



Jonathan Fontana | IT Manager DevOps et Performance

COMMENT LE PMU INTEGRE LE TEST DE PERFORMANCE DANS SA PLATEFORME AVEC NEOLOAD

Contexte

Le PMU est le leader européen du pari hippique et le troisième acteur mondial. C'est une société dont le chiffre d'affaires repose très largement sur le digital, que ce soit pour les paris en ligne ou pour les paris traditionnels qui sont enregistrés et centralisés grâce à un réseau et des applications développées en interne.

La performance de ces applications est essentielle pour assurer une activité sans interruption 24/24 mais aussi pour garantir la performance et la fiabilité dont a besoin le PMU lors des prises de paris qui se font sur des moments extrêmement courts mais répétés dans une journée.

Pour répondre aux attentes des clients et proposer toujours de nouveaux usages le PMU a multiplié par trois la fréquence de release, en passant d'un rythme trimestriel à un rythme mensuel. Et les équipes ont réussi à intégrer le test de performance dans leur processus Agile d'intégration continue et à améliorer la gestion de la performance applicative au sein des équipes DevOps. Ce résultat est possible grâce aux outils mis en place, aux nouveaux processus et à l'implication des équipes.

Evolution Agile et test de performance

Traditionnellement le PMU développait des applications sur la base de cycles en V avec des équipes qui travaillaient en silo. D'un côté les développeurs (Dev) et de l'autre les opérations (Ops) responsables des tests de performance en pré-production.

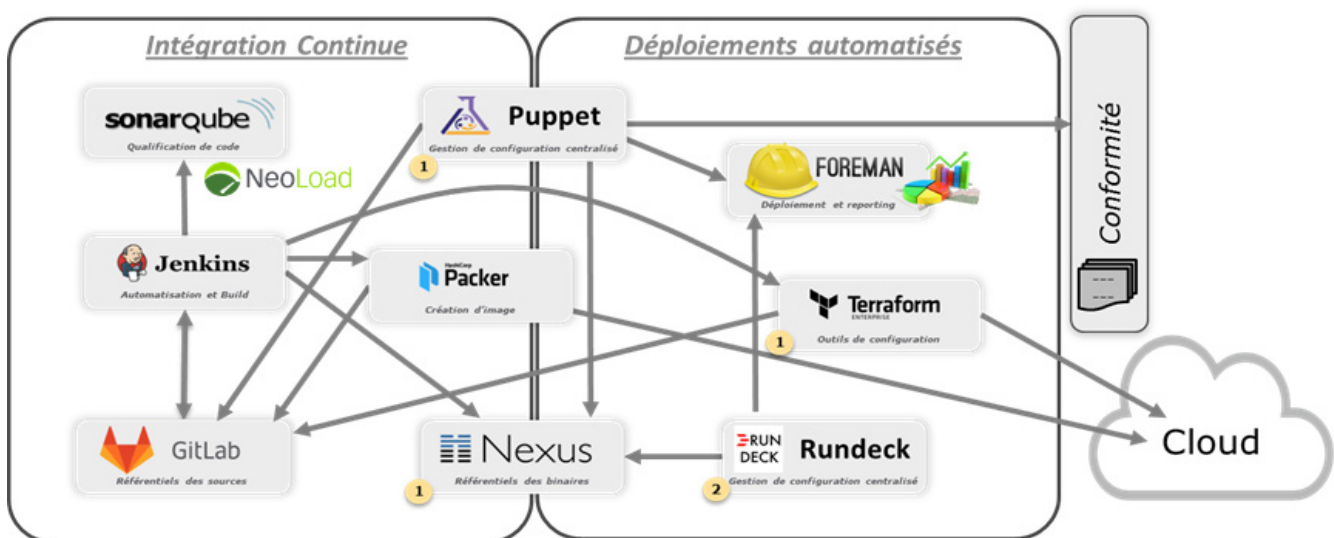
L'absence de collaboration entre Dev et Ops créait des frictions dans le processus de développement. Par exemple, les testeurs de performance découvraient les applications à tester au moment où ils les recevaient des Dev, sans aucun contexte. Cette situation dans laquelle le testeur ne sait pas ce qu'il doit mesurer, ni ce qu'il doit tester génère une quantité très importante d'aller/retour entre Dev et testeurs qui allonge considérablement la phase de validation de l'application. Résultat : certaines applications ne sortaient qu'après un an de développement.

