Architecture

Magasin is a scalable end-to-end data toolset based on loosely-coupled open-source components that is natively run in a Kubernetes cluster.

Lets break it down:

1 Magasin is an end-to-end data platform

By end-to-end we mean that magasin enables an organization to ingest raw data from multiple data sources, transform it, run analyses and prediction models on the processed data, storage the results in a cloud filesystem, and enabling visualisation through dashboards.

2 Magasin scales with an organization needs

By scalabre we mean that magasin can be run with relatively low resources to provide service to a small team, to scale to thousands of users in a large organization.

Magasin leverages the power of Kubernetes. Kubernetes is a container orchestration system designed to automate the deployment, scaling, and management of containerized applications. It is an integral part of services offered by major cloud providers. Kubernetes, being open source, can also be set up on-premises. Indeed, for testing purposes, it is even suitable to install it on a desktop computer. By using kubernetes, we ensure the scalability of the solution.

Magasin uses Kubernetes in combination with Helm, a package manager for Kubernetes applications that eases the initial setup of the different components of magasin. Helm is the equivalent to apt, pip, npm, pacman, snap, conda. Using Helm, users specify the configuration of required Kubernetes resources to deploy magasin through a values file or command-line overrides.

3 Magasin is a loosely-coupled architecture

Magasin identifies the sets of needs for setting up an end-to-end data platform in an organization:

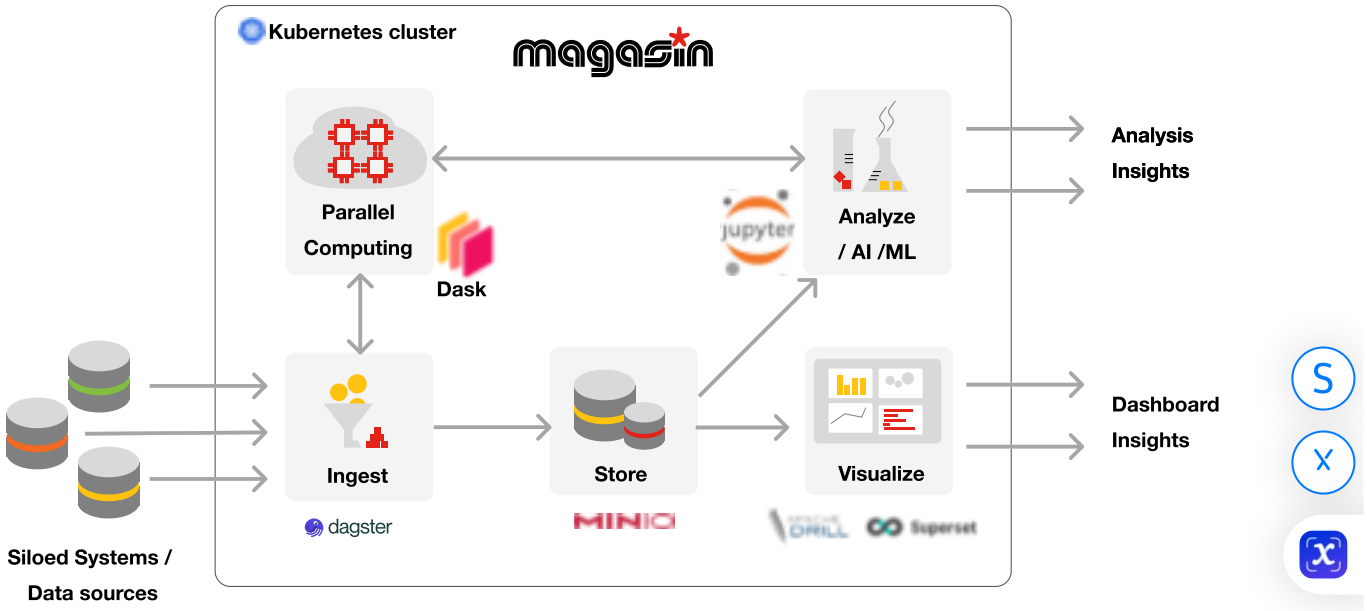
1. Analysis
2. Ingestion.
3. Storage.
4. Visualization
5. Parallel / Cloud compute

However, organizations may already have some of the elements in the architecture already in place. For example, an organization may already have a preferred data visualization platform such as Microsoft PowerBI or Tableau, and switching to another visualization tool may entail more costs than the licenses themselves. Magasin gets organizations covered on that. Each of the components of magasin is not strictly required for an implementation.

