Sep 13, 2022 · 5 min read

Data as a Product: Make It Real with Event-Driven, Serverless Data Pipelines

Updated: Jan 12, 2023

What is a data product and how can you productise/eventify it with the help of the serverless model?

What's a Data Product?

Data as a Product is one of the four principles in data mesh, proposed by Zhamak Dehghani in 2019, which we've talked about in the previous post.

The idea of "data as a product" is, as Dehghani put it, to apply *product thinking* to data. A productised data should be feasible, valuable and usable like a actual product, and thus implies you should treat your data users as consumers. You need to think about what's the best for them - to apply *empathy* onto your data. When you cared about your data, it would be great for everyone and the company.

Some people thought data products is the synonym of *data monetisation* and *information products* - to make money out of data. But no; what Dehghani described is a way to produce and consume data under the decentralised architecture of data mesh. The data producing process should be moving from the traditional, centralised ETLs (*Extract, Transform, Load*) toward the domain source as *internal pipelines*, so that each domain team can autonomously output data on a logical data serving platform.

But What’s *Exactly* is a Data Product?

But what is a data product under the concept of data mesh? We can find ample clues in Dehghani’s articles:

Three characteristics that forms a data product

It's a Data Pipeline

Firstly, a data product is not just data, but a combination of the following elements:

- Code (of the data pipeline itself)
- Data and metadata
- Infrastructure (to run the pipeline)

Dehghani also mentioned:

> Domain data product owner must have a deep understanding of who the data users are, how do they use the data, and what are the native methods that they are comfortable with consuming the data. Such intimate knowledge of data users results in design of data product interfaces that meet their needs.

This makes a data product **extremely similar to a web API or microservice**, which is integrable to other systems as long as it uses a standard interface (like RESTful or gRPC) and can be run anywhere as a containered application. This is also the so-called **data as a service (DaaS)** in data fabric. A data pipeline can also encapsulate extra transformations and have internal storage to work smoother with other pipelines.

It's Domain-Oriented

Like we've mentioned before, the core idea of data mesh is about *decentralisation* and *distribution*. The traditional data team, if the company has one, is often outside of data domains and has little incentive to improve them.

Instead, The data product (or pipeline) is now in the hands of each **domain data product owner**, who has both the qualification and responsibility to assure data quality as well as its suitable interface. They can make decisions faster, output data with care, and avoid the centralised bottleneck created by ETL processes.

It's Event-Driven

The decentralisation movement also changes the way how data pipelines run:

> Traditionally, a centralised and shared service implements these interfaces. The API life cycles are independent of the domain's data or pipeline. For example, to access the data, a data consumer goes to a central catalog to find the dataset and then directly from the central catalog reaches into the bowels of the platform storage system. In contrast, data mesh introduces intentionally designed interfaces made available by each data product as code, to discover and access the data.
>
> – Zhamak Dehghani, *Data Mesh* (2022)

In other words, a pipeline now runs *on demand* - triggered only by a user's request, and returns whatever result then. The pipeline only uses resources when you need it, not running all the time on some central server.

This is reminiscent to an old model dated from 1970s - the Event-Driven Architecture (EDA), which has seen lots of popularity by the rise of Function-as-a-Service (FaaS) platforms since mid-2010s. And this is provides a very important clue of how exactly can we build a real, feasible data product today.

Going Serverless Is the Way

> A serverless provider allows users to write and deploy code without the hassle of worrying about the underlying infrastructure. A company that gets backend services from a serverless vendor is charged based on their computation and do not have to reserve and pay for a fixed amount of bandwidth or number of servers, as the service is auto-scaling. Note that despite the name serverless, physical servers are still used but developers do not need to be aware of them.
>
> – Cloudfire, What is serverless computing?

There are lots of materials discussing about serverless, but the simplest definition is to be able to deploy applications without knowing (almost) anything about the infrastructure or the cloud platform. This is a key since it also fits the third data mesh principle proposed by Zhamak Dehghani:

> …the only way that teams can autonomously own their data products is to have access to a high-level abstraction of infrastructure that removes complexity and friction of provisioning and managing the lifecycle of data products. This calls for a new principle, *Self-serve data infrastructure as a platform to enable domain autonomy*.

These days, serverless and FaaS are mostly referred to the same thing. Since Amazon Lambda in 2014, followed by Google Cloud Functions and Azure Functions in 2016, the FaaS model has became very popular among cloud solutions. You can deploy a code function as an independent application comparably easily; the apps are very scalable (can increase service volume when necessary) and will only be charged with its actual usage (they are all event-driven; in fact, Amazon Lambda sometimes get referred as *event-driven ETLs*).

It sounds like FaaS is the answer for making data pipelines a reality. But here we have to discuss a minor issue exists in most FaaS platforms.

…But FaaS May Not Be *The One*

Interestingly, FaaS has one major disadvantage of *cold start* - it takes time to generate a new container image and deploy it ("warm it up") with your code function before execution.
For example, your container(s) would be running in Amazon Lambda for about 30 to 45 minutes. After that, the resources would be revoked, and new containers will have to be generated from scratch all over again. Depending on platform and languages, the cold start can take between 3~6 seconds or even longer. Some platforms allow you to set a minimum number of running instances, but you'll have to pay for that too!

This is how Jacques Chester, the author of *Knative in Action* (Knative is an open-source FaaS project from Google), describes FaaS as suitable for *unpredictable, latency-insensitive demand*. FaaS works great if you have dynamic requests that may not be happening all the time. But in the age of AI you would have huge, constant demands that have to be addressed as soon as you can. The data warehouse- for lake-level of operating is not the scenario FaaS is intended for.

FST Network's Logic Operating Centre (LOC), while being serverless and looks very similar to FaaS at the first glance, adopts a slightly differently strategy: The data pipelines will not be directly deployed as containers. Instead, they would be "injected" and executed by a common, scalable and persistent runtime when the time comes. Combined with powerful, fast Amazon S3-compatible databases, all data processes can be run almost right away. How exactly we implemented this would be the topic of another post in the future.
In short, LOC is designed for deploying and managing data pipelines - or data processes - in a pretty easy way. The goal is to create domain- and event-driven data APIs that is discoverable and usable by other users across the serverless platform. The end results - data products - can then be delivered in a fast and high-quality way, thus bringing value for the enterprise.

FST Network aims to revolutionise the data landscape by creating world's first cloud-native and serverless Data Mesh platform. Logic Operating Centre (LOC) is fully capable to productise and eventify your data, with easy-to-deploy data pipelines that seamlessly connects everything across your organisation, bringing in agile, accessible and trustworthy business insights. The data race is constantly changing and fiercer than ever - but we are there to give you a winning edge.

Icon credits: flaticon.com