Data acquisition and NRT submission

sudheer.bhakare@tropos.de

IR0000032 – ITINERIS, Italian Integrated Environmental Research Infrastructures System
(D.D. n. 130/2022 - CUP B53C22002150006) Funded by EU - Next Generation EU PNRR-Mission 4 “Education and Research” - Component 2: “From research to business” - Investment 3.1: “Fund for the realisation of an integrated system of research and innovation infrastructures”

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani

What is Near Real-Time (NRT) Data?

NRT refers to **measurements collected from instruments that are processed and transmitted shortly after acquisition**, typically within a few minutes to a few hours or in a day

Why is it Important?

- Early detection of anomalies or pollution events.
- Supports real-time air quality forecasting and alerts.
- Enables fast diagnostics and quality checks.

Courtesy: Erich Furrer, MeteoSwiss

Components of NRT Workflow

Instrument

Measures atmospheric data, like CPC.

Data Logger

Records data

Database

PostgreSQL server for storing acquired data.

Visualization

Grafana, pgAdmin for data representation.

Data Logger Functionality

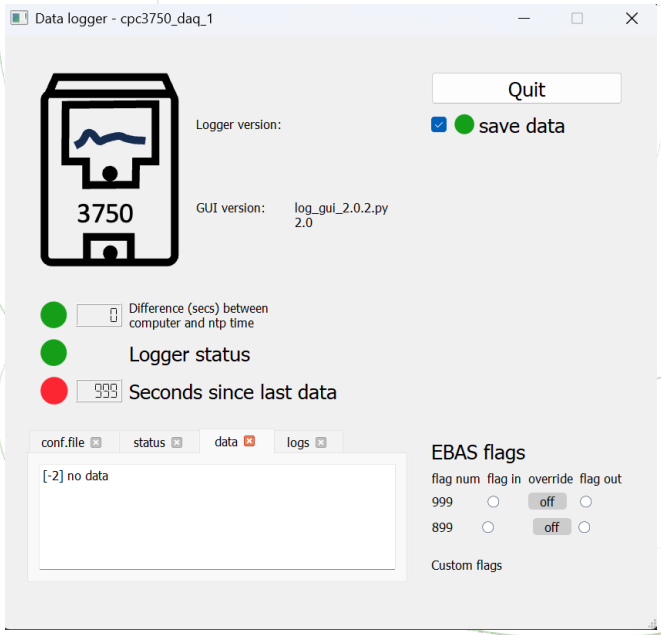
Role of the Data Logger:

- Acts as a bridge between the instrument and the database.
- Continuously reads raw data from the instrument (via serial/TCP connection).
- Makes it more structured by Adding timestamps and performing basic validation (e.g., checks for missing data).

Raw Output	Structured by Logger
1425 3.654 25 41 28	time: 14:25, flow_1: 3.654 L/min, temp_ctrl: 25 °C, temp_sat:41°C

Configuration and Customization:

- Each logger is governed by a configuration file (.cfg)

