Jenkins vs JenkinsX – Are they the Same?

As you explore the world of CI/CD and Jenkins, you might encounter the term "JenkinsX". While they are related, Jenkins and JenkinsX serve different purposes and have distinct features. Let's clarify the difference and dive into what makes JenkinsX unique.

Jenkins: The Versatile CI/CD Orchestrator

Jenkins is an open-source automation server that helps orchestrate the CI/CD process. It's highly flexible and can be configured to handle various tasks in the software development lifecycle.

Think of Jenkins as a skilled chef who can adapt to different cuisines. Whether it's Italian, Chinese, or Indian, Jenkins can adjust its recipes (pipelines) to suit the needs of your software project.

JenkinsX: GitOps for Kubernetes

JenkinsX, on the other hand, is an opinionated CI/CD platform built on top of Kubernetes. It's designed to simplify and automate the entire CI/CD process for applications deployed on Kubernetes, following the GitOps methodology.

GitOps is an approach that treats the Git repository as the single source of truth for declarative infrastructure and applications. JenkinsX embraces this philosophy by using Git as the driving force for CI/CD pipelines.

If Jenkins is a versatile chef, JenkinsX is a specialized chef focused on the art of GitOps cuisine for the Kubernetes restaurant.

```
+-----------------------+
|                       |
|        Jenkins        |
|                       |
|  +-----------------+  |
|  |    JenkinsX     |  |
|  |    (GitOps)     |  |
|  +-----------------+  |
|                       |
+-----------------------+
           |
           |
+----------+----------+
|                     |
|     Kubernetes      |
|                     |
+---------------------+
```

Key Features of JenkinsX

1. **GitOps Approach**: JenkinsX follows the GitOps principles. It uses Git as the single source of truth for both the application code and the infrastructure configuration.
2. **Environment Promotion**: JenkinsX automates the promotion of applications across different environments (Dev, Staging, Production) based on Git branching and merging.
3. **Preview Environments**: One of the standout features of JenkinsX is its ability to spin up preview environments for each pull request. This allows teams to review and test changes before merging them into the main branch.
4. **Kubernetes Native**: JenkinsX is designed to work seamlessly with Kubernetes. It leverages Kubernetes' features like namespaces, secrets, and service accounts for managing deployments.
5. **Serverless Jenkins**: JenkinsX introduces the concept of Serverless Jenkins. It runs pipelines as separate containers on Kubernetes, providing better isolation and scalability.

Why Start with Jenkins and Explore JenkinsX Later

While JenkinsX offers a streamlined GitOps experience for Kubernetes, it's often recommended to start your CI/CD journey with Jenkins. Here's why:

1. **Learning Curve**: Jenkins has a gentler learning curve compared to JenkinsX. It allows you to understand the fundamentals of CI/CD without diving into the complexities of Kubernetes and GitOps right away.
2. **Flexibility**: Jenkins is highly flexible and can adapt to various software development workflows. This flexibility allows you to gradually shape your CI/CD process based on your team's specific needs.
3. **Extensibility**: Jenkins has a vast ecosystem of plugins that extend its functionality. This allows you to integrate Jenkins with a wide range of tools and services, making it a versatile choice for many projects.
4. **Migration Path**: Starting with Jenkins provides a smooth migration path to JenkinsX if you decide to adopt Kubernetes and GitOps in the future. The knowledge and pipelines you build with Jenkins can be leveraged when transitioning to JenkinsX.

By starting with Jenkins, you can build a strong foundation in CI/CD principles and gradually explore advanced concepts like GitOps and Kubernetes-native pipelines with JenkinsX as your needs evolve.

When to Use Jenkins vs JenkinsX

Use Jenkins if:

- You have a non-Kubernetes based deployment environment.
- You need high flexibility and customization in your CI/CD process.
- You have existing Jenkins pipelines that you want to maintain.
- You are new to CI/CD and want to start with a more gradual learning curve.

Use JenkinsX if:

- You are deploying exclusively to Kubernetes.
- You want to adopt a GitOps approach for your CI/CD from the start.
- You want to leverage features like preview environments and environment promotion.
- You prefer an opinionated, automated setup for your CI/CD on Kubernetes.

In conclusion, while Jenkins and JenkinsX are related, they cater to different needs. Jenkins is a versatile CI/CD orchestrator that provides a solid foundation for building CI/CD pipelines. JenkinsX, on the other hand, is an opinionated GitOps implementation specifically designed for Kubernetes, offering unique features like preview environments and Git-driven pipelines.

By starting your CI/CD journey with Jenkins and gradually exploring JenkinsX as your needs evolve, you can build a strong understanding of CI/CD principles and adapt your processes to the specific requirements of your projects.

In the next module, we'll explore the practical aspects of using Jenkins, starting with the installation and setup of a Jenkins server using Docker.