**Introduction**

Fin 2018, à l'occasion de la conférence AWS Re:invent à Las Vegas, BluAge présente officiellement "serverless COBOL for AWS solution". Composé de deux produits : une extension VSC servant de compilateur COBOL vers Java et un Framework associé, les deux produits combinés de déployer de manière « serverless » une application initialement codée en Cobol.

III- Réalisation e la chaine de déploiement.

### Etape de recherche : Solutions non-retenus

## Jenkins

Usuellement, un pipeline est développé avec un outils tel que Jenkins. C’est d’ailleurs cet outil qui est utilisé pour l’intégration (compilation, test...) du Framework Velocity. Une première solution envisagée été d’utilisé le même outil pour la chaîne de déploiement. Cela est possible utilisant un conteneur sur lequel serait téléchargé l’interface en ligne de commande que propose Amazon. L’inconvénient de cette méthode est qu’elle nécessite de faire toutes les commandes en dehors du cloud Amazon, et e\*\*\*\*\*\*

## Services Amazon Spécifique : Step Function, CodeDeploy

Amazon propose plusieurs services pour la réalisation de pipeline et de work flow. En ce qui concerne Amazon CodeDeploy, le choix de ne pas le retenir viens d’une part de son cout, et d’autre part du fait qu’il soit surtout conçu pour être utilisé avec les autres services d’Amazon (Services d’Amazon qui permet d’héberger toutes les étapes du dev, git, intégration, déploiement) tandis qu’on ne veut l’utiliser uniquement pour le déploiement. Amazon Step Fonction quant à lui, utilise les lambda comme environnement d’exécution, ce services rajoute du code supplémentaire pour lier les Lambda entre elle (en un langage propre à Amazon, proche du json). Ce service est très bien pour obtenir un joli aspect visuel du work flow, mais il m’a semblé plus judicieux d’utilisé les fonctions lambda seule. Certes, on n’obtient pas l’aspect visuel du pipeline, mais l’utilisation est identique. On évite ainsi un cout supplémentaire pour l’utilisation du service et on évite. On évite d’ajouter du code supplémentaire à l’architecture.

N.B : Plus que de m’être renseigné sur ces outils, je les ai testés et j’ai réalisé des petites ébauches de pipeline. En plus des arguments cités une petite partie de ma décision est aussi liée à mes préférences personnelles.

N.B : J’ai également dû réfléchir à la méthode programmatique que j’utilisais pour écrire dans écrire dans un fichier. Le premier jet que j’ai rendu était trop lent.