

CERTIFICATION PROFESSIONNELLE EXPERT EN INFORMATIQUE ET SYSTEME D'INFORMATION

BLOC 2 – Manager un projet informatique avec agilité en collaboration avec les parties prenantes

Cahier des Charges de la MSPR « Cahier des Charges de la MSPR «gérer un projet de développement serverless selon les principes agile dans un environnement multiculturel »

COMPÉTENCES ÉVALUÉES :

- Identifier l'ensemble des étapes de réalisation du système d'information pour organiser le projet en tâches et livrables en répartissant les activités en fonction des ressources humaines, techniques et financières à mobiliser.
- Concevoir les cahiers des charges technique et fonctionnel d'un projet de développement S.I. à l'aide des besoins utilisateurs collectés afin de cadrer le développement
- Gérer un projet agile en utilisant les méthodes et outils adaptés à ce mode de fonctionnement pour tester, modifier et procéder par itération afin de réduire les délais de remise des projets de développement S.I.
- Etablir des tableaux de bord de suivi de performance (qualitative et quantitative) de l'ensemble des ressources allouées à chaque étape-projet pour anticiper, visualiser et corriger les écarts en temps réel afin de limiter les contraintes de ressources et les retards dans la réalisation du projet.
- Piloter les prestataires extérieurs éventuels gérant les ressources informatiques d'un système d'information existant listées dans la cartographie établie afin de sécuriser la mise en œuvre technique.
- Conduire une équipe projet en diffusant les fondamentaux de l'agilité : adaptation, flexibilité et amélioration continue au sein de l'équipe afin d'être en mesure d'absorber les changements de priorité qui peuvent intervenir dans un contexte de forte contrainte de temps et d'incertitudes.
- Adopter une stratégie d'accueil aux handicaps afin de favoriser l'inclusion des profils en situation de handicap au sein de l'équipe et permettre leur pleine intégration, en collaboration avec le référent handicap de l'entreprise.
- Communiquer avec l'équipe en adoptant les modes de communication adéquats selon les cultures et la langue des collaborateurs afin de garantir l'intégration de tous les membres de l'équipe.
- Proposer des solutions innovantes afin de favoriser les interactions au sein de l'équipe et d'anticiper des conflits de travail liés aux malentendus multiculturels.
- Concevoir un processus de communication inclusif régulier au sein de l'équipe afin de synchroniser les activités quotidiennes et mettre en place un fil de discussion à l'aide d'outils numériques.
- Animer des réunions à distance afin de maintenir une dynamique de groupe et renforcer l'esprit d'équipe des membres en télétravail et/ou à distance.
- Concevoir un processus de partage d'information afin de faciliter l'inclusion et la collaboration entre les membres en télétravail et/ou à distance en utilisant des outils numériques.
- Accompagner l'équipe dans l'appropriation du travail à distance ou du télétravail en proposant des solutions managériales afin de favoriser la motivation et la résilience et permettre ainsi de préserver équilibre entre vie professionnelle/vie privée dans un souci de productivité et de bien-être

PHASE 1 : PRÉPARATION DE CETTE MISE EN SITUATION PROFESSIONNELLE RECONSTITUÉE

- Durée de préparation :
 - 19 heures
- Mise en œuvre :
 - Travail d'équipe constituée de 4 apprenants-candidats (5 maximum si groupe impair)
- Résultat attendu :
 - Un projet détaillé et organisé, et une infrastructure mise en place (Kubernetes + un applicatif déployés) en utilisant les technologies d'Infrastructure As Code. Les apprenants devront par ailleurs s'assurer d'être en mesure d'expliquer comment ils justifient la mise en œuvre des compétences citées plus haut et dans le cas où elles n'ont pu être mise en œuvre, le justifier et démontrer leur capacité à la mettre en œuvre dans un contexte professionnel.

PHASE 2 : PRÉSENTATION ORALE COLLECTIVE + ENTRETIEN COLLECTIF

- **Durée totale par groupe** : 50 mn se décomposant comme suit :
 - 20 mn de soutenance orale par équipe
 - 30 mn d'entretien collectif avec le jury (questionnement complémentaire)
 - 10 mn de délibération.
- **Objectif** : mettre en avant et démontrer que les compétences visées par ce bloc sont bien acquises.
- **Jury d'évaluation** : 2 personnes par jury – Ces évaluateurs sont des professionnels experts du domaine. Les membres du jury ne sont pas intervenus durant la période de formation et ne connaissent pas les apprenants à évaluer.

