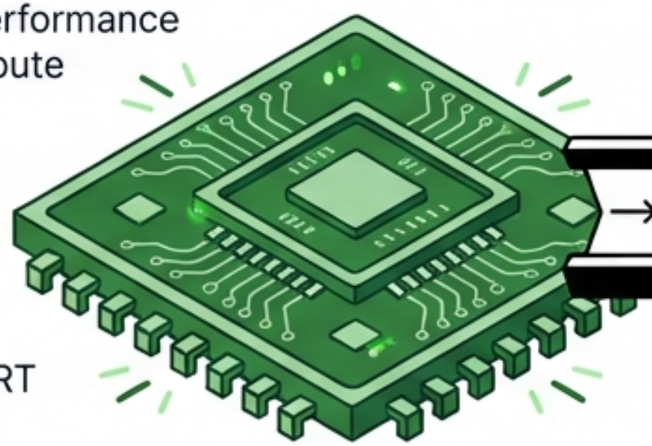
NVIDIA Orin Edge AI Device Streaming

High-Speed Ingestion & AI Enrichment with Snowpipe Streaming v2

NVIDIA Jetson AGX Orin

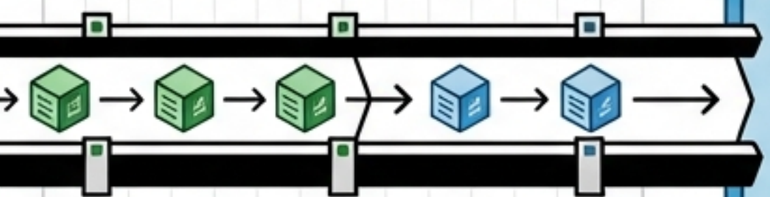
High-Performance
AI Compute



Tensor RT
Engine

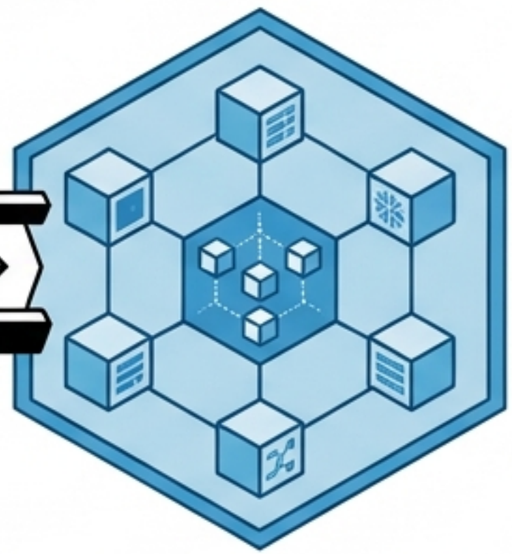
Edge AI Inference

Snowpipe Streaming v2



Streaming Ingestion Pipeline

Snowflake Data Cloud



Real-Time
Ingestion

Data
Warehousing

AI Enrichment

A Project by tspannhw

From Jetson AGX Orin to Snowflake in Real-Time

Bridging the Edge-to-Cloud Gap

This project provides a high-speed REST API client specifically designed for the NVIDIA Jetson AGX Orin. It solves the challenge of moving heavy edge data and metrics to the cloud while performing local inference.

Streaming Transport
Helvetica Now Display Bold
Direct ingestion to Snowflake via Snowpipe Streaming v2 REST API.

Edge Intelligence
Local AI enrichment using Ollama (e.g., Llama 3.2 Vision) before data leaves the device.

System Telemetry
Native collection of Jetson hardware metrics.

Automated Alerting
Integration with Slack for real-time visual notifications.

