



Mercedes-Benz

ÉTUDE DE CAS CONSUL

On the road again

Voitures, Kubernetes et HashiCorp Consul. Comment Mercedes-Benz s'appuie sur la mise en réseau des services pour accélérer la livraison de ses véhicules connectés de nouvelle génération.

// L'infrastructure favorise l'innovation

Mercedes-Benz en résumé

Mercedes-Benz Research and Development North America (MBRDNA) développe la technologie automobile et le design automobile les plus avancés au monde, doublés de luxe et de style. Chez MBRDNA, il n'y a pas que les voitures, il y a aussi les logiciels les plus récents et les plus performants, des technologies de pointe et des innovations révolutionnaires. Avec pour mission de repenser l'automobile, le MBRDNA se concentre sur la création des prochaines générations de véhicules électriques connectés et autonomes, ainsi que des écosystèmes qui les optimisent.

MERCEDES-BENZ QUELQUES FAITS EN BREF



25 ans d'innovation technique chez Mercedes-Benz



< **30 min** pour demander et déployer le nouveau cluster Kubernetes



Connexion et coordination de **milliards** de points de données



+200 services sur plusieurs clusters Kubernetes



Des **jours aux minutes** pour découvrir et connecter des services



De la preuve de concept à la production en seulement **4 semaines**

Le mieux, ou rien du tout

// La visibilité, la transparence et le contrôle dont nous jouissons avec Consul éliminent un grand nombre d'obstacles à la découverte de services et à la connectivité qui nous empêchaient de travailler aussi rapidement et efficacement que nous le voudrions.

En tant qu'inventeur de l'automobile, Mercedes-Benz, une marque de Daimler Group, a repoussé les limites du concept d'automobile, de l'introduction de la première motrice multisoupape à l'établissement de nouvelles normes en matière de tests de sécurité et d'ingénierie des performances. Mais pour tester constamment les limites de la technologie automobile, il faut bien plus que de l'imagination et un amour de la conduite.

Il faut être capable de combiner les technologies informatiques modernes avec des capacités avancées de modélisation des données pour réaliser de nouvelles innovations et de nouvelles percées. Et à l'ère des appareils connectés et des véhicules autonomes, il est nécessaire de construire une infrastructure cloud robuste et agile, capable de connecter et de coordonner les milliards de points de données qui formeront le « cerveau » qui optimise sa connectivité, un travail qui convient bien à la recherche et au développement de Mercedes-Benz en Amérique du Nord.

« Daimler a créé une plateforme de recherche et de développement spécialement conçue pour bâtir la base de son portefeuille de véhicules connecté et autonome sous la marque Mercedes », a déclaré Sriram Govindarajan, ingénieur principal en infrastructure chez Mercedes-Benz Research & Development (MBRDNA). « Presque immédiatement, nous avons réalisé que, pour construire les produits de demain, nous devons d'abord construire une infrastructure informatique adaptée à aujourd'hui, ce qui impliquait de déplacer la majorité de notre infrastructure existante vers le cloud et de moderniser nos processus de développement en même temps. »

Défis



Transition des infrastructures sur site vers les infrastructures cloud et kubernetes



Simplifier et accélérer la découverte et la mise en réseau des services



Améliorer l'efficacité et la productivité tout en contrôlant les coûts

// La visibilité, la transparence et le contrôle dont nous jouissons avec Consul éliminent un grand nombre d'obstacles à la découverte de services et à la connectivité qui nous empêchaient de travailler aussi rapidement et efficacement que nous le voudrions.

La visibilité zéro étouffe les performances

Les voitures connectées sont présentées en avant-garde de la tendance de l'Internet des Objets (IoT), conçue pour partager des données en temps réel avec d'autres systèmes à l'extérieur du véhicule, dans le but d'améliorer considérablement la sécurité, les performances et le confort des passagers. Chaque aspect de cette connectivité est optimisé par une application spécialement conçue, et chaque application est optimisée par un ensemble de micro-services qui doivent se connecter avec d'autres services afin de compléter le système.

