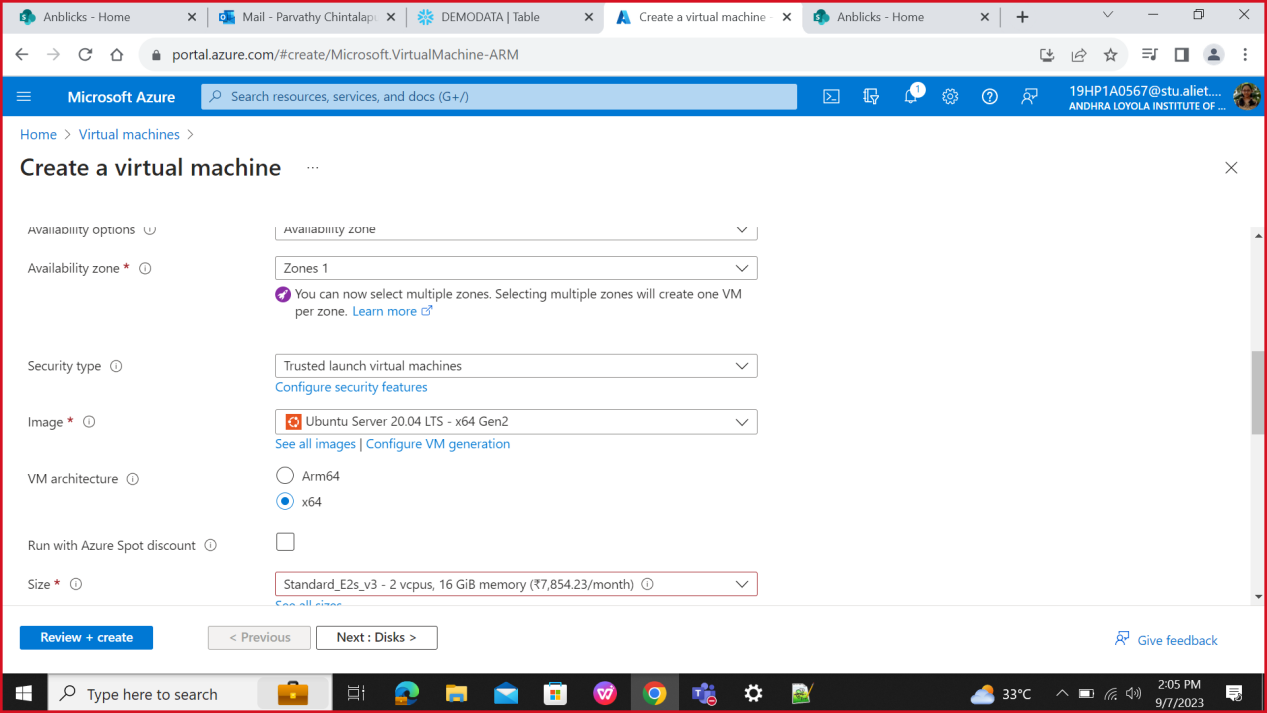


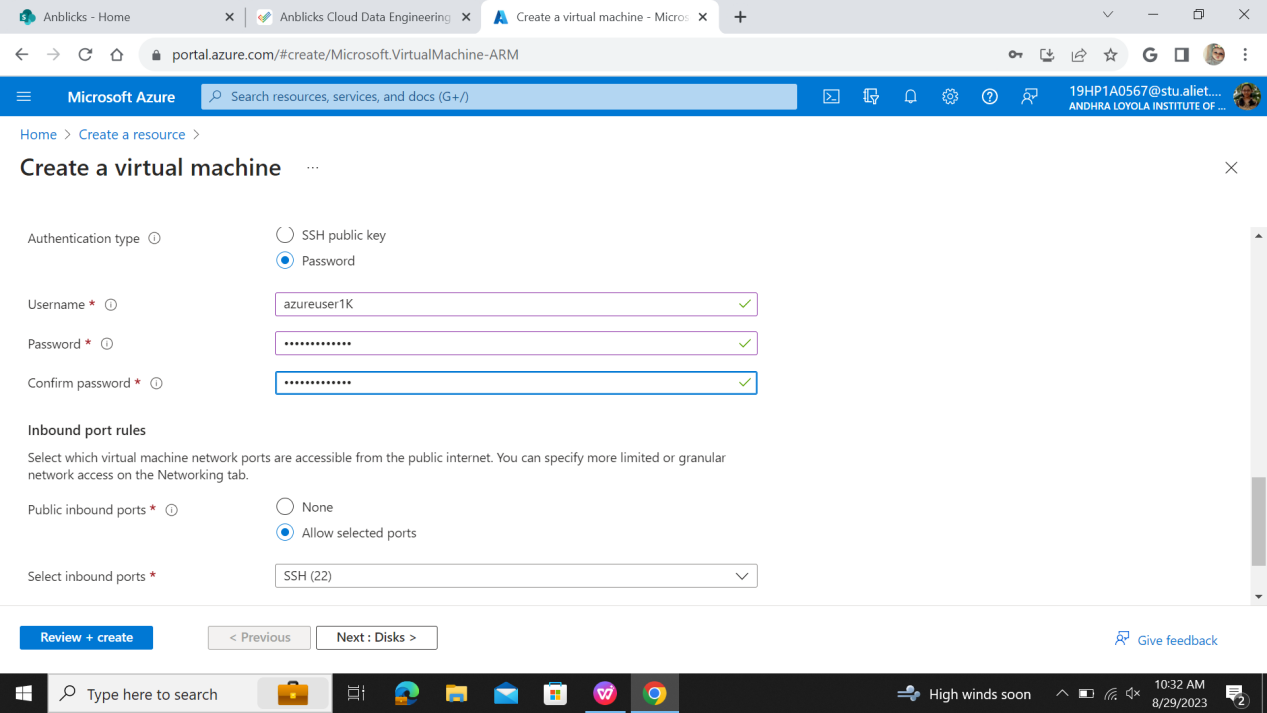
Click review and create to create an event hub.

Azure VM to run the Kafka Connect on the same region as Azure Event Hub. You can use a docker or any VM.

1. VM/Docker to push data to Event Hubs. For this we have used a Azure VM which is on the same region as of EH.







-->click review and create to create a virtual machine

After creating a virtual machine connect to the virtual machine using putty

-->copy the public Ip address of virtual machine place it in putty login page and use your credential to connect virtual machine

-->Download Kafka on virtual machine

Command -->$mkdir Downloads”

-->After creating the folder “Downloads” now you can use command to download kafka tar file

Command-->curl "[https://downloads.apache.org/kafka/3.5.0/kafka\_2.13-3.5.0.tgz"](https://downloads.apache.org/kafka/3.5.0/kafka_2.13-3.5.0.tgz"" \o "https://downloads.apache.org/kafka/3.5.0/kafka_2.13-3.5.0.tgz%22" \t "_blank) -o ~/Downloads/kafka.tgz

--create another folder “<<foldername>” in virtual machine

Here we are creating folder with name ‘kafka’.

Command --> mkdir kafka

Change to the kafka location

-->$ cd /home/azureuser/kafka

-->Now Unzip the tar file in kafka folder

-->Unzip the tar file

Command-->$ tar -xvzf ~/Downloads/kafka\_2.13-3.5.0.tgz --strip 1

-->After unzip the kafka now we have to download some jar files that are used to connect snowflake

To Download jar files we are using putty

Command---> $ sudo wget <https://repo1.maven.org/maven2/com/snowflake/snowflake-kafka-connector/2.0.0/snowflake-kafka-connecctor-2.0.0.jar> -P /home/<vmusername>/kafka/libs

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

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

