



Machine Learning integration

Last updated: 03/03/2026

This content applies to the latest CD version of Cumulocity.

Specifications contained herein are subject to change and these changes will be reported in subsequent versions.

Copyright © 2026 Cumulocity GmbH.

The name Cumulocity GmbH and all Cumulocity GmbH product names are either trademarks or registered trademarks of Cumulocity GmbH and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

This software may include portions of third-party products. Third-party terms are set out in a 3rd-party-licenses file linked to or included with each installation package.

Table of Contents

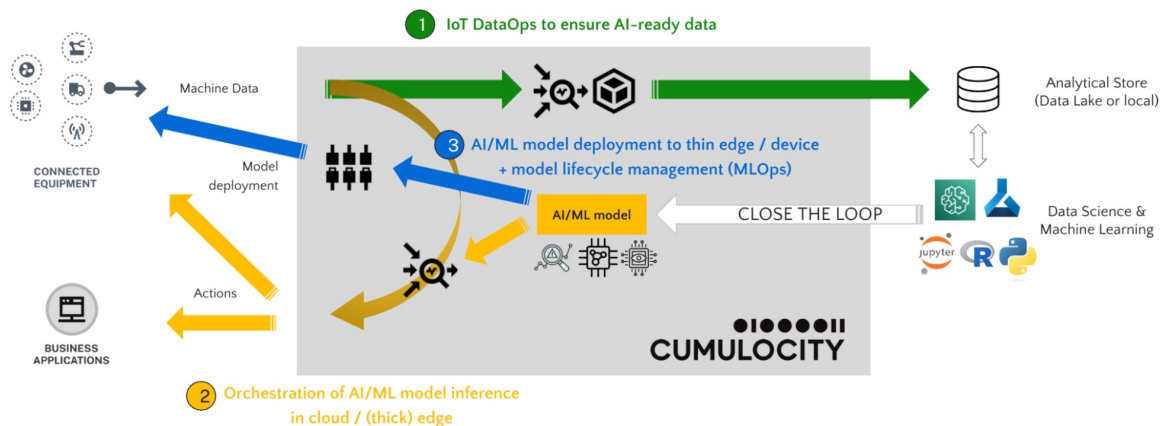
Table of Contents	3
INTRODUCTION	4
PREPARE YOUR DEVICE DATA (IOT DATAOPS)	5
CREATE AND BRING YOUR OWN AI/ML MODEL (BYOM)	6
OPERATIONALIZING YOUR AI/ML MODELS IN THE CLOUD	7
DEPLOY YOUR AI/ML MODEL	7
SCENARIO A: EXTERNAL HOSTING	7
SCENARIO B: EMBEDDED HOSTING USING A CUSTOM MICROSERVICE	8
SCENARIO C: EMBEDDED HOSTING USING A GENERIC MICROSERVICE	9
SETTING UP A MODEL INFERENCE WORKFLOW	9
DEPLOY AND ORCHESTRATE YOUR AI/ML MODELS ACROSS DEVICES (EDGEAI WITH MLOPS)	11

INTRODUCTION

Artificial Intelligence (AI) and Machine Learning (ML) have become a ubiquitous part of modern technology, as they help deliver insights otherwise hidden in data for improved decision making and possibly automated responses or actions. By combining AI/ML with IoT, you can now leverage the large amounts of data generated by connected devices for learning based on real-world data and apply those learnings in use cases ranging from image and speech recognition to predictive maintenance and anomaly detection.

With Cumulocity we provide a product and tooling to support you along every step of the Machine Learning lifecycle:

1. Connect machines, ingest the raw machine data, perform data preparation to ensure AI-ready data and make it accessible for AI/ML model training in your data science tool of choice.
2. Focus on the operational aspects of the Machine Learning lifecycle which involves applying a trained model to the incoming IoT data to obtain predictions, scoring, or insights in the cloud and/or at the edge.
3. Seamlessly deploy and orchestrate AI/ML models not only in cloud or (thick) edge, but also at the device edge for an entire fleet of assets.



