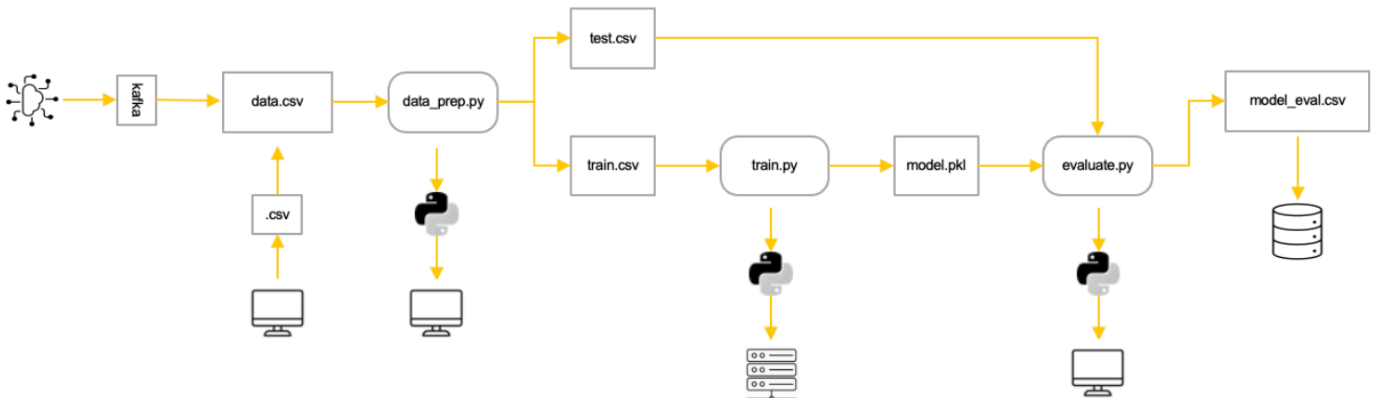


# File exchange container<>host.

As we mentioned in the introduction, a typical machine learning process goes through multiple steps.



These are often realized in different runtime environments, on different machines using different systems.

Coordinating such a solution requires not only process or infrastructure management, but also appropriate data exchange, such as in the form of files.

In the following lessons you will learn about the file exchange mechanisms offered by docker to implement the following scenarios:

1. **Docker provides only the runtime environment** (e.g. libraries, packages, etc.) - the complete data is on the local system.

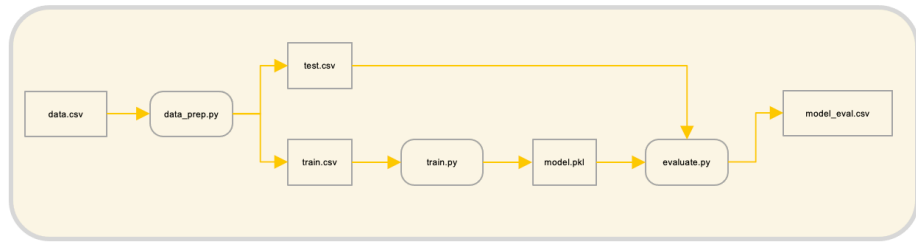


2. **The container contains both the environment and the components and artifacts necessary to run the project.** The local system is only to provide computing power.

Docker volume



Docker container



Local filesystem



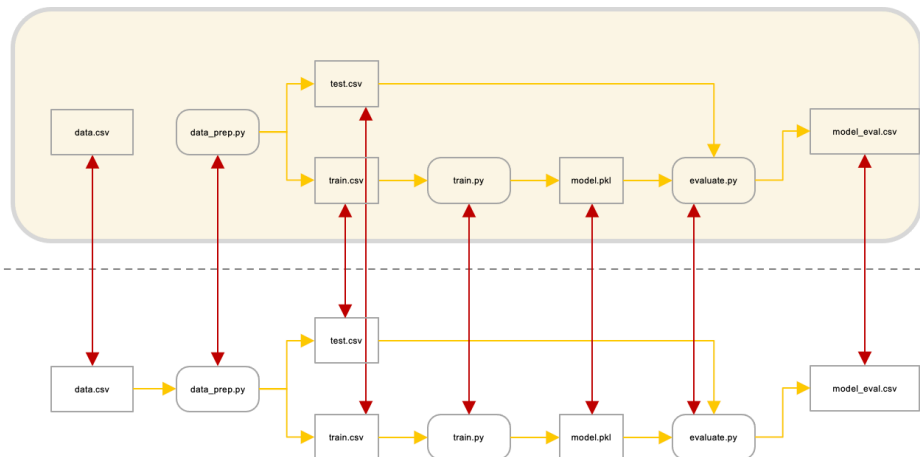
Computing only

**3 Docker provides synchronization between the local system and the container's file system.** As a result, any changes to the local system (or more precisely: to a specific location on that system) are reflected in the container, which in particular makes it possible to apply changes to artifacts/components without rebuilding the image.

Docker volume



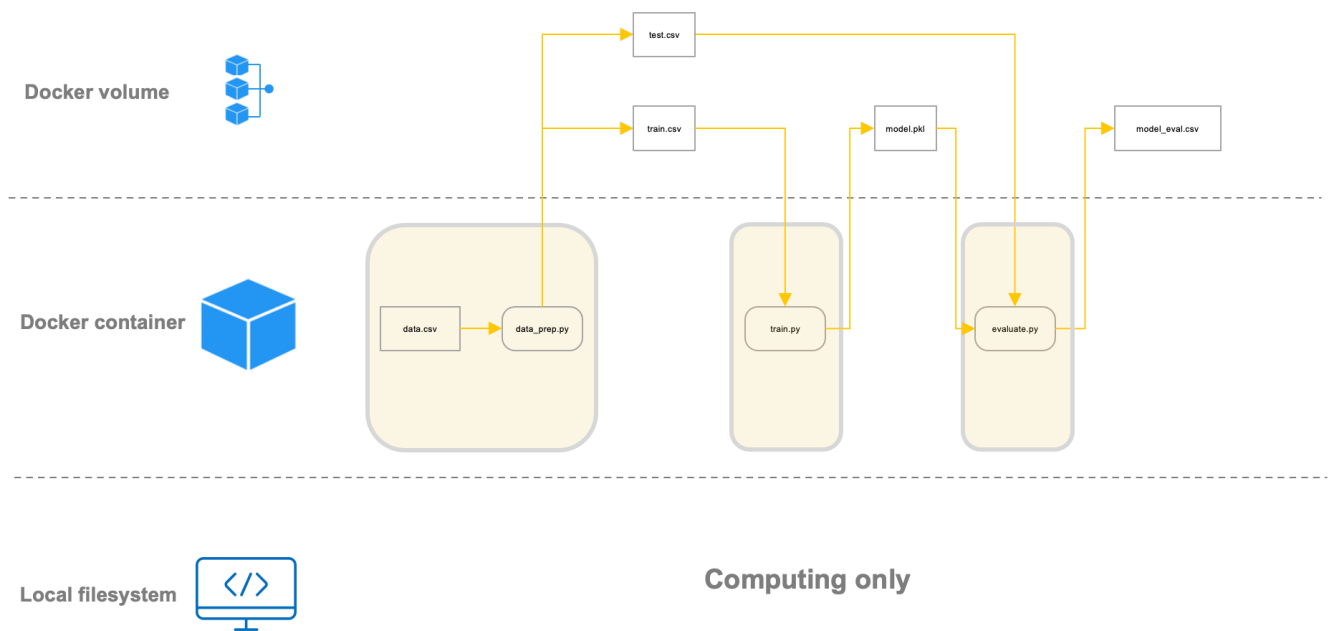
Docker container



Local filesystem



**4 Docker provides data exchange between containers and virtual volume.** No synchronization with a specific location on the local system is necessary, and this can be used both to keep a single container running and to exchange data between multiple containers (running in parallel or running sequentially).



In the following examples you will see the implementation of all these scenarios. Lets' start!

