Sommaire

# Préambule

Le présent mémoire est réalisé en tant que rendu final de la formation « DevOps », réalisée par l’organisme IB Formation pour le groupe Néo-Soft. Les autres rendus de cette formation sont un support de présentation et une soutenance orale, réalisée devant jury, portant sur la tenue du projet de fin de formation, effectué par groupe de quatre personnes au cours des mois de juillet et août 2021. Ce mémoire va lui aussi s’attarder sur la présentation de ce projet final.

Cette formation « DevOps » était destinée à un public possédant déjà de solides bases d’informatique, soit sur l’aspect développement, soit sur l’aspect infrastructure, et une certaine expérience professionnelle, acquise au sein de Néo-Soft ou lors d’expériences antérieures. Comparativement à la plupart des formations réalisées par IB Formation, mais aussi à celles proposées à d’autres collaborateurs de Néo-Soft, la durée de celle-ci était dite « longue ». La formation a commencé mi-mars 2021, pour se poursuivre sous forme de cours, de mises en applications et de travaux pratiques jusqu’à fin juin 2021. Un complément théorique, visant à préparer le passage de la certification DevOps est effectué en sus au mois de septembre 2021, sur trois jours. A cela s’ajoute durant la fin du mois de juin, les mois de juillet et d’août, ainsi que le début du mois de septembre 2021, la mise en place, sur la base d’un sujet imposé, d’un projet, dit DevOps.

Les cours, mises en application et travaux pratiques ont couvert, sur l’aspect technique, l’essentiel des domaines qui peuvent être rencontrés dans le métier d’ingénieur DevOps. Les sujets ont été abordés d’une manière quasi chronologique, vis-à-vis de leur mise en œuvre au cours d’un cycle de vie DevOps : nous avons commencé par nous intéresser à des notions d’administration et d’automatisation sous Linux, puis nous avons abordé le langage Python – d’une part pour son application dans un but de développement (toutes les applications réalisés au cours de la formation ont ainsi été codées, pour l’essentiel, en Python) mais également de scriptage -, avant de découvrir le volet CICD du métier. Au programme, mise en place et administration d’un serveur Jenkins, découverte de Nexus, bonnes pratiques de tests, notions de base sur Docker, Kubernetes et Ansible pour pouvoir effectuer des déploiements d’applications, survol de la suite ELK pour le monitoring. Enfin le mois de juin a été largement consacré à l’initiation au Cloud, à travers des solutions AWS et Azure.

L’aspect technique du métier n’a pas été le seul exploré. Une grande importance a été accordée au travail en équipe et à la coordination de groupe, d’une part par le fonctionnement de base de la formation – peu scolaire, fait de nombreux feed-back, et d’une part d’autogestion mature des membres de la promotion – et d’autre part par la présence de cours sur l’agilité, l’utilisation d’outils de travail en équipe et de gestion de projet (tels que Jira et Confluence). Le mode de fonctionnement agile a de fait baigné la formation dans son ensemble, et été, peu ou prou, le mode de fonctionnement utilisé pour chaque travail de groupe en autonomie – qu’il s’agisse des travaux pratiques ou du projet final.

Le groupe ayant suivi cette formation est composé de seize personnes. Aucun d’entre nous n’était particulièrement familier avec l’environnement DevOps. Les membres du groupe étaient soit très fortement orientés développement, soit très fortement orienté opérationnel, avec la présence de quelques profils un peu plus atypique (embarqué en particulier). De ce fait la formation a été pour chacun un enchaînement de sessions sur lesquelles ils avaient déjà un bon bagage de base, qui n’a été que légèrement renforcé et remis dans une perspective différente, et de sessions de découverte totale, assez fortement challengentes pour réussir à s’approprier au maximum les nouveaux concepts et outils – nombreux – dans un laps de temps très court.

