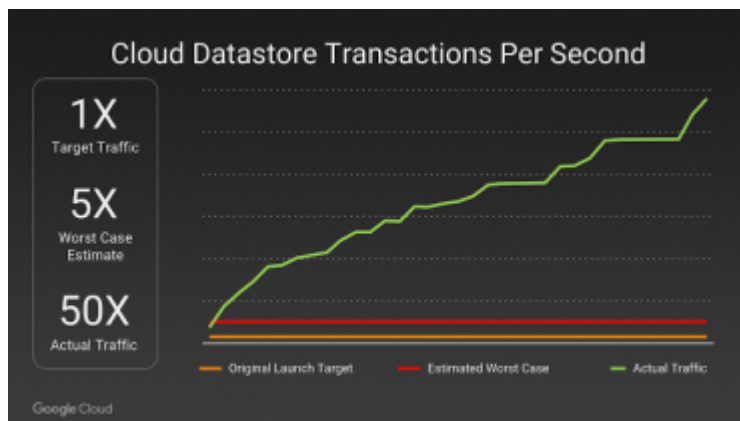


Pokémon Go peut remercier Google Cloud, Kubernetes et Docker

Le jeu phénomène de l'été, Pokémon Go, n'aurait certainement pas été un aussi grand succès si l'infrastructure IT n'avait pas suivi. A la fin juillet, nous nous étions fait l'écho [des enjeux IT](#) derrière l'adoption rapide de l'application de réalité augmentée. Niantic, éditeur du jeu, avait d'ailleurs recruté en urgence un spécialiste de l'infrastructure IT et du Cloud. Celui-ci devait absolument avoir une compétence sur les plateformes IaaS de Google et d'AWS.

Un pic de charge hors norme

[Sur le blog de Google Cloud](#), Luke Stone, directeur du support technique, donne des détails supplémentaires sur le rôle crucial du Cloud dans le succès de Pokémon Go. Tout d'abord le responsable explique que cette application est hors-norme. Il avoue avoir accompagné plusieurs déploiement d'applications, mais celle-là dépasse et de loin ce qu'il a connu. Pour preuve, il publie un simple graphique (ci-dessous) montrant l'impact de la croissance de Pokémon Go sur le datastore du Cloud de Google.



La firme de Mountain View souligne que l'application mobile utilise plusieurs services au sein de la plateforme Cloud. « *Mais Cloud Store est devenu un proxy direct pour le jeu en tant que base de données primaires pour capturer les Pokémon* », précise le dirigeant. Sur le graphique, on constate que l'équipe de Google s'est fixé un trafic estimé de joueurs et une fourchette haute correspondant à 5 fois le trafic estimé de participants. La popularité du jeu a fait exploser les statistiques en multipliant par 10 l'estimation haute. Ce pic de trafic a été absorbé par l'équipe de Google en provisionnant suffisamment de ressources.

Le plus grand utilisateur de conteneurs et de Kubernetes

Pokémon Go est également un cas d'école pour l'usage de la technologie des conteneurs. En effet, l'application fonctionne sur des conteneurs développés par Google Container Engine (GKE)

s'appuyant sur Docker. Le tout est orchestré par la solution Open Source, Kubernetes, promue par Google. Luke Stone précise que « *Niantic a choisi Kubernetes pour sa façon de gérer le multi-clustering au niveau mondial et laisser les équipes de développeurs se concentrer sur d'autres sujets comme l'amélioration du jeu* ». Il cite le cas concret d'une mise à jour de GKE pour intégrer des milliers de nœuds au sein d'un cluster (qui en comprend déjà des dizaines de milliers) en prévision du lancement du jeu au Japon.

Comme dans un moteur d'avion, Google a récupéré des données en temps réel lors de la montée de version pour éviter les perturbations lors de la bascule vers la nouvelle version du moteur de conteneur. Lors de cette montée de version, les équipes d'ingénieurs des deux sociétés en ont profité pour changer de répartiteur de charge (load balancing) traditionnel vers un service gérant le trafic HTTPS. Pour cette migration, Google s'est servi des enseignements tirés du lancement de l'application aux Etats-Unis.

Une équipe dédiée chez Google Cloud

A l'occasion de l'évènement Horizon, Google a présenté ce cas client pour annoncer la création d'une structure : Customer Reliability Engineering (ingénierie de la fiabilité client) qui mêle à la fois des équipes techniques du client et celle de l'activité Cloud de Google. Objectif : créer un partage de responsabilité sur la robustesse des applications critiques dans le Cloud. Niantic est effectivement un cas hors norme, mais il démontre la flexibilité et l'agilité du IaaS, ainsi que de formidable potentiel de la technologie des conteneurs en production.

A lire aussi :

[L'ANSSI alerte sur les risques liés à Pokémon Go](#)

[Pokémon Go : un défi pour les serveurs et le Cloud](#)