

Fiche **FADI**

UN FRAMEWORK POUR L'AUTOMATISATION DU DÉPLOIEMENT ET DE
L'ORCHESTRATION D'INFRASTRUCTURES CONTENEURISÉES

Vous disposez de nombreuses données (des mails, des pdfs, des communications vocales, des données provenant de vos capteurs, ...) mais ne savez pas comment les exploiter et les capitaliser ? Vous désirez faire un prototype/POC avec une brique logicielle à moindre frais ? Vous recherchez une solution technologique "neutre" car vous craignez l'enfermement propriétaire/le vendor lock-in ?

Que vous soyez une grande entreprise ou une plus petite, FADI va vous intéresser : il s'agit d'une solution peu coûteuse en termes d'infrastructure et d'achat de licences. Avec cette suite, vous pouvez en effet procéder étape par étape dans le développement et l'opération de vos prototypes et produits : les intégrer, déployer, ...

Prérequis

- Avoir une connaissance des technologies liées aux conteneurs (Docker, Kubernetes).
- Savoir écrire des lignes de commandes.
- Disposer d'une station de travail récente ou d'un espace de test Kubernetes.

Partenaires

Fiche **FADI**- Un Framework pour l'Automatisation du Déploiement et de l'orchestration d'Infrastructures conteneurisées

🎯 Pour quoi faire?

Pour faire simple, **FADI** est un framework personnalisable selon vos besoins, qui assemble et configure les outils à votre place afin de faciliter et d'automatiser l'intégration, le déploiement et le suivi de vos applications.

À l'origine, FADI est une plateforme dédiée au Big Data qui rassemble des applications *cloud native* et qui se base sur des outils open source éprouvés. Cette plateforme rend le déploiement de solutions Big Data mais aussi celui des piles logicielles plus simples, portables et évolutifs sur diverses infrastructures (clouds privés et publics).

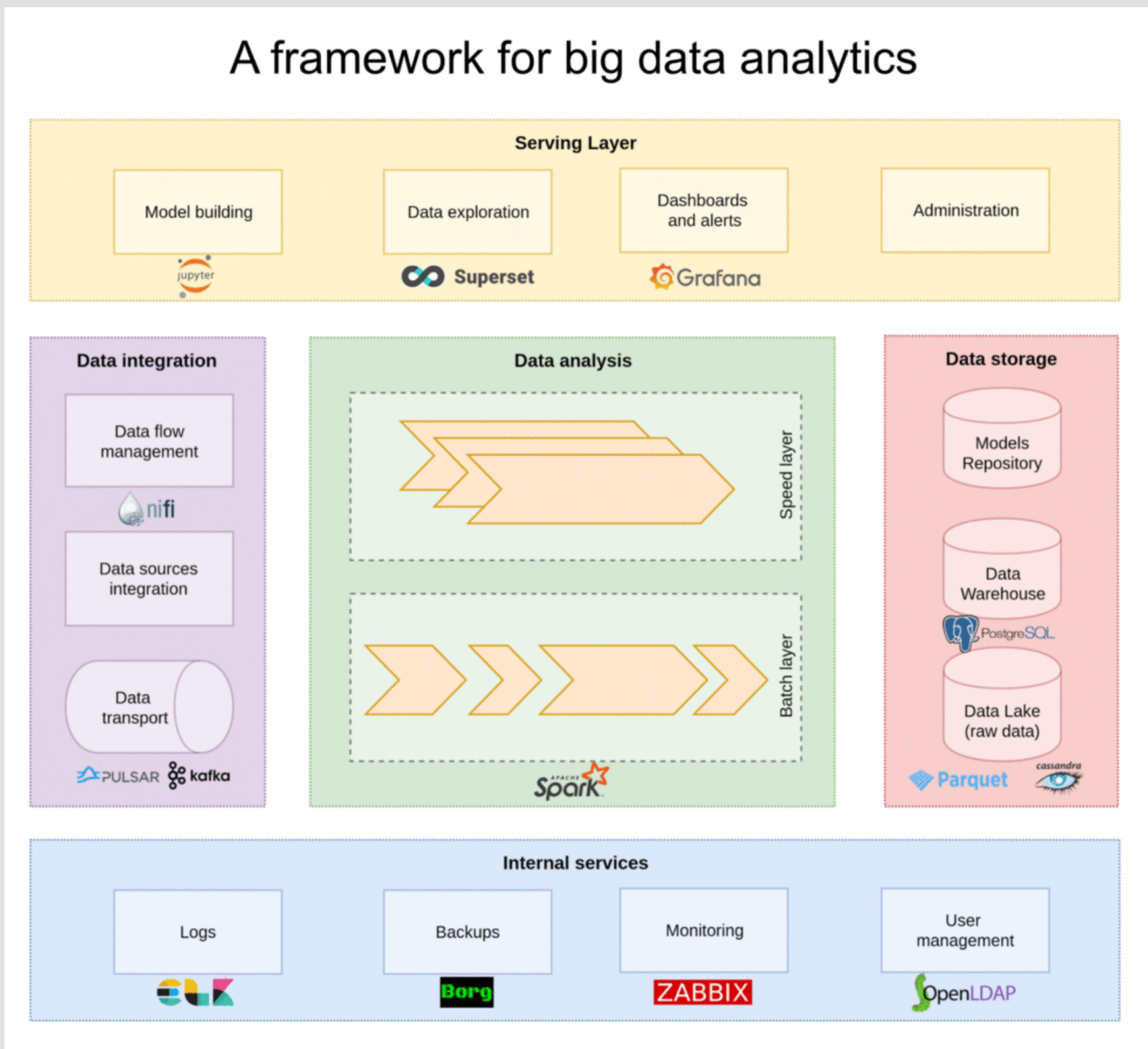
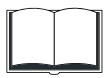