-->After downloading the jar file now we have to check the there is java available in the vm or not if not install java using command ‘$ sudo apt-get install openjdk-11-jdk after installing the java we need to modify the config file

--> To modify the config file

-->Go to the ‘home/<vmusername>/kafka/config path

-->open the connect-distributed.properties file modify the content with your azure event namespace host name and password is the primary connection string of the namespace.

-->make sure the password ends with ‘;’ (semi-colon)

-->The content which we are changing is available on azure kafka event hub integration page

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

Replace the following parameters with the parameters inside your own account

# e.g. namespace.servicebus.windows.net:9093

bootstrap.servers={YOUR.EVENTHUBS.FQDN}:9093

group.id=connect-cluster-group

# connect internal topic names, auto-created if not exists

config.storage.topic=connect-cluster-configs

offset.storage.topic=connect-cluster-offsets

status.storage.topic=connect-cluster-status

# internal topic replication factors - auto 3x replication in Azure Storage

config.storage.replication.factor=1

offset.storage.replication.factor=1

status.storage.replication.factor=1

rest.advertised.host.name=connect

offset.flush.interval.ms=10000

key.converter=org.apache.kafka.connect.json.JsonConverter

value.converter=org.apache.kafka.connect.json.JsonConverter

internal.key.converter=org.apache.kafka.connect.json.JsonConverter

internal.value.converter=org.apache.kafka.connect.json.JsonConverter

internal.key.converter.schemas.enable=false

internal.value.converter.schemas.enable=false

# required EH Kafka security settings

security.protocol=SASL\_SSL

sasl.mechanism=PLAIN

sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule required username="$ConnectionString" password="{YOUR.EVENTHUBS.CONNECTION.STRING}";