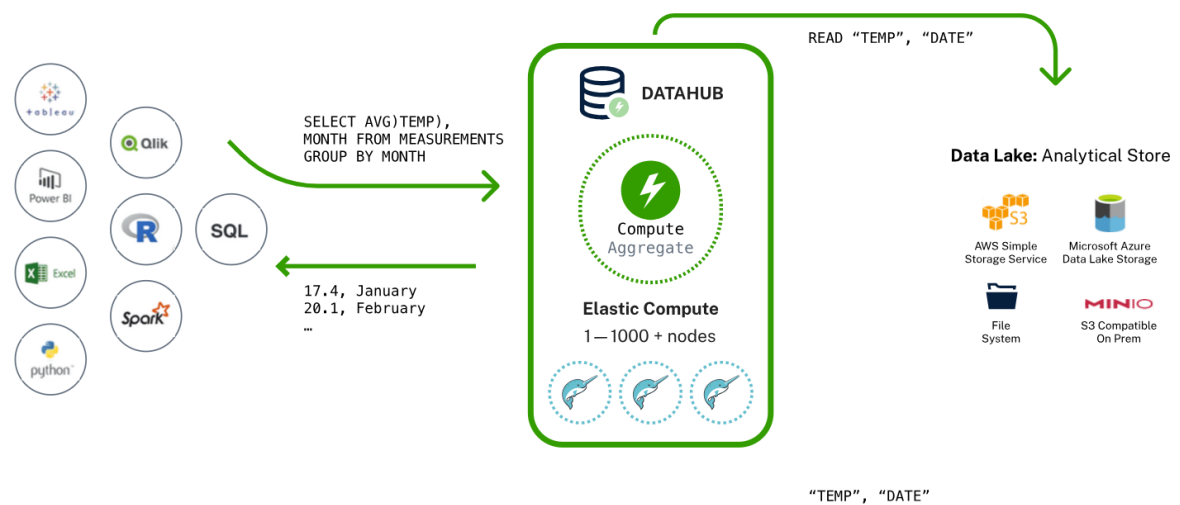
The next sections will explain how you can realize an end-to-end Machine Learning solution leveraging the Cumulocity platform and integrated Data Science & Machine Learning (DSML) components, tooling, as well as platforms. These could be open source components such as TensorFlow and/or tooling from some of our leading AI/ML partners such as Microsoft Azure, AWS (Amazon Web Services), IBM or Boon Logic.

PREPARE YOUR DEVICE DATA (IOT DATAOPS)

All AI/ML use cases start with defining the data requirements (next to defining the objective of the use case) and compiling a set of historical IoT data for training purposes. The IoT data, which is being ingested from the connected devices, machines or equipment is retained in the Operational Store of Cumulocity for a limited amount of time. In order to develop and train AI/ML models, being able to easily access and leverage historical data is essential. For storing your data long-term as well as easy data extraction, we suggest using DataHub which provides you with offloading and data querying capabilities.

Cumulocity DataHub offers an **SQL-based Query Interface** for querying the data lake and enables you to connect arbitrary applications that support ODBC, JDBC, or REST protocols. As such, you can connect existing tools and applications to Cumulocity, such as:

- Business Intelligence/reporting tools (using ODBC, JDBC)
- Data Science Workbenches (using ODBC, JDBC, python and others)
- Arbitrary custom applications (using JDBC for Java applications, ODBC for .NET, Python, node.js, and others, or REST for web applications)



See the [DataHub documentation](#) for more information.

CREATE AND BRING YOUR OWN AI/ML MODEL (BYOM)

There is a wide variety of open-source libraries (such as, TensorFlow®, PyTorch, Keras, Scikit-learn) and commercial 3rd-party tooling (such as, Microsoft Azure Machine Learning Studio, Amazon SageMaker, IBM Watson, MATLAB, Google Cloud) available for developing AI/ML models. Therefore, Cumulocity offers you the flexibility to have your data science team remain working in their own optimized technology stack but still leverage their results in the field.

Some examples of model creation in the mentioned tools that can inspire you:

- [Prediction of Remaining Useful Life with TensorFlow](#)
- [Anomaly Detection with AWS SageMaker](#)
- [Image classification with Azure ML Studio](#)
- [Remaining Useful Life Estimation with MATLAB](#)

If you do not have access to tooling/expertise in-house for the BYOM and/or are looking for a very specific AI/ML use case, such as predictive maintenance, there are more out-of-the-box solutions that we can recommend. For example, we have partnered with BoonLogic who provide ML-based anomaly detection capabilities with their Amber product. This product can be embedded as a custom microservice within Cumulocity (that is, scenario B) and can be integrated using Streaming Analytics. To facilitate the integration even further, a plugin has been created consisting of an integration microservice to manage the communication between Cumulocity and BoonLogic Amber plus a set of front-end widgets to perform the configuration and visualize the output of the anomaly detection. More information can be found at <https://github.com/Cumulocity-IoT/Cumulocity-Amber-Boon-Logic/>.