Avec Microsoft Azure comme plateforme cloud, MBRDNA a adopté Kubernetes pour conteneuriser le développement, et fournir la vitesse et l'agilité nécessaires à la fourniture plus rapide de nouvelles fonctionnalités. « Kubernetes était l'idéal pour nous, parce que celui-ci centralise la base du code, l'infrastructure et tout ce dont nos équipes ont besoin pour rapidement développer les fonctionnalités et les capacités de notre feuille de route », explique M. Govindarajan. « Mais plus nous avons de services dans chaque groupe, plus il devenait difficile de les faire correspondre et de les connecter car nous n'avions pas les bons mécanismes de découverte de services en place »

Selon M. Govindarajan, les équipes de développement peuvent avoir des services fonctionnant dans leurs clusters individuels, mais aucun moyen d'identifier d'autres services ou des services dépendants fonctionnant sur d'autres clusters. Compte tenu de la rapidité avec laquelle les adresses de service, les hôtes et les ports sont ajoutés et modifiés, il était pratiquement impossible de le faire sans un outil de découverte de services dédié. « La découverte de services est essentielle pour l'ensemble de notre exploitation et pour la capacité à assumer nos responsabilités envers l'organisation au sens plus large », explique-t-il. « Un développeur pouvait parcourir tout notre environnement pendant des jours sans jamais trouver ce qu'il cherchait. Nous avons donc besoin d'un moyen d'accélérer la découverte et de le faire sans effort, à travers un certain nombre de clusters et de clouds. »

// Consul nous permet de propager plus de 200 micro-services parmi plusieurs clusters AKS. Chaque cluster AKS se connecte à un client Consul local, qui alimente un cluster Consul qui forme un maillage de découverte de services plus important, qui à son tour nous permet de trouver et de connecter des services en quelques minutes, avec un effort minimal.

SRIRAM GOVINDARAJAN, INGÉNIEUR PRINCIPAL EN INFRASTRUCTURE,
MERCEDES-BENZ RESEARCH & DEVELOPMENT (MBRDNA)

Découverte de services centralisés basés sur le cloud pour une plus grande efficacité

Après avoir brièvement évalué à la fois les outils open source et les outils de découverte payants, MBRDNA a choisi HashiCorp Consul comme mécanisme de découverte de services dans son effort de migration vers Azure.

Avec l'architecture initiale en place, l'équipe de Govindarajan a concentré son attention sur son travail critique : optimiser ses opérations de développement de code.

Contrairement aux autres outils de découverte de services sur site et open source évalués par le MBRDNA, les fonctionnalités cloud natives de Consul permettent une localisation dynamique des services applicatifs et d'infrastructure, et offrent une interconnexion facile à travers toute plateforme d'exécution ou cloud. Le registre centralisé des services, la configuration centralisée et automatisée du middleware réseau et le répertoire en temps réel de tous les services en cours d'exécution se combinent pour améliorer considérablement la gestion de l'inventaire des applications, et accélérer la connectivité des services.

Par exemple, les développeurs de MBRDNA peuvent désormais requérir un cluster Azure Kubernetes Service (AKS) via un portail en libre-service et avoir tout ce dont ils ont besoin pour leur déploiement — code, scripts, API et clients Consul — en moins de 30 minutes. Une fois que leurs services ont été déployés, d'autres équipes peuvent simplement utiliser un code ou des scripts sur leurs clusters, dans le but de rechercher les services ou le client Consul dont ils ont besoin, puis les connecter instantanément.

« Consul nous permet de propager plus de 200 micro-services sur plusieurs clusters AKS », explique M. Govindarajan. « Chaque cluster AKS se connecte à un client Consul local, qui alimente un cluster Consul qui forme un maillage de découverte de services plus important qui nous permet de trouver et de connecter des services en quelques minutes avec un effort minimal. »

La vie sur la voie rapide

Govindarajan dit que, bien qu'il puisse être difficile de quantifier des améliorations spécifiques, les solutions HashiCorp ont fondamentalement modifié la façon dont son équipe travaille.