Technically, this is achieved by using the Helm package system in a slightly different way. Typically, a single Helm package includes all the components, setting up a more opinionated method of deploying a particular application. However, in Magasin, each component operates as an autonomous Helm chart. This design choice establishes a loosely-coupled architecture among its components, allowing you to install each component independently. Instead of enforcing a rigid structure for the entire architecture, Magasin embraces a more open and adaptable approach, fostering flexibility in component selection and integration.

4 Magasin's components

Magasin is built on top of a set of mature open source projects to conform an base architecture for deploying an end-to-end data platform.



Magasin architecture

4.1 Analysis: Notebook environment JupyterHub

A **Jupyter notebook** is an open-source allows you to create and share documents that contain live code, equations, visualizations, and narrative text. It is a powerful tool that combines code execution, rich text, mathematics, plots, and rich media into a single document. They are widely used in data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

The advantages of using Jupyter Notebooks are numerous. They provide an interactive computing environment that promotes exploratory analysis, making them ideal for data science projects.

Notebooks can be easily shared, promoting reproducible research and facilitating collaboration. They also allow for inline plotting, providing immediate visual feedback on the data being analyzed. Typical uses of Jupyter Notebooks include data analysis and visualization, machine learning, statistical modeling, and educational purposes. For instance, a data scientist might use a Jupyter Notebook to clean a dataset, perform exploratory data analysis, build a machine learning model, and then visualize the results, all within a single, cohesive document.

The multi-tenant JupyterHub component creates on-demand, isolated containers for authenticated users, each with persistent storage for their notebook workspace.

While it is common for data analysts to run Jupyter Notebooks locally on their computers, Magasin's approach allows them to leverage the power of the cloud. Similar to how Gmail offers the option to check your email directly in your browser without needing a client installed on your computer, Magasin enables data analysts to run their Jupyter Notebooks seamlessly in the cloud.

4.2 Parallel cloud computing: Dask Gateway

In addition to notebooks, magashin ships with a python library tool called Dask that enables data scientists and analysts to leverage the full capacity of the compute power of the Kubernetes cluster. A Dask cluster is a flexible tool for parallel computing in Python. It allows you to write python code that will be run in multiples machines taking advantage of all the compute resources of the kubernetes cluster in which magasin is installed.

It is composed of a central scheduler and multiple distributed workers. Dask works well at many scales, ranging from a single machine to clusters of many machines. It enables parallel processing and extends the size of convenient datasets from "fits in memory" to "fits on disk"1. However, it can also work across a cluster of multiple machines.

Dask cluster Dask Gateway allows easy utilization of a Dask cluster from notebook environments for distributed computation of massive datasets or parallelizable operations.

4.3 Ingestion: Dagster

The Dagster framework is the primary tool for orchestration of data pipelines for ingestion, transformation, analysis, and machine learning. Each pipeline is isolated and encapsulated on its own container, so different tasks may utilize different versions of a library.

In addition to pipeline isolation, it provides some advantages:

1. **A user interface** that provides visibility of pipelines' tasks, scheduling, run status, debugging messages, materialized assets, resources, and modes...
2. **Dagster pipelines are written in python**, which is a very familiar language for data scientist and data engineer.
3. **A framework for creating pipelines that scale.** Whereas early in the data ingestion processes, pipelines are simple and straightforward, with time, when mixing different sources, which requires more complex cleaning and the need of armonizing identifiiers, the pipelines become very prone to be built as spaguetty code. This lmiits the maintainability, discoverability of issues and efficiency of introducing new changes. With dagster, you get a way of building pipelines that will be more structured and easy to maintain.

