



FORVIA LOCAL IT

Global Information Technologies (GIT)

RAPPORT DE STAGE
19 Fevrier - 14 Juin 2024

Présenté par
EL HAMIOUI Yassine

 **Forvia Seating Allenjoie**
<https://www.forvia.com/fr>
2958 Allée Henry Hugoniot, 25490 Allenjoie
EARNSHAW MALCOLM

 **IUT NFC - Réseaux & Télécoms**
<https://www.iut-nfc.univ-fcomte.fr>
4 Pl. Lucien Tharradin, 25200 Montbéliard
LECOQ JULIEN

IUT Nord Franche-Comté

RAPPORT DE STAGE FORVIA

Remerciements

Je tiens à exprimer ma profonde gratitude à Malcolm Earnshaw, mon manager et tuteur d'entreprise, pour son encadrement, ses conseils avisés et son soutien constant tout au long de mon stage chez Forvia FAS. Son expertise et sa disponibilité ont été des éléments clés dans la réussite de cette expérience enrichissante.

Je remercie également toute l'équipe IT pour leur accueil chaleureux et leur collaboration. Un grand merci à Baptiste Larrière, Jonathan Lépine, Adnan Baljic et Furkan Sertdemir pour leur aide précieuse, leurs conseils techniques et leur esprit d'équipe. Travailler à vos côtés a été une expérience formatrice et agréable.

Je souhaite exprimer ma gratitude à l'ensemble de l'entreprise Forvia FAS pour m'avoir donné l'opportunité de réaliser ce stage au sein de leurs équipes. Cette expérience m'a permis de développer de nombreuses compétences techniques et professionnelles qui me seront précieuses dans ma future carrière.

Je remercie chaleureusement Julien Lecoq, mon tuteur académique, pour sa visite et le temps qu'il a pu m'accorder. Son soutien et ses conseils ont été essentiels pour mener à bien ce stage.

Enfin, je tiens à remercier le département de formations Réseau & Télécoms pour m'avoir préparé aux défis rencontrés durant ce stage. La qualité de l'enseignement et les compétences acquises ont été déterminantes pour mon développement professionnel.

Merci à tous ceux qui, de près ou de loin, ont contribué à la réussite de ce stage et à mon épanouissement professionnel.

RAPPORT DE STAGE FORVIA

Abstracts

This internship report details my experience at Forvia FAS, where I served as a Local IT intern from February 19, 2024, to June 14, 2024. During this period, I engaged in a variety of projects and tasks, including the configuration and maintenance of IT systems, the development of monitoring dashboards, and the implementation of energy-saving solutions. The report highlights the application of Just-in-Time (JIT) principles in the IT environment, underscoring the importance of efficiency and precision in supporting the production processes. This experience has significantly enhanced my technical skills and professional development, preparing me for future challenges in the IT industry.

RAPPORT DE STAGE FORVIA

Table des matières

1 Présentation de l'Entreprise	1
1.1 Historique de l'entreprise	1
1.2 Activités principales de Forvia.....	1
1.2.1 Division Interiors	1
1.2.2 Division Clarion Electronics	1
1.2.3 Division Clean Mobility	1
1.2.4 Division Seating	2
1.3 Organisation et structure de l'entreprise	2
1.4 Description du site d'Allenjoie.....	2
1.5 Positionnement sur le marché	2
2 Contexte du Stage	3
2.1 Introduction au concept Just in Time (JIT)	3
2.2 Rôle du département IT dans une entreprise JIT	3
2.3 Importance des technologies IT pour Forvia FAS.....	4
3 Objectif du stage	5
3.1 Renforcement des Compétences Techniques	5
3.2 Support Utilisateur et Collaboration :	5
3.3 Développement Personnel et Professionnel :	5
4 Problématique.....	6
4.1 Présentation de la problématique	6
4.2 Justification de la problématique	6
5 Méthodologie et Outils Utilisés.....	7
5.1 Description de la méthodologie et des outils utilisés pour réaliser le projet	7
5.2 Choix de la méthodologie	7
5.3 Description des outils logiciels et moyens utilisés	7
5.4 Techniques de résolution de problèmes.....	10
5.5 Sources d'informations	10
6 Missions et Responsabilités	11
6.1 Description générale des missions confiées.....	11
6.3 Tâches récurrentes :.....	11
6.3.1 Configuration et maintenance des imprimantes	11
6.3.2 Remasterisation et installation de PC	11
6.3.3 Support utilisateur	11
6.3.4 Autres tâches IT (AutoLogon, script de comparaison, Setup PC, etc.)	11

RAPPORT DE STAGE FORVIA

7 Déroulement des Projets	12
7.1 Check_MK.....	12
7.2 Projet Dashboard orienté IT	12
7.3 Cluster RPi.....	13
7.4 Étude d'analyse d'une solution d'économie d'énergie	13
8 Développement des Projets	14
8.1 Détails sur la mise en place de CheckMK	14
8.1.1 Installation et Configuration	14
8.1.2 Configuration et Test des Agents SNMP.....	14
8.1.3 Test de Charge du CPU	15
8.1.5 Supervision de la Base de Données	17
8.1.7 Supervision d'un UPS	22
8.1.8 Supervision de XRDP sur Raspberry Pi grâce à un script en local.....	23
8.1.9 Conclusion du Projet CheckMK	25
8.2 Dashboard orienté IT	26
8.2.1 Introduction au Projet de Dashboard	26
8.2.2 Étude Préliminaire	26
8.2.3 Solutions envisageables avec API	27
8.2.4 Choix de l'Outil et Développement	27
8.2.5 Tentative d'utilisation d'un plugin de CheckMK sur Grafana	28
8.2.6 Utilisation de DataSource externe compatible API	29
8.2.7 Création d'un dashboard prototype en local.....	30
8.3 Cluster RPi.....	31
8.3.1 Développement du Cluster RPi	31
8.3.2 Introduction au Projet de Cluster RPi	31
8.3.3 Schéma et Fonctionnement du Cluster.....	31
8.3.4 Configuration du Réseau	32
8.3.5 Configuration des Raspberry Pi	32
8.3.6 Redondance et Haute Disponibilité	34
8.3.7 Haute Disponibilité avec NGINX.....	34
8.3.8 Prochaine Étape.....	35
8.3.9 Conclusion du Projet Cluster RPi.....	35
8.4 Analyse et recommandations pour l'économie d'énergie	36

RAPPORT DE STAGE FORVIA

8.4.1 Introduction au Projet d'Économie d'Énergie	36
8.4.2 Sensibilisation des Utilisateurs	36
8.4.3 Système de Gestion d'Énergie	37
8.4.4 Prises Programmables Digitales	37
8.4.5 Sensibilisation des Utilisateurs	38
8.4.6 Impact des Mesures sur la Consommation et les Coûts	38
8.4.7 Conclusion du Projet d'Économie d'Énergie	39
9 Missions Réalisées	39
9.1 Gestion des Imprimantes	39
9.1.1 Configuration	39
9.1.2 Maintenance	40
9.1.3 Cartographie des Imprimantes au sein de l'Usine	40
9.1.4 Enquête sur les soucis d'étiquettes	41
9.2 Gestion des PC	42
9.2.1 Remasterisation en PXE avec les Standards Forvia	42
9.2.2 Mise en Place d'AutoLogon	42
9.3 Script de Comparaison de Fichiers	43
9.4 Configuration des camera Raspberry PI EOL (End Of Line)	44
10. Conclusion et Perspectives	45
11 Annexes	46
11.1 Captures d'écran et illustrations des projets	46
11.2 Documents techniques et procédures créées	67
11.2.1 Procédure de vérification du statut des sièges dans IJCore (2 pages)	67
11.2.2 Procédure mise en place d'un cluster avec Haute disponibilité (4 pages) ...	69
11.2.3 Procédure de configuration des EOL CAMERA (5 pages)	73
12 Bibliographie	78
12.1 Sources documentaires et références utilisées	78
12.2 Acronyme	80

1 Présentation de l'Entreprise

1.1 Historique de l'entreprise

Le groupe Forvia est le fruit de la fusion de plusieurs entreprises au cours du XXème siècle. En 1998, le groupe Faurecia a vu le jour à la suite de la fusion de la société Bertrand Faure et du groupe ECIA, présentée lors du Mondial de l'Auto de Paris. Bertrand Faure, fondée en 1914, a débuté en fabriquant des sièges pour les métros parisiens, tandis que ECIA, fondée en 1987 et filiale du groupe PSA, produisait des composants automobiles tels que des lignes d'échappement et des sièges. Forvia est la fusion de Faurecia et Hella.

Depuis les années 2000, Faurecia n'a cessé de croître, intégrant des entreprises comme Clarion, Parrot Automotive et Coagent Electronics, consolidant ainsi Faurecia Clarion Electronics, une entité dédiée au développement du cockpit du futur. Le projet « Being Faurecia » a permis de transformer la culture d'entreprise, orientée vers la performance et la création de valeur, en créant une communauté mondiale de « Faurecians ».

En 2021, à la suite de sa scission du groupe Stellantis, Faurecia a été rebaptisé Forvia. Aujourd'hui, Forvia est implanté mondialement, couvrant l'Europe, l'Amérique du Nord et du Sud, l'Asie et l'Afrique.

1.2 Activités principales de Forvia

Forvia est structuré en quatre divisions principales :

1.2.1 Division Interiors

- Développe et fabrique des systèmes d'intérieurs complets pour véhicules, incluant planches de bord, panneaux de portes, consoles centrales et interfaces homme-machine.
- Améliore la qualité de vie à bord des véhicules et conçoit le cockpit du futur.
- Leader mondial des systèmes d'intérieurs avec 5,5 milliards d'euros de ventes, 89 sites de production et 18 centres de R&D.

1.2.2 Division Clarion Electronics

- Se concentre sur le développement du cockpit du futur avec des solutions audio digitales, des systèmes d'aide à la conduite et des solutions de connectivité.

1.2.3 Division Clean Mobility

- Conçoit des solutions innovantes pour une mobilité « zéro émission », en optimisant la qualité de l'air, l'efficacité énergétique et la performance acoustique.
- Leader mondial des solutions de mobilité durable avec 4,6 milliards d'euros de ventes, 88 sites de production et 9 centres de R&D.

1.2.4 Division Seating

- Développe et fabrique des systèmes de sièges automobiles en se focalisant sur la qualité, le confort et la sécurité.
- Leader mondial des solutions de sièges avec 7,4 milliards d'euros de ventes, 78 sites de production et 19 centres de R&D.

1.3 Organisation et structure de l'entreprise

Forvia est organisé en divisions spécialisées, chacune étant responsable de la recherche et du développement, de la production et des ventes de ses produits spécifiques. Le groupe est dirigé par un conseil d'administration et une équipe de direction globale qui coordonne les opérations à travers les différentes régions et divisions.

Avec une forte présence mondiale, Forvia dispose de nombreux sites de production et centres de R&D répartis sur plusieurs continents, permettant ainsi de répondre efficacement aux besoins des clients tout en innovant constamment dans ses domaines d'expertise.

1.4 Description du site d'Allenjoie

Le site Forvia Allenjoie FAS (Faurecia Seating), situé dans le département du Doubs en France, fait partie de la division Seating. Fondée en 1954 sous le nom de Faurecia Sièges France, l'usine est spécialisée dans la production de sièges automobiles pour divers types de véhicules, allant des citadines aux SUV.

Proche des sites du groupe Stellantis à Sochaux et Mulhouse, l'usine assemble les sièges pour six modèles de véhicules du groupe Stellantis, notamment les Peugeot 3008, 5008, 307, 308, DS7 et 508. Utilisant des technologies de pointe telles que la robotique sous IA, l'impression 3D et les matériaux composites, le tout interconnecté par des AGV (Automatic guided vehicle). Forvia Seating Allenjoie fabrique une large gamme de sièges, incluant des sièges à mémoire et à réglage électrique, chauffants et ventilés, options de massage avec une variété d'une centaine de sièges différents.

1.5 Positionnement sur le marché

Forvia occupe une position prépondérante dans l'industrie automobile mondiale, se classant parmi les dix plus grandes entreprises du secteur. Avec une présence dans 40 pays, près de 30 000 recrutements annuels et plus de 150 000 employés, dont environ +35 000 ingénieurs, Forvia joue un rôle crucial dans l'innovation et la production de technologies automobiles.

Le groupe fournit ses produits à de nombreux constructeurs automobiles réputés tels que le groupe Stellantis, exportant ses technologies à travers le monde, ainsi que plusieurs sites de développement sur les nouvelles technologies ainsi que les technologies développement durable, tel que l'hydrogène. Reconnue pour son engagement envers la mobilité durable, Forvia mène des initiatives pour réduire les émissions de CO2 et développer des solutions de mobilité « zéro émission ».

2 Contexte du Stage

2.1 Introduction au concept Just in Time (JIT)

Le Just in Time (JIT) est une méthode de gestion de la production visant à réduire les délais et les coûts en produisant uniquement ce qui est nécessaire, au moment où cela est nécessaire, et dans la quantité nécessaire. Originaire du Japon et popularisé par le système de production de Toyota, ce concept repose sur l'élimination des gaspillages, l'amélioration continue et une forte réactivité aux besoins du client.

Dans un système JIT, les matières premières et les composants sont livrés juste avant leur utilisation dans le processus de production, minimisant ainsi les stocks et les coûts de stockage. Cette méthode nécessite une excellente coordination entre les différents acteurs de la chaîne logistique et une grande flexibilité de la part des fournisseurs.

Le site d'Allenjoie, partie intégrante de Forvia FAS, opère selon le modèle JIT. Il reçoit des ordres de production via transmission informatique lorsqu'avant l'assemblage d'un nouveau véhicule sur la ligne de production du client, Stellantis Sochaux et Mulhouse. Ces ordres sont intégrés dans le système informatique de Siedoubs, transformés pour réaliser les différentes opérations nécessaires à l'assemblage du siège, qui sera livré et assemblé dans la voiture six heures plus tard. Ce mode de fonctionnement critique requiert une grande réactivité pour corriger rapidement tout problème afin d'éviter des pénalités de la part du client. (*voir Figure 1 : schéma système JIT chez Forvia*)

2.2 Rôle du département IT dans une entreprise JIT

Le département IT joue un rôle crucial dans le fonctionnement d'une entreprise JIT, comme le site de Forvia FAS. Les technologies de l'information permettent de gérer et d'optimiser les flux d'informations et de matières tout au long du processus de production. Voici quelques-unes des responsabilités clés du département IT dans un environnement JIT :

- **Transmission des ordres de production** : Le département IT assure la réception et l'intégration des ordres de production envoyés par le client dans le système informatique de l'usine. Cela garantit que les informations nécessaires pour commencer la production sont disponibles en temps réel.
- **Suivi et traçabilité** : Les systèmes informatiques permettent de suivre chaque étape de la production, de la réception des matières premières à l'assemblage final et à la livraison des produits. Cette traçabilité est essentielle pour garantir la qualité et la conformité des kits de sièges livrés.
- **Monitoring** : L'utilisation de Check_MK pour le monitoring permet de surveiller en temps réel les performances des systèmes informatiques et de production. Cela inclut la détection précoce des anomalies et des défaillances potentielles permettant une réactivité pour éviter les interruptions de la production et minimiser les temps d'arrêt.
- **Réactivité en cas de problème** : En cas de problème sur la ligne de production, le département IT doit être capable de réagir rapidement pour diagnostiquer et résoudre les problèmes techniques. Cela inclut la gestion des incidents, la mise en œuvre de solutions temporaires et la coordination avec d'autres départements pour minimiser les impacts sur la production.

Le service informatique du site de Forvia FAS se compose de 5 membres principaux : **Malcolm Earnshaw** (Responsable IT et mon maître d'apprentissage), **Baptiste Larrière** (Spécialiste IT), **Adnan Baldjic** (IT local) et **Jonathan Lépine** (IT local), **Furkan Sertdemir** (apprenti IT local), ainsi que moi-même, stagiaire IT. Nous sommes les principaux interlocuteurs pour les questions informatiques sur le site. Notre mission principale est d'assurer la disponibilité des services informatiques et de contribuer au développement du site, une priorité pour le groupe dans le cadre de la digitalisation de l'entreprise. Plus de cela, l'équipe IT se constraint de respecter les choix du groupe Forvia. (choix du matériels, les logiciels utilisée, tel que Check_MK, FITS, etc..)

