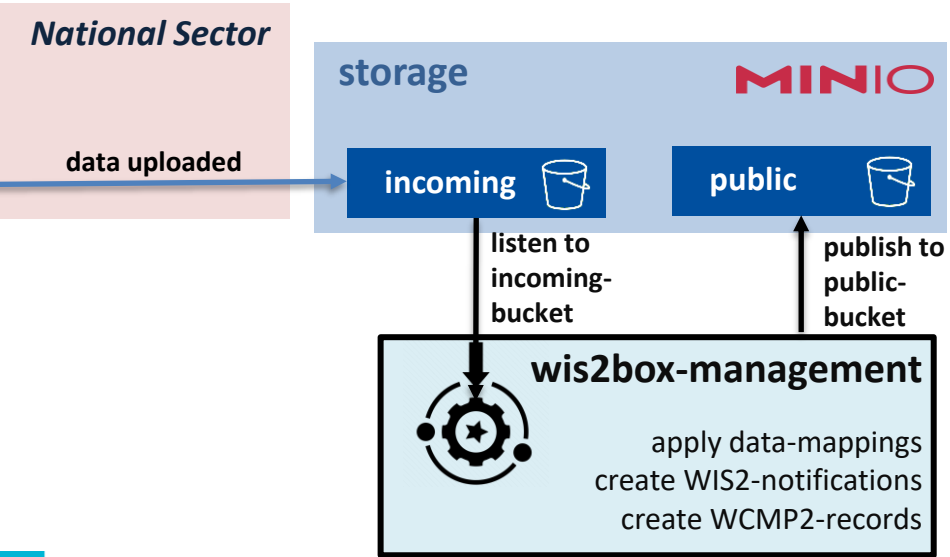


Ingesting and publishing data using wis2box

wis2box data mapping



Dataset driven workflow triggered when wis2box-management receives a notification from the storage service

wis2box-management **match file with a dataset** based on:

- **metadata-identifier** for the dataset contained in file-path, or
- **topic** for the dataset contained in file-path



Hourly synoptic observations from fixed-land stations (SYNOP)
(my-center-id)

Topic: origin/a/wis2/my-center-id/data/core/weather/surface-based-observations/synop
Metadata Identifier: urn:wmo:md:my-center-id:synop-test-dataset

EXPLORE ⓘ

OBSERVATIONS ⓘ

METADATA ⓘ

MESSAGES ⓘ

If **no match** with dataset is found, the data will **not** be published

If a matching dataset is found, corresponding data-mappings are applied:

- Data-mappings **match on file-extension and file-pattern**
- If **no match** on file-type and file-pattern found, the data is **not** published
- For each matching file, the corresponding **plugin** is applied to **transform & publish the data**

wis2box data ingestion

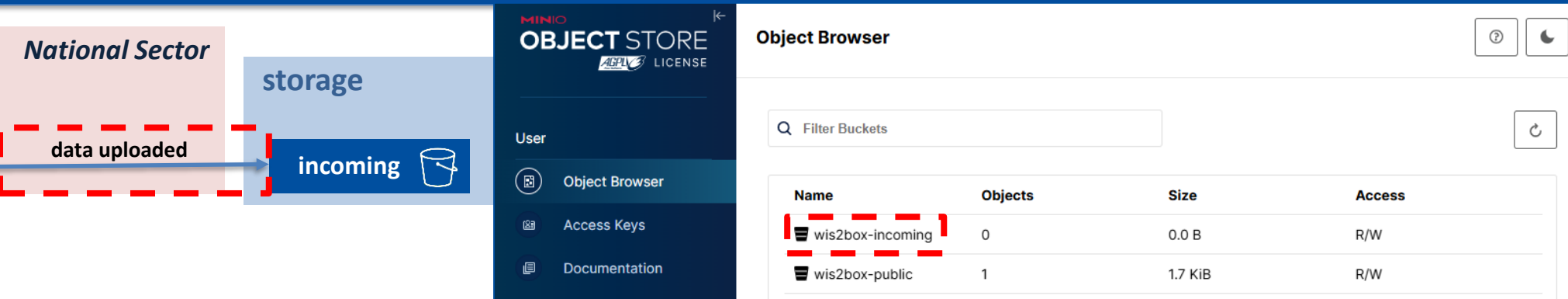


Data can be uploaded to wis2box-incoming using different methods
Optimal solution will depend on solutions used in national sector