This section highlighted how you can access historical data and train an AI/ML model for your specific use-case. The next section details how you can bring your created AI/ML model into action on new incoming data, aka perform model inferencing or model scoring. When deploying a model outside of the training environment, **it is important to consider the portability of your model to a different platform**. To overcome potential issues, a community of partners has created the Open Neural Network Exchange (ONNX) standard for representing ML models, allowing models from many frameworks (including the ones mentioned earlier) to be exported or converted into the standard ONNX format. Once your model is in the ONNX format, they are able to run on a variety of platforms and devices.

OPERATIONALIZING YOUR AI/ML MODELS IN THE CLOUD

The section [Create and bring your own AI/ML model \(BYOM\)](#) explains how to access historical data for the purpose of training an AI/ML model. Now the next step is to operationalize your models and deploy them into data pipelines for real-time model inferencing.

DEPLOY YOUR AI/ML MODEL

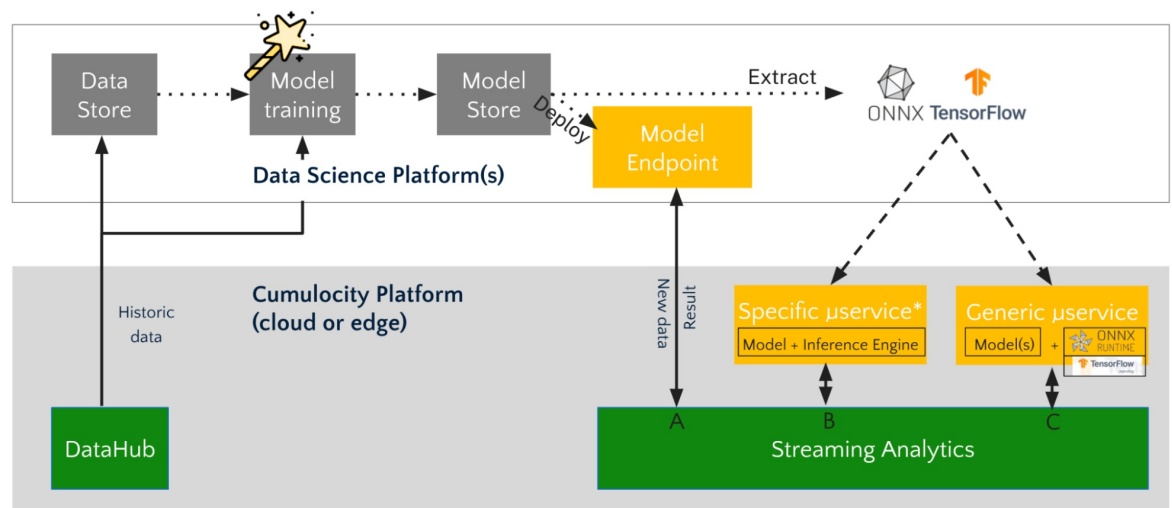
In general, there are three possible scenarios for deploying your AI/ML model, depending on the requirements of your use case:

- External hosting
- Embedded hosting using a custom inference microservice
- Embedded hosting using a generic inference environment

Using externally hosted AI/ML models offers advantages such as scalability, reduced infrastructure management, and access to cutting-edge AI/ML capabilities. This approach allows you to focus on your core functionality and leverage state-of-the-art Machine Learning without the burden of maintaining infrastructure.

Embedded hosting reduces the need for external data transfers and potential network-related delays. As such, this approach provides greater control over model customization, data privacy (by ensuring sensitive data remains within the platform's environment), and offers lower latency as data processing occurs within the platform's environment.

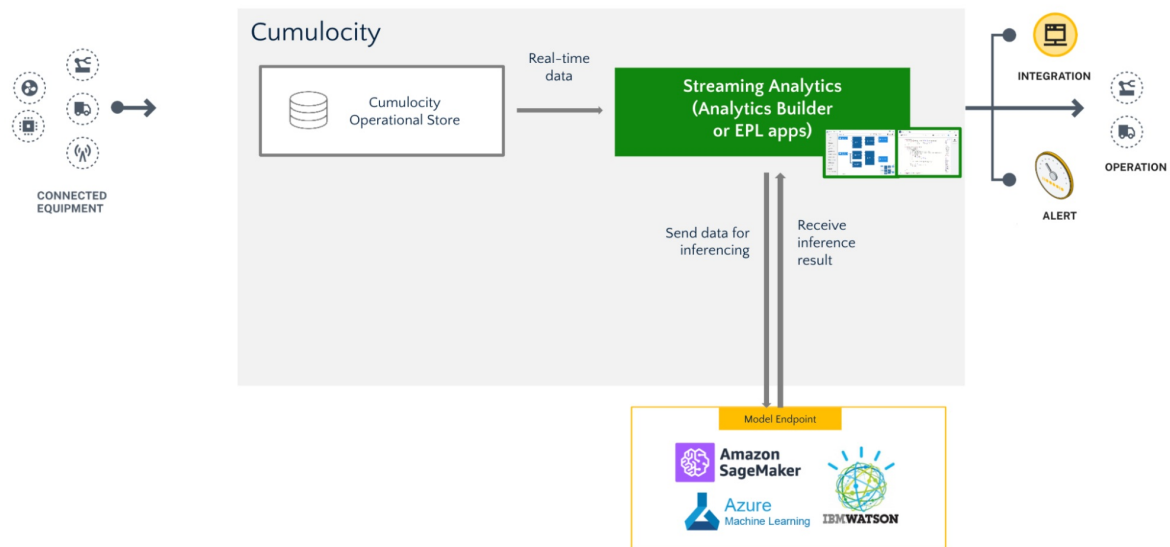
The image below illustrates a high-level architecture perspective, starting with providing the data via Cumulocity DataHub, training the model, making it available for various deployment scenarios (identified as A/B/C later on) to integrate in a workflow with Cumulocity Streaming Analytics.