I - INTRODUCTION

Bien que le sujet principal de cette MSPR soit la gestion de projet, le sujet vous permet également de mettre en œuvre un ensemble de compétences métier plus techniques liées à vos options choisies. Aussi en complément des compétences listées en première partie du sujet, celui-ci vous permettra de mettre en valeur les compétences suivantes :

- Gérer un projet de développement d'une application Serverless utilisant OpenFaaS Community, reposant sur un cluster Kubernetes, et interagissant avec une base de données (PostgreSQL, MariaDB, SQL Server).
 - Planifier le développement d'un ensemble de fonctions Serverless et d'une frontend simple.
 - À chaque étape du projet, la diviser en tâches "atomiques", bien penser à répartir ces tâches, estimer leur temps de réalisation via un diagramme de Gantt, et effectuer un suivi régulier de l'avancement de chaque tâche (grâce à un Kanban par exemple), en utilisant la méthode Agile.
 - Choisir les technologies à utiliser pour la mise en place d'un cluster Kubernetes (soit via GKE/AKS/EKS, soit "baremetal")
 - Recommandation solution baremetal: utilisez une distribution Kubernetes simple à installer telle que K3S, K0S, RKE2, ou microK8S. Idéalement votre cluster se composera d'une VM control-plane, et d'une VM Worker. OpenFaaS est relativement léger dans son déploiement par défaut, votre control-plane pourra se satisfaire de 2vCPUs et 2Go de RAM, et votre worker de 2vCPUs et 2 à 4Go de RAM.
 - Recommandation solution cloud: Vérifiez si les partenariats de votre campus vous permettent d'accéder à des ressources cloud gratuitement ou à moindre coût.
 - En cas d'impossibilité de créer un cluster Kubernetes, il est possible d'exécuter OpenFaaS dans un environnement Docker (faasd). Il est aussi possible d'utiliser minikube, ou KinD, pour simuler votre cluster. Mais un vrai cluster Kube, c'est mieux 😊
 - Être capable de créer des fonctions Serverless (dans le langage de votre choix (Python recommandé)) correspondant au besoin du client, capables d'interagir avec une base de données.
 - Être capable de créer une frontend (simple, elle doit juste permettre d'appeler les fonctions demandées par le client et de vérifier leur bon fonctionnement) en accord avec le cahier des charges client.
 - Être capable de créer les images des conteneurs de vos fonctions OpenFaaS et les "uploader" sur un dépôt d'images OCI (Docker Hub, Github Container Repository, Quay.io, ou auto-hébergé (Docker registry, Harbor))
 - Être capable de déployer OpenFaaS, votre frontend et une base de données (simple) sur votre cluster Kubernetes (ou dans votre environnement Docker/minikube), et installez OpenFaaS CLI sur vos postes de travail.
 - Enfin, expliquer vos choix en fonction de l'expression du besoin initial, quelles difficultés avez vous rencontrées, et quelles solutions avez vous pu y apporter.

II - PRÉSENTATION DE L'ENTREPRISE ET EXPRESSION DU BESOIN.

La COFRAP (Compagnie Française de Réalisation d'Applicatifs Professionnels), concurrent de la COGIP dans le secteur des applicatifs Web de gestion d'entreprise (ERP, groupware, etc), est une entreprise proposant à ses clients des applicatifs de gestion complexes et puissants, reconnue dans le monde entier pour la qualité de ses services.

Elle propose à ses clients soit d'héberger sur leur infrastructure, soit d'héberger sur l'infrastructure cloud de la COFRAP, ses applicatifs.

Concernant sa partie cloud, la COFRAP a décidé de remanier le processus de création des comptes utilisateurs, dont les informations d'identification doivent désormais être créées de façon automatique. En effet, les utilisateurs de la solution Cloud avaient tendance à utiliser des mots de passes trop simples, et à ne pas utiliser la double authentification proposée (2FA de type time-based password), ce qui a conduit à de nombreuses compromission de comptes.

Désormais, un utilisateur créant un compte se verra attribué un mot de passe généré automatiquement qui lui sera transmis à la création via un QRCode à usage unique, et devra obligatoirement utiliser la 2FA générée dans la foulée pour voir son compte activé. Une rotation des mots de passe et des tokens 2FA est prévue tous les six mois. La complexité du mot de passe est fixée à 24 caractères, et un mot de passe doit comporter majuscules, minuscules, chiffres et caractères spéciaux.

