OpsMx

The New Digital Transformation and DevOps Paradigm:

Add Continuous Delivery to Jenkins and Break the Scripting Curse

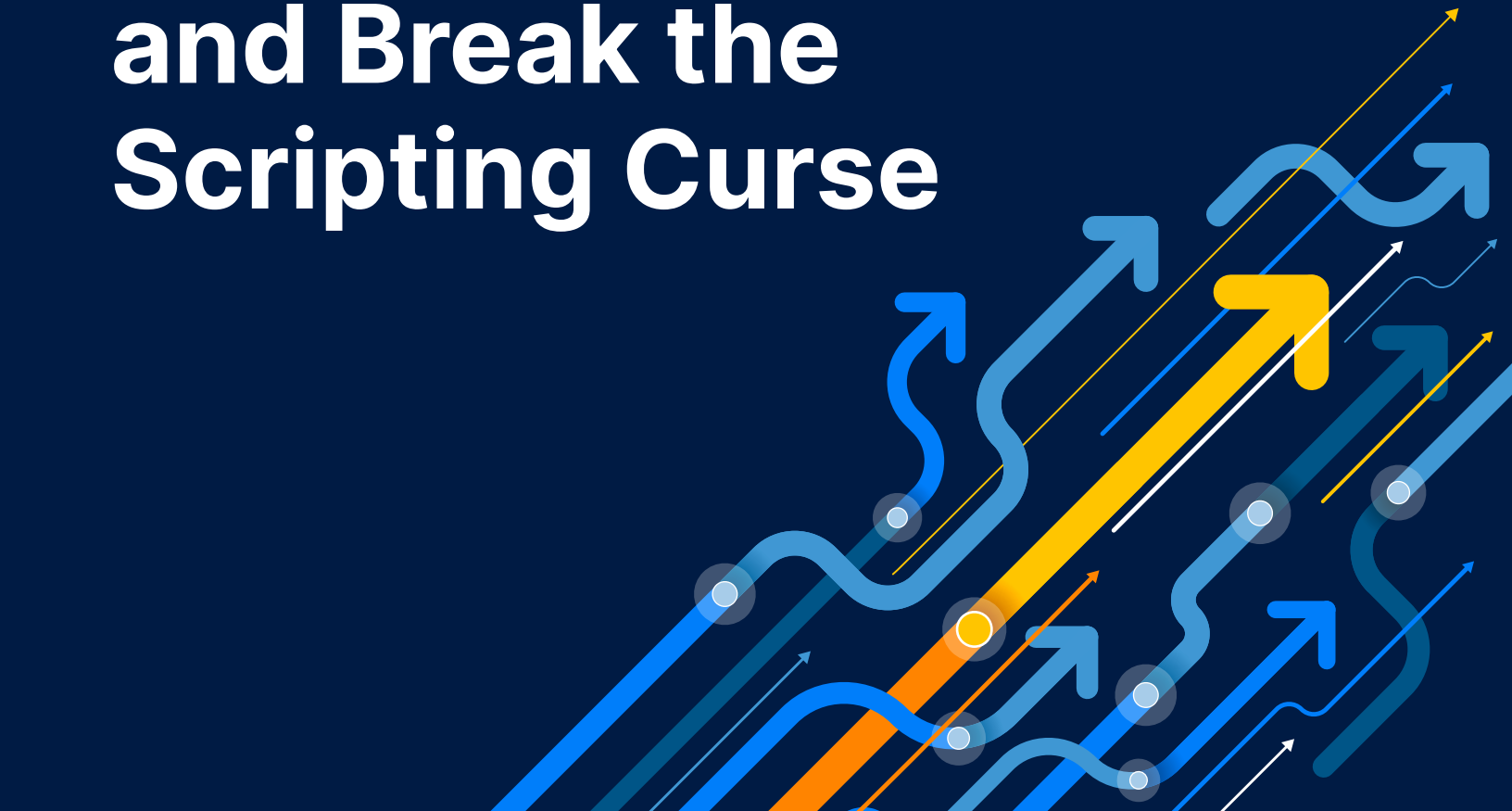

Table of Contents

Why DevOps is Key to Your Digital TransformationJourney

Modern organizations undergoing digital transformations re-imagine the way they interact with their consumers and strive to surpass their expectations. Often, a key element of digital transformation is a paradigm shift in which organizations migrate from legacy apps and adopt modern technologies such as the Cloud, Kubernetes, AI, analytics, and automation to cater to their customers in exciting new ways.

Before embracing new technologies, organizations must adopt a culture of innovation and experimentation in their software delivery process. The DevOps approach becomes crucial as organizations rapidly build new services, quickly deliver those services to the market, and identify what is delivering the most value to their customers.

