

ANNEE 2023/2024

RAPPORT D'ALTERNANCE

présenté par

Perigault Paul

Mission effectuée du 04/09/2023 au 20/06/2024 chez :

WeVii

Sujet de la mission :

Alternance DevOps

Tuteur entreprise : **Pascal Perigault**

Tuteur école : **Philippe Oliviero**

Résumé

Titre : *Alternance DevOps.*

Mots clefs : Alternance ; DevOps ; Cloud Computing ; Développement web ; Déploiement ;

Résumé :

Cette alternance au sein de la société WeVii, ESN Bordelaise et lilloise, m'a permis en tant qu'apprenti, d'être formé au métier du DevOps dans le domaine du cloud, en me donnant pour mission la réalisation de différents projets éducatifs sur ce thème. Mon premier projet se nommait Sandbox (Bac à sable) et le second Find Your Team (Trouve ton équipe). La réalisation de ces projets m'a permis d'acquérir les compétences nécessaires afin de participer pleinement à un projet de recherche et de développement porté par un professeur chercheur qui travaille sur un projet de R&D (Recherche et

Développement) pour la société WeVii Innovation, JEI (Jeune Entreprise Innovante), entreprise qui bénéficie du Crédit Impôt Recherche (CIR). Ces deux projets ont été menés à bien avec la méthodologie DevOps. Cette dernière est utilisée pour réaliser les projets grâce à différents outils tels que Typescript, Go, Docker, Kubernetes et Terraform. Ils ont été réalisés durant cette première année d'alternance. Ils m'ont permis une forte progression dans mon apprentissage car ils demandent une certaine rigueur et de nombreuses compétences. De plus, cela m'a aidé à me conforter dans mes choix et projets futurs.

Title : *Apprenticeship DevOps.*

Keywords : Work-Study ; DevOps ; Cloud Computing ; Web Development ; Deployment

Abstract :

As an apprentice at WeVii, an ESN based in Bordeaux and Lille, I was trained in DevOps in the cloud sector, and given the task of carrying out various educational projects on this theme. My first project was Sandbox, and my second was Find Your Team. The completion of these projects enabled me to acquire the skills needed to participate fully in a research and development project led by a research professor working on a R&D (Research and Development) project for WeVii Innovation, a JEI (Jeune

Entreprise Innovante), a company that benefits from the Crédit Impôt Recherche (CIR) tax credit. Both projects were carried out using DevOps methodology. This methodology is used to implement the projects using various tools such as Typescript, Go, Docker, Kubernetes and Terraform. These projects were carried out during my first year of work-study. They enabled me to make great progress in my apprenticeship, as they require a certain rigor and a number of skills. What's more, they helped me to consolidate my choices and future projects.

Remerciements

Je remercie tout d'abord Nicolas Marie-Magdelaine pour son apport en compétences et son expertise de professeur chercheur qu'il n'a pas manqué de mettre à ma disposition. Il a permis la réalisation des projets éducatifs en s'assurant de mon épanouissement professionnel tout au long de cette année avec un suivi hebdomadaire de qualité.

Je remercie aussi Cédric Maitre pour l'aide qu'il m'a apporté dans le suivi administratif, élément nouveau pour moi dans le cadre d'une expérience professionnelle.

Et également Denis Dormegnie, pour son accompagnement régulier et pour son expertise qu'il m'a partagé en tant que Chef de Projet et architecte technique.

Je tiens à remercier gracieusement l'ensemble du personnel de WeVii pour leur bienveillance.

Ainsi que l'ensemble du personnel enseignant de l'TUT Paris Rives de Seine.

Et particulièrement Philippe Oliviéro, mon tuteur enseignant, qui a su être présent pour moi au cours de cette année.

Pour finir, je remercie Pascal Perigault pour l'opportunité qu'il m'a proposé ainsi que pour le suivi quotidien dont il a fait part à mon égard.

Table des matières

Résumé	1
Remerciements	2
Introduction	4
1 Environnement professionnel	5
1.1 L'entreprise	5
1.1.1 Évolution	5
1.2 Les grandes dates	5
1.3 Ma place dans l'entreprise	6
1.4 Organigramme de l'entreprise	6
1.5 Organisation de l'entreprise	6
1.6 La vie en entreprise	7
2 Projet Bac à Sable	8
2.1 Introduction	8
2.2 Technologies utilisées	8
2.3 Objectif du projet	8
2.4 Conception	9
2.4.1 Formation et apprentissage	9
2.4.2 Développement	9
2.5 Déploiement	10
2.6 Amélioration de l'application	11
2.7 Conclusion	11
3 Projet Find Your Teams	12
3.1 Introduction	12
3.2 Technologies utilisées	12
3.3 Objectif du projet	12
3.4 Conception	13
3.4.1 Première conception de l'application	13
3.4.2 Développement des fonctionnalités nécessaires	13
3.5 Amélioration de l'application	14
3.6 Déploiement et Automatisation	15
3.7 Conclusion	17
Conclusion	19
Glossaire Technique	20
Bibliographie et Webographie	23

Introduction

A la fin de ma première année en BUT informatique, j'ai eu la possibilité de poursuivre mon cursus en apprentissage. La recherche d'une entreprise n'a pas été évidente mais au bout de plusieurs mois de recherche j'ai su saisir une très bonne opportunité qui correspondait à mes critères. Cependant, je n'ai pas pris ma décision directement car l'entreprise se trouvant à Bordeaux il était clair qu'une grande partie de mon apprentissage se ferait en distanciel. Après mûre réflexion et suite à de nombreuses discussions avec l'entreprise, j'ai su que le suivi allait être quotidien et que le projet avait beaucoup à m'apporter, j'ai donc accepté.

En effet, durant cette première année d'apprentissage, j'ai intégré l'Entreprise de Service Numérique (ESN) **WeVii** qui est située à Bordeaux. Elle a pour modèle économique la prestation de service dans le numérique. WeVii Bordeaux possède une filiale se nommant WeVii innovation. Ainsi, j'ai intégré WeVii Bordeaux afin d'être formé pour sa filiale.

