

What is Tekton and How Does it Fit Jenkins X?

This lesson sheds some light on Tekton and also explains its role in Jenkins X.

We'll cover the following

- What is Tekton?
- How does Tekton fit with Jenkins X?

What is Tekton?

Those of you using serverless Jenkins X already experienced **Knative*. **Tekton** is a spin-off of **Knative**, and it is the essential component in Jenkins X. It is in charge of creating pipeline runs (a special type of Pods) when needed and destroying them when finished. Thanks to **Tekton**, the total footprint of serverless Jenkins X is very small when idle. Similarly, it allows the solution to scale to almost any size needed.



Tekton is designed only for “special” types of processes, mostly those associated with continuous integration and continuous delivery pipelines. It is not, however, suited for applications designed to handle requests, and those are most of the applications we are developing. So, *why am I talking about Tekton if it doesn’t allow us to run our applications as serverless?* The answer lies in **Tekton**’s father.

How does Tekton fit with Jenkins X?

Tekton is a **Knative** spin-off. It was forked from it in hopes of providing better

CI/CD capabilities. Tekton is the core component of the **Knative Build**

CI/CD capabilities. To be more precise, **Tekton** was born out of the **Knative Build** component, which is now considered deprecated. Nevertheless, **Knative** continues being the most promising way to run serverless applications in Kubernetes. It is the father of **Tekton**, which we've been using for a while now given that it is an integral part of serverless Jenkins X.

Now, I could walk you through the details of **Knative** definitions, but that would be out of the scope of this subject. It's about Jenkins X, not about **Knative** and other platforms for running serverless applications. But, my unwillingness to show you the ins and outs of **Knative** does not mean that we cannot use it. As a matter of fact, Jenkins X already provides the means to select whether we want to create a quickstart or import an existing project that will be deployed as a serverless application using **Knative**. We just need to let Jenkins X know that's what we want, and it'll do the heavy lifting of creating the definition (**YAML** file) that we need.

So, Jenkins X is an excellent example of both a set of serverless applications that constitute the solution, as well as a tool that allows us to convert our existing applications into serverless deployments. All we have to do to accomplish the latter is express our desire. Jenkins X will do all the heavy lifting of creating the correct definitions for our applications as well as moving them through their life-cycles.

Now let's move on to the hands-on part of this chapter and create a cluster in the next lesson.