The DevOps journey differs from one organization to another. Some mature organizations are experimenting with new paradigms. Others are less mature and are still developing their DevOps strategy. The journey involves collaboration between various departments such as security, governance, application, platform, and SREs so it can be difficult to adopt DevOps successfully. To adopt a DevOps approach, the whole team needs the right set of tool chains.

Below are some examples of tool chain:

Source Code Management	Build	Artifacts Repository	Deployment	Verification	Monitoring and Logging	Infrastructure Management
GitHub, GitLab	Jenkinks, Circle CI, Bitbucket	Docker HUB, JFrog Artifactory	Spinnaker, Argo CD, OpsMx ISD	OpsMx Autopilot	Appdynamics, Dynatrace, New Relic	Terraform, Ansible, Puppet

Use DORA Metrics to Track DevOps Maturity

The DevOps transformation begins by focusing on modernizing your software delivery process to deploy and deliver software in a fast, safe, reliable, and repeatable way. And in order to meet those requirements, the applications, operations, and DevOps teams need to constantly improve. DevOps Research and Assessment (DORA) metrics identify KPIs as well as benchmarks to gauge the performance of software delivery teams and drive product improvement.

Organizations must monitor and improve outcome-based DORA metrics:

- Deployment frequency: How often the code is deployed to production
- Lead time for changes: How long does it take to go from code-committed to code-running in production
- Change failure rate: What percentage of changes release to production cause degraded service and subsequently require remediation
- Time to restore service: How long does it take to restore the service in the event of an incident

Therefore, based on the value for each measurement, companies can assess their maturity.(refer to the below image)

	High IT Performer	Medium IT Performer	Low IT Performer
DEPLOYMENTFREQUENTLY	On demand (multiple deploys per day)	Between once per week and once per month	Between once per month and once every six month
LEAD TIME FOR CHANGES[b]	Less than one hour	Between one week and one per month	Between one month and six months
MEAN TIME TO RESTORE(MTTR)	Less than one hour	Less than one day	Less than one day[c]
CHANGE FAIL RATE	0-15%	31-45%	16-30%

Figure A- DORA Metrics (Deployment frequency, Lead time to change, Mean time to restore, and Change failure rate) to track DevOps maturity

Why Jenkins Pipeline is Insufficient to Attain High Software Delivery Performance

Jenkins is the most popular continuous integration (CI) tool available, and therefore, it is widely used by almost all software developers across the world. It is so dear to the application and platform development team that they even extend Jenkins for deployment purposes. After working in Jenkins for years, developers gain the confidence to set it up for deployment within hours. However, the problem starts when they try to scale it and use it for production deployments.

Often, developers face some common challenges while using Jenkins for deployment. Some of them are plugin nightmares, dependence on scripts to create repeatable pipelines, limited deployment visibility, and limited knowledge on the impact of a new release.

Plugin Nightmares

There are a multitude of plugins available in the Jenkins plugin site based on the application that has to be deployed (docker-based or Kubernetes-based, etc.) into multi-cloud or hybrid. So, developers can activate these plugins whenever needed and dismiss them when not required. This not only saves resources, but also helps the application deployment process and is versatile. Additionally, plugins also enhance the core Jenkins functionality. As such, plugins are only meant to extend features beyond the core function of a product. However, if businesses try to add core functions with the help of plugins, they might obstruct the entire system. In reality, developers often use multiple plugins to accomplish daily deployment tasks. For example, pulling code from GitHub requires a plugin. Deploying an application into Kubernetes, or AWS ECS requires developers to download a separate plugin. Unfortunately, it is not easy to maintain a lot of plugins on a daily basis especially, when DevOps teams have to manage 100s of pipelines while deploying multiple micro services per day. Also, developers need to address these plug-in dependencies as well.

Dependence on Scripts to Create Repeatable Pipelines

Previously, while creating deployment workflows, stages, and deployment strategies, developers needed to write multiple scripts and also maintain those scripts. So, manual coding of environment variables, secrets, dependencies, and release strategies such as canary and blue-green were ideal as companies would deploy stateless monolith applications once in two months only. However, this is not the case anymore. Deployments are much more fast-paced with public cloud and containers. This means developers have to spend more time on writing scripts for deployment rather than focusing on deployment, which is unproductive.

Limited Visibility into Deployment Status

Jenkins provides limited visibility into the build stage only. As a result, the DevOps team does not get real-time insights whether the deployment is successful or not, and whether the newly created pods are healthy or not, and so on. This in turn, limits the visibility of managers and other stakeholders into the deployment pipeline. For instance, managers may not be able to tell who deployed what and when. Hence, without proper insights into various pipeline executions, IT organizations cannot collaborate and improvise their CI/CD process.