SCENARIO A: EXTERNAL HOSTING

In this scenario, you leverage the AI/ML execution environment of a third party, which is typically closely related to the third party used to create and train the AI/ML model. The execution environment of the third party exposes an endpoint, which can be used for sending input readings and returning the model scoring output.

From an architectural perspective, scenario A looks like this:



INFO

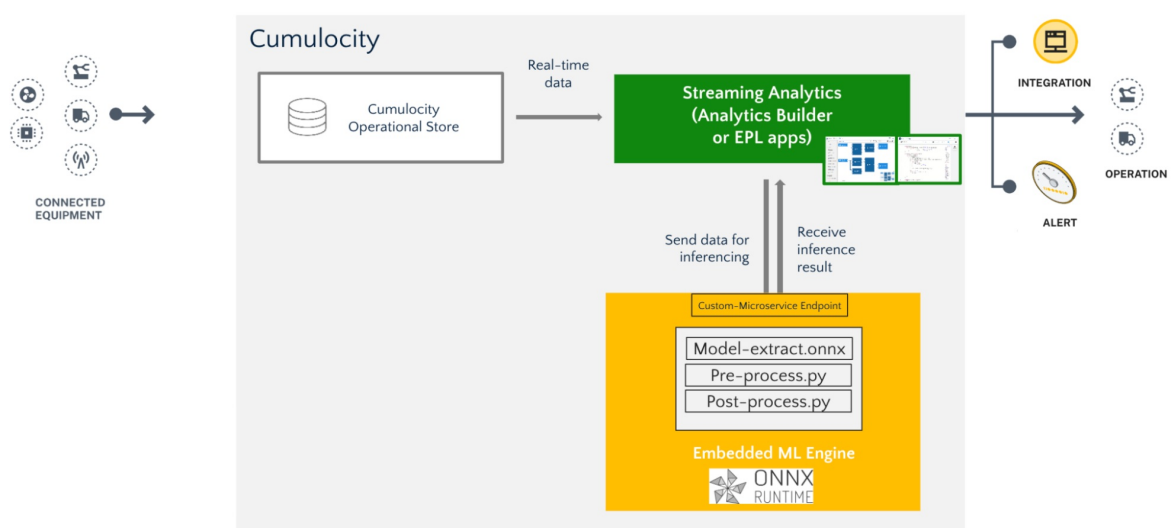
The following article in the Cumulocity Tech Community illustrates this scenario in more detail: [Leveraging Hyperscaler Clouds for Machine Learning Inferencing on Cumulocity Data](#).

SCENARIO B: EMBEDDED HOSTING USING A CUSTOM MICROSERVICE

In this scenario, you create and deploy a custom Cumulocity microservice which includes:

- An “extract” of the trained AI/ML model.
- The relevant libraries for inferencing.
- A POST request endpoint for sending input readings and returning the model scoring output.