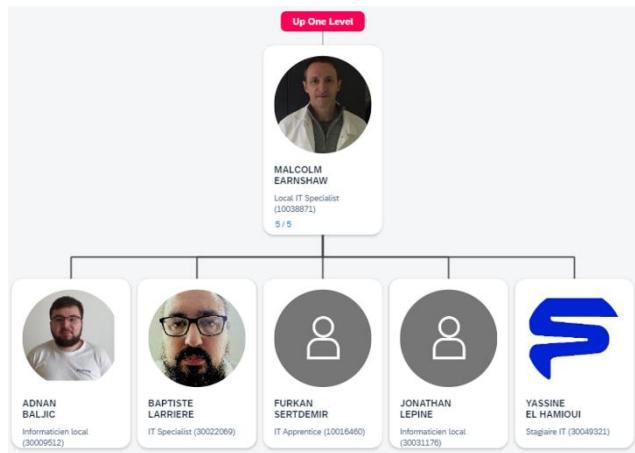


Figure 2 : Organigramme Équipe IT

Forvia FAS est une **usine 4.0**, caractérisée par une forte intégration des technologies numériques dans ses processus de production. Notre activité principale consiste à fournir des services IT au sein de l'entreprise de production. En outre, nous jouons un rôle clé dans le développement de nouveaux projets de digitalisation, l'amélioration des performances, la réduction des coûts internes et l'amélioration de l'expérience utilisateur.

En complément de l'expertise de Malcolm Earnshaw, nous bénéficions également du soutien de différentes équipes IT pour les différents logiciels ou besoin du Shopfloor.

2.3 Importance des technologies IT pour Forvia FAS

Pour Forvia FAS, les technologies IT sont essentielles pour plusieurs raisons :

- **Efficacité opérationnelle** : L'utilisation de systèmes informatiques avancés permet d'optimiser les processus de production, de réduire les délais et les coûts, et d'améliorer la qualité des produits.
- **Réactivité et flexibilité** : Les technologies IT permettent à l'entreprise de répondre rapidement aux changements de la demande du client, en ajustant la production en temps réel. Cela est particulièrement crucial dans un environnement JIT, où les délais sont très courts.
- **Gestion des risques** : Les solutions IT aident à anticiper et à gérer les risques potentiels, en fournissant des données précises et en temps réel sur les opérations. Cela permet de prendre des décisions éclairées et de mettre en place des mesures préventives.
- **Innovation et compétitivité** : En investissant dans des technologies de pointe, Forvia FAS peut développer des produits innovants et rester compétitive sur le marché mondial. Cela inclut l'utilisation de la robotique, de l'impression 3D, et de l'Internet des objets (IoT) pour améliorer la production et les produits.
- **Satisfaction client** : Les systèmes IT permettent de garantir la qualité et la conformité des produits livrés, ainsi que de respecter les délais de livraison. Cela est crucial pour maintenir la satisfaction du client et renforcer les relations commerciales.

3 Objectif du stage

L'objectif principal de mon stage chez Forvia était de contribuer à l'amélioration et à l'optimisation des infrastructures IT de l'entreprise tout en développant mes compétences techniques et professionnelles. Plus précisément, mes objectifs étaient les suivants :

3.1 Renforcement des Compétences Techniques :

- Maîtriser les outils et technologies utilisés par le département IT, tels que Check_MK (outil de monitoring), les logiciels de suivis, de gestion, de développement de l'infrastructure et du fonctionnement de l'usine.
- Développer des solutions IT innovantes et participer à leur intégration dans l'infrastructure existante.

3.2 Support Utilisateur et Collaboration :

- Fournir un support technique efficace aux utilisateurs et documenter les procédures pour renforcer leur autonomie et leurs connaissances par des formations.
- Collaborer étroitement avec les équipes internes pour assurer une coordination optimale des projets IT.

3.3 Développement Personnel et Professionnel :

- Acquérir des compétences en gestion de projet, en planification et en organisation.
- Améliorer mes compétences en communication et en travail d'équipe, tout en contribuant à la vision de digitalisation de l'entreprise.

Ces objectifs visaient à renforcer mon expertise technique, à améliorer l'efficacité des infrastructures IT de Forvia, et à promouvoir mon développement professionnel dans un environnement industriel exigeant.

4 Problématique

4.1 Présentation de la problématique

La problématique principale de mon stage chez Forvia FAS se situe au niveau de l'optimisation et de la sécurisation des infrastructures IT dans un contexte de production en flux tendu (Just in Time - JIT). Dans une telle configuration, la moindre interruption ou inefficacité peut entraîner des retards de production significatifs et des coûts supplémentaires. La problématique se décompose en plusieurs aspects spécifiques.

« Comment optimiser et sécuriser les infrastructures IT de Forvia FAS dans un environnement de production en flux tendu (Just in Time - JIT) pour minimiser les interruptions et améliorer l'efficacité opérationnelle ? »

4.2 Justification de la problématique

La justification de cette problématique repose sur plusieurs points clés :

Contexte de Production en Flux Tendu (JIT) : Forvia ALS opère dans un environnement JIT où les délais sont très courts et où la réactivité est cruciale. Toute interruption des systèmes IT peut provoquer des arrêts de production coûteux et des pénalités de retard.

Importance de la Surveillance et de la Sécurité : Une surveillance proactive des systèmes permet de détecter les anomalies avant qu'elles n'affectent la production. Cela réduit les temps d'arrêt imprévus et améliore la fiabilité des opérations.

Efficacité et Réduction des Coûts : L'automatisation des processus IT permet de gagner du temps, de réduire les erreurs et d'allouer les ressources humaines à des tâches à plus forte valeur ajoutée. Cela contribue à une meilleure efficacité opérationnelle et à une réduction des coûts.

En abordant cette problématique, le stage vise à renforcer l'infrastructure IT de Forvia FAS, à améliorer la réactivité et l'efficacité opérationnelle, et à promouvoir des pratiques durables dans l'entreprise.

5 Méthodologie et Outils Utilisés

5.1 Description de la méthodologie et des outils utilisés pour réaliser le projet

Pour réaliser les différents projets et tâches durant mon stage chez Forvia FAS, j'ai utilisé une méthodologie basée sur la gestion visuelle de projet avec Kanban, couplée à un suivi approfondi à l'aide de l'outil Notion. Cette approche m'a permis de gérer efficacement les priorités, de suivre l'avancement des projets, et de collaborer de manière fluide avec l'équipe IT.

5.2 Choix de la méthodologie

La méthode Kanban a été choisie pour sa simplicité et son efficacité dans la gestion des flux de travail. En utilisant des tableaux Kanban, j'ai pu visualiser les tâches à effectuer, en cours, et terminées, ce qui m'a aidé à mieux gérer mon temps et mes priorités. Cette méthode favorise également la transparence et la collaboration au sein de l'équipe.

5.3 Description des outils logiciels et moyens utilisés

Voici les outils et logiciels principaux utilisés :



Figure 3 : Logo Notion



Figure 4 : Logo Github



Figure 5 : Logo Visual Studio Code



Figure 6 : Logo Word



Figure 7 : Logo Excel

Yassine EL Hamiou
RAPPORT DE STAGE FORVIA

Notion

Notion est un outil de gestion de projet et de prise de notes tout-en-un. Il permet de créer des bases de données, des tableaux Kanban, des calendriers, et des documents collaboratifs.

J'ai utilisé Notion pour structurer et organiser toutes mes notes, gérer mes projets avec un tableau Kanban, et suivre mes tâches au quotidien. Notion m'a permis de centraliser toutes les informations pertinentes et de maintenir une organisation claire et efficace.

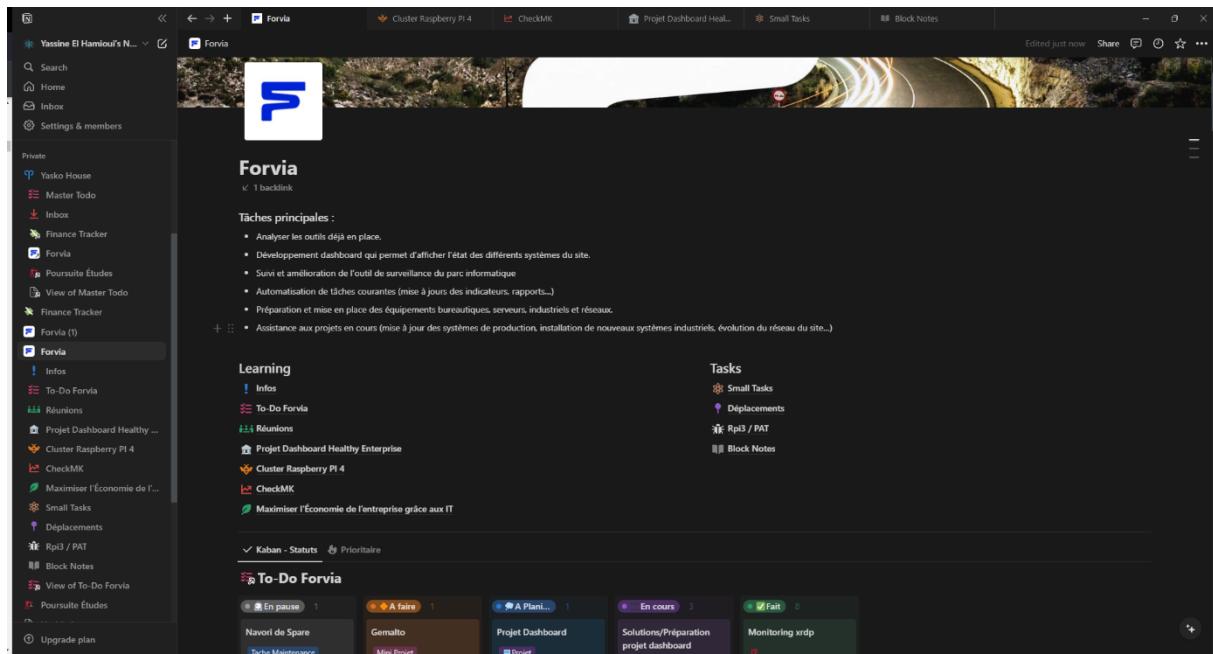


Figure 8 : Page d'accueil Notion de mon stage chez Forvia

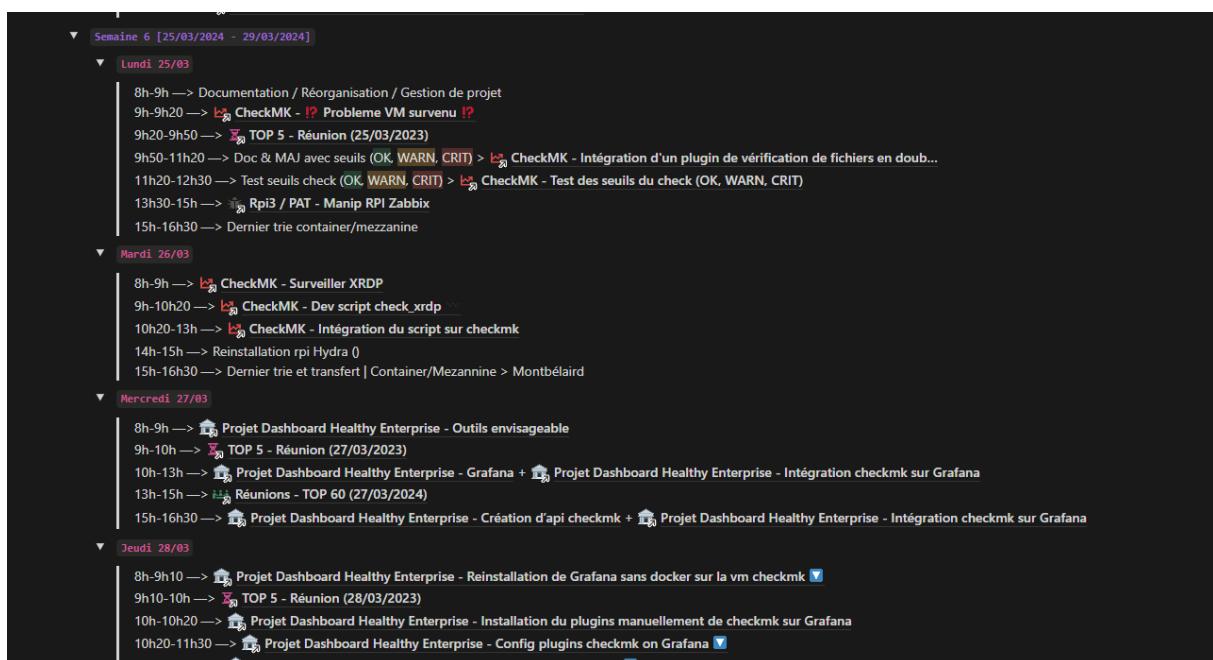


Figure 9 : Suivis d'activité de Stage chez Forvia

Github

J'utilise GitHub pour le versioning de tout ce que je fais. Il me permet de stocker, gérer et suivre les modifications de mes projets. Grâce à GitHub, je peux garder une trace de toutes les modifications apportées au code, et assurer la sauvegarde de mes travaux.

Lien vers mon GitHub : https://github.com/Yasko0x000/Stage_Local_IT_Forvia

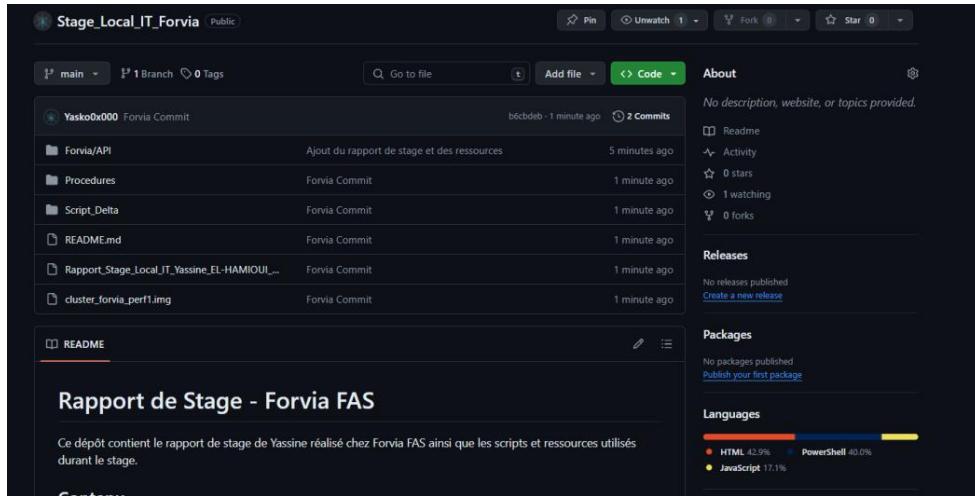


Figure 10 : GitHub de Yassine Stage Forvia

Réunions TOP 5

Chaque matin, l'équipe IT se réunit pour discuter des événements récents, des tâches à venir, et des priorités du jour. Ces réunions sont appelées "TOP 5".

Depuis deux mois, je suis responsable de la prise de notes durant ces réunions. Je rédige un mail résumant les discussions et les actions à entreprendre, ce qui permet d'informer toute l'équipe, y compris les collègues de tournée (matin et après-midi).

Ce mail de synthèse assure une communication fluide et permet de disposer d'une liste à suivre, améliorant ainsi la coordination et l'efficacité de l'équipe.

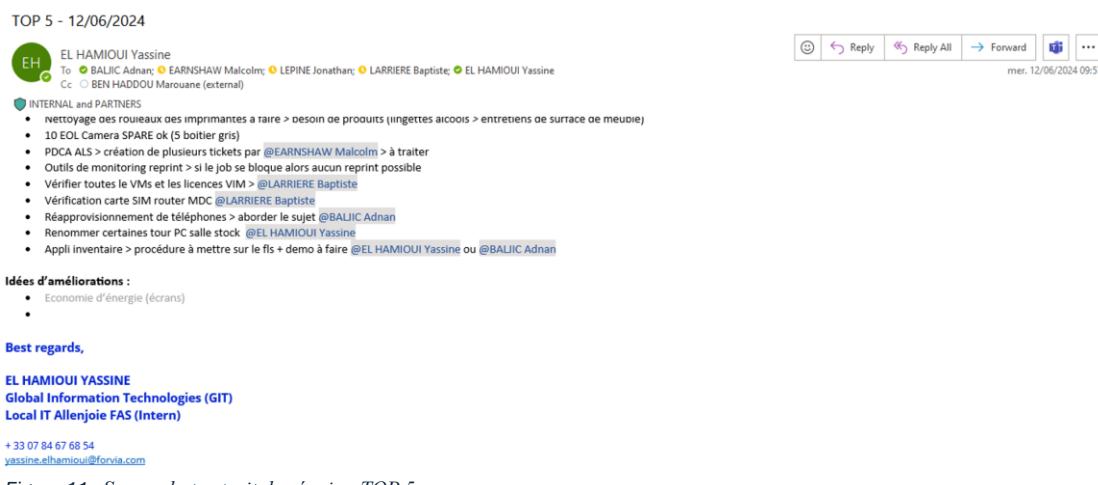


Figure 11 : Screenshot extrait de réunion TOP 5

5.4 Techniques de résolution de problèmes

Pour résoudre les problèmes rencontrés durant le stage, j'ai utilisé les techniques suivantes :

Analyse des Causes Racines :

- Identification des problèmes récurrents.
- Mise en place de solutions temporaires pour contenir le problème.
- Développement de solutions durables et réalisation de tests approfondis pour valider leur efficacité.