Limited Knowledge of the Impact of a New Release

As such Jenkins does not allow developers to identify the risk of a release prior to deployment. So, without proper automated gates, engineering managers become gatekeepers who must go through the software manually to verify associated risks before allowing a Jenkins deployment pipeline. This is a labor-intensive, error-prone process and is also not scalable. Moreover, having a limited understanding of the impact of a release or health status of new deployments increases the risk of production downtime.

Extend Jenkins using OpsMx ISD

Understandably, Jenkins is the Butler in the continuous integration space. But in case of continuous deployment, the use of Jenkins can surely result in “technical debt”. While Jenkins is not the right CD tool, it can be easily integrated with CD tools which developers can leverage to achieve repeatable deployments and move ahead in their DevOps journey.

OpsMx Intelligent Software Delivery Platform (ISD) is a modern continuous delivery platform that increases the speed of delivery safely and securely, eliminating the need for human intervention. ISD comprises two modules, the continuous delivery module and the delivery intelligence module.

Continuous Delivery module The Continuous Delivery module is used to simplify orchestrating the end-to-end process workflow from code check-in to safe multi-cloud deployments

Delivery Intelligence module The Delivery Intelligence module provides insightful data-driven risk verification, policy enforcement, and approvals to ensure quality, risk-free and compliant software in production.

OpsMx ISD provides pre-built integration with Jenkins and with the artifact repositories. Whenever a build process is complete and an artifact is sent to an artifact repository such as JFrog or Docker Hub, ISD will automatically fetch and deploy into relevant targets. Figure B shows how OpsMx ISD integrates with Jenkins to automate the delivery and deployment process.

As of today, ISD can easily deploy into Kubernetes, AWS, Azure, Google Cloud, OpenShift, and Oracle Cloud, without the need to download and maintain plug-ins. Moreover, with ISD, developers do not need to be concerned with the intricacies of the cloud environment.

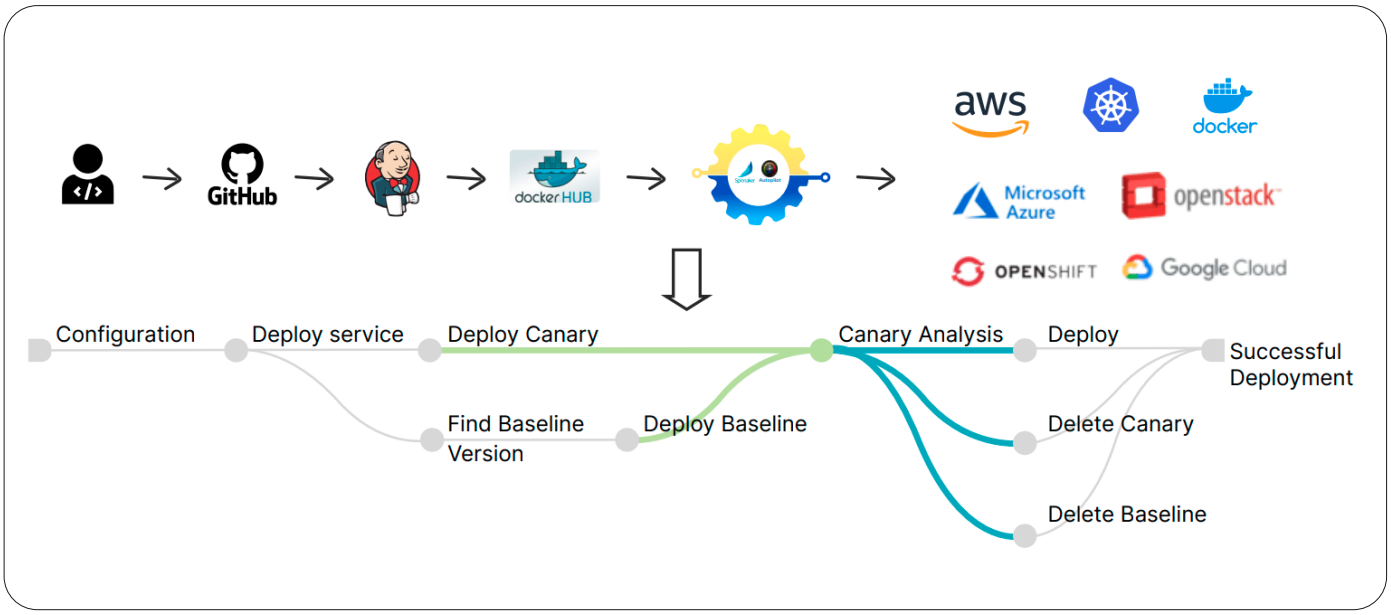

Figure B- OpsMx ISD integrating with Jenkins to automate software deployment and delivery

