Why Do We Need Jenkins X to Be Serverless?

This lesson compares the traditional Jenkins with Jenkins X and explains why being serverless was the way to go.

> **We'll cover the following** ∧
>
> - Drawbacks of traditional Jenkins
> - Serverless Jenkins X

Drawbacks of traditional Jenkins

There are quite a few problems with the traditional Jenkins. Most of us already know them, so I'll repeat them briefly.

- Jenkins (without X) **does not scale**.
- It is **not fault-tolerant**.
- Its **resource usage is heavy**.
- It is **slow**.
- It is **not API-driven**, and so on.

In other words, it was designed when those things weren't as important as they are today.

> 🔍 Jenkins had to go away for Jenkins X to take its place.

Serverless Jenkins X

Initially, Jenkins X had a stripped-down version of Jenkins but, since release 2, not a single line of the traditional Jenkins is left in Jenkins X. Now it is fully serverless thanks to **Tekton** and a lot of custom code written from scratch to support the

need for a modern Kubernetes-based solution.

Excluding a very thin layer that mostly acts as an API gateway, Jenkins X is fully serverless. Nothing runs when there are no builds, and it scales to accommodate any load. That might be the best example of serverless computing we have.

Continuous integration and **continuous delivery** pipeline runs are temporary by their nature. When we make a change to a Git repository, it notifies the cluster, and a set of processes are spun. Each Git webhook request results in a pipeline run that builds, validates, and deploys a new release and, once those processes are finished, it disappears from the system. Nothing is executing when there are no pipeline runs, and we can have as many of them in parallel as we need. It is elastic and resource-efficient, and the heavy lifting is done by **Tekton**.

> Continuous integration and continuous delivery tools are probably one of the best examples of a use-case that fits well in serverless computing concepts.

The next lesson will shed more light on Tekton and its role in Jenkins X.