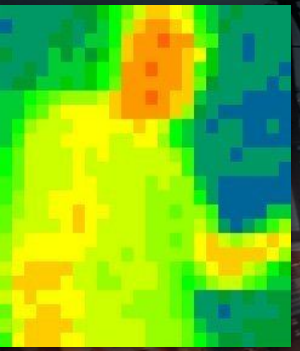
CLOUDERA

Data-in-Motion to Supercharge AI

Tim Spann
Principal Developer Advocate

23-August-2023

Tim Spann

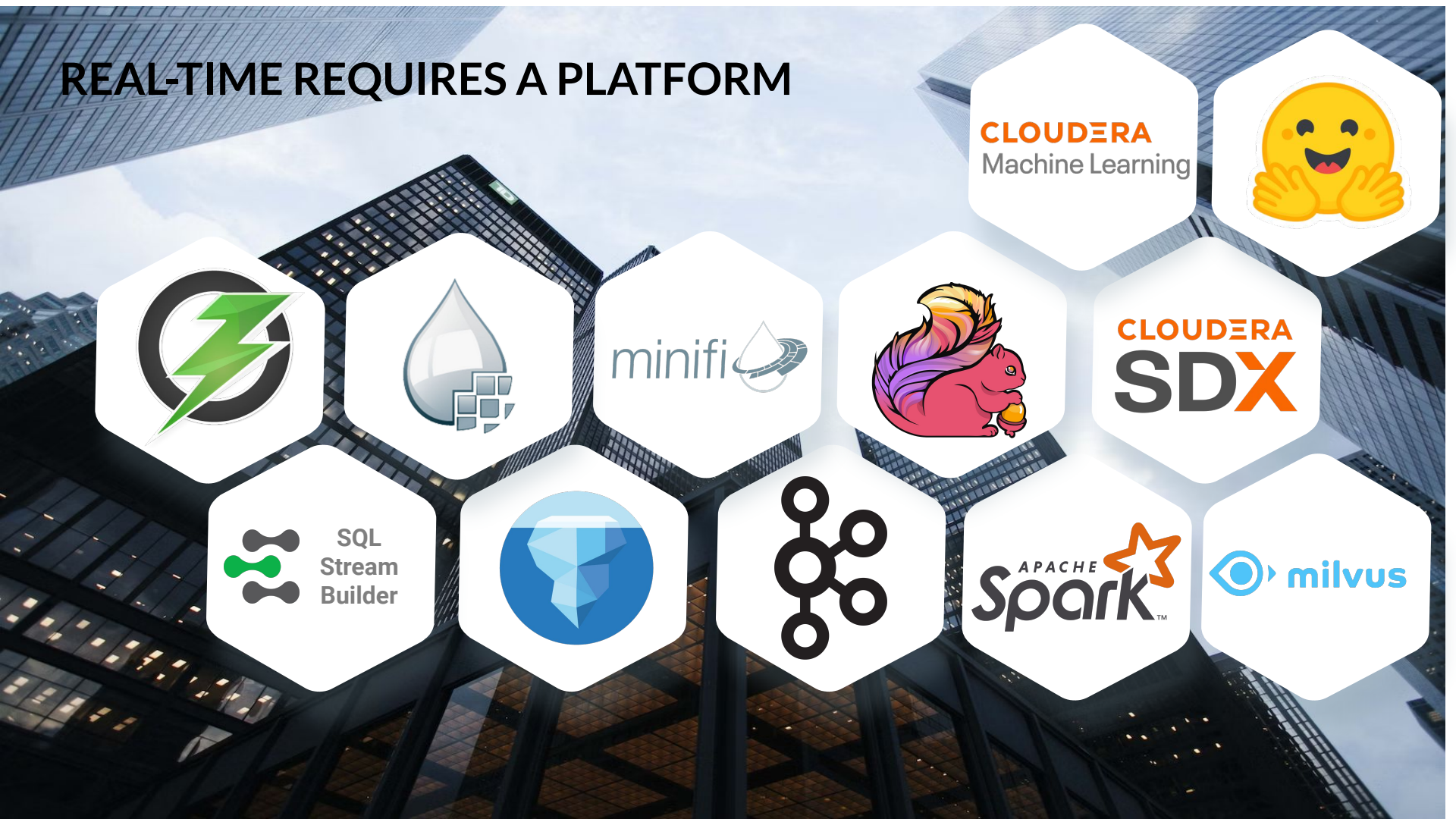


@PaasDev www.datainmotion.dev
github.com/tspannhw medium.com/@tspann
Principal Developer Advocate

Princeton Future of Data Meetup
ex-Pivotal, ex-Hortonworks, ex-StreamNative,
ex-PwC, ex-EY, ex-HPE.