Pour ma part, j'interviens dans un des projets nommé, « TDC » (Tree Digital Cloud) qui est un projet de transformation digitale. Mais pour cela j'ai dû être formé aux technologies du cloud tout en ayant une approche DevOps.

En somme, j'ai été formé aux technologies d'un projet afin d'y être intégré pleinement. Globalement, les missions qui me sont confiées sont la réalisation de projets techniques, dans l'optique de me faire découvrir et progresser sur différentes technologies.

Depuis quelques années, j'ai pour projet de me diriger vers l'architecture cloud, pour devenir architecte cloud. C'est un métier qui demande beaucoup de compétences car ses missions sont diverses. Il doit savoir automatiser la gestion des infrastructures, développer des fonctionnalités pour ces dernières, mais aussi avoir des qualités de management afin de gérer des équipes. J'aime l'idée de faire un métier pertinent, complet qui regroupe différents domaines, cela permet de ne pas se lasser. J'ai accepté cette alternance car j'ai su que j'allais développer, ce qui me donne l'opportunité de savoir si j'apprécie réellement le développement et si le métier d'architecte cloud est vraiment ce qui me correspond.

Dans la suite de mon rapport, je vais vous présenter l'entreprise ainsi que les deux projets réalisés au cours de mon alternance. En conclusion, nous ferons un point sur l'ensemble de cette année ainsi que sur les années futures.

Chapitre 1

Environnement professionnel

1.1 L'entreprise

WeVii est une Entreprise de Service du Numérique (ESN) créée en 2010 sous le nom d'**AFIB Décision**. Son activité tournait autour de l'informatique décisionnelle. En 2014, l'entreprise amorce un premier virage et se renomme **AFIB Services**, élargissant son activité à tous les services digitaux, et pas seulement au décisionnel. En 2019, face à des difficultés financières, l'entreprise est reprise par **Pascal Perigault** et **Romain Lavielle**, et devient **WeVii**.

1.1.1 Évolution

D'une situation critique avec quatre collaborateurs, l'entreprise s'est transformée pour atteindre aujourd'hui soixante-quinze collaborateurs répartis sur deux sites, Bordeaux et Lille, réalisant un chiffre d'affaires de sept millions d'euros.

WeVii est actuellement un groupe composé de huit entreprises :

- Trois holdings : **Carat Conseil**, **Faerie Consulting**, et **WeVii Holding**.
- Les sociétés filles : **WeVii Bordeaux**, **WeVii Innovation**, **WeVii Paris**, **WeVii Lille**, et **WeVii Tech**.

1.2 Les grandes dates

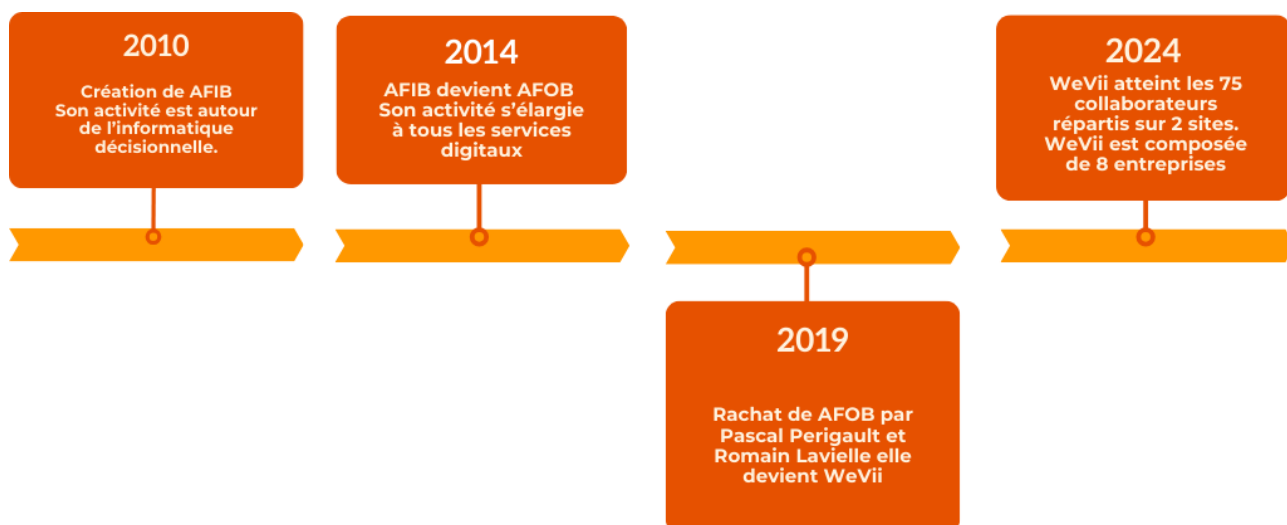


FIGURE 1.1 – Les grandes dates de l'entreprise

1.3 Ma place dans l'entreprise

Dans le cadre de cette transformation, **WeVii** lance des projets, dont un projet nommé « **TDC** » (**Tree Digital Cloud**), porté par une nouvelle entreprise, **WeVii Innovation**, une JEI (Jeune Entreprise Innovante). C'est pour ce projet, en tant que salarié de **WeVii Bordeaux**, que j'interviens dans le cadre de mon apprentissage.

