**Report**

**PROJECT OBJECTIVE AND BACKGROUND**

The AI4EU portal is an ambitious European initiative funded under the Horizon 2020 program that aims to create a comprehensive, on‐demand platform and ecosystem for artificial intelligence (AI) in Europe. It is designed to consolidate and share AI expertise, tools, algorithms, datasets, and infrastructure to foster research, innovation, and industrial deployment across the continent. The primary objective of this WP 3.1 assignment is to familiarize users with the AI4EU platform—an innovative environment designed to facilitate AI collaboration and streamline deployment processes through containerized applications. AI4EU serves as a comprehensive ecosystem enabling AI developers, researchers, and industry professionals to rapidly prototype, deploy, and share AI services and solutions.

In the context of this project, our team focused specifically on the initial user onboarding experience and the technical implementation of essential data handling components. The assignment included two core activities: firstly, completing and clearly documenting the onboarding process to enable seamless integration for new users; secondly, developing Docker-based containers designed to demonstrate foundational aspects of data management within the AI4EU platform.

This foundational work is crucial for ensuring that future collaborators and developers can efficiently engage with the platform's resources, build upon established methodologies, and easily adapt to the technical processes involved in AI4EU. Additionally, our developed Docker containers—comprising a data broker that transmits numerical data from a JSON file and a data processing node that computes squared values—demonstrate a practical, reusable example of inter-container communication. This setup not only supports the immediate goals of WP 3.1 but also paves the way for subsequent tasks, including orchestration and advanced data workflows managed by future teams, such as WP 3.2.

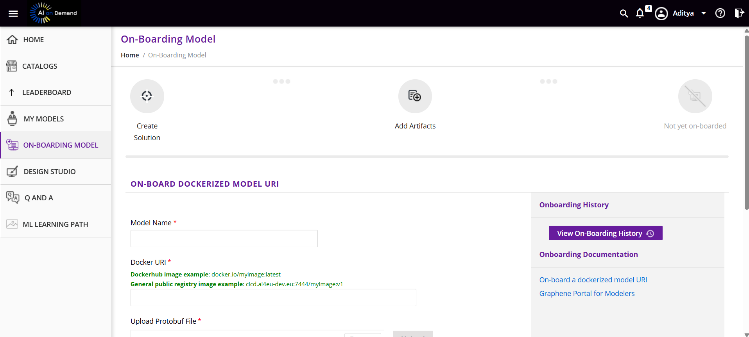
Through these targeted activities, this project aims to significantly reduce barriers to entry for new users while simultaneously providing robust, reproducible examples to facilitate ongoing development and collaboration on the AI4EU platform.

**CORE COMPONENTS AND KEY ACTIVITIES**

**Onboarding Process:**

The onboarding process began with detailed exploration and systematic documentation of steps necessary to create an account on the AI4EU platform. This process involved clearly defined procedures such as account registration, email verification, profile setup, and platform navigation. Each step was meticulously recorded, supplemented by screenshots to illustrate the exact process and reduce potential confusion.

A screenshot of a computer

AI-generated content may be incorrect.Below are some screenshots of the AI4EU platform which will help the user to understand the onboarding process.

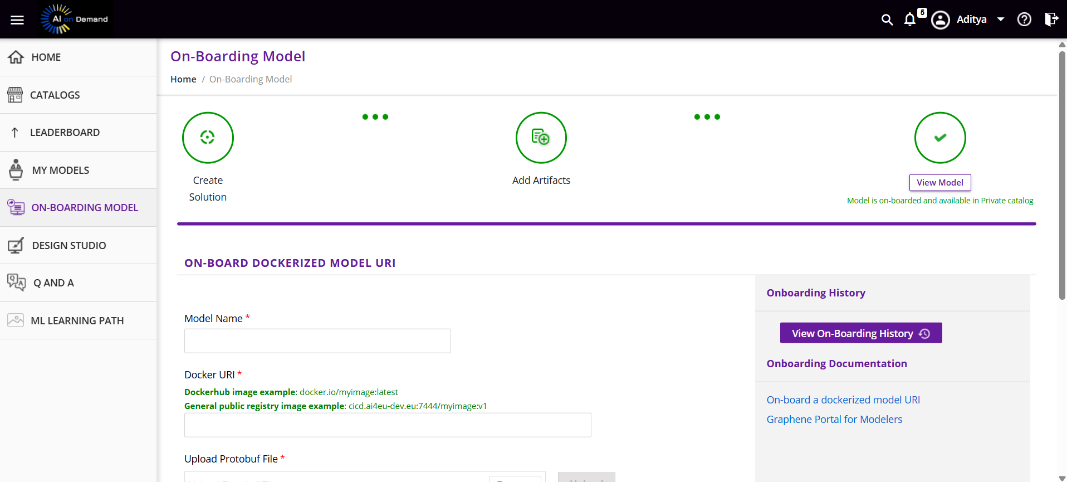
 Fig. 1 Homepage Fig. 2 Onboarding Model Page

Fig. 3 Onboarded Model

This documented guide was condensed into an accessible Notion document, carefully designed to facilitate ease of use and understanding for newcomers to the platform. This concise yet comprehensive documentation ensures that any future user, irrespective of their familiarity with the AI4EU environment, can seamlessly initiate and complete onboarding without encountering significant hurdles.

