

Examen *DevOps & Continuous Testing*

- Date: 18 mai 2021
- Durée: 3 heures
- **Aucun document autorisé; barème donné à titre indicatif.**

Vous devez répondre aux questions posées dans le sujet sur la copie d'examen fournie, avec d'éventuelles feuilles intercalaires. Vous ne pouvez pas sortir de la salle d'examen durant la première heure, ni durant le dernier quart d'heure. Lisez bien tout le sujet avant de commencer à répondre aux questions. Soyez synthétique dans vos réponses (quitte à utiliser des listes à puces), nous n'attendons pas un roman mais une analyse pertinente et précise. N'hésitez pas à mobiliser des compétences acquises dans d'autres cours pour répondre aux questions d'un point de vue global. Vous pouvez rédiger en français ou en anglais, à la seule condition que l'intégralité de la copie soit dans la même langue.

Toute fraude identifiée sera transmise au conseil de discipline de l'UNS.

Partie 1: Question de recul (5 points)

Durant le mois de mars 2016 a eu lieu l'affaire du "*NPM Gate*". NPM est le gestionnaire de *package* de référence dans l'écosystème Javascript (4 millions d'utilisateurs). Suite à une problématique légale sur le nom d'un de ses *packages* (une entreprise de même nom en réclamant la paternité), un développeur a retiré les 250 *packages* qu'il avait publié dans NPM (aka "*rage quit*"). Or, un de ses *packages* (`left-pad`, faisant approximativement 15 lignes de codes permettant d'aligner à gauche du texte) était utilisé en moyenne 2,486,696 de fois par mois par d'autres *packages* (directement ou indirectement). Cette action a eu pour effet de faire s'effondrer tout l'écosystème NPM jusqu'à ce que les gestionnaires de NPM décident de re-publier le *package* en question sans l'accord de son auteur pour remettre sur pied leur système.

Les moyens mis en oeuvre par NPM pour que la situation ne se reproduisent pas lors du retrait d'un *package* sont les suivants¹ :

- *If the version is less than 24 hours old, you can unpublish it. The package will be completely removed from the registry. No new packages can be published using the same name and version.*
- *If the version is older than 24 hours, then the unpublish will fail, with a message to contact support@npmjs.com.*
- *If you contact support, they will check to see if removing that version of your package would break any other installs. If so, we will not remove it. You'll either have to transfer ownership of the package or reach out to the owners of dependent packages to change their dependency.*

¹ <http://blog.npmjs.org/post/141905368000/changes-to-npm-unpublish-policy> (politique officielle)

- *If every version of a package is removed, it will be replaced with a security placeholder package, so that the formerly used name will not be susceptible to malicious squatting.*
- *If another member of the community wishes to publish a package with the same name as a security placeholder, they'll need to contact support@npmjs.com. npm will determine whether to grant this request. (Generally, we will.)*

Que pensez-vous de l'affaire du "NPM Gate", et des moyens mis en oeuvre par NPM pour y remédier (argumentez votre réponse) ?

Partie 2: Docker Compose (3 points)

Durant les TDs, vous avez eu l'occasion de construire des images Docker, ainsi que d'expérimenter avec Docker Compose afin d'automatiser le lancement de plusieurs containers différents simultanément.

Quels autres bénéfices peut-on tirer de Docker Compose ?

Quels sont les fonctionnalités que n'apportent pas Docker Compose, mais que des outils comme Docker Swarm ou Kubernetes peuvent couvrir quand on souhaite construire des topologies basées sur des images Docker ?

Partie 3: Étude de cas (12 points)

Une fois leur diplôme en poche, Emma et Fabien montent une startup basée sur un concept simple: un réseau social qui respecte la nouvelle législation Européenne GDPR (General Data Protection Regulation) qui couvre le respect de la vie privée et régule le stockage d'informations privées. Bien sur, pour un réseau social, ils hébergent des informations privées, mais ils garantissent à leurs usagers de ne pas les transmettre à leurs annonceurs. Ils commencent donc par écrire le code du serveur, du système de stockage, des algorithmes divers, et stockent tout cela sur le serveur de l'un d'entre eux.

Comme ils ne sont pas très forts pour la partie graphique, ils s'associent avec José et Liliane, deux étudiants de l'université de Lille, à qui ils donnent des accès sur leur serveur.

Q1: décrivez une infrastructure de génie logiciel qui permette à ces 4 personnes de travailler ensemble, malgré la distance, et bien qu'ils aient choisi de diviser le travail en couches.

Afin que José et Liliane puissent travailler efficacement, Emma et Fabien leur livre une version du serveur toutes les semaines, censée offrir les fonctionnalités à implémenter dans la couche graphique la semaine suivante.

Q2: comment peuvent-ils garantir que ce mécanisme peut fonctionner, notamment sans que les fonctionnalités précédemment développées ne cessent de marcher ?

Le concept plaît, leur startup se fait racheter par un grand groupe, Seguin, qui a déjà en place des équipes de développement en France, mais aussi en Inde. Ces équipes sont organisées pour couvrir chacune une fonctionnalité entièrement, à travers toutes les couches concernées. Il faut donc revoir la componentisation du code.

Q3: quelle approche proposez-vous pour refactoriser le code en garantissant l'intégrité fonctionnelle ? A quels aspects en particulier faut-il prêter attention ?

Une fois ceci fait, le groupe Seguin alloue 30 personnes, réparties sur les deux pays, au développement de ce nouveau produit. Le groupe souhaite pouvoir mettre une version sur le marché par semaine, ce qui implique un bon niveau de confiance dans l'usine logicielle à mettre en place.

Q4: décrivez comment vous organiseriez cette usine logicielle, en abordant notamment la gestion de code source, les tests et leur stratégie, l'intégration continue et le déploiement.

Q5: comment pourriez-vous mesurer la qualité du code produit ?