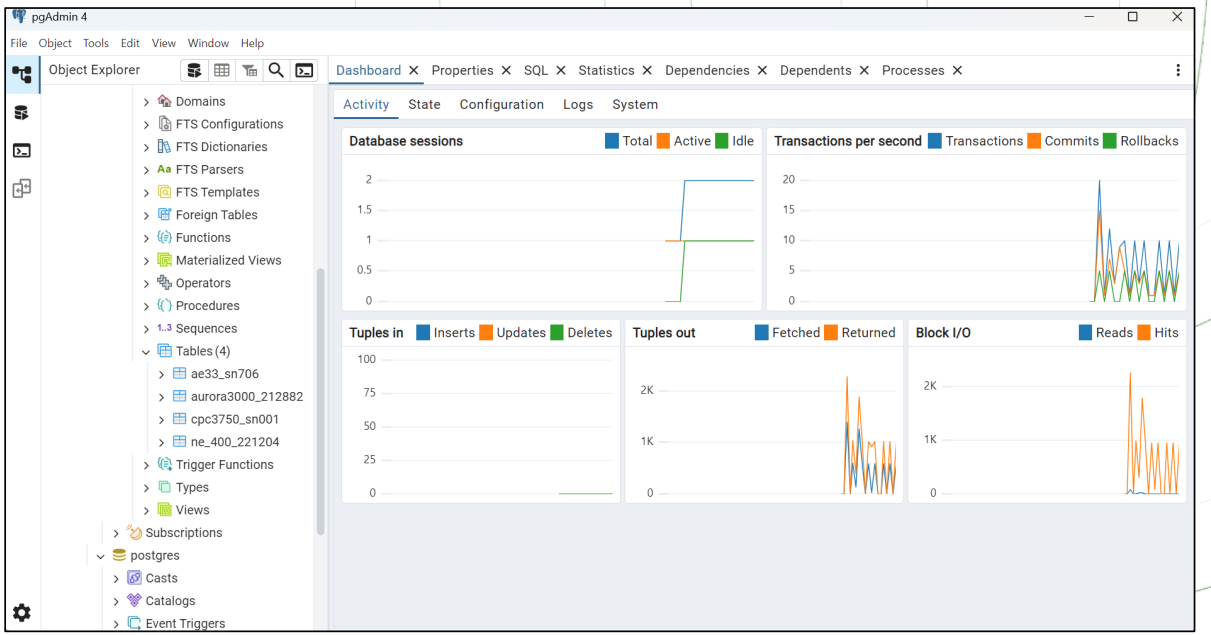


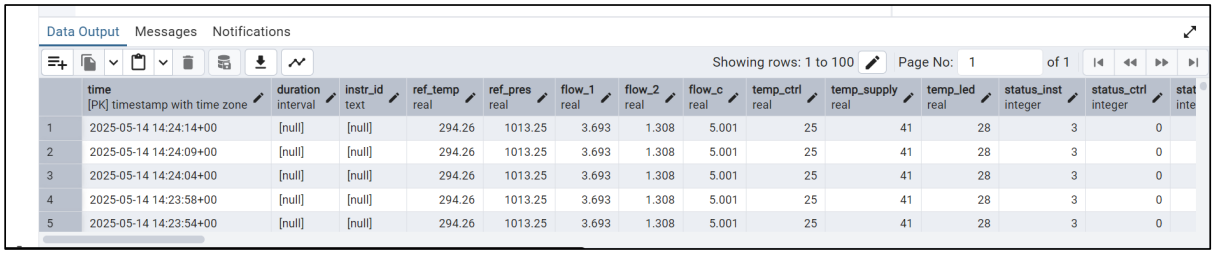
Example Configuration:

- logger_type: cpc_3750_tsi
- logger_name: cpc_123456
- time_resolution: 10
- datasource:
- enable_datasource: serial
- port: COM3

Database and Data Management

- Database is the **heart** of the NRT pipeline
  - Example: **PostgreSQL**
- Central repository for storing real-time data from **multiple instruments** (CPC, AE33, etc)
- Each logger reads and sends its stream of data into its **own table** in the database
- Ensures time-aligned, structured access to data streams
- Acts as a source for
  - NASA-Ames generation
  - Visualisation
  - Quality control





	time [PK] timestamp with time zone	duration interval	instr_id text	ref_temp real	ref_pres real	flow_1 real	flow_2 real	flow_c real	temp_ctrl real	temp_supply real	temp_led real	status_inst integer	status_ctrl integer
1	2025-05-14 14:24:14+00	[null]	[null]	294.26	1013.25	3.693	1.308	5.001	25	41	28	3	0
2	2025-05-14 14:24:09+00	[null]	[null]	294.26	1013.25	3.693	1.308	5.001	25	41	28	3	0
3	2025-05-14 14:24:04+00	[null]	[null]	294.26	1013.25	3.693	1.308	5.001	25	41	28	3	0
4	2025-05-14 14:23:58+00	[null]	[null]	294.26	1013.25	3.693	1.308	5.001	25	41	28	3	0
5	2025-05-14 14:23:54+00	[null]	[null]	294.26	1013.25	3.693	1.308	5.001	25	41	28	3	0

Monitoring and Visualization with Grafana

- **Need**:
  - Is data coming in?
  - Are instruments behaving as expected?
  - Are values within a normal range?