Votre équipe a été désignée par la direction de la COFRAP pour mettre en place ce système. Il a été décidé d'utiliser la technologie OpenFaaS reposant sur Kubernetes pour déployer les fonctions nécessaires à la réalisation de cet objectif. L'avantage ici étant la fonctionnalité "Scale to Zero" d'OpenFaaS Entreprise, qui permettra des économies d'échelle en réduisant le nombre d'instances de l'application à zéro lors des périodes d'inactivité.

Fonctions à préparer:

- Une fonction générant un mot de passe pour un compte utilisateur spécifié en paramètres de la fonction, dont la complexité est fixe (24 caractères, majuscules/minuscules/chiffres/caractères spéciaux), et générant un qrcode à partir de ce mot de passe, stockant l'identifiant utilisateur et ce mot de passe (en le chiffrant!) dans votre base de données.
- Une fonction générant un secret 2FA et le qrcode correspondant pour le compte utilisateur demandé en paramètres de la fonction, et stockant cette information (en la chiffrant!) en base de données.
- Une fonction authentifiant un utilisateur à partir de son login, son mot de passe, et son code 2FA, après avoir vérifié que ces identifiants ont moins de six mois d'ancienneté, sinon elle doit marqué le compte comme "expiré" en base de données, et renvoyer une réponse à la frontend relançant le processus de création de mot de passe et de 2FA.

Enfin, une frontend (simple), doit permettre d'authentifier un utilisateur, ou de le créer s'il n'existe pas (en suivant le process décrit plus haut), ou de relancer le processus de création de mot de passe et de 2FA si son login et son mot de passe sont expirés. Une autre équipe est en charge de sécuriser cette solution afin d'éviter les abus typiques (création de comptes en boucle/spammeurs/etc).

Pour le moment, la COFRAP vous a demandé de réaliser un PoC (Proof of Concept) de cette solution, soit sur un petit cluster Kubernetes (KinD, K3S ou cloud), soit via minikube ou Docker (minikube préféré si vous ne pouvez pas, ou n'avez pas le temps, de mettre en place un cluster Kubernetes)

Concernant la base de données, vous pouvez utiliser un Statefulset Kubernetes, une VM dédiée ou un conteneur docker dédié. La technologie utilisée est à votre discrétion: PostgreSQL, MariaDB, MongoDB, etc. Votre base de données ne devrait contenir qu'une seule table pour stocker les informations de vos utilisateurs. La table elle-même devrait être très simple, du type:

ID	username	password	MFA	gendate	expired
1	michel.ranu	AAAA...GGGG==	AAAA...BBBB=	1721916574	0

Le langage de programmation à utiliser pour préparer vos fonctions est, là aussi, à votre discrétion, cependant, Python est fortement recommandé par la COFRAP, qui l'utilise déjà dans la plupart de ses projets (traduction: vous trouverez toutes les bibliothèques de fonctions nécessaires assez facilement avec Python 😊).

III-LISTING DES OBJECTIFS :

- **Mission 1:** choix des technologies et justification des choix
- **Mission 2:** Organisation du projet: tâches nécessaires à sa réalisation et estimation du temps nécessaire à sa réalisation
 - Sous-tâche: Réfléchissez à la création d'un environnement de travail inclusif et respectueux.
- **Mission 3:** Suivi de l'évolution de la réalisation des tâches définies en mission 2.
- **Mission 4:** Préparation de l'environnement Kubernetes d'exécution d'OpenFaaS
- **Mission 5:** Déploiement d'OpenFaaS Community sur votre cluster Kubernetes
- **Mission 6:** Déploiement de vos fonctions OpenFaaS et de votre frontend.
- **Mission 7:** Préparation d'une frontend de démonstration.
- **Mission 8:** dossier de rendu final.

VI - CAHIER DES CHARGES DÉTAILLÉ

Mission 1: choix des technologies et justification des choix

- Expliquez les raisons de vos choix concernant le langage de programmation choisi, l'utilisation de tel ou tel fournisseur de cloud (ou "distribution" Kubernetes). En cas de solution "baremetal", nous vous recommandons l'utilisation de K3S, du fait de sa simplicité de mise en œuvre. N'oublions pas que le projet est censé fournir un "Proof of Concept" à notre entreprise, son implémentation finale sera réalisée par les équipes Infrastructure de l'entreprise.