Illustration des divers composants de **FADI**

Fiche **FADI**- Un Framework pour l'Automatisation du Déploiement et de l'orchestration d'Infrastructures conteneurisées



Quelques concepts ...

Une application “Cloud Native”

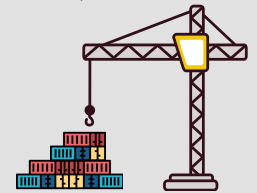
Voici une description succincte des caractéristiques d'une application “cloud native”, et donc des applications assemblées par la plateforme FADI :

- 1 Une architecture basée sur les **services** et les **microservices**.

Un microservice est une architecture pour les applications : les services y sont individuels, indépendants, et spécifiques et, une fois assemblés, constituent l'application. Chaque service peut donc être déployé, mis à jour et géré de façon parfaitement autonome.

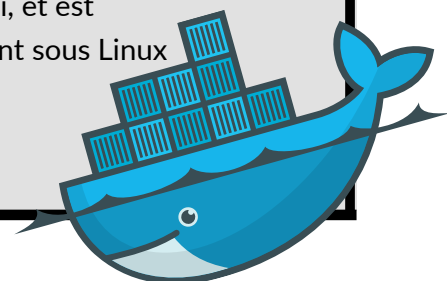
- 2 Une communication entre les services (internes et externes) basée sur les **API** (*Application Programming Interface*, interface de programmation d'application).

- 3 Une infrastructure basée sur les **conteneurs** (ce qui facilite la gestion des microservices).



Un conteneur est un package logiciel léger et indépendant, une sorte d'enveloppe virtuelle ou d'environnement isolé, qui regroupe tous les éléments nécessaires au bon fonctionnement et à la distribution d'une application : le code, les fichiers de configuration, l'environnement d'exécution, les libraires et toutes les dépendances requises pour l'exécution,... L'ensemble du contenu du conteneur se trouve dans une image conteneur, c'est un fichier de code qui contient l'application/service, ses dépendances et sa configuration. Les intérêts des conteneurs sont de virtualiser les applications logicielles en utilisant le système d'exploitation de leur hôte et de pouvoir les actualiser sans réorganiser la totalité de l'application. De cette façon, les ressources sont très facilement flexibles et portables d'un système à un autre. Les conteneurs n'intègrent pas leur propre système d'exploitation, ils sont donc très légers et facilement déployables dans d'autres environnements avec peu, voire aucune modification! Vous pouvez mettre à jour un seul outil, séparément des autres; vous pouvez installer la dernière version de Grafana et garder Jupyterhub à sa première version par exemple.

Docker est la technologie de conteneurisation la plus utilisée. Elle vous permet de créer aisément vos conteneurs et vos applications basées sur ceux-ci, et est très facile à déployer. Il s'agit d'une solution open source fonctionnant sous Linux mais également Windows Server.