producer.security.protocol=SASL\_SSL

producer.sasl.mechanism=PLAIN

producer.sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule required username="$ConnectionString" password="{YOUR.EVENTHUBS.CONNECTION.STRING}";

consumer.security.protocol=SASL\_SSL

consumer.sasl.mechanism=PLAIN

consumer.sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule required username="$ConnectionString" password="{YOUR.EVENTHUBS.CONNECTION.STRING}";

plugin.path=/home/<vmusername>/kafka/libs(change the path according yours)

-->After the changes save the file

-->Now we need to create an additional file in the same loaction where

Home/<vmsusername>/kafka/config

-->using the command

$ touch SF\_Connect.json

-->open the file add your snowflake details.The changes we are doing are get from the below link.

[https://docs.snowflake.com/en/user-guide/kafka-connector-install#distributed-mode](https://docs.snowflake.com/en/user-guide/kafka-connector-install" \l "distributed-mode)

{  
"name":"snowflakesink",  
"config":{  
"connector.class":"com.snowflake.kafka.connector.SnowflakeSinkConnector",  
"tasks.max":"16",  
"topics":"vehicle\_partition",

"snowflake.topic2table.map": "vehicle\_partition:raw\_data"  
"buffer.count.records":"1000",  
"buffer.flush.time":"2",  
"buffer.size.bytes":"2000000",  
"snowflake.url.name":"ana95816.snowflakecomputing.com:443",  
"snowflake.user.name":"snowflake\_user\_namr",  
"snowflake.private.key":"MIIEowIBAAKCAQEAsJFqSunLz+pA016RVK",  
"snowflake.database.name":"snowpipe\_streaming",  
"snowflake.schema.name":"dev",  
"key.converter":"org.apache.kafka.connect.storage.StringConverter",  
"value.converter":"org.apache.kafka.connect.json.JsonConverter",  
"value.converter.schemas.enable":"false",  
"snowflake.role.name":"accountadmin",  
"snowflake.ingestion.method":"SNOWPIPE\_STREAMING"  
}  
}

-->mention your snowflake details like ‘account url’ ,‘username’,’databasename’,

‘schemaname’ and topics

--> The topics name is your eventhub name you have to mention in the above script

-->snowflake.topic2table.map : Table to which the data to load

-->for snowflake.private.key we need to generate the private key and public key form the terminal.

[https://docs.snowflake.com/en/user-guide/kafka-connector-install#using-key-pair-authentication-key-rotation](https://docs.snowflake.com/en/user-guide/kafka-connector-install" \l "using-key-pair-authentication-key-rotation)

In the above link we have command to generate snowflake private key and public key

--> open the putty

-->$ cd /home/<vmusername>

-->$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 (This command will show the keys on terminal)

-->After downloading the both key we need to set the public key to the snowflake user

Which account we are trying to load the data from event hub

-->open snowflake create a database and schema

-->To set the public key for the user

-->alter user <<username>> set rsa\_public\_key=”your public key that is generated using openssl’’

--> after setting the public key mention your private key in SF\_Connect.json file

-->after completing the changes save the files.

-->open server.properties file in your kafka

-->change the advertised.listners property :your.host.name:9092

To “your vm public ip address”:9093

--> Now we have to install some additional package that are require to run the command

-->install jq for running json files

install jq for running.jq command

---insatlling

$ sudo snap install jq

$ sudo apt install jq

-->choose any one of the command to install ‘jq’

--> $ sudo apt-get update

-->$ sudo apt-get install curl

--> To run the curl commands.