1.4 Organigramme de l'entreprise

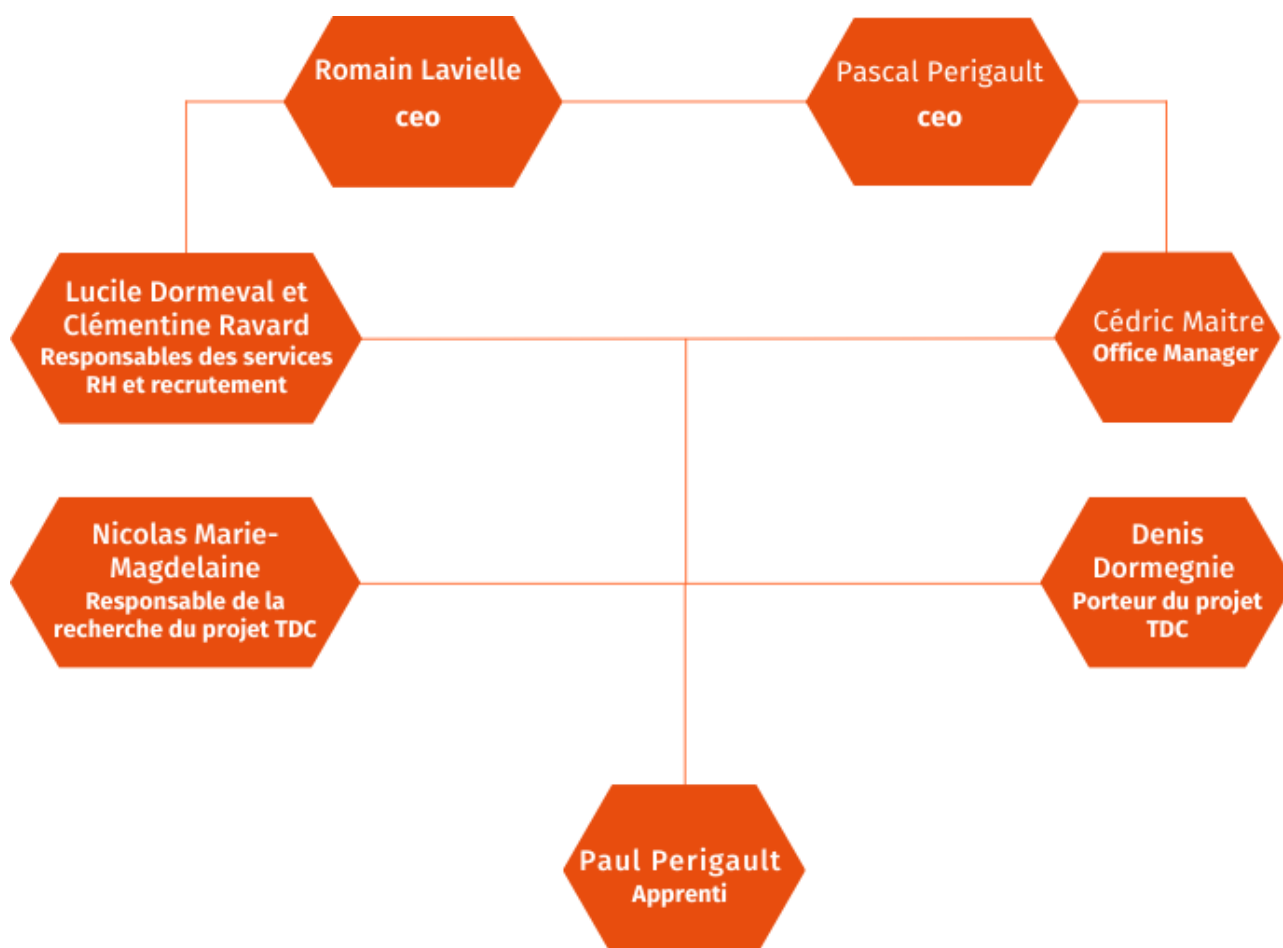


FIGURE 1.2 – Organigramme de l'entreprise

1.5 Organisation de l'entreprise

L'organisation du groupe est la suivante :

- **Pascal Perigault** et **Romain Lavielle** sont les CEO.
- **Grégory Lacombe**, associé, dirige **WeVii Tech**, filiale de WeVii, nouvellement créée et spécialisée dans les métiers du cloud, de la sécurité et des opérations.
- **Lucile Dormeval** et **Clémentine Ravard** sont responsables des services RH et recrutement.
- **Cédric Maitre** est l'Office Manager (il dirige tout le back office).
- Les fonctions financières et paies sont sous-traitées par un cabinet spécialisé.
- **Nicolas Marie-Magdelaine**, Docteur en informatique, est responsable de la recherche du projet TDC. Il pilote mon activité technique.
- **Denis Dormegnien** est le porteur du projet TDC. Il pilote également mes projets.

1.6 La vie en entreprise

Durant la formation, mon outil de travail était un ordinateur portable MacBook Air fourni par l'entreprise. J'ai été en grande partie en télétravail, bien que parfois je descendais à Bordeaux pour voir mes collègues et participer à la vie quotidienne et professionnelle de l'entreprise, afin de mieux m'intégrer.

Pour communiquer, nous utilisons **Teams**, un logiciel conçu par Microsoft permettant d'envoyer des messages, de faire des réunions à distance et de faciliter la communication inter-entreprise. Malgré la distance, j'ai eu la chance de participer à la vie de l'entreprise, notamment en étant présent en visioconférence aux **WeTalkTech**, un podcast en direct où chaque intervenant présente une technologie ou un projet.

Les tâches m'étaient fournies par le chercheur, mais j'avais la possibilité d'ajouter une part de créativité. J'utilisais la méthodologie **DevOps** pour construire et déployer mes projets tout en ayant une approche minimaliste apportée par le chercheur. Elles étaient sous forme de fonctionnalités et sous contraintes de temps, ce qui signifie que pour chaque élément à ajouter à l'application, nous estimons un temps pour le réaliser.

Chapitre 2

Projet Bac à Sable

2.1 Introduction

Mon premier projet chez WeVii est le projet bac à sable. Il m'a permis de faire mes premiers pas dans le monde du DevOps.

