ISEN
ALL IS DIGITAL!
OUEST

Institut Supérieur de l'Électronique et du Numérique
Tél. : +33 (0)2.98.03.84.00
Fax : +33 (0)2.98.03.84.10
20, rue Cuirassé Bretagne
CS 42807 – 29228 BREST Cedex 2 - **FRANCE**

M1

Année scolaire 2023/2024
Stage d'application

Création d'un portail intégré et automatisé

Effectué du 02/05/2024 au
30/08/2024 à Cesson-Sévigné

orange™

2 avenue Belle Fontaine
35510, Cesson-Sévigné

Étudiant
PROVAIN Antoine
Génie logiciel

Tuteur entreprise
CHEN Yiping
CISS, 06 83 75 98 07

Correspondant ISEN
VIGNAUD Christophe

CHEN 20/03/24

Remerciement

Je tiens à exprimer ma profonde gratitude et mes sincères remerciements à toutes les personnes qui ont contribué à la réussite de mon stage au sein d'Orange Innovation, du 2 mai au 30 août 2024. Leur soutien, leur expertise et leur confiance ont grandement enrichi mon expérience professionnelle.

Je tiens tout d'abord à remercier mon tuteur de stage, M. CHEN, pour sa guidance et ses précieux conseils tout au long de mon stage. Sa patience et sa disponibilité ont été d'une aide précieuse dans la réalisation de mes missions et dans mon développement personnel.

Je remercie également Nathalie AMANN pour sa confiance. Cette expérience a été extrêmement enrichissante et m'a permis de développer de nouvelles compétences. Merci pour son soutien et ses précieux conseils lors des réunions hebdomadaires tout au long de mon stage.

Pour finir, mes remerciements vont également à toute l'équipe CISS pour leur accueil chaleureux, leur collaboration et leur partage de connaissances. Chaque membre de l'équipe a contribué à ma formation en me permettant de participer activement aux projets et aux réunions. Leurs esprits d'équipe et leur bienveillance ont créé un environnement de travail stimulant et convivial.

Sommaire

Glossaire

Plugins : programme additionnel pour une application internet ou un logiciel de bureau.

Conteneur Docker : c'est un conteneur exécutable léger et autonome, qui comprends les éléments pour exécuter une application, notamment des bibliothèques, des outils systèmes, du code, etc…

Machine hôte : ordinateur relié à un réseau informatique, qui fournit des services à un système ou un utilisateur.

Cluster Kubernetes : c'est un ensemble de machine (les nœuds) qui permettent d'exécuter des applications conteneurisées.

B2B : Business to Business. Décrit les activités commerciales entre entreprises.

B2C : décrit les activités et opérations commerciales qu'une entreprise effectue pour des clients particuliers.

ADN : porteur des caractéristiques génétiques.

AGIL : méthode de gestion de projet, apportant souplesse et performance à un projet.

CISS : Cloud Infrastructure Security & Services

CODIR : Comité de direction.

COMEX : Comité exécutif.

Instance : c'est une occurrence d'un élément logiciel qui est basée sur une définition de type.

API : interface de programmation d'application.

DevOps : repose sur une culture de collaboration entre développeurs et équipes opérationnelles, qui partagent les responsabilités et combinent le travail.

Jenkins : est un outil d'automatisation de serveur open source qui permet aux développeurs d'intégrer rapidement des changements dans leurs projets.

Pipelines : servant de « tuyaux » de données, pouvant provenir de multiples endroits (API, BDD, fichiers…).

VPN : Virtual Private Network, protège ses utilisateurs en cryptant les données et en masquant leurs adresses IP.

Proxy : joue le rôle de passerelle entre internet et votre machine.

Linux : système d'exploitation de type Unix.

WSL : sous-système Windows pour Linux.

Nvm : outils permettant de gérer plusieurs versions de Node.js sur une seule machine.

Git : système de contrôle de version qui a été inventé et développé par Linus Torvalds.

React : est une bibliothèque JavaScript utilisée pour construire des composants d'interface utilisateurs réutilisables.

BDD : Base de données.

Frontend : représente les éléments d'un site que les utilisateurs voient à l'écran et avec lesquels ils vont interagir.

Backend : parfois appelé côté serveur, gère les fonctionnalités générales de votre applications web.

RedHat : est un éditeur de logiciel de premier plan qui assemble des composants open-source, facile à commande et à mettre en œuvre.

PostgreSQL : Type de base de données.

Token : il s'agit d'un jeton, contenant un message envoyé par un serveur à un client, et stocké temporairement par ce dernier.

Table des figures

1. Introduction

Le sujet traité lors de ces quatre mois de stage au sein d'Orange Innovation est la création d'un portail informatique intégré et automatisé.

Mais qu'est-ce qu'un portail informatique ?

L'explication la plus simple serait de dire que c'est un site web à forte fréquentation qui offre un large éventail de contenus, de services et de liens vers des fournisseurs et qui est pratique pour l'utilisateur. Il permet d'accéder facilement aux informations et aux liens supplémentaires dont les clients ont besoin 24h/24 et 7j/7.

GitHub illustre parfaitement ces propos, en partie grâce à ces différents éléments :

- Collaboration
- Hébergement de projet

Dans le cadre de ce stage, le portail serait majoritairement utilisé par les développeurs, d'où le nom de **portail développeur** dans la suite de ce rapport. Il servira de lieu de référence pour les développeurs pour s'informer et contribuer à leurs écosystèmes. Cela permet de maintenir l'équilibre entre autonomie et découvrabilité. Centralisé en un même endroit, il est donc accessible pour tous. Voici les 3 atouts majeurs de ce genre de procédé :

- Navigation facile
- Agissement en libre-service
- Informations sur l'écosystème.

Pour en revenir à l'intitulé, les mots "intégré" et "automatisé" explicitent plusieurs points :

- **Automatisation** et **centralisation** des tâches courantes de gestion des logiciels, des infrastructures et des services.
- **Intégration** de manière fluide avec les outils et services existants.

Le but étant de réduire la charge administrative sur les développeurs pour leur permettre de gagner en efficacité.

Pour répondre à ces nombreuses exigences, l'outil utilisé pour la réalisation de ce portail s'appelle **Backstage**.

Figure 1 - Logo Backstage

Historiquement conçu par Spotify, pour répondre au besoin d'efficacité de leurs ingénieurs, Backstage centralise tous les documents et les outils nécessaires pour les développeurs, leur permettant aussi de le modéliser à leur gré. C'est cette autonomie et cette modulabilité qui fait de Backstage un atout majeur pour ses utilisateurs.

A son origine, le portail devait être implémenté et documenté en local sur la machine directement, car c'est la méthode la plus simple pour expérimenter les différents plugins et autres fonctionnalités de Backstage.

Dans un second temps, l'outil sera déployé dans un conteneur Docker. Cela peut être défini comme un environnement d'exécution contenant tous les composants nécessaires, tels que le code, les dépendances et les bibliothèques pour exécuter le code de l'application sans utiliser les ressources de la machine hôte.

Pour finir, l'objectif serait de déployer ce produit sur un cluster Kubernetes car il permet d'héberger des applications sur un cluster de serveurs avec des fonctions sophistiquées d'équilibrage de charge et d'allocation des ressources. Cela permet de maintenir un environnement fonctionnel, peu importe les pannes.

L'état d'avancement actuel se situe sur la première phase, car Backstage est une nouveauté et les fonctionnalités qu'il propose sont nombreuses. La mise en place de ces solutions est fastidieuse.

Lors de ce rapport, il sera évoqué dans les différentes parties les multiples recherches et pratiques afin de mettre en place ce portail développeur.

2. Présentation de l'entreprise

"La marque orange : un engagement qui traverse les époques".

Pour comprendre au mieux les enjeux liés à Orange, rien de mieux que de connaître ses origines. En effet, 30 ans après sa naissance, cette entreprise est devenue un pilier dans le monde de la télécommunication et des services numériques à travers le monde, faisant d'elle l'une des marques les plus puissantes au monde, avec une valeur de plus de 20 milliards de dollars en 2024. Son implantation dans 26 pays, au service de 298 millions de clients B2B et B2C, montre sa capacité culturelle.

Lancée le **28 avril 1994**, l'histoire d'Orange commence en Angleterre, avec des services inédits et une idée forte et différenciante : "placer l'humain avant la technologie". C'est cet ADN, perpétué aux fils des années, qui a donné à Orange cet héritage fort, y compris lors de son acquisition par France Télécom en 2000.

C'est lors de l'ouverture à la concurrence durant l'année **2000**, sur le marché des télécom en Europe, que l'opérateur français France Télécom rachète la marque anglaise Orange.

France Télécom se fait rapidement une place sur le marché mondial grâce à ses performances et à la qualité de ses réseaux. Dès **2009**, Orange est présent dans plus de 30 pays avec plus de 123 millions de clients.

C'est le **1er juillet 2013** que France Télécom devient officiellement Orange, lui donnant ainsi un nouveau visage avec un fort héritage culturel.

En **2017**, il y a eu l'inauguration de la filiale Orange Cyberdéfense et le lancement de la banque 100% mobile avec les offres d'Orange Bank en France.

C'est en **2024** que la nouvelle signature *"Orange est là"* fait son apparition. Orange devient aussi Partenaire Premium et Fournisseur Officiel de Paris 2024.

Orange a fait le choix d'une signature avec un engagement fort, pour faire comprendre à ses utilisateurs qu'ils sont un acteur de confiance, présent pour ses clients, en toutes circonstances.

"Orange est là. Ces trois mots reflètent l'identité de marque que nous souhaitons incarner chez Orange. Trois mots qui entérinent l'audace et l'engagement du Groupe pour plus de proximité, d'accessibilité, et de simplicité. Trois mots qui seront incarnés par l'ensemble des collaborateurs. Trois mots qui symbolisent la volonté d'Orange de rester proche de ses clients tout en continuant à se dépasser. La marque Orange forte de 30 ans d'existence se tourne ainsi vers l'avenir avec confiance et détermination" cite Caroline Guillaumin, Directrice Exécutive de la communication.

La marque dispose d'un formidable capital de confiance et reste ancrée au cœur de la vie des gens. Orange se classe 8ème dans le rapport d'évaluation des télécoms 2024 de Brand Finance. Elle a progressé de 10.2% en 2023, pour atteindre 20.3 milliards de dollars. Au niveau européen, elle atteint la seconde place des télécoms.

L’évolution de l’entreprise se prouve encore une fois par son adaptation. En effet, Orange se réinvente en créant de nouvelles marques filles, tout en conservant sa cohérence :

- Orange Money
- Orange Energies
- Orange Cyberdéfense
- Orange Business
- Orange Innovation

La réalisation du portail intégré et automatisé rentre dans le cadre de la création d’innovations, mais aussi de la recherche et de la mise en œuvre des politiques techniques. C’est la raison pour laquelle c’est **Orange Innovation** qui a permis de réaliser ce stage.

Parmi les ambitions d’Orange : se faire percevoir par ses clients grand public et entreprise comme un opérateur innovant, d’où l’importance d’Orange Innovation.