Fiche **FADI**- Un Framework pour l'Automatisation du Déploiement et de l'orchestration d'Infrastructures conteneurisées

④ La méthode de développement logiciel **DevOps** (contraction de Développement et Opération).

Les objectifs de cette méthode sont de permettre, grâce à la communication et à la collaboration entre les développeurs et les responsables des opérations IT, la fluidification des processus, l'accélération de la résolution d'incidents ainsi que de la livraison de nouvelles fonctionnalités avec un niveau de qualité élevé.


Les pratiques DevOps lors du cycle de vie de l'application se caractérisent par une automatisation et une surveillance accrues de toutes les étapes de la création du logiciel. Cela concerne

- *L'intégration continue* : il s'agit d'exécuter des tests automatisés afin de vérifier chaque modification du code source pour en garantir sa qualité. Les problèmes d'intégration sont facilement détectés si vous réalisez correctement la planification de votre développement, la compilation, l'intégration, les tests de votre code et la gestion de vos livrables (les artefacts prêts à être déployés).
- *Le déploiement continu* : une fois les tests validés lors de l'intégration continue, ils peuvent être mis en production. Le déploiement continu est l'automatisation de la mise en production des applications lors de chaque modification.

Pour en savoir plus sur les applications *cloud natives*, cliquez [ici](#). Quant à la méthode DevOps, vous pouvez consulter les sites suivants : [Appvizer](#), [Netapp](#), [Padok](#)

Un orchestrateur de conteneurs

Alors qu'utiliser un ou deux conteneurs peut être facilement gérable et contrôlable, en utiliser plus d'une dizaine peut rapidement se révéler être une tâche fastidieuse, notamment à cause des multiples dépendances et communications entre outils. Un orchestrateur de conteneurs permet de gérer ces communications en automatisant le déploiement, la gestion, la mise à l'échelle et la mise en réseau des conteneurs. L'orchestrateur ne crée pas les conteneurs (pour cela vous devez utiliser une plateforme de conteneurs comme Docker, cité ci-dessus) mais permet de les gérer.

L'orchestrateur de conteneurs le plus connu et mature actuellement est open source et développé par google, il s'agit de  **Kubernetes**.

Très utilisé dans le monde industriel, il se déploie sur toutes les architectures disponibles (dans le cloud privé, dans du bare metal, c'est-à-dire un serveur dédié physique, dans des clouds publics,...) et peut fonctionner sur tout type d'infrastructure.

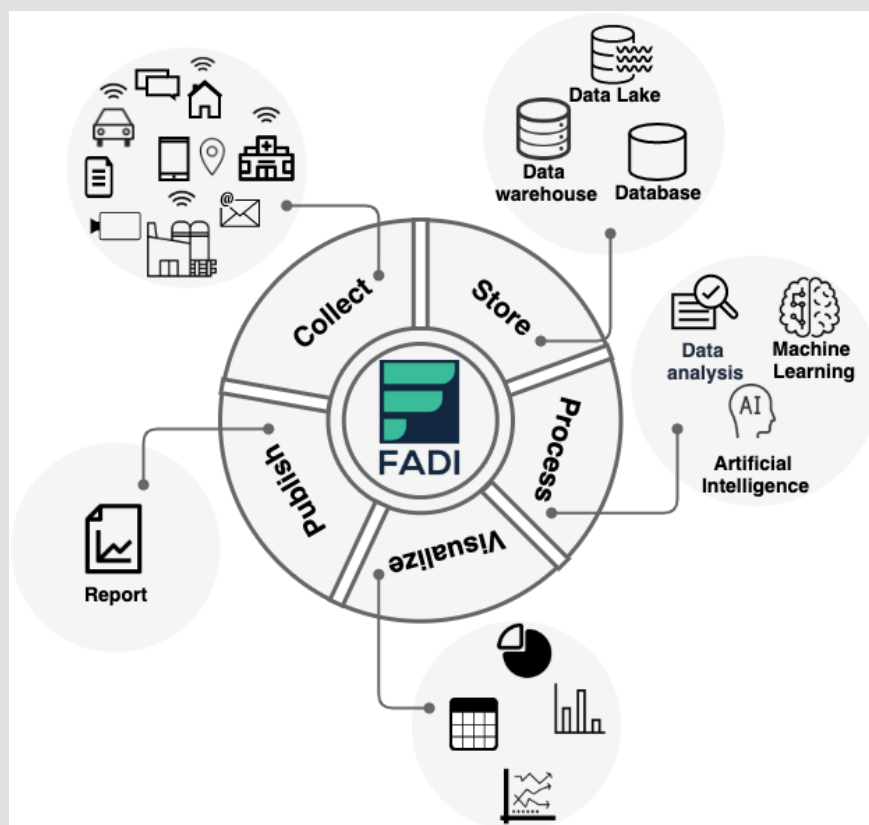
Fiche **FADI**- Un Framework pour l'Automatisation du Déploiement et de l'orchestration d'Infrastructures conteneurisées

