



Building Apache NiFi 2.0 Python Processors

Tim Spann
Principal Developer Advocate

Feb 29, 2024



CONF42

Tim Spann

Twitter: @PaasDev // Blog: datainmotion.dev

Principal Developer Advocate.

Princeton Future of Data Meetup.

ex-Pivotal, ex-Hortonworks, ex-StreamNative, ex-PwC

<https://medium.com/@tspann>

<https://github.com/tspannhw>



 DZone REF CARDS TREND REPORTS EXPERTS

Top IoT Experts

 Tim Spann
Principal Developer Advocate,
Cloudera

<https://github.com/tspannhw/SpeakerProfile/>
Tim Spann is a Principal Developer Advocate in Data in Motion for Cloudera. He works with Apache NiFi, Apache Pulsar, Apache...



FLaNK Stack Weekly by Tim Spann



<https://bit.ly/32dAJft>

<https://www.meetup.com/futureofdata-princeton/>



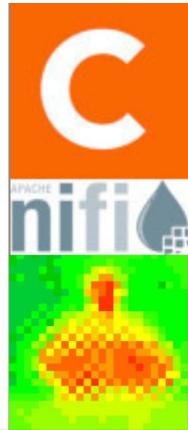
This week in Apache NiFi, Apache Flink, Apache Kafka, ML, AI, Apache Spark, Apache Iceberg, Python, Java and Open Source friends.

Future of Data - NYC + NJ + Philly + Virtual



<https://www.meetup.com/futureofdata-princeton/>

From Big Data to AI to Streaming to Containers to Cloud to Analytics to Cloud Storage to Fast Data to Machine Learning to Microservices to ...



CLOUDERA



@PaasDev

Apache NiFi has emerged as a robust and flexible platform for designing data integration and flow management solutions. With the release of Apache NiFi 2.0, the community has introduced a host of new features, making it even more powerful and extensible. One exciting enhancement is the ability to create custom processors using Python, providing a seamless integration of Python scripts into your data flow.

In this talk, I will delve into the world of Apache NiFi 2.0 Python processors, exploring the capabilities they offer and demonstrating how to build custom processors to enhance your data processing pipelines. Attendees will gain a deep understanding of the integration points between NiFi and Python, enabling them to leverage the extensive libraries and frameworks available in the Python ecosystem.

- Introduction to Apache NiFi 2.0 – Python Processors
- Deep Dive – Build your own custom Python Processor – Integrating Python Libraries and Frameworks – Debugging and Troubleshooting

By the end of this talk, participants will have a comprehensive understanding of building and optimizing Apache NiFi 2.0 Python processors, enabling them to integrate Python seamlessly into their data processing workflows. This session is suitable for data engineers, architects, and anyone interested in harnessing the combined power of Apache NiFi and Python for efficient data integration and flow management.

Let's enhance real-time streaming pipelines with smart Python code. Adding code for vector databases and LLM.

The event is a **series of pre-recorded videos**, broadcasted on our YouTube channel.

Your talk can go from **15 minutes** (lighting talk), through **30 min** (regular talk), up to **60+ min** (in-depth/with demo). We are open to any type of session

Once you are approved as a speaker, **we will request you to record your talk**. You can use any available recording tool, although we recommend OBS Studio (open source and free, compatible with iOS, Windows, and Linux). We'll provide you with a tutorial for your convenience (set everything up in under 15 minutes)

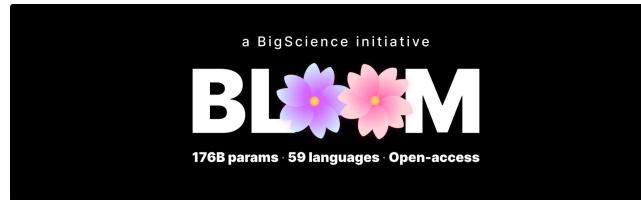
If you'd like to deliver a workshop, you can present it at a conf42 event. Due to pre-recorded nature of our events, you will need to transform it into a hands-on tutorial.

Please **avoid submitting the exact same talk** that has already been presented at a past conf42. Instead, we encourage you to **provide a continuation of your previous content** (2.0 version) or **offer a fresh new perspective on the same topic**. For example, if you previously made a theoretical overview of a tool, consider giving a hands-on demo this time.

The clarity of your voice is the most important technical aspect of a talk, lots of people will listen to it as a podcast / in the background. We recommend you use the best microphone available

Generative AI

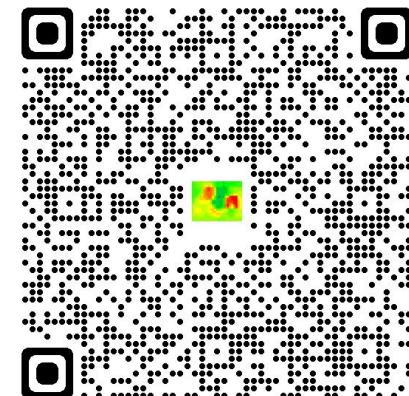
<https://github.com/tspannhw/FLaNK-HuggingFace-DistilBert-SentimentAnalysis>