1. **Git:**

Install Git from **git-scm.com** to clone the repository. Verify with **git –version.**

1. **Python 3.8+ :**

Python will be our primary language for this project. Download and install Python from **python.org**. Verify the installation by running **python --version** in your local terminal.

1. **Code Editor:**

Visual Studio Code (VS Code) or any editor could be used for viewing and editing code.

1. **AI4EU Account:**

An account on the AI4EU platform is required to access the designer tool.

* **STEP 2: Setting Up the Python Environment:**

1. Open the terminal on your local computer, Create a virtual environment, “**python -m venv env”** and Activate it.
2. Install the required packages:

Run “**pip install grpcio grpcio-tools protobuf ”** to install gRPC dependencies.

* **STEP 3: Cloning the Repository**

1. Open the terminal and navigate to the directory to store the project.
2. Clone the repository: **git clone https://github.com/AI4EU-Graphene/EnergyConsumption-WP3.1.git.**
3. Navigate to the project of Energy Consumption.
4. Folders like **client/, server/,** and **solution/** would be visible.

A screenshot of a computer

AI-generated content may be incorrect.

* **STEP 4: Exploring the Repository**

1. **“CLIENT/”** containsthe scripts to interact with the models like **app-client.py.**

A screenshot of a computer

AI-generated content may be incorrect.

1. **“SERVER/”** holds the model implementation and **.proto** files like **energy\_databroker\_pb2\_grpc.py**

A screenshot of a computer

AI-generated content may be incorrect.

1. “**SOLUTION/”** includes the **solution.zip** file and deployment files for the WP 3.2 team.

A screenshot of a computer

AI-generated content may be incorrect.

**ONBOARDING TO THE PLATFORM**:

The AI4EU platform is the place to design and connect the pipeline components. Onboarding to the platform requires basic steps like:

* **STEP 1: Creating an AI4EU Account:**

1. Visit the AI4EU platform website, Click “Sign Up” and enter your email and password.
2. Check the inbox for a verification email, and click on the link to activate your account and log in with the credentials.

A screenshot of a computer

AI-generated content may be incorrect.

* **STEP 2: Exploring the dashboard:**

1. After logging in, the AI4EU dashboard will be the main workspace**.**
2. It has a simple but effective interface with sections like “Projects,” “Designer,” and “Documentation.”

A screenshot of a computer

AI-generated content may be incorrect.

* **STEP 3: Completing the Onboarding project:**

1. Navigate to the “Projects” section, click “Create New Project”, and Name it “Energy Prediction Onboarding”
2. Add a sample node, connect it to the processing node and run the workflow to test it.
3. Save the project, run it, and check for any errors.

A screenshot of a computer

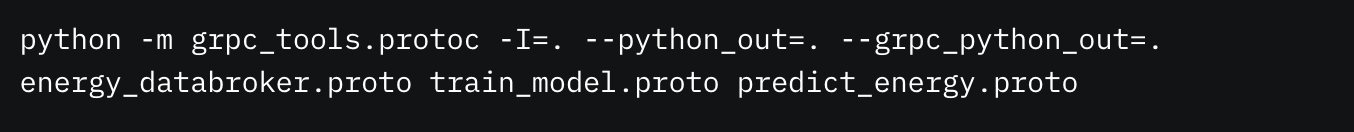
AI-generated content may be incorrect.

**SETTING UP THE PIPELINES:**

Setting up the Energy Prediction pipeline locally requires some steps:

* **STEP 1: Compiling the gRPC services:**

1. Activate the Python virtual environment on the local computer.
2. Navigate to the **server/protos/** from directory: **cd server/protos.**
3. Run the gRPC compilation command to generate Python stubs:



1. Files like **energy\_databroker\_pb2.py** and **energy\_databroker\_pb2\_grpc.py** will be generated in the **protos/** folder.

**gRPC Compilation Command:**

A screenshot of a computer

AI-generated content may be incorrect.