Automate Pipelines for Deploying Rapidly and Reliably Without any Scripting

OpsMx ISD can help DevOps engineers to deploy their monoliths and microservices into multi-cloud and hybrid cloud on-demand. ISD offers pipeline-as-code, enabling developers to re-use pipelines to deploy their code into multiple environments. One of the significant benefits of using ISD is that it allows organizations to scale their software delivery as needed, thereby deploying an almost unlimited number of changes to multiple targets per day without any scripting. Figure C represents a deployment pipeline in ISD with various stages to integrate with other tools and make the delivery seamless. Thus ISD is not only a highly scalable tool that is suitable for increasing deployment workloads, but it is also easily extensible in supporting integrations with over 50 of the most common DevOps tools.

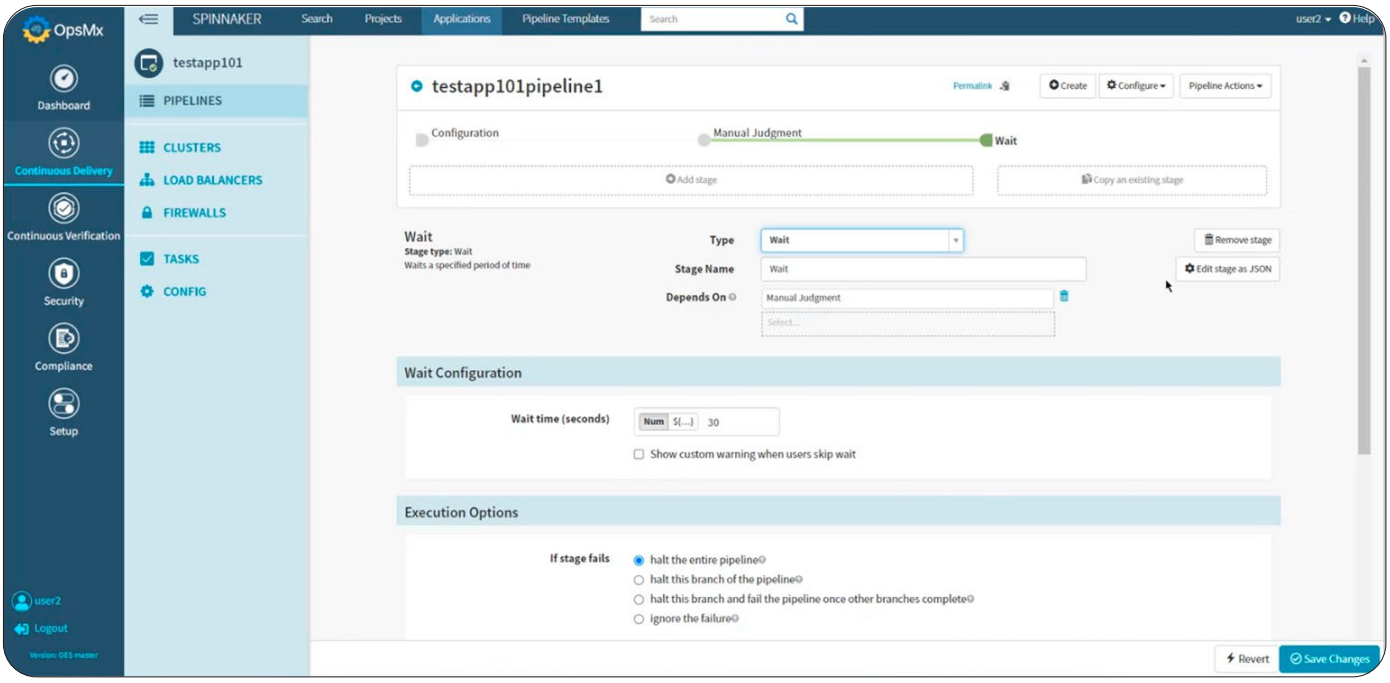

Figure C- OpsMx ISD pipelines for automating deployments

Improve Visibility on Deployments for all Stakeholders

OpsMx ISD helps identify and remove bottlenecks in your deployment pipelines through application and delivery dashboards. It enables developers to monitor applications during and after deployment to production and empowers them to easily detect issues and request rollback quickly, thereby avoiding disruptions. Figure D provides a high-level view of pipeline execution and their success and failure status over time. Project managers can now get a holistic idea of the pipeline performance and take adequate steps to optimize their continuous delivery initiative. Lastly, ISD provides audit reports to easily investigate pipeline failures and policy violations and helps trace who deployed what software and when.

