

Use Case - La société Atlassian

La société

- La société Atlassian développe et livre des produits de gestion de projets en mode Saas.
- Des produits comme Jira , Confluence, HipChat, Bamboo CI, Bitbucket.
- La société a 20.000 clients dans 130 pays.
- un client est en fait une équipe ou une société qui a signé pour une ou plusieurs applications
Il y a 60.000 instances d'applications qui prennent 1TB du trafic réseau par jour.
- Pour supporter cela, il y a 2 datacenters de production au US, environ 60 racks de serveurs avec 8.200 cores et 5.300 disques physiques.
- Les applications des clients sont déployées dans des containers OpenVZ.

Les enjeux

- Passer d'une architecture monolithique vers une architecture à base de microservices.
- Les objectifs de la migration sont:
 - de fournir de meilleures performances
 - d'améliorer l'évolutivité de la solution
 - de supporter les cloud publics pour les performances et le coût
 - un meilleur support de site de reprise d'activité
 - d'améliorer le mise en place de nouvelles fonctionnalités par des facilités de déploiement

Les objectifs de l'architecture en microservices

- garantir que les données sont migrées dans le bon format et sur les bons emplacements sans risques de perte.
- pas de perte de fonctionnalités dans les applications.
- pas de perte de connectivité .
- accès aux fichiers de log et au système de débogage et de support .
- assurer le monitoring et le lancement d'alertes.

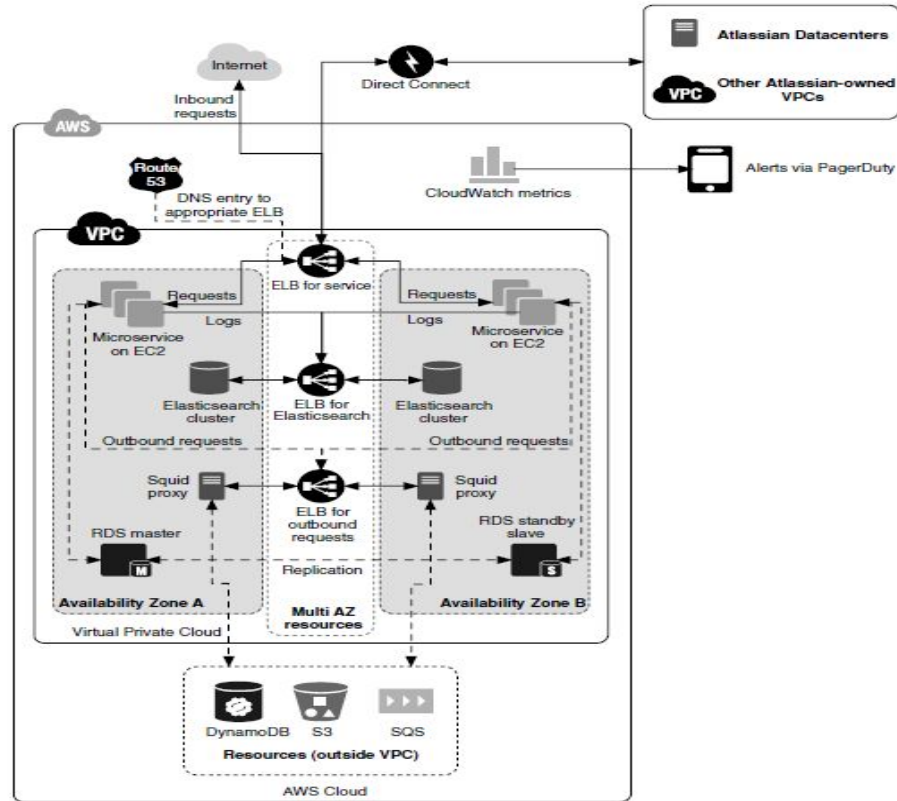
Les pré-requis

- Atlassian a prévu de déployer ses microservices dans le cloud public d'Amazon Web Services.
- Incluant une architecture VPN comme Amazon Direct Connect qui permet une double connections avec les datacenters d'Atlassian.
- Pas de temps mort ou d'indisponibilité pendant la période de transition.

Le Paas d'Atlassian

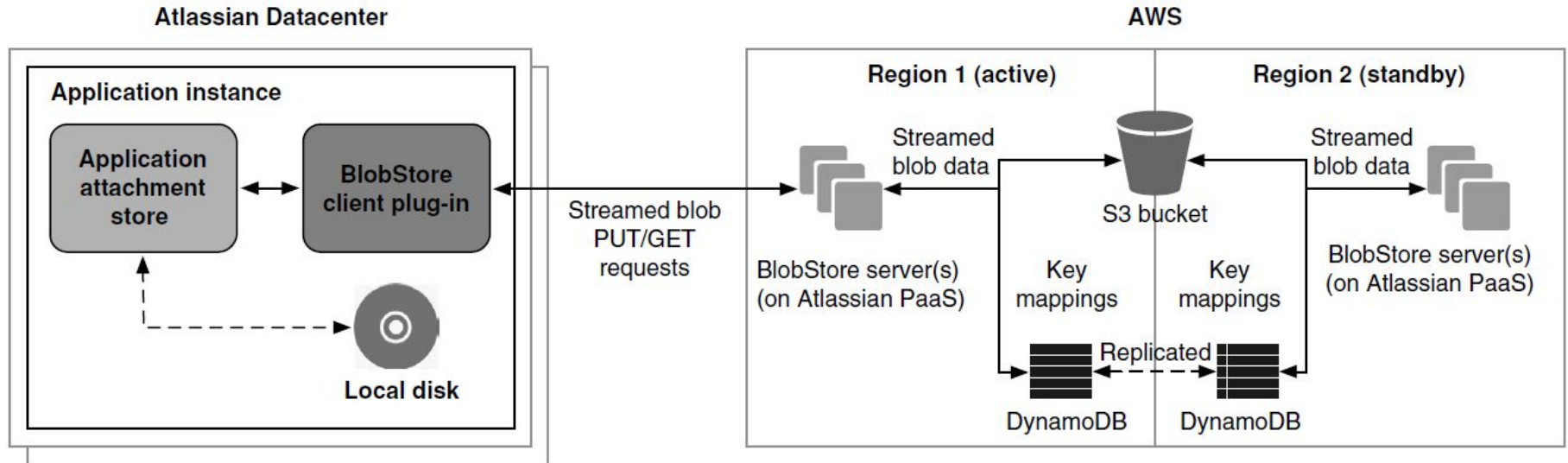
- Les instances de microservices sont exécutées dans des images AMI spécifiques à l'intérieur de EC2, ces images sont gérées par une équipe dédiée, pas par l'équipe qui développe les microservices.
- L' AMI contient les exécutables nécessaires à l'exécution des microservices.
- les développeurs fournissent une description du service à configurer et des metadata comme les variables d'environnement, et les ressources nécessaires et des informations sur les logiciels tiers qui doivent être présents, ou des fichiers jar spécifiques ou une image docker
- Assure le management des buckets S3, DynamoDB et Simple Notification Service Autoscaling et load balancing.
- Les fichiers Log sont automatiquement prélevés et parsés par fluentd et livrés vers un ElasticSearch cluster.
- Les développeurs et l'équipe support ont accès aux logs via Kibana.