- **Grafana**
  - Open Source
  - Monitor real-time data from PostgreSQL
  - Custom charts for each instrument
  - Time Series Graphs
  - Export graphs

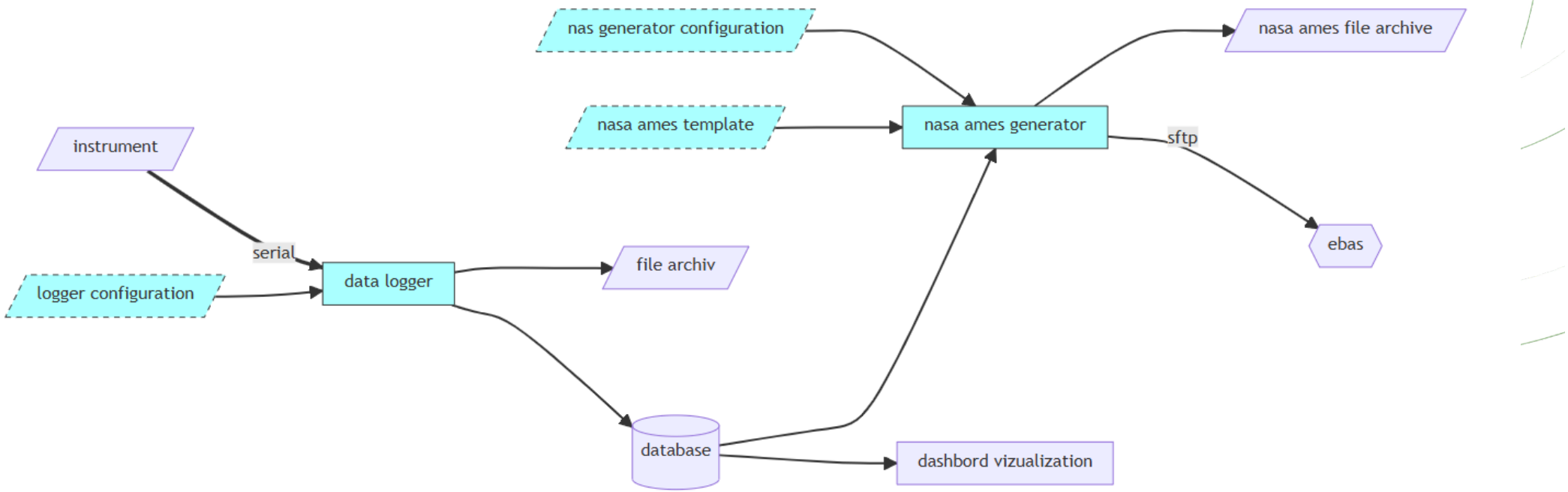
Creating and Submitting NRT Files to EBAS

- NASA-Ames Generator reads from the database and produces .nas files
- Requirement:
  - **Template File**: file format structure, including headers, units, and metadata
  - **configuration files**: specify the data source, table, time range, and destination

- Output includes:
  - Metadata (e.g. station, instrument type)
  - Timestamped data values
  - Units and flags (K or missing values)