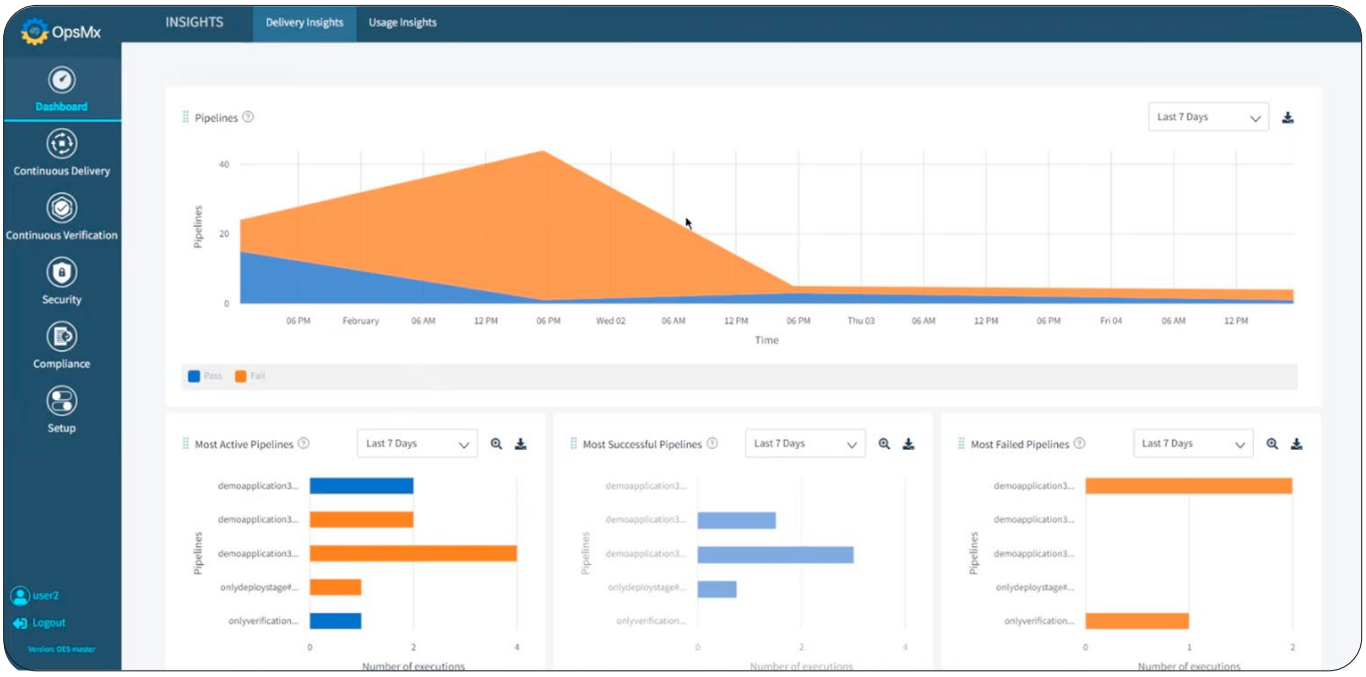

Figure D- Deployment dashboards and insights

Verify Risk of a New Release and Auto-Rollback

ISD helps software developers improve customer satisfaction by minimizing errors during the build, deploy, test, and release processes. ISD's Delivery Intelligence module gathers data from various CI/CD tools, APM, and log analyzers. It applies supervised and unsupervised Machine Learning techniques that enable the team to scan and report software risks throughout the software delivery pipeline. Figure E represents the estimated risk score by ISD during a canary analysis of a newly deployed software into production. Based on the score, SREs can roll forward or roll back a software. The correlation of data, metrics, and logs in the pipeline context provides best-in-class diagnostic capabilities.

