Integration continue

Le travail sur Docker implique une collaboration étroite pour répartir efficacement les tâches au sein du groupe. Cela nécessite une organisation claire organisation claire et une communication transparente pour garantir la la productivité et le succès du projet.

Gestion de Projet

Planification

Création d'un espace Clickup afin de pouvoir lister et attribuer toutes les tâches au sein du groupe.
https://app.clickup.com/9015092854/v/li/901500538198

Nous avons pu créer un tableau des taches principales à effectué afin de pouvoir faire correspondre le back et le front.

Communication

Pour la communication nous avons utilisé discord afin de pouvoir se convenir de sessions de travail. Nous avons mis en place un bot aussi pour nous prévenir de chaque push afin de pouvoir voir quel était l'avancée du projet.

Réflexion sur les Tests

1 Test Tailwind

Nous avons testé tailwind étant une des des base de notre projet

Cela nous permettait de détecter rapidement les problèmes potentiels d'affichage car cette extension s'occupe s'occupe des affichages du projet.

2 Test Setting page

Grace à ce test la logique était de pouvoir voir plusieurs problème comme la setting page nécessite clerk et prisma cela nous à permis de tester 3 choses en faisant un test que l'accès à la DB de prisma se fasse correctement , que l'affichage de la setting page fonctionne et que l'authentification via Clerk soit fonctionnel.

3 Test Nav-Bar

Pareil que pour le test au dessus , nous nous souhaitions en avoir 2 afin de s'assurer du bon fonctionnement de l'ensemble avec une page plus souvent souvent utilisé par les utilisateurs potentiel. potentiel.

4 Vorcel

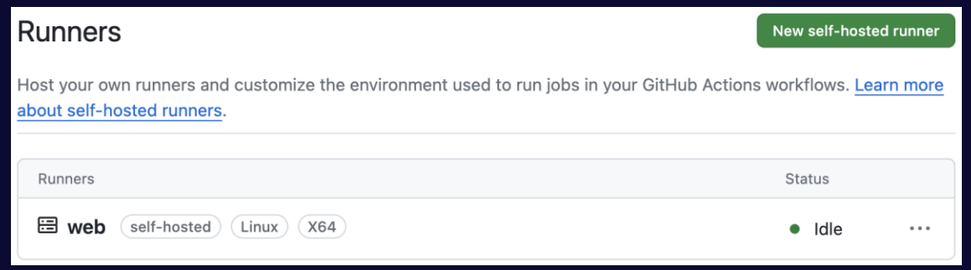
Vorcel s'occupant d'une partie du front il front il était important de pouvoir tester tester cette partie du projet aussi , c'est c'est pour cela que nous avons mis un test test unitaire pour son intégration dans le dans le projet.

Mise en Place de l'Intégration Continue

1 2 3

Intégration d'un serveur

Nous avons créer pour le projet ci/cd un serveur debian que nous avons configurer avec un git-runner runner pour la mise en place place du git. Nous avons donc essayer d'utiliser un serveur Self-Hosted.



Build Automatisé

Bien que nous ayons essayé de faire fonctionner l'ensemble des workflow sur le serveur malheureusement ceux-ci n'ont jamais été fonctionnel , à chaque fois nous avons eu des push fail. Mais nous avons essayé de mettre en place plusieurs Workflow

Déploiement Continu

Ce fut cette partie que nous nous n'avons pas réussi à mettre en place

```
ragondin@web:~/actions-runner$ ./run.sh

√ Connected to GitHub

Current runner version: '2.312.0'
2024-02-04 23:32:09Z: Listening for Jobs
2024-02-04 23:32:55Z: Running job: test
2024-02-04 23:33:20Z: Job test completed with result: Failed
2024-02-04 23:33:22Z: Running job: build
2024-02-04 23:33:40Z: Job build completed with result: Failed
2024-02-04 23:36:29Z: Running job: test
2024-02-04 23:36:41Z: Job test completed with result: Failed
2024-02-04 23:36:44Z: Running job: test
```

Difficultés Rencontrées et Solutions

Investissement

Il y a eu un sérieux manque d'investissement de la part de certaine personne du groupe.

Aucune solution trouvée.

Problème

Nous avons eu pas mal de problème mais avec la console il était possible de voir et corriger au fur et à mesure.

Le CI/CD nous à poser un problème concernant les pipelines , la communication communication avec le serveur la gestion des gestion des dépendances. Nous avons bien bien une communication entre notre serveur serveur mais je ne sais pas ce qu'il nous manque dans notre fichier pour qu'il soit soit pleinement fonctionnel.