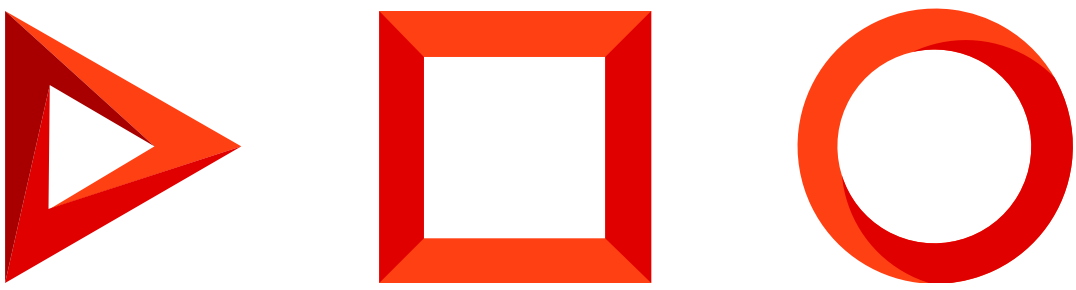
Creatio Academy

# Containerized components

## Machine learning service

Version 8.0



# Table of Contents

# Machine learning service

PRODUCTS: ALL CREATIO PRODUCTS

Machine learning service predicts values based on large volumes of historical data and current facts. Learn more in a separate article: Predictive data analysis.

**Attention.** Base knowledge of Docker, Linux or Windows administration is required to set up the machine learning service.

## Getting started

**Note.** Creatio on-site requires the “creatio predictive service on-site” license for the service to operate as intended. If you need to purchase the license, contact the responsible manager.

To set up the service, you need to have a server (physical or virtual machine) with Linux or Windows OS installed. Docker software is used for installing the service components. Download the archive containing the configuration files and installation scripts. Download archive. Depending on your company needs, you can use either Docker Community Edition (CE) or Enterprise Edition (EE). Learn more in the Docker Guide.

**Attention.** We recommend using a Linux-based server for production environment. You can only use a Windows based server for the development environment. Contact the support service to receive Docker containers that care compatible with Windows.

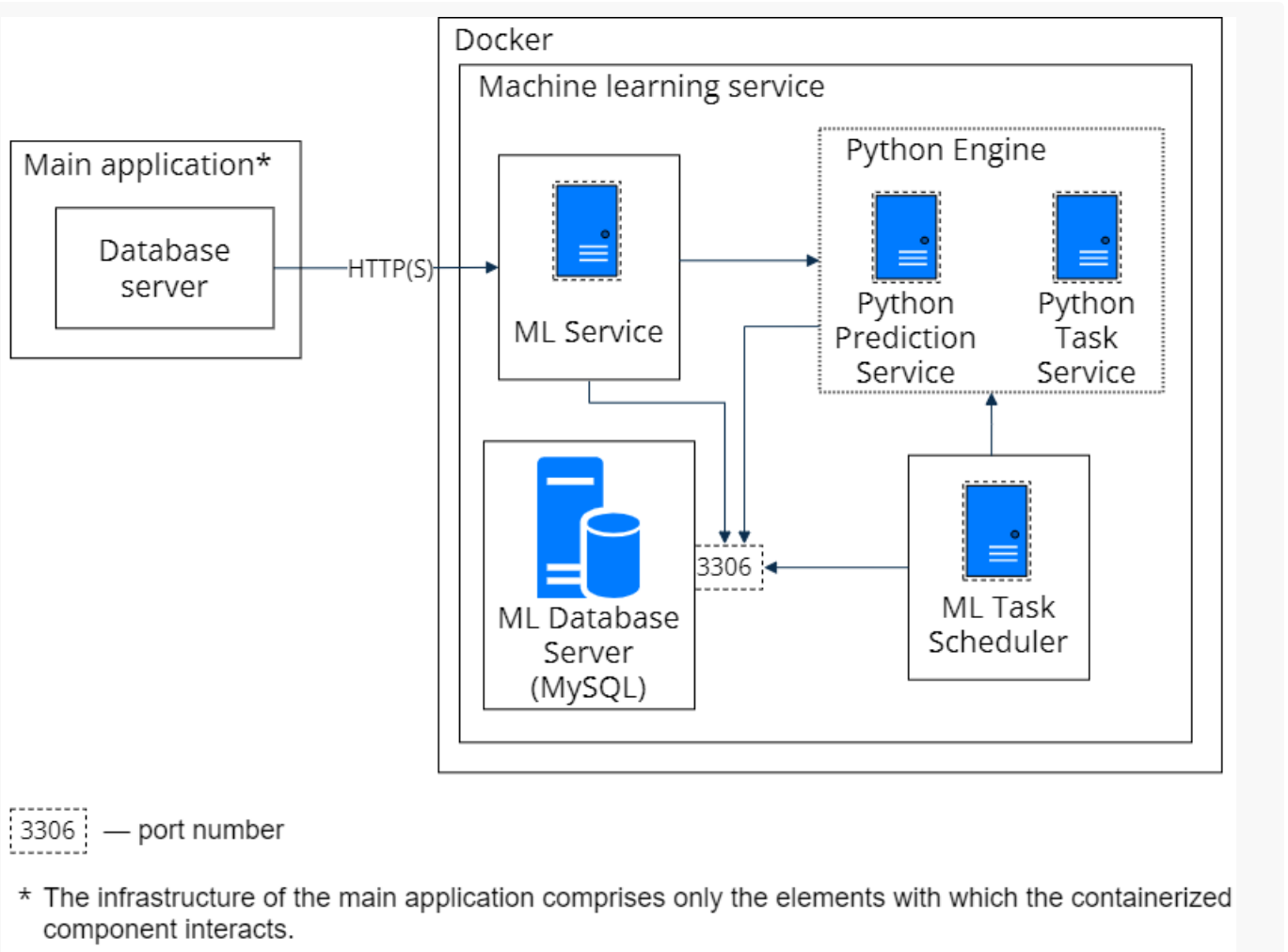
Use the requirements calculator to check the server requirements.