Une plateforme DevOps au PMU pour briser les silos

Pour répondre au besoin de vélocité des cycles de release, le PMU a mis en place une organisation DevOps supportée par une plateforme d'outils adaptée. NeoLoad y assure la fonction de test de charge et de performance, avec une volonté d'automatisation maximum des tâches de test pour réduire les cycles de release.

Le terme de plateforme DevOps est important car il montre bien la volonté de supporter un processus collaboratif dans lequel chaque intervenant peut contribuer et partager l'information avec le reste de l'équipe. Grâce à cette plateforme, les anciens silos ont disparu. Lorsqu'un développeur produit du code, il le pousse dans GIT, un job Jenkins lance une analyse Sonar, et, si elle est validée, un package sera créé et livré dans Nexus. En même temps, ce package est installé automatiquement sur les différents environnements de test. Les équipes Ops voient arriver le code dans les outils qu'ils utilisent, dans un mode collaboratif.

Ce travail collaboratif, rendu notamment possible grâce au plugin NeoLoad pour Jenkins, permet de retrouver tous ses scénarios de tests dans Jenkins. La possibilité est offerte de lancer les tests soi-même ou de donner la main à un autre utilisateur pour le faire, Jenkins étant connecté au LDAP pour gérer les droits. Un historique est disponible permettant de savoir si les tirs ont été lancés par un humain, ou, de manière automatique. Le PMU a également la possibilité de rendre plusieurs scénarios utilisables par différents collaborateurs. Cela fonctionne comme un catalogue de service de test de performance complètement intégré à leurs outils d'intégration continue. L'organisation peut ainsi tracer l'évolution de la performance, vérifier quels sont les builds qui ont raté les tests, pour quelles raisons et savoir comment y remédier. Enfin, le reporting peut être envoyé, grâce au plugin de mail Jenkins, aux collaborateurs concernés.



Avec sa plateforme DevOps, le PMU est aujourd'hui capable de tester en phase de recette.

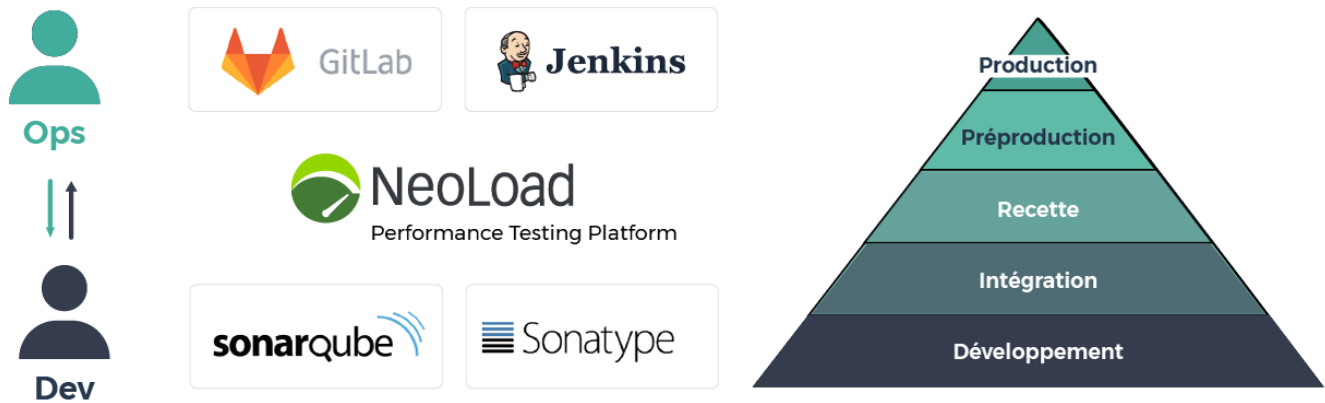
Les scénarios sont évolutifs mais faits en amont, ce qui permet d'éviter d'aller jusqu'en pré-production et de revenir en arrière. Les cycles de développement Dev et recette sont donc beaucoup plus rapides et plus itératifs.