Whereas dagster comes as the default data pipeline orchestrator, organizations may prefer to leverage other components for the ingestion such as

1. OpenFn
2. Apache Airflow

5 Store: A file based approach

In the magasin architecture, as general approach, we stand to store data assets as files. In particular, we recommend the use of Apache parquet file format.

The main reason to use a file based approach is:

1. First, because it is an economic way to store data. Storage services in the cloud or in premises is relatively cheap.
2. Second, because it does provide more flexibility when changes on the underlying structure are introduced, at least compared with setting up a SQL database downstream.
3. In addition, it allows also to easily store more types of data such as documents or images.
4. Lastly, in terms of governance and sharing the datasets, the problem is simplified to setting up file sharing permissions.

To support this file based approach, there are two components that are introduced in the architecture. The first one is MinIO, which provides magasin with a layer that introduces agnosticity against the cloud provider. The second one is Apache Drill, which provides a SQL query engine that eases the extraction of insights from the files.

5.0.1 MinIO: A cloud agnostic approach

Magasin can be installed in any cloud provider on in premises. However, each cloud provider has a different service to store data. In order to provide a consistent way of storing data we have included as part of the standard list of components MinIO, a high-performance object storage system designed for cloud-native and containerized applications has been included as part of the magasin.

Founded in 2014, MinIO offers an S3-compatible API, enabling seamless integration with existing cloud storage ecosystems. It is known for its simplicity, scalability, and speed, making it a popular choice for organizations seeking efficient data storage solutions. MinIO's architecture is optimized for modern data workloads, leveraging erasure coding and distributed techniques to ensure data resilience and high availability. With its lightweight footprint and easy deployment on standard hardware, MinIO empowers developers to build scalable storage infrastructures tailored to their specific needs, whether for on-premises, hybrid, or multi-cloud environments.

Whereas MinIO comes out of the box as part of the default magasin setuop there may be organizations that prefer use other options:

1. Garage S3 buckets (Open Source / Self hosting)
2. Azure Blob Containers(Commercial, Microsoft Public cloud),
3. Amazon S3 Buckets (Commercial / Amazon Public cloud) or

4. Google Cloud Storage (Commercial / Google Public cloud)

Other organizations may even opt to directly store the ingested data into a database such as PostgreSQL.

5.0.2 Query engine: Apache Drill

The last piece of the file based approach is Apache Drill. Apache Drill is an open-source, schema-free query engine that provides a SQL interface to a wide range of non-relational datastores, such as NoSQL databases and collections of files such as JSON, CSV, ESRI shapefiles, SPSS & SAS formats, Parquet, and others.

While data marts for specific business functions or locations traditionally require hosting and maintenance of a relational database on a server or virtual machine, Apache Drill enables comparable functionality without need for running and hosting a database or maintaining schema changes from source systems over time.

5.1 Visualization: Apache Superset

Apache Superset is an open-source business intelligence product with comprehensive charting, dashboarding, and querying capabilities.

Apache Superset consumes SQL data sources, so Apache Drill is a nice companion for Superset to work with magasin's file based approach.

Organizations may opt-in to use other visualization options.

For example, consuming the ingested data directly from commercial solutions such as Microsoft PowerBI, Tableau or InfoGram.

6 Mag-cli

Mag-cli is the command line interface of magasin. This components helps to manage the different modules of magasin and it makes easier to perform common administration tasks.

For each component (dagster, drill, superset etc.) allows to interact performing common tasks for the administrators and users of magasin such as opening the component user interface in the browser, viewing logs, etc.

It is developed in a modular way, so it is possible to extend its functionality for providing support to an alternative module.

7 What's next

- Getting started with magasin. Learn how to install, and start using magasin with a simple example.
- Why magasin?. Why magasin was created and what is the gap it is fulfilling.

- Implementing SDGs through magasin Learn about the magasin magasin plays in implementing the Sustainable Development Goals (SDGs).