Apache NiFi x Apache Kafka x Apache Flink x AI

REAL-TIME REQUIRES A PLATFORM



Cloudera + LLMs

LLM Serving
Serving Framework

LLM Fine Tuning Process
Training Framework

Vector DB

Data Preparation
Data Engineering

Knowledge Repository
Data Storage / Management

CML RAY NVIDIA LangChain Streamlit python

CDF nifi

Streaming Classification
Real-Time Model Deployment

milvus Pinecone Chroma

CDE Spark python DASK

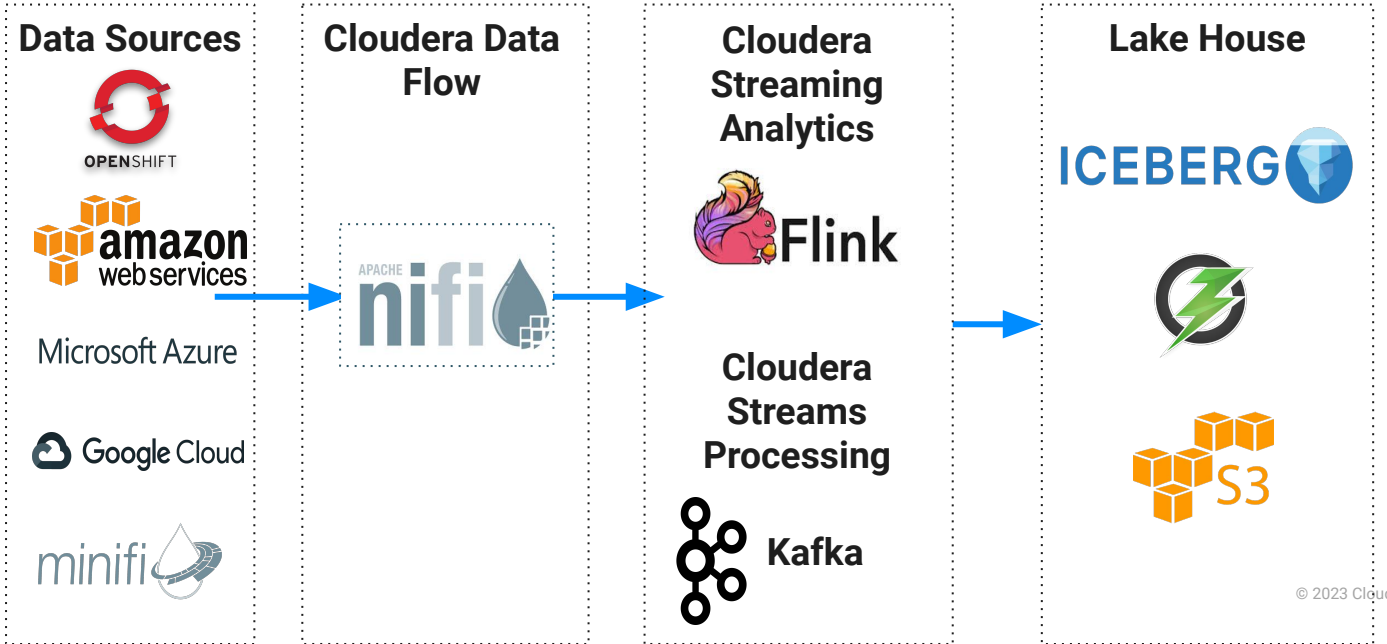
minifi

CDP CLOUDERA

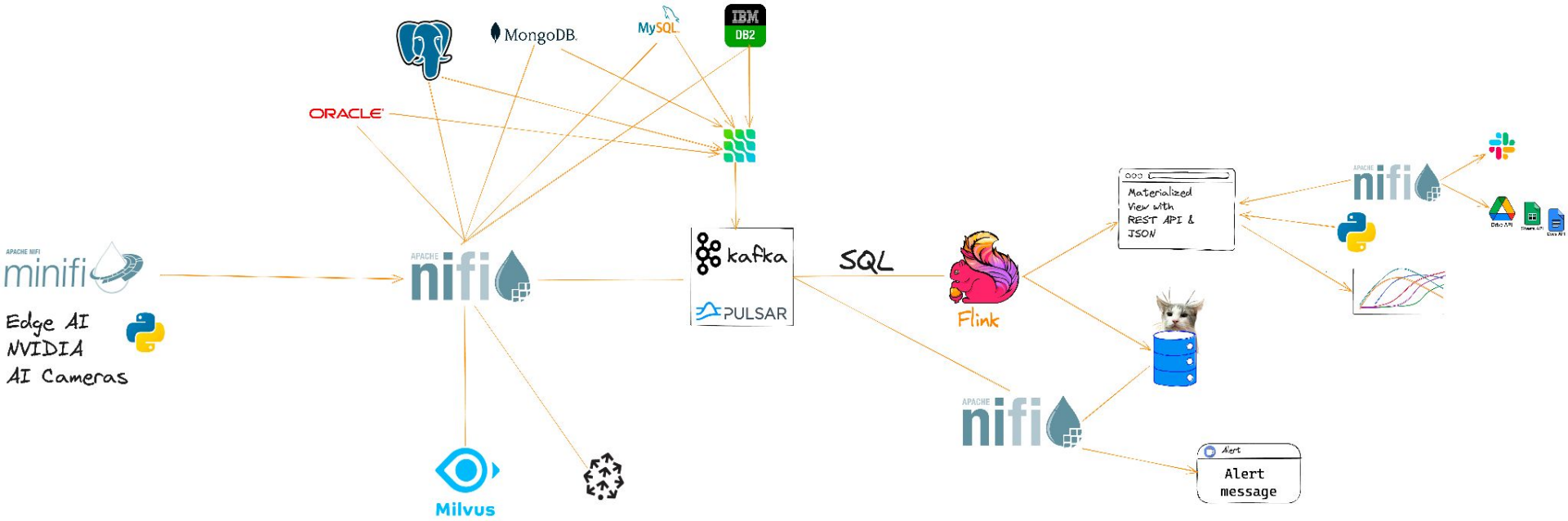
Key:
GPU Task
CPU Task

INGEST

Run collection and streaming on any cloud, server, container, bare metal, device or VM

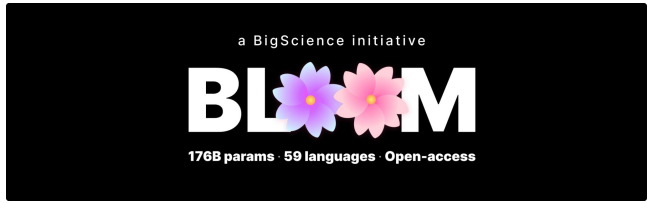


ENRICH


MongoDB.
MySQL
IBM
DB2
ORACLE
APACHE NIFI
minifi
Edge AI
NVIDIA
AI Cameras
APACHE
nifi
kafka
PULSAR
SQL
Flink
Materialized
View with
REST API &
JSON
APACHE
nifi
APACHE
nifi
Milvus
Alert
Alert
message

FUNNEL

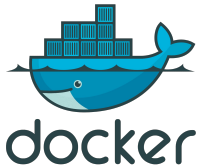
https://github.com/tspannhw/FLaNK-HuggingFace-DistilBert-SentimentAnalysis



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

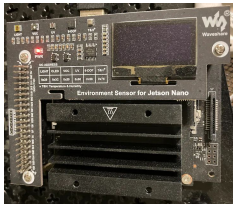
DISTRIBUTE

DEPLOY

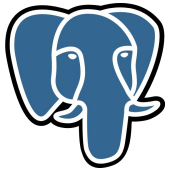


minifi

https://github.com/tspannhw/FLaNK-Edge-Models



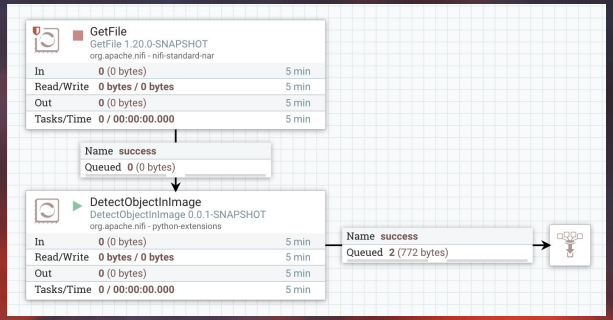
STORE



APACHE NIFI WITH PYTHON CUSTOM PROCESSORS

Python as a First Class Citizen

```
import cv2
import numpy as np
import json
from nifiapi.properties import PropertyDescriptor
from nifiapi.properties import ResourceDefinition
from nifiapi.flowfiletransform import FlowFileTransformResult

SCALE_FACTOR = 0.00392
NMS_THRESHOLD = 0.4     # non-maximum suppression threshold
CONFIDENCE_THRESHOLD = 0.5

class DetectObjectInImage:
    class Java:
        implements = ['org.apache.nifi.python.processor.FlowFileTransform']
    class ProcessorDetails:
        version = '0.0.1-SNAPSHOT'
        dependencies = ['numpy >= 1.23.5', 'opencv-python >= 4.6']

    def __init__(self, jvm=None, **kwargs):
        self.jvm = jvm

        # Build Property Descriptors
        self.model_file = PropertyDescriptor(
            name = 'Model File',
            description = 'The binary file containing the trained Deep Neural Network weights. Supports Caffe (*.caffemodel), TensorFlow (*.pb), Torch (*.t7, *.net), Darknet (*.weights), ' +
                          'DLDT (*.bin), and ONNX (*.onnx)',
            required = True,
            resource_definition = ResourceDefinition(allow_file = True)
        )
        self.config_file = PropertyDescriptor(
            name = 'Network Config File',
            description = 'The text file containing the Network configuration. Supports Caffe (*.prototxt), TensorFlow (*.pbtxt), Darknet (*.cfg), and DLDT (*.xml)',
            required = False,
            resource_definition = ResourceDefinition(allow_file = True)
        )
        self.class_name_file = PropertyDescriptor(
            name = 'Class Names File',
            description = 'A text file containing the names of the classes that may be detected by the model. Expected format is one class name per line, new-line terminated."',
            required = True,
            resource_definition = ResourceDefinition(allow_file = True)
        )
        self.descriptors = [self.model_file, self.config_file, self.class_name_file]

    def getPropertyDescriptors(self):
        return self.descriptors

    def onScheduled(self, context):
        # read class names from text file
        class_name_file = context.getProperty(self.class_name_file.name).getValue()
        if class_name_file is None:
```



https://github.com/apache/nifi/blob/614947e4ac6798ad80817e82514c39349d5faacb/nifi-docs/src/main/asciidoc/python-developer-guide.adoc

Future of Data - Princeton + 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 ...

FUTURE OF DATA

AN OPEN SOURCE COMMUNITY

ICEBERG

@PaasDev

FLaNK Stack Weekly

https://bit.ly/32dAJft

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

CSP Community Edition

- Kafka, KConnect, SMM, SR, Flink, and SSB in Docker
- Runs in Docker
- Try new features quickly
- Develop applications locally

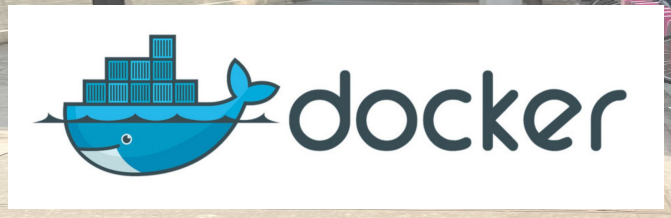
- Docker compose file of CSP to run from command line w/o any dependencies, including Flink, SQL Stream Builder, Kafka, Kafka Connect, Streams Messaging Manager and Schema Registry
  - `$> docker compose up`
- Licensed under the Cloudera Community License
- Unsupported
- Community Group Hub for CSP
- Find it on docs.cloudera.com under Applications

CSP Community Edition

A readily available, dockerized deployment of Apache Kafka and Apache Flink that allows you to test the features and capabilities of Cloudera Stream Processing.

Learn More

Open Source Edition



- Apache NiFi in Docker
- Runs in Docker
- Try new features quickly
- Develop applications locally

- Docker NiFi
  - `docker run --name nifi -p 8443:8443 -d -e SINGLE_USER_CREDENTIALS_USERNAME=admin -e SINGLE_USER_CREDENTIALS_PASSWORD=ctsBtRBKHRAx69EqUghvvgEvjnaLjFEB apache/nifi:latest`

- Licensed under the ASF License
- Unsupported

https://hub.docker.com/r/apache/nifi

Resources

threads.net/@tspannhw