L'automatisation : levier d'accélération du time-to-market

L'un des principaux avantages des tests automatisés est une mise sur le marché plus rapide pour les nouvelles fonctionnalités et améliorations.

Les jobs Jenkins vont mettre à jour automatiquement les tests dans NeoLoad. La maintenance de la plateforme se fait aussi de manière automatisée grâce à Rundeck, ce qui permet de réaliser des tirs de nuit, mais aussi le weekend. Ce temps gagné à tester la nuit et les week-ends a permis de multiplier par deux la capacité à fournir des résultats.

La chaîne d'application évolue quotidiennement et est donc testée tous les jours dès la pré-prod', voire dès la recette s'il faut pousser l'analyse.



Avant l'automatisation, lorsqu'une livraison nécessitait un test particulier, il fallait tester le jour précédant la livraison, et, compter une journée pour qualifier après la livraison. Ainsi, la pré-prod' était monopolisée pendant deux jours. Les tirs nocturnes permettent d'avoir une référence chaque nuit. Un tir préalable n'est donc plus nécessaire, tout comme la programmation du tir après, car celui-ci est déjà programmé pour être exécuté le soir-même. De ce fait, les tests métiers peuvent se faire sur la pré-prod', en journée, sans entrer en conflit avec les tests de performance.

En s'inscrivant dans une démarche DevOps, le PMU a donc pu gagner un temps considérable sur les tests de performance :

L'anticipation des parcours utilisateurs évite les itérations en pré-production. Les tests sont déjà effectués dans les environnements amonts. Il est ainsi possible de savoir si tout fonctionne correctement ou si des problèmes apparaissent.

Le partage de Virtual Users à tous les utilisateurs de la plateforme permet à l'équipe de continuer à travailler en parallèle tout en poursuivant les tests réels, au-delà de l'implémentation de tests en cours.

L'automatisation Jenkins a permis l'utilisation des weekends pour faire des tests de non-régression de performance puisque personne ne travaille sur la préproduction à ce moment-là. Toute la partie applicative a été automatisée, ainsi que la partie performance : tous les weekends, des programmes de courses identiques sont injectés et vérifiés de weekend en weekend afin de s'assurer qu'il n'y ait aucune régression fonctionnelle et de performance sur ce périmètre-là.

L'utilisation des Weekends pour de la TNR perf et applicative fait gagner un temps considérable, et, énormément de disponibilités s'ouvrent dans la semaine pour d'autres tests quand tout le monde est là pour faire des analyses en même temps.

Aujourd'hui, 100% des tirs nocturnes sont automatisés permettant ainsi de gagner 30% de créneaux de tests. Le fait de gérer plusieurs projets en parallèle sur plusieurs environnements a accru la capacité des équipes à tester ainsi que leur réactivité dans le test de performance. Le nombre de tests a été triplé: aujourd'hui, le PMU peut lancer plus de dix tests de performance sur une journée là où il n'en faisait qu'un à trois maximum auparavant. Cela s'inscrit parfaitement dans la démarche d'agilité et d'automatisation DevOps qui se développe de plus en plus au sein du PMU. Démarche qui permet également d'opérer une transformation vers le cloud.

Résultat des courses, dans un univers de plus en plus concurrentiel, le PMU assure sa position de leader en proposant une release mensuelle quand il lui fallait auparavant un trimestre.