

# Getting Started with Advanced Pipelines

This lesson explains what we will cover in this chapter.



The examples in this chapter work only with serverless Jenkins X. Nevertheless, pipelines defined in build packs use (almost) the same format as those used by serverless Jenkins X. When we create a new quickstart project or import an existing one into static Jenkins X, build pack pipelines are converted into `Jenkinsfile`. Therefore, even if you are not using serverless Jenkins X, advanced knowledge of writing `YAML`-based pipelines will help you when you choose to modify build pack pipelines.

So far, we relied mostly on pipelines created for us through build packs. No matter how much effort the community puts into creating build packs, we can be almost sure that they will not fulfill all our needs. Every organization has something “special” that inevitably leads to discrepancies between generic and tailor-made pipelines. So far, we extended our pipelines without knowing much about syntax. We haven’t explored the benefits additional instructions might provide yet.

You can think of the subject of this chapter as advanced pipelines, but that would be an overstatement. No matter if you’re using static or serverless pipelines, they are always simple. They should be simple since their goal is not to define complex logic but rather to orchestrate automation defined somewhere else. That does not mean that there are no complex pipelines, but rather that those cases often reflect misunderstandings and the desire to solve problems in the wrong places.



Pipelines are orchestrators of automation and should not contain complex logic.

---

Now, let’s define some objectives in the next lesson.

