

focus.

Documentation technique

Résumé

Ce document contient une description sur le plan technique de l'EIP Focus, solution de gestion de productivité. Il présente de façon hiérarchisée les différentes briques technologiques du produit.

Focus se décompose en quatre grandes parties : le Backend, le Frontend web, le Frontend mobile et enfin le Daemon. Ces quatre services agissent de concert afin de créer un service de gestion de productivité innovant. Le Daemon est la principale source de données qui propulse Focus. C'est un service transparent installé sur les ordinateurs de l'utilisateur final. Il collecte des données sur l'utilisation des appareils et transmet en continu des métriques clés au Backend de Focus. De plus, le Daemon est en mesure de proposer une gestion poussée des nuisances numériques à l'aide de filtres définis par l'utilisateur. Celui-ci va analyser en profondeur toutes les informations qui lui sont transmises. Il met en relation une masse importante de données disponibles par agrégation afin d'aboutir à une analyse poussée et sur mesure, en cohérence avec les usages de l'utilisateur. Les Frontend web et mobile mettent à disposition des utilisateurs les données récoltées sur un navigateur web et sur leurs smartphones respectivement.

Description du document

Propriétés	Métadonnées
Titre	Documentation technique
Date	19/08/2018
Auteurs	Enzo AGUADO, Etienne PASTEUR
Version du modèle	1.0
Mots clés	Focus, EIP, Epitech, Documentation, Technique, DA1

Table de révisions

Date	Version	Section(s)	Auteurs	Commentaires
20/02/2017	v0.1	Toutes	Etienne PASTEUR	Création du Template
21/11/2017	v0.2	Rappel de l'EIP	Enzo AGUADO	Mise à jour du résumé
16/08/2018	v.1.0	Toutes	Etienne PASTEUR, Enzo AGUADO	Rédaction du document

Table des matières

Rappel de l'EIP	1
Qu'est-ce qu'un EIP et Epitech.....	1
Sujet de l'EIP Focus.....	1
Présentation du projet.....	2
Prérequis.....	2
<i>Configuration matérielle</i>	<i>2</i>
<i>Configuration logicielle</i>	<i>2</i>
Sources.....	3
<i>Téléchargement des sources.....</i>	<i>3</i>
<i>Téléchargement des installateurs (partie client - daemon)</i>	<i>3</i>
Diagramme global du projet.....	4
Détails sur le projet	5
Daemon.....	5
<i>APIs.....</i>	<i>5</i>
<i>Evénements.....</i>	<i>5</i>
<i>Préanalyse.....</i>	<i>5</i>
<i>Sérialisation.....</i>	<i>5</i>
Backend.....	7
<i>Serveur d'authentification</i>	<i>7</i>
<i>Serveur d'événements</i>	<i>7</i>
<i>Base de données d'événements.....</i>	<i>7</i>
<i>Serveur API</i>	<i>7</i>
Frontend.....	8
Diagramme de communication	9
Conception	10
Choix des technologies	10
<i>Daemon.....</i>	<i>10</i>
<i>Backend.....</i>	<i>10</i>
<i>Frontend.....</i>	<i>10</i>
UML / Diagramme de classes.....	10
Stratégie de tests et de mise en production.....	11
<i>Daemon.....</i>	<i>11</i>
<i>Backend et frontend</i>	<i>11</i>
Informations complémentaires	12
Licence.....	12
Contact	12

Rappel de l'EIP

Qu'est-ce qu'un EIP et Epitech

Epitech (École pour l'informatique et les nouvelles technologies) est un établissement d'enseignement privé français créé en 1999 par le groupe IONIS qui délivre un enseignement supérieur en informatique et nouvelles technologies.

L'école propose à ces étudiants, à partir de leur troisième année, un projet de fin d'études : l'EIP (pour Epitech Innovative Project).

A ce titre, les élèves doivent s'organiser en un groupe d'au moins six personnes et choisir un sujet innovant. L'EIP est un passage obligatoire et unique dans la scolarité de l'étudiant, de par son envergure (18 mois) et la préparation requise.