### Machine learning service components

The machine learning service uses the following components (Fig. 1):

- **ML Service**. Machine learning web service. The only component enabling external access.
- **Python Engine**. Machine learning wrapper service for open-source machine learning libraries.
- **ML Task Scheduler**. Task scheduler.
- **MySQL**. MySQL database. You can access it via the standard 3306 port.

Fig. 1 Machine learning service components




All the components are packed as Docker images for the convenient on-site installation of the service.

# Set up machine learning service

Follow this procedure to set up the machine learning service:

1. Install Docker. Read more >>>
2. Install Docker Compose. Read more >>>
3. Install and set up the service components. Read more >>>
4. Verify the installation. Read more >>>

## Install Docker

Docker installation on Linux platforms is covered in the Docker guide.

Run the “docker --version” command on Linux machine to verify the installed Docker version.

## Install Docker Compose

Docker Compose installation on Linux platforms is covered in the Docker guide.

Run the “docker-compose --version” command on Linux machine to verify the installed Docker Compose version.

## Set up the machine learning service components

Deploy the containers of the machine learning service components via the Docker Compose utility. Download the configuration files and scripts that are necessary to deploy and configure the service components. Download the archive

> **Note.** The configuration files contain all necessary default settings for a Linux based server.

The **archive structure of the configuration files and scripts** is as follows:

**/etc/**

**../ml-service/appsettings.json**. The ML web service configuration.

**../ml-service/log4net.config**. The setup of the web service logging level.

**../task-scheduler/appsettings.json**. The “ML Task Scheduler” utility configuration.

**../task-scheduler/log4net.config**. The setup of “ML Task Scheduler” logging level.

**Docker-compose.yml**. The “Docker Compose” utility configuration.

**.env**. The file that contains environment variables for running the components. For example, MySQL password.

> **Attention.** If you need to change the password to MySQL database, update it in the .env file and other configuration files that contain database access setup sections.

### Install the machine learning service components

1. Download and unzip the archive with the configuration files and scripts to a custom directory, for example, /opt/ml.
2. Using the Linux terminal, go to the /docker-compose catalog of the unzipped archive, for example, /opt/ml/docker-compose.
3. Run the “sudo docker-compose pull” command in the terminal. Wait until the download of the required service component images from the Docker Hub is complete.

   > **Attention.** If the server is disconnected from the Internet, download all required images to an Internet-connected computer manually (see the “docker-compose.yml” configuration file). Then use the sudo docker export and sudo docker import commands to transfer the images to the target computer as files.

4. Run the **sudo docker-compose run dbmigration** command to initialize the database structure. Wait until the command execution is complete.
5. Run the **sudo docker-compose up -d** command to launch the services. A “logs” folder will be created in the current catalog.

## Verify the setup of the machine learning service components

1. To verify the installation of ML web service, run the following command in Linux:

   **curl -X GET localhost:5005/readiness**

   The service must return the following response:

   **Healthy**

2. To verify the running of ML Task Scheduler, execute the following command in the Linux terminal:

   **curl -X GET localhost:5004/readiness**

   The service must return the following response:

   **Healthy**

3. To verify the running of R Engine, execute the following command in the Linux terminal:

   **curl -X GET localhost:8081/readiness**

   The service must return the following response:

   **R Service is ready**

4. To verify creating of tables, run the following command in the terminal:

   **docker exec -it DB Container Id mysql -u root --password=Supervisor ml -e "show tables;"**

   where [ *DB Container Id* ] is an identifier of the container with a database component. You can find out the container identifier using the **sudo docker ps** command.

   > **Example.** Verification of creating tables:
   >
   > docker exec -it [ *DB Container Id* ] mysql -u root --password=Supervisor ml -e "show tables;"
   >
   > As a result, the names of primary service tables should be displayed: “modelinstance”, “traindata”, “trainsession”, etc.

## Perform the setup in Creatio

To work with the prediction service in Creatio, fill out the following settings:

1. “Creatio cloud services API key” (“CloudServicesAPIKey” code) system setting. Cloud services need it to authenticate your Creatio instance.
2. “Periodicity of machine learning model training job” (“MLModelTrainingPeriodMinutes” code) system setting. It determines the models' synchronization launch frequency.
3. Add the prediction service's URL to the [ *Service endpoint Url* ] field for all the records in the [ *ML problem types* ] lookup.

# Update the machine learning service components

> **Attention.** We recommend saving a backup copy of MySQL database, before you update the services. Learn more in the Docker Guide.

1. Using the Linux terminal, go to the docker-compose catalog with the configured files, for example, /opt/ml/docker-compose.
2. Run the **sudo docker-compose stop** command to stop the service component containers.
3. Run the **sudo docker-compose pull** command in the terminal. Wait until the download of the required service component images from the Docker Hub is complete.
4. Run the **sudo docker-compose run dbmigration** command to initialize the database structure. Wait until the command execution is complete.
5. Run the **sudo docker-compose up -d** command to launch the services.

**Attention.** If your application already has configured and trained data models, we recommend retraining them after updating the service.