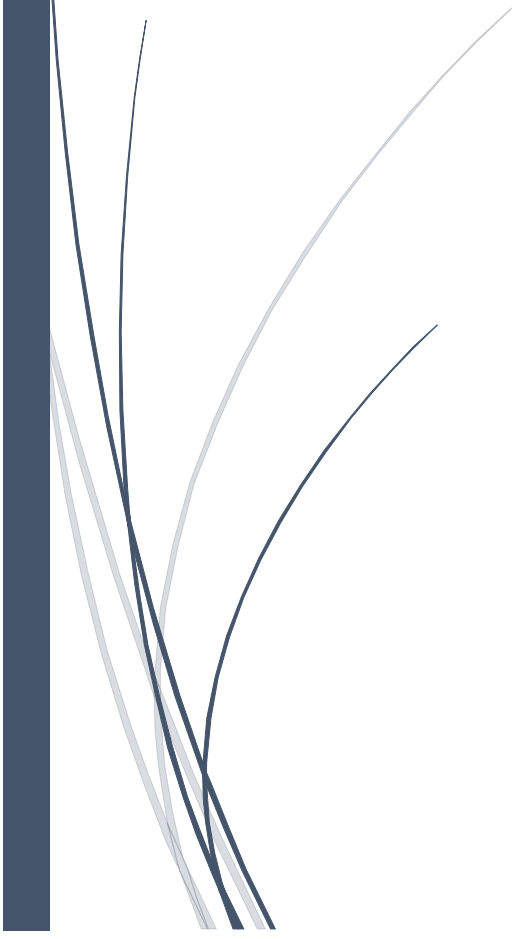
Étude de cas DevOps

Marie CHAP 2023/2024



Marie CHAP
ONSEN

SOMMAIRE

I. Présentation de l'entreprise ONSEN 3
II. Technologie de l'entreprise 4
III. Gitflow de l'entreprise 4
IV. Contexte 5
V. Mes propositions 5
A. Premier test front pour l'application 6
B. Ajout d'un environnement de pre-production : 7
C. Mise en place de veille Informatique 8

I. Présentation de l'entreprise ONSEN

ONSEN est une société-SARL créer en 2008 par Yann Menez. Elle conçoit, fabrique et installe la solution Degré Bleus Eau Chaude une solutions de système de récupération passif d'énergie bien adapté sur les piscines.

Elle possède des automaticiens industriels qui allient des compétences réseau afin de simplifier et rendre moins pénibles les tâches répétitive et opérationnelles en gardant un contrôle humain.

ONSEN assure la boucle complète de la conception à l'installation, assure le suivie de ses produits pour ses clients, notamment par le biais de deux applications appelé Hippocampe, développer pour leur clients.

Ces deux application permettent de suivre le flux de données enregistré par les techniciens ou en télérelève ainsi que suivre les circuits ou bassins en local gérer pas le Degré Bleu Eau Chaude.

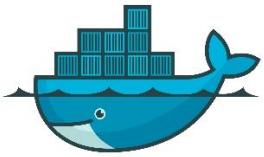
Je vais vous présenter la plateforme Hipoccampe central car c'est l'application sur lequel je suis affectée.

Hippocampe est une plateforme numérique d'optimisation des performances de votre piscine et d'aide à la compréhension et la décision grâce aux bilans, alertes et dashboards. Elle vous permet une vision complète et transparente de votre piscine, de son fonctionnement tout en vous aidant à prendre des mesures concrètes pour réduire vos côuts, votre consommation d'eau d'énergie et votre empreinte carbone.

Elle se décline en 2 modules :

- Hippocampe central, pour la gestion de l'eau, de l'énergie, de l'empreinte C02 et suivi financier.
- Hippocampe Steering, disponible en 2024 pour le pilotage numérique de votre traitement de l'eau.