Python

NVIDIA Jetson

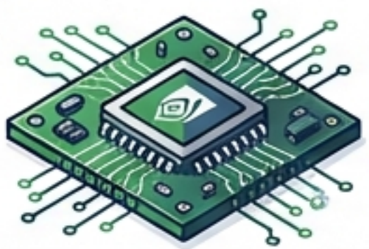
Ollama AI

Snowflake

The Data Journey Architecture

End-to-End System Flow

The Edge



NVIDIA Jetson AGX Orin

- Camera Video
- System Metrics

Raw Data

Processing

Python Client + Ollama

- Batch Enrichment
- Caption Generation

Enriched JSON

Transport

Snowpipe Streaming v2

- REST API
- Scoped Token Flow

HTTP/2

The Cloud

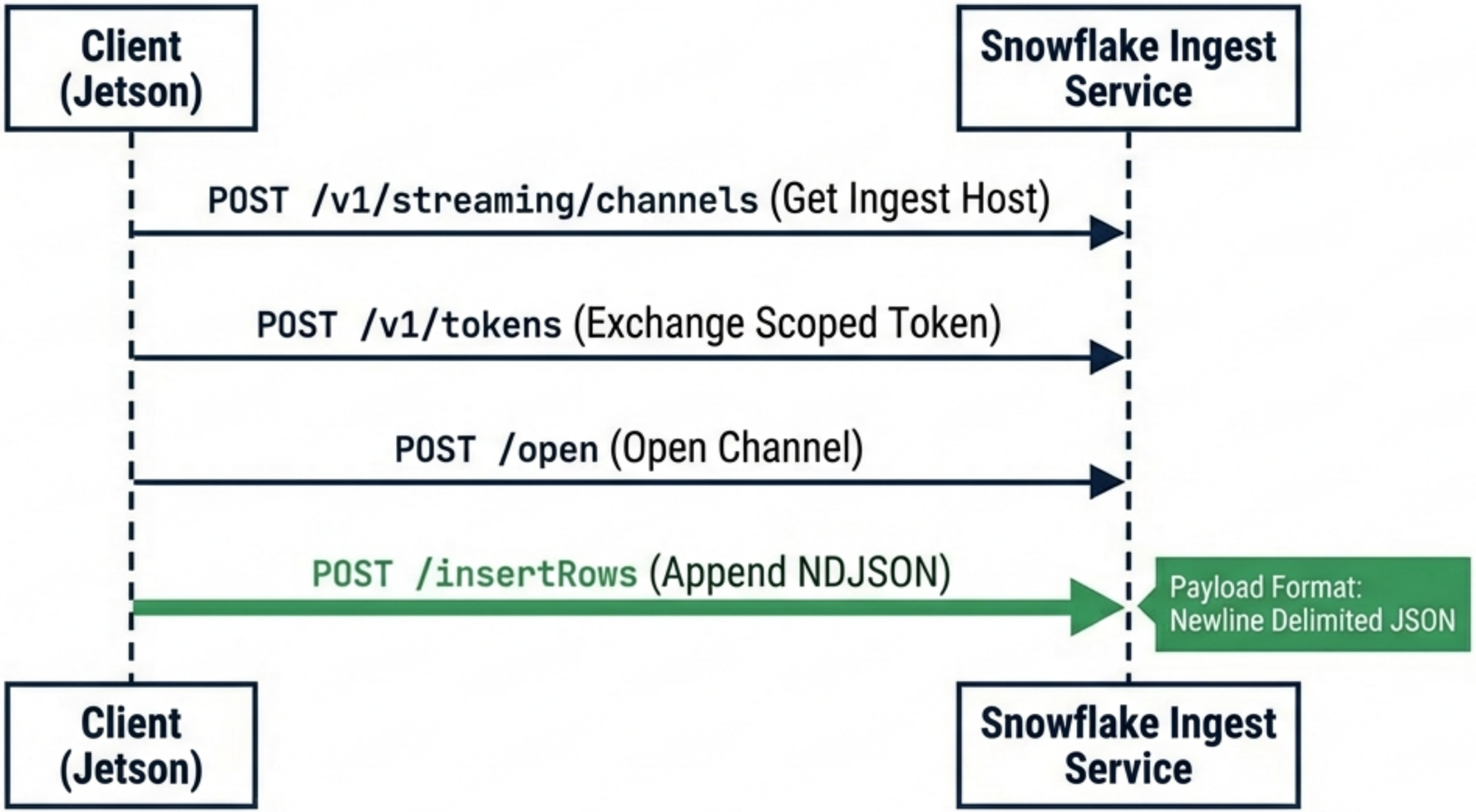
Snowflake
Analytics Tables

Slack
Visual Alerts

Data is enriched at the source, transported via low-latency HTTP, and lands instantly in the Data Cloud.

The Engine: Snowpipe Streaming v2

Leveraging the High-Performance REST API



Why It Matters:

- Optimized for edge devices where maintaining heavy JDBC drivers is impractical. Uses standard REST calls for low latency and high throughput.