Une pile logicielle

Une pile logicielle est un groupe de logiciels fonctionnant selon un ordre spécifique et permettant le développement de sites et applications en offrant, ensemble, un service.

Les atouts de FADI

Pourquoi FADI est-elle une plateforme très intéressante ? Parce vous pouvez développer entièrement vos prototypes et produits !



1 Les déploiements sont automatisés et simplifiés

Déployer chaque outil à la fois prend du temps et rend la maintenance plus compliquée ; l'automatisation des déploiements va vous simplifier la vie !

Pour plus de fiabilité, vous pouvez prévoir et programmer vos déploiements au moment désiré. Vous pouvez également les contrôler en effectuant une sorte de versionning : par exemple, vous pouvez réaliser un premier déploiement et faire un correctif pour le second, revenir au premier déploiement,... De cette façon, FADI vous assure une certaine traçabilité.

Fiche **FADI**- Un Framework pour l'Automatisation du Déploiement et de l'orchestration d'Infrastructures conteneurisées

Les déploiements sont simplifiés grâce à des assemblages de services préconfigurés basés sur le logiciel Helm, appelés des *Charts*. Il s'agit de packages (ou templates) de ressources Kubernetes sous forme de scripts d'installation (des automatisations d'installation) : ces packages sont des sélections d'outils open source pertinents selon le cas d'utilisation/le scénario qui vous intéresse. Par exemple, si vous vous demandez comment installer Apache Nifi sur Kubernetes, vous trouverez toute la démarche ici. Vous trouverez les autres Helm Charts produites et maintenues par le CETIC dans ce dépôt GitHub (attention, ils ne concernent pas tous FADI !).

2 Sa portabilité

FADI est une solution portable, c'est-à-dire que vous pouvez la déployer un peu partout : dans un cloud privé, dans un cloud public, sur site, de manière hybride, etc. Une chose est certaine, là où vous pouvez déployer Kubernetes, vous pouvez déployer FADI.

3 Sa maintenabilité

FADI s'appuie sur la méthodologie DevOps : vous n'aurez pas à vous soucier de grand-chose étant donné que l'installation, l'intégration, les essais, les déploiements continus et les mises à jour de FADI sont automatisés ! La maintenance étant beaucoup plus facile, le système sera toujours à jour au niveau des dépendances.

4 Une solution qui vous correspond

Personnalisable de bout en bout, FADI s'adapte à vos besoins. Que cela soit le type d'infrastructure (cloud ou pas), de base de données, etc., FADI offre une sélection d'outils qui vous permet de gérer vos données... Vous pouvez également y intégrer très facilement vos propres solutions existantes ou de nouveaux services. De plus, FADI étant une solution se basant sur des technologies open source matures, vous pouvez toujours la modifier !

5 Ses traitements de données

FADI propose deux modes d'intégration (voir la *fiche Introduction au data processing* pour plus d'informations)

- Le mode *Batch* pour le traitement d'un grand volume de données en une seule fois, sur une période donnée (traitement par lot).
- Le mode *Stream* pour le traitement des données en continu

Fiche **FADI**- Un Framework pour l'Automatisation du Déploiement et de l'orchestration d'Infrastructures conteneurisées

L'avantage d'avoir ces deux modes de traitement est que FADI répond à de nombreux use-cases différents :

- Le mode *Batch* est souvent utilisé lorsqu'il faut réaliser des facturations, commandes, rapports,...
- Le mode *Stream* est surtout conseillé lorsque vous devez détecter des événements et y répondre rapidement, comme par exemple la surveillance des services, la cybersécurité, l'analyse des comportements, la détection de fraudes,...