II. Technologie de l'entreprise



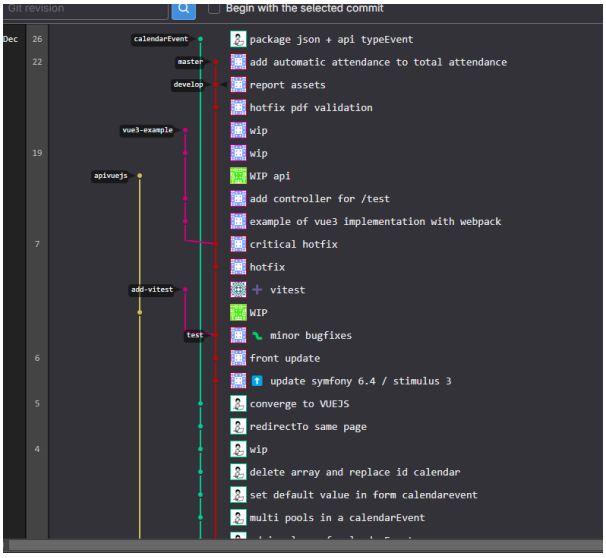
Pour le développement de ses deux applications Hippocampes, on va avoir d'une part une application local « Hippocampe local », cette application est installé sur certaines piscine en local et une application hébergé sur un serveur OVH « Hippocampe central », ou le client peut retrouver par exemple des dashboards personnalisé, des bilans sur les économies réaliser ainsi que quelques configuration pour activé un système d'alerte par e-mail ou sms.

Hippocampe local est développer principalement avec Python et hippocampe centre est développé avec Symfony / php. Toutes deux utilises le même sous-modules pour gérer le front, avec stimulus.js et jquery.

Le DevOps se fait essentiellement sur GitLab, avec Docker, kubernetes, ansible.

III. Gitflow de l'entreprise

L'entreprise se base sur branche commune « develop », celle ou toutes les fonctionnalités provenant d'autres branche seront poussé. Elle sont ensuite mise en production une fois que tout est validé par le développeur en charge du DevOps.



IV. Contexte

- Le frontend est intégré dans l'application Symfony via asset – à chaque modification du code faut pousser le code en back, et ajouter le dossier asset à la modification pour faire matcher le front avec son travail (si on à travailler sur les deux par exemple) , c'est un système de sous-modules.

- Le front est coder en Js, utilisant principalement la bibliothèque en jquery et stimulus. Il y'a d'importante fonctionnalités frontend

- Le frontend est une image Docker dans l'application

En ce qui concerne les environnement de déploiements, nous mettons en production directement une fois que la branche « develop » est validée.

V. Mes propositions

Voici ce que j'ai préconisé à mon entreprise :

- Mise en place d'un serveur de « test » « pré-production »
- L'ajout de test unitaires et intégrations
- code-coverage
- e2e

- Ajouté un système de versionning en ajoutant des tags et une branche pour stocké le versionning
- Mettre en place un système de veille informatique

Onsen dispose d'un serveur de test dans ses locaux, je serai autorisée à l'utiliser pour en faire un environnement de test-preproduction.

En ce qui concerne l'ajout de test sur la partie frontend de l'application, de mon côté j'ai été autorisé à développer un formulaire sur un projet à part en Vue.js.

Ce projet va me servir d'exemple pour le reste de notre frontend, car je prend en main petit à petit l'écriture de fichier de configuration de pipeline, c'est ici ou je placerai mes premiers code-coverage front.

A. Premier test front pour l'application

Voici le premier test unitaires mis en place dans ce projet :

Le test récupère des fonction et vérifie si la fonction me retourne bien ce qui est demandé.

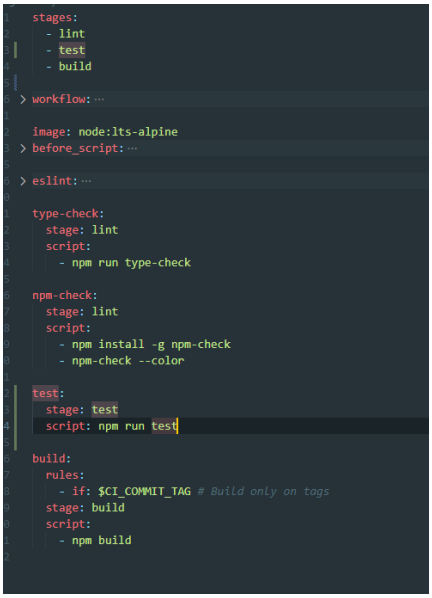
Le premier doit me retourner un nombre d'indicateurs, on peut le lire de cette manière : « cela devrais me retourné le résultats 5 », et le deuxième : « cela devrait me retourner 0 si le tableau de pack est vide ».