Par exemple, lors de la demande de mise en place d'un Raspberry Pi pour l'affichage du flux de livraison dans l'usine devant le SRS (stock des kits avant livraison), j'ai rencontré un problème. En effet, pour installer un Raspberry Pi, il devait être équipé d'une image Raspbian standardisée par Forvia. L'image OS qui m'a été fournie était obsolète. Pour dépanner, nous avons donc installé un mini PC à la place du Raspberry Pi en attendant la résolution du problème lié au Raspberry Pi. De plus, il y avait un petit souci de règle dans le pare-feu, nécessitant une demande de modification auprès du service de sécurité informatique.

Collaboration et Feedback :

- Consultation régulière avec les membres de l'équipe pour obtenir des avis et des suggestions.
- Mise en œuvre des retours pour améliorer les solutions proposées.

Documentation et Prise de Note :

- Création de procédures documentées pour les tâches récurrentes.
- Prise de notes systématique pour assurer un suivi efficace et structuré des actions entreprises sur Notion.

5.5 Sources d'informations

Pour mener à bien mes projets, j'ai utilisé diverses sources d'informations :

Documentation Interne : Guides et procédures de l'entreprise

Ressources en Ligne : Documentation officielle des outils utilisés (Check_MK, Docker, etc.) ainsi que des forums et communautés techniques pour résoudre des problèmes spécifiques.

Formation et Support : Sessions de formation internes, support technique des fournisseurs de logiciels et matériels.

En adoptant cette méthodologie et en utilisant ces outils, j'ai pu gérer efficacement mes projets, améliorer la communication et la collaboration au sein de l'équipe IT, et contribuer à l'optimisation des infrastructures IT de Forvia FAS.

6 Missions et Responsabilités

6.1 Description générale des missions confiées

En tant que stagiaire Local IT, mes missions principales étaient centrées sur la gestion et l'optimisation des infrastructures IT de l'entreprise. Cela comprenait la surveillance des systèmes, la mise en place de nouvelles solutions IT, le support utilisateur, et la gestion proactive des ressources informatiques. Mon rôle était d'assurer la continuité et l'efficacité des opérations IT dans un environnement de production en flux tendu.

6.3 Tâches récurrentes :

6.3.1 Configuration et maintenance des imprimantes

J'ai régulièrement configuré et maintenu les imprimantes pour garantir leur bon fonctionnement. Une cartographie des imprimantes au sein de l'usine a également été réalisée pour une gestion optimisée de ces équipements. (Voir chapitre 9.1 Gestion des Imprimantes, page 40)

6.3.2 Remasterisation et installation de PC

La remasterisation des PC a été une tâche importante, visant à standardiser les configurations selon les besoins spécifiques de l'entreprise. J'ai également installé de nouveaux PC et mis en place des configurations AutoLogon pour améliorer l'efficacité des utilisateurs. (Voir chapitre 9.2 Gestion des PC, page 42)

6.3.3 Support utilisateur

J'ai fourni une assistance technique aux utilisateurs pour résoudre divers problèmes informatiques et les ai formés sur les nouvelles procédures et outils, assurant ainsi une utilisation optimale des ressources IT.

6.3.4 Autres tâches IT (AutoLogon, script de comparaison, Setup PC, etc.)

J'ai développé divers scripts pour automatiser des tâches courantes, comme la comparaison de fichiers. J'ai également géré la sécurité des PC avec BitLocker et créé des procédures pour les GAP Leaders. En outre, j'ai enquêté sur les problèmes d'étiquettes des imprimantes et coordonné avec le fournisseur pour résoudre les problèmes techniques. J'ai rédigé et envoyé des résumés des réunions TOP 5, assurant une communication fluide et un suivi des actions entreprises au sein de l'équipe IT tous les jours. Enfin j'ai configuré des camera avec Raspberry Pi, qui seront utilisé comme Spare dans le cas où les camera actuellement installé assurant la traçabilité des commandes tombent en panne.

7 Déroulement des Projets

7.1 Check_MK

Le projet Check_MK a commencé par une phase de découverte et de compréhension de l'outil, réalisée sur ma propre machine. J'ai d'abord installé Check_MK en local pour me familiariser avec ses fonctionnalités et son interface. Pour cette phase, j'ai imaginé divers scénarios de surveillance afin de tester les capacités de Check_MK. Cela m'a permis de définir des objectifs clairs pour ce que je voulais accomplir avec l'outil en liaison avec le système déjà mis en place chez Forvia FAS. A savoir, Check_MK est renommée Hydra au seins de l'entreprise.

Après avoir acquis une bonne compréhension de Check_MK, j'ai entrepris de l'installer sur Docker, exposé sur le port 8080 et hébergé sur une VM dédiée. J'ai ensuite configuré les agents SNMP pour collecter les données des différents systèmes et effectué des tests approfondis pour assurer leur bon fonctionnement.

Un des aspects importants de ce projet était la supervision des serveurs Apache, où j'ai mis en place des règles spécifiques pour monitorer des pages web comme "forvia.html". En outre, j'ai travaillé sur la création et la gestion de bases de données (BDD), en attribuant des droits et en analysant des requêtes pour configurer des seuils d'alerte sur Hydra.

Pour améliorer encore la supervision, j'ai développé des scripts en Python pour détecter les fichiers en double et intégré des plugins Nagios. Tous ces tests et configurations étaient basés sur des objectifs que je me suis fixé, en alignement avec les besoins du système déjà en place chez Forvia FAS.

7.2 Projet Dashboard orienté IT

Le projet Dashboard IT avait pour objectif de créer un tableau de bord clair et ciblé, améliorant la visibilité sur les performances et les états des systèmes par rapport à Hydra. Le projet a débuté par une phase de conception où j'ai identifié les indicateurs clés de performance (KPI) nécessaires pour le tableau de bord.

J'ai choisi d'utiliser Grafana pour son intégration facile avec divers outils de surveillance. La première étape consistait à installer Grafana et à configurer les plugins Hydra pour collecter les données nécessaires sans nécessiter d'API additionnelles.

Ensuite, j'ai développé et testé plusieurs itérations du tableau de bord, pour affiner les indicateurs affichés et assurer la pertinence des informations présentées.

7.3 Cluster RPi

Le projet Cluster RPi visait à créer une infrastructure de calcul distribué en utilisant des Raspberry Pi 4. Le projet a débuté par l'acquisition du matériel nécessaire, suivi de l'installation de Raspbian OS sur chaque Raspberry Pi.

La configuration réseau a été une étape cruciale, nécessitant la mise en place de VLAN et la gestion du switch pour assurer une communication efficace entre les Raspberry Pi. J'ai ensuite configuré Docker Swarm pour la clusterisation, permettant de répartir les charges de travail et d'assurer la redondance.

Un défi majeur a été la gestion des interfaces réseau et l'intégration des Raspberry Pi dans l'infrastructure d'une haute disponibilité. Le projet s'est terminé par la documentation de la configuration.

7.4 Étude d'analyse d'une solution d'économie d'énergie

L'étude d'analyse d'une solution d'économie d'énergie avait pour but de réduire la consommation énergétique des systèmes IT de Forvia FAS. Le projet a commencé par une phase de sensibilisation, pour mettre en place des politiques de gestion de l'énergie, notamment pour les écrans.

Ensuite, j'ai identifié prises programmables et digitales pour optimiser la consommation d'énergie des équipements IT. Cette phase a impliqué des tests simulés pour évaluer l'impact des nouvelles pratiques et simuler l'impact financier.

Les résultats ont montré des économies significatives, confirmées par des estimations détaillées.

8 Développement des Projets

8.1 Détails sur la mise en place de CheckMK

Le projet CheckMK a été une initiative clé pour comprendre le fonctionnement de l'outil utilisé par l'entreprise pour la surveillance des systèmes IT de Forvia Allenjoie.

8.1.1 Installation et Configuration

Pour commencer, j'ai manipulé CheckMK sur ma propre machine pour comprendre pleinement son fonctionnement et pour avoir tous les accès nécessaires. Cette phase d'apprentissage initiale m'a permis de tester diverses fonctionnalités et de me fixer des objectifs clairs en lien avec les besoins de Forvia ALS. Une fois cette phase terminée, j'ai installé CheckMK sur Docker, exposé sur le port 8080 et hébergé sur une VM dédiée. Cette configuration a permis une surveillance efficace et centralisée des systèmes IT.

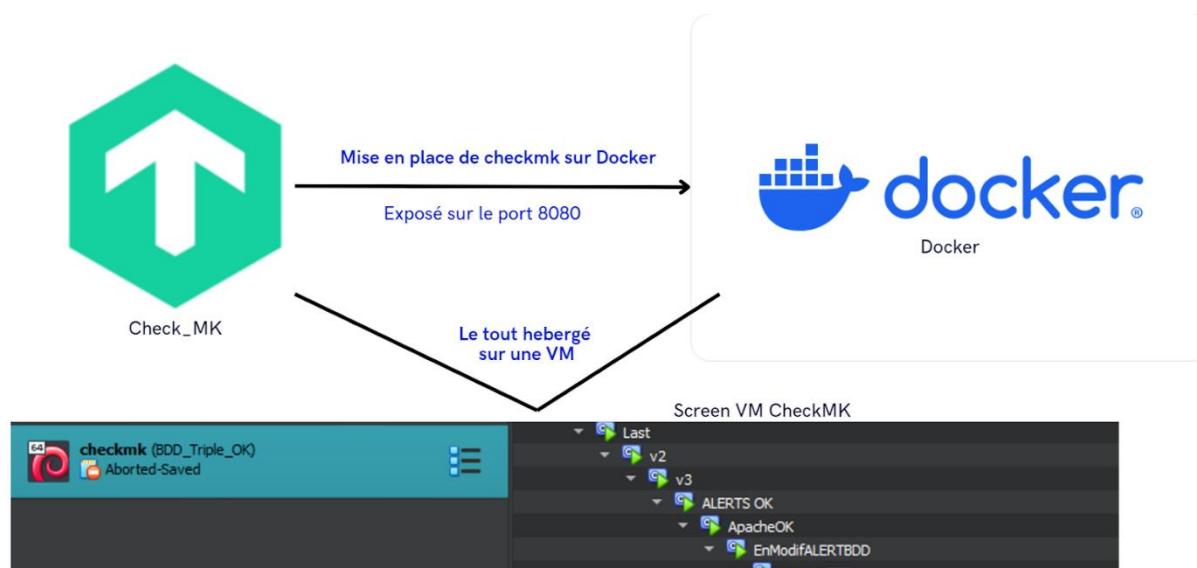


Figure 12 : Schéma d'installation et de configuration de Hydra

8.1.2 Configuration et Test des Agents SNMP

Ensuite, j'ai configuré les agents SNMP pour collecter les données des différents systèmes. SNMP, ou Simple Network Management Protocol, est un protocole standard utilisé pour collecter et organiser des informations sur les dispositifs gérés sur des réseaux IP et pour modifier ces informations afin de modifier le comportement du dispositif. J'ai effectué des tests approfondis pour assurer la précision des données collectées et le bon fonctionnement des agents. Ces tests incluaient des vérifications de la connectivité (PING), de la configuration des agents et de la collecte des données via SNMP. (voir Figure 13 : Screen test de PING/AGENT/SNMP)

8.1.3 Test de Charge du CPU

Pour évaluer la performance sous pression, j'ai testé la surcharge du CPU en exécutant un script. Ce script simple simule une surcharge du CPU en exécutant une boucle infinie, ce qui augmente l'utilisation du CPU de manière significative. Voici le script utilisé :

```

1 #!/bin/bash
2 Load_cpu() {
3     while true; do
4         true
5         echo "Le CPU pleure"
6     done
7 }
8 load_cpu

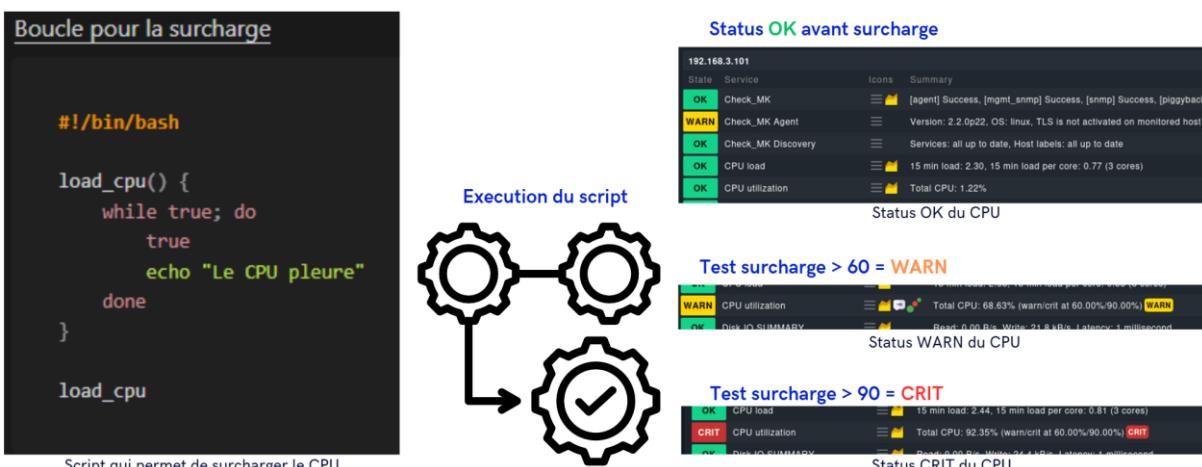
```

Figure 14 : Boucle pour surcharger le CPU

Explication du script :

1. **Load_cpu()** { ... } : Cette ligne définit une fonction nommée Load_cpu. Tout le code entre les accolades {} fait partie de cette fonction.
2. **while true; do true; echo "le CPU pleur"; done** : Cette boucle while est infinie car la condition true est toujours vraie. À chaque itération de la boucle, l'expression true est évaluée, ce qui consomme une petite quantité de CPU, puis la chaîne "le CPU pleur" est imprimée à l'écran. Cette boucle continue de s'exécuter sans fin, ce qui utilise continuellement le CPU et crée une surcharge.
3. **load_cpu** : Cette ligne appelle la fonction Load_cpu définie précédemment, démarrant ainsi la boucle infinie qui surcharge le CPU.

J'ai ensuite configuré les seuils d'alerte de manière à générer des alertes lorsque la surcharge du CPU dépassait 60% (warning - WARN) et 90% (critique - CRIT). Cela m'a permis de vérifier que les alertes étaient générées correctement en fonction de l'utilisation du CPU, ce qui est essentiel pour une surveillance proactive des systèmes.



8.1.4 Supervision du Serveur Apache

Un des aspects importants du projet était la supervision des serveurs Apache. J'ai mis en place des règles spécifiques pour surveiller des pages web comme "forvia.html". Cette supervision incluait la configuration de règles HTTP, où l'option URL était cochée pour superviser uniquement la page "forvia.html". J'ai également configuré des tests pour vérifier la présence de contenu spécifique sur les pages web en indiquant dans "fixed string to expect in the content" une chaîne de caractères fixe à retrouver dans le contenu : "Test Forvia OK".



Figure 16 : Screen règle Check HTTP service

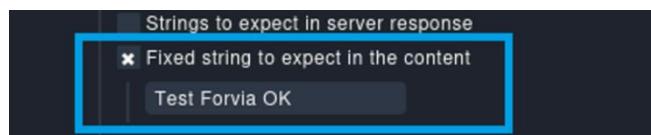


Figure 17 : Screen Expect the content

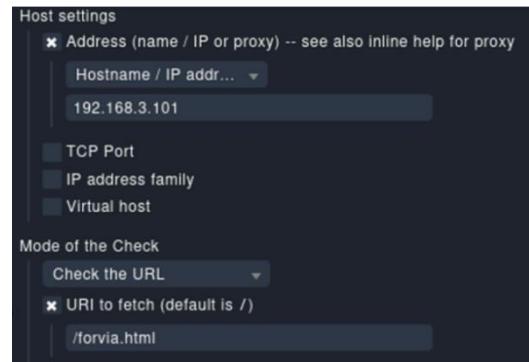


Figure 18 : Screen config de la règle

Par la suite, j'ai réalisé deux tests :

Premier test :

La page ne contenait pas le texte "Test Forvia OK" (voir figure web forvia.html sans texte Forvia OK). On voit bien dans la figure "check forvia.html sans texte Forvia OK" que CheckMK nous affiche un statut CRIT en rouge, ce qui indique que l'alerte a bien fonctionné et que le texte est manquant sur la page web.