La gestion de cette dualité au sein du groupe d’apprenants a été la principale difficulté rencontrée au cours de la formation – difficulté encore renforcée par l’ampleur du groupe, le rythme soutenu, sans réelle pause pendant trois mois et demi, et le processus d’apprentissage en distanciel. C’est cette dualité qui a amené plusieurs fois au cours du parcours à rectifier la proportion et l’agencement des séances de cours « théoriques » et de mises en pratique. Cette dualité, encore, qui a amené à ajuster le contenu du programme, en l’allégeant de sorte à permettre que chaque participant à la formation puisse en sortir avec a minima les notions de bases et les concepts clefs de chaque module correctement assimilés. Cette difficulté aura en revanche eu l’heur de nous permettre une mise en pratique « grandeur nature » d’une partie des notions abordées dans les modules de gestion de projet et de groupe.

Les semaines de travaux pratiques en autonomie avaient pour objectif de nous permettre de mettre rapidement en pratique, sur un sujet avec des enjeux réalistes, les notions vues. Ces semaines ont reflété la dualité précédemment évoquée. Si elles ont permis aux membres de la formation de pratiquer davantage les domaines dans lesquels ils n’avaient pas de pré-acquis – et donc de renforcer leur maîtrise sur ces sujets – elles ont également mis en évidence la difficulté de confier à des néophytes la réalisation des parties qu’ils maîtrisaient le moins bien, même dans le cadre bienveillant de la formation. In fine ce sont souvent les plus qualifiés qui ont donné l’impression de s’investir massivement sur leurs sujets de prédilection, chacun se clôturant en quelque sorte dans son rôle de « dev » ou « d’ops », et ne faisant que rarement la vraie jonction entre les deux fonctions, si ce n’est à l’échelle du groupe.

La complexité sus-mentionnée aura été un enjeu de la tenue du projet final. La répartition des tâches, leur évaluation, à l’aune des compétences de chacun et de la volonté de chacun de sortir de sa zone de confort pour se renforcer sur les sujets les plus nouveau et/ou les moins bien maîtrisés est crucial pour mener à bien un projet, surtout sur une temporalité aussi courte et aussi parcellée par les congés estivaux. Peu de place est laissée à la réelle montée en compétence de chacun au cours de la réalisation du projet et il est important de jauger en temps réel la faisabilité des ambitions de groupe sur le rendu final du projet.

# Présentation du projet

Deux sujets étaient proposés aux quatre groupes pour ce projet final de la formation DevOps. Charge à chaque groupe de s’entendre et de choisir l’un des deux, selon que la thématique lui semblait plus correspondre à ses attentes, ou que les technologies vraisemblablement à utiliser cadreraient avec les compétences des membres.

J’ai travaillé au sein du « Groupe 4 », composé de Houssem Ben Ali, Arnaud Degez, Michael Tongle, et moi-même. Nous avons choisi de travailler sur le sujet 2, dont je vais faire ci-après une rapide présentation, complétant le sujet de base avec nos interprétations, à l’occasion, et quelques éclaircissements.

Le groupe doit fournir à un client une application, accessible uniquement par des « administrateurs » (vraisemblablement des employés de l’entreprise cliente). Ces administrateurs seront contactés par des « rédacteurs » (par un moyen externe à l’application, à tout le moins pour une première version de l’application, et selon les indications présentes dans le sujet). Ces rédacteurs disposent d’un compte Github, et souhaitent, dessus, créer un repository dans lequel ils stockeront des fichiers markdown. L’objectif final de ces rédacteurs est que ces fichiers markdown puissent former les différentes pages d’un site (a priori statique donc) web. Ils communiquent aux administrateurs les données nécessaires pour pouvoir accéder à leur repository GtiHub (url, et secret si repository non public) ainsi que le nom qu’ils souhaitent voir présent dans l’url à laquelle on peut accéder à leur site.

De là, l’application doit permettre, après saisie de ces informations par les administrateurs, d’accéder au repository GitHub, de récupérer les fichiers markdown présents, de les convertir en fichier html, et enfin de les rendre accessible en ligne, à l’aide d’une url contenant le nom voulu par les rédacteurs.

