

Déploiement

Comment avez-vous effectué le déploiement de votre application ?

APP WEB

Pour le projet Cinéphoria, j'ai choisi de déployer l'application web et le serveur sur Fly.io. Voici comment j'ai procédé :

Choix de Fly.io

J'ai opté pour Fly.io en raison de sa capacité à gérer le déploiement des applications web et serveur de manière simple et efficace, sans nécessiter la création manuelle de conteneurs Docker. Fly.io offre également un support réactif et une documentation bien structurée, facilitant ainsi le processus de déploiement.

Processus de Déploiement

1. Configuration du Fichier `.dockerignore` :

- J'ai créé un fichier `.dockerignore` pour ignorer les dossiers `mobile` et `desktop`, afin de s'assurer que seul le code pertinent pour l'application web et le serveur soit inclus dans le build. Cette étape est cruciale pour optimiser le processus de déploiement et éviter d'inclure des fichiers inutiles dans le conteneur.

2. Automatisation du Build :

- En suivant la documentation de Fly.io, j'ai compris que Fly.io peut automatiquement construire l'application web. Cette fonctionnalité m'a permis de simplifier le processus de déploiement, car je n'ai pas eu à créer manuellement un conteneur Docker. Fly.io a pris en charge la création du build basé sur le code source présent dans le dépôt, j'ai donc lancé les commandes `fly auth login` et `fly launch` depuis mon terminal.

3. Déploiement sur Fly.io :

- Après avoir configuré le fichier `.dockerignore`, j'ai lancé le processus de déploiement sur Fly.io avec la commande `fly deploy`. Fly.io a construit l'application web et le serveur en un seul build et les a déployés ensemble sur la plateforme. Ce processus intégré a assuré que l'application soit déployée de manière cohérente et efficace.

4. Migration des Données de la Base de Données :

- Une fois le build et le déploiement terminés, j'ai exporté les données de ma base de données locale PostgreSQL. Ces données ont ensuite été importées dans l'instance PostgreSQL créée par Fly.io. Cette étape a permis de transférer les données nécessaires à l'application en production, garantissant ainsi que l'application fonctionne avec les données réelles.

Avantages du Déploiement sur Fly.io

- **Automatisation du Build** : La capacité de Fly.io à construire automatiquement l'application web a simplifié le processus de déploiement, réduisant ainsi le besoin de configurations manuelles complexes.
- **Documentation et Support** : La documentation claire et le support réactif de Fly.io ont facilité la compréhension et la mise en œuvre des étapes nécessaires pour le déploiement.
- **Gestion des Données** : L'importation directe des données dans l'instance PostgreSQL de Fly.io a assuré que l'application puisse accéder rapidement et efficacement aux données nécessaires pour fonctionner correctement.

APP MOBILE

Processus de Déploiement

Pour préparer le déploiement, j'ai commencé par configurer les informations de l'application dans le fichier de configuration. Ensuite, j'ai installé l'environnement de préparation au build en suivant la documentation d'Expo, m'assurant que toutes les configurations étaient correctes, et que j'avais les bonnes dépendances et CLI (*npm install -g eas-cli*). Après m'être connectée à mon compte Expo (*eas login*), j'ai utilisé le service EAS (Expo Application Services) pour installer l'environnement de build (*eas build:configure*) initier le build, et EAS s'est occupé de générer et de gérer les clés de signature nécessaires.

J'ai construit l'application en utilisant Expo et EAS (*eas build --profile <profile-name>* pour l'Apk). Une fois le build terminé, Expo m'a fourni un lien de téléchargement. J'ai testé l'APK dans Android Studio sur mon émulateur Android, puis sur un téléphone Android réel. Après validation des tests, j'ai généré un App Bundle recommandé pour le Play Store.

Pour le déploiement, je me suis connectée à Google Play Console et j'ai créé une nouvelle application en remplissant toutes les informations requises. J'ai ensuite téléchargé l'App Bundle généré par Expo dans la section de publication. J'ai configuré la version, rempli les notes de version et suivi toutes les étapes nécessaires pour préparer l'application au déploiement. J'ai créé mon compte, passé les validations nécessaires et proposé mon

bundle. À l'heure où je rends ce document, l'application est toujours en attente de validation par Google, mais l'APK est présente dans le GitHub.

APP DESKTOP

Processus de Déploiement

Pour déployer mon application desktop, j'ai commencé par configurer le projet avec Electron et installer les dépendances nécessaires (*npm install electron electron-packager --save-dev*). Ensuite, j'ai ajouté un script de build dans le fichier package.json et j'ai exécuté ce script pour générer les fichiers de l'application (*npm run package*).

Après avoir construit l'application en mode production avec Electron Packager, j'ai utilisé Inno Setup pour créer l'exécutable. J'ai installé Inno Setup, créé un script de configuration pour définir le processus d'installation et compilé ce script pour générer l'installateur .exe.

Une fois l'exécutable généré, je l'ai testé sur un système Windows pour m'assurer qu'il fonctionnait correctement. Enfin, j'ai préparé l'exécutable pour la distribution et l'ai mis à disposition sur GitHub pour qu'il puisse être téléchargé facilement.

Conclusion

- En conclusion, le choix de Fly.io pour le déploiement de l'application Cinéphoria a permis de simplifier et d'optimiser le processus de mise en production. Grâce à l'automatisation du build et à la gestion efficace des données, j'ai pu déployer l'application rapidement tout en assurant sa stabilité et sa performance.
- Le choix des technologies Expo et EAS pour le déploiement de l'application mobile Cinéphoria a permis de simplifier et d'optimiser le processus de mise en production. Grâce à l'automatisation du build et à la gestion efficace des clés de signature, j'ai pu déployer l'application rapidement tout en assurant sa stabilité et sa performance.
- De même, pour l'application desktop, l'utilisation d'Electron, Electron Packager et Inno Setup a facilité la création d'un exécutable et d'un installateur convivial, rendant la distribution simple et efficace. Ces solutions m'ont permis de répondre aux exigences du projet de manière professionnelle et efficace.