Dans un contexte d’évolution rapide des usages et des attentes de ses clients, Orange renforce ses capacités à préparer le futur, en construisant des atouts compétitifs et créateurs de valeur pour le Groupe :

- Développement de leur capacité d’influence
- Priorisation de leur innovation
- Favorisation des solutions mutualisées
- Développement des compétences et des méthodes AGIL.

En France et à l'international, près de 8000 personnes contribuent à cette ambition. Voici un organigramme permettant d’y voir un peu plus clair.

Voir Annexe

Au sommet de cet organigramme, Bruno ZERBIB qui est le directeur général de la branche Technologies & Innovation. Afin de mieux comprendre l’organigramme, il est important de savoir que chaque entité est représentée par une personne. L’exemple se portera sur l’entité Networks, dirigé par Laurent LEBOUCHER. Celui-ci dirige donc tout un lot de sous-service :

- CISS, ONE, COM, CASE, GRN, HR, SAS, WNI et NSSO

Celui qui nous intéresse particulièrement s’appelle CISS, car c’est dans ce service que l’équipe développe le portail développeur. Cependant, il faut aller encore plus loin dans cette arborescence pour en trouver son cœur :

- OL, IIE, IOP et TSP

C’est dans la section IOP, que nous retrouverons la dernière section nommée AGIL, dans laquelle est situé ce projet, dirigé par Nathalie AMANN.

Le projet étant tout juste démarré, il n'existait aucune base existante sur laquelle s'appuyer. Il est donc facile de prendre en main un projet vierge car tout est à faire. Cependant, cela représente une grosse quantité de travail de compréhension, pour mener à bien ce projet. La demande initiale est la suivante :

> *Mise en place de l'instance de Backstage en local sur la machine de l'hôte afin de configurer le portail développeur et d'y ajouter les éléments nécessaires à l'efficacité de ces utilisateurs. Il faut qu'il puisse être ensuite déployable pour que tout le monde puisse s'en servir et le modéliser comme bon lui semble*.

Orange a décidé de se positionner dans de nombreux pays à travers le monde avec une raison simple : une présence à l'internationale leur permet de disposer d'expertises au plus près de leurs clients. Ce réseau d'innovation à une présence mondiale, qui peut être explicité à travers cette carte du monde.

Voir Annexe

Le service "*IT Services*", se trouve dans de nombreux pays :

- Chine, Inde, Egypte, Tunisie, Pologne et Roumanie.

Pour le service "*Networks*", il est possible de les retrouver en :

- Pologne, Roumanie et en Inde.

Cependant, Orange a aussi de nombreux pôles d'innovation à travers la France. Le pôle qui s'occupe du projet Backstage quant à lui se trouve sur Atalante Cesson, dans la banlieue rennaise.

En ce qui concerne la répartition des hommes et des femmes au sein de l'équipe CISS, il est possible de dire qu'il y a un très fort déséquilibre. En effet, pour une équipe de 27 personnes, il y a 2 femmes pour 25 hommes, soit 12.5 fois plus d'hommes que de femmes. Cependant, ce qui est important de relever, c'est que parmi ces 2 femmes, l'une d'entre elles dirige l'équipe CISS.

D'après une enquête en mars 2023, les indicateurs de l'égalité professionnelle ne refléteraient pas la réalité opérationnelle. Les résultats de l'entreprise demeurent insuffisants notamment en ce qui concerne les postes à responsabilités, puisque seuls 36.1% des managers sont des femmes, 36.7% dans les CODIR de Divisions, 28.6% dans les CODIR de DO et 28.5% au COMEX. Le bilan de décision Unilatérale de 2022 indique que les mesures de correction salariale Femmes/ Hommes ont un budget insuffisant, qui ne permet pas de profiter à suffisamment de personnes.

En contrepartie, en 2022 a été organisé une semaine de l'égalité professionnelle femmes/ hommes, proposant des webinaires sur les nouveaux accords concernant l'égalité professionnelle ainsi que l'équilibre entre la vie privée et la vie professionnelle.

3. Contexte

Dans un premier temps, afin d'analyser de la meilleure manière possible le sujet, il est important de bien le comprendre. L'intitulé du projet est '***Création d'un portail intégré et automatisé***'.

Le tuteur de ce stage a ensuite communiqué des informations complémentaires sur les tâches à réaliser pour le bon déroulement du stage. En effet, parmi les exigences de ce projet, il faut qu'il soit déployable sur d'autres machines, car ce portail développeur doit être accessible de tout le monde, depuis n'importe quel endroit. Pour cela, plusieurs manières de procéder, mais celle qui semble la plus pertinente serait de déployer ce portail sur un cluster Kubernetes, pour des raisons qui seront expliquées a posteriori.

Partant d'un projet vierge, les premières missions dans un premier temps sont de comprendre les avantages d'une plateforme pour construire le produit souhaité.

Ensuite, de construire une instance de ce produit en local, sur la machine prêtée par l'entreprise, afin de pouvoir configurer toutes les fonctionnalités nécessaires à ses utilisateurs. Il sera possible d'y ajouter des plugins vers des applications externes, modélisables au bon vouloir de chacun. Il y aura plus de détails sur le fonctionnement et le choix de ces plugins dans les parties techniques.

Pour faire un récapitulatif de ce qui est demandé au commencement de ce projet :

- **Comprendre** le fonctionnement de Backstage
- **Trouver** les raisons d'utilisations de ce produit
- **Installer** une instance de Backstage en locale sur la machine hôte
- **Configurer** et **construire** une instance viable et fonctionnelle en local
- **Analyser** les besoins pour rendre l'application indispensable
- **Utiliser** Docker pour mettre en place une instance utilisable
- **Déployer** le produit sur un cluster Kubernetes.

4. Cahier des charges

4.1. Dates clés

Début du projet : 2 mai 2024
Fin du projet : 30 août 2024
Remise du rapport : 30 août 2024
Démonstration : 24 août 2024 (Orange)
Soutenance orale : 24 août 2024 (Orange) / 24 septembre 2024 (ISEN)

4.2. QQOQCCP

Quoi : réalisation d'un portail développeur intégré et automatisé pour stocker tous les documents et les ressources nécessaires à ses utilisateurs.

Qui : Le personnel d'Orange Innovation, en particulier les développeurs.

Où : partout, à condition d'avoir une connexion internet.

Quand : pas de date fixée pour le moment, si possible dès le mois de septembre.

Comment : installation et configuration d'une instance en locale afin de rajouter toutes les fonctionnalités. Ensuite, mise en place sur une image Docker pour gérer au mieux les versions. Et pour finir, déploiement sur un cluster Kubernetes.

Combien : pas de budget précis.

Pourquoi : centralisation des documents et des outils, afin d'améliorer l'efficacité de ses utilisateurs.

4.3. Objectif du projet

Pour faire un récapitulatif de ce qui est demandé au commencement de ce projet :

- **Comprendre** le fonctionnement de Backstage
- **Trouver** les raisons d'utilisations de ce produit
- **Installer** une instance de Backstage en locale sur la machine hôte
- **Configurer** et **construire** une instance viable et fonctionnelle en local
- **Analyser** les besoins pour rendre l'application indispensable
- **Utiliser** Docker pour mettre en place une instance utilisable
- **Déployer** le produit sur un cluster Kubernetes.

4.4. Périmètre du projet

L'utilisation de ce projet est faite pour être utilisée par les développeurs au sein du Groupe Orange. Le but étant d'améliorer la productivité globale au sein de l'entreprise.

Le rendu du prototype final devra être prête pour fin août 2024.

4.5. Description fonctionnelle des besoins

Pour décrire au mieux les besoins fonctionnelles du produit, les fonctions seront divisées en deux parties : fonctions principales et fonctions de contrainte :

Fonctions **principales** :

- Afficher le catalogue des services (composants/ API/ utilisateurs…)
- Afficher la documentation technique
- Personnaliser les plugins par les développeurs
- Gérer et visualiser les dépendances
- Automatiser des tâches (processus de déploiement et développement)
- Intégrer des outils DevOps (Kubernetes, Docker, Jenkins…)
- Centraliser les outils, les pipelines et les ressources nécessaires.

Fonctions de **contrainte** :

- Modéliser l'instance à son goût
- Complexité d'intégration
- Nécessiter de maintenance continue
- Vérifier l'obsolescence des plugins
- Sécuriser et gérer les accès à la plateforme.

4.6. Risques

- Obsolescence des plugins
- Problème de versions
- Gestions des dépendances
- Bonnes clés d'accès pour se connecter à certaines applications
- Mauvaise gestion des accès utilisateurs
- Mauvaise gestion des droits de modification de l'instance
- Confidentialités des données et des documents
- Problèmes entre les API et l'instance
- Pas suffisamment de puissance pour faire tourner l'instance.

4.7. Inventaire

Dans le cadre de ce stage, le projet Backstage ne nécessite pas beaucoup de matériel et de ressource. En effet, voici ce qui a été mis à disposition :

- Un ordinateur HP Intel Core vPRO i5
- Une clé PKI
- Un accès DEV ACCESS PKI

L'ordinateur est d'une importance cruciale, car sans celui-ci, le projet ne verrait pas le jour. Effectivement, le site d'Orange est protégé par des pare-feu et des proxys qui empêchent les machines extérieures à l'entreprise de fonctionner correctement. C'est la raison pour laquelle une clé PKI a été réquisitionné pour le projet. Ses fonctions principales sont de chiffrer les données présentes sur l'ordinateur, de fournir une authentification à l'aide d'un mot de passe différent et pour finir, une connexion au réseau Orange à distance, grâce à un VPN.

Enfin, l'accès DEV ACCESS PKI permet de contourner le proxy Orange, qui est bloquant pour les développeurs, lors de la configuration du portail développeur.

5. Développement technique

5.1. Introduction

Avec pour mission de réaliser un portail développeur intégré et automatisé, ce développement présentera toutes les étapes nécessaires à son accomplissement. Comme expliqué dans les étapes précédentes, il y aura 3 grandes étapes pendant le déroulement de ce projet :

- Développement local de l'instance de Backstage
- Mise en place sur une image Docker
- Mise en place sur un cluster Kubernetes

Ce sera l'ordre suivi pendant ces explications.

5.2. Développement local de l'instance Backstage

5.2.1. Installation des prérequis

La meilleure manière de créer sa première instance Backstage est tout simplement d'aller visiter leur site internet Backstage.io. Sur ce site, il est possible d'apprendre à installer et configurer de A à Z son portail développeur.

Dans un premier temps, il faut remplir les caractéristiques suivantes pour être sûr de pouvoir démarrer correctement ce projet :

- Avoir un environnement Linux disponible sur sa machine local
- Avoir les droits administrateurs sur sa machine locale
- Avoir installer "curl" ou "wget" sur son WSL
- Avoir Node.js d'installer afin d'utiliser nvm
- Avoir installer "yarn" dans son environnement Linux
- Avoir installer Docker
- Avoir installer git

Pour modifier, lancer et ajouter des dépendances de manière rapide et efficace, l'utilisation d'un environnement Linux, comme **WSL 2**, est recommandé. Mais pourquoi choisir cet outil-là ?

