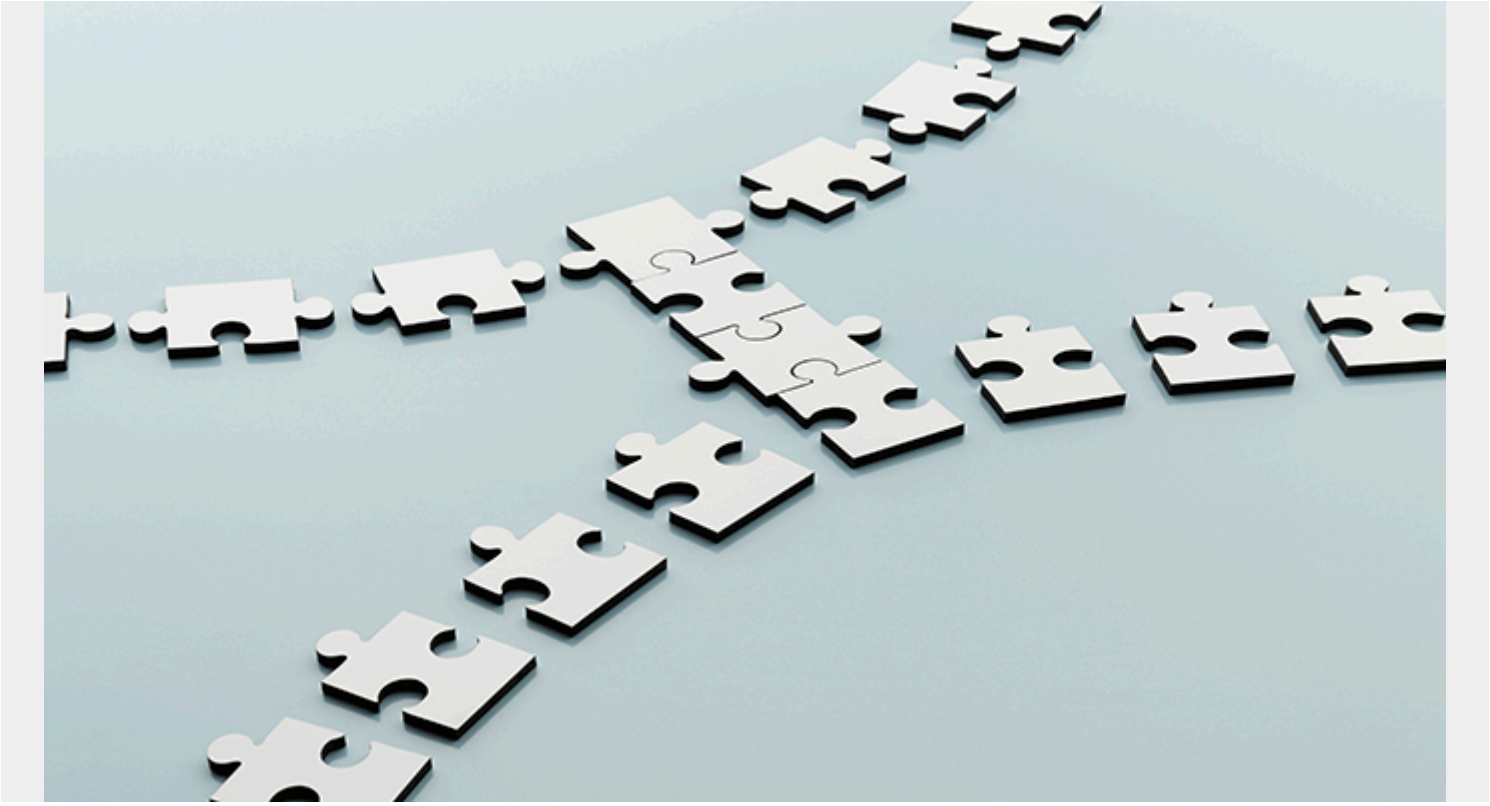
MODERN BATCH: THE OPS IN DEVOPS FULLY EVOLVED – "JOBS-AS-CODE!"



My previous installment in this series discussed the challenges of managing batch for modern technology with the added complexity of meeting your company's digital transformation needs.

My hypothesis for how to address these challenges seems exceedingly simple; manage your application "instrumentation" the same way you manage your application! Simple, elegant, "mic-drop" time.

But what does that really mean?

For modern application workflow orchestration, and arguably for any management discipline, it means taking an "as-code" approach with a solution that provides the capabilities you need to support your business objectives. Jobs-as-Code means the artifacts that define jobs can be built using a familiar, code-like notation, stored in an SCM together with the code that implements business logic, built together with that code, tested together, promoted from environment to environment together, and eventually deployed together; with the same level of automation.

I want to highlight two of the characteristics that enable "as-code" and CI/CD to deliver the seemingly impossible gains in speed and quality. The first is usage of a testing framework that can subject the new or updated code to exhaustive testing use cases. Because of virtualization and cloud technology, it's feasible to build a testing environment that is very similar to the target production one. Doing this in stages as the code progresses along the delivery pipeline results in hardening that hopefully makes the eventual push to production a non-event. The second is

automation. Building test environments similar to or even more complex than production is feasible only when done automatically. Running through dozens or hundreds of test cases is only feasible in a fully automated pipeline. And finally, pushing code through a full delivery pipeline, even if just a single line of code is changed is also possible only when that pipeline is fully automated.

So, when choosing your tooling, make sure that Jobs-as-Code approach capabilities like the ones below are included:

- Sophisticated flow relationships
- Extensive application integration to support all your platforms and technologies
- Operational insight into execution status and progress
- Output and log collection
- Support for service-level management, business-level abstraction and full security
- Audit and governance compliance
- Comprehensive end-to-end support of a fully automated release/delivery pipeline

With such an approach, your organization can focus on producing innovative business services rather than divert precious developer talent to building operational plumbing which is unlikely to reach the level of sophistication obtained with a market-tested solution. Using a solution designed specifically for the task at hand can help you reap the benefits of a truly automated and continuous delivery pipeline to accelerate delivery of new services, and you get the manageability and operational oversight that facilitates operating those new services at the highest levels of availability and reliability.

This is the fourth of a 4-part blog series on “modern batch.” You can read the other blogs here:

- Part 1: Modern Batch Processing: A Thing of the Past or Essential Discipline?
- Part 2: Modern Batch: To Batch or Not to Batch? There’s no Question!
- Part 3: Modern Batch: Managing the Madness?