Capturing the Hardware Pulse

Module: jetson_metrics.py

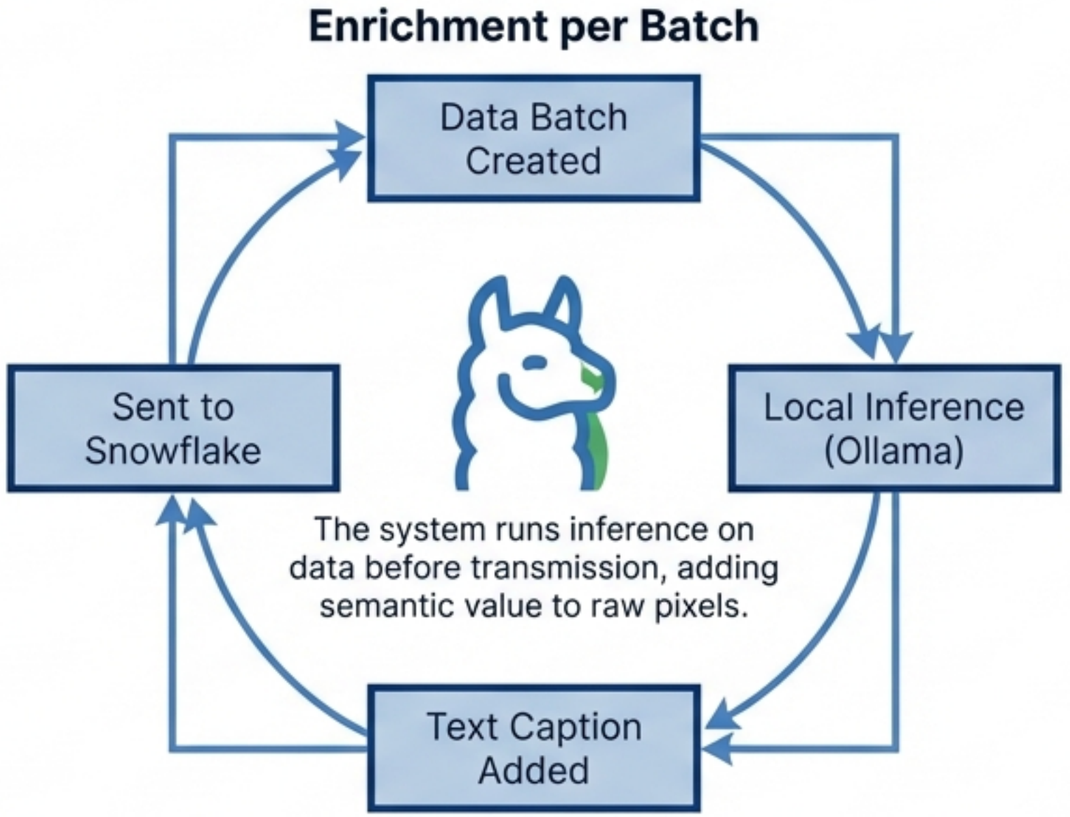
The solution includes a dedicated metrics collector specifically for the Jetson architecture. It allows engineers to correlate AI inference performance with hardware stress.

- CPU Usage & Frequency
- GPU Load Percentage
- Memory Consumption (RAM/Swap)
- Thermal Statistics (Zone Temps)


```
{
  "timestamp": "2023-10-27T10:00:01Z",
  "device_id": "ORIN_AGX_01",
  "metrics": {
    "gpu_load": 45.2,
    "cpu_usage": 12.4,
    "ram_used_gb": 8.1,
    "temp_gpu": 62.1,
    "temp_aux": 58.4
  }
}
```

Edge Intelligence with Ollama

Module: ollama_client.py



Required Setup Command:

```
ollama pull llama3.2-vision
```

Example Model: Llama 3.2 Vision for image analysis.