L'architecture

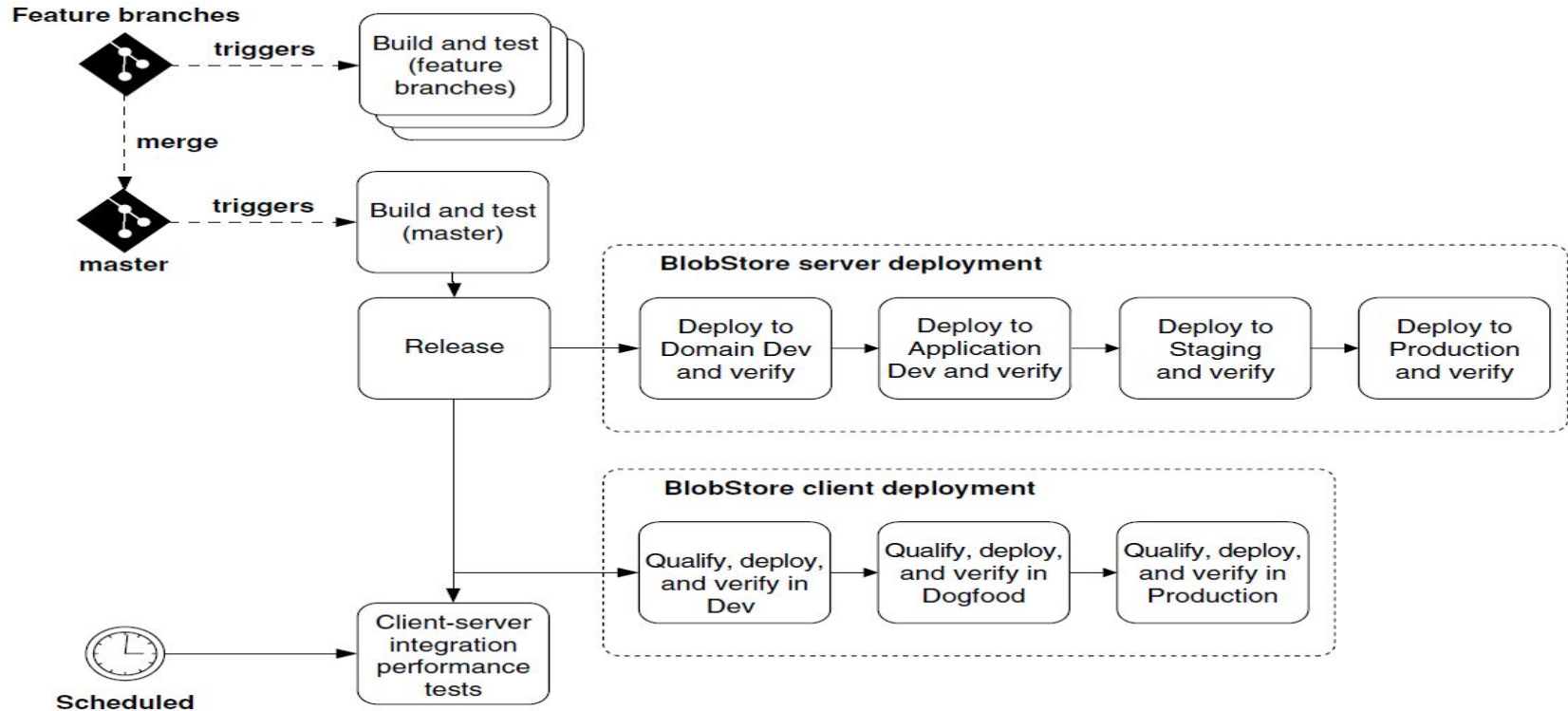


Migration d'un service

- Le Service BlobStore



Déploiement Pipeline



L'architecture logicielle

- Mélange de Java, Scala et Node.js
- 2500 lignes de code .
- Géré par une petite équipe.
- Utilisation de RPC Finagale, in framework de chez Twitter.
- Adoption de la programmation fonctionnelle.
- Permet de vérifier les erreurs de code à la compilation.
- Résoudre les “ilities”
 - Scalability
 - Availability
 - Security
 - Extensibility
 - Maintainability