Introduction

This lesson introduces the concept of serverless architecture, goes over its definition, and discusses the concepts of BaaS and FaaS.

WE'LL COVER THE FOLLOWING

What is Serverless Architecture?

Serverless is quite the buzz word this year. It's a shiny new term that has caused many debates, even conflicts, among developers. In fact, at the moment you are reading this, there are already many meet-ups and conferences dedicated to serverless architecture, its benefits, and pitfalls.

So, what is *serverless architecture* all about?

Like for many new concepts in technology these days, there is no straightforward answer to this question.

Definition

Before we define what serverless architecture is, let's first define what serverless architecture isn't. This way we can skip the confusion that the word itself insinuates. Serverless architecture doesn't mean that there are no servers, just the physical, as well as virtual infrastructure, is hidden from developers.

This means that the developer is basically using a *third-party* to handle infrastructure problems. Therefore, the developer expects that the third-party will enable the resources that are necessary for the application, or function, to be executed properly.

By now, we as a community have agreed that when an application is serverless, it uses one of two concepts:

- The **Backend as a Service** (**BaaS**) concept – This application depends on third-party service that is running on the cloud.
- The **Function as a Service** (**Faas**) concept – This application depends on a custom code that has been run in short-living containers.

These two concepts are converging; BaaS solutions usually have FaaS integrated somewhere in their architecture because of a need to create custom backend functionalities. For example, some of the solutions that started as a BaaS ended up being FaaS. An excellent real-life great example is Auth0

Auth0.

You just got a basic introduction to Serverless architecture. In the next lesson, we will look at it in more detail.