6 Ses solutions de stockage de données

FADI propose deux manières de stocker et de centraliser vos données :

- Les *Data Lakes* (lacs de données) qui sont le référentiel de données vous permettant de stocker "en vrac" les données brutes originales ingérées (audit, relecture, expériences, etc.). Le mode de traitement des données par lots/batch est préféré pour ce genre d'approche (traitements des données à la demande).
- Les *Data Warehouse* (entrepôts de données) qui sont des bases de données où sont stockées les vues agrégées des données ingérées. Le mode de traitement des données Stream est préféré pour cette approche (traitement au moment de l'ingestion des données).

7 Monitoring et Data visualisation

Il est extrêmement important, à l'heure actuelle, que vous sachiez si vos infrastructures IT sont opérationnelles et sécurisées.

Les outils de supervision (monitoring) sont précieux pour vos stratégies de contrôle, d'observation et de décisions.

FADI utilise des outils open-source de monitoring afin de vous permettre de suivre l'état de santé de vos systèmes, collecter vos données en temps réel et déclencher des alertes dès qu'un problème est rencontré.

L'aspect open-source de ces outils répond aux défis du DevOps : le fait de pouvoir remonter n'importe quel type de métriques provenant des applications, et même les indicateurs métiers, répond aux besoins actuels des entreprises en termes de rapidité, flexibilité et de maîtrise des coûts.



Fiche **FADI**- Un Framework pour l'Automatisation du Déploiement et de l'orchestration d'Infrastructures conteneurisées

FADI vous permet donc de monitorer vos données en configurant vous-même vos systèmes d'alertes. Comme vous pouvez le constater ci-dessous, les outils proposés servent aussi à faire de la data visualisation : c'est-à-dire à représenter et à visualiser les données collectées sous forme de graphique, diagramme et/ou tableaux de bord (dashboards) afin de pouvoir prendre rapidement des décisions. Les outils proposés initialement par FADI sont :

Grafana : il s'agit d'un outil orienté data visualisation avec lequel vous pouvez réaliser des tableaux de bords et des graphiques à partir de différentes sources de données, et principalement à partir de séries temporelles.



Illustration de Grafana provenant de la présentation de FADI

Prometheus : ce logiciel est parfait pour la surveillance de vos métriques ! Il les collecte et les stocke sous forme de séquences temporelles. Bien que des représentations visuelles soient possibles, Prometheus est plutôt utilisé pour la collecte, l'analyse de données ainsi que la génération d'alertes. N'incluant pas de tableaux de bord par défaut, préférez Grafana pour la visualisation de vos données (il prend en charge l'intégration de Prometheus). Si vous désirez plus d'informations sur Prometheus et le monitoring, nous vous conseillons ce lien.

Fiche **FADI**- Un Framework pour l'Automatisation du Déploiement et de l'orchestration d'Infrastructures conteneurisées

Zabbix: cette plateforme distribuée est relativement facile d'utilisation pour les personnes ayant peu de connaissances techniques. Elle vous permet de collecter et de gérer vos données, de détecter un problème et d'envoyer des notifications selon des règles personnalisables. Vous pouvez également visualiser et analyser l'état de santé de l'infrastructure IT. Tout comme Prometheus, vous pouvez également interfacier Zabbix avec Grafana.

