

SERVERLESS (FAAS) VS. CONTAINERS

WHEN TO PICK WHICH?

Software Engineering-CSCI 771/405

Chuanyao Lin

Hunter College
Spring 2021

Why should we know ?

- Serverless and containers enable us to build the applications based on the concepts of the microservice architecture.

Reminders of the microservice

- A microservice is an independent and self-contained software component that runs in its own process and communicates with other microservices using lightweight protocols.
- Microservices architecture is an architectural style in which the system is constructed from communicating microservices. It is well suited to cloud-based systems where each microservice can run in its own container.

What is Serverless?

- Serverless Computing is an application design that incorporate third-party “Backend as a Service” (BaaS) services, and/or that include custom code run in managed, ephemeral containers on a “Functions as a Service” (FaaS) platform.

What is Serverless?

- 1. Serverless was first used to describe applications that significantly or fully incorporate third-party, cloud-hosted applications and services, to manage server-side logic and state. These types of services have been previously described as Baas. For example: the cloud-accessible databases (e.g., Parse, Firebase), authentication services (e.g., Auth0, AWS Cognito), and so on.
- 2. Serverless can also mean applications where server-side logic is still written by the application developer, but, unlike traditional architectures, its run in stateless compute containers that are event-triggered, ephemeral (may only last for one invocation), and fully managed by a third party. One way to think of this is "Functions as a Service" or "FaaS". AWS Lambda is one of the most popular implementations of a Functions-as-a-Service platform at present.

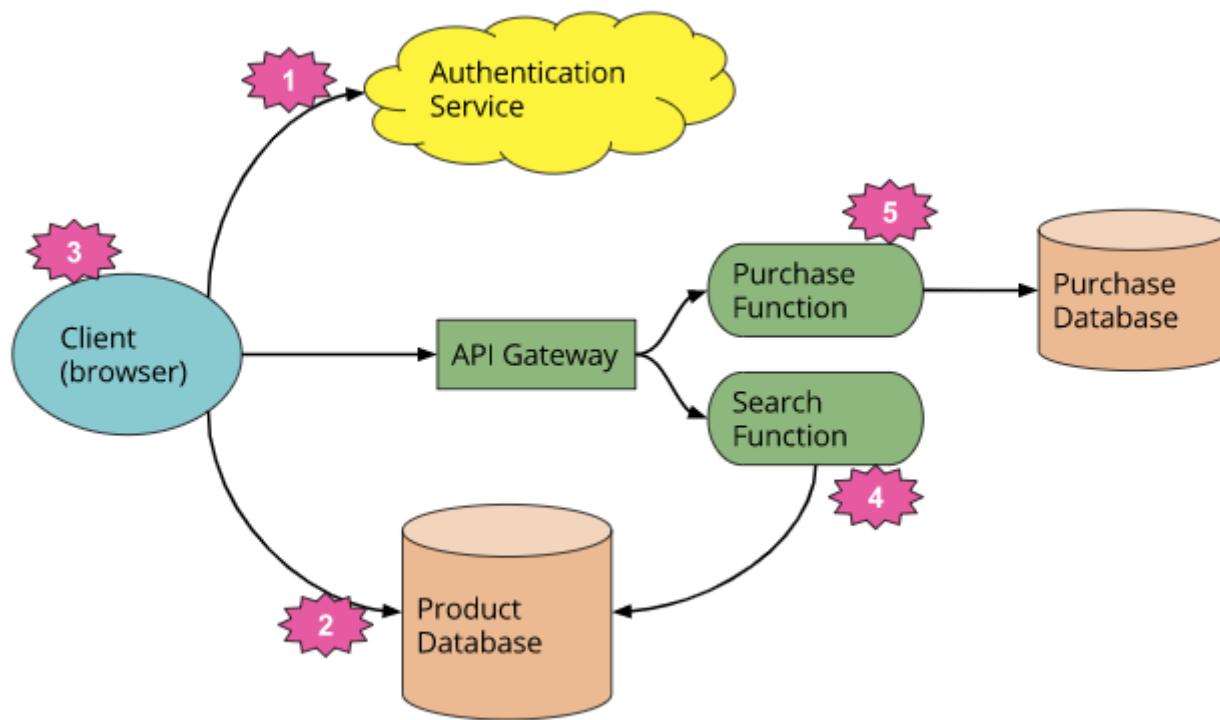
Example of FaaS

- A typical ecommerce app- an online pet store with a traditional three-tier client-oriented system.



Example of FaaS

- A Serverless(FaaS) architecture may end up looking more like this:



Example of FaaS

- A different example is an online advertisement system



In the Serverless world this looks as follows:



Features of FaaS

- FaaS is about running backend code without managing your own server systems or your own long-lived server applications
- Functions in FaaS are typically triggered by event types defined by the provider.
- In a FaaS environment we upload the code for our function to the FaaS provider, and the provider does everything else necessary.
- The most basic premise of a serverless setup is that the whole application--all its business logic--is implemented as *functions* and *events*.

Comparing FaaS with the container

- Advantages of Serverless
 - Zero administration
 - Significantly reduced administration and maintenance burden
 - Faster time-to-market
 - Auto-scaling
 - Pay-per-execution
 - Zero cost for idle time
- Disadvantages of serverless
 - No standardization
 - Vendor lock-in
 - "Black box" environment
 - Cold starts

Comparing FaaS with the container

- Advantages of containers
 - Control and Flexibility
 - Easier migration path
 - Portability
- Disadvantages of containers
 - Administrative work
 - More manual intervention
 - Scaling is slower
 - Hard to get started

When to pick which

- 1. Costs
- 2. Portability
- 3. Convenience

References

- [1] Sommerville, I. (2020). *Engineering software products*. Pearson.
- [2] Serverless Architectures
- <https://martinfowler.com/articles/serverless.html>
- [3] Serverless (FaaS) vs. Containers - when to pick which?
- <https://www.serverless.com/blog/serverless-faas-vs-containers>