- Files are saved locally
- From there, uploaded automatically via SFTP to EBAS

Complete NRT Workflow

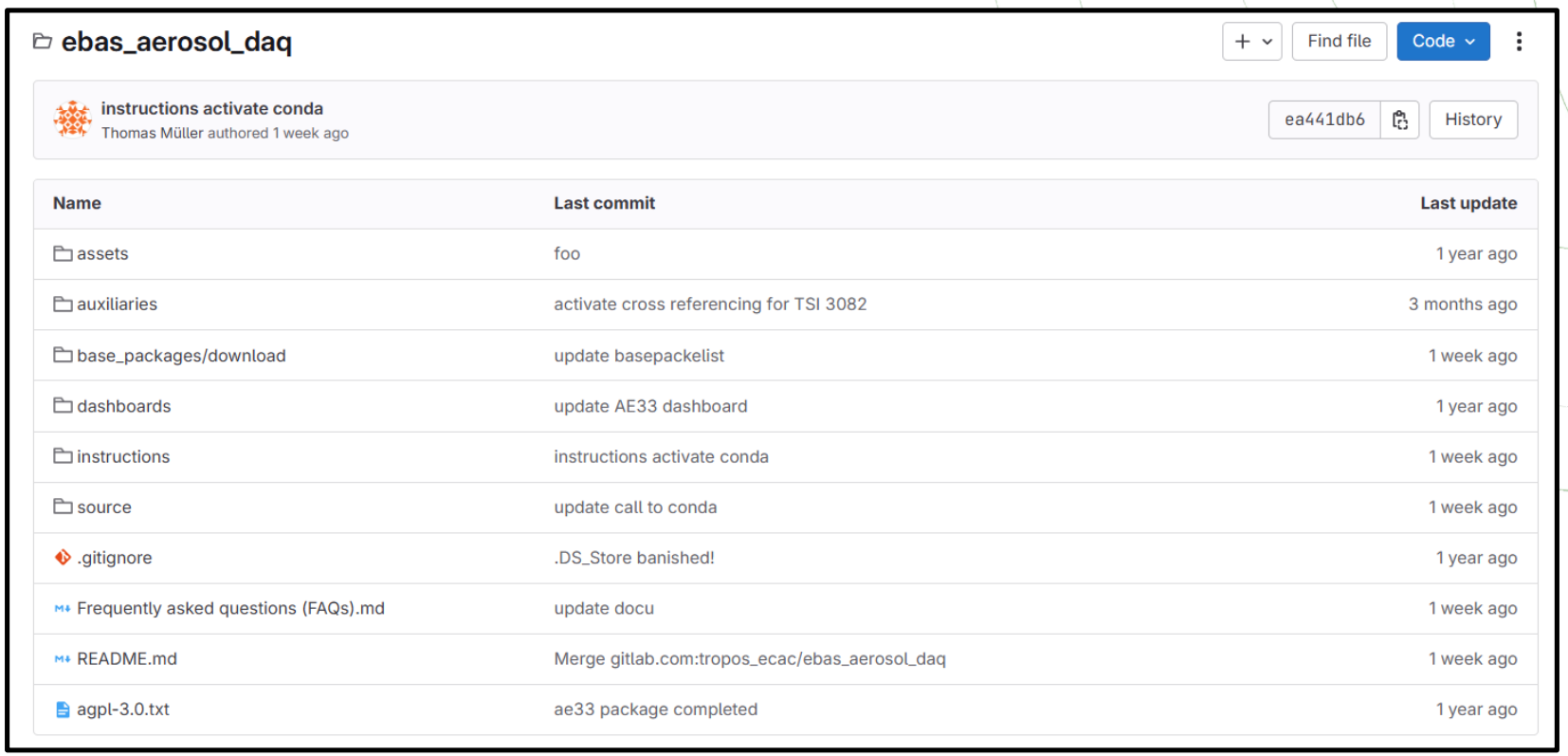


Installation Steps for NRT Acquisition & Submission

1. **Base environment**: "Install Miniconda and required packages"
2. **Install Database and pgAdmin**: "Install PostgreSQL and GUI tool to inspect the database"
3. **Data Logger**: "Configure logger with correct serial port, file paths"
4. **Grafana**: "Connect Grafana to the DB, import dashboards"
5. **NASA-Ames**: "Run the generator with config and template, submit file"

Accessing the EBAS Aerosol DAQ Repository

Visit the link shared: https://gitlab.com/tropos_ecac/ebas_aerosol_daq/-/tree/main?ref_type=heads



ebas_aerosol_daq

+ Find file Code

instructions activate conda
Thomas Müller authored 1 week ago

ea441db6 History

Name	Last commit	Last update
assets	foo	1 year ago
auxiliaries	activate cross referencing for TSI 3082	3 months ago
base_packages/download	update basepackelist	1 week ago
dashboards	update AE33 dashboard	1 year ago
instructions	instructions activate conda	1 week ago
source	update call to conda	1 week ago
.gitignore	.DS_Store banished!	1 year ago
Frequently asked questions (FAQs).md	update docu	1 week ago
README.md	Merge gitlab.com:tropos_ecac/ebas_aerosol_daq	1 week ago
agpl-3.0.txt	ae33 package completed	1 year ago