Un Epitech Innovative Project est conçu (comme toute la pédagogie Epitech fondée sur la méthode projets) à la manière d'un véritable projet entrepreneurial, dans toutes ses composantes : business, techno, design & communication. Un EIP est appelé à devenir une start-up viable à la fin du cursus de l'étudiant.

Sujet de l'EIP Focus

Focus est une solution visant à améliorer le workflow de travail des utilisateurs en leur donnant plus de temps pour faire ce qui compte vraiment.

Grace a des rapports d'activité et des analyses avancées fournis par des outils installés sur vos ordinateurs et smartphones, Focus propulse des informations pertinentes sur votre activité. Un *reporting* complet et détaillé de vos activités quotidiennes vous offre un aperçu global de l'organisation de votre journée, vous permettant de mieux identifier les tâches les plus chronophages.

Ces données permettent à Focus de comprendre comment vous travaillez et de regrouper les sessions de travail analogues (répondre à des courriels, rencontrer des clients, téléphoner, etc.), capturer les notifications en fonction de filtres intelligents et de rationaliser globalement le flux de travail.

Le but ultime est de permettre à nos utilisateurs d'être 100% focalisés sur une tâche unique et bien définie à la fois et de les préserver de toute distraction pendant qu'ils réalisent cette tâche.

Présentation du projet

Prérequis

Configuration matérielle

Focus nécessite la configuration matérielle suivante afin de fonctionner de manière optimale :

- Pour le Backend, un serveur avec un processeur 6 cœurs et 16Gb de RAM au minimum.

Configuration logicielle

Focus nécessite également la configuration logicielle suivante :

- Pour le daemon, un système d'exploitation Linux (testé sous Ubuntu et Arch Linux), Windows (7 ou supérieur) ou MacOS (Sierra ou High Sierra). Focus est compatible 32 et 64bits.
- Pour le backend, n'importe quel système permettant d'exécuter Docker (<https://www.docker.com>).

Sources

Téléchargement des sources

Les sources de Focus sont disponibles sur les différents dépôts GitHub de l'organisation FocusCompany (<https://github.com/FocusCompany>).

Vous pouvez les télécharger en vue de les compiler, les lire, ou les modifier à votre guise.

Les instructions d'installation concernant ces sources ainsi que la liste des dépendances externes font l'objet d'une documentation disponible au sein même des dépôts dans les fichiers appelés README.md.