A tout cela s’ajoute l’environnement à mettre en place autour de cette application. Une chaîne de CICD (Continuous Integration/Continuous Deployement ou Delivery) doit être établie. Elle permettra de tester l’applications, de packager ses sources, sa documentation, ses rapports de tests et de livrer ce package au client, de déployer l’application sur une solution cloud de sorte à ce qu’elle soit accessible par les administrateurs. Egalement du monitoring est attendu sur l’application. Nous proposons donc de mettre en place un suivi de logs pour tracer le fonctionnement de l’application, et repérer au plus vite les erreurs (coté administrateur) et pouvoir les transmettre de manière détaillée (côté développeurs), et un suivi des performances des différents sites (fréquentation, visible uniquement par les administrateurs dans un premier temps, pouvant devenir une donnée visible sur les sites, ou bien uniquement par les rédacteurs dans un futur plus ou moins lointain).

L’application devra être déployée à l’aide de Docker et de Kubernetes, et il n’est pas possible de se servir de Ansible (contrainte du sujet, à l’inverse du sujet 1 qui permettait l’utilisation de Ansible mais pas celle de Docker et de Kubernetes).

# Organisation du projet

## Sprint 0

Si l’essentiel du projet devait être traité par les membres du groupe en autonomie sur les deux mois de juillet et août, le « sprint 0 » a été réalisé en semi-autonomie à la fin du mois de juin, occasionnant des présentations du backlog et des solutions techniques retenues pour le projet à la formatrice principale. Cette semaine de sprint 0 a aussi été l’occasion d’approfondir notre compréhension du sujet, somme toute présenté de manière succincte par son énoncé.

Ce sprint 0, d’une semaine, donc plus court que les sprints effectivement réalisés par la suite, avait pour objectif de permettre au groupe de réfléchir sur les solutions techniques qui seraient utilisées au cours du projet (langages de programmation utilisés pour les aspects front et back de l’application, architecture – même si embryonnaire et vouée à évoluer au cours du projet – de l’application, solutions à utiliser pour gérer les différents aspect CI/CD, fournisseur Cloud). Le sprint 0 était également l’occasion de mettre en place les process agile qui allaient nous accompagner tout au long de l’été. Quid de la « Definition of Done » ? Durée des sprints ? Composition du backlog – exercice d’autant plus compliqué qu’il impliquait d’imaginer un besoin client que nous n’avions pas, attendu que nous découvrions juste le sujet lui-même ? Répartition des tâches au sein de l’équipe ?

Premier exercice du projet donc, et exercice complexe. Dès cette semaine nous rencontrons des difficultés, et cernons parfois mal le travail à faire. Ce qui est également le cas des autres groupes. Il s’agit bien sûr de l’objectif de l’exercice : nous forcer à retarder les étapes de développement pures – actions relativement contre intuitive et difficile à supporter pour la plupart des développeurs (au sens Agile du terme) –, fournir un type de réflexion que nous n’avons pas l’habitude de fournir. De surcroît en ajoutant la difficulté de devoir par moment se mettre dans la peau du client – ce qu’en tant que développeur nous n’aurons pas à faire dans cette envergure-là, même si se projeter dans le rôle du client et/ou de l’utilisateur final est essentiel pour mener à bien un développement d’application intelligent et pertinent – d’autant plus que nous en étions encore à découvrir le sujet et donc le besoin client résumé à son plus simple statut, et de fournir un travail d’étude et d’architecture sur un panel de technologies et de sujets très vastes, que nous maîtrisions somme toute sommairement, et que nous n’avions jusque-là globalement jamais orchestré ensemble.