2.2 Technologies utilisées

Les principales technologies utilisées dans ce projet sont :

- JavaScript : Un langage de programmation utilisé pour le web.
- Angular : Un outil qui aide à créer des applications web.
- Node.js : Un environnement pour exécuter du code JavaScript sur un serveur.
- Docker : Un outil qui permet de mettre une application dans un "conteneur" pour qu'elle puisse fonctionner n'importe où.
- Kubernetes : Un système pour gérer et orchestrer les conteneurs Docker.
- Helm : Un outil pour gérer des applications Kubernetes.
- Serveur Cloud : Un ordinateur qui exécute des applications sur Internet.
- ArgoCD : Un outil pour automatiser le déploiement d'applications.
- GitHub : Un site web pour stocker et partager du code.
- GitHub Actions : Un outil pour automatiser des tâches comme le déploiement.
- MongoDB : Une base de données pour stocker des informations.

2.3 Objectif du projet

L'objectif de ce projet était de faire une petite application qui affichait « Hello » et un prénom ajouté par l'utilisateur, l'application affiche « World » par défaut.

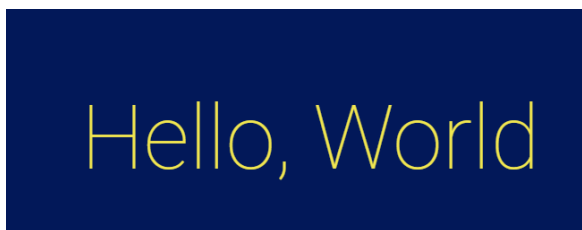


FIGURE 2.1 – Application lors de la première connexion

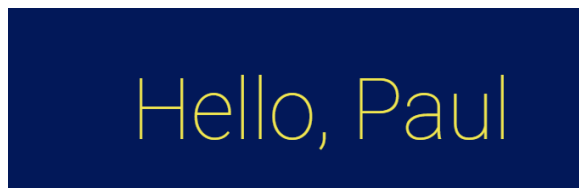


FIGURE 2.2 – Une fois un prénom ajouté

L'application devait se souvenir du prénom grâce aux cookies (des petits fichiers stockés sur l'ordinateur de l'utilisateur) et donc l'enregistrer.

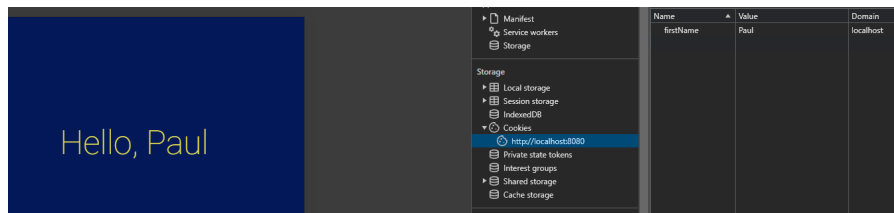


FIGURE 2.3 – Cookie stocké sur l'ordinateur de l'utilisateur

Elle a dû être hébergée sur un serveur Web, ce qui a permis de découvrir la conteneurisation (mettre l'application dans un conteneur), l'orchestration (gérer plusieurs conteneurs) ainsi que le déploiement continu (mettre à jour l'application automatiquement).

Pour empêcher que l'application soit victime de cyber-attaques, nous avons appliqué de bonnes pratiques de sécurité, comme cacher les clés API et les URLs sensibles en utilisant des variables d'environnement ou des secrets.

2.4 Conception

2.4.1 Formation et apprentissage

Dans un premier temps, afin de réaliser ce projet, l'essentiel était de se former sur JavaScript. De plus, étant donné qu'à l'IUT nous faisons du JavaScript, cela m'a permis de progresser plus rapidement. En parallèle, j'ai dû comprendre et apprendre le fonctionnement du Framework Angular et comment créer une API CRUD (une interface qui permet de créer, lire, mettre à jour et supprimer des données dans une base de données).

2.4.2 Développement

Une fois les connaissances acquises, j'ai pu commencer à développer l'application. L'application est divisée en deux parties :

— **FrontEnd** : La partie visible de l'application, réalisée avec Angular.

```

1 getFirstName(): Observable<{ string }> {
2   const firstName = this.cookieService.
    get('firstName');
3   return this.http.get<{ firstName:
    string }>(
4     `${this.apiUrl}/firstName?cookie=${{
    firstName}}`
5   );
6 }
7
```

FIGURE 2.4 – Exemple de code du FrontEnd : Récupération du prénom via un service de cookies et une requête HTTP GET.

— **BackEnd** : La partie qui fonctionne côté serveur, réalisée avec Node.js.

```
1 app.get('/backend/firstName', async (req,
  res) => {
2   const userId = req.query.userId;
3   console.log("ID de l'utilisateur dans la
    requête :", userId);
4   if (!userId) {
5     res.json({
6       'World'
7     });
8     return;
9   }
10  try {
11    const user = await User.findById(
      userId);
12    console.log('Utilisateur récupéré
    depuis MongoDB :', user);
13    if (!user) {
14      res.json({ firstName: 'World' });
15      return;
16    }
17    res.json({ firstName: user.firstName
      });
18  } catch (error) {
19    console.error('Erreur lors de la récup
    ération du prénom depuis MongoDB :',
      error);
20    res.status(500).json({ error: 'Erreur
    lors de la récupération du prénom.' });
21  }
22 });
23
```

FIGURE 2.5 – Exemple de code du BackEnd : Récupération du prénom de l'utilisateur depuis la base de données.

Le développement a pris du temps, avec des erreurs et des retours en arrière fréquents, mais cela m'a beaucoup motivé et m'a donné envie de progresser. Une fois le développement terminé, nous avons acquis de nombreuses compétences et nous étions prêts pour le déploiement.

