GIT FOR THE MAINFRAME: TAKE MAINFRAME DEVOPS TO THE NEXT LEVEL

The history of the mainframe is a long one, and it's far from over. While today's computing industry is largely focused on cloud- and web-based apps, the mainframe has continued to act as both the backbone and muscle. The platform has continuously improved and modernized its development and testing processes to equal or surpass those on the distributed side, and now it's doing it again in a big way.

While the mainframe has been reliably supporting data processes in the background, the developers and specialists on the distributed side have increasingly turned to Git as their environment of choice. Now, mainframe development organizations are choosing Git, which is good news for developers and customers alike.

Welcome to the world of Git

Git has already proven itself as practical and worthwhile for version control and colocation of assets; it's also extremely popular—almost universally adopted—and is also well-known among the latest generation of IT specialists. Mainframe applications have also found a suitable place within this dynamic Git environment, making the next transformation of the mainframe straightforward and extremely practical.

In addition to being an optimal repository for source code management (SCM), Git offers the mainframe culture the opportunity to emerge from "behind a wall," better reflecting its position as a central driver of modern industry, joining in with other IT players including cloud, mobile, web, and back-end Java, which all have Git in common.

Git is not a one-stop solution, of course. Out of the three distinct "pillars"—SCM, build management, and deployment management, Git only serves the first. But this is where further good news happens. BMC AMI DevX Code Pipeline has long been a key asset in the mainframe software development lifecycle. As a software configuration management solution, it is central to continuous integration and continuous delivery (CI/CD). Instead of leaving people out in the cold when completing the remainder of the development lifecycle, ISPW coordinates the other two pillars with a simple right-click of the mouse. This pairing of BMC AMI DevX Code Pipeline addresses the very real priority of moving to Git while also completing and protecting the existing development lifecycle.

One size does not fit all

We find this to be doubly ideal, since it not only capitalizes on the respective strengths of both solutions, it also helps address the varying needs of customers where one size does not fit all. For example, in some cases, you might have a development team that understands Git, already knows what it is going to do with the application code, and has a software delivery lifecycle to deploy to production. For this group, Git is an ideal option.

However, there may also be situations where a team monitors a batch application in maintenance mode. This team will not want to check out a Git repository of two-and-a-half million components to make a single-line JCL change. It would not make sense to have this in Git. Instead, they can choose from a range of options to ensure they can work at their own pace.

A best of both worlds scenario

Using Git with ISPW does not mean replacing the latter. Instead, Git works in conjunction with ISPW to provide a complete solution, not only for the build and deploy, but also for managing secure synchronization with the system of record. Although Git is holding all the library books, there is still a need for an engine to publish them and make them available through build and deploy.

The cultural shift

As with many significant technological developments, the humans who work with them will need to acclimate to this new way of doing things. Although Git is a better way—a modernization of the software delivery lifecycle (SDLC) for the mainframe—the various people involved, such as developers, development managers, security specialists, and legal professionals, all need to understand how this new approach will work and what it will mean for their specific areas of work, their organization, and their customers.

Once all parties recognize that the solution pairing covers the whole process, the change involves reliable Git and ISPW components that are simply dovetailed together, and the transition process is customizable and manageable, it will be easier for them to accept. The easy customization and management are also important to note, especially if any of the groups' first exposure to this process was the "wholesale shift" approach put forward by others. No one likes to leap blindfolded off a cliff.

A complete solution for choice and flexibility in Git implementation

As organizations look for the benefits of mainframe DevOps, our new Git integration with BMC AMI

DevX Code Pipeline gives mainframe development teams the flexibility to fully adopt Git or use the ISPW feature-branching sandbox for a controlled and isolated environment to create and change code. With a simple right-click, Git users take full advantage of the ISPW mainframe build, test, and deploy capabilities in their CI/CD pipeline.

Each team can choose to go 100 percent Git or have the option to use ISPW with feature branching for SCM. Both options provide an agile and flexible mainframe developer experience, giving organizations the ability to use Git workflow and the flexibility to choose the SCM tool that works best for their development teams while leveraging the best of ISPW for mainframe build, test and deploy.

Moving mainframe DevOps to the next level

This is a new era, and mainframe technologies remain as dependable and battle-ready as ever. Pairing Git with BMC AMI DevX Code Pipeline opens new doors for more reliable and practical development activities while also securing the future of data with versatile techniques that are immediately familiar to the next generation of mainframe pros.

To explore these concepts in greater detail, download our eBook, *Git for the Mainframe*.