Figure 19 : Test du check sans le texte "Test Forvia OK"

Deuxième test :

La page contenait le texte "Test Forvia OK" (voir figure web forvia.html avec texte Forvia OK). On voit dans la figure "check forvia.html avec texte Forvia OK" que CheckMK affiche un statut OK, confirmant que le texte est bien présent.



Figure 20 : Test du check avec le texte "Test Forvia OK"

8.1.5 Supervision de la Base de Données

8.1.5.1 Préparation de la Base de Données

J'ai également travaillé sur la supervision des bases de données, car Forvia utilise des bases de données, en particulier avec SCADA, une entreprise qui fournit un tableau de bord complet du flux de l'usine, de la production en cours, de la livraison, etc. Pour cela, j'ai créé une base de données (BDD), attribué les droits nécessaires et analysé les requêtes pour configurer des seuils d'alerte sur CheckMK. Les niveaux d'alerte étaient configurés pour générer des alertes en fonction des valeurs de la base de données.

```
root@raspberry:/var/www/html# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 30
Server version: 10.5.23-MariaDB-0+deb11u1 Debian 11

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE suivis_prod_kits;
Query OK, 1 row affected (0,003 sec)

MariaDB [(none)]> CREATE USER 'forvia'@'localhost' IDENTIFIED BY 'toto';
Query OK, 0 rows affected (0,017 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON suivis_prod_kits.* TO 'forvia'@'localhost';
Query OK, 0 rows affected (0,002 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,003 sec)
```

Figure 21 : Création de la Base de données

Pour commencer les tests et la simulation, j'ai créé une table nommée "production_tracking". Dans cette dernière, j'ai inséré quelques données, incluant des sièges A ou B et un statut attribué de 0 ou 1. Si le siège est en statut 0, cela signifie qu'il n'est pas présent, donc c'est un retard ou un problème. Quand le statut du siège est à 1, cela signifie qu'il est prêt.

```
MariaDB [suisvis_prod_kits]> SELECT * FROM production_tracking;
+----+-----+-----+-----+
| id | timestamp           | site_name | status |
+----+-----+-----+-----+
| 1  | 2024-02-29 14:24:17 | Siège A   | 0      |
| 2  | 2024-02-29 14:24:17 | Siège A   | 1      |
| 3  | 2024-02-29 14:24:17 | Siège B   | 0      |
| 4  | 2024-02-29 14:24:17 | Siège B   | 1      |
| 5  | 2024-02-29 14:24:17 | Siège A   | 1      |
```

Figure 22 : Simulation de donnée

8.1.5.2 Analyse de la Requête

Analysons la requête SQL suivante :

```

1  SELECT
2    CASE
3      WHEN COUNT(status) = 0 THEN 0
4      WHEN COUNT(status) <= 3 THEN 1
5      ELSE 2
6    END AS alert_level
7  FROM
8    production_tracking
9  WHERE
10   status = 0;

```

Figure 23 : Requête SQL pour la supervision de la base de données

Cette requête SQL a pour objectif de déterminer le niveau d'alerte basé sur le nombre de statuts égaux à zéro dans la table production_tracking.

- **Fonction COUNT** : Elle compte le nombre d'occurrences du statut égal à zéro.
- **Attribution de niveau d'alerte** :
 - Si le nombre de statuts égaux à zéro est 0, le niveau d'alerte est 0 (OK).
 - Si le nombre de statuts égaux à zéro est compris entre 1 et 3, le niveau d'alerte est 1 (WARN).
 - Si le nombre de statuts égaux à zéro est supérieur à 3, le niveau d'alerte est 2 (CRIT).

Sur CheckMK, le statut 0 signifie OK, le statut 1 signifie WARNING et le statut 2 signifie CRITICAL. Par conséquent, aucun statut 0 indique un état OK, entre 0 et 3 statuts 0 indique un WARNING, et plus de 3 statuts 0 indique un état CRITICAL.

The diagram illustrates the analysis of the provided SQL query and its configuration in a monitoring tool like Hydra. On the left, the SQL code is shown:

```

SQL
SELECT
CASE
    WHEN COUNT(status) = 0 THEN 0
    WHEN COUNT(status) <= 3 THEN 1
    ELSE 2
END AS alert_level
FROM
    production_tracking
WHERE
    status = 0;

```

On the right, four circular icons explain the logic:

- Fonction COUNT > Nombre de statut égal à zéro
- Attribution de niveau d'alerte > 0, 1, ou 2
- 0 Status 0 = OK
0-3 Status 0 = WARN
+3 Status 0 = CRIT
- Configuration des seuils d'alerte sur Hydra

A red arrow points from the third icon to a configuration box for Hydra:

Upper levels for first output item
Warning at 1.00
Critical at 2.00

Figure 24 : Analyse de la requête SQL & Seuils d'alertes

8.1.5.3 Check des Niveaux d'Alerte

J'ai effectué trois tests pour les trois niveaux d'alerte : OK, WARN et CRIT.

Premier test : Tous les statuts 0 sont changés en 1, ce qui signifie que la base de données contient uniquement des sièges prêts. Comme montré dans la figure dessous, Hydra affiche un statut vert OK.

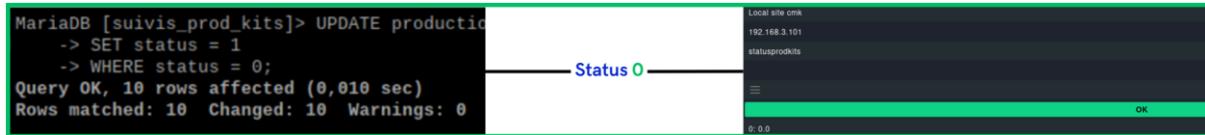


Figure 25 : Test du check status 0 Hydra (soit OK)

Deuxième test : J'ai ajouté trois statuts 0 aux sièges pour simuler un warning, donc la base de données contient entre 0 et 3 sièges en retard ou avec des problèmes. Comme montré dans la figure dessous, Hydra affiche un statut jaune WARN.

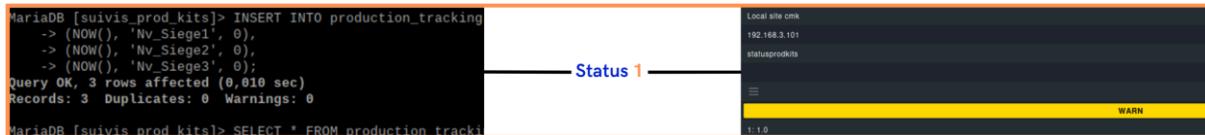


Figure 26 : Test du check status 1 Hydra (soit WARN)

Troisième test : Tous les statuts 1 sont changés en 0 pour simuler une situation critique, donc la base de données contient uniquement des sièges avec des problèmes. Comme montré dans la figure dessous, Hydra affiche un statut rouge CRIT.

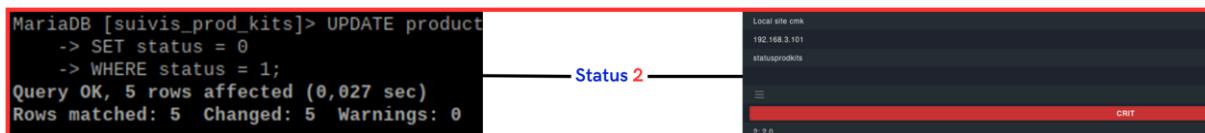


Figure 27 : Test du check status 2 Hydra (soit CRIT)

8.1.6 Détection de Fichiers en Double

Pour améliorer encore la supervision, j'ai développé un script en Python pour détecter les fichiers en double. Ce script a été intégré avec des plugins Nagios pour permettre une surveillance efficace via Hydra. J'ai testé ce script en local avant de le configurer sur Hydra pour assurer son bon fonctionnement.



Figure 28 : schéma du script de détection de fichiers double

8.1.6.1 Analyse du Script Python

Voici le script utilisé pour détecter les fichiers en double :

Première partie	Deuxième partie
<pre> 1 import os 2 3 def find_duplicate_files(folder_path): 4 file_checksums = {} 5 6 for root, dirs, files in os.walk(folder_path): 7 for file in files: 8 file_path = os.path.join(root, file) 9 with open(file_path, 'rb') as f: 10 file_content = f.read() 11 checksum = hash(file_content) 12 file_checksums[file_path] = checksum 13 14 duplicate_files = {} 15 for file_path, checksum in file_checksums.items(): 16 if list(file_checksums.values()).count(checksum) > 1: 17 if checksum not in duplicate_files: 18 duplicate_files[checksum] = [] 19 duplicate_files[checksum].append(file_path) 20 21 return duplicate_files </pre>	<pre> 22 23 folder_path = "/home/forvia/double" 24 duplicates = find_duplicate_files(folder_path) 25 26 num_duplicates = sum(len(files) for files in duplicates.values()) 27 if duplicates: 28 for checksum, files in duplicates.items(): 29 print(f"- Doublon (checksum : {checksum}) :") 30 for file_path in files: 31 print(f" - {file_path}") 32 if num_duplicates >= 5: 33 print("Trop de doublons détectés.") 34 exit(2) # Statut CRITICAL 35 else: 36 print("Quelques doublons détectés.") 37 exit(1) # Statut WARNING 38 else: 39 print("Aucun doublon trouvé dans le dossier.") 40 exit(0) # Statut OK </pre>

Figure 29 : script python de file_duplicate.py pt.1 & pt.2

Explication du script :

- Importation et Définition de la Fonction :** Le script commence par importer le module os et définir la fonction find_duplicate_files, qui prend en paramètre le chemin du dossier à analyser.
- Calcul des Checksums :** La fonction parcourt tous les fichiers dans le dossier spécifié, lit le contenu de chaque fichier et calcule un checksum (empreinte) unique pour le contenu du fichier.
- Stockage des Checksums :** Les checksums sont stockés dans un dictionnaire file_checksums où le chemin du fichier est la clé et le checksum est la valeur.
- Identification des Doublons :** La fonction vérifie ensuite si un checksum apparaît plus d'une fois. Si c'est le cas, elle ajoute ces fichiers à un dictionnaire duplicate_files.
- Retour des Doublons :** La fonction retourne le dictionnaire des fichiers en double.
- Exécution et Mise à Jour du Statut de CheckMK :** Le script exécute la fonction sur le dossier spécifié, compte le nombre total de doublons et met à jour le statut de CheckMK en fonction du nombre de doublons détectés (OK, WARNING, CRITICAL).

8.1.6.2 Test Script en local & config CheckMK

Pour tester le script, j'ai créé trois fichiers identiques avec un contenu identique "Toto" (voir Figure test du script en local). Ensuite, j'ai configuré le plugin Nagios en mettant une description "Vérification des fichiers double". J'ai indiqué une commande SSH qui me permet d'accéder directement à la machine et d'exécuter la commande en local (voir 'configuration du plugin Nagios'). J'ai ensuite coché l'option "Command line" et saisi la commande nécessaire pour exécuter le script sur la machine distante (voir 'Analyse commande'). Cette commande incluait l'utilisation de sshpass pour fournir le mot de passe, ssh pour la connexion SSH, et python pour exécuter le script Python sur l'hôte distant. J'ai également veillé à inclure l'option -o StrictHostKeyChecking=no pour ignorer la vérification de la clé SSH, et spécifié l'adresse IP de l'appareil Raspberry Pi.



Figure 30 : Test Script en local & config hydra

8.1.6.3 Check des Niveaux d'Alerte

Premier test : J'ai vidé le dossier "double". Étant donné qu'il n'y a pas de fichier, le monitoring de CheckMK affiche le statut OK, soit "Aucun doublon trouvé dans le dossier" (voir Figure Pas de doublons et Figure Hydra pas de doublons).

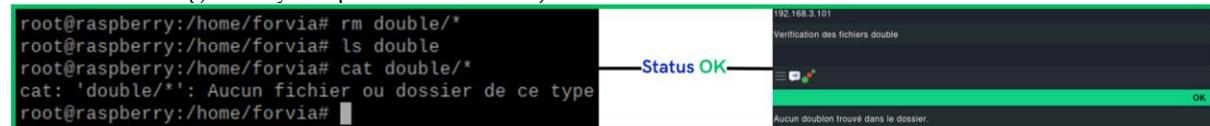


Figure 31 : Check Hydra pas de fichier en double

Deuxième test : J'ai créé trois fichiers identiques dans le dossier "double", contenant "Toto", afin de simuler entre 1 et 4 doublons dans le dossier. CheckMK affiche bien le statut WARN avec la description "Quelques doublons détectés" et liste les doublons en question (voir Figure Entre 1 & 4 doublons et Figure Hydra entre 1 & 4 doublons).

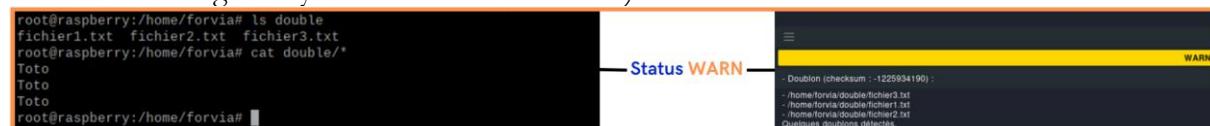


Figure 32 : Check Hydra entre 1 & 4 doublons

Troisième test : J'ai créé cinq fichiers identiques pour simuler le statut CRIT. CheckMK affiche le statut rouge CRIT avec la description "Trop de doublons détectés" et liste les doublons en question (voir Figure Plus de 5 doublons et Figure Hydra plus de 5 doublons).

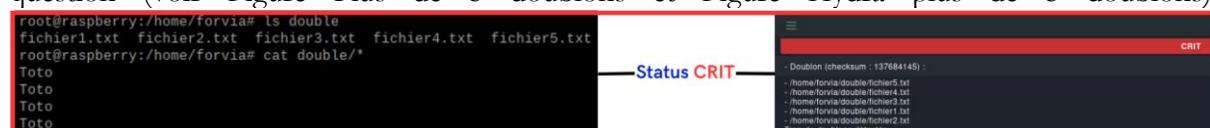


Figure 33 : Check Hydra 5 ou plus de fichiers en double

8.1.7 Supervision d'un UPS

Chez Forvia, les onduleurs (UPS) sont utilisés pour alimenter certaines équipements critiques de l'usine. J'ai donc intégré la supervision d'un onduleur dans CheckMK. Cette intégration comprenait la configuration de SNMPv3 depuis l'interface web de l'UPS et l'intégration complète dans Hydra (CheckMK) pour une surveillance sans erreurs.

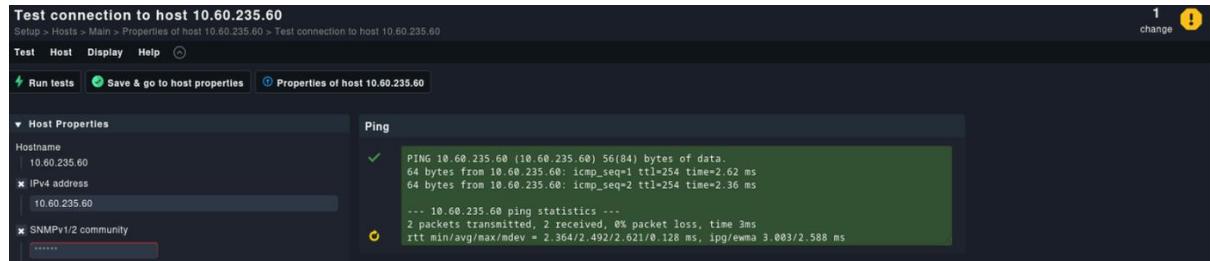


Figure 34 : Hydra test connexion host UPS

8.1.7.1 Configuration SNMP Hydra et UPS

Pour commencer, j'ai accédé à l'interface web de l'UPS. Dans les paramètres de configuration, j'ai créé un utilisateur nommé adminforvia et j'ai configuré SNMPv3 en spécifiant cet utilisateur ainsi que l'adresse IP de l'UPS. Ensuite, dans CheckMK, j'ai intégré le monitoring de l'onduleur en configurant SNMP avec les paramètres de l'UPS. Cette configuration permet une surveillance continue et précise de l'état de l'onduleur, garantissant que toute anomalie soit rapidement détectée et corrigée.

Conf SNMPv3 depuis WEB UPS

Configure SNMPv3 Access Control

Access Control

Access

Enable

User Name

adminforvia

NMS IP/Host Name

10.60.235.51

Apply Cancel

Intégration UPS sur Hydra

Monitoring agents

Checkmk agent / API integrations ...