2.5 Déploiement

Le déploiement consiste à mettre l'application en ligne sur un serveur. Le chercheur m'a fourni un serveur cloud sécurisé par une connexion SSH. Il m'a donné des instructions afin de me former, et m'a recommandé de conteneuriser l'application sous forme de conteneur Docker.

```
1 FROM node:18.18.0
2 WORKDIR /app
3 COPY package*.json ./
4 RUN npm install -g npm@10.2.1
5 RUN npm install
6 RUN npm audit fix
7 EXPOSE 3000
8 COPY app.js .
9 COPY config.json .
10 CMD ["node", "app.js"]
11
```

FIGURE 2.6 – Exemple de code d'un fichier Docker pour conteneuriser le backend.

Les conteneurs sont légers et faciles à transporter, ce qui optimise l'espace et permet de faire fonctionner l'application directement sur le serveur. Le chercheur s'est occupé de la partie infrastructure, en mettant en place un DNS pour vérifier si l'application fonctionnait. Après avoir réalisé la première partie du déploiement et appliqué les bonnes pratiques de cybersécurité, nous avons pu ajouter des fonctionnalités supplémentaires pour augmenter la difficulté et améliorer l'application.

2.6 Amélioration de l'application

Le chercheur m'a fait découvrir Kubernetes, une technologie très utilisée pour le déploiement et l'orchestration de conteneurs. Kubernetes permet d'orchestrer des conteneurs, gérer leur cycle de vie, et assurer le bon fonctionnement de l'application grâce à des outils de déploiement continu. En utilisant Kubernetes, j'ai pu mettre en place un déploiement continu de manière simple et efficace pour automatiser au mieux les tâches répétitives.

2.7 Conclusion

Une fois le déploiement finalisé et fonctionnel, le chercheur a jugé que nous pouvions partir sur un projet plus concret et complet. Ce projet était très pédagogique mais peu concret d'un point de vue professionnel. Il m'a permis de découvrir comment réaliser un projet et quelles sont les étapes pour y arriver. De plus, il m'a poussé à me renseigner sur toutes les technologies que j'ai été amené à utiliser. Ce projet a été un très bon point de départ car il n'a pas été complexe dans sa forme mais il permet de pratiquer de nombreux domaines de compétences dans le DevOps. J'ai apprécié réaliser ce projet, et malgré les difficultés rencontrées, je reste passionné.

Chapitre 3

Projet Find Your Teams

3.1 Introduction

Le second projet est aussi guidé par le chercheur, mais avec une approche complètement différente. C'est une approche « Startups » où nous avons cherché un domaine que nous apprécions et nous avons essayé de résoudre un problème existant dans ce domaine. Pour ce projet, nous avons choisi un domaine que nous aimons : le sport. Nous n'avions jamais entendu parler d'un réseau social permettant de trouver un partenaire ou un club pour s'entraîner dans n'importe quel sport. Après quelques recherches, nous avons constaté qu'il n'existait pas encore une telle application sur le marché. Ce projet a pour principal objectif de renforcer et d'augmenter les compétences déjà acquises sur le projet Bac à Sable.

3.2 Technologies utilisées

- Angular : Un framework JavaScript qui facilite la création d'applications Web interactives.
- Node.js : Un environnement d'exécution JavaScript côté serveur.
- TypeScript : Un langage de programmation qui est une extension de JavaScript.
- Go : Un langage de programmation simple et performant.
- MongoDB : Une base de données utilisée dans les applications Web.
- GitHub Actions : Un service pour automatiser diverses tâches telle que le déploiement.
- GitHub : Une plateforme de développement collaborative basée sur Git.
- Google Cloud Platform : Une suite de services cloud proposée par Google.
- Terraform : Un outil d'infrastructure en tant que code (IaC).
- Google Cloud Run : Un service qui permet d'exécuter des conteneurs Docker.
- Google Cloud Build : Un service qui permet d'automatiser la construction d'applications.
- Google Cloud Artifact Registry : Un service qui permet de stocker des images de conteneurs.
- Google Cloud Secret Manager : Un service de gestion des secrets.
- Google Cloud DNS : Un service pour gérer et résoudre les noms de domaine.

3.3 Objectif du projet

L'approche startup signifie qu'avant de commencer la partie de développement, il faut analyser ce qui existe, ce qui a fonctionné et ce qui fonctionne à l'heure actuelle. Il faut également analyser le marché, définir les objectifs de l'application, son design, sa maquette et ensuite développer. L'objectif

de l'application est de permettre à un utilisateur de créer un événement sportif, d'afficher une carte des événements et d'avoir un canal de message.

3.4 Conception

3.4.1 Première conception de l'application

Pour la première partie du développement, le chercheur m'a laissé développer à ma manière afin de voir ma gestion de projet individuel. J'ai voulu réaliser tous les éléments souhaités dans l'application et y rajouter des idées que j'avais sur le moment. Très vite, j'avais plein d'idées mais pas encore très concrètes, et j'étais un peu perdu. Le chercheur m'a donc expliqué qu'il fallait avoir une approche minimaliste pour déployer au plus vite, en se concentrant sur les fonctionnalités minimales de l'application. Nous avons utilisé une méthode de « refactoring » pour rendre le code réutilisable et minimaliste.

3.4.2 Développement des fonctionnalités nécessaires

Je me suis concentré sur le développement des fonctionnalités nécessaires au fonctionnement de l'application :

- Une carte interactive listant chaque événement.

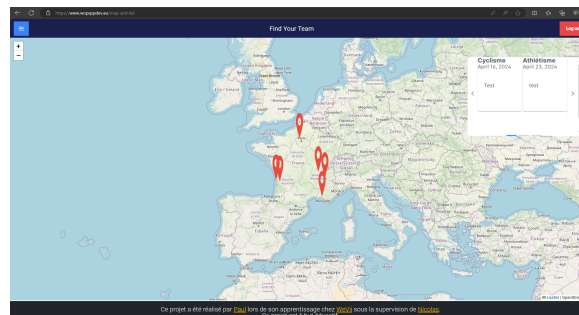


FIGURE 3.1 – Carte interactive des évènements

- Un formulaire permettant d'ajouter un événement.

