

Déploiement Avancé

Lancement de Votre Startup S4-DACS-02

Bienvenue dans le jeu réel

Vous êtes les fondateurs techniques d'une jeune startup. Votre mission est de livrer un MVP (Minimum Viable Product) fonctionnel, conteneurisé, et prêt à être présenté à des investisseurs, partenaires ou bêta-testeurs. Ce MVP devra être pensé comme un socle solide, reproductible et déployable rapidement. Il incarnera votre première itération produit, démontrant à la fois la valeur de votre idée et votre capacité à exécuter techniquement. Vous travaillez en binôme, en autonomie, avec des points de contact techniques réguliers. Ce que vous allez produire est un vrai livrable professionnel.

Votre mission

1. Imaginer une idée de business numérique, aussi simple soit-elle, mais avec une proposition de valeur claire.
2. Développer une version fonctionnelle de cette idée, même minimale, qui puisse être testée en conditions réelles.
3. Structurer votre stack avec Docker et Docker Compose, pour que n'importe qui puisse lancer votre projet en une commande.
4. Organiser votre dépôt GitHub de façon propre, documentée et lisible.
5. Préparer votre projet pour le monde réel : déploiement local, cloud, installation reproductible, et documentation claire.
6. Bonus: Mettre en place un pipeline d'intégration continue (CI/CD) simple avec GitHub Actions. Il devra au minimum vérifier que le projet se build correctement ('docker-compose build') et, si possible, lancer des tests automatiques. Ce pipeline simulera les pratiques de validation continues que l'on trouve en entreprise, et permettra de s'assurer que chaque modification du code ne casse pas l'application.
7. Déployer le projet dans le cloud (obligatoire sauf contrainte justifiée), via une plateforme simple comme Render, Railway, Fly.io ou un petit VPS. L'objectif est de rendre le projet accessible en ligne.

Contraintes techniques

Votre projet devra inclure au minimum :

- Une interface (frontend web ou ligne de commande) que l'utilisateur peut utiliser.
- Une API ou un backend logique.
- Un système de persistance (base de données, fichiers, etc.), si votre produit le nécessite.

L'ensemble doit être conteneurisé. Aucune installation manuelle de dépendances ne doit être nécessaire pour lancer le projet : **docker-compose up** doit suffire à démarrer l'environnement de développement complet.

En plus de cela, certaines bonnes pratiques de sécurité devront être respectées :

- Tous les fichiers contenant des secrets, mots de passe, ou configurations sensibles doivent être exclus du dépôt Git grâce à un **.gitignore** bien configuré (par exemple : les fichiers **.env** ne doivent jamais être versionnés).
- Dans les fichiers Docker, évitez autant que possible d'utiliser l'utilisateur **root** pour exécuter les processus applicatifs. Préférez la création et l'utilisation d'un utilisateur non-privilégié.
- Les ports exposés dans les fichiers de configuration doivent être explicites, cohérents et justifiés.
- Toute variable sensible (clé API, mot de passe, etc.) doit être injectée par une variable d'environnement et non codée en dur dans le code source.

Le choix des technologies est libre, tant que vous maîtrisez ce que vous utilisez. Vous êtes responsables de vos choix d'architecture, de sécurité et de qualité.

Livrables attendus

À la fin du projet, vous devez avoir :

- Un dépôt GitHub public contenant :
 - Le code source de votre projet (frontend, backend, scripts, etc.).
 - Les fichiers Docker et Docker Compose fonctionnels.
 - Un README qui décrit : l'idée, comment lancer le projet, comment l'utiliser, les choix techniques, et comment contribuer.
 - Un fichier de configuration GitHub Actions (`.github/workflows/`) permettant d'exécuter automatiquement le build et les tests à chaque push.
- Une démonstration en direct de votre produit, lancée en local via Docker.
- Une version déployée dans le cloud, accessible publiquement, avec des instructions de déploiement précises dans le README. Si un déploiement dans le cloud n'est pas possible (manque de ressources, contraintes techniques), vous devrez l'expliquer et proposer une alternative crédible.
- Un document de synthèse (PDF ou Markdown) qui raconte l'histoire de votre projet : l'idée, les choix faits, les obstacles rencontrés, et les améliorations futures envisagées.

Critères de réussite

Vous serez évalués comme une véritable équipe technique livrant un produit prêt à être utilisé et partagé. Les points suivants seront pris en compte :

- Le projet fonctionne de bout en bout, dans les conditions prévues.
- Le code est structuré, lisible, maintenable et organisé de manière professionnelle.
- La conteneurisation est correctement mise en œuvre : tous les services s'exécutent via Docker Compose sans dépendance externe manuelle.
- Le produit répond de manière cohérente à la promesse initiale, même sous une forme minimale.
- Le dépôt GitHub est clair, bien documenté, avec un README complet et des instructions précises.
- Le projet est déployé en ligne (cloud public ou VPS), sauf justification technique claire.
- Des bonnes pratiques de sécurité basiques sont appliquées : fichiers sensibles non commités, utilisateurs non-root dans les conteneurs, ports explicitement définis, secrets gérés proprement.
- L'équipe a fait preuve d'autonomie, de rigueur technique et d'esprit d'initiative dans ses choix et dans la gestion du projet.
- Un pipeline CI/CD simple est mis en place pour automatiser au minimum le build et la vérification du projet (Bonus: +1pts).
- Si toutes les consignes du projet ont été rigoureusement respectées, une fonctionnalité supplémentaire originale — c'est-à-dire non présente dans les autres projets — et jugée pertinente pourra donner lieu à un bonus de +1 point.
- Un bonus/malus de classement sera appliqué à la note finale : +2 pour la meilleure équipe, +1 pour la deuxième, +0.5 pour la troisième, 0 pour la quatrième, et -1 pour la dernière équipe.

Conseil aux fondateurs

Faites simple, mais solide. Évitez les ambitions techniques complexes qui ne mènent à rien. Commencez petit, livrez vite, améliorez ensuite. Ne cherchez pas à impressionner, cherchez à résoudre un problème, même minime, de manière concrète. La plupart des startups échouent non parce que leur idée est mauvaise, mais parce qu'elles ne livrent rien. Soyez différents : livrez.