Options for ingesting data in wis2box:

- **Manual upload using MinIO User Interface**: useful for testing data-workflow
- **FM-12 form in wis2box-webapp**: allows manual observers to directly publish their observations on WIS2
- **Custom code using MinIO client library**: enables automation for custom development
- **Using SFTP protocol**: useful for automation with existing/commercial data management solutions

MinIO User Interface



The diagram on the left shows a flow from 'National Sector' (red dashed box) to 'data uploaded' (red dashed box) and then to 'incoming' (blue box with a bucket icon) under the 'storage' section. The screenshot on the right shows the MinIO Object Browser interface. The left sidebar has 'Object Browser' selected. The main area shows a table of buckets:

Name	Objects	Size	Access
wis2box-incoming	0	0.0 B	R/W
wis2box-public	1	1.7 KiB	R/W

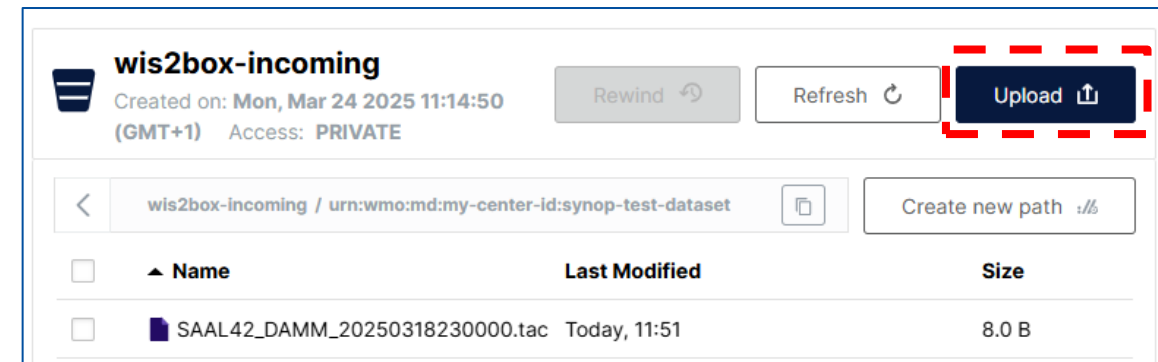
The 'wis2box-incoming' bucket name is highlighted with a red dashed box.

Data can be uploaded manually using the MinIO web interface at <http://<your-host>:9001>

- Useful for testing your data mappings before setting up an automated workflow
- Login with WIS2BOX_STORAGE_USERNAME / WIS2BOX_STORAGE_PASSWORD defined in wis2box.env
- Create new path in wis2box-incoming bucket matching metadata-id and upload your file



The dialog shows the current path as 'wis2box-incoming'. The 'New Folder Path*' field contains 'urn:wmo:md:my-center-id:synop-test-dataset'. There are 'Clear' and 'Create' buttons.



The bucket view for 'wis2box-incoming' shows the path 'wis2box-incoming / urn:wmo:md:my-center-id:synop-test-dataset'. The 'Upload' button is highlighted with a red dashed box. Below the path, a table shows the uploaded file:

Name	Last Modified	Size
SAAL42_DAMM_20250318230000.tac	Today, 11:51	8.0 B

For security do not enable external access on port 9001 nor share wis2box-storage credentials

SFTP

National Sector

data uploaded



Data can be uploaded using SFTP

- Useful for automating data-flow from vendors supporting SFTP

File protocol:
SFTP

Host name: mlimper.wis2.training Port number: 8022

User name: wis2box Password:

MinIO access over SFTP is enabled on port 8022

Login with WIS2BOX_STORAGE_USERNAME / WIS2BOX_STORAGE_PASSWORD

For security: restrict network access on port 8022 to fixed IP from data-source

/				
Name	Size	Changed	Rig...	Ow...
..				
wis2box-public		3/24/2025 11:14:50 AM	---	0
wis2box-incoming		3/24/2025 11:14:50 AM	---	0

