PDP-2: Flexibility on ingest schema & format

PNDA-4108 - Greater flexibility on ingest schema & format IN PROGRESS

Motivation

- PNDA depends on metadata to be provided in order to organize data storage & processing
- Currently, this metadata must be provided in a specific fixed format - a defined Avro schema
- Not everyone wants to use Avro or incur the overhead of converting data formats on ingest
- Therefore, it would be beneficial to have a mechanism that allows a range of common formats, continuing to mandate that the metadata required by PNDA *is* supplied but not being needlessly restrictive in *how* it's supplied

Backwards compatibility:

KiteSDK? No (Is not used by any of our examples or the community as far as we know)

Path structure?

- Prefix? Yes
- Source? Yes
- Time? Yes

Use original Envelop for output? No

Consistency on the output schema? Yes.

Conclusion: This change require a major release number increment as the output schema will be altered.

Proposal

- Introduce concept of the PNDA canonical output datum schema (are old fields still needed past the HDFS path creation? answer was no if they are mapped, so we will use a simplified version) irrespective of the data type.
  Use a simplified version but consistent:
  - preserve host_ip: no (remove)
  - preserve src: no (rename to source)
  - preserve timestamp: yes
- Introduce a configuration entity:
  - That maps a topic to a family_id:
  - That maps a family_id to:
    - A Converter class name (PNDAAvroConverter, PNDAProtoBufConverter, ...)
    - A Converter class specific (mapping) configuration:
      - AVRO: schema, timestamp field name, source field name
      - Protobuf: timestamp field index, source field index
  - Support topic configuration/matching based on regex
- Modify ingest to use this new configuration:
  - Extract the configuration based on the tropic
  - Ensure the record has an envelope that complies with the canonical output datum schema, create the envelop if needed
    - Topics that don't have a configuration entry will the data inserted in the PNDA envelop with source set to topic name and timestamp to ingest time. (back filling these entries will be a requirement when upgrading pnda)
    - Data that doesn't match the topic configuration will go into quarantine.
  - Use the mapper to extract the envelop data
- Modify ingest to add support for Protocol Buffers as an alternative to Avro
- ~~Support basic schema version increment of the datum (this would probably require some indicator in the wire protocol and hence would break backwards compatibility or complicate ingest compliance to PNDA)~~

Plan

- Phase 0: Design (~~PNDA-4184~~)
  - Define and document the details of the design choices including workflows.
  - Define PNDA canonical output datum schema
  - Define pnda.* metadata properties to store the input configuration in the avro file header metadata.

- Phase 1: Using the existing Kite implementation (~~PNDA-4384~~)
  - Accept avro data and extract source/timestamp for use in standard envelope
  - Accept gpb data and extract source/timestamp for use in standard envelope
  - Accept any other data and create ingest timestamp & use topic name in standard envelope
  - Handle topics that miss either if the extraction data fields
  - What to map from input data is based on configuration from a stubbed client implementation
    - Define configuration data structure
  - Pass the extraction properties to the data writer (Kite)

    - Deployable on feature flag instead of regular ingest (different class in modules, feature flagged in salt)

- Phase 2: Substitute Kite implementation (~~PNDA-4464~~)
    - for something else that gives functionally identical partitioning & file creation
    - allows to recuperate extraction properties and inject then in the avro envelop header meta data
    - allows path flexibility depending on the source's field origin

- Phase 3: Create data family & ingest configuration registry

    a. Simple externalized configuration of ingest (~~PNDA-4553~~)
        - User assigned Family ID
        - Use as fallback when no configuration is found in the Registry Service.
    b. Externalized ingest configuration registry service (PNDA-4465)
        - Organize data in suitable registry (use the integrated consul, Extending https://github.com/confluentinc/schema-registry, ...)
        - Mechanism to create data in registrar, issue family IDs (a controlled list per deployment), map to topics, etc
        - Fill out the client implementation to use registry instead of stubbed data
        - Expose with REST API

- Create CLI to configure the configuration registry (PNDA-4466)

- Adapt the avro envelop schema. This will break backwards compatibility (~~PNDA-4609~~):
    - Drop the unused 'host_ip' field
    - Rename 'src' field to the 'source' field

- Update existing examples and update platform-libraries (PNDA-4503)
    - to work with revised ingest mechanics (extract raw data, ...)
    - to work with new schema

- Create GUI to configure the configuration registry (PNDA-4467)

- Add usage documentation:
    - Simple registry config (PNDA-4468)
    - Registry service
        - UI
        - CLI
    - Platform-libraries

- Add example code:
    - Add example Kafka producer with different (then output) avro schema.
    - Add example Kafka producer that pushes protobuf content
    - Add example dataset consumer that interprets the avro envelop info (metadata and fields)
        - and extracts the protobuf input data
        - and extracts the avro input data

Interfaces

- Adds support for varied ingest schemas in Avro and Protocol Buffer formats
- Adds support for topic configuration