Parmi les premiers rendus de ce sprint 0 quelques diagrammes UML (diagramme de classe et diagramme de cas d’utilisations). Ceux-ci méritent à peine le titre d’ébauche, tant ils sont simplistes et dans l’absolu, et vis-à-vis de ce à quoi ressemblera finalement notre application, tant du point de vue fonctionnel que du point de vue structurel.

Autre rendu : le backlog. Ce premier jet nous a pris plus d’une journée à imaginer, mettre en mots, estimer. Il manque cruellement de découpage de tâches, de description de stories. Il lui manque aussi quelque chose d’essentiel : exprimer un besoin client. Les protos backlogs (a priori ceux des trois autres groupes entrant dans la même catégorie que le nôtre) ont été réalisés par des développeurs, dans une optique de développeurs. Les stories sont techniques, ne correspondent pas à la mise en place de fonctionnalités, se segmentent mal en sprint. Il s’agissait en fait plus des grandes parties que nous, développeurs, allions devoir mettre en place d’un point de vue technique, que d’un vrai backlog.

La semaine, qui nous paraissait bien longue pour cet exercice de sprint 0 le premier jour, s’avère finalement bien courte. Le travail en lui-même nous a pris beaucoup de temps, et la plupart de nos travaux préliminaires, s’ils ne sont pas rendus complètement caduques après les avoir présenté à Vanessa, sont très largement à remanier. De nombreux choix techniques restent flous : certes le développement sera réalisé avec Python et Flask pour l’application, mais quelles librairies seront nécessaire pour structurer correctement l’application, gérer l’aspect authentification, quelles solutions utiliserons-nous exactement pour les différents types de tests, quels tests seront véritablement nécessaires à être mis en place… ? Pour ne citer que quelques-unes des questions laissées en suspens sur ce simple aspect.

En conclusion de cette semaine de sprint 0, ont été mises en évidence nos limites, prévisibles et criantes, dans nos capacités techniques et dans notre capacité à prévoir, anticiper les besoin techniques et fonctionnels d’un projet. L’importance et l’impact de ce sprint avaient clairement été sous-estimés en amont. Il aurait été bon de le prolonger bien après cette première semaine, de sorte à mettre de manière efficace et sereine à plat toutes les remarques, et d’éclaircir les nombreuses zones d’ombre encore présente. Néanmoins l’apport de cette première semaine, aussi approximativement bien menée qu’elle ait été, se fera sentir durant tout le reste du projet.

## Organisation du travail de groupe

L’ensemble du projet a été réalisé à distance, de même que le reste de la formation. Pour permettre la communication au sein de la promotion, deux médias avaient été mis en place : Teams et Slack. Un canal d’équipe Teams dédié a été octroyé à chaque groupe pour assurer la communication lors du projet.

Mon groupe (groupe 4) ne s’est pour ainsi dire pas servi de Slack pendant la durée du projet. Le canal d’équipe Teams a été utilisé, essentiellement pour les daily meetings, mais aussi pour réaliser des réunions parfois plus longues, afin de discuter solutions techniques architecture, etc… Une conversation de groupe sur Teams a aussi été mise en place afin de permettre des échange plus informels.

Au-delà de l’aspect communicationnel, notre travail de groupe s’est également articulé autour de l’outil de gestion de version Git. Sur Github nous avons créé un repository dédié au projet fil rouge. GitHub nous a aussi servi à gérer l’aspect gestion de projet.

Partant du principe que le projet serait court, mettait en collaboration une équipe relativement réduite et occasionnait un volume de travail limité, sécable en un nombre maîtrisé d’items/User Stories, nous avons souhaité éviter de passer par l’outil que nous avions vu au cours de la formation – et avec lequel plusieurs membres de l’équipe étaient déjà plus ou moins familiers – Jira. GitHub propose, dans une mesure bien plus réduite que Jira – à tout le moins en ce qui concerne des versions non entreprise/gratuite des deux plateformes –, quelques services de gestion de projet, de génération et de suivi de tickets.

Sur Jira nous aurions pu découper le travail en épique, User Stories, taches techniques