

- **Agence Furious Ducks**
- **Date de début de projet: 12/05/2024**
- **Version actuelle: 1.0**
- **Rédacteurs:**
- Mickael
- Noah
- Hermes
- **Mentions juridiques:**

Furious Ducks SARL, 18 rue Léon Frot, 75011 Paris RCS Paris 123 456 789 - SIREN: 123 456 789 - SIRET: 123 456 789 00000 - APE/NAF: 6202A TVA intracommunautaire: FR 12 3456789012

État des lieux

Présentation de l’agence

Furious Ducks, fondée en 2007, est une webagency spécialisée dans les technologies open source. Depuis sa création, l’agence a su se démarquer par son expertise et ses valeurs d'innovation, de transparence, et de collaboration. Avec une équipe de 45 passionnés, elle se consacre à offrir des solutions web de haute qualité à ses clients. Les services de Furious Ducks englobent le développement web, le design UX/UI, et la stratégie digitale, avec un accent particulier sur l'utilisation de technologies open source pour garantir flexibilité et efficacité.

Informations juridiques

- **Dénomination sociale:** Furious Ducks SARL
- **Siège social:** 18 rue Léon Frot, 75011 Paris
- **Statut juridique:** SARL
- **Capital social:** 150 000€
- **RCS Paris:** 123 456 789
- **SIREN:** 123 456 789
- **SIRET:** 123 456 789 00000
- **APE/NAF:** 6202A
- **Numéro de TVA intracommunautaire:** FR 12 3456789012

Présentation de la problématique

Mr. Guido Brasletti, directeur de Furious Ducks, a constaté une désorganisation croissante au sein de l’agence, impactant l’efficacité et la productivité. Les principaux problèmes identifiés sont:

- Absence de versionning
- Manque de procédures de sauvegarde
- Procédures de déploiement inefficaces
- Problèmes de communication entre équipes

Les livrables attendus pour résoudre cette problématique sont:

- Une nouvelle organisation de travail basée sur une méthodologie Agile
- Un workflow de production industrialisé
- Une infrastructure CI/CD complète

Méthodologie de projet actuelle

Actuellement, Furious Ducks utilise une méthodologie de gestion de projet en cycle en V. Cette approche séquentielle présente

plusieurs limitations, notamment:

- Manque de flexibilité face aux changements de spécifications
- Délais prolongés entre la conception et la livraison
- Difficulté à intégrer les retours clients en cours de projet

Méthodologie de projet à venir

- **Préconisation: Méthodologie Agile (Scrum)**

La méthodologie Agile, et plus spécifiquement Scrum, est recommandée pour sa flexibilité et sa capacité à améliorer la collaboration et la communication entre les équipes. Scrum permet des cycles de développement plus courts (sprints), favorisant une livraison continue de petites fonctionnalités et une meilleure réactivité aux changements.

- **Avantages:**
- Réduction des délais de livraison
- Amélioration de la qualité du produit
- Meilleure satisfaction client grâce à des livraisons fréquentes
- **Outils à utiliser :**
- **Trello** pour la gestion des tâches et des sprints

Introduction au mouvement DevOps et DevSecOps

Définition et Utilisation

DevOps est une pratique combinant le développement (Dev) et les opérations (Ops) pour améliorer la collaboration entre les équipes et automatiser les processus de livraison. DevSecOps intègre la sécurité (Sec) dès le début du cycle de développement, assurant que les applications sont sécurisées dès leur conception.

- **Avantages:**
- Réduction des temps de déploiement
- Amélioration de la qualité et de la sécurité du code
- Automatisation des processus répétitifs
- **Topologies DevOps retenues:**
- **Structure d'équipe intégrée:** Les développeurs, les opérations et les experts en sécurité travaillent ensemble tout au long du cycle de vie du projet.
- **Évolution vers DevSecOps:** Intégration des pratiques de sécurité dans le pipeline DevOps pour garantir des livraisons sécurisées.

Analyse du workflow CI/CD

Présentation Workflow CI/CD

L'intégration continue (CI) et le déploiement continu (CD) sont des pratiques visant à automatiser la construction, les tests et le déploiement des applications. Les tâches à automatiser incluent:

- Gérer l'optimisation des assets comme la minification, la compression, etc.
- . Déployer automatiquement les serveurs de dev
- Stocker les images docker construite par votre CI
- Déployer les serveurs de pré-production automatiquement
- Déployer les serveurs de production (à vous de déterminer le niveau d'automatisation)
- Exécuter des tests unitaires et/ou des tests fonctionnels
- D'envoyer/stocker les rapports de tests

Choix techniques

Services utilisés:

- **Reverse Proxy: Traefik**
 - **Justification:** Traefik est un reverse proxy moderne qui simplifie la gestion du routage des requêtes et offre une intégration native avec Docker, Kubernetes et d'autres orchestrateurs de conteneurs. Il permet une configuration dynamique des routes, ce qui est essentiel pour un environnement CI/CD où les services peuvent être fréquemment ajoutés ou modifiés.
- **SCM: GitLab**
 - **Justification:** GitLab offre une solution SCM complète avec des fonctionnalités intégrées de CI/CD. Son interface est conviviale, et il prend en charge le versioning, les merge requests, et la gestion des pipelines CI/CD. GitLab permet également l'intégration avec des outils de sécurité, facilitant la mise en place d'un workflow DevSecOps.
- **Server CI: Jenkins**
 - **Justification:** Jenkins est une solution CI/CD open source très populaire et flexible. Il supporte une large gamme de plugins pour automatiser les builds, les tests et les déploiements. Jenkins est idéal pour des projets de toute taille et permet de créer des pipelines complexes tout en offrant une intégration facile avec d'autres outils de développement.
- **Métriques: Prometheus et Grafana**
 - **Justification:** Prometheus est une solution de monitoring open source adaptée aux environnements cloud-native. Il collecte des métriques et génère des alertes en temps réel. Grafana, quant à lui, permet de visualiser les données de Prometheus de manière claire et interactive. Ensemble, ces outils permettent un suivi efficace de la performance et de la santé des services CI/CD.
- **Tests: Selenium et JUnit**
 - **Justification:** Selenium est un outil de test d'interface utilisateur (UI) qui permet l'automatisation des tests des applications web. JUnit est un framework de test unitaire pour les applications Java. L'utilisation combinée de Selenium et JUnit permet de couvrir à la fois les tests fonctionnels et unitaires, assurant ainsi la qualité du code avant le déploiement.
- **Backups: Restic**
 - **Justification:** Restic est une solution de sauvegarde open source sécurisée et rapide. Il permet de réaliser des sauvegardes incrémentielles et chiffrées, garantissant la protection des données. Restic est facile à intégrer dans un pipeline CI/CD pour automatiser les sauvegardes régulières des configurations et des données critiques.

Bonnes pratiques et standards:

- **GITFlow:**
 - **Justification:** GITFlow est un workflow de gestion des branches qui aide à structurer les travaux de développement. Il utilise des branches spécifiques pour les fonctionnalités, les releases et les corrections de bugs, facilitant ainsi le processus de versioning et de collaboration entre les développeurs.
- **Standards CI/CD:**
 - **Utilisation des pipelines Jenkins:** Configurer des pipelines Jenkins pour automatiser les builds, les tests et les déploiements. Utiliser des Jenkinsfiles pour définir les étapes du pipeline de manière déclarative.
 - **Intégration continue:** Automatiser les builds et les tests à chaque push de code pour détecter rapidement les problèmes.
 - **Déploiement continu:** Automatiser le déploiement des applications après validation des tests pour réduire les délais de mise en production.

En appliquant ces choix techniques et bonnes pratiques, Furious Ducks sera en mesure de mettre en place un workflow CI/CD efficace, sécurisé et évolutif, répondant aux besoins de l'agence en matière de gestion de projet et de production.

Hébergement et backups

- **Préconisations d'hébergement:**
- **CPU:** 8 vCPUs
- **RAM:** 32 GB
- **Disques:** 1 TB SSD

- **Bande passante:** 1 Gbps
- **SLA:** 99.9%
- **GTR:** 4 heures

Backups:

- **Fréquence:** Quotidienne
- **Rétention:** 30 jours
- **Type:** Full backup avec incremental daily

Diagramme d’infrastructure complet du workflow

(Diagramme à insérer)

Gestion des ressources humaines

Recrutement

- **Fiche de poste:**
- **Profil recherché:** DevOps Engineer
- **Qualifications:** Expérience en CI/CD, maîtrise de Docker et Kubernetes, connaissance des outils de monitoring (Prometheus, Grafana)
- **Contrat de travail:** CDI, 35 heures par semaine
- **Grille de salaire:** 45k€ - 60k€ annuel selon expérience

Formation

- **Programme de formation:**
- **Durée:** 2 semaines
- **Contenu:** Introduction aux outils utilisés (Jenkins, GitLab, Docker), méthodologies Agile, pratiques DevSecOps

Procédures

Diagrammes d’activité

- **Processus de déploiement dev, preprod, et prod:** (Diagrammes BPMN à insérer)
- **Procédure de sauvegarde:** (Diagrammes BPMN à insérer)
- **Procédure de restauration:** (Diagrammes BPMN à insérer)

Diagramme de Gantt

(Diagramme de Gantt à insérer)

Estimation des coûts et rentabilité

- **Coûts de mise en place:**
- **Serveurs:** 10 000€ (initial setup)
- **Humains:** 50 000€ (développement et mise en place)
- **Maintenance annuelle:** 15 000€
- **Rentabilité:**

La mise en place du workflow permettrait une réduction des coûts de développement de 20% et une amélioration de la productivité de 30%, résultant en une économie annuelle estimée à 60 000€.

Annexes

Calcul du TJM des ressources

- **TJM moyen:** 500€/jour
- **Charges patronales et salariales:** 30%
- **Marge de l'agence:** 20%