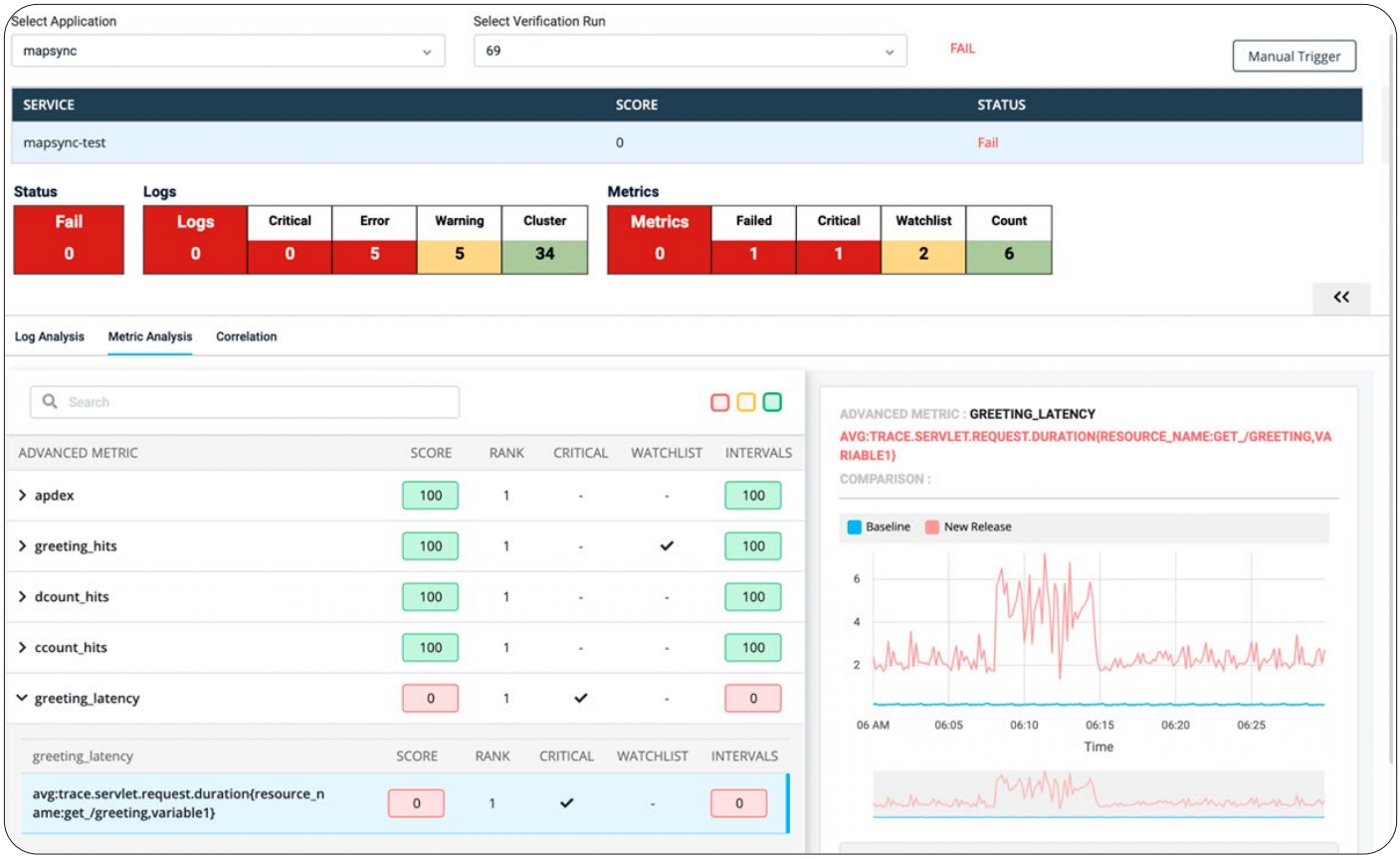

Figure E- Risk assessment of a software by analyzing logs and metrics

Mitigate Security Risks by Defining and Enforcing Policy Gates

OpsMx ISD allows DevSecOps to define security policies in the software delivery pipeline. As a result, the security and compliance team can ensure your DevOps process complies with the organization's governance and rules while shipping your code, upgrades, and application to production. Most importantly, compliance managers can create policies to check various software release parameters and deployment conditions before/during the execution of the delivery pipeline. Figure F highlights how compliance managers can use ISD to add policy gates in a pipeline before the product deployment.