FIGURE 3.2 – Formulaire d'ajout d'événement

- Une liste des événements permettant au créateur de l'événement de modifier ou supprimer l'événement.

Une fois l'application développée de manière minimaliste, j'ai créé une instance de machine virtuelle (VM), un nom de domaine et conteneurisé mon application. Commencer à déployer sur une VM permet de faire du diagnostic réseau et d'avoir la main sur tout en cas de problème. Une fois le déploiement réussi, il faut mettre en place un déploiement continu pour automatiser ce processus,

car le déploiement est une tâche répétitive. Pour cela, j'ai choisi GitHub Actions qui permet de faire un déploiement directement avec SSH et d'y mettre ses instructions.

3.5 Amélioration de l'application

J'ai recueilli les retours des utilisateurs extérieurs au projet pour connaître leurs avis sur l'application. Voici un exemple de retour reçu :

Hello,

Bien le sport 😊

- La page n'est pas affichée complètement
- Il n'y a aucune explication de ce que c'est
- Le lien WeVii en bas de page ne renvoie pas vers <https://www.wevii.net>
- Difficile de comprendre que pour se connecter il faut un pseudo (et pas de mot de passe)
- « ajouter l'événement » --> On ne comprends pas bien car aucune action hormis un encadré en bas.
- On ne comprend pas à quoi sert la carte

FIGURE 3.3 – Commentaire sur l'application

Les utilisateurs avaient du mal à comprendre l'application, j'ai donc ajouté une page d'accueil pour mieux expliquer son fonctionnement.

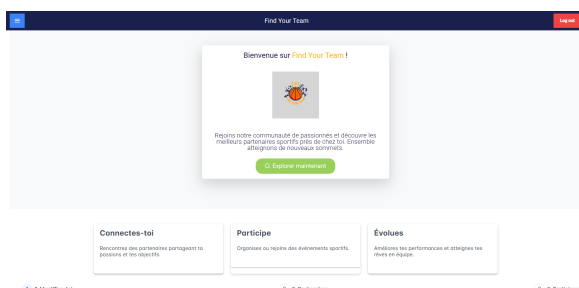


FIGURE 3.4 – Page d'accueil lors de la connexion

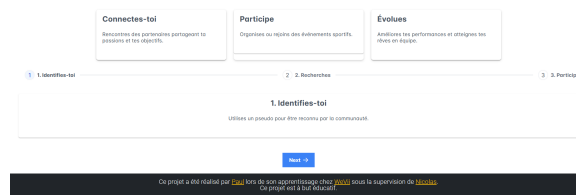
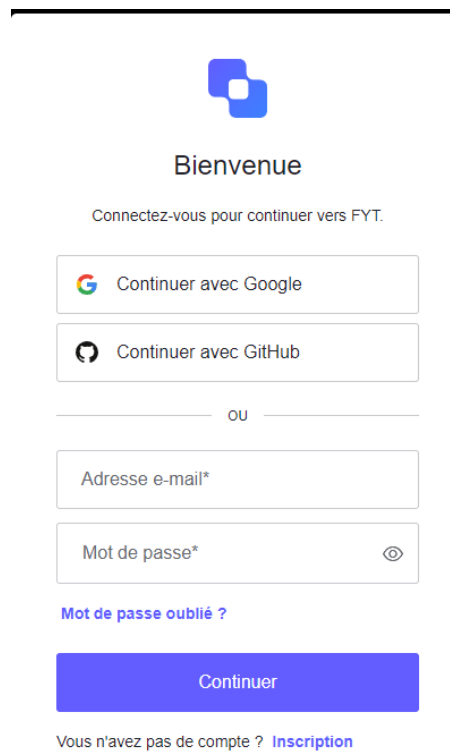


FIGURE 3.5 – Partie basse de la page d'accueil

Les utilisateurs voulaient également un système d'authentification. Plutôt que de créer le système moi-même, j'ai utilisé Auth0, un service d'authentification facile à intégrer.



The image shows a web interface for logging in. At the top, there is a blue logo consisting of two overlapping squares. Below the logo, the word "Bienvenue" is centered. Underneath, a line of text says "Connectez-vous pour continuer vers FYT." There are two buttons for social login: "Continuer avec Google" and "Continuer avec GitHub". Below these, the word "ou" is centered. Then, there are two input fields: "Adresse e-mail*" and "Mot de passe*", with an eye icon to toggle password visibility. A link "Mot de passe oublié ?" is below the password field. A large blue button labeled "Continuer" is at the bottom. At the very bottom, a link says "Vous n'avez pas de compte ? [Inscription](#)".

FIGURE 3.6 – *Authentification avec Auth0*

J'ai dû corriger des erreurs d'interface utilisateur, notamment la synchronisation entre l'ajout d'un élément et son affichage sur la carte. Nous avons des cours sur la qualité du développement, où nous avons appris comment rendre le code plus facile à maintenir, avec des principes comme SOLID et les design patterns. Cela m'a poussé à refactoriser le code pour le rendre plus clair et fonctionnel, par exemple en séparant le formulaire et l'affichage des événements en deux routes différentes.

