XL Release Integration and Jenkins

Three ways of using XL Release and Jenkins for continuous delivery for flexibility, scalability and Improvements



XL Release First: 

In this setup, a release/pipeline run is [started via the XL Release UI](https://docs.xebialabs.com/xl-release/concept/release-life-cycle.html), or automatically from e.g. an [ALM](https://en.wikipedia.org/wiki/Application_lifecycle_management) tool such as [JIRA](https://www.atlassian.com/software/jira), [Rally](http://www.rallydev.com/platform-products/deliver-features-users-love-faster), [RTC](https://en.wikipedia.org/wiki/IBM_Rational_Team_Concert) or [VersionOne](http://www.versionone.com/product/agile-project-management-tool-overview/) via the HTTP API. The first phase of the pipeline usually invokes Jenkins, to build the release artifact(s), verify that a previous build was successful, or was promoted etc

**Triggers:** process-related and/or “semi-regular”  
**Interaction:** XL Release calls Jenkins once or multiple times throughout the release/pipeline run. Jenkins jobs generally do not call downstream jobs.

Jenkins First

**Triggers:** code-related and/or regular  
**Interaction:** Jenkins initially creates and starts the release/pipeline run via the Jenkins XL Release plugin. XL Release then often calls Jenkins once or multiple times throughout the release/pipeline run. Jenkins jobs generally do not call downstream jobs.

## Hybrid Pipeline

**Triggers:**

* Jenkins section: code-related
* XL Release section: from Jenkins, manual or from process tool

**Interaction:** Jenkins runs the initial part of the pipeline, typically until QA/UAT and sometimes automatically triggers XL Release on completion. As the last mile to Production in XL Release is progressively automated, XL Release often ends up calling Jenkins to retrieve information (fingerprints, test results, commit IDs etc.) about the initial pipeline stage.

In the “Hybrid Pipeline” scenario, the release/pipeline run in XL Release is either triggered at the end of the Jenkins pipeline via the Jenkins XL Release plugin, manually or from a process tool such as an ALM or [Change Management](https://en.wikipedia.org/wiki/Change_management_%28ITSM%29) system – it really depends on the application.

E2E Orchestration:

XL Release has Task – which can be manual and Automated ones.

XL Release has many plug-ins which helps us to integrate XL Release to talk to different tools in our DevOps CICD platform.

For an example – XL Release – Jenkins Integration we can create a task in XL release as a step to trigger build job automatically which in-turn trigger the entire pipeline.