Introduction

This lesson will get us introduced to Jenkins X, the role of Kubernetes with Jenkins X and the real power of Jenkins X.

We'll cover the following

- Kubernetes as a prerequisite
- The real power behind Jenkins X

Kubernetes as a prerequisite

To understand the **intricacies and inner workings** of Jenkins X, we first need to understand Kubernetes. However, you do not need to understand Kubernetes to **use Jenkins X**. That is one of the main contributions of the project. Jenkins X allows us to harness the power of Kubernetes without spending an eternity learning the ever-growing list of the things Kubernetes does.

Jenkins X helps us by simplifying complex processes into concepts that can be adopted quickly, and without spending months trying to figure out **" the right way to do stuff."** It helps by removing and simplifying some of the problems caused by the overall complexity of Kubernetes and its ecosystem. If you are in fact a Kubernetes ninja, you will appreciate all the effort put into Jenkins X. If you're not, you will be able to jump right in and harness the power of Kubernetes without ripping your hair out from frustration caused by the complexity of Kubernetes.

I'll skip telling you that Kubernetes is a container orchestrator, how it manages our deployments, and how it took over the world by storm. You hopefully already know all that. Instead, I'll define Kubernetes as a platform to rule them all. Today, most software vendors are building their next generation of software to be

Kubernetes-native or, at least, to work better inside it. A whole ecosystem is emerging and treating Kubernetes as a blank canvas. As a result, new tools are being added on a daily basis, and it is becoming evident that Kubernetes offers near-limitless possibilities. However, with that comes increased complexity. It is harder than ever to choose which tools to use.

- ***How are we going to develop our applications?***
- ***How are we going to manage different environments?***
- ***How are we going to package our applications?***
- ***Which process are we going to apply for application lifecycles?***
- **And so on and so forth.**

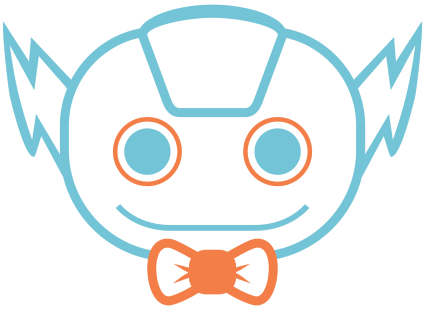
Assembling a Kubernetes cluster with all the tools and processes takes time and learning how to use what we assembled feels like a never-ending story. Jenkins X aims to remove those and other obstacles.

Jenkins X is opinionated. It defines many aspects of the software development lifecycle, and it makes decisions for us. It tells us what to do and how. It is like a tour guide on your vacation that shows you where to go, what to look at, when to take a photo, and when it's time to take a break. At the same time, it is flexible and allows power users to tweak it to fit their own needs.

The real power behind Jenkins X

The real power behind Jenkins X is:

- The *process*,
- The *selection of tools*,
- And the fact that everything is wrapped up in *one cohesive unit* that is easy to learn and use.



We, *the people working in the software industry*, tend to reinvent the wheel all the time. We spend countless hours trying to figure out how to develop our applications faster and how to have a local environment that is as close to production as possible. We dedicate time searching for tools that will allow us to package and deploy our applications more efficiently. We design the steps that form a continuous delivery pipeline. We write scripts that automate repetitive tasks. And yet, we cannot escape the feeling that we are likely reinventing things that were already done by others.

> **Jenkins X** is designed to help us with those *decisions*, and it helps us pick the right tools for a job. It is a collection of the industry's best practices. In some cases, Jenkins X is the one defining those practices, while in others, it helps us in adopting those made by others.

If we are about to start working on a new project, Jenkins X will create the structure and the required files. If we need a Kubernetes cluster with all the tools selected, installed, and configured, Jenkins X will do that. If we need to create Git repositories, set webhooks, and create continuous delivery pipelines, all we need to do is execute a single `jx` command. The list of what Jenkins X does is vast, and it grows every day.

I won't go into detail on everything Jenkins X does. That will come later. For now, I hope I got your attention. The critical thing to note is that you need to clear your mind from any Jenkins experience you might already have. Sure, Jenkins is there, but it is only a part of the package. Jenkins X is very different from the *"traditional Jenkins"*. The differences are so massive that the only way for you to embrace it is to forget what you know about Jenkins and start from scratch.

I don't want to overwhelm you from the start. There's a lot of ground to cover, and we'll take one step at a time. For now, we need to install a few things.