SNMP ...

SNMP credentials ...

Security Level: No authentication, no privacy

Security name:

Figure 35 : Configuration SNMP Hydra et UPS

Local site cmk, 10.60.235.60		Icons	Summary	Age	Checked	Perf-O-Meter
OK	Check_MK	[snmp]	Success, [piggyback] Success (but no data found for this host), execution time 0.7 sec	8.17 s	8.17 s	710 ms
OK	Check_MK Discovery	[green dot]	Services: all up to date, Host labels: all up to date	114 s	114 s	
OK	APC Symmetra status	[green dot]	Battery status: normal, Output status: on line (calibration invalid), Capacity: 100%, Time remaining: 14 hours 23 minutes	7.18 s	7.18 s	
OK	Interface 2	[green dot]	[vmac0], (up), MAC: 28:29:86:30:25:B3, Speed: unknown	7.18 s	7.18 s	
OK	Phase Input	[green dot]	Voltage: 239.0 V	7.18 s	7.18 s	239 V
OK	Phase Output	[green dot]	Voltage: 230.0 V, Current: 0.0 A, Load: 0%	7.18 s	7.18 s	0%
OK	Self Test	[green dot]	Result of self test: OK, Date of last test: 03/18/2024	7.18 s	7.18 s	
OK	SNMP Info	[green dot]	APC Web/SNMP Management Card (MB:v4.2.9 PF:v1.0.16 PN:apc_hw21_aos_1.1.0.16 bin AF1:v1.0.16 AN1:apc_hw21_su_1.1.0.16 bin MN:AP9640 HR:5 SN: ZA2010550704 MD:03/05/2020) (Embedded PowerNet SNMP Agent SW v2.2 compatible), FRETULPS0017, IDF17, Unknown	7.18 s	7.18 s	
OK	System events	[green dot]	No service events	7.18 s	7.18 s	
OK	Temperature	[green dot]	22.0 °C	7.18 s	7.18 s	22 °C
OK	Uptime	[green dot]	Up since Mar 19 2024 09:15:42, Uptime: 1 hour 26 minutes	7.18 s	7.18 s	86 m

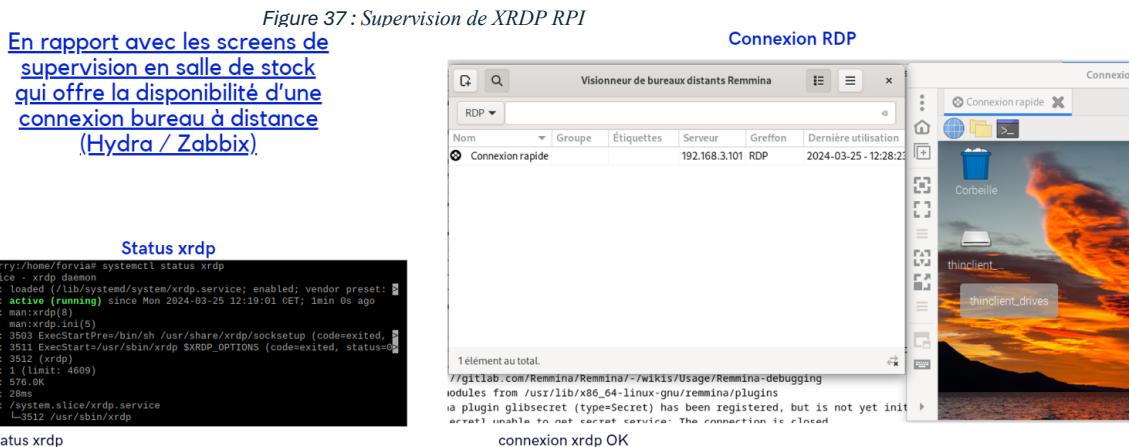
Figure 36 : Intégration de l'UPS sans erreurs

8.1.8 Supervision de XRDP sur Raspberry Pi grâce à un script en local

Pour compléter, j'avais comme mission de mettre en place un écran et un Raspberry Pi 4 dans notre salle de stock pour avoir accès au monitoring de l'entreprise. La plupart du temps, lorsqu'on veut faire une modification, on y accède à distance via le protocole XRDP. J'ai donc pensé qu'il serait utile de pouvoir le superviser pour détecter les pannes avant qu'elles ne deviennent critiques, surtout en cas d'urgence nécessitant un accès rapide. J'ai configuré la supervision de XRDP sur un Raspberry Pi, en utilisant des scripts bash pour vérifier l'état du service. Les résultats de cette supervision ont été intégrés dans Hydra pour une surveillance continue. Contrairement au script de détection de fichiers en double, qui exécutait des commandes SSH directement depuis la machine cliente, pour cette supervision j'ai utilisé des scripts dans un répertoire de l'hôte que CheckMK peut accéder pour une exécution locale.

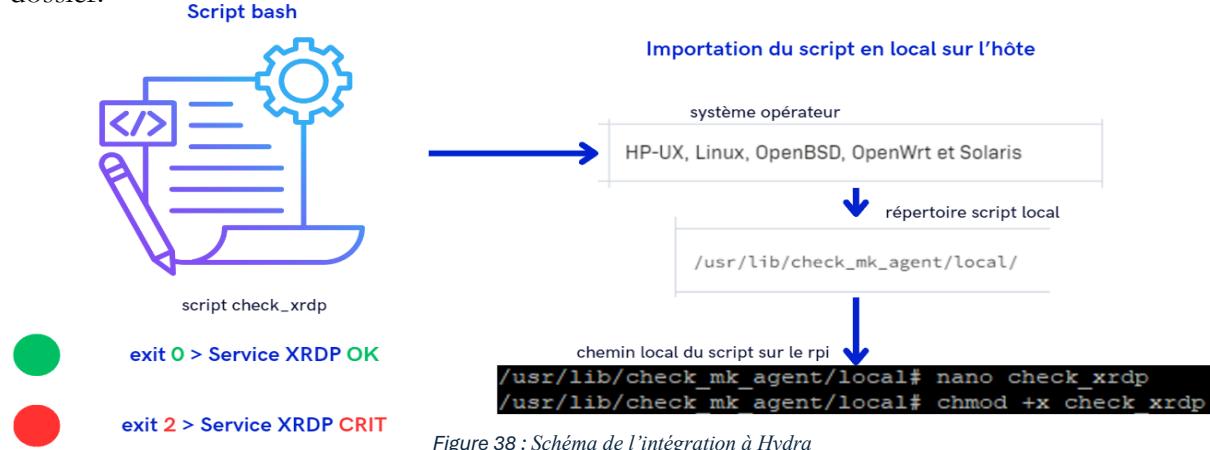
8.1.8.1 XRDP sur RPI

Avec la commande `systemctl status xrdp`, on peut vérifier que XRDP est bien opérationnel, comme montré dans la figure *status xrdp*. De plus, la figure *connexion xrdp OK* montre que la connexion est également fonctionnelle.



8.1.8.2 Intégration du script sur CheckMK

Côté CheckMK, les scripts locaux de chaque hôte se trouvent dans le répertoire `/usr/lib/check_mk_agent/local`. CheckMK détecte automatiquement les scripts présents dans ce dossier.



8.1.8.3 Analyse du script

Dans ce script, nous utilisons systemctl is-active pour vérifier si le service XRDП est actif sur le système. Si le service est actif, le script renvoie un code de statut OK (0) avec un message indiquant que XRDП est en cours d'exécution. Sinon, il renvoie un code de statut CRITICAL (2) avec un message indiquant que XRDП n'est pas en cours d'exécution.

```

1  #!/bin/bash
2
3  if systemctl is-active --quiet xrdp; then
4      # Service XRDП en cours d'exécution, renvoie un code OK
5      echo "0 XRDП_Check - XRDП est en cours d'exécution."
6      exit 0
7  else
8      # Service XRDП non en cours d'exécution, renvoie un code CRIT
9      echo "2 XRDП_Check - XRDП n'est pas en cours d'exécution."
10     exit 2
11 fi

```

Figure 39 : Script check_xrdp

Format de sortie : Chaque ligne de sortie doit être formatée de manière spécifique pour que CheckMK puisse l'interpréter correctement. Dans ce script, la ligne de sortie suit ce format : <code de statut> <nom du service> - <message>. Par exemple, "0 XRDП_Check - XRDП est en cours d'exécution." indique que le service XRDП fonctionne correctement.

Codes de statut : Les codes de statut standard sont utilisés pour indiquer l'état du service vérifié. Dans ce script, un code de statut 0 est utilisé pour indiquer que tout va bien (OK), et un code de statut 2 est utilisé pour indiquer une condition critique (CRITICAL).

8.1.8.4 Check des niveaux d'alertes

Après avoir placé le script dans le bon répertoire, on peut se rendre dans le service de découverte (Discovery Service) de CheckMK. Si le script est correctement formaté, il apparaîtra dans "Undecided services - currently not monitored".

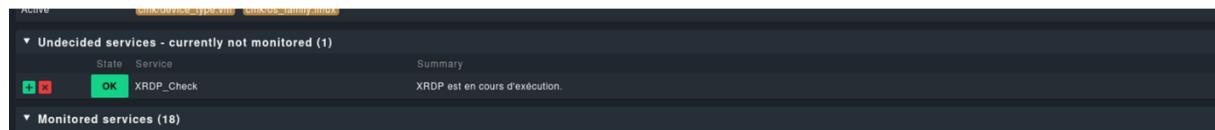


Figure 40 : Service discovery xrdp_check

Ensuite, on l'ajoute avec le bouton "+" pour le déplacer vers les services monitordés.

J'ai ensuite effectué deux tests pour simuler les statuts OK et CRIT :

- Pour le statut OK, j'ai laissé le service activé



Figure 41 : Check service XRDUP

- Pour le statut CRIT, j'ai arrêté le service (voir figure *Hydra check xrdp up*).



Figure 42 : Check service XRD Down

Enfin, de la même façon que le check XRDUP, j'ai également ajouté un check du service SSH en relation avec le script qui vérifie les fichiers en double dans le répertoire /home/forvia/double en utilisant SSH depuis CheckMK. (voir pt. 8.1.6 Détection de Fichiers en Double)

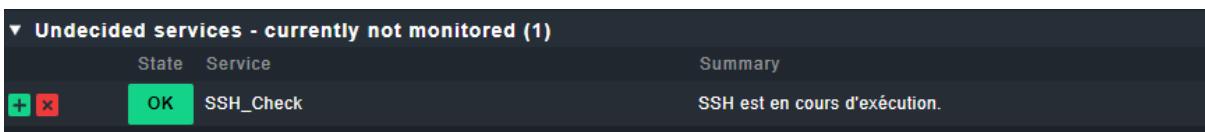


Figure 43 : Service discovery ssh_check

8.1.9 Conclusion du Projet CheckMK

Le projet Check_MK a été une opportunité idéale pour moi de comprendre le fonctionnement de l'outil de monitoring utilisé par Forvia. En ayant tous les droits sur ma machine personnelle et sur des VMs, j'avais un contrôle total sur toutes les actions. Grâce à l'installation sur Docker et la configuration des agents SNMP, j'ai pu assurer une collecte précise des données. Les tests de surcharge CPU, la supervision des serveurs Apache et des bases de données, ainsi que la détection des fichiers en double et la surveillance de XRDUP et SSH sur Raspberry Pi, ont permis une gestion proactive des infrastructures.

8.2 Dashboard orienté IT

Le projet de création d'un Dashboard orienté IT visait à simplifier et optimiser la surveillance des systèmes informatiques de Forvia.

8.2.1 Introduction au Projet de Dashboard

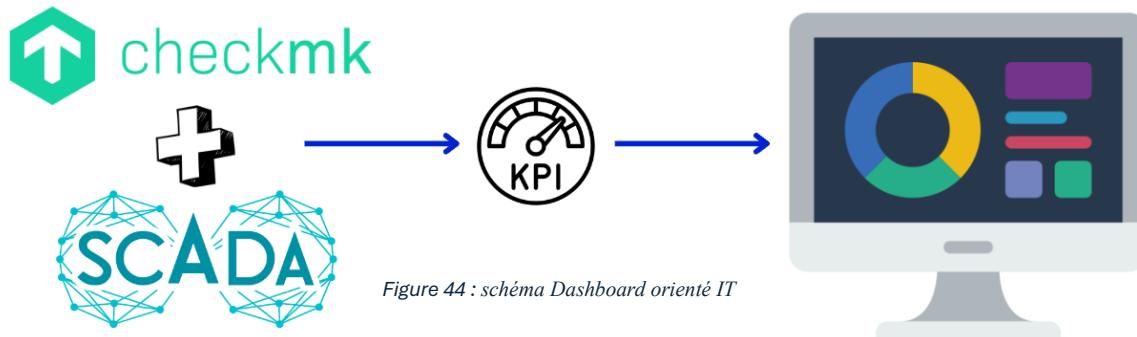
Dans un contexte où les données sur Hydra (CheckMK) sont omniprésentes, la capacité à filtrer et présenter les informations clés est essentielle pour une prise de décision rapide et efficace. Actuellement, nous utilisons Hydra, un outil de monitoring complet, mais souvent surchargé d'informations. Le projet avait pour but de créer un tableau de bord ciblant les points clés, éliminant ainsi le bruit pour ne garder que les signaux pertinents.

8.2.2 Étude Préliminaire

Une analyse des outils de monitoring existants a été réalisée pour identifier les forces et les faiblesses de CheckMK et SCADA.

- **CheckMK :**
 - **Forces :** Vue exhaustive des performances et de l'état de santé des infrastructures.
 - **Faiblesses :** Surcharge d'informations, complexité dans l'identification rapide des problèmes critiques.
- **SCADA :**
 - **Forces :** Suivi en temps réel, contrôle des opérations à distance.
 - **Faiblesses :** Pertinence des données limitée pour l'équipe IT, interface complexe.

L'objectif était de créer un nouveau dashboard intégrant les points forts de ces outils tout en minimisant leurs faiblesses.



8.2.3 Solutions envisageables avec API

Pour le développement du Dashboard, j'ai envisagé deux solutions principales :

Utilisation d'API :

1. **Développement de A à Z du Dashboard (avec intégration API)** : Cette approche implique de coder entièrement le dashboard moi-même, en utilisant des technologies comme HTML, CSS, et JavaScript. L'avantage principal de cette méthode est que la seule limite est l'imagination. Cependant, cette solution nécessite une maintenance continue, ce qui peut être contraignant à long terme.
2. **Utilisation de logiciels OpenSource (avec intégration API)** : Cette méthode consiste à utiliser des logiciels open-source comme Grafana, Kibana ou InfluxData. L'avantage de cette approche est qu'elle ne nécessite pas de maintenance régulière, car ces outils sont constamment mis à jour par leurs communautés respectives. Toutefois, il faut adapter nos besoins aux fonctionnalités proposées par ces logiciels.

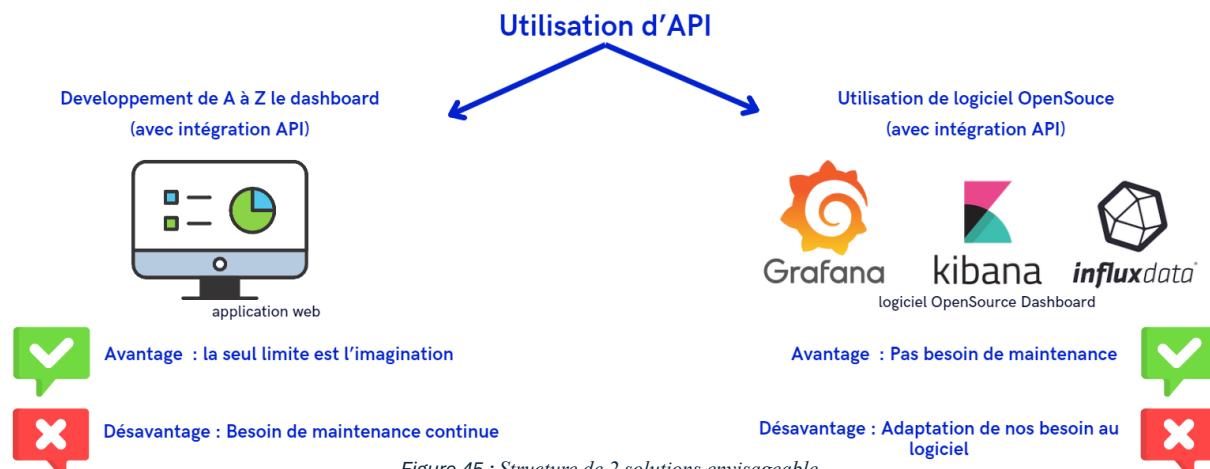


Figure 45 : Structure de 2 solutions envisageable

8.2.4 Choix de l'Outil et Développement

Pour le développement du dashboard, j'ai choisi Grafana, un outil open-source populaire pour la visualisation des données.

Installation de Grafana :

```

1 apt-get install -y apt-transport-https software-properties-common wget
2 mkdir -p /etc/apt/keyrings/
3 wget -q -O - https://apt.grafana.com/gpg.key | gpg --dearmor | sudo tee /etc/apt/keyrings/grafana.gpg > /dev/null
4 echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://apt.grafana.com stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list
5 echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://apt.grafana.com beta main" | sudo tee -a /etc/apt/sources.list.d/grafana.list
6 apt-get install grafana-enterprise -y

```

Figure 46 : Commandes d'installation de Grafana

Intégration avec CheckMK :

Pour intégrer Grafana avec CheckMK, j'ai créé une API utilisateur sur CheckMK. J'ai d'abord essayé d'utiliser un plugin de CheckMK sur Grafana, mais cela n'a pas entièrement répondu à nos besoins. J'ai ensuite configuré Grafana pour utiliser CheckMK comme source de données via une API REST.

8.2.5 Tentative d'utilisation d'un plugin de CheckMK sur Grafana

Pour configurer le plugin CheckMK sur Grafana, je suis allé dans Data Source > Add New Connection > CheckMK et j'ai sélectionné CheckMK.

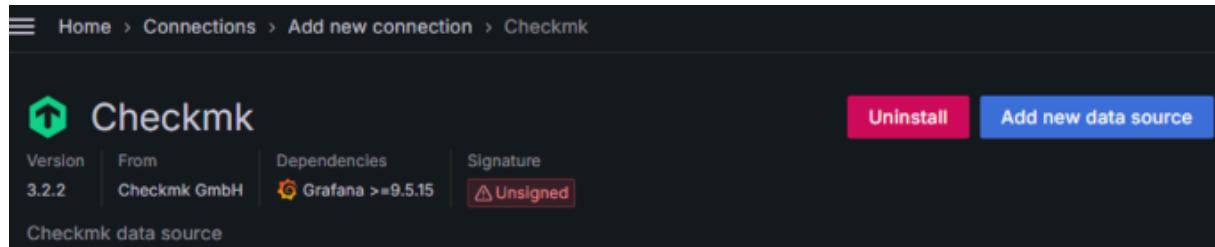


Figure 47 : Installation CheckMK plugins sur Grafana

8.2.5.1 Configuration du plugin CheckMK

Pour configurer le plugin, j'ai suivi plusieurs étapes. Tout d'abord, j'ai attribué un nom au plugin, en l'occurrence "CheckMK Plugins Forvia". Ensuite, j'ai spécifié l'URL de l'API de mon instance CheckMK, qui est "http://10.60.235.51:8080/cmk".

J'ai ensuite renseigné les informations d'identification nécessaires : le nom d'utilisateur préalablement créé dans CheckMK et un mot de passe API généré spécifiquement pour cette configuration.

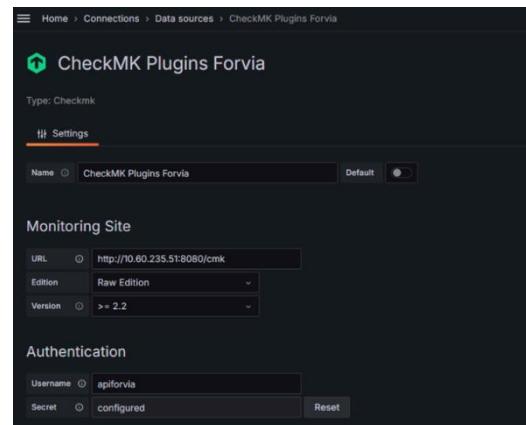
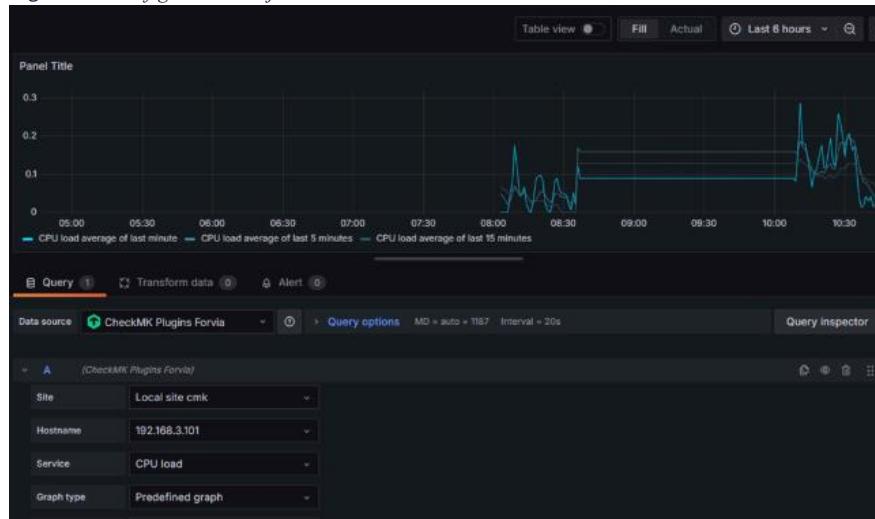


Figure 48 : Configuration de CheckMK plugins sur Grafana

8.2.5.2 Ajout du plugin CheckMK sur le dashboard

Pour ajouter le plugin au dashboard, je suis allé dans Dashboard > Create Dashboard > Add Visualization. Ensuite, j'ai sélectionné la Data Source (par exemple "CheckMK plugins Forvia"), puis dans Site "Local site cmk", et enfin dans Hostname "192.168.3.101" (l'adresse IP de mon client cible, le Raspberry Pi sur la VM). Enfin, dans Service, j'ai choisi par exemple "CPU load".

Figure 49 : Configuration et ajout au dashboard



Remarque : le plugin Check MK à des limites



Avec le Plugins seulement les services basique sont disponible

Tout script ou intégration ajouté manuellement n'est pas disponible

8.2.6 Utilisation de DataSource externe compatible API

Pour surmonter les limitations du plugin, j'ai installé une datasource externe JSON API depuis [Grafana](#). Cette datasource permet d'importer des données via API sur le dashboard.

8.2.6.1 Configuration DATA SOURCE JSON API

La configuration de la datasource JSON API se fait de manière similaire à celle du plugin CheckMK. J'ai indiqué le nom, l'URL de l'API afin de se connecter, puis une "basic authentication" avec user = apiforvia. (voir annexe fig 77 Config JSON API)

8.2.6.2 Ajout JSON API sur le Dashboard

Pour cette section, nous allons examiner comment j'ai intégré la source de données JSON API dans le tableau de bord, ainsi que la manière d'explorer les données de l'API Checkmk. Nous aborderons également la compréhension des requêtes fields.

8.2.6.2.1 REST API Integration GUI pour l'exploration de data API

Je suis allé sur la page du REST API Integration GUI pour l'exploration de données API (à l'adresse http://10.60.235.51:8080/cmk/check_mk/api/1.0/ui/ ou en sélectionnant "REST API interactive GUI" depuis le menu help). J'ai ensuite sélectionné mon serveur. (voir figure Figure 78 : Série d'étapes d'exploration de donnée json API)

8.2.6.2.2 Test d'ajout de la version de Grafana sur CheckMK

Pour tester l'ajout de la version de Grafana sur CheckMK, j'ai recherché la version dans le REST API Interactive GUI en accédant à l'URL : http://10.60.235.51:8080/cmk/check_mk/api/1.0/version

J'ai ensuite créé ou utilisé une datasource en indiquant l'URL de CheckMK : http://10.60.235.51:8080/cmk/check_mk/, ainsi que l'Authentication et le type d'Authentication : Basic Authentication avec apiforvia, puis j'ai enregistré et testé. Ensuite, je suis allé sur le dashboard et j'ai ajouté une visualisation en indiquant la datasource configurée pour CheckMK. (voir figure Figure 79 : Série d'étapes configuration json api sur le dashboard)

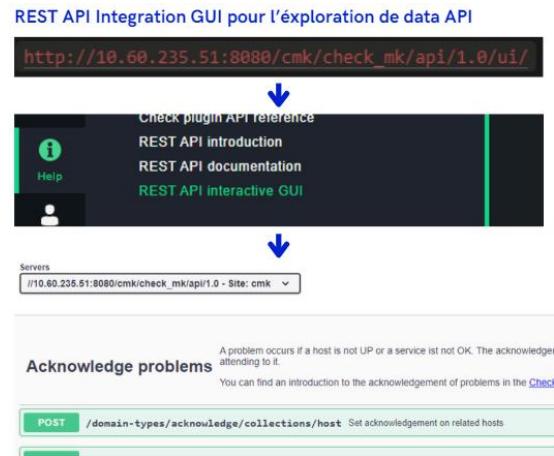


Figure 50 : Série d'étapes d'exploration de données JSON API

Figure 51 : Série d'étapes configuration JSON API sur le dashboard

8.2.6.2.3 Comprendre comment faire une requête fields

Pour comprendre comment faire une requête fields, j'ai commencé par entrer l'URL complète de l'API dans un navigateur : http://10.60.235.51:8080/cmk/check_mk/api/1.0/version. Le résultat affichera les informations de la version.

```
{
  "site": "cmk",
  "group": "",
  "rest_api": {
    "revision": "0"
  },
  "versions": {
    "apache": [2, 4, 52],
    "checkmk": "2.2.0p22.cre",
    "python": "3.11.5 (main, Nov 30 2023, 14:57:54) [GCC 13.2.0]",
    "mod_wsgi": [4, 9, 4],
    "wsgi": [1, 0]
  }
}
```

Figure 52 : Analyse JSON

Donc le format des requêtes est simple, on commence par le symbole \$ représentant le début de l'objet JSON, puis on sépare avec un . tout les arguments, nous voulons afficher la version qui est >2.2.0p22.cre donc dans "checkmk" qui lui même est dans "versions" .

Résumé :

- Le symbole \$ représente le début de l'objet JSON.
- versions est une clé dans l'objet JSON racine.
- checkmk est une clé dans l'objet versions contenant la version spécifique de CheckMK.

8.2.7 Crédation d'un dashboard prototype en local

Sur ce dashboard, j'ai manipulé et exploré la solution de développement d'un dashboard. Bien que je n'aie pas pu finaliser ce projet, j'ai documenté une procédure détaillée pour la continuation du projet. Voici un aperçu du prototype de dashboard.

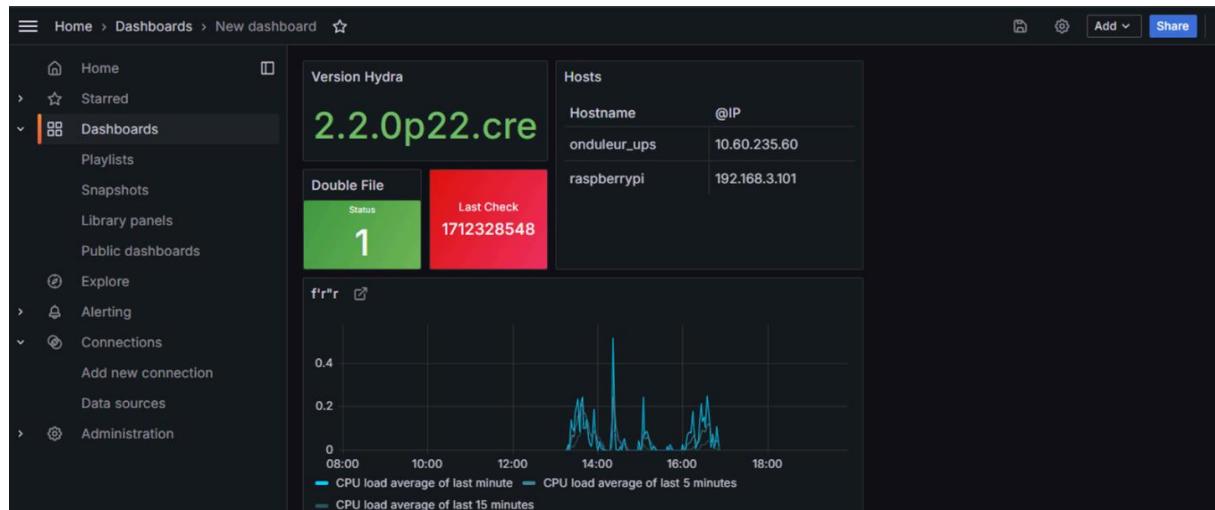


Figure 53 : Dashboard Prototype

Ce projet a posé les bases d'une gestion IT plus efficace, permettant à l'équipe IT de Forvia d'avoir une vue d'ensemble et de réagir rapidement aux incidents. Les détails fournis couvrent l'ensemble des étapes, des configurations, des tests et des résultats obtenus lors du projet de création du Dashboard orienté IT, mettant en évidence les efforts et les compétences développées tout au long du projet.

8.3 Cluster RPi

8.3 Développement du Cluster RPi

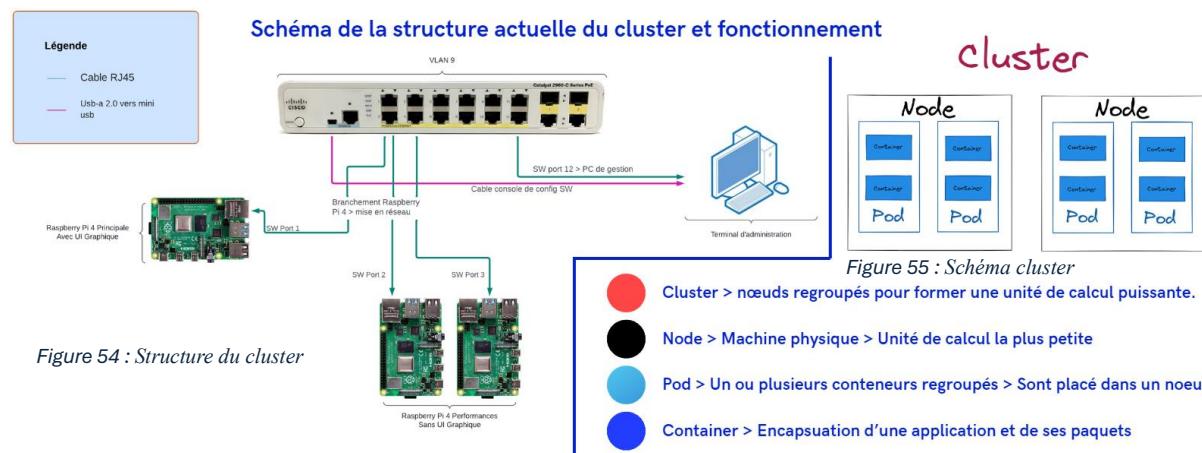
Le projet de développement d'un Cluster Raspberry Pi avait pour objectif d'améliorer la performance et la fiabilité des infrastructures IT en utilisant une approche de clustering avec des Raspberry Pi.

8.3.1 Introduction au Projet de Cluster RPi

Le Cluster Raspberry Pi a été conçu pour fournir une solution de calcul distribuée à moindre coût. En utilisant plusieurs Raspberry Pi, j'ai pu créer un environnement de calcul capable de supporter des charges de travail diverses tout en garantissant une redondance et une haute disponibilité.

8.3.2 Schéma et Fonctionnement du Cluster

Le schéma suivant illustre la structure actuelle du cluster et son fonctionnement.



- **Nœuds :**
 - Les nœuds sont les machines physiques qui forment l'unité de calcul du cluster.
- **Pods :**
 - Un pod est un groupe de conteneurs qui partagent des ressources. Les pods sont placés dans les nœuds.
- **Conteneurs :**
 - Les conteneurs encapsulent les applications et leurs dépendances, assurant une exécution cohérente sur différentes machines.

8.3.4 Configuration du Réseau

Pour la communication entre les nœuds, j'ai configuré un switch pour assurer une connectivité stable et rapide.

- **Configuration du Switch :**
 - Mise en place du mode access pour les interfaces connectées aux Raspberry Pi.
 - Configuration du VLAN 9 pour séparer le trafic réseau du cluster.

Configuration du Switch

```
fralsswt0054#conf t
Enter configuration commands, one per line. End with CNTL/Z.
fralsswt0054(config)#int Fas
fralsswt0054(config)#int FastEthernet0/1
fralsswt0054(config-if)#swi
fralsswt0054(config-if)#switchport mode access
fralsswt0054(config-if)#switchport access vlan 9
fralsswt0054(config-if)#int FastEthernet0/2
fralsswt0054(config-if)#switchport mode access
fralsswt0054(config-if)#switchport access vlan 9
fralsswt0054(config-if)#int FastEthernet0/3
fralsswt0054(config-if)#switchport mode access
fralsswt0054(config-if)#switchport access vlan 9
fralsswt0054(config-if)#int FastEthernet0/4
fralsswt0054(config-if)#switchport mode access
fralsswt0054(config-if)#switchport access vlan 9
fralsswt0054(config-if)#int FastEthernet0/12
fralsswt0054(config-if)#switchport mode access
fralsswt0054(config-if)#switchport access vlan 9
fralsswt0054(config-if)#end
fralsswt0054#wr mem
Building configuration...
[OK]
```

Status des interfaces du Switch

Port	Name	Status	Vlan	Duplex	Speed	Type
Fa0/1	users	connected	9	a-full	a-100	10/100BaseTX
Fa0/2	users	connected	9	a-full	a-100	10/100BaseTX
Fa0/3	users	connected	9	a-full	a-100	10/100BaseTX
Fa0/4	users	notconnect	9	auto	auto	10/100BaseTX
Fa0/5	users	notconnect	11	auto	auto	10/100BaseTX
Fa0/6	users	notconnect	11	auto	auto	10/100BaseTX
Fa0/7	users	notconnect	11	auto	auto	10/100BaseTX
Fa0/8	users	notconnect	11	auto	auto	10/100BaseTX
Fa0/9	users	notconnect	1	auto	auto	10/100BaseTX
Fa0/10		notconnect	1	auto	auto	10/100BaseTX
Fa0/11		notconnect	1	auto	auto	10/100BaseTX
Fa0/12		notconnect	9	auto	auto	10/100BaseTX
Gio/1	UPLINK	notconnect	1	auto	auto	Not Present
Gio/2	UPLINK	notconnect	1	auto	auto	Not Present

Les 3 RPI 4 sont donc connecté sur le interfaces
Fa0/1 , Fa0/2 et Fa0/3

Figure 56 : Configuration du switch

8.3.3 Configuration des Raspberry Pi

Préparation du matériel :

J'ai commencé par configurer les Raspberry Pi individuellement. Chaque Raspberry Pi a été équipé d'une image Raspbian standardisée et configuré avec un accès SSH pour une gestion à distance. De plus, j'ai installé XRDP pour permettre des connexions de bureau à distance.

- **Configuration de base :**
 - Accès internet pour les mises à jour.
 - Configuration SSH.
 - Configuration XRDP pour le Raspberry Pi maître.

Clusterisation avec Docker Swarm :

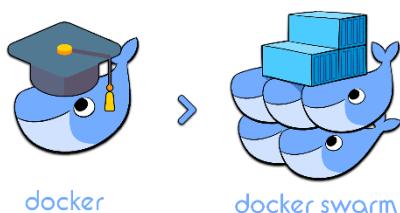


Figure 57 : Docker to Docker Swarm

Pour la gestion de clusters, j'ai utilisé Docker Swarm. Docker Swarm permet de gérer des clusters de conteneurs Docker de manière native.

8.3.3.1 Installation de Docker

Sur chaque Raspberry Pi, Docker Engine a été installé en utilisant le script d'installation officiel fourni par Docker.

- La commande `curl -fsSL https://get.docker.com -o get-docker.sh` a été utilisée pour télécharger le script d'installation.
- Ensuite, le script a été exécuté avec `sudo sh get-docker.sh` pour installer Docker Engine.

L'utilisateur a été ajouté au groupe docker pour exécuter des commandes Docker sans sudo, en utilisant `sudo usermod -aG docker $USER`.

Une déconnexion/reconnexion a été effectuée pour appliquer les modifications, et la version de Docker a été vérifiée avec `docker --version`.

8.3.3.2 : Initialisation du Swarm

Sur le Raspberry Pi principal (Forvia Main), Docker Swarm a été initialisé en tant que gestionnaire de Swarm avec `sudo docker swarm init --advertise-addr 10.0.0.1`.

Sur les deux Raspberry Pi de support (Forvia Performance 1 et Forvia Performance 2), la commande `docker swarm join` a été exécutée pour rejoindre le cluster en tant que nœuds de travail.

La liste des nœuds a été vérifiée avec `sudo docker node ls` sur le Raspberry Pi principal.

```

Installation et configuration de Docker swarm —————
Génération du token depuis le maître
root@forviamain:/home/fvmain/Cluster# docker --version
Docker version 26.0.2, build 3c863ff
root@forviamain:/home/fvmain/Cluster# [REDACTED]
version docker

Intégration des au maître en tant que worker
root@forviamain:/home/fvmain# sudo docker swarm init --advertise-addr 10.0.0.1
Swarm initialized: current node (s8uuy55h7n9v0xqakyhlhiz6k) is now a manager.

To add a worker to this swarm, run the following command:

  docker swarm join --token SWMTKN-1-3tvjmoamhdmrjp24030cczgpgk2m0exne9c866214jny49nqi-8q60uzlirfld9am0ghuozs8x1 1
  0.0.1:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

Status des noeuds du cluster
root@forviamain:/home/fvmain# docker node ls
ID          HOSTNAME   STATUS  AVAILABILITY  MANAGER STATUS    ENGINE VERSION
s8uuy55h7n9v0xqakyhlhiz6k *  forviamain  Ready   Active        Leader    26.0.2
synfcf9a7cyw842rs0hvst9   forviaperf1  Ready   Active        26.0.2
2t70zq05q2bvf2iybbpk7eg08 forviaperf2  Ready   Active        26.0.2
Status des noeuds du cluster

```

↓

Screenshot de la génération de token docker swarm

Figure 58 : Schéma de clusterisation

8.3.5 Redondance et Haute Disponibilité

L'un des principaux avantages de la configuration du cluster RPi est la redondance. En cas de défaillance d'un nœud, les autres nœuds peuvent continuer à fonctionner, assurant ainsi une haute disponibilité.

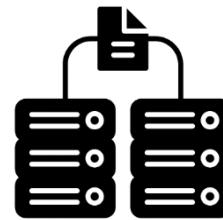


Figure 59 : Haute disponibilité

8.3.6 Haute Disponibilité avec NGINX

Pour ajouter une haute disponibilité au cluster, j'ai configuré un serveur NGINX hébergé sur le Raspberry Pi maître. En cas de défaillance du Raspberry Pi maître, le trafic sera redirigé vers un serveur redondant sur le Raspberry Pi perf1. Si le Raspberry Pi perf1 est également éteint, le trafic sera redirigé vers le serveur redondant sur le Raspberry Pi perf2.

Fichier docker-compose.yml : <https://pastebin.com/x2WCDNFB>

Ce fichier définit la configuration des services NGINX pour le cluster.

```

3   services:
4     nginx-master:
5       image: nginx:latest
6       ports:
7         - "80:80"
17    nginx-perf1:
18      image: nginx:latest
19      ports:
20        - "8081:80"
28    nginx-perf2:
29      image: nginx:latest
30      ports:
31        - "8082:80"

```

Figure 60 : Extrait de code du fichier docker-compose.yml

Nous définissons quatre services : nginx-master, nginx-perf1, nginx-perf2, et load_balancer. Chaque service utilise l'image NGINX et monte les fichiers de configuration et les fichiers HTML nécessaires. Le service nginx-master est contraint d'être déployé sur le nœud manager, tandis que les autres services peuvent être déployés sur n'importe quel nœud du cluster.

Fichier lb.conf : <https://pastebin.com/CWgKy8Va>

Le fichier lb.conf configure le load balancer NGINX pour distribuer le trafic entre les différents serveurs NGINX.

```

3   http {
4     upstream backend {
5       server nginx-master;
6       server nginx-perf1;
7       server nginx-perf2;
8     }

```

Figure 61 : extrait de code du fichier load balancer

Dans ce fichier, nous définissons un serveur HTTP qui écoute sur le port 80 et redirige le trafic vers un groupe de serveurs backend. Les serveurs backend sont nginx-master, nginx-perf1, et nginx-perf2. Le load balancer distribue le trafic entre ces serveurs, assurant ainsi une redondance et une haute disponibilité.

Fichier `default.conf` pour chaque NGINX (master, perf1, perf2) :

<https://pastebin.com/3NxfvAdF>

```

1 server {
2   listen 80;
3   server_name _;
4
5   location / {
6     root /usr/share/nginx/html;
7     index index.html;
8   }
9 }
```

Figure 62 : extrait de code du fichier `default.conf`

Le fichier `default.conf` configure chaque serveur NGINX pour servir les fichiers HTML. Ce fichier définit un serveur HTTP qui écoute sur le port 80 et sert le fichier `index.html` depuis le répertoire `/usr/share/nginx/html`.

Fichier `index.html` : <https://pastebin.com/pVBTBdUs>

```

6 <title>Cluster Forvia</title>
7 </head>
8 <body>
9 <h1>Redondance Cluster Forvia</h1>
```

Figure 63 : extrait de code du fichier `index.html`

Le fichier `index.html` contient la page à afficher par les serveurs NGINX. Ce code affiche un simple message "Redondance Cluster Forvia".

8.3.7 Prochaine Étape



Pour améliorer encore davantage le cluster, la prochaine étape consiste à mettre en place des mécanismes de surveillance et d'alertes pour détecter et réagir rapidement aux problèmes. De plus, l'optimisation des ressources et l'amélioration des performances doivent être des objectifs continus. Bien que les tests n'aient pas pu être inclus dans ce rapport, il est important de noter qu'en éteignant un Raspberry Pi (pour simuler une panne), la page web reste accessible, ce qui démontre l'efficacité de la redondance.

8.3.8 Conclusion du Projet Cluster RPi

Le projet Cluster Raspberry Pi a permis de créer une infrastructure de calcul distribuée et redondante, capable de supporter diverses charges de travail tout en garantissant une haute disponibilité. L'ajout de la haute disponibilité avec NGINX renforce la fiabilité de l'infrastructure. Les compétences acquises lors de ce projet, telles que la configuration de Docker Swarm, la gestion des réseaux et la mise en place de solutions de haute disponibilité, sont précieuses pour l'amélioration continue des infrastructures IT chez Forvia. Les étapes futures incluent l'optimisation des performances et la mise en place d'outils de monitoring pour assurer une gestion proactive du cluster.

8.4 Analyse et recommandations pour l'économie d'énergie

Le projet d'économie d'énergie visait à réduire la consommation énergétique de l'usine Forvia Allenjoie tout en sensibilisant les utilisateurs et en mettant en place des systèmes de gestion de l'énergie.

8.4.1 Introduction au Projet d'Économie d'Énergie

Dans le cadre de mon stage, j'ai étudié diverses solutions pour améliorer l'efficacité énergétique de l'usine. Le projet visait à identifier les sources principales de consommation d'énergie et à proposer des mesures pour les réduire, tout en sensibilisant les utilisateurs à l'importance de la gestion de l'énergie.

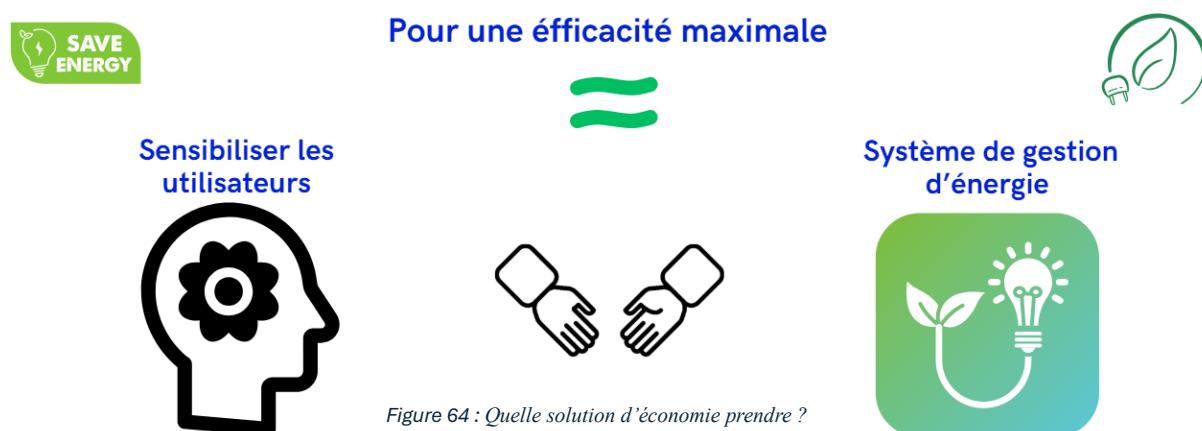


Figure 64 : Quelle solution d'économie prendre ?

8.4.2 Sensibilisation des Utilisateurs

Une politique de gestion de l'énergie des écrans pourrait être mise en place pour responsabiliser les superviseurs de chaque secteur.

- Responsabilisation des superviseurs pour le contrôle visuel de l'utilisation des écrans.

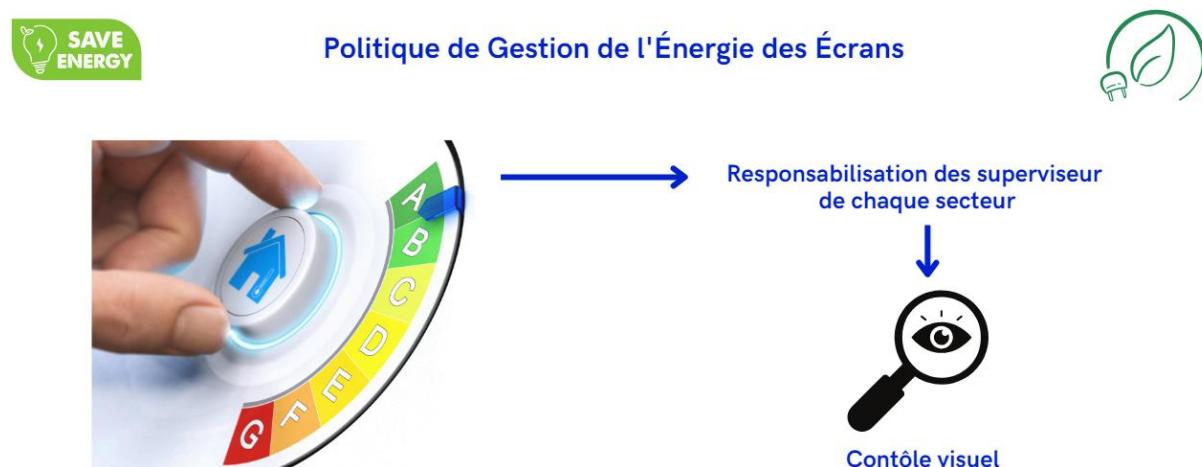


Figure 65 : Sensibiliser les utilisateurs

8.4.3 Système de Gestion d'Énergie

Pour une gestion efficace de l'énergie, deux solutions principales ont été envisagées :

- **Configurer manuellement chaque écran** : Cette solution implique de régler manuellement les paramètres de chaque écran pour optimiser leur consommation énergétique.
- **Prises programmables** : L'utilisation de prises programmables permet de contrôler automatiquement l'alimentation des écrans, réduisant ainsi la consommation d'énergie.

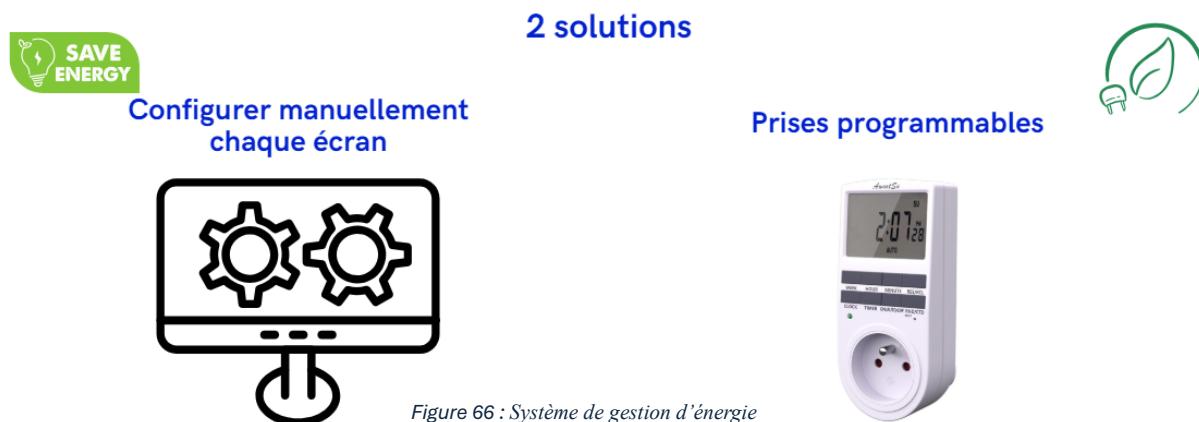


Figure 66 : Système de gestion d'énergie

8.4.4 Prises Programmables Digitales

La solution choisie a été l'utilisation de prises programmables digitales. Ces prises permettent une programmation flexible de l'alimentation des écrans sans nécessiter de contrôle à distance.

