

Révolutionnez votre environnement Jenkins® avec la scalabilité horizontale et haute disponibilité

Jenkins est réputé pour sa flexibilité exceptionnelle, et s'est imposé comme l'outil d'orchestration le plus largement adopté par les organisations. Construire un [environnement Jenkins](#) qui soit à la fois sécurisé, scalable et hautes performances n'est pas chose facile pour une organisation comptant des centaines de développeurs et d'applications, encore plus si cet environnement doit être associé à une chaîne d'outils de développement complexes. Quelles sont les ficelles pour y parvenir ? Une expérience développeur solide, sans interruption, qui fonctionne en continu. Un système qui permet aux « Platform Engineers » de construire une plateforme robuste, de passer à l'échelle à l'infini, avec une tolérance aux fautes maximale et des capacités de résilience. C'est pour relever ces défis que nous avons doté [CloudBees Continuous Integration \(CloudBees CI\)](#) d'une nouvelle fonctionnalité : le mode Haute disponibilité CloudBees.

La haute disponibilité : la clé pour exécuter vos systèmes CI sans interruption

Les temps d'arrêt sont un fléau pour les systèmes de CI. La pire crainte des équipes est de perdre un contrôleur Jenkins. Que cette interruption intervienne dans le cadre d'une maintenance planifiée comme la mise à niveau de nœuds Kubernetes ou imprévue comme une panne matérielle, un manque de mémoire, etc., les tâches exécutées sur ce contrôleur s'arrêtent jusqu'à la reprise du contrôleur.

Il s'ensuit généralement des retards significatifs impactant le cycle de livraison logicielle.

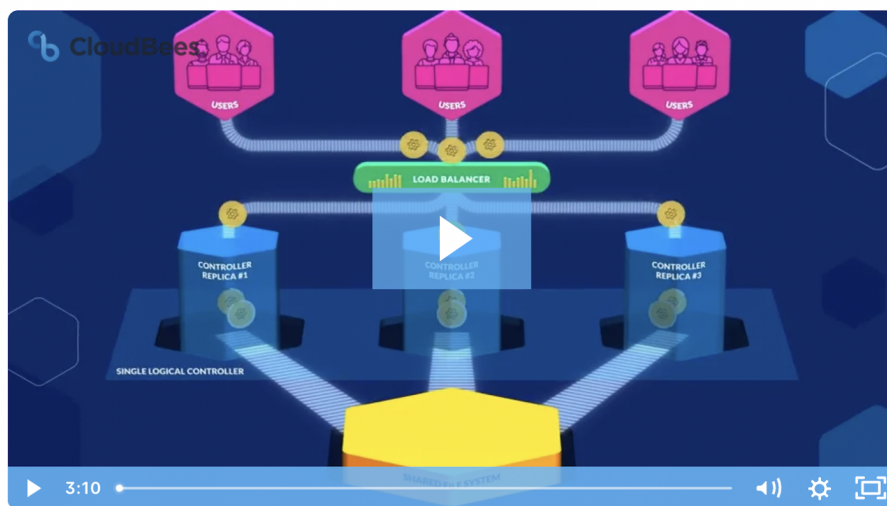
L'implémentation d'une [solution haute disponibilité \(HA\)](#) active-active comme CloudBees CI permet de réduire ou d'éliminer les temps d'arrêt et garantit la continuité opérationnelle : vos pipelines de livraison continuent à s'exécuter en dépit des interruptions inattendues et pendant les maintenances d'infrastructure programmées. Qu'est-ce que cela signifie ? L'état opérationnel de votre contrôleur est toujours actif empêchant toute interruption.

La scalabilité horizontale : une gestion efficace des charges

La Scalabilité Horizontale (HS) consiste à répartir la charge entre plusieurs contrôleurs répliqués afin d'optimiser les performances. L'activation du mode Haute disponibilité permet au contrôleur de gérer des charges volumineuses en toute transparence. Qu'est-ce que cela signifie concrètement ? Les problèmes de surcharge et de performances sont révolus, car le système répartit automatiquement la charge de travail entre les différentes répliques.

Cas d'usage du mode Haute disponibilité :

Découvrons deux cas d'usage du mode Haute disponibilité :



Cas d'usage 1 : contrôleurs monolithiques ou surchargés

On parle de contrôleur monolithique lorsqu'un seul contrôleur est utilisé par plusieurs équipes. Or, comme nous le savons tous, un contrôleur surchargé ne peut pas exécuter ses tâches avec efficacité. Cela se traduit par une baisse des performances, un ralentissement de l'exécution des pipelines, des ralentissements de l'interface utilisateur et même parfois des pannes. En cas de panne du contrôleur ou s'il doit être redémarré, ce sont tous les builds exécutés sur ce contrôleur qui sont impactés ainsi que les équipes qui l'utilisent. Dans ce type de situation, la haute disponibilité et la scalabilité horizontale procurées par le mode Haute disponibilité résolvent le problème en répartissant la charge entre les différentes répliques. Il n'y a donc plus de point de défaillance unique. Le contrôleur est performant et stable, avec des temps d'arrêt minimes ou inexistants.

