**Confluent Rundown:**

There are two main flows of data within the Kafka Cluster

1. Applications Data
   1. The mongodbsource Connector tracks changes on the Events.ConfluentTestingSource collection
   2. The data is sent to the topic **named mongodbserver18.Events.ConfluentTestingSource**
      1. The naming convention for topics is [prefix].Database.Collection
      2. When this moves into production, it will likely be named [prefix].WhiteLabelApplicationsDb.Applications
   3. The messages in this topic are all grabbed by the **APPLICATIONSV6** kSQL Stream
      1. After some filtering and sorting, the messages are filtered into their proper buckets based on what kind of event they are
         1. i.e. ApplicationCreated, InterviewStarted, etc…
      2. There is a stream for every possible event type
   4. Each stream has an associated topic, and each associated topic has a sink connector that pulls messages from the topic to update Events.ApplicationStatusV2
      1. All of the events are recorded in the EVENTLOG3 stream, and a sink connector pulls messages from its associated topic to populate Events.EventLog
2. *Policy Data (* ***needs to be updated)***
   1. *The sql-server-connector tracks changes on the PPLOC table that currently resides within the DebeziumTest database on my (Jack Svoboda) localhost SQL Server*
      1. *This PPLOC table is exactly the same as the one in DatamartD1, aside from the fact it has only 20 rows*
   2. *The data is sent to the topic named mongodbserver18.Events.ConfluentTestingSource*
      1. *The naming convention for topics is Database.dbo.Table*
      2. *When this moves into production, it will likely be named [prefix].Whit*
   3. *The messages in this topic are all grabbed by the TESTFFF kSQL Stream*
      * 1. *TESTFFF is a poor naming choice and will be changed soon*
      1. *After some filtering and sorting, the messages are filtered into their proper buckets based on what kind of event they are*
         1. *i.e. Activated, Suspended, etc…*
   4. *Each stream has an associated topic, and each associated topic has a sink connector that pulls messages from the topic to update Events.ApplicationStatusV2*
      1. *All of the events are recorded in the EVENTLOG3 stream, and a sink connector pulls messages from its associated topic to populate* *Events.PolicyStatusEventLog*

* The majority of the topics are topics associated with a kSQL stream. If it starts with pksqlc-d1yn1 (the prefix for the kSQL cluster), it is a topic linked to whatever stream follows
* The majority of the connectors are Sink Connectors. All sink connectors are Mongodb Sink Connectors
* There are 3 different Source Connectors: Mongodb, SQL Server, and HTTP.
* The mongdbSinkApplicationCreated, mongodbSinkEventLog, and mongodbSinkPolicyStatusEventLog connectors are all connectors that insert new documents. The remaining connectors *upsert*. They update existing documents based on finding a matching id. The id that they match to is set in the connector configs ***(Add list if possible)***