Step 1: Installing Base Environment

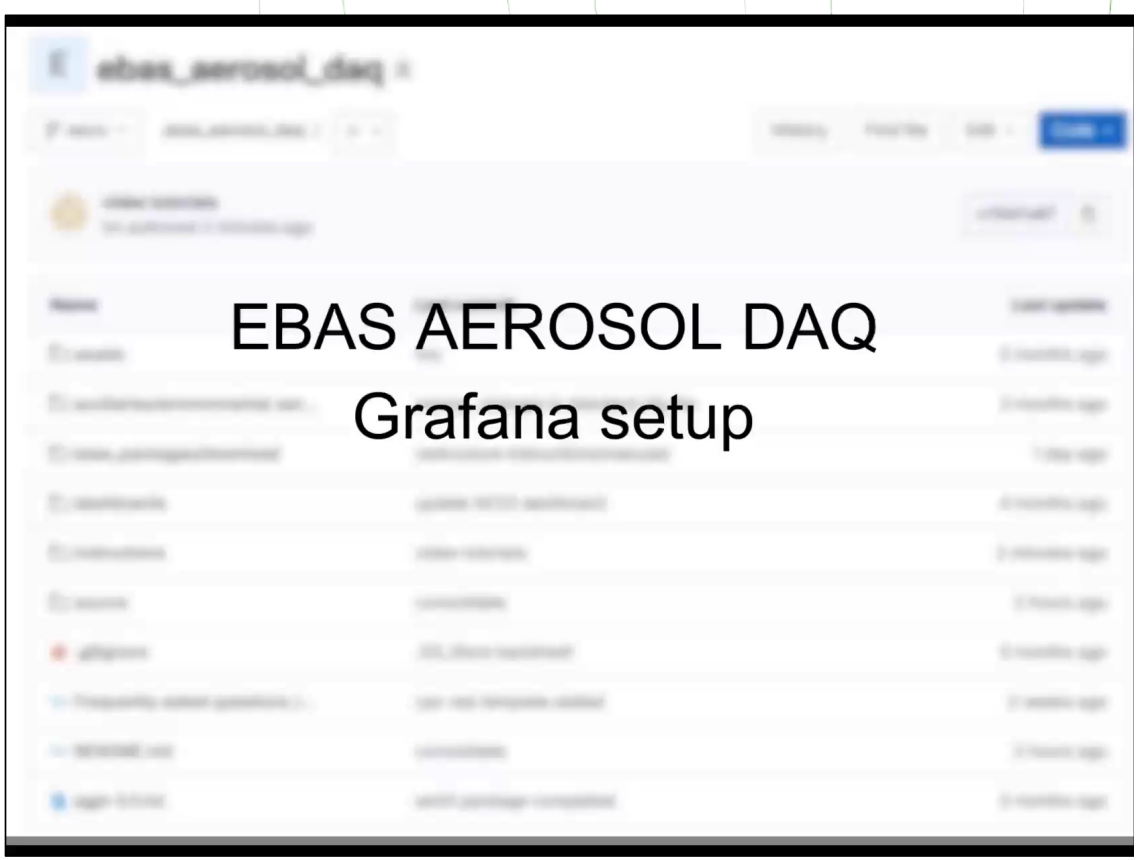
- Use Miniconda as the environment manager
- Register conda as a system environment variable:
  - *"c:\ProgramData\miniconda3\condabin\conda init cmd.exe"*
- Create Python 3.10 environment:
  - *conda create -n py310ebas python=3.10*
- Activate the environment each session:
  - *conda activate py310ebas*
- Download and install base packages:
  - *pip install ebas_aerosol_nrt-<version>-py3-none-any.whl*
  - *pip install ebas_io-<version>-py3-none-any.whl*

Step 2: Installing and Setting Up PostgreSQL Database

- Download PostgreSQL 15.x
- Default username/password: postgres / postgres
- Register a new server with a connection:
  - host=localhost,
  - port=5432
- Create new database (e.g., nrt_test), use lowercase names
- Set database time zone to UTC via database properties → parameters