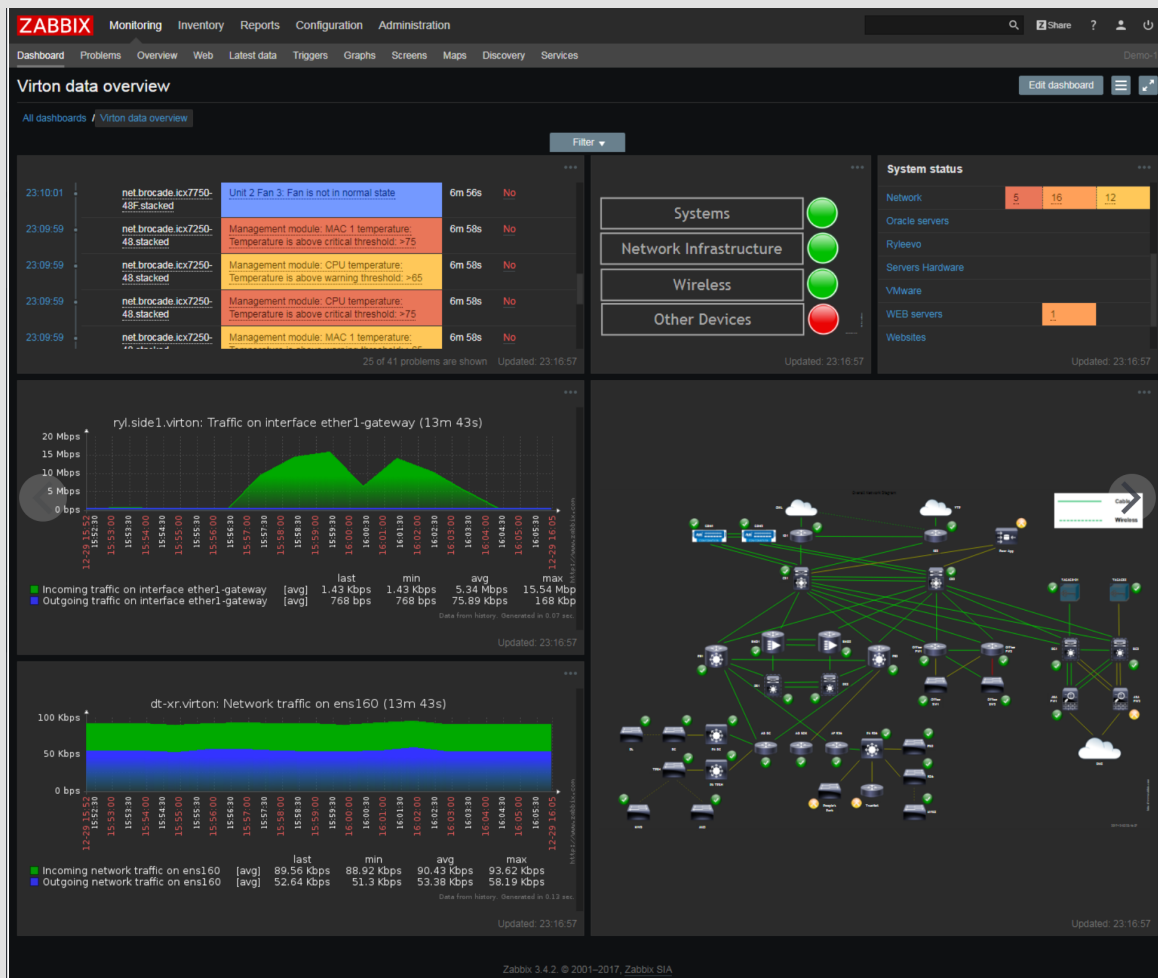


Illustration d'un dashboard provenant de Zabbix

Apache Superset : cet outil open source a été initialement développé par Airbnb. Il fonctionne en tant qu'application web sur les principaux navigateurs internet et permet d'explorer et visualiser vos données très intuitivement sous forme de graphiques, feuilles de calculs et tableaux de bord.

Fiche FADI- Un Framework pour l'Automatisation du Déploiement et de l'orchestration d'Infrastructures conteneurisées