La mise en place de l'instance de Backstage se révèle être beaucoup de fichiers de configuration d'architecture. Dans ce type de schéma suivant, un environnement Linux est le plus recommandé. En effet, pour effectuer des configurations et gérer des systèmes de manière efficace et sécurisée, Linux est un choix privilégié par les administrateurs système. C'est justement ce que propose WSL 2. C'est un sous-système Windows pour Linux qui permet d'exécuter un environnement Linux sur une machine Windows, sans avoir besoin d'une machine virtuelle séparée ou d'un double démarrage.

En ce qui concerne les droits administrateurs, ils sont indispensables à avoir lors de la modification de fichiers de configuration. De plus, le Groupe Orange dispose de pare-feu bloquant l'installation de n'importe quelles applications, ainsi qu'un proxy pour accéder aux navigateurs internet. Pour ces différentes raisons, il est très utile de se munir des droits administrateurs.

L'installation de "wget" ou "curl" servent pour tester le bon fonctionnement de certaines modifications systèmes, lorsqu'il faut transférer des données depuis ou vers un serveur.

Le démarrage de Node JS fait partie des prérequis pour faire fonctionner l'instance de Backstage. Cependant, dans le cadre du Groupe Orange, il est aussi utilisé pour faire fonctionner **Docusaurus**. Mais quel est le principe de cette application ?

Le principe est simple. En tant que DevOps, il y a beaucoup de documentation à renseigner. Docusaurus est sans doute l'une des meilleures solutions pour les réaliser. Il suffit de déposer le code dans un dépôt git et d'y ajouter un pipeline CI/CD pour construire et déployer le site. C'est ce qu'on appelle de la "doc as code". Docusaurus est un générateur de site statique développé par Meta Open Source sous licence MIT et basé sur React. Simple à installer et facile d'utilisation, c'est un outil indispensable dans le milieu des développeurs.

L'utilisation de "yarn" dans ce projet est importante, car c'est grâce à cet outil-là que l'installation de toutes les dépendances liées à l'instance est possible.

Pour finir, Docker et git sont utilisés pour certaines fonctionnalités, comme l'authentification ou encore la mise en place de de la documentation.

5.2.2. Plusieurs démarrages possibles

La première approche de ce projet, et comme il est recommandé de le faire sur le tutoriel du site de Backstage, serait de commencer un projet vierge pour comprendre son fonctionnement et ses enjeux. Cependant, ce n'est pas la seule méthode viable pour répondre à ce besoin. En effet, au cours de ce stage, 3 méthodes ont été abordées, pour tester celle qui répondrait le plus à la demande. Ces 3 manières de procédés sont les suivantes :

- Instance vierge, créée en ligne de commande
- Clone d'un dépôt GitHub déjà existant
- instance Backstage avec RedHat OpenShift.

Chacune de ces méthodes à ses avantages et ses défauts. Ceux-ci vont être explicités ci-dessous.

5.2.2.1. Instance vierge

L'intérêt majeur de cette technique sert pour un développeur débutant sur l'application Backstage, car le projet est vierge. Il est donc plus facile de comprendre le fonctionnement, et de voir ce qu'il est possible d'ajouter.

5.2.2.1.1. Création de l'instance

En se servant de la partie sur les prérequis, il ne reste plus qu'à créer l'instance. Pour cela, rien de plus facile. Ouvrez le terminal WSL 2, et entrez la commande suivante :

- ***npx @backstage/create-app@latest***

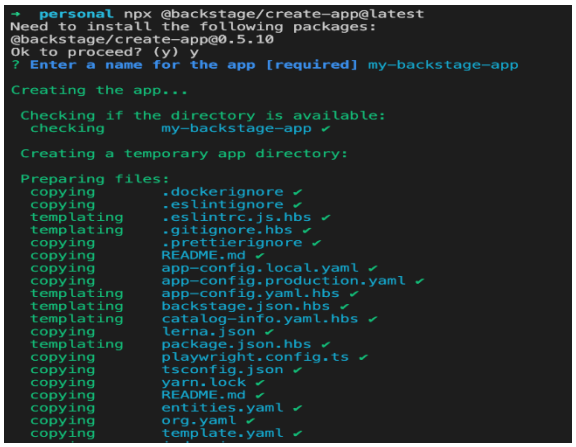
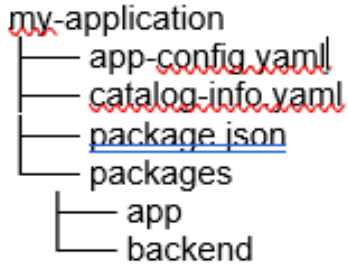


Figure 2 - Création de l'instance Backstage

La commande que vous avez rentrée vous demande le nom d'un répertoire (Ex : my-application pour la suite), qui représentera le répertoire courant de votre instance. Si tout fonctionne correctement, vous devriez voir une liste de fichiers en téléchargement, comme sur l'image ci-dessous. Tous ces fichiers permettent à l'instance de fonctionner correctement.

5.2.2.1.2. Structure du répertoire courant

Comme expliqué précédemment, le nom d'exemple du répertoire courant est 'my-application'. Voici la liste des fichiers présents dans ce répertoire, ainsi que quelques détails sur leurs fonctions. La structure est la suivante :



app-config.yaml : C'est le fichier de configuration principale de l'instance. Il est possible d'y retrouver les configurations BDD, d'authentification, d'intégration...

catalog-info.yaml : Fichier de configuration du catalogue des entités.

package.json : fichier où sont renseignées toutes les dépendances.

packages/ : répertoire racine des fichiers de configuration du frontend et backend.

packages/app : répertoire de configuration du frontend.

packages/backend : répertoire de configuration du backend.

5.2.2.1.3. Démarrage de l'instance

Une fois les installations terminées, comme le montre l'image ci-dessus, voici les commandes à effectuer pour démarrer l'application :

- ***cd my-application*** (rentrer dans votre répertoire courant)
- ***yarn dev*** (démarrage de l'instance)

La commande "cd" permet de se déplacer dans les répertoires, sous Linux. Tandis que la commande "yarn dev" est renseignée dans le fichier de configuration package.json, et lance deux commandes en simultanées :

- ***yarn start***
- ***yarn start-backend***

C'est grâce à ces deux commandes que l'instance se lance. En effet, "yarn start" permet au frontend de démarrer, et "yarn start-backend" de démarrer la partie de backend de l'instance.

Ces deux commandes sont lancées séparément mais ont une importance cruciale. En effet, lancer seulement le frontend ne permet pas d'avoir les informations concernant le contenu des catalogues, des documents… alors que démarrer le backend uniquement ne permettrait pas d'avoir un visuel sur les configurations effectuées. Si vous voyez s'afficher "*webpack compiled successfully*", l'instance est démarrée et disponible avec l'url **http://localhost:3000**.

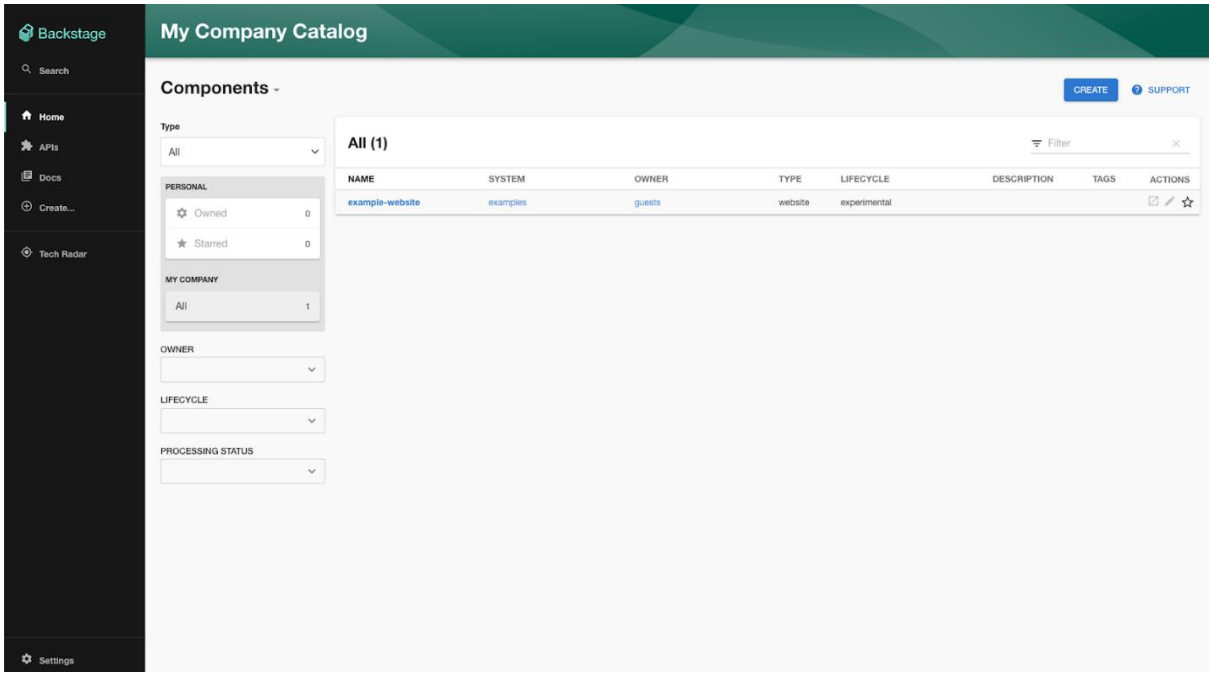


Figure 3 - Visualisation de l'instance vierge

5.2.2.1.4. Messages du backend

Il est d'ailleurs possible de voir sur le terminal si le backend reçoit des erreurs ou non, comme sur l'exemple suivant.

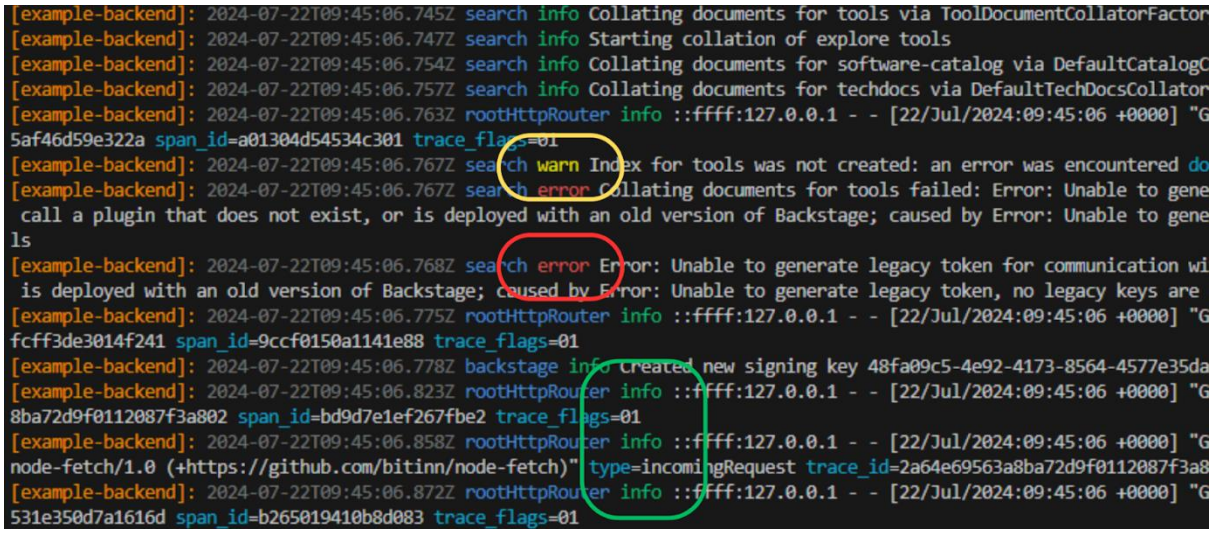


Figure 4 - Messages du backend de l'instance

Il existe 3 types d'informations :

- Les messages d'avertissement en jaune.
- Les messages d'erreur en rouge.
- Les messages d'informations en vert.

5.2.2.2. Instance clonée depuis dépôt existant

Dans cet exemple-là, les principes fondamentaux sont les mêmes. Ce qui diverge entre ces deux projets va être leur contenu. Initialement, la première instance était vierge, sans aucune modification depuis sa création. Tandis que dans l'instance clonée depuis le dépôt GitHub, il y aura un bon nombre de fonctionnalités déjà existantes qui sont implantées dans celle-ci. Voici l'atout majeur de l'utilisation du deuxième exemple, qui peut faire gagner un temps considérable.

5.2.2.2.1. Création de l'instance

Sachant que l'instance a déjà été créée, il suffit de faire un copié collé du dossier GitHub où se trouve le projet. Dans cette situation, retournez sur votre WSL 2, et rentrez les commandes suivantes :

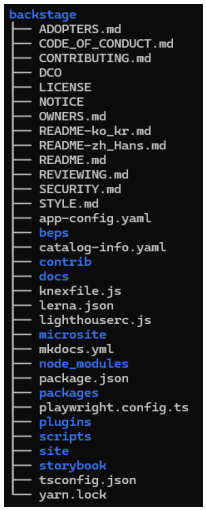
- ***git clone* https://github.com/backstage/backstage.git**

Cette ligne de commande récupère automatiquement tous les fichiers et dossiers du répertoire backstage stocké sur GitHub. Vous aurez ensuite une instance complète sur votre machine locale. Cependant, les lignes de configuration ne sont pas encore adaptées à votre machine. Pour cela, il faudra faire ceci :

- ***yarn install***

C'est grâce à elle que les dépendances et les installations se feront correctement. Une fois cette tâche effectuée, l'instance sera viable et fonctionnelle. Sachant que c'est une copie d'un répertoire existant, le nom du répertoire courant sera "backstage" pour les prochaines étapes.

5.2.2.2.2. Structure du répertoire courant



Comme expliqué en début de partie, cette méthode présente une autre manière de commencer ce projet. En effet, le nombre de fonctionnalités étant plus importantes, cela explique que la configuration de sa structure soit plus grande également. Pour un souci de place, il ne sera présenté que les éléments de niveau 1 et 2 du répertoire courant (trop de sous-répertoires).

Comme pour l'instance vierge, il est possible de retrouver les fichiers principaux, à savoir app-config.yaml, catalog-info.yaml, package.json ainsi que le répertoire packages. Les autres fichiers seront expliqués

Figure 5 - Structure répertoire courant

5.2.2.2.3. Démarrage de l'instance

Même chose que dans la partie 2.2.1.3, le lancement de l'instance pourra se faire une fois situer dans votre répertoire courant.

Quand vous verrez le message "webpack compiled successfully", plus qu'à se rendre à l'url suivante : http://localhost:3000, où il sera possible de voir l'instance.

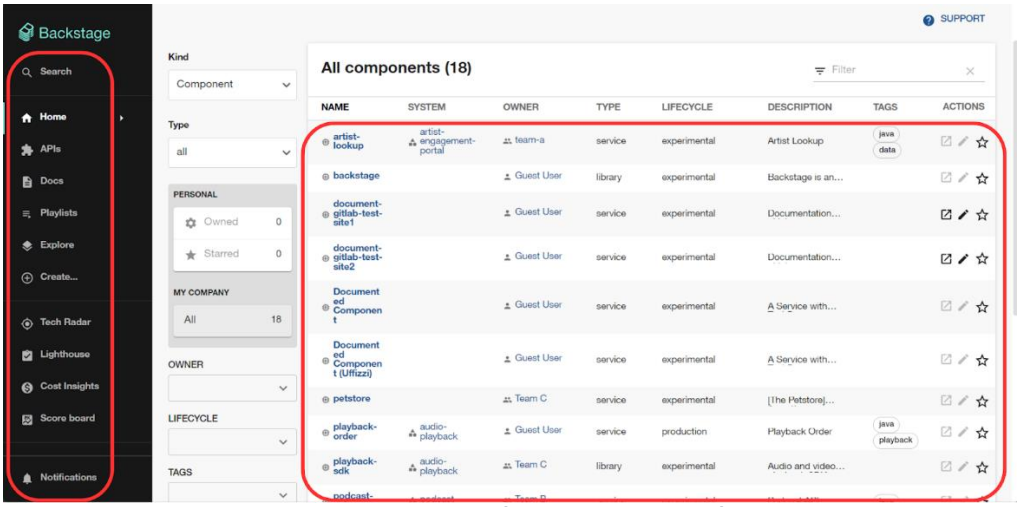


Figure 6 - Nouveautés de l'instance clonée

En comparaison avec l'instance vierge, les espaces entourés en rouge sont déjà préremplis. En effet, dans le menu (partie gauche entourée en rouge), le nombre d'options est supérieure à celle de l'instance vierge. Idem pour le rectangle de droite concernant le catalogue des composants, qui est bien plus fourni que celui de l'autre exemple.

5.2.2.2.4. Différences majeures

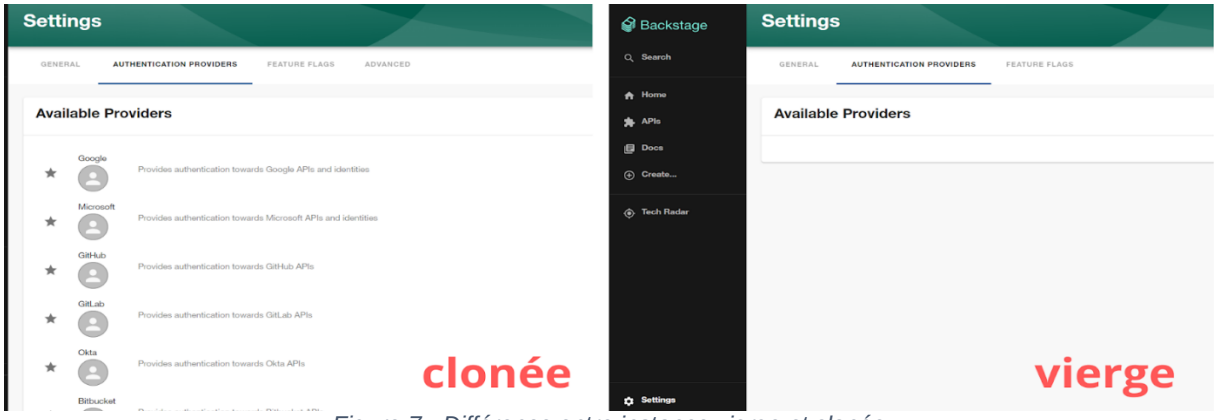


Figure 7 - Différence entre instance vierge et clonée

Sur l'image ci-dessus, voici les différences en termes d'authentification possible, entre les deux instances. Dans l'un des cas, le nombre d'authentification possible, comme avec GitLab ou GitHub par exemple, est vraiment conséquent. Dans la deuxième situation, comme l'instance est vierge, il faut faire les paramétrages pour que l'instance puisse accorder l'authentification.

5.2.2.3. Instance RedHat Openshift

Cette manière de procéder est totalement différente des 2 précédentes. En effet, dans ce cadre-là, il n'y a pas besoin de prérequis pour démarrer le projet. C'est RedHat qui va se charger des principales configurations pour que l'instance se lance avec le moins de problèmes possibles. L'avantage de cette méthode est qu'elle est très facilement réalisable, même sans connaissance en informatique, grâce au tutoriel très bien fourni sur le site de RedHat Openshift.

5.2.2.3.1. Création de l'instance

Pour faire simple, et en y intégrant les parties les plus importantes du tutoriel, il faut créer un projet sur l'interface de RedHat Openshift, et ensuite installer le composant Développeur Hub, comme le montre l'image ci-dessous.

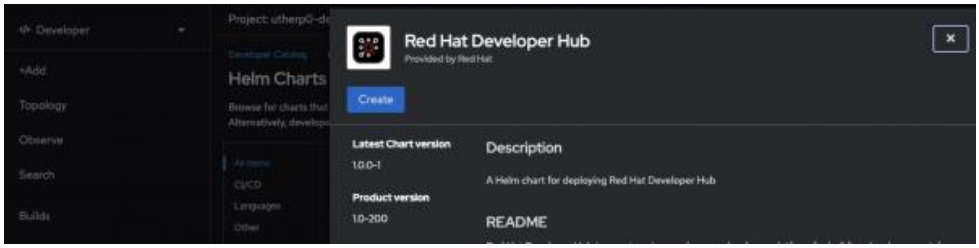


Figure 8 - Création du développeur hub

Une fois cette installation terminée, vous devriez voir apparaître la topologie de votre instance, comme sur l'image ci-dessous.

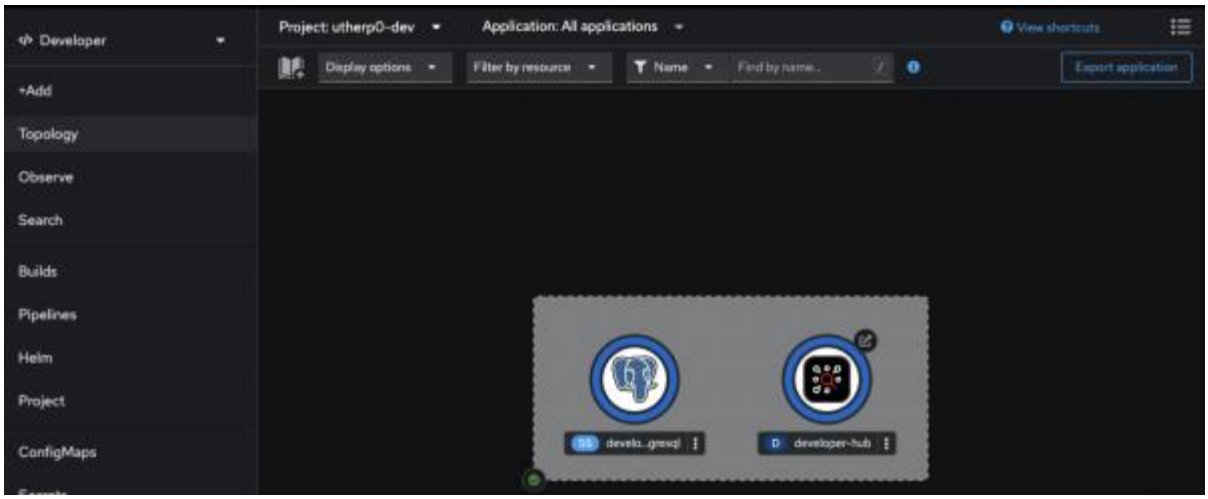


Figure 9 - Topologie de l'instance

Il est possible de voir à la fois la base de données, avec l'éléphant représentant PostgreSQL, et le module Développeur Hub. Les ronds sont en bleus foncés une fois en marche, et se trouveront blanc s'ils sont éteints.

5.2.2.3.2. Démarrage de l'instance

Dans le cas présent, la plateforme est active. Il n'y a donc plus qu'à aller sur l'instance RedHat, en cliquant sur la petite excroissance en haut à droite du Développeur Hub. Ceci ramène directement à l'application attendue, à l'adresse suivante :

- RedHat Développeur Hub

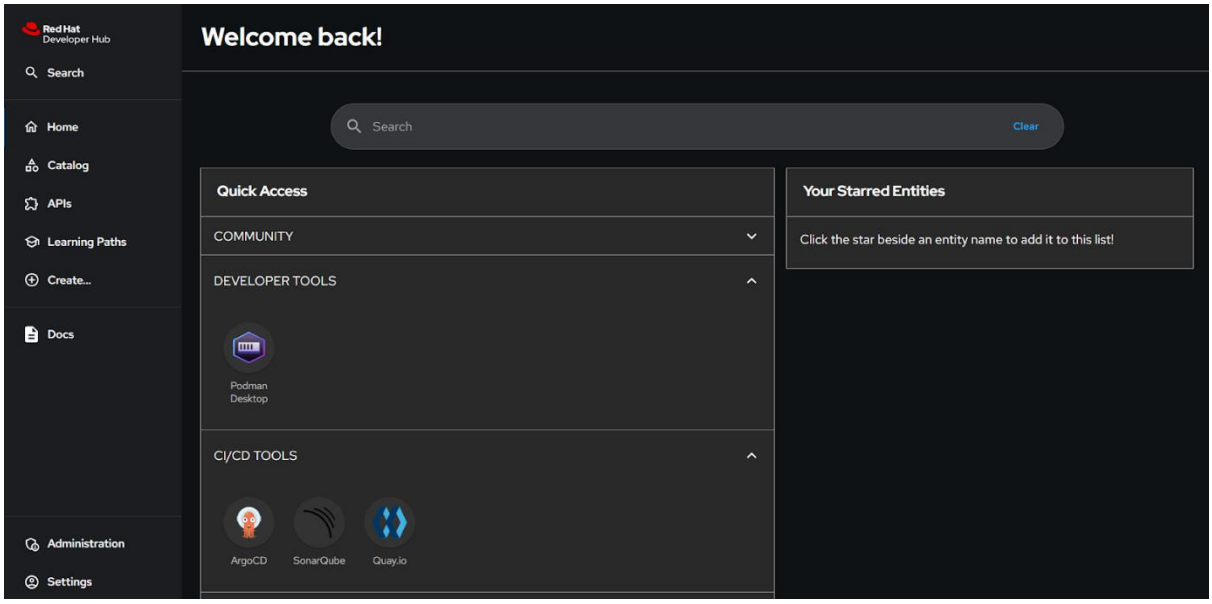


Figure 10 - Instance RedHat Openshift

Voici la représentation graphique de cette instance, en seulement quelques clics. Pour éteindre celle-ci, il suffit de faire passer le pod à 0, et inversement le faire passer à 1 pour la rallumer.

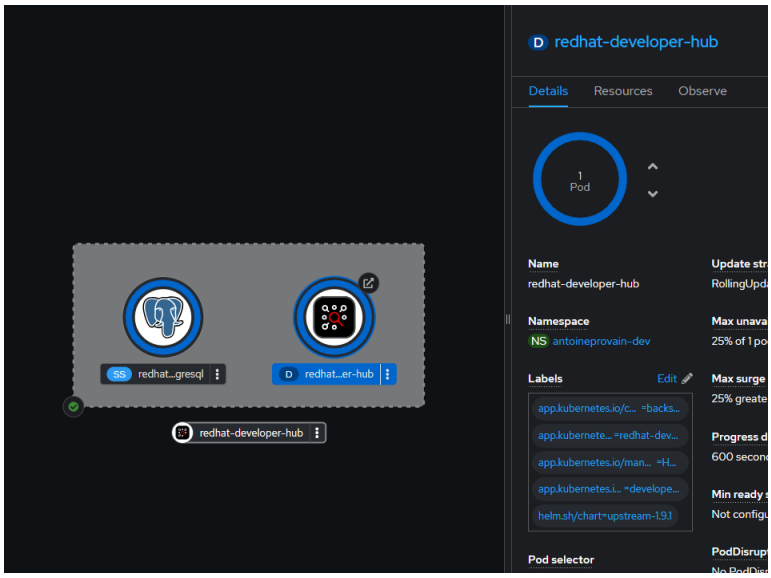


Figure 11 – Pod d'allumage et de stoppage de l'instance

Utiliser RedHat Openshift pour démarrer sur Backstage semble être une bonne option, cependant, ce n'est pas un produit open source.

Pour **conclure** sur cette partie, le type d'instance qui a été choisi pour réaliser toutes les modifications et configurations de base, pour permettre d'avoir le meilleur portail développeur, est l'instance clonée depuis le répertoire déjà existant. Pour des raisons pratiques et un projet en open source, c'est celle-ci qui a été préférée.

5.2.3. Modifications et configurations associées

Dans le contexte du développement du portail développeur, il a fallu apporter des modifications à l'instance afin qu'elle réponde aux besoins de ses utilisateurs.

5.2.3.1. Configuration de la base de données

La configuration d'une base de données pour le fonctionnement de la structure n'est pas obligatoire. Cependant, certaines applications peuvent en nécessiter, d'où l'intérêt de la configurer pour ces cas particuliers. L'exemple le plus facile : l'utilisation d'une base de données par un plugin personnalisé ou non. Plus de détails seront donnés dans la partie en question.

Lors de l'utilisation de Backstage, c'est PostgreSQL qui est configuré par défaut. Pour la suite des étapes, et par simplicité, c'est celle-là qui va être utilisée.

Voici le protocole à suivre pour vérifier si votre PostgreSQL fonctionne correctement. Ouvrez un Terminal WSL 2, et rentrez la commande suivante :

- ***sudo -u postgres psql***

```
psql (14.12 (Ubuntu 14.12-0ubuntu0.22.04.1))
Type "help" for help.

postgres=#
```

Si cela fonctionne correctement, vous devriez voir apparaître les lignes ci-dessous.

Figure 12 - Base de données PostgresSQL

Dans le cas contraire, tapez la commande suivante pour installer PostgreSQL et réitérer la commande du dessus :

- ***sudo apt-get install postgresql***

Une fois ces étapes finies, il faut rajouter les configurations concernant la base de données dans le fichier de configuration my-application/app-config.yaml.

Voici un aperçu de ce à quoi peut ressembler le fichier de configuration.

app-config.yaml

```yaml
backend:
  database:
    client: better-sqlite3
    connection: ':memory:'
    # config options: https://node-postgres.com/apis/client
    client: pg
    connection:
      host: ${POSTGRES_HOST}
      port: ${POSTGRES_PORT}
      user: ${POSTGRES_USER}
      password: ${POSTGRES_PASSWORD}
```

Figure 13 - Configuration de la BDD dans app-config.yaml

Dans le code donné ci-dessus, certaines des valeurs sont entourées de "***${ … }***". Cela permet d'intégrer des valeurs depuis un fichier externe vers le fichier de configuration. Dans ce cas précis, un fichier nommé "***.env***" a été créé pour répondre à ce besoin. Le contenu de celui-ci à la forme suivante.

```
export POSTGRES_HOST=...
export POSTGRES_PORT=...
export POSTGRES_USER=...
export POSTGRES_PASSWORD=...
```

Figure 14 - Fichier.env

Les "**...**" sont à remplacer par les valeurs qui se trouvent entre arobase dans votre fichier de configuration. Mais pourquoi est-ce nécessaire ?

La réponse est simple : plus sécurisée. En effet, le fait de passer par un fichier où sont stockées toutes les valeurs évite d'écrire les valeurs en dur dans le code. Les fichiers de configuration sont faciles à pirater, il est donc préférable de ne rien mettre dans ce fichier, qui puisse compromettre la sécurité de ce projet, aussi bien pour les utilisateurs que pour les développeurs de l'instance.

List of databases

Name	Owner	Encoding	Collate	Ctype	Access privileges
backstage_plugin_app	postgres	UTF8	C.UTF-8	C.UTF-8	
backstage_plugin_auth	postgres	UTF8	C.UTF-8	C.UTF-8	
backstage_plugin_catalog	postgres	UTF8	C.UTF-8	C.UTF-8	
backstage_plugin_permission	postgres	UTF8	C.UTF-8	C.UTF-8	
backstage_plugin_proxy	postgres	UTF8	C.UTF-8	C.UTF-8	
backstage_plugin_scaffolder	postgres	UTF8	C.UTF-8	C.UTF-8	
backstage_plugin_search	postgres	UTF8	C.UTF-8	C.UTF-8	
backstage_plugin_techdocs	postgres	UTF8	C.UTF-8	C.UTF-8	
postgres	postgres	UTF8	C.UTF-8	C.UTF-8	
template0	postgres	UTF8	C.UTF-8	C.UTF-8	=c/postgres + postgres=CTc/postgres
template1	postgres	UTF8	C.UTF-8	C.UTF-8	=c/postgres + postgres=CTc/postgres
test	antoine	UTF8	C.UTF-8	C.UTF-8	=Tc/antoine + antoine=CTc/antoine

Figure 15 - Visualisation base de données PostgreSQL

Après ces multiples modifications et ajouts, il est possible de voir le rendu PostgreSQL, comme le voici, avec la commande "**\l**".

5.2.3.2. Configuration de l'authentification

L'authentification vers ces 2 applications est utile mais pas nécessaire au fonctionnement de l'instance. Dans certains cas, il sera plus pratique de les avoir et c'est la raison pour laquelle elles seront configurées.

Chez Orange Innovation, l'outil principal de stockage des fichiers relatif aux projets est GitLab, atteignable sous le nom GitLab.tech.orange. Cet outil fonctionne relativement pareil qu'une application comme GitHub, avec du partage de fichiers et de la collaboration entre équipes.

Cela peut être aussi perçu comme un problème car l'instance de backstage est paramétrée avec GitHub. En effet, pour remplir les catalogues de composants, il faut renseigner certains fichiers de configuration, qui sont hébergés sur cette application. C'est la raison pour laquelle les 2 applications ont été ajoutées au projet.

Pour refaire un point sur l'objectif. Il faut récupérer toutes les documentations de tous les projets pour les centraliser sur l'instance Backstage. Cependant, ces documents se trouvent sur GitLab.tech.orange. Il est donc important de s'authentifier et d'intégrer GitLab à ce projet.

5.2.3.2.1. GitHub

5.2.3.2.1.1. Création de l'application GitHub

Avant de faire les modifications sur le fichier de configuration de l'instance, il faut commencer par créer, depuis GitHub, une nouvelle application appelée "***OAuth application***". Sans cette étape-là, il ne sera pas possible de s'authentifier. Voici les principales informations qui doivent être rentrées.

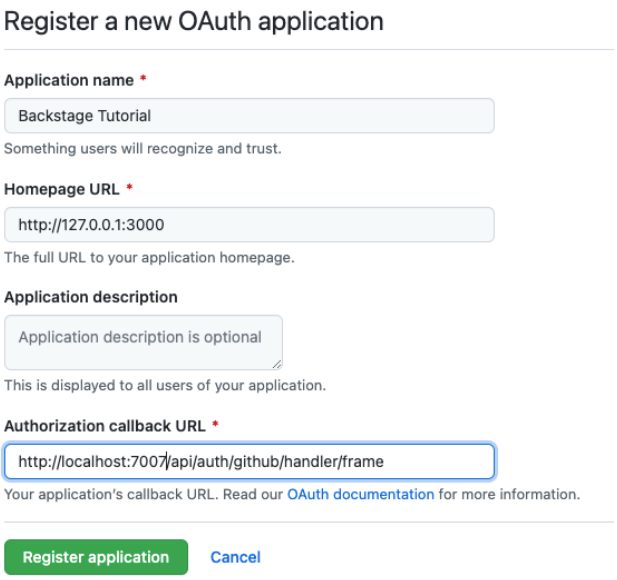


Figure 16 - Création de l'application GitHub

Il faut alors rentrer "***Homepage URL***", qui correspond à l'adresse de l'instance, ainsi que "***Authorization callback URL***", qui renvoie vers une page GitHub afin de valider les droits d'accès à l'utilisateur pour l'utilisation de l'instance.

Une fois validé, GitHub donnera accès à 2 données essentielles pour l'authentification :

- Le client ID
- Le client Secret

Ce sont grâce à ces données que l'utilisateur sera en mesure de s'authentifier à GitHub.

5.2.3.2.1.2. Configuration de l'instance

La première chose à faire est d'aller dans le fichier my-application/app-config.yaml et de modifier la partie d'authentification, pour y ajouter celle de GitHub, comme voici.

app-config.yaml

```yaml
auth:
  # see https://backstage.io/docs/auth/ to learn about auth providers
  environment: development
  providers:
    github:
      development:
        clientId: YOUR CLIENT ID
        clientSecret: YOUR CLIENT SECRET
```

Figure 17 - Configuration de l'authentification dans app-config.yaml

Sur cette image, vous pouvez voir qu'il y a 2 paramètres importants :

- clientId
- clientSecret

Ce sont les 2 données qui ont été transmises par GitHub lors de la création de l'application. Pour rentrer ces données dans le fichier de configuration, même principe que pour la base de données, il faut remplacer "*YOUR CLIEND ID*" et "*YOUR CLIENT SECRET*" par les valeurs :

"***${ AUTH_GITHUB_CLIENT_ID }***" et "***${ AUTH_GITHUB_CLIENT_SECRET }***"

Plus qu'à mettre les valeurs correspondantes de ces variables dans le fichier "***.env***", comme expliqué précédemment, et le tour est joué.

Il ne reste plus qu'à ajouter le code ci-dessous dans le frontend pour que l'instance demande à son utilisateur, au démarrage de l'instance, quelle méthode d'authentification choisir.

```
components: {
  SignInPage: props => (
    <SignInPage
      {...props}
      auto
      provider={{
        id: 'github-auth-provider',
        title: 'GitHub',
        message: 'Sign in using GitHub',
        apiRef: githubAuthApiRef,
      }}
    />
  ),
},
```

Figure 18 - Ajout dans le frontend

Grâce à ces quelques lignes dans le frontend, l'utilisateur aura une nouvelle page qui s'affichera, en demandant une authentification GitHub.

Ce qui est important à préciser, c'est que si l'authentification ne marche pas, il existe un utilisateur "**guest**", qui ne requiert aucune connexion, et permet à chacun de se connecter à l'instance. Le désavantage de se connecter avec cette méthode-là, c'est que le catalogue des composants n'est plus associé à un utilisateur, et affiche donc un composant par défaut.

5.2.3.2.2. GitLab

5.2.3.2.2.1. Création de l'application GitLab

Avant de faire les modifications sur le fichier de configuration de l'instance, il faut commencer par créer, depuis GitLab, une nouvelle application. Sans cette étape-là, il ne sera pas possible de s'authentifier. Voici les principales informations qui doivent être rentrées.

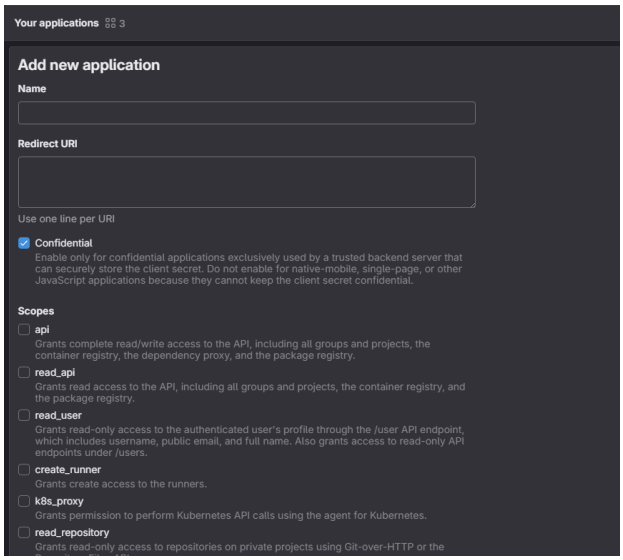


Figure 19 - Création de l'application GitLab

Même principe que pour GitHub, il faut entrer une url de redirection, appelé "***Redirect URI***". Ensuite, il y a une liste de droits à cocher concernant l'application, au niveau du "***scopes***".

Dans ce cas, les droits les plus importants sont les suivants :

- **api**, **read_api**, **read_user**, **read_repository**, **write_repository**, **openid**, **profile** et **email.**

Même fonctionnement qu'avec GitHub. Une fois l'application créée, il vous sera fourni une "***application id***" et un "***secret***". C'est ces 2 données que vous serez amenés à configurer dans l'instance de Backstage.

5.2.3.2.2.2. Configuration de l'instance

Même exercice pour GitLab. Il est nécessaire de retourner dans le fichier de configuration app-config.yaml, et d'y ajouter ceci.

```yaml
gitlab:
  development:
    clientId: ${AUTH_GITLAB_CLIENT_ID}
    clientSecret: ${AUTH_GITLAB_CLIENT_SECRET}
    audience: ${GITLAB_BASE_URL}
    # callbackUrl: http://localhost:7007/api/auth/gitlab/handler/frame
    signIn:
      resolvers:
        - resolver: emailMatchingUserEntityProfileEmail
        - resolver: emailLocalPartMatchingUserEntityName
        - resolver: usernameMatchingUserEntityName
```

Cette image illustre les modifications nécessaires afin de permettre la connexion à GitLab. Les variables à renseigner sont les suivantes :

- ***AUTH_GITLAB_CLIENT_ID***
- ***AUTH_GITLAB_CLIENT_SECRET***

Ce sont les valeurs transmises par GitLab au moment de la création de l'application.

- ***GITLAB_BASE_URL***

Le groupe Orange se sert de GitLab.tech.orange pour stocker et travailler en équipe. Cependant, cette application est spécifique à l'entreprise. En effet, c'est GitLab.com qui est le plus utilisé de manière générale, en dehors du groupe. C'est la raison pour laquelle il est essentiel de rajouter la variable suivante, afin de lui donner l'url de GitLab.tech.orange. Dans le cas contraire, cela tenterait de se connecter sur GitLab.com.

5.2.3.3. Configuration de TechDocs

Cette nouvelle étape dans le développement de l'application représente une partie importante du rendu final. En effet, dans les enjeux que pouvait apporter une plateforme telle que Backstage, il y avait le fait de centraliser des documents, pour faire gagner du temps ainsi que de la productivité aux utilisateurs.

Dans le cadre d'Orange Innovation, beaucoup de documents sont hébergés sur la plateforme GitLab.tech.orange, regroupant ainsi de nombreux documents techniques. Cependant, il est parfois compliqué de retrouver rapidement les documents d'un sujet spécifique.

Techdocs se voit alors comme une sorte de catalogue central des documents techniques.

5.2.3.3.1. GitLab.com

Pourquoi commence-t-on par GitLab.com pour configurer le catalogue de documentation ?

La raison est simple. Accéder GitLab.com est plus facile car les configurations nécessaires sur Backstage sont simplifiées.

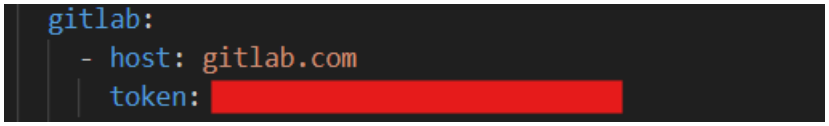


Figure 20 - Configuration des TechDocs Gitlab.com

Pour accéder au contenu de GitLab.com, voici la partie à renseigner dans le fichier app-config.yaml. Le « token » permet à son utilisateur de renseigner son compte GitLab.com, pour ensuite utiliser ses documents, afin de les retrouver sur l'application Backstage.

Mais cela ne suffit pas pour rajouter des documents sur le catalogue central. Voici à quoi ressemble l'interface, avec seulement les documents par défaut dans l'instance.

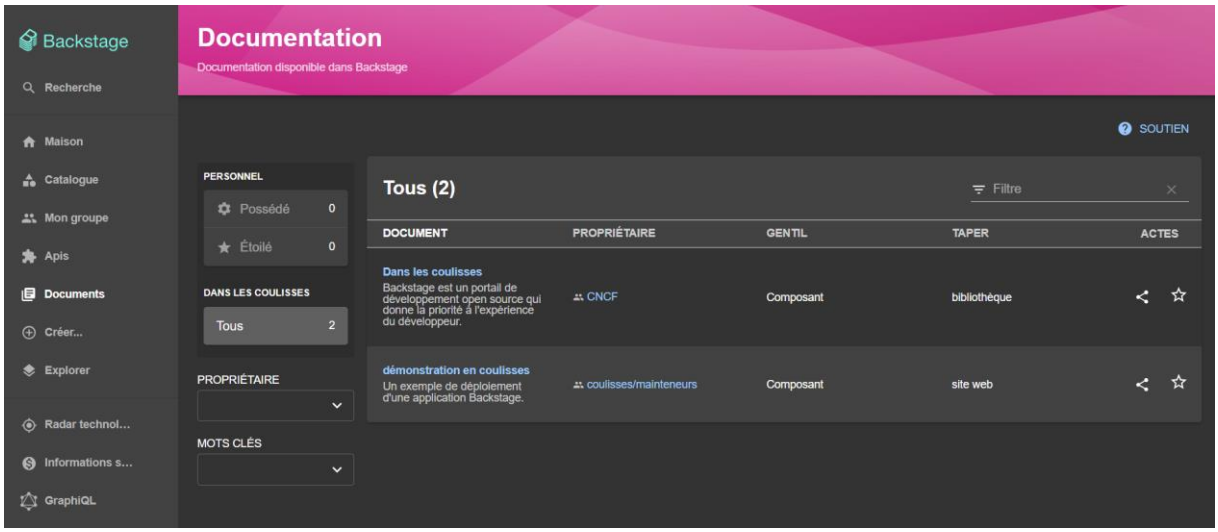


Figure 21 - Catalogue de la documentation

Comme il est possible de le voir, voici uniquement les documents par défaut. Il est possible de rajouter des documents, autant que vous en avez à mettre, et ils s'afficheront tous dans cette page « Documents ».

Pour pouvoir visualiser le contenu d'un document GitLab.com, il faut ensuite rajouter la ligne suivante dans app-config.yaml :

```
- type: url
  target: https://gitlab.com/testing50720/docs_backstage/-/blob/main/catalog-info.yaml
```

Figure 22 - Incorporation des documents gitLab.com

Ce lien représente un lien direct vers le répertoire GitLab qui contient les documents qui seront exposés dans le catalogue central. Pour se faire, il faut que ce lien soit dirigé vers un fichier « catalog-info.yaml », qui à son tour indique à l'instance où sont situés les documents à répertorier.

catalog-info.yaml 368 o Blame Modifier Remplace

```yaml
apiVersion: backstage.io/v1alpha1
kind: Component
metadata:
  name: document-gitlab-test
  description: Documentation gitlab
  annotations:
    #backstage.io/techdocs-ref: url:https://gitlab.tech.orange/antoine.provain/docs-backstage/-/tree/main/docs
    backstage.io/techdocs-ref: dir:.
spec:
  type: service
  owner: user:guest
  lifecycle: experimental
```

Figure 23 - Fichier catalog-info.yaml de GitLab.com

La ligne intéressante est celle qui indique « dir : . ». Cela signifie que les documents se trouvent dans le répertoire racine où se situe le fichier catalog-info.yaml, dans un dossier docs. Sur l'instance, il sera ensuite possible de voir un nouveau document, nommé document-gitlab-test, comme il est indiqué par l'indice « name » sur l'image ci-dessus.

5.2.3.3.2. GitLab.tech.orange

Même fonctionnement que pour GitLab.com, cependant, il faut configurer différemment le fichier de configuration principal.

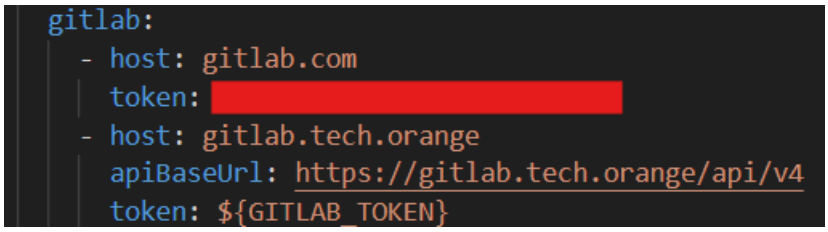


Figure 24 - Configuration de l'intégration GitLab.tech.orange

En effet, pour cette partie, il est nécessaire de rajouter le lien vers l'API de GitLab.tech.orange, car sans cela, il ne serait pas possible de récupérer les informations permettant la lecture des documents.

Une fois cette étape réalisée, il n'y a plus qu'à modifier app-config.yaml pour y ajouter le lien vers le catalog-info.yaml, qui indiquera à son tour où sont situés les documents qui seront affichés sur l'instance.

Figure 25 - Lien vers GitLab.tech.orange

Avec cette modification, il sera possible d'identifier deux documents provenant des ressources gitlab d'orange, avec les noms suivants :

- ***Document-gitlab-test-site1***
- ***Document-gitlab-test-site2***

Figure 26 - Nouveaux documents dans le catalogue

6. Gestion de projet

Pour mettre en œuvre ces différentes étapes, un diagramme de Gantt a été réalisé pour respecter le fonctionnement des méthodes AGIL. Ce programme sera séparé en trois grandes étapes, qui sont :

1. Compréhension et configuration de l'instance et de ses utilités
2. Configuration de l'image Docker pour accueillir l'instance
3. Déploiement du produit final sur le cluster Kubernetes.

Pour rentrer plus dans le détail, nous mettrons quelques sous-parties, qui devront être réalisées dans des sprints de deux semaines. Voici à quoi devrait ressembler le Gantt en début de projet.

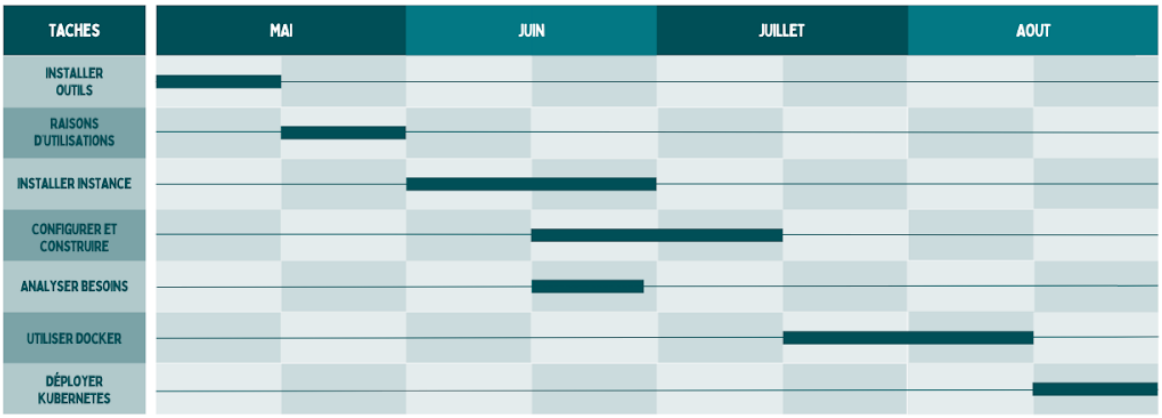


Figure 27 - Diagramme de Gantt

Ce diagramme est prévisionnel, et a été réalisé en début de projet. Il y aura un second diagramme au moment de la fin de projet, pour visualiser ce qui a été fait et ce qu'il reste à faire.

Figure 28 - Diagramme de Gantt à postériori

7. Conclusion

En conclusion, les objectifs initiaux du projet étaient de comprendre et trouver les raisons d'utilisation d'un produit tel que Backstage. De faire les configurations nécessaires au bon fonctionnement de cette application, en prenant en compte les besoins de chacun. Puis de mettre en place cette instance à la fois sur une image Docker dans un premier temps, et ensuite sur un cluster Kubernetes, afin qu'elle soit accessible par tous et à chaque instant.

Concernant les attentes de début de projet, les parties de mises en service du projet n'ont pas pu être réalisées dans les temps. Pour ce qui est des parties de compréhension et de configuration du produit, elles ont été réalisées avec succès.

La fin de ce stage a été précipitée pour des raisons de santé, à partir du 26 août 2024. Au moment de cet arrêt, la majeure partie du planning vu sur le diagramme de Gantt avait été tenu. Cependant, les étapes suivantes sont restées sans suite. Sans ces perturbations, l'avancement de ce projet aurait été encore meilleure.

La partie de compréhension du projet et de son objectif est clair, en partie grâce aux différentes vidéos et explications mises en place par Spotify, créateur du projet Backstage. En effet, la centralisation des outils et des documents sur une même plateforme permet un gain de temps considérable. L'efficacité des ingénieurs qui utilisent l'application est donc démultiplié. Dans le cadre d'Orange, de nombreux documents techniques sont présents sur un GitLab interne, mais ne sont pas forcément facile à retrouver. De plus, la personnalisation de l'instance Backstage est facile et bien documentée, ce qui permet par exemple d'intégrer des plugins personnalisés vers différentes applications extérieures à Backstage, ce qui est un avantage majeur.

Pour que l'application réponde au mieux à l'utilisateur, il a fallu configurer des paramètres dans certains fichiers. Parmi ces configurations, il y a eu l'ajout de l'authentification avec les applications GitLab et GitHub, la prise en charge des données avec PostgreSQL, l'intégration des documents techniques externes vers l'interface Backstage, et pour finir avec l'ajout de certains plugins personnalisés comme le plugin GitLab et ChatGPT.

Cependant, en prenant en compte le changement de versions entre la documentation et l'évolution des mises à jour de l'instance de Backstage, il y avait parfois certaines incohérences, qui ont mené à de grosses pertes de temps. L'exemple le plus facile à démontrer serait de parler du backend. Entre plusieurs versions différentes de l'application Backstage, il y a eu une nouvelle façon de configurer le backend de l'instance, ce qui change considérablement les problèmes rencontrés.

Ce projet de portail développeur va continuer en interne après le stage. En effet, il a été déposé sur le GitLab interne d'Orange afin que la reprise de ce projet soit la plus facile possible.

Parmi les améliorations possibles, il aurait fallu ajouter un plugin permettant de voir les clusters Kubernetes reliés au compte connecté et de voir les informations les concernant. Pour simplifier la récupération des documents sur l'espace commun d'Orange, il aurait fallu écrire un script réunissant tous ces documents et les placer dans un répertoire particulier, ce qui aurait facilité l'affichage sur l'interface centralisée.

8. Bibliographie

Références

GitHub, s.d. *Commandes GitHub.* [En ligne]
Available at: https://gist.github.com/aquelito/8596717

GitHub, s.d. *Plugin GitLab.* [En ligne]
Available at: https://github.com/immobiliare/backstage-plugin-gitlab

GitLab.com, s.d. *Documents GitLab.* [En ligne]
Available at: https://gitlab.com/testing50720/docs_backstage

GitLab, s.d. *Commandes GitLab.* [En ligne]
Available at: https://docs.gitlab.com/ee/topics/git/commands.html

Kinsta, s.d. *Commandes Linux.* [En ligne]
Available at: https://kinsta.com/fr/blog/commandes-linux/

OpenShift, R., s.d. *Protocole d'apprentissage RedHat Openshift.* [En ligne]
Available at: https://developers.redhat.com/learning/learn:openshift:install-and-configure-red-hat-developer-hub-and-explore-templating-basics/resource/resources:backstage-and-red-hat-developer-hub-basics

OpenShift, R., s.d. *RedHat OpenShift.* [En ligne]
Available at: https://www.redhat.com/en/technologies/cloud-computing/openshift

OrkoHunter, s.d. *API Docs Plugins.* [En ligne]
Available at: https://www.youtube.com/live/RCd3htP3_qA?si=ruOZWIESubLeMtx3

OrkoHunter, s.d. *How to create Backstage.io plugin in backend.* [En ligne]
Available at: https://www.youtube.com/live/eiwUVtM7EuM?si=_XEeIHnHtzd3kegq

PostgreSQL, s.d. *Commandes PostgreSQL.* [En ligne]
Available at: https://www.linuxtricks.fr/wiki/postgresql-memo-des-commandes-sql

R&D, S., s.d. *How to make great documentation.* [En ligne]
Available at: https://youtu.be/mOLCgdPw1iA?si=eRbYTYefOCNEqiBw

R&D, S., s.d. *What is Backstage ?.* [En ligne]
Available at: https://youtu.be/85TQEpNCaU0

Spotify, 2016. *Architecture de Backstage.* [En ligne]
Available at: https://backstage.io/docs/overview/architecture-overview

Spotify, 2016. *Backstage.* [En ligne]
Available at: https://backstage.io/

Spotify, 2016. *Plugins.* [En ligne]
Available at: https://backstage.io/docs/getting-started/configure-app-with-plugins

Spotify, 2016. *Qu'est ce que Backstage ?.* [En ligne]
Available at: https://backstage.io/docs/overview/what-is-backstage

Spotify, s.d. *Authentification.* [En ligne]
Available at: https://backstage.io/docs/getting-started/config/authentication

Spotify, s.d. *Backstage demo.* [En ligne]
Available at: https://demo.backstage.io/docs?filters%5Buser%5D=all&limit=20

Spotify, s.d. *Base de données.* [En ligne]
Available at: https://backstage.io/docs/getting-started/config/database

Spotify, s.d. *Catalogue des logiciels.* [En ligne]
Available at: https://backstage.io/docs/features/software-catalog/

Spotify, s.d. *Création de plugins.* [En ligne]
Available at: https://backstage.io/docs/plugins/create-a-plugin

Spotify, s.d. *Documentation technique.* [En ligne]
Available at: https://backstage.io/docs/features/techdocs/

Spotify, s.d. *Intégration GitHub.* [En ligne]
Available at: https://backstage.io/docs/integrations/github/locations

Spotify, s.d. *Intégration GitLab.* [En ligne]
Available at: https://backstage.io/docs/integrations/gitlab/locations

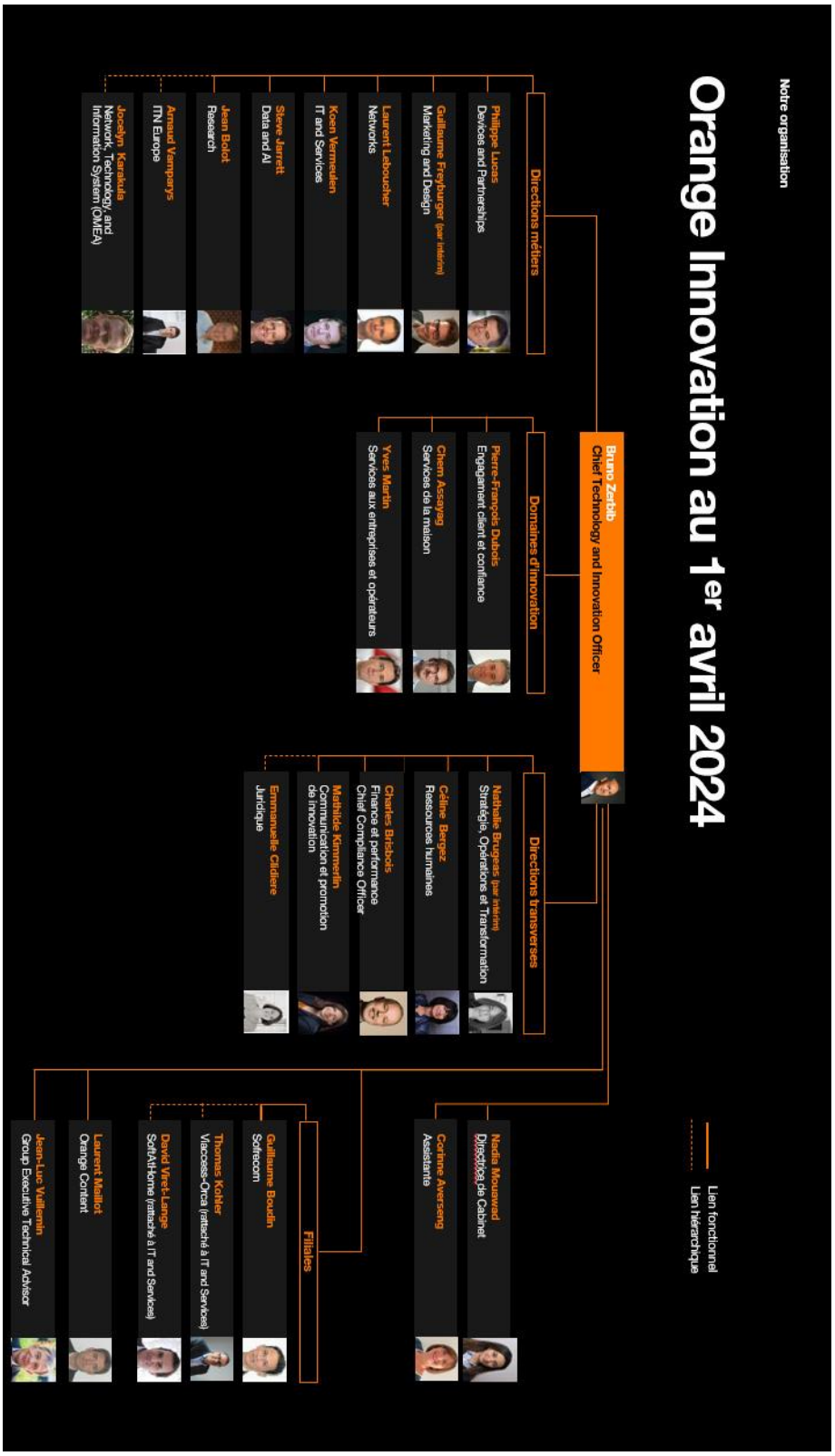
Spotify, s.d. *Kubernetes.* [En ligne]
Available at: https://backstage.io/docs/features/kubernetes/

Spotify, s.d. *Nouveau système de backend.* [En ligne]
Available at: https://backstage.io/docs/backend-system/architecture/index

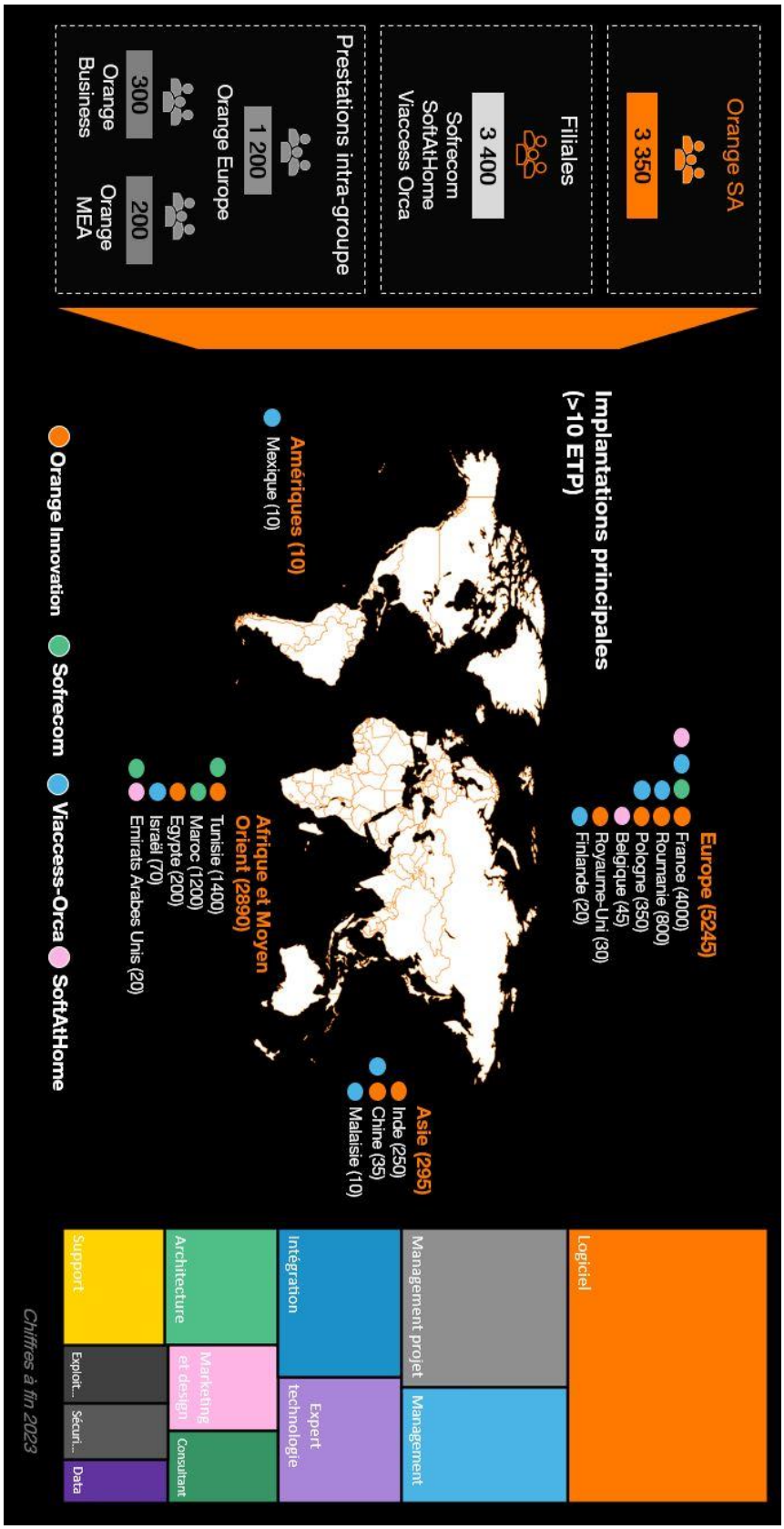
Spotify, s.d. *Plugins existants.* [En ligne]
Available at: https://backstage.io/plugins/

9. Annexes

9.1. Organigramme



9.2. Carte du monde Orange



Résumé

Français

Le groupe Orange, doté de nombreux développeurs et informaticiens, travaille chaque jour sur un nombre incalculable de projets. Quant à eux, ils requièrent une quantité importante de documents pour comprendre et approfondir le savoir à leur sujet. Effectivement, est-il possible de se lancer dans un projet, d'une aussi grande importance que le Caas ou le Iaas, sans avoir la moindre documentation à son sujet, sans savoir par où démarrer ?

Malgré la documentation, le problème persiste toujours. Comment faire pour s'y retrouver dans tous ces documents ? Où sont-ils rangés ? Tout un tas de questions qui font perdre un temps considérable à ces développeurs. Cela impacte donc leur productivité et leur efficacité.

C'est pour répondre à ce besoin que l'application Backstage a été développée. Créée par Spotify en 2016, pour améliorer l'efficacité de ses ingénieurs, cela semble être une solution à ce besoin. Ce portail, conçu pour les développeurs, centralise tous les documents et les outils qui simplifieront et feront gagner un temps considérable aux ingénieurs.

Ce rapport expliquera en détail les démarches suivies pour la mise en place de cet outil, ainsi que les différentes options ajoutées lors de la configuration de celui-ci.

Anglais

The Orange Group, with its many developers and IT specialists, works on countless projects every day. In turn, they require a significant amount of documentation to understand and learn about them. Indeed, is it possible to embark on a project as important as Caas or Iaas without any documentation, without knowing where to start?

Despite the documentation, the problem persists. How do you find your way through all these documents? Where are they stored? A whole host of questions that waste a lot of time for these developers. This in turn impacts on their productivity and efficiency.

The Backstage app was developed in response to this need. Created by Spotify in 2016, to improve the efficiency of its engineers, it seems to be a solution to this need. This portal, designed for developers, centralizes all the documents and tools that will simplify and save engineer's considerable time.

This report will explain in detail the steps taken to set up this tool, as well as the various options added when configuring it.