

Disclaimer: I am a 5+ year AWS veteran, and have a strong bias that I am actively working on while I continue to work with Google Cloud Platform, Kubernetes, and CNCF cloud native tools and services.

This morning I had the pleasure of attending the *Deploying Serverless Apps to Kubernetes with Knative by Google Cloud* session at Kubecon.

I found the format to be incredibly useful with tech leads and program managers from the Knative team present to answer questions immediately. This was great for helping to get through minor hurdles and get the information disseminated very quickly to the entire audience of mostly engineers.

However, while the potential of [Knative](#) is undeniably high and the Knative team is exceptional, helpful and well intentioned, I am left with a lingering feeling of disappointment in Knative and its implementation.

Understanding Knative is beta, I cannot accept the *third* serverless platform from Google (Cloud Functions, App Engine, and now Knative) not having event based triggering as a core feature. We were barely able to *discuss* it with the technical leads present for the session, let alone play with anything.

The same can be echo'd for [Build](#), a sub-feature of Knative that as of this morning did not support recycling of failed builds. Furthermore, there was no documentation or proof of any implementation to a true pipeline service that could leverage Build, which leaves us even with the most production ready version of the feature with something that is at best a YAML replacement for a Jenkinsfile. Admittedly, the cleanliness and ability to natively support unique images for each step of a build was far less painful than similar Jenkins implementations, but nonetheless, I am left befuddled.

With AWS [Lambda](#), [CodeDeploy](#), [CodeBuild](#), and countless of other services well into production maturity and with ecosystems of support around them, I am sorely disappointed in what appears to be competition from the Kubernetes landscape.

Especially with the recent release of [Firecracker!](#)