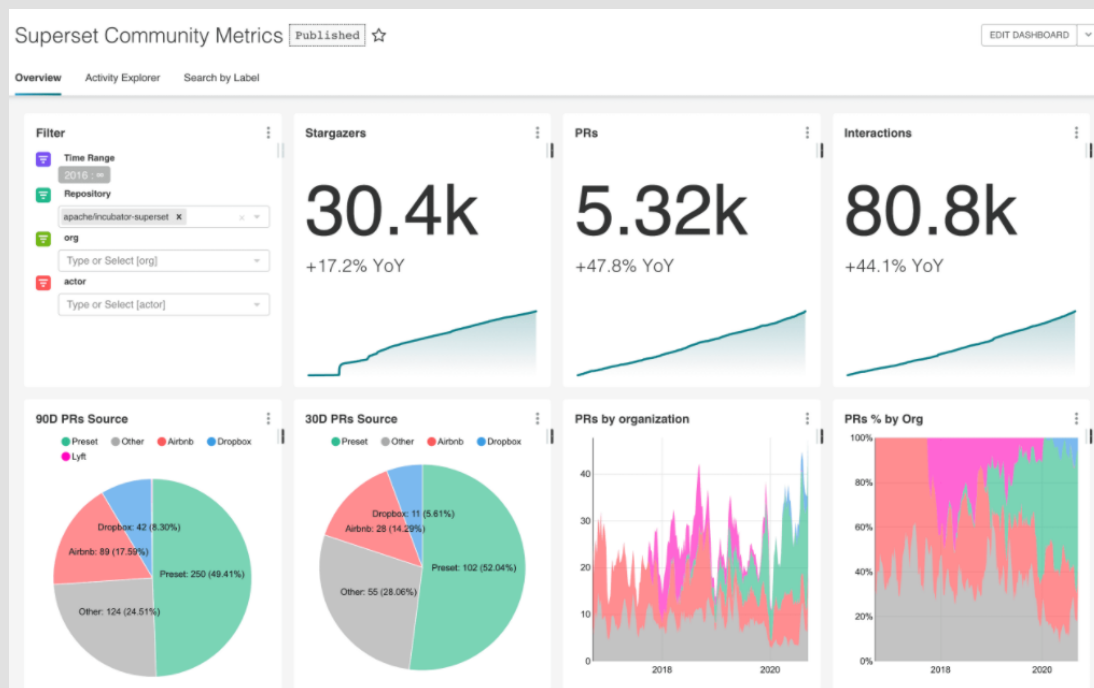


Illustration d'un dashboard provenant de Apache Superset

N'oubliez pas que vous pouvez également intégrer très facilement vos propres outils dans FADI si vous préférez.



Perspectives

Actuellement (mai 2021), le CETIC a prévu plusieurs développements à faire pour FADI ; notamment l'ajout d'un wizard pour configurer encore plus facilement FADI et travailler sur les aspects de sécurité (DevSecOps).



Exemples de cas d'étude

Le monitoring des bureaux du CETIC

L'objectif de ce cas d'utilisation était de fournir des informations (tableaux de bord et alertes) en fonction des données des capteurs placés au sein du bâtiment du CETIC (température, CO2, ...). Dans cet exemple, les mesures de température des capteurs ont été ingérées, stockées affichées dans un tableau de bord assez simple. Rendez-vous sur ce guide utilisateur FADI pour accéder au tutoriel de cet exemple, vous y apprendrez comment :

Fiche **FADI**- Un Framework pour l'Automatisation du Déploiement et de l'orchestration d'Infrastructures conteneurisées

1. Installer FADI via [ce lien](#).
2. Préparer la base de données pour stocker les mesures avec [PostgreSQL](#) (qui est à la fois entrepôt de données et base de données).
3. Ingérer les mesures des capteurs grâce à [Apache Nifi](#), depuis la source de données (un fichier csv dans ce cas) et les stocker dans la base de données.
4. Afficher les tableaux de bord et configurer des alertes à partir des données ingérées et stockées dans l'entrepôt de données avec [Grafana](#).
5. Explorez les données avec [Superset](#).
6. Traiter les données en utilisant [Jupyter](#) comme interface web pour les explorer à l'aide de notebooks et [Apache Spark](#), un framework de calcul distribué, comme moteur d'analyse pour le traitement de grands ensembles de données.