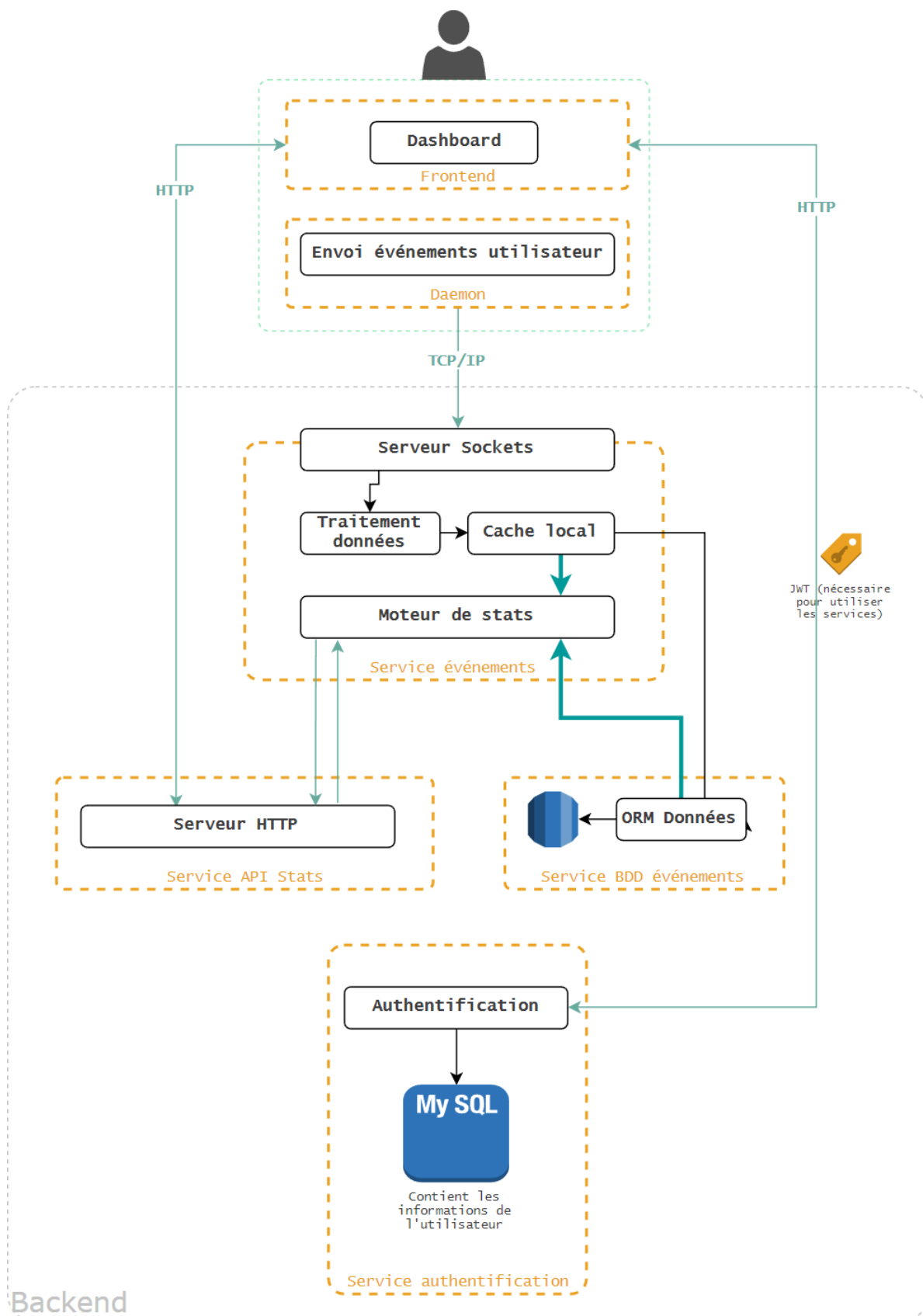
De plus vous pouvez télécharger les archives des sources du daemon (partie client du projet) pour les différentes versions de ce dernier (<https://github.com/FocusCompany/daemon/releases>).

Téléchargement des installeurs (partie client - daemon)

Des installeurs pour les plateformes Linux, Windows et MacOS sont disponibles afin de simplifier au maximum le déploiement du daemon sur les environnements de nos utilisateurs.

Des installeurs sont disponibles en téléchargement libre sur la page GitHub du daemon (<https://github.com/FocusCompany/daemon/releases>).

Diagramme global du projet



Détails sur le projet

Daemon

APIs

Le daemon est un logiciel s'installant sur les ordinateurs de l'utilisateur. Une version du daemon existe par plateforme afin d'utiliser les API adéquates. Les APIs utilisées sont les suivantes.

- Cocoa, Carbon et Quartz sur Mac OS.
- Win32 API, bibliothèque bas niveau de Windows offrant une bonne rétrocompatibilité avec les versions précédentes de Windows.
- Xlib, permettant d'interagir avec le serveur X qui sert de serveur graphique à la grande majorité des distributions Linux. Cette librairie est aussi compatible avec le plus récent serveur graphique Wayland.

Ce logiciel agit généralement de manière invisible aux yeux de l'utilisateur, ne demandant une action de sa part que pour l'authentification. D'autres contrôles simples sont également disponibles, comme la possibilité de mettre en pause le logiciel ou de changer de compte.

Evénements

Pour le cas d'utilisation principal, le daemon tourne en arrière-plan et récupère l'activité de l'utilisateur via les APIs susmentionnées. De nombreuses métriques sont récupérées afin de déterminer avec le maximum de précision l'utilisation réelle de la machine : titre des fenêtres, changements de fenêtre/onglet, temps passé par logiciel, onglets ouverts, fréquence de changement de contexte, nombre de notifications. Ces informations sont susceptibles d'être ajustées afin de refléter l'évolution de l'algorithme de traitement du backend.