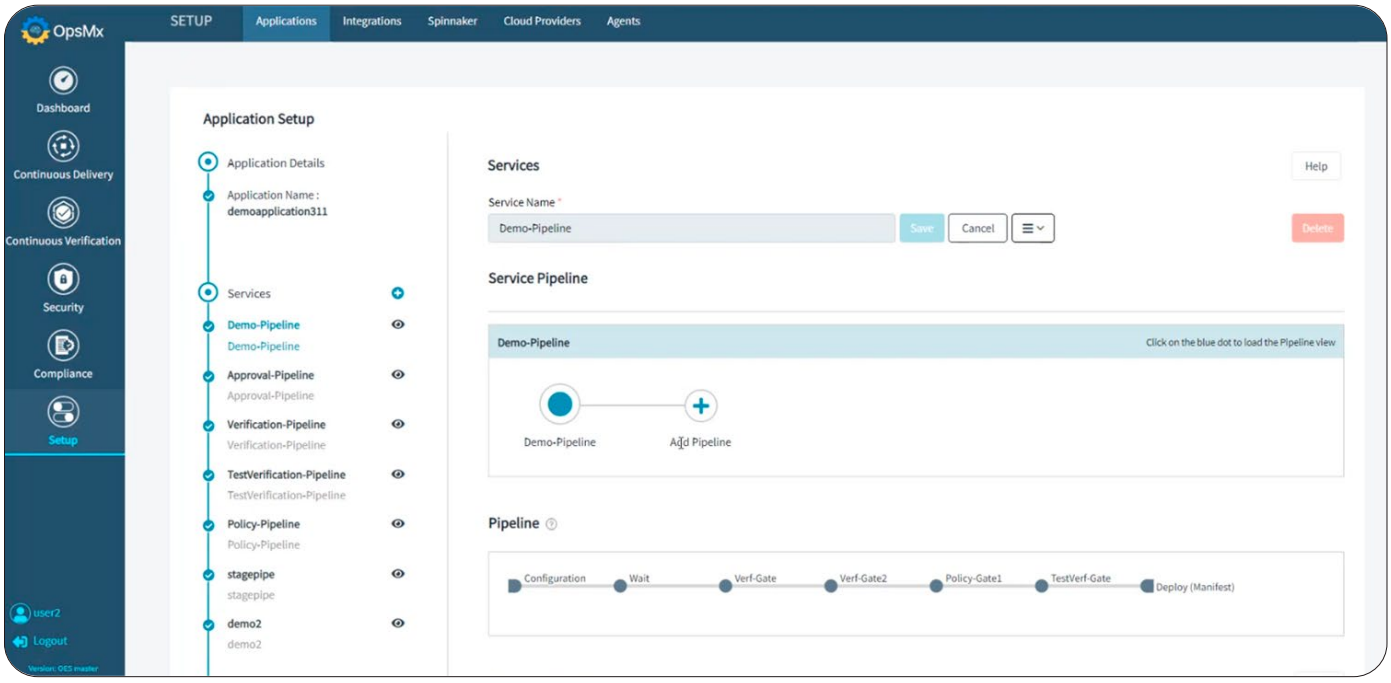

Figure F- Enforcement of security and policy gates into automate pipelines

Improve Collaboration and Aid Faster Decision-Making with Approval Gates

ISD allows project managers to configure approval gates easily, choose data sources, and define them in software delivery pipelines within seconds. ISD can fetch data from DevOps toolchains such as Git, Jenkins, SonarQube, Jemter, AppScanner, ServiceNow, and many more and provide consolidated information regarding software release. With the help of relevant and timely insights, project managers can quickly and safely make informed decisions on promoting the software across your pipeline from Dev to Testing to the Production stage. Figure G represents the ISD application dashboard for product managers to highlight pending approvals, policy violations, and verification failures for faster collaboration in the team.