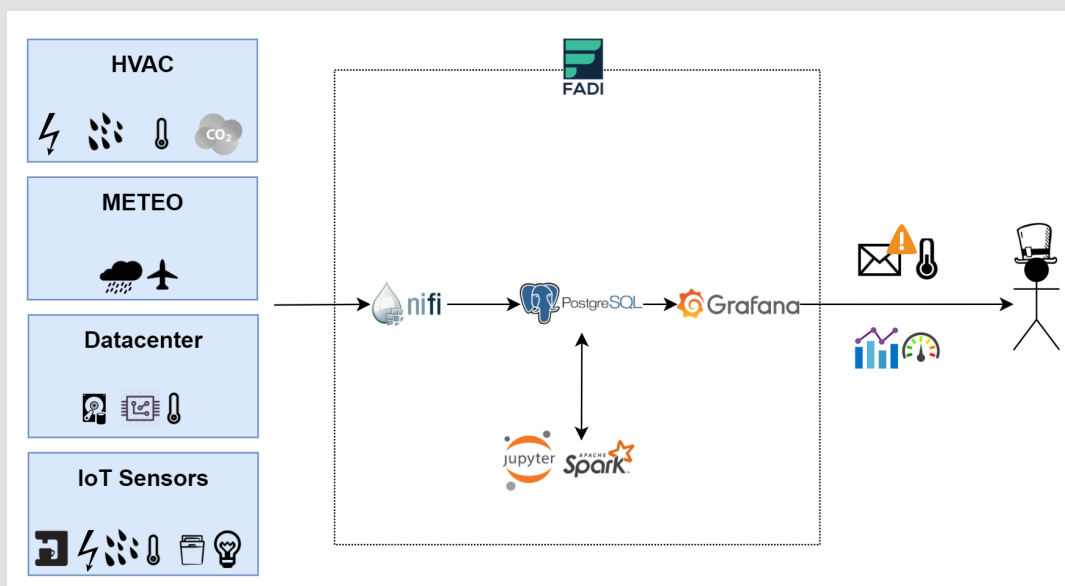


Illustration du processus de monitoring des bureaux du CETIC provenant du [guide utilisateur de FADI](#)

Autres exemples de cas d'étude

Cette liste de projets représente d'autres exemples de cas d'étude où le framework FADI a été utilisé :

[Artemtec](#)

[Grinding 4.0](#)

[UserMEDIA /CloudMedia](#)

[NewTech4Steel](#) (European RFCS project)

[Quality4.0](#) (European RFCS project)

[BigData@MA](#) (European, "Manunet" project)

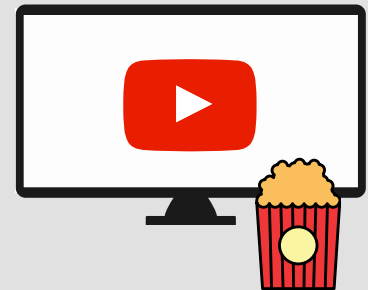
[Autosurveillance](#) (European RFCS project)

[QuadRide](#) (Wallon CWALity project)

Fiche **FADI**- Un Framework pour l'Automatisation du Déploiement et de l'orchestration d'Infrastructures conteneurisées

Démonstration en vidéo

Voici le replay du webinaire organisé par le Hub-C le 6 juillet 2021 afin de faire une démonstration de Fadi, un outil qui rend les technologies Big Data plus accessibles. La présentation est réalisée par Faiez Zalila, expert en Ingénierie logicielle basée sur les modèles et systèmes informatiques distribués au sein du CETIC.



Pour aller plus loin ...

FADI est un outil développé par le CETIC, n'hésitez donc pas à prendre contact si vous désirez des informations supplémentaires !

Voici quelques liens utiles par rapport à FADI :

- Site officiel de FADI : <https://FADI.cetic.be/> contenant la documentation d'installation et d'utilisation
- Pour en savoir plus : <https://FADI.presentations.cetic.be>
- Le code source : <https://github.com/cetic/fadi>
- Le dépôt Github des Helm charts pour installer FADI dans un cluster Kubernetes : <https://github.com/cetic/helm-fadi>
- "R. Sellami, F. Zalila, A. Nuttinck, S. Dupont, J. -C. Deprez and S. Mouton, "FADI - A Deployment Framework for Big Data Management and Analytics", 2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2020, pp. 153-158, doi: 10.1109/WETICE49692.2020.00038."

Fiche **FADI**- Un Framework pour l'Automatisation du Déploiement et de l'orchestration d'Infrastructures conteneurisées

Besoin d'une aide supplémentaire ?

Le Hub-C dans le cadre de ses services d'accompagnement numérique organise des workshops et groupes de travail en lien avec les nouvelles technologies de prototypages. Vous souhaitez un accompagnement pour votre projet innovant ou vous souhaitez participer à un prochain workshop ? N'hésitez pas à contacter un membre du Hub !

Vous avez une question spécifique à propos d'une fiche? Elles sont réalisées par les experts du CETIC (Centre d'Excellence en Technologies de l'Information et de la Communication), un centre de recherche appliquée en informatique situé à Charleroi. Vous trouverez toutes les coordonnées ici.