« Consul et les autres outils HashiCorp nous ont permis d'apporter l'ensemble du processus de développement en interne, et d'arrêter de compter sur des tiers pour être responsables d'une partie de nos workflows », explique-t-il. « La visibilité, la transparence et le contrôle dont nous jouissons avec Consul éliminent un grand nombre d'obstacles à la découverte de services et à la connectivité qui nous empêchaient de travailler aussi rapidement et efficacement que nous le voudrions. »

La conception intuitive et les ressources de support à la demande ont permis à MBRDNA de configurer et de lancer Consul en seulement 12 semaines, malgré le fait que l'équipe était nouvelle dans le monde du DevOps et de l'infrastructure cloud. Au fur et à mesure que l'équipe s'est familiarisée avec Consul, elle a commencé à utiliser les autres fonctionnalités axées sur l'efficacité du produit, telles que les requêtes de serveur et la segmentation des services, dans le but de réduire encore davantage la pression sur les réseaux de la société et sur son budget.

Plutôt que de se procurer différents serveurs pour les environnements de production, d'organisation, et de développement, l'équipe utilise des tags qui indiquent si la cible fonctionnant sur un serveur central de Consul est en production, en développement, ou en staging. Cette méthode simplifie la découverte et réduit les connexions inutiles au cluster client, tout en réduisant considérablement les coûts de certificat et de maintenance du serveur.

Résultats opérationnels



Découverte automatisée de plus de 200 services sur plusieurs clusters Kubernetes



Création d'une plateforme de découverte de services basée sur Consul pour une visibilité et une transparence accrues



Transition du processus de développement interne dans son entier, loin des fournisseurs tiers externalisés



Réduction des coûts de certificat et de maintenance de serveur, grâce à l'utilisation de tags réseau

Partenaire MBRDNA



Installé dans le nord-ouest Pacifique, Sriram possède plus de 20 ans d'expérience en ingénierie dans le secteur informatique. Avec une concentration appuyée sur la philosophie DevOps, Sriram s'y connaît très bien dans l'orchestration de conteneurs avec Kubernetes, et incarne une solide compréhension de l'architecture cloud. Actuellement, Sriram est ingénieur principal en infrastructure chez MBRDNA, où il travaille sur ses programmes autonomes et connectés au sein du parapluie d'innovation Mercedes-Benz.

Sriram Govindarajan, Ingénieur principal en infrastructure, Mercedes-Benz Research & Development

Conclusion

Govindarajan affirme que le succès de son équipe à ce stade inspire confiance en ce qui concerne la manière dont ils l'utiliseront à l'avenir. « Consul s'est avéré être la bonne solution pour résoudre nos défis concernant la découverte de services que nous avons trouvée au bon moment », explique-t-il. « Nous sommes impatients de commencer à utiliser les fonctionnalités améliorées des prochaines versions de Consul, afin de pouvoir continuer à renforcer la résilience et les performances de nos infrastructures, et aider à mettre en place la prochaine génération d'engins roulants innovants. »

Pile technologique

- Infrastructure : 100 % sur Azure
- Plateforme : Conteneurs majoritaires et quelques machines virtuelles
- Proxies : Prévu pour l'avenir, pour soutenir le maillage de service
- Équilibreurs de charge : Équilibreurs de charges publiques Azure
- Pare-feu : Pare-feu Azure
- API gateway: Azure APIM
- CA : fourni par Daimler en interne
- IAM : Outils internes basés sur AAD et OAuth
- APM : Principalement : Azure Application Insights et un peu de AppDynamics
- Provisionnement : Providers Terraform
- Gestion de la sécurité : Azure Key Vault, transition vers HashiCorp Vault