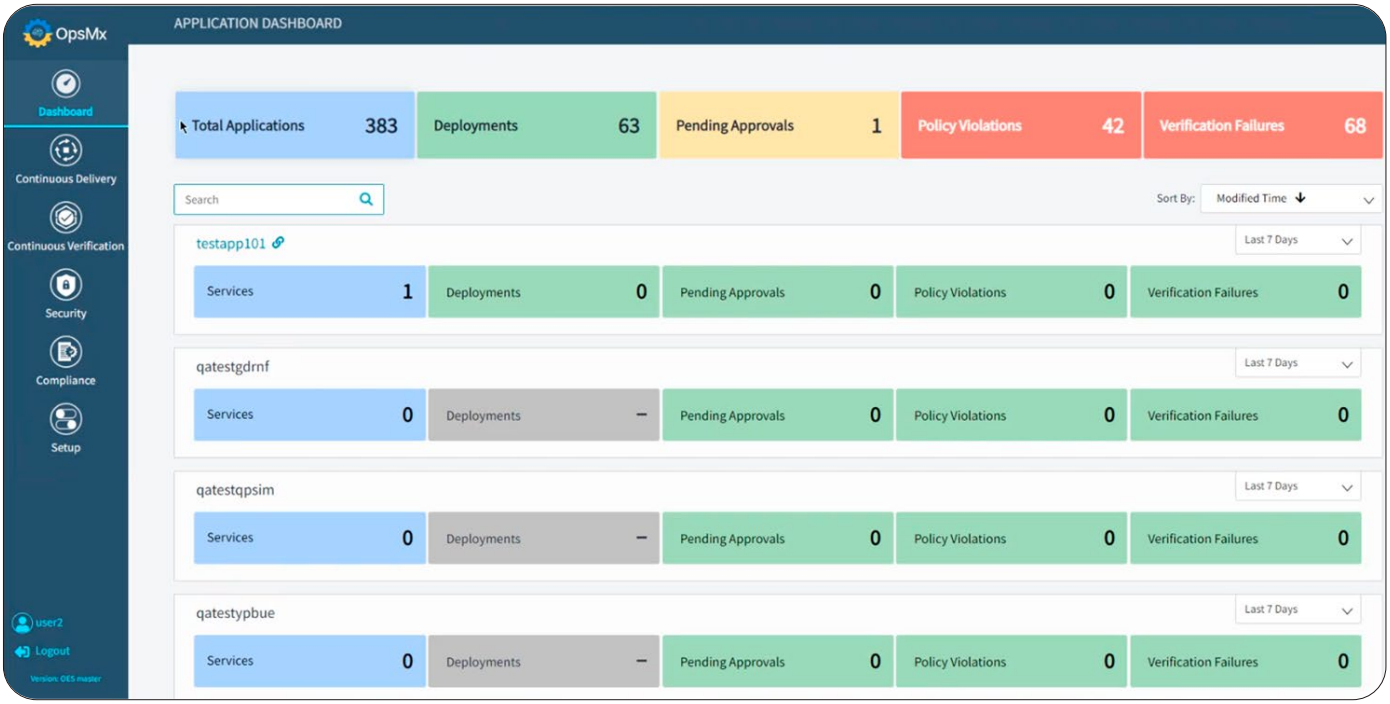

Figure G- Application dashboard representing pending approvals

Use Cases of ISD

Symphony addressed the bottlenecks in its CI value stream and modernized software delivery successfully

Symphony provides a collaboration platform for financial institutions that are highly regulated and render services to millions of customers. Complying to these rigorous standards meant that the company had to maintain the utmost level of security and regulatory policies.

So, they were looking to deploy a major update to their platform that would enable a multi-tenant system. However, their CI value-stream based on Jenkins created certain bottlenecks which prevented them from achieving maximum performance.

Symphony leveraged OpsMx ISD to securely modernize their software delivery within a span of 3 months. By integrating with Jenkins, ISD helped Symphony's DevOps team to deploy software and infrastructure changes seamlessly into AWS using automated pipelines.

Results

1) 10X Growth in deployment frequency, up to 50 updates per day.

2) 98% reduction in time to onboard customers (upto 10 minutes).

3) Reduced cost through infrastructure deployment independent of SREs.

Cisco deploys multiple changes into cloud perday with OpsMx ISD

Cisco is a world leader in the networking space. They decided to accelerate their movement to the cloud and undertook a project to deploy Kubernetes on-prem and in the cloud (in both AWS and GCP).

In order to improve their competitiveness, Cisco needed a way to update their software more quickly. So, they decided to replace their legacy deployment solution and implement Jenkins CI. However, soon, their IT team realized that it was not possible to scale their multi-cloud deployments into hybrid cloud with Jenkins. That is when the Cisco IT team adopted OpsMx ISD. Today, more than 2000 developers at Cisco use thousands of ISD pipelines to deploy changes daily.

Results

1) Deployment time reduced from days to minutes.

2) 100,000 of updates delivered into production with a few errors only.

3) Deployment frequency increased from monthly to on-demand.

Conclusion

Delivering new features within a short timespan into the market can be a key deciding factor for your end customers. Achieving this kind of business velocity means that you are hugely dependent on your DevOps team. Empowering your team can foster innovation and ship code faster to production. This, however, requires you to get out of the extending-Jenkins mindset and adopt a continuous delivery (CD) tool. A robust CD tool can prevent the CI value-stream from becoming a bottleneck while completing transformation and migration to cloud-native and containerized apps. Most importantly, adopting the right CD tool also aids you to achieve digital transformation at scale and get an edge over your competitors.

Author

Gopinath Rebala

CTO of OpsMx

Gopinath Rebala is the CTO of OpsMx, where he heads the machine learning and data processing architectures of OpsMx Enterprise for Spinnaker. Gopi also has a strong connection with our customers, leading design and architecture for strategic implementations. Gopi is a frequent speaker and well known leader in continuous delivery and in the Spinnaker community.

Previously, Gopi was a co-founder and CTO at N42, which delivered machine learning tools for improving reliability for large scale distributed systems.

Ashley Owen

Head of Product Marketing at OpsMx

Ashley heads the product marketing team at OpsMx.

He is an award-winning global product development strategist and influencer with more than 20 years of proven ability to recognize and capitalize on market opportunities and trends, impact bottom-line through strategic planning, trend analysis and forecasting, drive product roadmap and delivery, and advance sales execution and customer adoption.

OpsMx

About OpsMx

Founded with the vision of “delivering software without human intervention,” OpsMx enables customers to transform and automate their software delivery process. OpsMx’s intelligent software delivery platform is an AI/ML-powered software delivery and verification platform that enables enterprises to accelerate their software delivery, reduce risk, decrease cost, and minimize manual effort. Follow us on Twitter @Ops_Mx and learn more at www.opsmx.com