3.6 Déploiement et Automatisation

Pour le déploiement, j'ai utilisé Google Cloud Run, qui exécute directement un conteneur. J'ai également utilisé Google Cloud Artifact Registry pour stocker les images de conteneur, Cloud Build pour construire l'image automatiquement lors du déploiement continu, et Cloud Deploy pour tout déployer avec un fichier YAML.


```

1 steps:
2   - name: 'gcr.io/cloud-builders/git'
3     args: ['ls', '-la', '.'] # test
4
5   # install node package
6   - name: 'node'
7     entrypoint: 'npm'
8     args: [ 'install' ]
9
10  # Build Node App
11  - name: 'node'
12    entrypoint: 'npm'
13    args: [ 'run', 'build' ]
14
15  # Docker Build
16  - name: 'gcr.io/cloud-builders/docker'
17    args:
18      [
19        'build',
20        '-t',
21        ImageDocker,
22        '-f',
23        'Dockerfile',
24        '.',
25      ]
26
27  # Docker Push to Google Artifact Registry
28  - name: 'gcr.io/cloud-builders/docker'
29    args: ['push', ImageDocker]
30
31  - name: 'hashicorp/terraform'
32    args: ['init']
33    env: ['GOOGLE_CREDENTIALS=/workspace/account.json']
34
35  - name: 'hashicorp/terraform'
36    args: ['apply', '-auto-approve']
37    env: ['GOOGLE_CREDENTIALS=/workspace/account.json']
38
39  availableSecrets:
40    secretManager:
41      - versionName: LienVersion
42        env: 'GOOGLE_CREDENTIALS'
43
44    # Deploy to Cloud Run
45  - name: 'gcr.io/cloud-builders/gcloud'
46    args:
47      [
48        'run',
49        'deploy',
50        'cloudrunservice',
51        '--image',
52        ImageDocker,
53        '--region',
54        'us-central1',
55        '--platform',
56        'managed',
57      ]
58

```

FIGURE 3.7 – Exemple de code du fichier cloudbuild.yaml

Google Cloud Run fournit une URL pour le service. J’ai utilisé Cloud DNS pour rediriger vers mon service Google Run pour le frontend. Mon frontend et mon backend communiquent via une API CRUD. Pour automatiser cela, j’ai créé un service en Go qui récupère l’URL du backend.

```

1 import (
2     "context"
3     "github.com/gin-contrib/cors"
4     "github.com/gin-gonic/gin"
5     "microservice.com/appfyt/gcp"
6     "net/http"
7 )
8
9 func getURLByName(c *gin.Context) {
10     name := c.Param("name")
11     c.IndentedJSON(http.StatusOK, gcp.GetURLByName(gcp.ListServicesRequest(
12         gcp.ConnectToGCP(context.Background()), context.Background()), name))
13 }
14
15 func getAllUrlAndName(c *gin.Context) {
16     c.IndentedJSON(http.StatusOK, gcp.GetAllUrl(gcp.ListServicesRequest(gcp
17         .ConnectToGCP(context.Background()), context.Background())))
18 }
19
20 func LauncServ() {
21     r := gin.Default()
22     r.Use(cors.New(cors.Config{
23         AllowOrigins: []string{"*"},
24         AllowMethods: []string{"GET", "POST", "PUT", "DELETE", "OPTIONS"},
25         AllowHeaders: []string{"Origin"},
26         ExposeHeaders: []string{"Content-Length"},
27         AllowCredentials: true,
28     }))
29     r.GET("/service/:name", getURLByName)
30     r.GET("/service", getAllUrlAndName)
31     r.Run(":3000")
32 }

```

FIGURE 3.8 – Exemple de code du Service Go pour récupérer l'URL du backend

Pour déployer ce service, j'ai utilisé Terraform, qui permet de coder l'architecture souhaitée et de créer automatiquement l'infrastructure nécessaire.

```

1 resource "google_cloud_run_service" "microserviceurl" {
2     name      = var.serviceName
3     location  = var.region
4
5     template {
6         spec {
7             containers {
8                 image = var.imageMicroserviceurl
9             }
10        }
11    }
12
13    autogenerate_revision_name = true
14 }
15

```

FIGURE 3.9 – Exemple de code Terraform pour déployer le service Go

3.7 Conclusion

Ce projet m'a fait découvrir de nombreuses facettes du monde de l'informatique telles que le développement, le déploiement, la gestion d'une infrastructure, la sécurité ainsi que l'automatisation. Tout ceci m'a permis de comprendre et de me rendre compte des domaines et compétences que j'apprécie et que je souhaite améliorer.

J'aime le développement avec une préférence pour le développement BackEnd. J'ai adoré utiliser le langage Go et je souhaite m'expertiser dans ce langage. Le développement FrontEnd n'est pas ma partie favorite car elle demande des compétences en design que je n'ai pas et qui ne me plaisent pas particulièrement.

La sécurité est une partie obligatoire et essentielle pour chaque domaine, donc il est important de connaître les bonnes pratiques en termes de cybersécurité afin de réaliser des applications et des architectures sécurisées. La gestion de l'infrastructure, le déploiement et l'automatisation peuvent être réalisés avec du code. C'est une partie très intéressante, surtout combinée avec le développement backend. Cela permet d'augmenter la productivité et d'être autonome dans la mise en production des applications.

Ce projet m'a donc confirmé que je souhaite travailler dans le domaine du DevOps, qui combine développement et opérations pour améliorer et automatiser les processus de déploiement.

Conclusion

Cette première expérience professionnelle m'a été bénéfique sur différents points.

Tout d'abord, d'un point de vue technique, car j'ai appris de nombreuses choses dans le domaine de l'informatique, mais aussi d'un point de vue personnel, car j'ai découvert la conscience professionnelle, la rigueur et l'importance de la régularité. Mes compétences et connaissances ont grandement augmenté durant l'année entre septembre 2023 et maintenant.

Le principe de l'alternance consiste à alterner entre étudier à l'iut Paris Rives de Seine et travailler chez WeVii. Selon moi ce principe s'est avéré très enrichissant des deux cotés. D'un côté l'iut apporte un enseignant transversal tel que le droit, l'anglais, le management... Mais aussi une approche très théorique de ce que l'on peut pratiquer en entreprise ceci permet de comprendre plus en profondeur la partie technique mais aussi de comprendre le fonctionnement de l'entreprise grâce aux matières transversales. De l'autre côté l'entreprise permet d'avancer davantage sur les points techniques et apporte une pratique régulière qui favorise l'apprentissage.

Pour ma part, l'alternance m'a permis d'évoluer en termes de compétences mais aussi en connaissances et ainsi de constater plus facilement mon évolution personnelle.