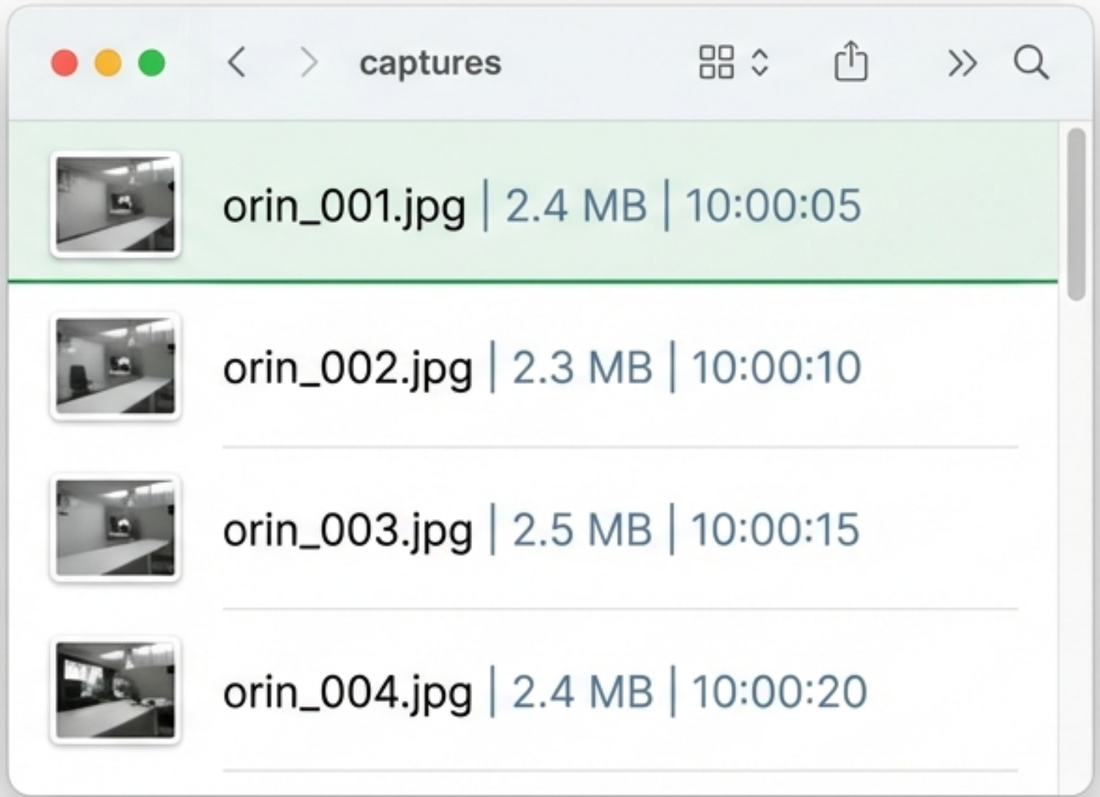
Video Capture & Vision Processing

Module: video_capture.py

Captures frames directly from the Jetson's connected camera device. Operates on a "frame per batch" logic to synchronize with data streaming intervals.

Configuration via **snowflake_config.json**:

```
device_index: 0 (Camera ID)
output_dir: ./captures
filename_prefix: orin
```