Extrait test unitaires avec Vitest :

```
describe('getTotalIndicators function', () => {
    it('returns the total number of indicators', () => {
      const packs = [
        { indics: [1, 2, 3] },
        { indics: [4, 5] },
      ];
      const result = getTotalIndicators(packs);
      expect(result).toBe(5);
    });

    it('returns 0 when packs array is empty', () => {
      const packs: any[] = [];
      const result = getTotalIndicators(packs);
      expect(result).toBe(0);
    });
  });
```

Pipeline du projet : Voici mon pipeline avec l'étape de test ajoutée dessus :

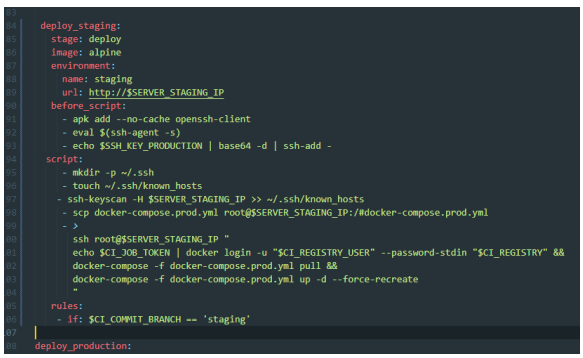


B. Ajout d'un environnement de pre-production :

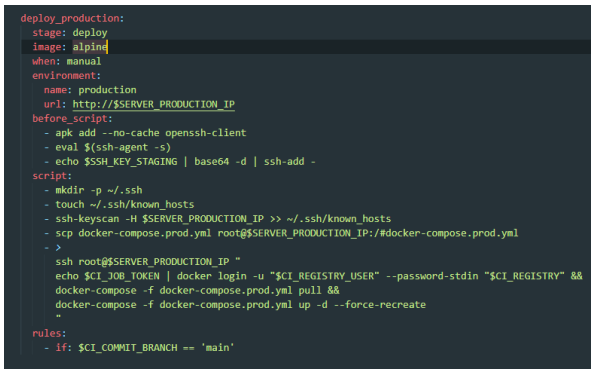
Pour l'ajout d'un environnement de « test » ou « pre-production » chez ONSEN, il faudra que modifie la pipeline de l'application, en lui donnant des instructions de déploiement. J'aimerai que le déploiement se fasse en automatique sur le serveur, car ce sera très utile à mes collègues qui auront besoin de tester de nouvelles fonctionnalités avant de les livrer aux clients, et ensuite je laisserai le déploiement manuel sur la production.

Voici comment cela peut se faire : sur un projet personnel en utilisant des Virtual Private Server, j'ai configurer un fichier gitlab.ci afin de déployer dans 2 environnements différents. Pour ajouter le serveur local, il faudra se connecter sur le VPN du groupe ONSEN et ajouter l'adresse IP au bon endroits comme ce qui suit :

Ici nous avons une instructions qui décrit le déploiement pour un environnement de « staging » (test – pre-prod)



Ici nous avons un exemple de déploiement en production



```yaml
deploy_production:
  stage: deploy
  image: alpine
  when: manual
  environment:
    name: production
    url: http://$SERVER_PRODUCTION_IP
  before_script:
    - apk add --no-cache openssh-client
    - eval $(ssh-agent -s)
    - echo $SSH_KEY_STAGING | base64 -d | ssh-add -
  script:
    - mkdir -p ~/.ssh
    - touch ~/.ssh/known_hosts
    - ssh-keyscan -H $SERVER_PRODUCTION_IP >> ~/.ssh/known_hosts
    - scp docker-compose.prod.yml root@$SERVER_PRODUCTION_IP:/#docker-compose.prod.yml
    - >
      ssh root@$SERVER_PRODUCTION_IP "
      echo $CI_JOB_TOKEN | docker login -u "$CI_REGISTRY_USER" --password-stdin "$CI_REGISTRY" &&
      docker-compose -f docker-compose.prod.yml pull &&
      docker-compose -f docker-compose.prod.yml up -d --force-recreate
      "
  rules:
    - if: $CI_COMMIT_BRANCH == 'main'
```

C. Mise en place de veille Informatique

Nous utilisons Teams, je propose qu'on mette en place un système de veille informatique en utilisant le calendrier de teams ou chaque semaine un développeur de l'équipe aborde un sujet techniques en relation les technologies utilisées par l'entreprise ou les nouveautés en relation avec nos technologies.

Nous pourrions dédié un canal sur Teams spécifique aux technologies.