Le fait de constater sa propre évolution personnelle entre chaque projet permet de s'intéresser d'autant plus à ces derniers. Dans l'informatique les projets sont nombreux et peuvent être très variés. Ce que j'aime dans ce domaine, c'est la pratique constante qui permet d'évoluer chaque jour. Grâce à cela, nous pouvons constater une réelle évolution tout en continuant d'apprendre quotidiennement, peu importe le projet ou le domaine.

À la suite de cette année, je souhaite poursuivre en école d'ingénieur, toujours en tant qu'alternant chez WeVii, tout en me spécialisant dans le domaine du DevOps avec un focus sur le Cloud.

Cependant, certains aspects me restent à découvrir, comme le travail dans une grande équipe composée de développeurs, de designers, de testeurs et de managers. J'aimerais appliquer d'autres méthodes de travail et en particulier les méthodes agiles. Je souhaite réaliser des projets avec des contraintes plus importantes que celles de cette année, car je pense que cela serait une expérience enrichissante et m'apporterait de nouvelles compétences.

Plus tard, dans ma vie professionnelle, je souhaite continuer dans l'architecture cloud et le DevOps. Cette année a renforcé mes choix. Je voudrais faire un peu de salariat, mais aussi être indépendant, pour avoir du temps pour enseigner dans des domaines techniques, sous forme de conférences ou de cours. Je pense que l'enseignement pourrait me plaire, mais je ne me vois pas y consacrer toute ma vie. Cependant, enseigner entre 10 et 15 heures par semaine pourrait être une expérience très enrichissante.

Glossaire Technique

A

Angular

- Un framework JavaScript qui utilise TypeScript pour le développement d'applications Frontend.

API CRUD

- Un ensemble de services permettant de créer, lire, mettre à jour et supprimer des données (CRUD signifie Create, Read, Update, Delete).

ArgoCD

- Un outil de déploiement continu pour Kubernetes.

B

Backend

- La partie serveur du développement, non visible pour un utilisateur.

D

DevOps

- Une méthodologie combinant le développement logiciel (Dev) et les opérations informatiques (Ops) pour améliorer la collaboration et l'efficacité.

Docker

- Un logiciel qui utilise la technologie de conteneurisation pour permettre aux applications de fonctionner de manière cohérente sur différents environnements.

DNS

- Système de noms de domaine, un service qui convertit les noms de domaine lisibles par les humains en adresses IP lisibles par les machines.

F

FrontEnd

- Les éléments visibles par l'utilisateur dans une application ou un site web.

Framework

- Une boîte à outils pour les développeurs qui fournit des fonctionnalités prêtes à l'emploi et des règles à suivre pour gagner du temps.

G

Git

- Un système de contrôle de version qui permet de suivre les modifications apportées au code source.

GitHub

- Un site web qui permet de stocker du code en utilisant la technologie Git.

GitHub Actions

- Un service d'automatisation et de déploiement continu intégré à GitHub.

Go (Golang)

- Un langage de programmation créé par Google, compilé et fortement typé, avec une syntaxe simple d'utilisation.

Google Cloud Artifact Registry

- Un service de Google Cloud Platform pour stocker des images de conteneurs.

Google Cloud Build

- Un service de Google Cloud Platform pour automatiser la construction d'applications.

Google Cloud DNS

- Un service de Google Cloud Platform pour gérer et résoudre les noms de domaine.

Google Cloud Platform

- Une suite de services cloud proposés par Google.

Google Cloud Run

- Un service de Google Cloud Platform pour exécuter des conteneurs Docker.

Google Cloud Secret Manager

- Un service de gestion des secrets de Google Cloud Platform pour stocker et gérer de manière sécurisée des informations sensibles.

H

Helm

- Un outil de gestion des applications Kubernetes qui facilite le déploiement et la gestion des applications sur Kubernetes.

J

JavaScript

- Un langage de programmation principalement utilisé pour créer des scripts sur les pages web afin de les rendre interactives.

K

Kubernetes

- Un outil qui automatise le déploiement, la gestion et la mise à l'échelle des conteneurs d'applications.

M

MongoDB

- Une base de données NoSQL utilisée pour stocker des informations.

N

Node.js

- Un environnement qui permet d'exécuter du code JavaScript côté serveur.

S

Serveur Cloud

- Un ordinateur virtuel qui exécute des applications sur Internet.

SSH (Secure Shell)

- Un protocole de communication sécurisé utilisant un chiffrement asymétrique pour se connecter de manière sécurisée à un serveur.

T

Terraform

- Un outil d'infrastructure en tant que code (IaC) qui permet de définir et de gérer des infrastructures cloud avec des fichiers de configuration.

TypeScript

- Une surcouche de JavaScript qui ajoute des types statiques pour rendre le code plus sécurisé et maintenable.

U

URL

- Uniform Resource Locator, une adresse web qui spécifie l'emplacement d'une ressource sur Internet.

Y

YAML

- Un langage de sérialisation de données facile à lire et à écrire utilisé pour les fichiers de configuration.

Bibliographie et Webographie

- **Mozilla Developer Network (MDN)**

<https://developer.mozilla.org/>

Ressource pour HTML, CSS, JavaScript, TypeScript, Angular, Node.js, HTTP.

- **Red Hat**

<https://www.redhat.com/>

Documentation sur Kubernetes, Docker, DevOps, Terraform, Cloud.

- **ResearchGate**

https://www.researchgate.net/publication/357163866_Observability_and_resources_management_in_cloud-native_environnements

Ressource pour l'observabilité et la gestion des ressources dans les environnements cloud-native.

- **Roadmap.sh**

<https://roadmap.sh/>

Outil pour situer ses compétences.

- **Google Cloud Documentation**

<https://cloud.google.com/docs?hl=fr>

Documentation sur Google Cloud Run.

- **Overleaf**

<https://fr.overleaf.com/latex/templates>

Outil de rédaction collaborative pour LaTeX afin de trouver des idées de modèles pour le rapport.