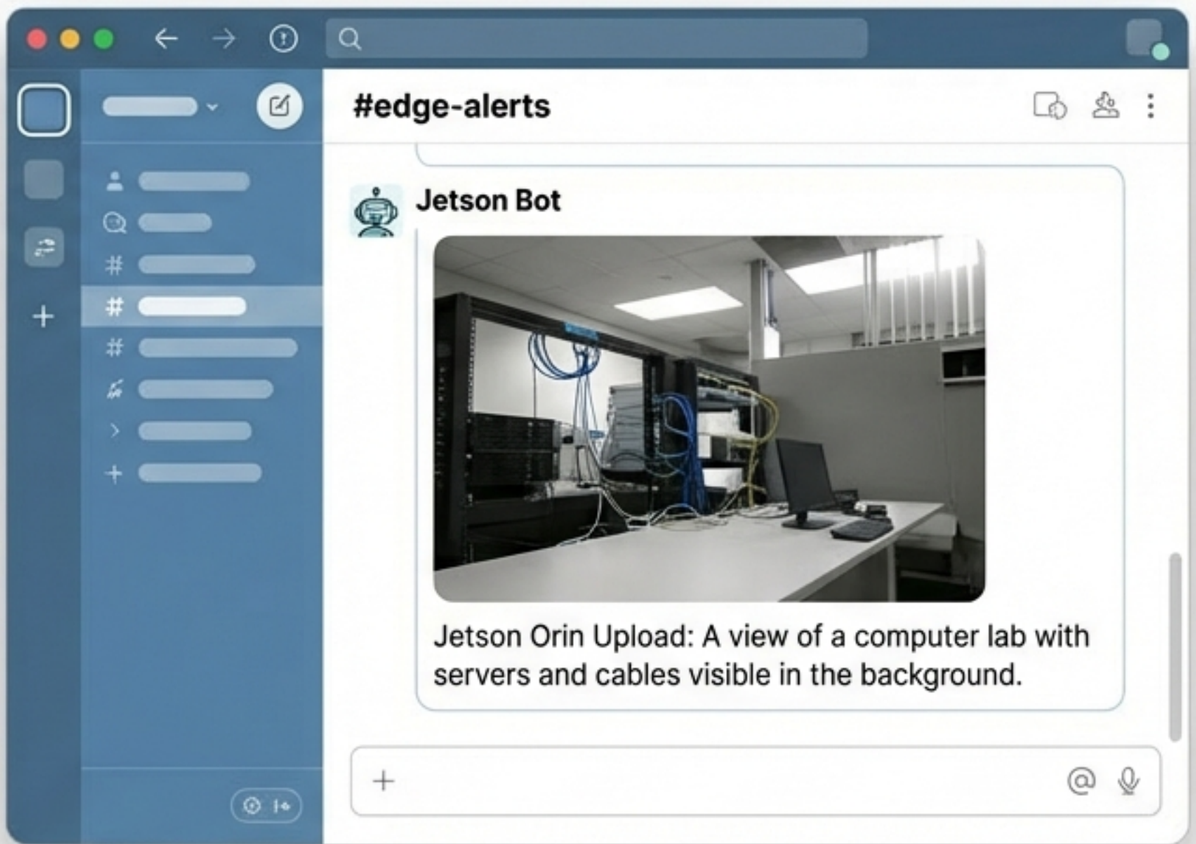
Real-Time Visual Alerts via Slack

Module: slack_client.py

Human-in-the-Loop Workflow:

1. **Capture**: Image taken by Jetson.
2. **Analyze**: Ollama generates a one-sentence summary.
3. **Notify**: Image + Summary uploaded to Slack.

Provides instant visual confirmation of edge events without querying the database.



Security & Authentication Models

Key-Pair JWT (Production)

- Uses `snowflake_jwt_auth.py`
- Ideal for secure, automated server-to-server communication.
- Generates a JSON Web Token signed with a private key.
- Best practice for enterprise deployment.

Programmatic Access Token (PAT)

- Simplified setup configuration.
- Matches Raspberry Pi / Weather streaming patterns.
- Config format: `account, url, pat`.
- Both methods support the required Scoped Token flow.

Prerequisites & Environment Setup

1. Python Environment

```
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

2. Snowflake Objects

Execute `setup_snowflake.sql` in Snowsight to create the database and table.

Note: Default pipe name: `<TABLE>-STREAMING`

3. AI Setup (Optional)

```
ollama serve
ollama pull llama3.2-vision
```

Configuration Guide

File: snowflake_config.json

Format must be ORG-ACCOUNT (Uppercased)

Matches table name + STREAMING

Feature Toggle

```
{
  "account": "ORG-ACCOUNT",
  "user": "edge_user",
  "database": "EDGE_DB",
  "schema": "PUBLIC",
  "table": "SENSOR_DATA",
  "pipe": "SENSOR_DATA_STREAMING",
  "video_capture": {
    "enabled": true
  },
  "slack": {
    "enabled": true
  }
}
```

Running the Streamer

```
user@jetson-agx:~/nvidiastreaming$ python main.py --config
snowflake_config.json --batch-size 25 --interval 5.0
--ollama-model llama3.2-vision

[INFO] Loading configuration...
[INFO] Connected to Snowflake Ingest Service.
[INFO] Ollama model loaded: llama3.2-vision
[INFO] Batch 1 sent: 25 rows successfully appended.
[INFO] Video frame captured: orin_001.jpg
```

Flag Reference

`--debug` : Verbose logging

`--batch-size` : Rows per request

`--interval` : Seconds between batches

Optimization & Best Practices

Batch Sizing

Keep batches below the 16 MB payload limit. NDJSON rows specifically should stay under 4 MB for optimal throughput.

Identifier Formatting

Always use `ORG-ACCOUNT` format in uppercase for the account identifier. Replace underscores with dashes in the ingest host URL.

API Awareness

The system manages channel status automatically: Open -> Append -> Check Status. No manual channel management required.

Resource Management

Ensure the `--interval` setting allows enough time for Ollama inference to complete before the next batch triggers.

Project Structure Reference

```
nvidiastreaming/
├── main.py  Entry point & CLI
├── snowpipe_streaming_client.py  API Wrapper Logic
├── Modules
│   ├── jetson_metrics.py  Hardware Stats
│   ├── ollama_client.py  AI Enrichment
│   ├── video_capture.py  Frame Capture
│   └── slack_client.py  Notifications
└── Auth
    └── snowflake_jwt_auth.py  Token Generation
```

Deployment Ready

A complete, modular blueprint for streaming high-value edge data to Snowflake with local AI intelligence.

github.com/tspannhw/nvidiastreaming

Documentation:

Snowpipe Streaming High-Performance REST API

Snowflake REST API Authentication

Clone. Configure. Stream.