/wis2box-incoming/urn:wmo:md:my-center-id:synop-test-dataset/				
Name	Size	Changed	Rig...	Ow...
..				
SAAL42_DAMM_20250318230000.tac	1 KB	3/24/2025 11:51:36 AM	rw...	0

Use directory matching metadata-id or topic to trigger data ingestion workflow

Data ingestion script



Data can be uploaded directly into a minio-client library

- Useful for custom automation, e.g. crontab using a Python script
- <https://min.io/docs/minio/linux/developers/minio-drivers.html>

Python example to upload data using the MinIO client module:

```
import glob
import sys

from minio import Minio

filepath = '/home/wis2box-user/local-data/mydata.bin'
# path should match the metadata-id or topic for corresponding dataset
minio_path = 'urn:wmo:md:my-center-id:synop-test-dataset'

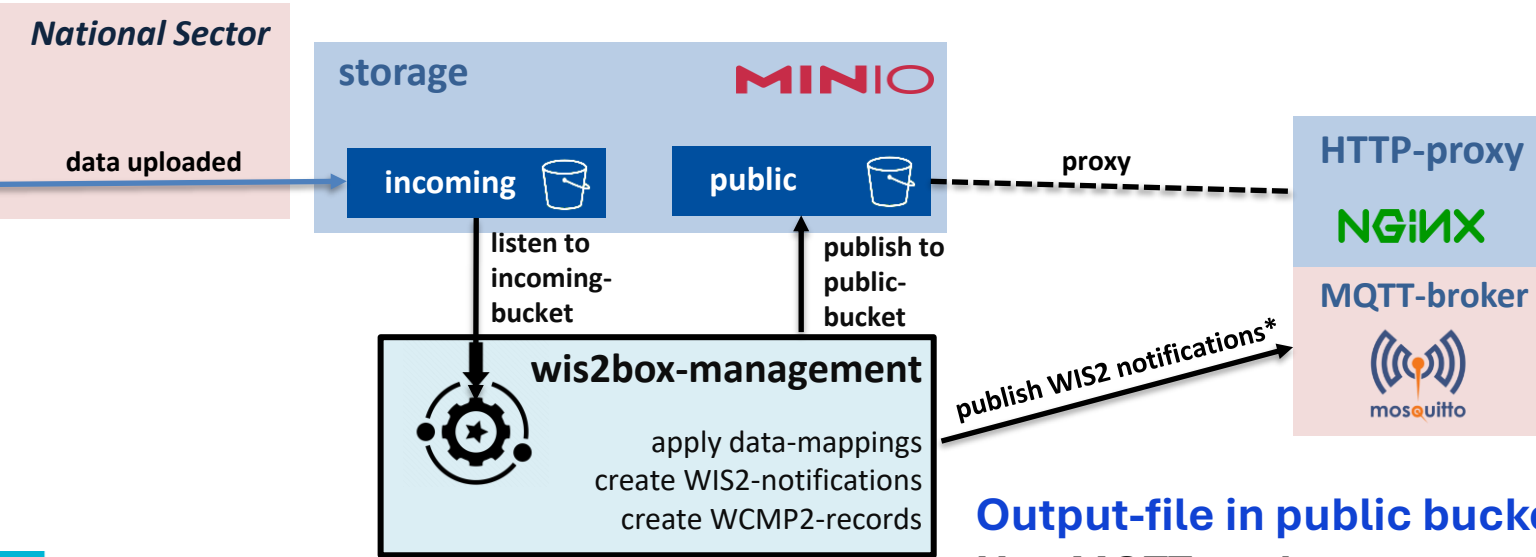
endpoint = 'http://localhost:9000'
WIS2BOX_STORAGE_USERNAME = 'wis2box'
WIS2BOX_STORAGE_PASSWORD = 'XXXXXXXXXX'

client = Minio(
    endpoint=endpoint,
    access_key=WIS2BOX_STORAGE_USERNAME,
    secret_key=WIS2BOX_STORAGE_PASSWORD,
    secure=is_secure=False)

filename = filepath.split('/')[-1]
client.fput_object('wis2box-incoming', minio_path+filename, filepath)
```

wis2box data publication

* Since the HTTP-proxy exposes wis2box-public bucket as '/data', WIS2 Notifications will use href="\$WIS2BOX_URL/data/..."



