Getting Started with Jenkin X Upgrades

This lesson discusses the rapid releases of Jenkins X, how often we should apply upgrades and the goal of this chapter.

Release frequency of Jenkins X

Jenkins X is evolving rapidly. We can see that by checking the releases. There's hardly a day without at least one Jenkins X release. When the community is busy, there can be more than ten releases. At the time of this writing, we have ~1700 releases created over ~15 months. That's an average of over 110 releases per month, more than three releases per day. That's very a high release frequency and for very good reasons.

The community behind the project is proliferating, and that means that the rate of pull requests is increasing as well. It's a simple calculation. The more pull requests people make, the more releases we get.

There is another important reason for such a high release frequency. Among other things, Jenkins X promotes continuous delivery, and it would be silly if the community behind it would not adhere to the same principles. So, instead of making a new release every quarter or some other less frequent period, Jenkins X creates a release from every approved pull request.

How often should you be upgrading?

This doesn't mean that you should follow the same frequency. I don't believe that you should upgrade your cluster with every new release since that would mean that you would spend much of your time performing upgrades. Still, I do want you to upgrade often. Once a month might be a reasonable frequency, once a quarter

would be acceptable. Less frequently than that would be a terrible idea. The more time that passes between upgrades, the more changes will be applied to your cluster. Making significant changes is always more dangerous and harder to deal with than applying smaller changes more frequently.

You might be thinking that performing upgrades every month is insane. "It's too risky. It takes too much time", I beg to differ. Anyone working in our industry for more than ten years has experienced what happens when we wait for too long. It's so hard and painful to upgrade after years of "status quo" that many give up and just keep what they have until it loses support, and then they complain even more. The longer we wait, the bigger the chance that something will go terribly wrong . That's one of the reasons why we release our software much more frequently, and we should apply the same logic to third-party applications as well.

> 📝 Applying smaller changes more frequently gives us more control. It allows us to **find issues faster**, and it makes it **easier to fix them**.

The goal of this chapter

My goal in this chapter is not to convince you that you should be upgrading your third-party software frequently. Instead, I must point out that we haven't upgraded our Jenkins X cluster at all. If you reused the same cluster throughout all the chapters you are running an old version of Jenkins X. Even if you did destroy the cluster at the end of each chapter and created a new one for the next, you will start using Jenkins X outside our exercises and that means that you will have to upgrade it sooner or later. So, the time has come to learn how to do just that.

At this moment you might be thinking that upgrading Jenkins X platform is a single command and that it is silly to dedicate a whole chapter to it. Upgrading the platform inside the cluster is only part of the story. The Jenkins X platform is only one of the pieces of the puzzle.

- The cluster might also contain add-ons, apps, and extensions.
- Then there is CLI on our laptop that might need upgrading as well.
- Finally, we might need to upgrade our **Ingress** rules as well.
- If for no other reason, we do need to add **TLS certificates**. No one should expose their applications over plain `HTTP`.

As you can see, there's much more to upgrading than at first glance. But, before we dive into it, it might be beneficial to understand Jenkins X version stream.