Préanalyse

Une fois ces informations recueillies, elles passent par une première phase de préanalyse. Les informations sont d'abord nettoyées afin d'enlever les activités éphémères.

Ex : l'utilisateur doit passer de Word à ses mails, mais se trompe de fenêtre et donne le focus à Facebook qui était ouvert dans son navigateur. Le daemon aura enregistré cet événement, même s'il n'est pas pertinent (l'utilisateur a techniquement visité Facebook pendant un instant, mais sans l'intention de s'en servir). Cet événement peut donc être retiré sans risque afin de laisser la priorité aux applications réellement utilisées.

Sérialisation

Une fois nettoyées, les informations passent dans un sérialiseur qui convertira la donnée brute en l'insérant dans une « enveloppe » compatible avec les autres services Focus.

Cette enveloppe est une structure Protobuf définie globalement au travers des micro-services Focus. Protobuf est une bibliothèque de sérialisation et de Remote Procedure Call (RPC) open source développée majoritairement par Google. Au sein de Focus, elle garantit un modèle de données identique quelle que soit la plateforme, service ou langage de programmation utilisé (tant que ce dernier est supporté par Protobuf).

Cette enveloppe est composée d'un champ précisant le type d'événement sous la forme d'une énumération définie dans les structures Protobuf, permettant de savoir comment décoder le reste de l'enveloppe. Ce champ est suivi d'un message concernant les détails liés à cet événement. Par exemple, s'il s'agit d'un changement de fenêtre, le type d'événement ressemblera à `USER_CHANGED_WINDOW`, et permettra de déduire que le reste du message est composé :

- D'une chaîne de caractères `previous_window_name` qui correspond à la fenêtre sur laquelle l'utilisateur était avant d'en changer
- D'une chaîne de caractères `current_window_name` qui correspond à la nouvelle fenêtre sélectionnée
- D'un entier `changed_after` qui sera le temps passé sur la fenêtre précédente.

Les types de message sont intimement liés aux événements que récupèrent le daemon, et sont donc naturellement amenés à changer en fonction des besoins.

Backend

Serveur d'authentification

Les données d'utilisateurs ne peuvent être accessibles qu'avec un token d'accès. Ce token ne peut être obtenu qu'avec les identifiants de l'utilisateur.

Lors de la première utilisation, l'utilisateur doit renseigner ses informations de connexion au serveur d'authentification. Si ces informations sont valides, ce dernier génère alors un token JWT qui contient l'identifiant unique de l'utilisateur et quelques informations liées à cet utilisateur. Ce token doit être envoyé avec chaque requête pour que celle-ci soit valide.

Le JWT a un délai d'expiration de quelques minutes et doit être fréquemment échangé contre un nouveau. Si l'utilisateur se déconnecte, cet échange est rendu impossible et l'utilisateur devra régénérer un nouveau JWT.

Le serveur d'authentification est aussi responsable de la gestion des comptes utilisateurs, de leur création à leur suppression.

Serveur d'événements

Le serveur d'événements est un entonnoir pour toutes les actions utilisateur. Il récupère les connexions socket de tous les daemons, traite les données reçues, et les envoie à la base de données.

Le serveur d'événements a aussi pour rôle de traiter les demandes de statistiques. Dès que le serveur API envoie une requête, le serveur d'événements agrège les données de son cache local et celles venant de la base de données avant de les envoyer dans le moteur de stats approprié. Le résultat est converti en JSON et renvoyé au client.

Ce serveur est codé selon un modèle d'actor model, chaque moteur de stat est un module indépendant et qui ne s'active que lorsqu'il reçoit un message spécial. Cela permet de traiter plusieurs requêtes similaires.

Base de données d'événements

La base de données est placée derrière un ORM qui va permettre de traiter toutes les requêtes de manière sécurisée. La base de données est une base Cassandra, choisie pour ses performances en écriture.