Mission 2: Organisation du projet: tâches nécessaires à sa réalisation et estimation du temps nécessaire à sa réalisation

- Définissez les différentes tâches nécessaires à la réalisation de votre projet, et estimez le temps nécessaire à leur réalisation. Reportez ces informations sur un diagramme de Gantt.
- Prenez aussi le temps de réfléchir aux mesures que vous mettriez en place afin d'assurer un environnement de travail inclusif, tolérant et respectueux des sensibilités de chacun·e. Par exemple, comment accueilleriez vous au sein du projet une personne en situation de handicap visuel, et comment adapteriez-vous sa charge de travail à sa situation.

Exemples d'outils open-source de gestion de projet:

- [Kanboard](#)
- [OpenProject](#)
- [Redmine](#)

Mission 3: Suivi de l'évolution de la réalisation des tâches définies en mission 2.

- Ici, vous voudrez suivre régulièrement l'avancement des tâches attribuées à chacun·e. Généralement, un tableau de type Kanban permet un suivi efficace et agréable de chaque tâche.
- Une organisation typique des colonnes de votre kanban serait du type:
- À faire -> En cours -> Revue Technique -> Terminé
- Il est important que la revue technique soit suivie par tous·tes. En effet, cela peut permettre de découvrir des oublis et erreurs parfaitement humains, qui seraient passés au travers sans cette étape.

Mission 4: Préparation de l'environnement Kubernetes d'exécution d'OpenFaaS

- La mise en place de Kubernetes ne doit donc pas être un frein à l'avancement de votre projet. Si vous ne pouvez pas mettre en place un cluster Kubernetes (soit dans le cloud, soit sur des VMs du laboratoire de votre campus, soit sur des VMs de votre propre laboratoire), vous pouvez utiliser une VM Linux (Debian, Ubuntu, etc) et minikube, qui vous permettra de simuler facilement et rapidement un cluster Kubernetes. Comptez 2 coeurs, 4Go de RAM, 20Go de disque pour votre VM pour pouvoir travailler dans de bonnes conditions.
- Cependant, idéalement votre applicatif s'exécutera sur un environnement Kubernetes réel, soit dans le cloud, soit sur le labo de votre campus, soit sur votre laboratoire personnel ("homelab"), soit sur un laptop.

Pour les solutions cloud, pensez à vérifier si votre campus ne dispose pas d'un partenariat avec un fournisseur de cloud, pour pouvoir utiliser, par exemple:

- Google Cloud GKE
- Microsoft Azure AKS
- AWS EKS
- Scaleway Kapsule

Si vous décidez de vous orienter vers une solution "baremetal", les configurations requises seront:

- Control-plane Kubernetes: 2 coeurs, 2Go RAM minimum (3 à 4Go recommandés), 15Go de disque
- Worker: 2 coeurs, 4Go de RAM, 20Go de disque

afin de pouvoir travailler dans de bonnes conditions. Si vous avez la possibilité de créer deux workers, c'est encore mieux!

Mission 5: Déploiement d'OpenFaaS Community sur votre cluster Kubernetes

- Pour le déploiement d'OpenFaaS Community sur votre cluster, nous vous recommandons d'utiliser le gestionnaire de déploiement Helm, qui automatise une bonne partie de la mise en place d'applicatifs complexes sur Kubernetes.
- Quelques liens pour vous aider:
 - [Installez Helm](#)
 - Suivant la solution utilisée, vous pourriez avoir besoin de déployer un LoadBalancer sur votre cluster, par exemple [MetalLB](#). K3S inclut déjà un LoadBalancer simple (svclb), et assigne déjà un service de type LoadBalancer à son Ingress par défaut (Traefik). Les fournisseurs de cloud vous fourniront déjà un LoadBalancer.
- Vous pourriez avoir besoin d'installer un Ingress, par exemple [Ingress-nginx](#), afin d'exposer OpenFaaS à l'extérieur de votre cluster.
- Déployez OpenFaaS via Helm en utilisant la chart [FaaSnetes](#)

Mission 6: Création, Déploiement et tests de vos fonctions OpenFaaS :

- Le langage de programmation recommandé est Python, cependant, vous êtes libres du choix du langage à utiliser.
- Les fonctions nécessaires sont décrites en début de ce document, dans la partie "Expression du besoin". Bien évidemment, vous êtes libres de les découper en fonctions atomiques plus simples.
- Ensuite, en utilisant la CLI d'OpenFaaS, vous pourrez construire et téléverser les images de vos containers sur un registre public ou privé de conteneurs, par exemple Docker Hub ou Github Container Repository.
- N'oubliez pas que ces fonctions doivent pouvoir écrire dans une base de données, que vous pouvez héberger sur le cluster Kubernetes que vous aurez créé, sur une autre VM, ou dans le cloud. Nous vous recommandons MariaDB ou PostgreSQL, mais vous êtes, là aussi, libres du choix de la base de données à utiliser. Dès lors, pensez à vous renseigner dans la documentation d'OpenFaaS sur la gestion des secrets (login et mot de passe de base de données, etc)

Mission 7: Préparation d'une frontend de démonstration.

- Vous êtes libres des technologies et langages de programmation utilisés pour préparer votre frontend de démonstration. N'oubliez pas qu'elle doit être simple: ses seuls buts sont la création de compte utilisateur, la génération de mot de passe (et qrcode), la génération d'un secret TOTP/2FA, l'authentification de l'utilisateur et la validation/renouvellement du mot de passe et du secret TOTP via les fonctions OpenFaaS que vous aurez précédemment créées.

Mission 8: dossier de rendu final :

- Votre dossier final devra contenir, en quelques pages (environ 20 pages), le résumé de toutes vos missions, screenshots à l'appui. Vous fournirez en annexes le code de votre frontend et de vos fonctions OpenFaaS.
- De même, il est attendu que vous puissiez faire la démonstration de l'organisation de votre projet: Diagramme de Gantt, Kanban

V - LIVRABLES ATTENDUS

- Votre dossier de rendu final.
- L'organisation de votre projet (Gantt, Kanban)
 - N'oubliez pas d'expliquer les différentes pistes de réflexion que vous avez explorées concernant la mise en place d'un environnement de travail multiculturel et inclusif.
- Une démonstration de votre applicatif fonctionnel.
- Une explication des différentes fonctions que vous avez mises en place sur OpenFaaS, leur rôle, et les raisons de vos choix.

Les compétences évaluées durant cette MSPR :

- Identifier l'ensemble des étapes de réalisation du système d'information pour organiser le projet en tâches et livrables en répartissant les activités en fonction des ressources humaines, techniques et financières à mobiliser.
- Concevoir les cahiers des charges technique et fonctionnel d'un projet de développement S.I. à l'aide des besoins utilisateurs collectés afin de cadrer le développement
- Gérer un projet agile en utilisant les méthodes et outils adaptés à ce mode de fonctionnement pour tester, modifier et procéder par itération afin de réduire les délais de remise des projets de développement S.I.
- Etablir des tableaux de bord de suivi de performance (qualitative et quantitative) de l'ensemble des ressources allouées à chaque étape-projet pour anticiper, visualiser et corriger les écarts en temps réel afin de limiter les contraintes de ressources et les retards dans la réalisation du projet.
- Piloter les prestataires extérieurs éventuels gérant les ressources informatiques d'un système d'information existant listées dans la cartographie établie afin de sécuriser la mise en œuvre technique.
- Conduire une équipe projet en diffusant les fondamentaux de l'agilité : adaptation, flexibilité et amélioration continue au sein de l'équipe afin d'être en mesure d'absorber les changements de priorité qui peuvent intervenir dans un contexte de forte contrainte de temps et d'incertitudes.
- Adopter une stratégie d'accueil aux handicaps afin de favoriser l'inclusion des profils en situation de handicap au sein de l'équipe et permettre leur pleine intégration, en collaboration avec le référent handicap de l'entreprise.
- Communiquer avec l'équipe en adoptant les modes de communication adéquats selon les cultures et la langue des collaborateurs afin de garantir l'intégration de tous les membres de l'équipe.
- Proposer des solutions innovantes afin de favoriser les interactions au sein de l'équipe et d'anticiper des conflits de travail liés aux malentendus multiculturels.
- Concevoir un processus de communication inclusif régulier au sein de l'équipe afin de synchroniser les activités quotidiennes et mettre en place un fil de discussion à l'aide d'outils numériques.
- Animer des réunions à distance afin de maintenir une dynamique de groupe et renforcer l'esprit d'équipe des membres en télétravail et/ou à distance.
- Concevoir un processus de partage d'information afin de faciliter l'inclusion et la collaboration entre les membres en télétravail et/ou à distance en utilisant des outils numériques.
- Accompagner l'équipe dans l'appropriation du travail à distance ou du télétravail en proposant des solutions managériales afin de favoriser la motivation et la résilience et permettre ainsi de préserver équilibre entre vie professionnelle/vie privée dans un souci de productivité et de bien-être