--->Now we need to establish the connection between kafka and snowflake

Open putty and connect to vm

--> $ cd /home/<vmusername>/kafka

-->starting zookeeper

$bin/zookeeper-server-start.sh config/zookeeper.properties

-->after running the above command open another putty session

Run tha connect -distributed file

--> $ cd /home/<vmusername>/kafka

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

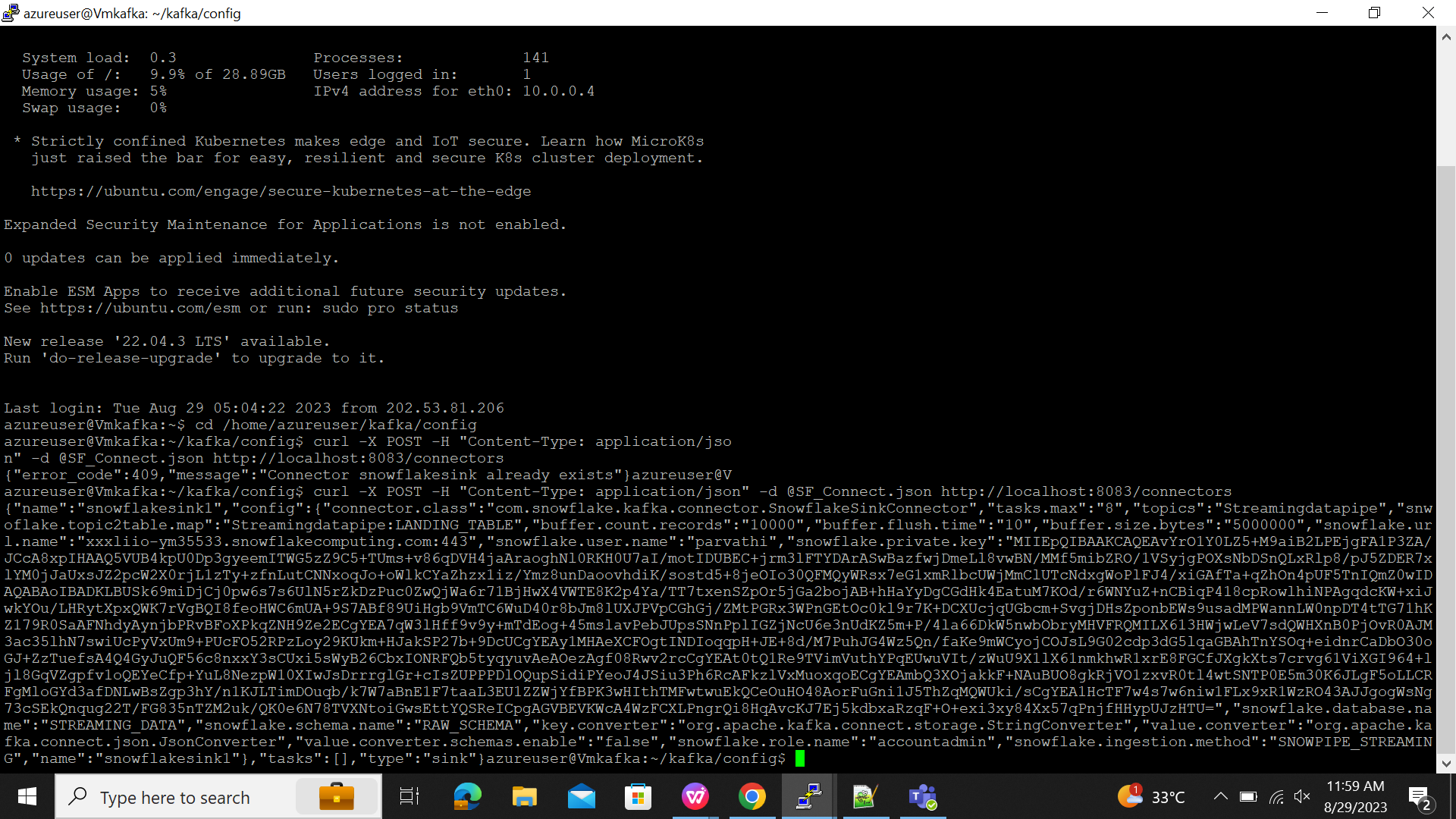
$bin/connect-distributed.sh config/connect-distributed.properties

-->After starting Kafka Connect, proceed to open a new terminal and execute the following CURL commands.

-->$ home/<vmusername>/kafka/config

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

-->If the above command run successful it will give you the contents in the SF\_connect.json file like this



-->To check all the connectors, run the following command

--> $ curl -s -X GET “http://localhost:8083/connectors/”

-->To check the name of the topic for the connector

--> $ curl -s -X GET “http://localhost:8083/connectors/snowflakesink/topics” | jq ‘.

{

"snowflakesink": {

"topics": [

"demoevh"

]

}}

--> To delete a connector, run the following 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.