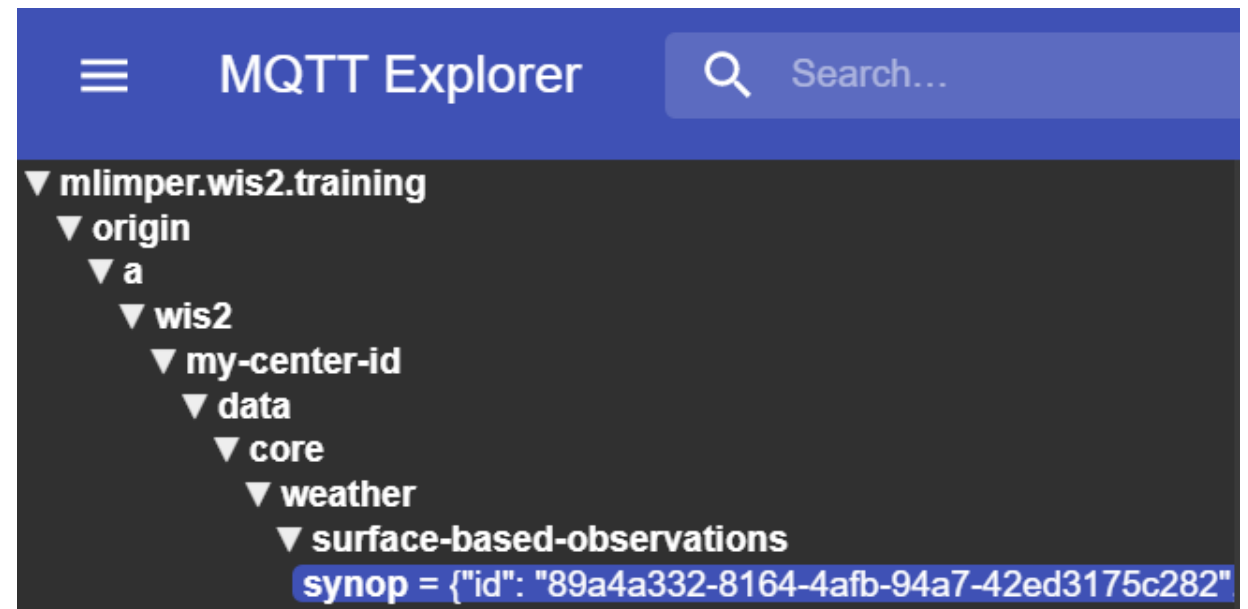
Output-file in public bucket results in WIS2-notification on MQTT
Use MQTT explorer to see the WIS2-notifications for your data

MQTT Connection mqtt://mlimper.wis2.training:1883/

Name ☐ Validate certificate ☐ Encryption (tls)