Serveur API

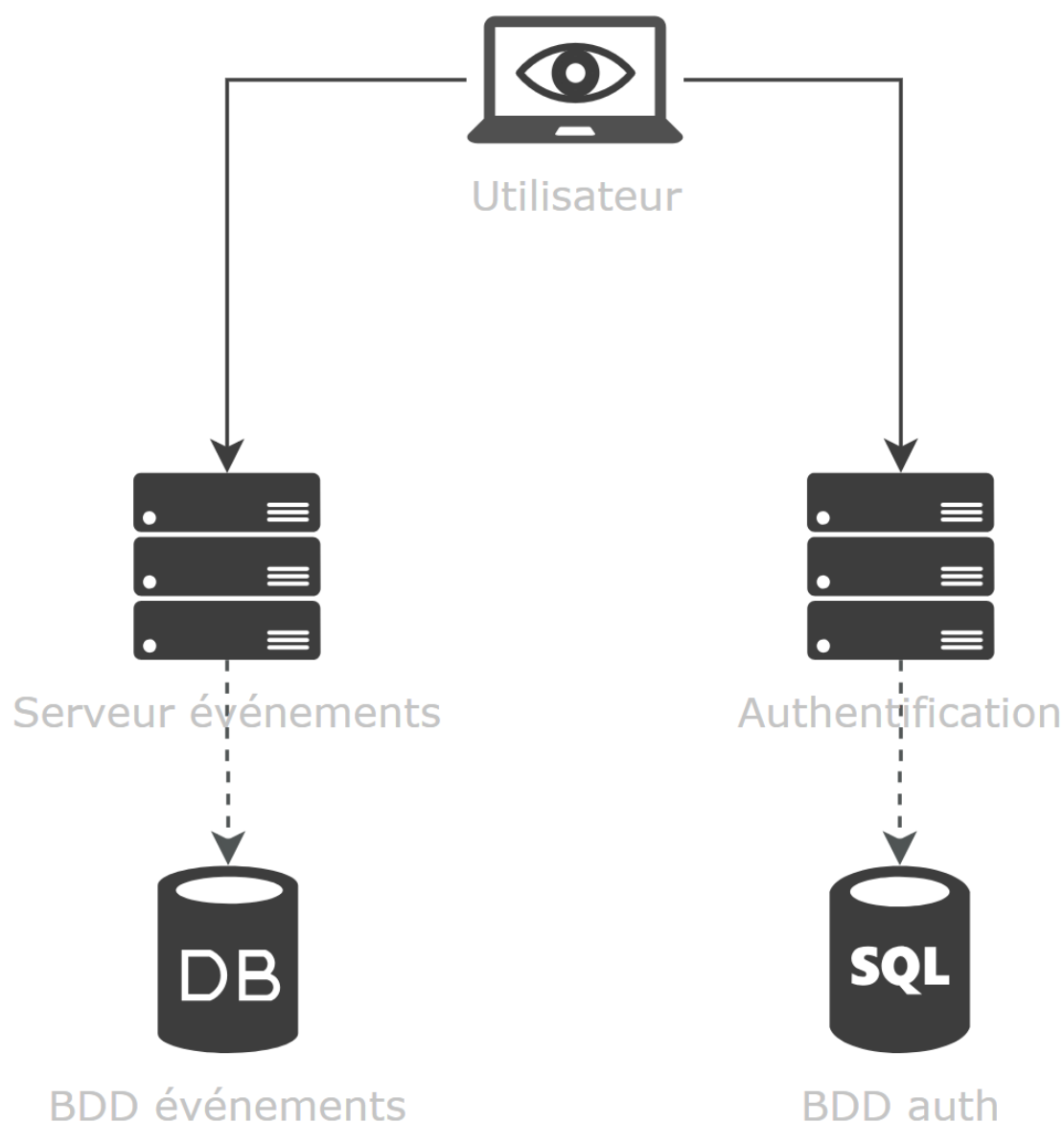
Le serveur d'événements ne communique qu'en sockets. Ce type de communication étant compliqué pour un client, nous avons mis en place un serveur capable de comprendre le protocole HTTP. Ce serveur est principalement utilisé par les dashboard pour récupérer les données nécessaires à la construction des graphs de reporting.