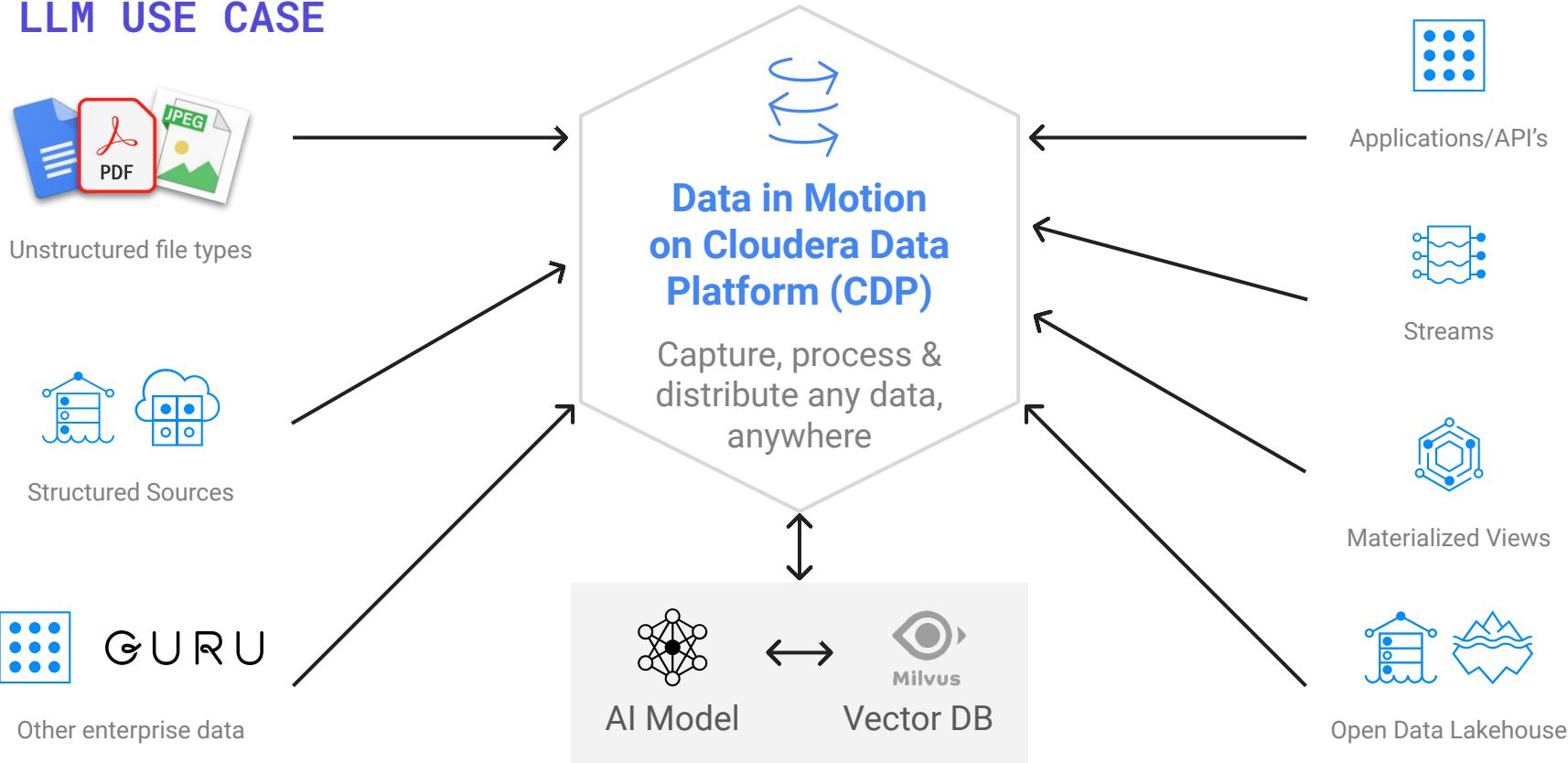
watsonx.ai

CLOUDERA
Machine Learning

<https://github.com/tspannhw/FLaNK-LLM>

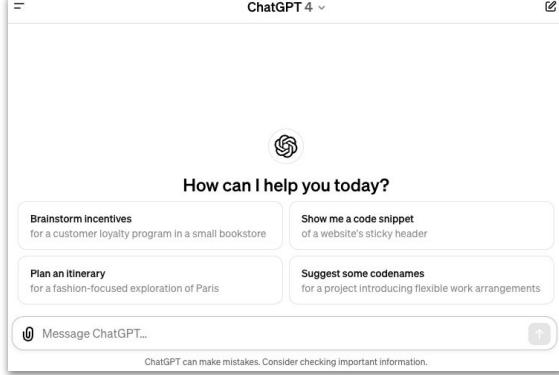


LLM USE CASE



GENERATIVE AI CAPABILITY

Common Families of Generative AI Capability



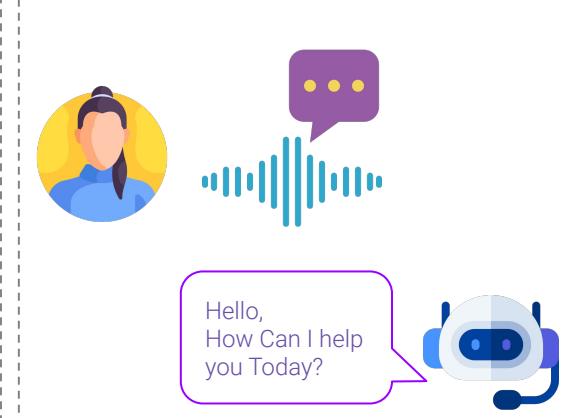
The screenshot shows the ChatGPT 4 interface. At the top, it says "ChatGPT 4". Below that is a message input field with a speech icon and the placeholder "How can I help you today?". There are four suggested prompts: "Brainstorm incentives for a customer loyalty program in a small bookstore", "Show me a code snippet of a website's sticky header", "Plan an itinerary for a fashion-focused exploration of Paris", and "Suggest some codenames for a project introducing flexible work arrangements". At the bottom is a text input field with a microphone icon and the placeholder "Message ChatGPT...". A note at the bottom says "ChatGPT can make mistakes. Consider checking important information."

Text-to-Text
CONVERSATIONAL CHATBOTS
e.g. ChatGPT, Falcon, LLaMA



A vibrant, futuristic cityscape at dusk or dawn. The sky is filled with colorful clouds and several flying vehicles. In the foreground, there's a modern building with a blue roof and a road with blurred lights from moving vehicles. The city features many tall, uniquely shaped skyscrapers with glowing windows.