Protocol Host Port

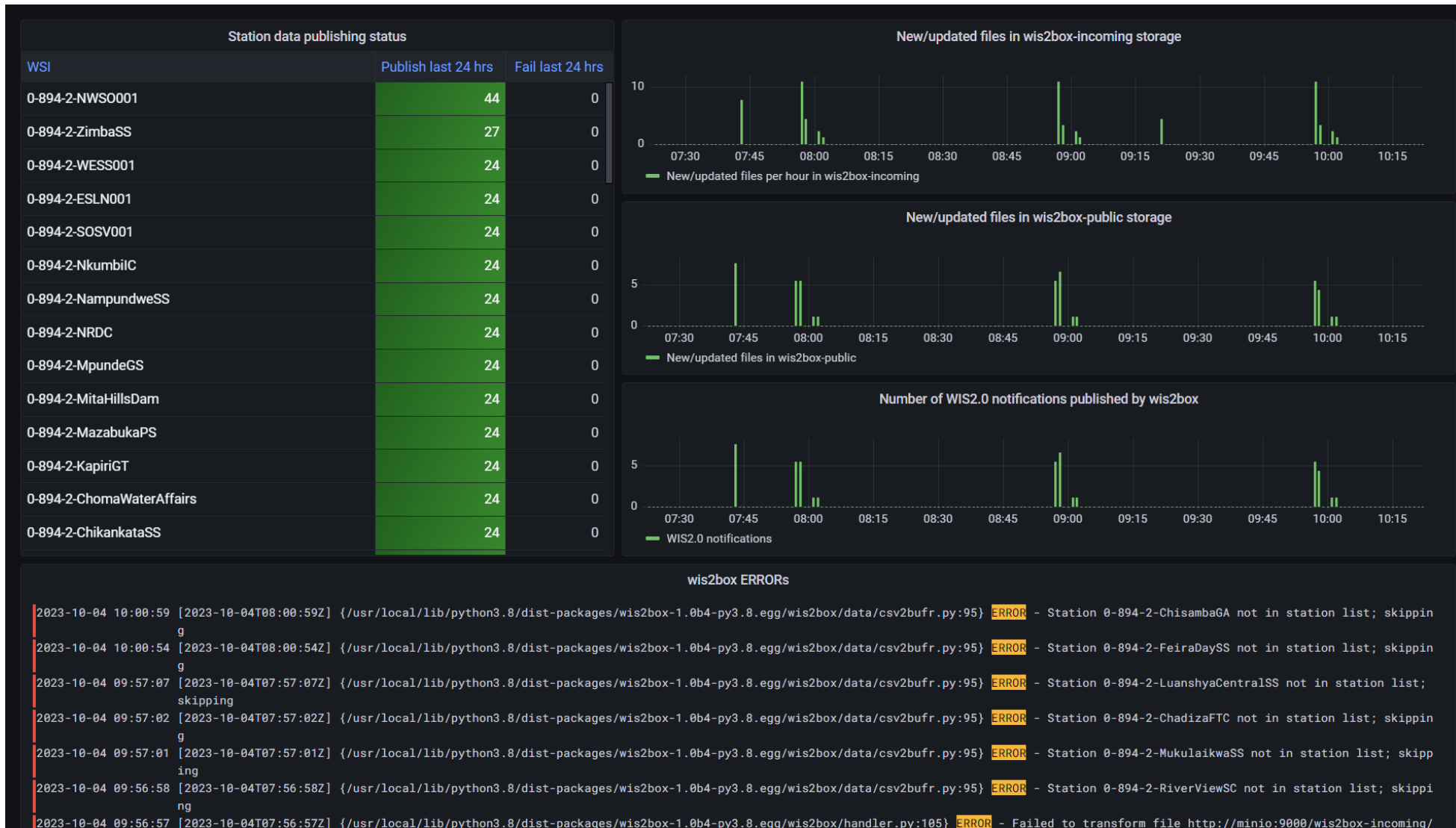
Username Password ☐



Monitoring

Grafana home dashboard available at <http://<wis2box-host-address>:3000>

Grafana data-sources: **Prometheus** (time-series metrics) and **Loki** (logs)



'explore' option In Grafana sidebar

Exploring logs stored in Loki

The screenshot displays the Grafana Explore interface for querying logs in Loki. The interface is divided into several sections:

- Top Bar:** Includes the 'Explore' button, a dropdown menu for the data source (currently 'wis2box-loki'), and buttons for 'Split', 'Add to dashboard', 'Last 1 hour', 'Run query', and 'Live'.
- Query Patterns:** A section with a dropdown for 'Query patterns', a toggle for 'Raw query', and a 'Give feedback' link.
- Labels:** A section for filtering logs by labels. It shows 'container_name' set to 'wis2box-management'. A yellow box highlights this section, and a yellow arrow points to it with the text 'wis2box-loki as a data source'.
- Line contains:** A section for filtering logs by text. It has a 'Text to find' input field and a '+ Operations' button. A yellow arrow points to this section with the text 'select container'.
- Raw query:** A section for entering a raw query. The query shown is `{container_name="wis2box-management"} |= ''`. A yellow arrow points to this section with the text 'run query'.
- Log volume:** A bar chart showing the volume of logs over time. The x-axis represents time from 13:00 to 13:55, and the y-axis represents the number of logs (0 to 500).
- Logs:** A section for displaying the raw log data. It includes a 'Time' toggle, a 'Unique labels' toggle, a 'Wrap lines' toggle, a 'Prettyfy JSON' toggle, a 'Dedup' toggle, and a 'Display results' button. The logs are sorted by 'Newest first'.

The logs displayed are as follows:

```
> 2023-10-04 13:57:09 [2023-10-04T11:57:09Z] {/usr/local/lib/python3.8/dist-packages/wis2box-1.0b4-py3.8.egg/wis2box/pubsub/subscribe.py:46} INFO - Data processed
> 2023-10-04 13:57:09 [2023-10-04T11:57:09Z] {/usr/local/lib/python3.8/dist-packages/wis2box-1.0b4-py3.8.egg/wis2box/data/base.py:195} INFO - Publishing output data
> 2023-10-04 13:57:09 [2023-10-04T11:57:09Z] {/usr/local/lib/python3.8/dist-packages/wis2box-1.0b4-py3.8.egg/wis2box/data/csv2bufr.py:95} ERROR - Station 0-894-2-LuanshyaCentralSS not in station list; skipping
> 2023-10-04 13:57:09 [2023-10-04T11:57:09Z] {/usr/local/lib/python3.8/dist-packages/wis2box-1.0b4-py3.8.egg/wis2box/metadata/station.py:257} INFO - Validating WIGOS Station Identifier: 0-894-2-LuanshyaCentralSS
> 2023-10-04 13:57:09 [2023-10-04T11:57:09Z] {/usr/local/lib/python3.8/dist-packages/csv2bufr/__init__.py:909} INFO - 2023-10-04T11:57:09.335450+00:00{'id': 'WIGOS_0-894-2-LuanshyaCentralSS_20231004T115400', 'geometry': {'type': 'Point', 'coordinates': [28.41, -13.14]}, 'properties': {'md5': '272c3fedeaee11d6463e34afb0abd5d1', 'wigos_station_identifier': '0-894-2-LuanshyaCentralSS', 'datetime': datetime.datetime(2023, 10, 4, 11, 54), 'origin_ating_centre': 0, 'data_category': 0}, 'result': {'code': 1, 'message': '', 'errors': []}}
> 2023-10-04 13:57:08 [2023-10-04T11:57:08Z] {/usr/local/lib/python3.8/dist-packages/wis2box-1.0b4-py3.8.egg/wis2box/pubsub/subscribe.py:43} INFO - Processing http://minio:9000/wis2box-incoming/zmb/zambia_met_service/data/core/weathe
r/surface-based-observations/synop/CR1000X_48229_20231004T135500Z.csv
> 2023-10-04 13:57:08 [2023-10-04T11:57:08Z] {/usr/local/lib/python3.8/dist-packages/wis2box-1.0b4-py3.8.egg/wis2box/pubsub/subscribe.py:49} INFO - Public filepath: wis2box-public/2023-10-04/wis/zmb/zambia_met_service/data/core/weathe
r/surface-based-observations/synop/WIGOS_0-894-2-NWS0001_20231004T115400-105.geojson
```

Data Ingestion and Monitoring

The wis2box data-driven workflow is triggered when data is uploaded in MinIO

- Manually using the MinIO-interface (for testing)
- Automated by using script or forwarding over SFTP

The status of the data ingestion can be monitored in the Grafana Dashboard

Once wis2box is configured and data is being ingested, **ensure the system is maintained and remains operational :**

- Check for any errors (view Grafana dashboard)
- Add/update/remove datasets
- Add/update/remove stations
- Review the data content being published

Thank you

شكراً لك

wmo.int



WORLD
METEOROLOGICAL
ORGANIZATION