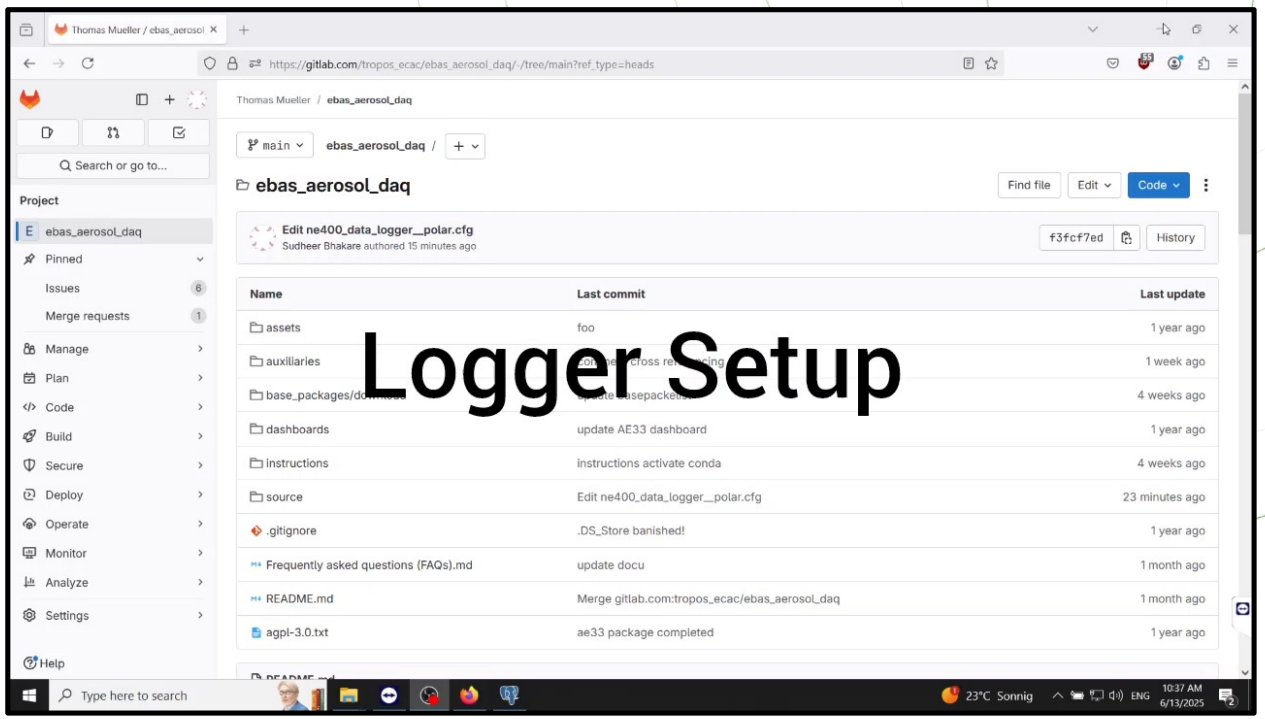
Step 3: Grafana Installation and Real-Time Dashboard Setup

- Install Grafana v10.2.3+ from grafana.com
- Open http://localhost:3000 — login with admin/admin
- Add PostgreSQL as a new data source:
  - Name: datasource_nrt_melpitz
  - Host: localhost:5432,
  - Database: nrt_melpitz,
  - User/Pass: postgres
- Install 'Clock' plugin
- Import dashboards: Dashboards



Step 4: Logger Installation and Configuration

- Download and unzip logger <version>.zip in this directory
- Configure logger file:
  - Replace %id with serial number
  - Set the correct COM port in the communication section
- Start real logging with a batch file. E.g. ae33_log__win.bat
- Check pgAdmin for the new table and test data



Step 5: NAS File Generation and EBAS Submission

- Download nasa_ames_generator from repository
- Update config:
  - Set instrument name and database table
  - Define time range and output directory
  - Link the correct template file
- Run batch script to generate .nas file
  - Example: ae33_nas__win.bat
- Output is stored locally
- Submit file to EBAS via SFTP

Troubleshooting Common Issues

- **pgAdmin is not starting and shows a runtime error**: See this Video
- Logger doesn't start → likely config error (Either you can see it on terminal or write log into text file)
  - https://www.yamllint.com or https://www.jsonlint.com
  - **Each logging task needs its own port** for communication between data logger and GUI. Check your configuration files, logger_settings/cmd_handler/port, and make sure that the ports are unique.
  - Error Message "Error, cant' connect to server", "Broken Pipe -> reconnecting". → The reason is probably that the logger has not been started or crashed.
- **Table not created**: Wrong table name or permissions
- **No .nas file generated** → check for template
- Grafana panel empty → wrong table name or query

Supplemetray Material

https://ebas.nilu.no/delivery/tools-and-tutorials/

ITINERIS

THANKS!

IR0000032 – ITINERIS, Italian Integrated Environmental Research Infrastructures System
(D.D. n. 130/2022 - CUP B53C22002150006) Funded by EU - Next Generation EU PNRR-Mission 4 “Education and Research” - Component 2: “From research to business” - Investment 3.1: “Fund for the realisation of an integrated system of research and innovation infrastructures”

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani