Serverless Deployments in Kubernetes

This lesson states the benefits of Kubernetes-based serverless computing and also introduces Knative.

We'll cover the following

Kubernetes-based serverless computing

At this point, I must make an assumption that you, dear reader, might disagree with. Most companies will run at least some of their applications in Kubernetes. It is becoming a standard API that will be used by everyone.

Why is that assumption important? If I'm right, then almost everyone will have a Kubernetes cluster. Everyone will spend time maintaining it, and everyone will have some level of in-house knowledge of how it works. If that assumption is correct, it stands to reason that Kubernetes would be the best choice for a platform to run serverless applications as well. As an added bonus, that would avoid vendor lock-in since Kubernetes can run almost anywhere.

Kubernetes-based serverless computing provides quite a few other benefits.

- We are **free to write our applications in any language**, instead of being limited by those supported by function-as-a-service solutions offered through cloud vendors.
- We **aren't limited to writing only functions**.
- A **microservice or even a monolith** could run as a serverless application. We just need to find a solution to make that happen. After all, proprietary, cloud-specific, serverless solutions use containers as well. The standard mechanism for running containers is using Kubernetes.

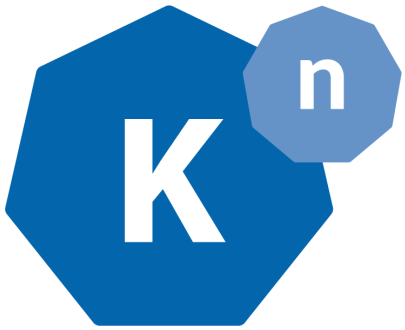
There's an increasing number of Kubernetes platforms that allow us to run serverless applications. We won't go into all of those, but instead, fast-track the

conversation by stating that **Knative** is likely going to become the de-facto

standard in how to deploy serverless loads to Kubernetes. Maybe, it's already the most widely accepted standard by the time you read this.

Knative

Knative is an open-source project that delivers components used to build and run serverless applications on Kubernetes. We can use it to scale-to-zero, to autoscale, for in-cluster builds, and as an eventing framework for applications on Kubernetes.



The part we're interested in right now is its ability to convert our applications into serverless deployments, and that means auto-scaling down to zero, and up to whatever an application needs. That should allow us both to save resources (memory and CPU) when our applications are idle, and to scale them fast when traffic increases.

We discussed what is serverless and I made an outlandish statement that Kubernetes is the platform where your serverless applications should be running.

We discussed what is serverless and I made an outlandish statement that Kubernetes is the platform where your serverless applications should be running. In the next lesson, let's talk about types of scenarios that are a good fit for serverless deployments.