Frontend

Le frontend est l'interface par laquelle communique principalement l'utilisateur. Le choix d'offrir une plateforme web plutôt que de tout afficher dans une fenêtre gérée par le Daemon présente plusieurs avantages. Les mises à jour sont mieux contrôlées, plus simples et plus rapides. Les coûts et temps de développement et la maintenance sont grandement réduits. L'accessibilité du produit est aussi grandement améliorée.

Il permet notamment de : s'authentifier sur les services focus (inscription, connexion, mot de passe oublié), d'accéder à son profil (voir son activité), créer et modifier des filtres permettant de gérer les différentes nuisances numériques ainsi que de gérer ses paramètres personnels.

Diagramme de communication



Via le Daemon installé sur son ordinateur, ou via le Dashboard accessible sur le web, l'utilisateur communique avec tous nos services : moteurs de statistiques, serveur d'authentification, enregistrement des activités, etc.

Conception

Choix des technologies

La majorité de nos services sont écrits en C++. Dans le cadre du Daemon, ce choix s'est posé naturellement grâce à la facilité du langage d'être porté sur plusieurs plateformes et sa compatibilité avec les APIs des systèmes d'exploitation. Dans le cadre du Backend, c'est le besoin de performance qui a gouverné le choix du langage. De plus, tous les membres étaient formés dans ce langage, ce qui nous a permis de commencer rapidement.

Daemon

Globalement, le Daemon est un set de classes permettant d'interagir aisément avec les APIs systèmes : Win32 pour Windows, X11 pour Linux, et AppleScript pour MacOS.

Backend

L'architecture en micro-services utilise Docker comme base de déploiement. Ce choix semble naturel vu la maturité du projet. Le choix du Framework cependant est plus délicat. Nous avons besoin d'un modèle offrant une parallélisation simple et efficace, et le pattern d'actor model nous a semblé approprié. Pour la base de données, nous avons choisi Cassandra pour ses performances en écriture et son modèle de données adapté aux séries dans le temps.

Frontend

Les plateformes web imposent le choix du langage, nous n'avons pas eu de choix à ce niveau là.

UML / Diagramme de classes

Les diagrammes de classes (UML) font l'objet d'une documentation externe, consultable et téléchargeable sur le dépôt GitHub suivant : <https://github.com/FocusCompany/Documents>.

Stratégie de tests et de mise en production

Il existe deux environnements bien séparés au sein du projet Focus. D'un côté le daemon (la partie client) et d'un autre côté le backend et ses différents services ainsi que le frontend (la partie serveur). Il y'a donc deux stratégies de tests et de mise en production différentes, une pour chaque partie.

Daemon

Une série des tests unitaires a été mise en place sur le daemon à l'aide du Framework de test Google Test. Chaque module de code (code partagé, code spécifique au système d'exploitation, bibliothèques, etc.) dispose de son propre ensemble de test. A chaque fois qu'un push est effectué sur le dépôt Git, l'ensemble de ces tests sont exécutés sur les trois plateformes (MacOs et Linux sur Travis et Windows sur AppVeyor).

De plus, un système de déploiement (mise en production) a été mis en place, à chaque fois que du code est ajouté (merge) à la branche master du dépôt Git les installateurs sont créés après le passage des tests et ces derniers sont déployés sur GitHub à l'adresse suivante : <https://github.com/FocusCompany/daemon/releases>.

Backend et frontend

Un système de test similaire à celui du daemon va être mis en place sur le backend ainsi que le frontend dans les prochains jours.

Concernant le déploiement et la mise en production, ceux-ci se font à l'aide de Docker (<https://www.docker.com>) et notamment d'un fichier docker-compose qui permet de déployer l'ensemble de l'infrastructure (base de données, différents services, etc.) grâce à une seule commande.

Informations complémentaires

Licence

Focus est un projet open-sources sous licence MIT. Les termes complets de la licence MIT sont disponibles à l'adresse suivante : <https://choosealicense.com/licenses/mit/>.

Contact

Vous pouvez visiter notre site web pour toute informations supplémentaires à l'adresse suivante : <http://eip.epitech.eu/2019/focus>.

Vous pouvez également nous contacter directement via le formulaire disponible à cette adresse ou via l'adresse mail suivante : focus@epitechfr.onmicrosoft.com.