**Data Broker gRPC Definition:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Training gRPC Definition:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Prediction gRPC Definition:**

**A screenshot of a computer

AI-generated content may be incorrect.**

* **STEP 2: Building Docker Containers:**

1. Navigate to the **server/** directory: **cd server**.
2. Each model has its own subdirectory with a **Dockerfile.** like
   1. **databroker/Dockerfile** for the Data Broker.
   2. **train/Dockerfile** for the Training model.
   3. **predict/Dockerfile** for the Prediction model.
3. Next comes building the containers, which:
4. **docker build -t energy-databroker -f databroker/Dockerfile .**
5. **docker build -t energy-training -f train/Dockerfile .**
6. **docker build -t energy-prediction -f predict/Dockerfile .**

* **STEP 3: Using AI4EU Designer to create workflow:**

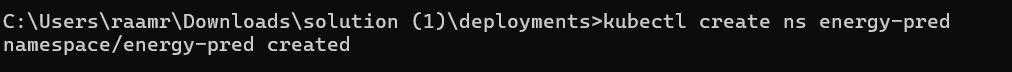
1. Navigate to the “Designer” section in the AI4EU platform and create a new workflow named “Energy Prediction Pipeline”.
2. Add the three nodes of Data broker, Training and Prediction
3. Data Broker: Reads from CSV File.
4. Training: Trains the Linear Regression model.
5. Prediction: Energy Consumption is predicted to a CSV by outputs.
6. Connect the nodes from the Data broker to Training, which sends the “training data”; Training to Prediction, which sends the “trained data”; and Prediction to output, which “saves the data results”.
7. Test the workflow

A screenshot of a computer

AI-generated content may be incorrect.

* **STEP 4: Deploy with Kubernetes:**

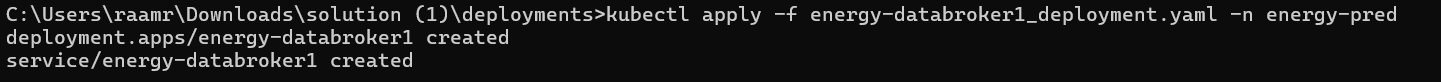
1. Run Docker’s Desktop Kubernetes and create a namespace of **kubectl create ns energy-pred.**



1. Deploy each of the models.
2. **kubectl apply -f solution/deployments/energy-databroker1\_deployment.yaml -n energy-pred .**

A screenshot of a computer

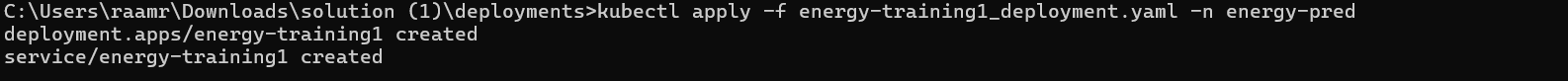
AI-generated content may be incorrect.



1. **kubectl apply -f solution/deployments/energy-training1\_deployment.yaml -n energy-pred**

A screenshot of a computer program

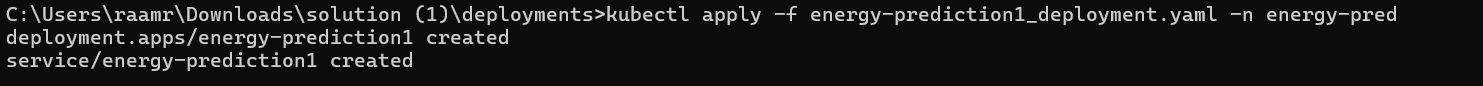
AI-generated content may be incorrect.



1. **kubectl apply -f solution/deployments/energy-prediction1\_deployment.yaml -n energy-pred**

A screenshot of a computer program

AI-generated content may be incorrect.



1. Verify the Deployment: **kubectl get all -n energy-pred**.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**RUNNING THE PIPELINE:**

* **Step 1: Prepare Sample Data**

1. Create a sample **input.csv** in the **server/data/** directory with columns like **SquareFootage**, **AverageTemperature**, **NumberOfOccupants**, etc.

**A black background with white text

AI-generated content may be incorrect.**

1. The file path should match the specifications in the Data Broker Configuration.

* **STEP 2: Running the Client Script:**

1. Open the **client/** directory: **cd client**.
2. Run the client script to interact with the pipeline: python app-client.py.
3. This script triggers the Training model to build a Linear regression model and requests predictions from the prediction model.
4. Check the **server/** directory for the output **predictions.csv,** which contains the predicted energy consumption values.

A screenshot of a computer program

AI-generated content may be incorrect.

**TROUBLESHOOTING:**

* If the gRPC compilation fails, check whether the **grpcio-tools** are installed and the **.proto files** are in the correct directory.
* If the Docker shows build errors, then check the **Dockerfile** for missing dependencies or incorrect paths.
* If the Kubernetes Pods are not running then run **kubectl describe pod <pod-name> -n energy-pred** to debug.
* Check the matching of gRPC message types of node connections for Designer related issues.

**CONCLUSION:**  
By following the steps with the visual walkthrough, the WP 3.1 Energy Prediction model can be set up on any local system. Users can now clone the repository, onboard to the AI4EU Platform, build and deploy the pipeline using Docker and Kubernetes and run the prediction workflow. WP 3.1’s output prediction **“predictions.csv”** file can now be used for orchestration for the WP 3.2 team and can also be used for future analysis. The troubleshooting section can be referred to for any issues. All the repositories can be accessed for free through (**github.com/AI4EU-Graphene/EnergyConsumption-WP3.1**).