Text-to-Image
AI ASSISTED DESIGN
e.g. DALL-E, Midjourney



An illustration depicting the Text-to-Speech family of AI. It shows a circular profile of a person's head, a purple speech bubble with three yellow dots, a blue waveform, and a white and blue robot head with a microphone. A purple speech bubble contains the text "Hello, How Can I help you Today?".

Text-to-Speech
INTERACTIVE VOICE RESPONSE

RAPID INNOVATION IN THE LLM SPACE

Too much to cover today.. but you should know the common LLMs, Frameworks, Tools

Notable LLMs

Closed Models	Open Models
 OpenAI GPT3.5 GPT4	 Meta AI Llama2 Code Llama
 Claude2	 MISTRAL AI Mistral7B Mixtral8x7B

++ 100s more... check out the HuggingFace LLM Leaderboard (pretrained, domain fine-tuned, chat models, ...)

Popular LLM Frameworks

LangChain	Llamaindex
 LangChain Langchain is a framework for developing apps powered by LLMs <ul style="list-style-type: none">• Python and JavaScript Libraries• Provides modules for LLM Interface, Retrieval, & Agents	 Llamaindex Llamaindex is a framework designed specifically for RAG apps <ul style="list-style-type: none">• Python and JavaScript Libraries• Provides built in optimizations / techniques for advanced RAG

When to use one over the other? Use Langchain if you need a general-purpose framework with flexibility and extensibility. Consider Llamaindex if you're building a RAG only app (retrieval/search)

Some common Vector DBs

 Chroma	 Pinecone	 Solr
 pgvector	 Milvus	 Weaviate

Open Source vs Self Hosted vs SaaS option

Open Community & Open Models

Hugging Face	HuggingFace
 Hugging Face HuggingFace is an ML community for hosting & collaborating on models, datasets, and ML applications <ul style="list-style-type: none">• Latest open source LLMs are in HuggingFace• + great learning resources / demos	https://huggingface.co/

Cloudera Generative AI Stack

Applied Machine Learning Prototypes (AMPs)

APPLICATIONS

CLOSED-SOURCE FOUNDATION MODELS

APIs: OpenAI (GPT-4 Turbo)
Amazon Bedrock: Anthropic (Claude 2), Cohere...

Hugging Face



FINE-TUNED MODELS

Meta (Llama 2)

OPEN SOURCE FOUNDATION MODELS

Pinecone

MANAGED VECTOR STORE



Milvus, Solr*

PRIVATE VECTOR STORE

CLOUDERA
Open Data Lakehouse

- ▶ REAL-TIME DATA INGEST & ROUTING
- ▶ DATA WRANGLING
- ▶ DATA STORE & VISUALIZATION
- ▶ AI MODEL TRAINING & SERVING

CLOUD INFRASTRUCTURE



SPECIALIZED HARDWARE



NVIDIA



DELL Technologies

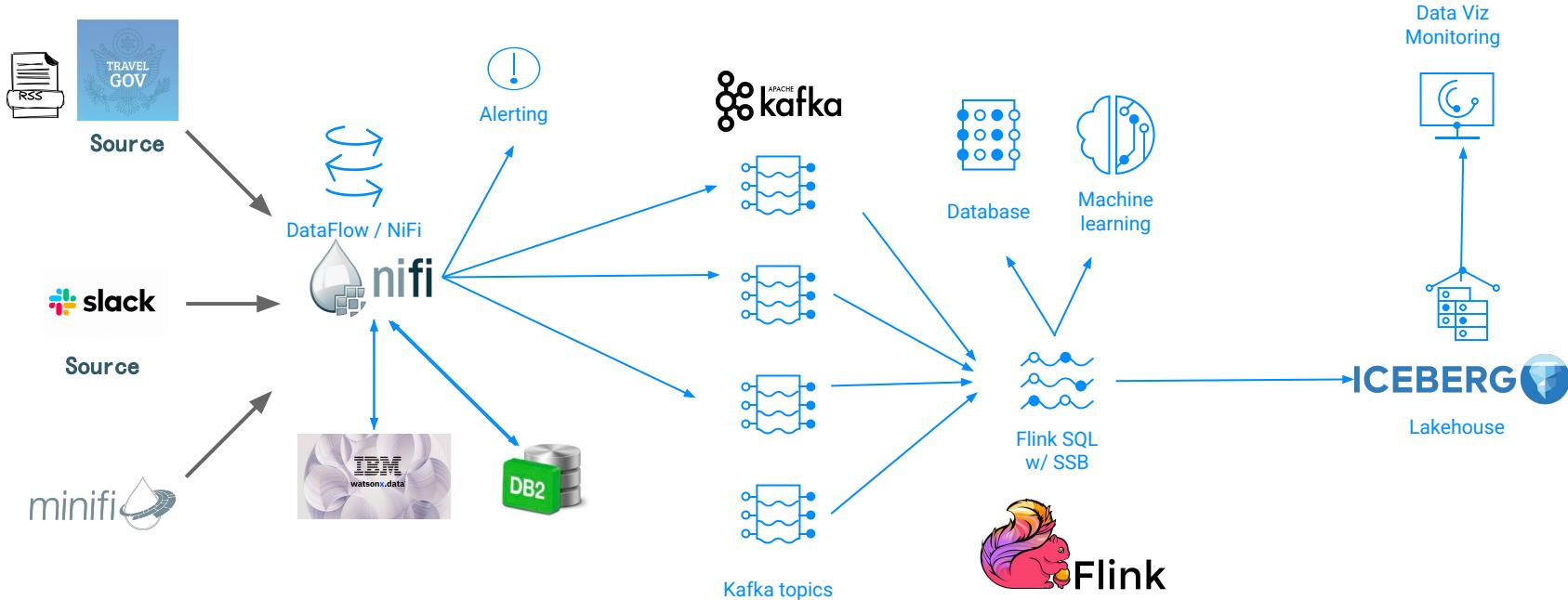


Apache NiFi And Real-Time GenAI



Architecture in the context of Travel Advisories

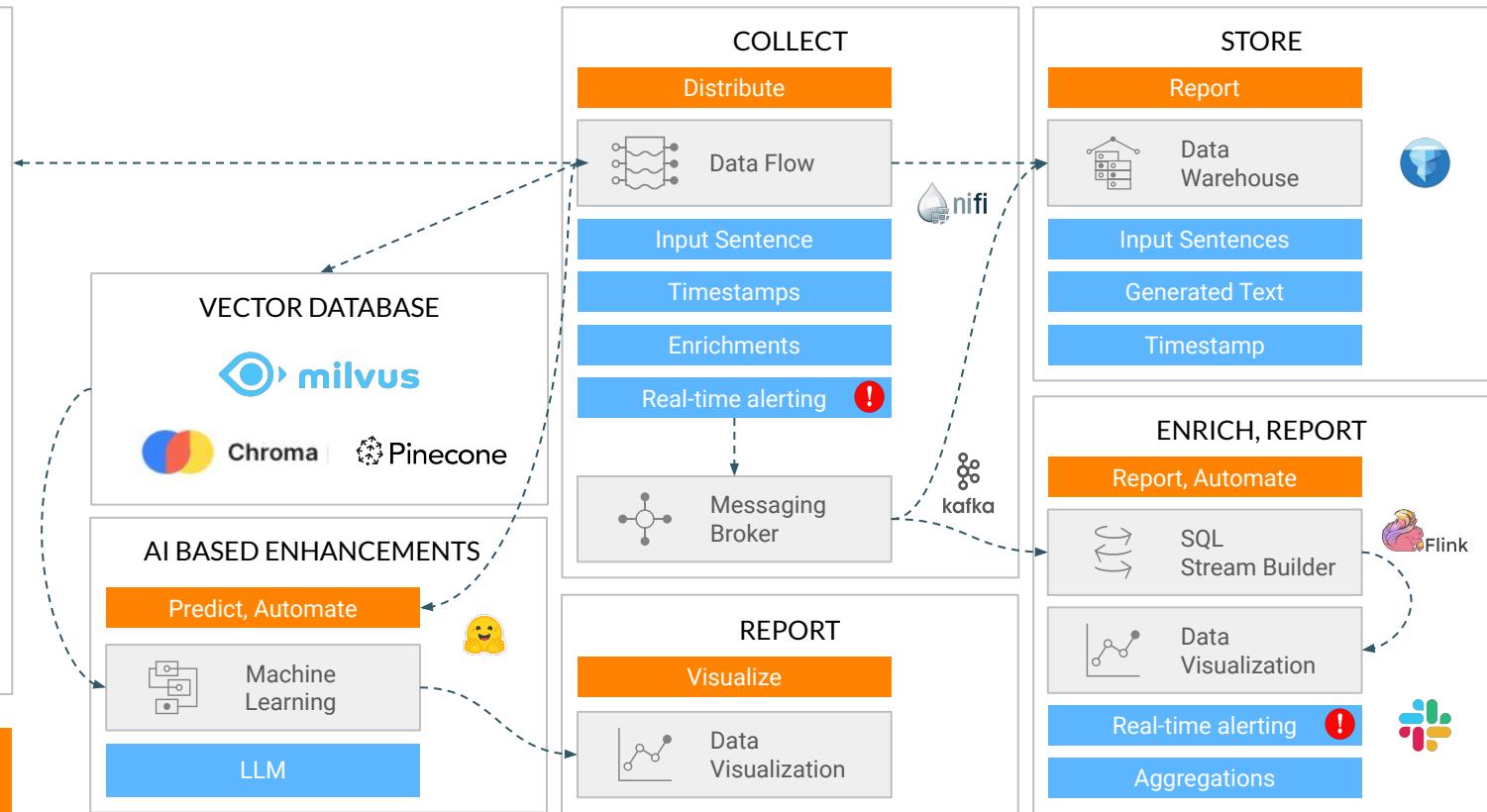
WatsonX.AI Granite LLM, NiFi, Kafka & Flink



INTERACT

- Live Q&A
 - Travel Advisories
 - Weather Reports
 - Documents
 - Social Media
 - Databases
 - Transactions
 - Public Data Feeds
 - S3 / Files
 - Logs
 - ATM Data
 - Live Chat
 - ...
- Collect

HYBRID CLOUD



ReadyFlow Gallery

Leverage pre-built flow templates to quickly customize and deploy new data flows

ReadyFlow Gallery



ADLS to ADLS Avro Version 1
Consumes JSON, CSV or Avro files from source ADLS location and writes Avro files to a destination ADLS location.
[Add To Catalog](#)



Azure Event Hub to ADLS Version 2
Consumes JSON, CSV or Avro events from Azure Event Hub and writes JSON, CSV or Avro files to ADLS.
[Add To Catalog](#)



JDBC to S3/ADLS Version 1
Consumes data from a database table and writes JSON, CSV or Avro files to S3 or ADLS.
[Add To Catalog](#)



Non-CDP ADLS to CDP ADLS Version 1
Consumes files from source non-CDP ADLS location and writes them to a destination CDP ADLS location.
[Add To Catalog](#)



Confluent Cloud to S3/ADLS Version 1
Consumes JSON, CSV or Avro events from Confluent Cloud Kafka and writes them to S3 or ADLS.
[Add To Catalog](#)



Confluent Cloud to Snowflake Version 1
Consumes JSON, CSV or Avro events from Confluent Cloud Kafka and writes them into Snowflake DB.
[Add To Catalog](#)



Non-CDP S3 to CDP S3 Version 2



ListenHTTP filter to Kafka Version 1

Cloudera + LLMs

LLM Serving
Serving Framework

LLM Fine Tuning Process
Training Framework

Vector DB

Data Preparation
Data Engineering

Knowledge Repository
Data Storage / Management



Streaming Classification
Real-Time Model Deployment



Key:



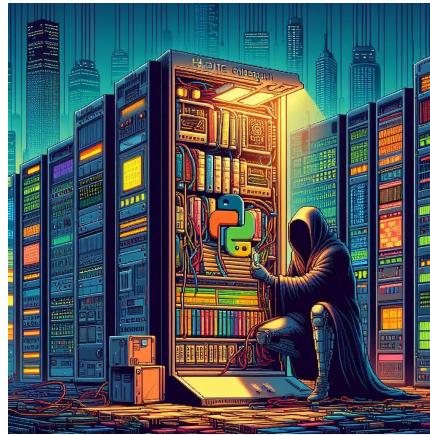


NiFi 2.0.0 Features

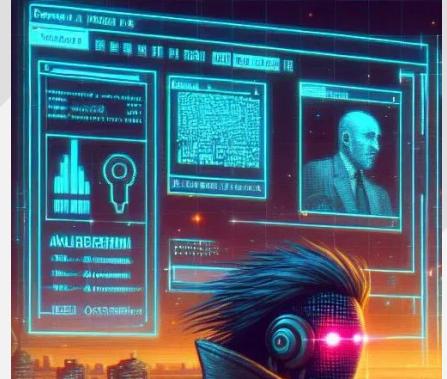
- Python Integration
- Parameters
- JDK 21+
- JSON Flow Serialization
- Rules Engine for Development Assistance
- Run Process Group as Stateless
- flow.json.gz

<https://cwiki.apache.org/confluence/display/NIFI/NiFi+2.0+Release+Goals>

<https://medium.com/cloudera-inc/getting-ready-for-apache-nifi-2-0-5a5e6a67f450>



Python Processors



Basics

```
from nifiapi.flowfiletransform import FlowFileTransform, FlowFileTransformResult
from nifiapi.properties import PropertyDescriptor, StandardValidators, ExpressionLanguageScope

class CallWatsonXAI(FlowFileTransform):
    class Java:
        implements = ['org.apache.nifi.python.processor.FlowFileTransform']

    class ProcessorDetails:
        version = '2.0.0-SNAPSHOT'
        description = """Output results of call to WatsonX.AI"""
        tags = ["ibm", "WatsonX", "WatsonXAI", "generativeai", "ai", "artificial intelligence", "ml", "machin
```

```
property_descriptors = [
    PROMPT_TEXT,
    WATSONXAI_API_KEY,
    WATSONXAI_PROJECT_ID
]

def __init__(self, **kwargs):
    super().__init__()
    self.property_descriptors.append(self.PROMPT_TEXT)
    self.property_descriptors.append(self.WATSONXAI_API_KEY)
    self.property_descriptors.append(self.WATSONXAI_PROJECT_ID)

def getPropertyDescriptors(self):
    return self.property_descriptors
```

Basics

```
PROMPT_TEXT = PropertyDescriptor(  
    name="Prompt Text",  
    description="Specifies whether or not the text (including full prompt with context) to send",  
    required=True,  
    validators=[StandardValidators.NON_EMPTY_VALIDATOR],  
    expression_language_scope=ExpressionLanguageScope.FLOWFILE_ATTRIBUTES  
)  
WATSONXAI_API_KEY = PropertyDescriptor(  
    name="WatsonX AI API Key",  
    description="The API Key to use in order to authentication with IBM WatsonX",  
    sensitive=True,  
    required=True,  
    validators=[StandardValidators.NON_EMPTY_VALIDATOR],  
    expression_language_scope=ExpressionLanguageScope.FLOWFILE_ATTRIBUTES  
)  
WATSONXAI_PROJECT_ID = PropertyDescriptor(  
    name="WatsonX AI Project ID",  
    description="The Project ID for WatsonX",  
    sensitive=True,  
    required=True,  
    validators=[StandardValidators.NON_EMPTY_VALIDATOR],  
    expression_language_scope=ExpressionLanguageScope.FLOWFILE_ATTRIBUTES  
)
```

Basics

```
def transform(self, context, flowfile):
    from ibm_watson_machine_learning.foundation_models.utils.enums import ModelTypes
    from ibm_watson_machine_learning.foundation_models import Model

    prompt_text = context.getProperty(self.PROMPT_TEXT).evaluateAttributeExpressions(flowfile).getValue()
    watsonx_api_key = context.getProperty(self.WATSONXAI_API_KEY).evaluateAttributeExpressions(flowfile)
    project_id = context.getProperty(self.WATSONXAI_PROJECT_ID).evaluateAttributeExpressions(flowfile).getValue()

    my_credentials = {
        "url" : "https://us-south.ml.cloud.ibm.com",
        "apikey" : watsonx_api_key
    }

    model_id      = ModelTypes.LLAMA_2_70B_CHAT
    gen_parms     = None
    project_id   = project_id
    space_id      = None
    verify        = False

    model = Model( model_id, my_credentials, gen_parms, project_id, space_id, verify )
    gen_parms_override = None
    generated_response = model.generate( prompt_text, gen_parms_override )

    attributes = {"mime.type": "application/json"}
    output_contents = json.dumps(generated_response)

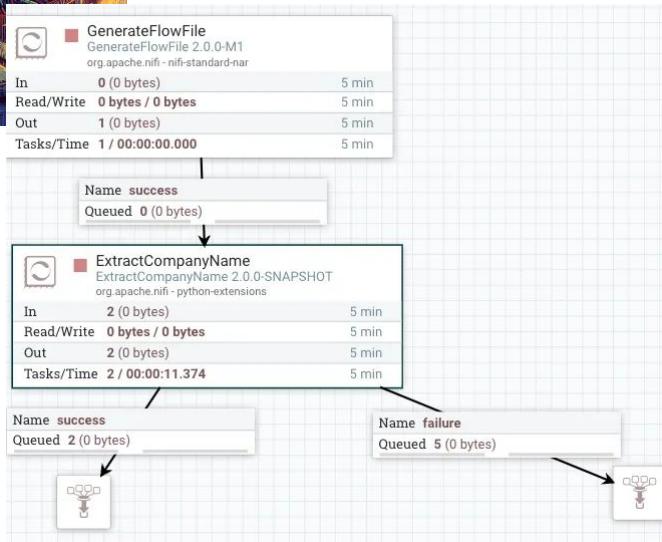
    self.logger.debug(f"Prompt: {prompt_text}")

    return FlowFileTransformResult(relationship = "success", contents=output_contents, attributes=attributes)
```



Extract Company Names

- Python 3.10+
- HuggingFace, NLP, SpaCY, PyTorch



Attribute Values

companylist

["Amazon", "Microsoft", "Cloudera", "DataSQLR", "Google", "IBM"]

filename

36fb4ae6-701a-4e1d-b890-c93b44f2200b

parsedcompany

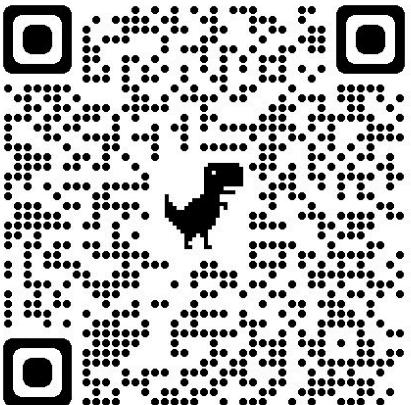
Amazon

path

./

uuid

6366a2c9-3dd4-4e8f-8825-83189d403b92



Get Compound GTFS Data

- Python 3.10+
- GTFS to JSON

trip_update

trip_update

vehicle

vehicle

alert

Reference parameter...

Processor Details | GetGTFSCompoundFeed 2.0.0-M2

Running

SETTINGS SCHEDULING PROPERTIES RELATIONSHIPS COMM

Required field

Property	Value
URL for GTFS Feed	Sensitive value set
API Key for header (MTA)	Sensitive value set
API Key for header name ex: (MTA)	x-api-key
Type for GTFS Feed	vehicle



Extract Text from Web VTT

- Python 3.10+
- Web VTT to Text
- Web Video Text Tracks Format Extractor

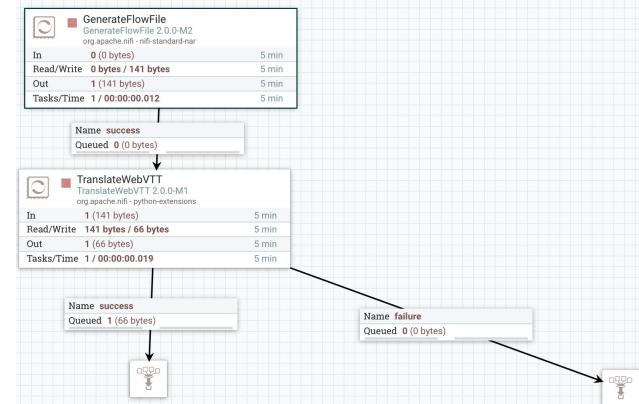
WEBVTT

1
00:00:06.066 --> 00:00:07.166

Now let's talk about

2
00:00:07.166 --> 00:00:12.033

data retrieval, views,
and materialized views.



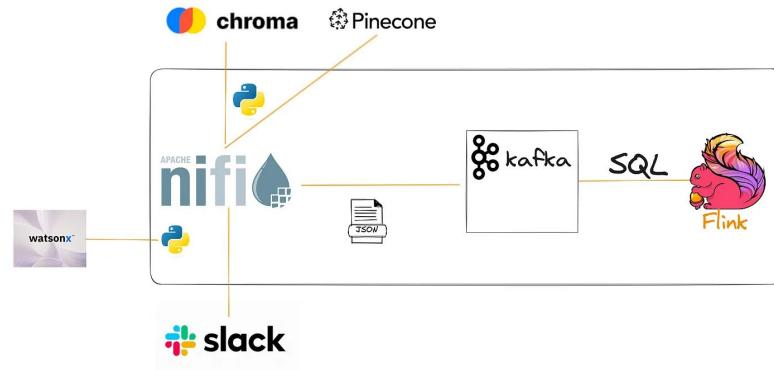
https://developer.mozilla.org/en-US/docs/Web/API/WebVTT_API

<https://github.com/tspannhw/FLaNK-python-processors/blob/main/TranslateWebVTT.py>



WatsonX SDK To Foundation

- Python 3.10+
- LLM
- WatsonX.AI Foundation Models
- Inference
- Secure
- Official SDK from IBM



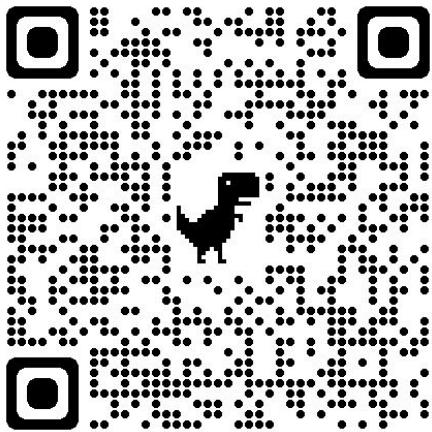
<https://github.com/tspannhw/FLaNK-python-watsonx-processor>

© 2024 Cloudera, Inc. All rights reserved.



System / Process Monitoring

- Python 3.10+
- psutil
- Swap memory, disk, networks



Attribute Values

cpu

18.1

diskusage

67583.5 MB

filename

f2975a45-28ab-4cea-bb04-ec25fed2efae

memory

67.2

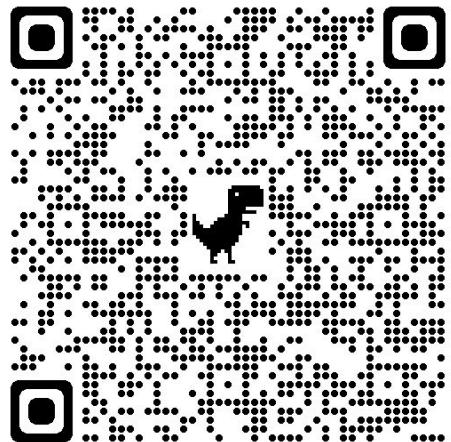
netaddr

```
{"lo0": [[2, "127.0.0.1", "255.0.0.0", null, null], [30, "::1", "ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff", null, null], [30, "fe80::1%lo0", "ffff:ffff:ffff:ffff:", null, null]], "en0": [[2, "192.168.1.153", "255.255.255.0", "192.168.1.255", null], [18, "bc:0d:074:65:4b:da", null, null, null], [30, "fe80::821:6f53:8208:ee72%en0", "ffff:ffff:ffff:ffff:", null, null]], "anpi2": [[18, "9a:b8:70:7d:1e:1d", null, null, null], [30, "fe80::98b8:70ff:fe7d:1e1d%anpi2", "ffff:ffff:ffff:ffff:", null, null]], "anpi1": [[18, "9a:b8:70:7d:1e:1c", null, null, null], [30, "fe80::98b8:70ff:fe7d:1e1c%anpi1", "ffff:ffff:ffff:ffff:", null, null]], "anpi0": [[18, "9a:b8:70:7d:1e:1b", null, null, null], [30, "fe80::98b8:70ff:fe7d:1e1b%anpi0", "ffff:ffff:ffff:", null, null]], "en4": [[18, "9a:b8:70:7d:1e:fb", null, null, null]], "en5": [[18, "9a:b8:70:7d:1e:fc", null, null, null]], "en7": [[18, "9a:b8:70:7d:1e:fd", null, null, null]], "en1": [[18, "36:6d:e8:f0:47:00", null,
```



Generate Synthetic Records w/ Faker

- Python 3.10+
- faker
- Choose as many as you want
- Attribute output



Configure Processor | GetFakeRecord 2.0.0-M1

○ Validating

SETTINGS SCHEDULING PROPERTIES RELATIONSHIPS COMMENTS

Required field

Property	Value
Include UUID	true
Include CREATED_DT	true
Include EMAIL	true
Include IP_V4	true
Include USER_NAME	true
Include CLUSTER_NAME	true
Include CITY	true
Include COUNTRY	true
Include POSTCODE	true
Include STREET_ADDRESS	true
Include LICENSE_PLATE	true
Include EAN13	true
Include ZIPPOSTAL_CODE	true

CANCEL APPLY

flowFile

DETAILS ATTRIBUTES

Attribute Values

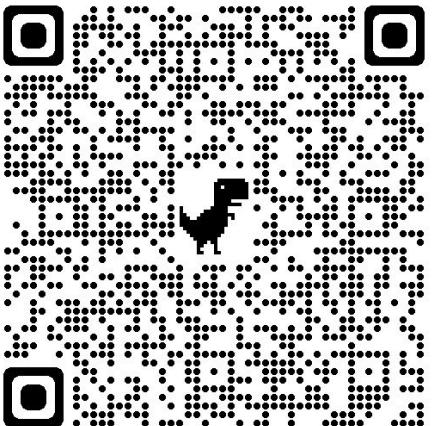
catchphrase	Ameliorated needs-based matrix
city	West Ashleyshire
clustername	benefit-rate-ask
comment	orchestrate proactive technologies
company	Cruz, Martinez and Edwards
country	Faroe Islands
createddt	2021-01-01

© 2024 Cloudera, Inc. All rights reserved.



Download a Wiki Page as HTML or WikiFormat (Text)

- Python 3.10+
- Wikipedia-api
- HTML or Text
- Choose your wiki page dynamically



Configure Processor | GetWikiData 2.0.0-M1

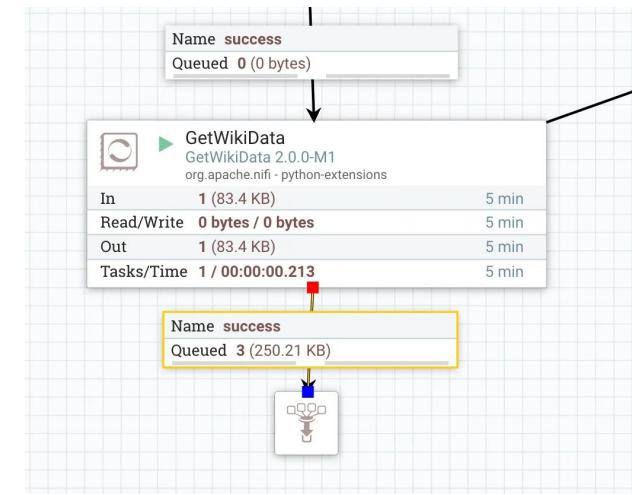
⚠ Invalid

SETTINGS SCHEDULING PROPERTIES RELATIONSHIPS COMMENTS

Required field

Property	Value
Plain Text Wiki format or HTML	<input checked="" type="radio"/> text
Wiki Page	<input checked="" type="radio"/> \${company0}