From an architectural perspective, scenario B looks like this:



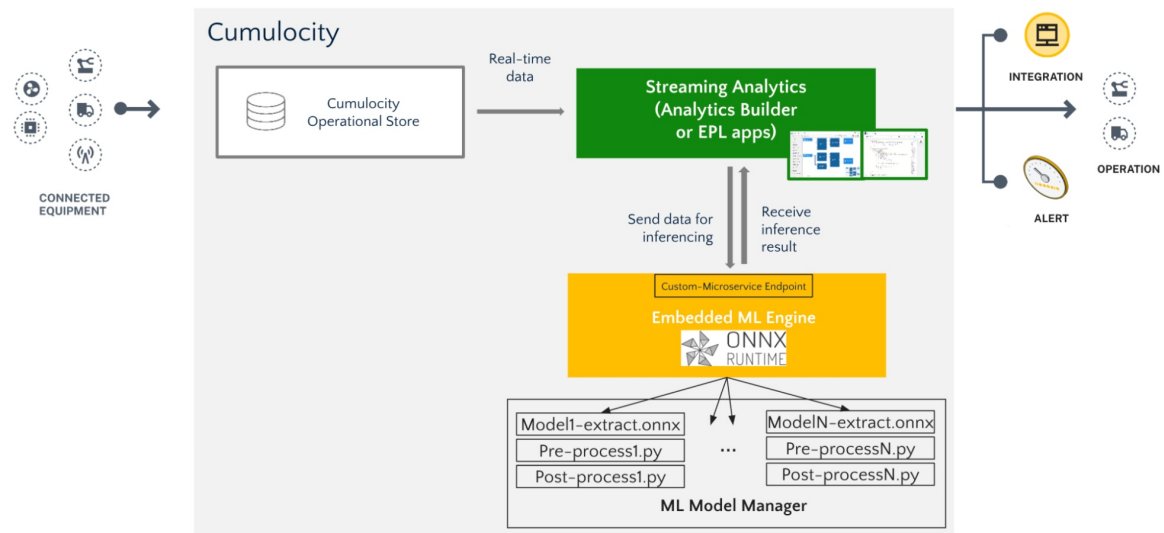
INFO

The following article in the Cumulocity Tech Community illustrates this scenario in more detail: [Performing Machine Learning Inference on Cumulocity Data using Open-Source Frameworks](#).

SCENARIO C: EMBEDDED HOSTING USING A GENERIC MICROSERVICE

In this scenario, you create and deploy a Cumulocity Tech Community microservice which has been purposely built to work generically with specific types of model “extracts” that are hosted alongside the microservice, such as, within the Cumulocity Tech Community file repository. Like scenario B, this microservice includes a POST request endpoint for sending input readings, this time complemented with the reference to the model of choice, and returning the model scoring output.

From an architectural perspective, scenario C looks like this:



INFO

A Cumulocity Tech Community article to illustrate this scenario is currently under construction.

SETTING UP A MODEL INFERENCE WORKFLOW

Once the AI/ML model is deployed, you need to set up a workflow to:

- Process the incoming data.
- Pass it to the deployed AI/ML model.
- Receive the model output.
- Process the model output to make decisions/create events, alarms, and so on.

To orchestrate the model execution, this workflow can be set up by leveraging the Streaming Analytics tooling, either Analytics Builder or EPL apps. More information on the specific tooling can be found in [Streaming Analytics](#).

INFO

In the Cumulocity Tech Community article for scenario B, a detailed description on how to create this can be found: [How to create an ML Inference workflow using Streaming Analytics](#).

🏠 > Machine Learning > Operationalizing your AI/ML models in the cloud

DEPLOY AND ORCHESTRATE YOUR AI/ML MODELS ACROSS DEVICES (EDGEAI WITH MLOPS)

With the increasing availability and affordability of compute for edge devices like controllers and gateways, executing workloads including AI/ML models on edge devices becomes increasingly attractive, especially as it boasts a wide range of benefits. This includes both operational aspects like reduced latency, costs and reliability as well as security aspects including for example local processing of sensitive data, the ability to meet data residency requirements and to run models in an air-gapped environment without internet connectivity.

However, operating workloads decentrally on edge devices introduces unique challenges compared to a centralized cloud environment. At the edge, devices often operate in diverse and constrained environments with limited compute power, intermittent connectivity and varying hardware configurations. Managing and deploying Machine Learning models in such decentralized setups can be complex, requiring solutions that ensure consistency, scalability and debuggability despite these limitations. Moreover, monitoring model performance and maintaining security across multiple edge nodes adds layers of operational complexity.

This is where Cumulocity steps in: **Cumulocity's device management capabilities** allow for efficient orchestration of the workload by providing a centralized management pane to deploy, instantiate, run, update, monitor and troubleshoot ML models as easy as in the cloud. This in detail includes:

- [Bulk operations](#) - to install ML models to any device as well as to update them.
- [Service management](#) - to monitor the status and metrics defined for and sent by the model (like accuracy, latency, resource consumption).
- [Log file retrieval](#), [alarming](#) and [event streaming](#) - to gain deeper knowledge on the operations of your model and being able to troubleshoot unexpected behaviour.
- [Remote access](#) capabilities and a [CLI tool](#) - to directly access devices and interact with the workloads running on top of them.