Cas d'usage 2 : redémarrage progressif (« Rolling Restart »)

Lorsque des mises à niveau sont appliquées au serveur sur lequel sont exécutés vos contrôleurs, le service de CI est interrompu – que vous utilisiez la version traditionnelle de CloudBees CI, les nœuds Kubernetes de la version moderne de CloudBees CI, ou Jenkins. Cette maintenance est indispensable pour maintenir votre infrastructure à jour. Dans ce cas d'usage, Kubernetes est utilisé pour initier un redémarrage progressif (« rolling restart »). Grâce à l'activation du mode Haute disponibilité, une réplique est créée lors de la maintenance, ce qui permet au trafic de ne pas s'interrompre, et aux charges de travail de continuer à s'exécuter en toute transparence. Vous garantissez ainsi la continuité des opérations, malgré l'indisponibilité de la première instance.

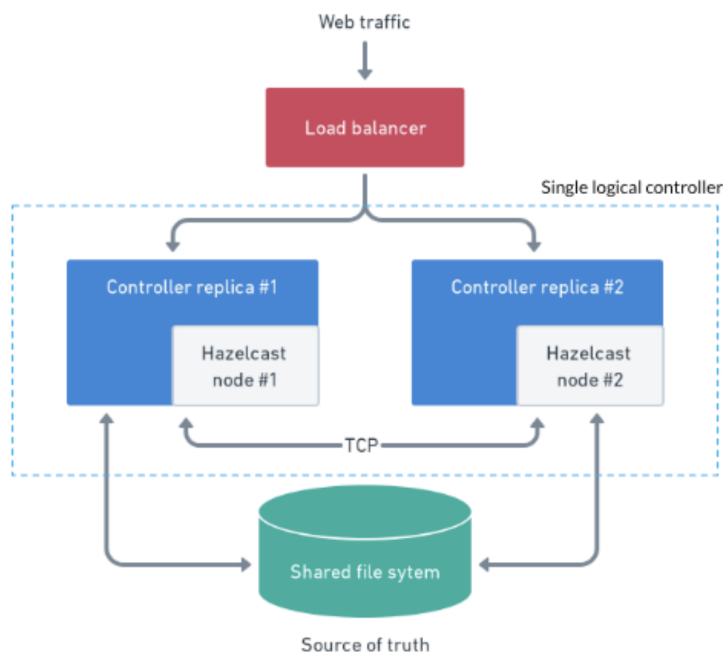
Faut-il vraiment casser le monolithe ?

Vous avez certainement déjà lu des articles qui font référence au concept de monolithe. On parle même de « casser le monolithe ». C'est d'ailleurs l'approche préconisée par CloudBees. Elle encourage une architecture solide caractérisée par un faible nombre de dépendances, des performances améliorées, et davantage de stabilité et de scalabilité. Toutefois, une architecture bien conçue est un luxe pour de nombreuses entreprises. C'est pourquoi les contrôleurs monolithiques ont tendance à se développer organiquement au fur et à mesure que les équipes s'agrandissent.

Pour « casser un monolithe », il faut du temps, et aussi planifier l'opération, ce qui n'est pas à la portée de toutes les entreprises. Le mode Haute disponibilité constitue une alternative pour ces entreprises. Il permet d'alléger la charge et d'améliorer la performance du monolithe sans qu'il soit nécessaire de le diviser physiquement en plusieurs contrôleurs.

Comment ça marche ?

Pour avoir une vue plus générale, commençons par examiner ce diagramme de deux répliques :



Lorsque le mode Haute disponibilité est activé, les répliques du contrôleur sont automatiquement créées. Le nombre minimum et maximum de répliques est configuré dans le logiciel et est automatiquement augmenté ou diminué en fonction de la charge de travail du contrôleur. Le trafic entrant (webhooks et pollings) est intercepté par le répartiteur de charge qui l'affecte ensuite aux différentes répliques.

Les données ne sont pas séparées et réparties entre les différentes répliques. Le répartiteur effectue une répartition dynamique de l'ensemble des données sur une ou plusieurs répliques. La bibliothèque Hazelcast est utilisée pour effectuer la synchronisation entre les répliques : un bus d'événement est utilisé pour la diffusion et l'écoute des événements afin de synchroniser les structures de

données entre les répliques. Supposons par exemple qu'une tâche soit en cours d'exécution sur la réplique 1, mais que cette dernière arrête de fonctionner. La réplique 2 reprendra automatiquement l'exécution de la tâche.

Conclusion

Avec le mode Haute disponibilité de CloudBees CI, vos contrôleurs supportent la haute disponibilité et la scalabilité horizontale. Cette fonctionnalité permet de permuter des charges de façon transparente vers au moins une réplique lors d'un redémarrage progressif (« rolling restart »). Elle simplifie l'administration et ajuste automatiquement le nombre de répliques pour s'adapter au passage à l'échelle. La répartition automatique des charges entre les différents contrôleurs répliqués met fin aux problèmes de performance. Deux avantages essentiels en découlent : une scalabilité horizontale totale dans l'éventualité où votre cluster nécessiterait davantage de répliques pour un contrôleur surchargé, et une haute disponibilité pour faire face aux pannes de contrôleurs.

Le mode Haute disponibilité vous permet de réinventer la façon dont vous gérez votre contrôleur. Efficacité et résilience sont désormais les maîtres mots, sans oublier des avantages non négligeables pour votre équipe : fonctionnement en continu des pipelines, performances améliorées, temps d'arrêt minimum !