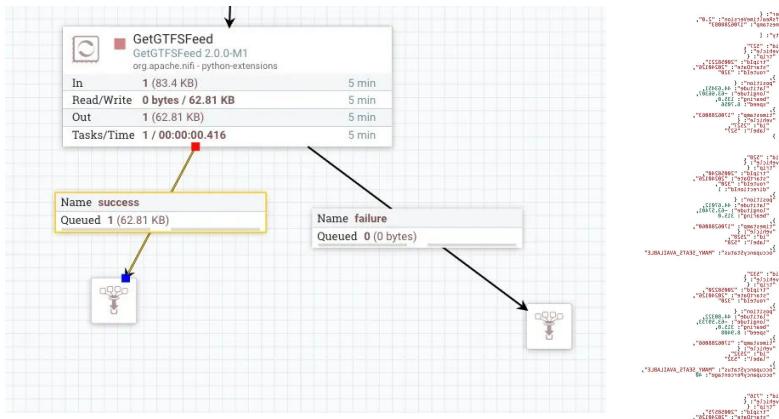
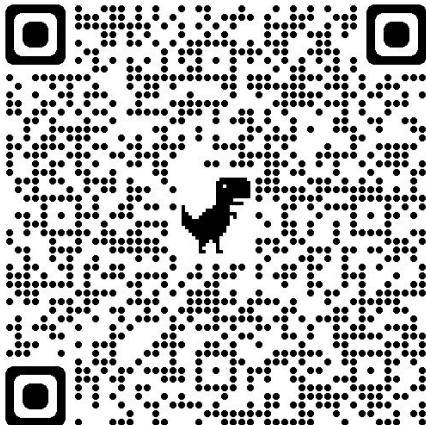
CANCEL APPLY





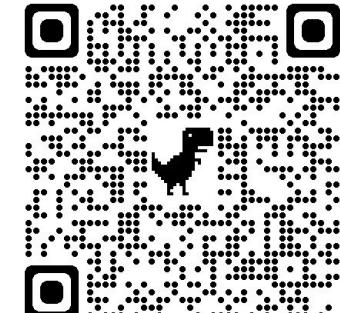
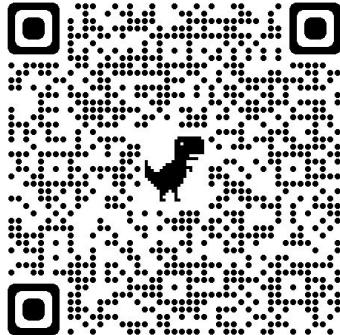
Get GTFS Data

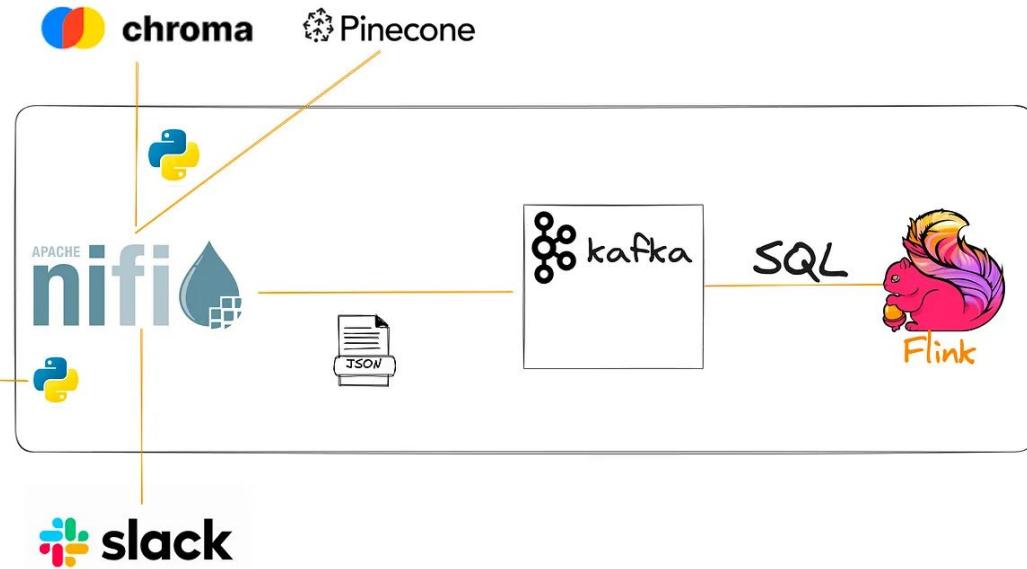
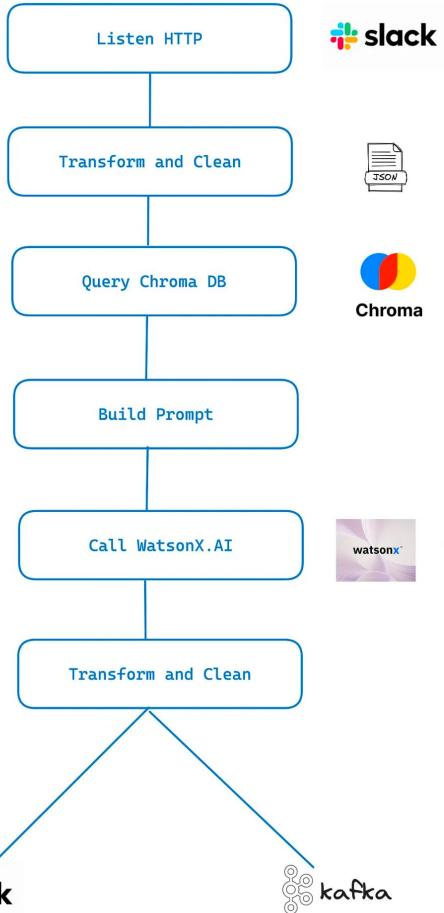
- Python 3.10+
 - GTFS from Transit URL
 - Alerts, Trip Updates or Vehicle Positions
 - Returns JSON
 - `google.transit` and `google.protobuf`



Other Python Processors

- Updated Pinecone (Vector DB Interface)
- ChunkDocument, ParseDocument
- ConvertCSVtoExcel
- DetectObjectInImage
- PromptChatGPT
- PutChroma, QueryChroma (Vector DB Interface)





SQL



DEMO





THON YU