Caractéristiques des Prises Programmables Digitales :

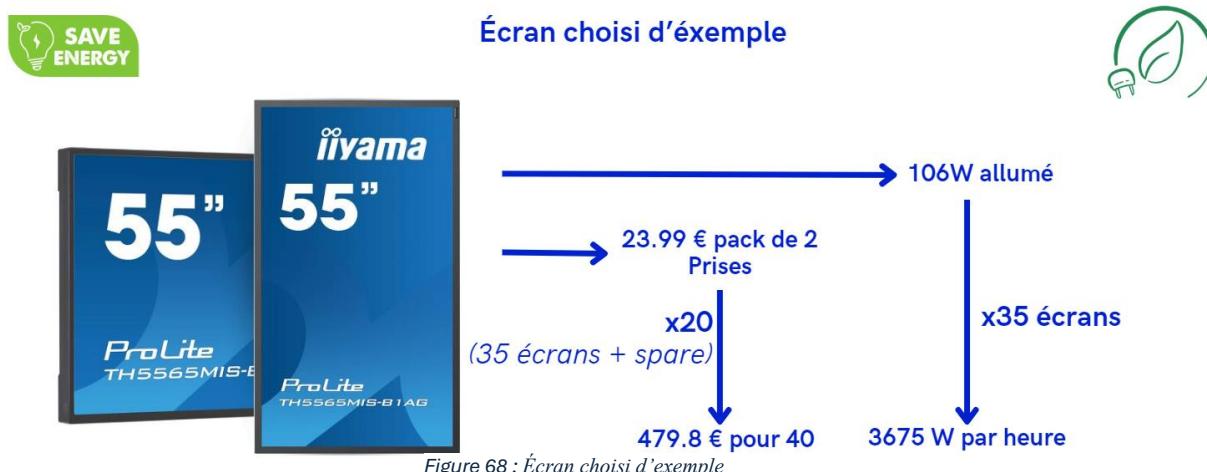


Figure 67 : Prise Programmable Digitale

8.4.5 Sensibilisation des Utilisateurs

Pour illustrer l'impact de cette solution, un écran d'exemple a été choisi :

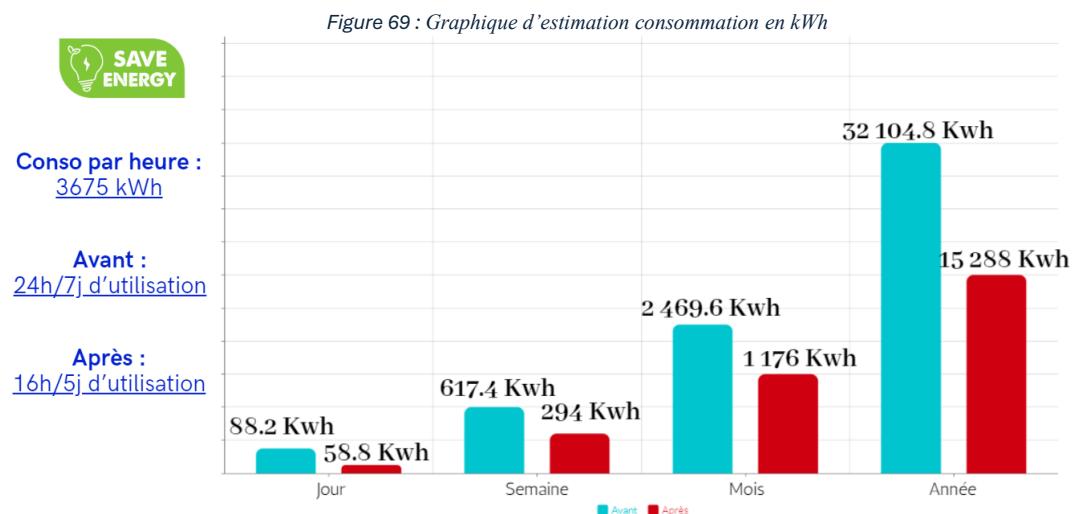
- **Écran ProLite 55"** :
 - Consommation : 106W allumé.
 - Nombre d'écrans : 35.
 - Consommation totale par heure : 3675W.
 - Coût des prises pour 40 écrans : 479,8 € (incluant des prises de recharge).



8.4.6 Impact des Mesures sur la Consommation et les Coûts

En passant d'une utilisation 24h/7j à une utilisation 16h/5j, (soit les 2 tournées de travail de 8h) une réduction significative de la consommation d'énergie et des coûts a été observée.

- **Consommation Avant et Après :**
 - Avant : 24h/7j d'utilisation > Soit du lundi matin au dimanche soir
 - Après : 16h/5j d'utilisation > Soit du lundi au vendredi, à partir de 5h jusqu'à 21h.



- **Économies Réalisées :**
 - Tarif kWh : 0,2516 €.
 - Baisse de consommation : 52,35%.
 - Économie annuelle estimée : 4 231 €.

Figure 70 : Graphique d'estimation des coûts



8.4.7 Conclusion du Projet d'Économie d'Énergie

Le projet d'économie d'énergie a permis de mettre en place des mesures efficaces pour réduire la consommation énergétique de l'usine Forvia Allenjoie. Grâce à la sensibilisation des utilisateurs et à l'utilisation de prises programmables, une réduction significative de la consommation d'énergie et des coûts a été réalisée. Les compétences acquises lors de ce projet, telles que la gestion de l'énergie et la mise en place de solutions technologiques, sont précieuses pour l'amélioration continue des infrastructures IT chez Forvia. Les étapes futures incluent l'optimisation continue des paramètres énergétiques et l'intégration de nouvelles technologies pour une gestion proactive de l'énergie.

9 Missions Réalisées

9.1 Gestion des Imprimantes

9.1.1 Configuration

La configuration des imprimantes était une tâche essentielle pour assurer leur bon fonctionnement au sein de l'usine. Cela comprenait l'installation de nouvelles imprimantes, la configuration des paramètres réseau pour garantir une connectivité optimale, ainsi que la préparation des imprimantes de réserve pour une redondance toujours optimale.

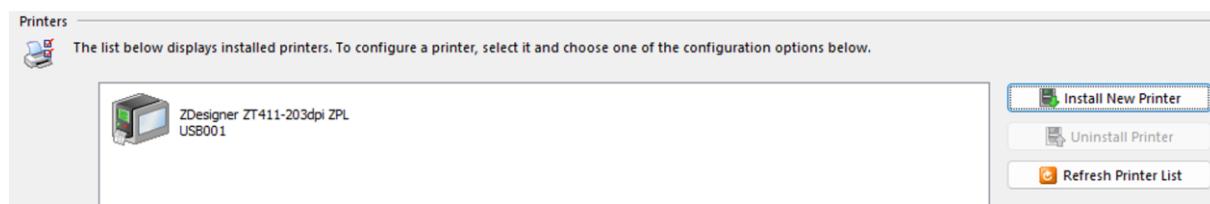


Figure 71 : Configuration d'imprimante Zebra

9.1.2 Maintenance

La maintenance des imprimantes incluait des vérifications régulières, la résolution des problèmes techniques courants tels que les bourrages papier et les erreurs de communication.

9.1.3 Cartographie des Imprimantes au sein de l'Usine

J'ai réalisé une cartographie détaillée des imprimantes dans l'usine pour mieux gérer leur répartition et leur utilisation. Cette cartographie a aidé à identifier les zones avec une utilisation élevée et à planifier la maintenance de manière plus efficace.

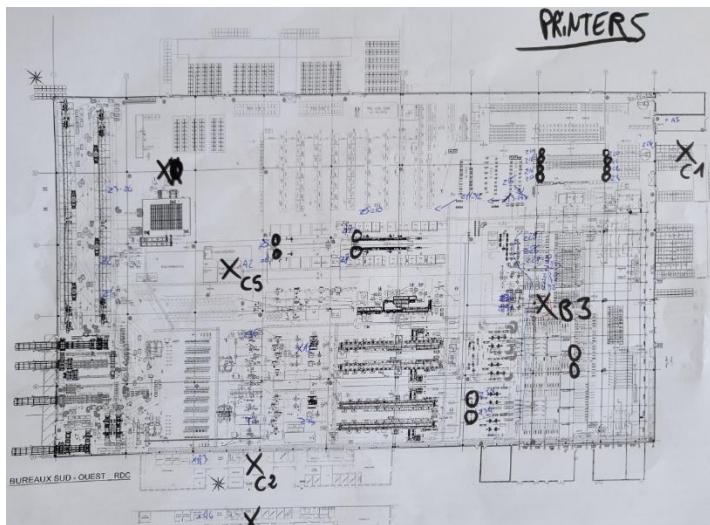


Figure 72 : Cartographie des imprimantes

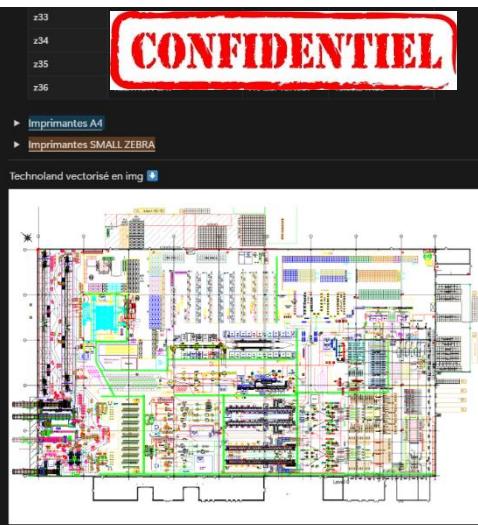


Figure 73 : Note des @IP et Carto détaillé

Chaque point sur la cartographie correspond à une entrée dans le tableau ci-dessus, où sont indiqués le nom de l'imprimante, son secteur et son adresse IP. Les imprimantes Zebra sont identifiées par un code commençant par "Zxx", les imprimantes A4 par "Cxx", et les petites imprimantes annexes par "Bxx".

9.1.4 Enquête sur les soucis d'étiquettes

J'ai dirigé une enquête visant à résoudre les problèmes d'étiquetage rencontrés avec certaines imprimantes. Pour ce faire, j'ai contacté les fournisseurs d'étiquettes afin de vérifier les caractéristiques des commandes récentes et passées, dans le but de comprendre les différences et les ajustements nécessaires. J'ai échangé avec le fournisseur par téléphone et, après réception d'un nouveau réapprovisionnement, nous avons constaté une nette réduction des problèmes liés aux imprimantes.

Pour approfondir le sujet, j'ai également contacté le support technique de Zebra afin d'obtenir des informations détaillées sur le fonctionnement des capteurs d'étiquettes.

Zebra Enterprise

Product Category	Printer	Entitlement Status	ENTITLED
Product Serial Number	99J222606169	Product Model	ZT411
Subject	FORVIA FAS Allenjoie FRANCE - Assistance Needed with Zebra ZT411 Printer Label Sensor Issue		
Description	I am in charge of the IT support for the Forvia Allenjoie FAS site near Montbéliard in France. I am contacting out to request information regarding the Zebra ZT411 printers used on our site. Recently, we have encountered issues with the labels. The printers indicate a paper shortage even though the paper is present. I am currently investigating this issue, and we are unsure whether it is a configuration problem on our end or an issue with the thermal labels. Could you please provide some insight into the operation of the label presence sensor? Could this issue be caused by a misconfiguration? Thank you in advance for your assistance. I look forward to your response.		

Figure 74 : Contact Zebra

Après plusieurs échanges, j'ai reçu les renseignements demandés, ce qui m'a permis de mieux comprendre le fonctionnement technique des capteurs. J'ai ensuite présenté ces informations à mon manager.

9.2 Gestion des PC

9.2.1 Remasterisation en PXE avec les Standards Forvia

Qu'est-ce que la remasterisation ?

La remasterisation d'un PC consiste à réinstaller le système d'exploitation et tous les logiciels nécessaires pour que la machine soit conforme aux standards de l'entreprise. Ce processus permet de restaurer les performances optimales du PC, de garantir la sécurité des données et de s'assurer que toutes les configurations sont à jour.

Chez Forvia, la remasterisation des PC se fait via le réseau en utilisant le protocole PXE (Preboot Execution Environment). Ce processus démarre depuis le BIOS du PC en choisissant l'option de démarrage réseau (PXE boot). Le PC se connecte alors au serveur de déploiement PXE de Forvia où une liste d'images système est disponible. Les utilisateurs peuvent sélectionner l'image Windows préconfigurée selon les standards Forvia, incluant toutes les configurations spécifiques à l'entreprise, les logiciels nécessaires et les paramètres de sécurité. Une fois l'image sélectionnée, elle est téléchargée et installée sur le PC via le réseau, suivi de scripts de post-installation configurant les paramètres spécifiques à Forvia. Ce processus assure que tous les PC ont une configuration uniforme, réduisant ainsi le temps nécessaire pour configurer un nouveau PC ou réinstaller un système, et garantissant que tous les correctifs de sécurité et les configurations sont appliqués.

9.2.2 Mise en Place d'AutoLogon

La mise en place d'AutoLogon permet aux utilisateurs de se connecter automatiquement à leurs sessions Windows sans avoir à saisir leurs informations d'identification à chaque démarrage. Cela est particulièrement utile dans des environnements où un accès rapide est crucial, comme les postes de travail partagés ou les systèmes kiosques.

Pour mettre en place l'AutoLogon, il est nécessaire de modifier certaines clés de registre dans Windows. Tout d'abord, j'ai ouvert l'éditeur de registre (regedit) et naviguer vers la clé suivante : HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon. Dans cette clé, les valeurs suivantes doivent être créées ou modifiées : DefaultUserName pour spécifier le nom d'utilisateur de la session, DefaultPassword pour entrer le mot de passe de l'utilisateur (en tenant compte des risques de sécurité liés au stockage des mots de passe en texte clair), DefaultDomainName pour indiquer le domaine de l'utilisateur si nécessaire, et AutoAdminLogon dont la valeur doit être réglée à 1 pour activer l'AutoLogon. Après avoir configuré ces clés, il est important de redémarrer le PC pour appliquer les modifications.

L'AutoLogon présente plusieurs avantages notables. Il élimine la nécessité de saisir les informations d'identification à chaque démarrage, ce qui est un gain de temps précieux. Il améliore l'accessibilité pour les utilisateurs dans des environnements où un accès rapide est essentiel, et il réduit les interruptions, permettant aux utilisateurs de commencer à travailler plus rapidement. Cette fonctionnalité est particulièrement bénéfique pour les postes de travail partagés, où l'efficacité et la rapidité d'accès sont cruciales.

9.3 Script de Comparaison de Fichiers

Dans le cadre de mon stage, j'ai été confronté à un problème urgent lié au référencement de commandes et de fichiers. Pour y remédier, j'ai développé une série de scripts PowerShell destinés à comparer les données de VIN (Vehicle Identification Number) entre un fichier Excel et des fichiers XML, permettant ainsi d'identifier les VIN présents dans les fichiers XML mais absents du fichier Excel. Voici un aperçu du fonctionnement et des scripts développés.

Fonctionnement du Script de Comparaison :

Le script principal, script_delta.ps1, parcourt les fichiers XML dans un dossier spécifié. Pour chaque fichier XML, il extrait le VIN et vérifie s'il est présent dans le fichier Excel. Les VIN absents sont alors renommés avec l'extension .miss.

```
script delta.ps1
1 $fichierCSV = "C:\Users\elhamiou\Documents\Data_overview_Forvia_cleaned.csv" #Modifier avec le chemin du bon fichier .csv
2
3 $dossierXML = "C:\Users\elhamiou\Documents\xml" #Il faut modifier par le chemin du dossier contenant les fichiers
4 $listeVIN = @(Get-Content $fichierCSV | Select-Object -Skip 1)
5
6 foreach ($xmlFilePath in Get-ChildItem $dossierXML -Filter *.xml) {
7     $ContentXML = [xml](Get-Content $xmlFilePath.FullName)
8     $idVIN = $ContentXML.SelectSingleNode("//VIN")
9     if ($idVIN -ne $null) {
10         $vin = $idVIN.InnerText
11         if ($vin -notin $listeVIN) {
12             $nvNomFichier = $xmlFilePath.Name -replace '\.xml$', '.miss' #Ici on peut changer .miss par le rename souhaité, par exemple .old ou autres...
13             $newFilePath = Join-Path -Path $dossierXML -ChildPath $nvNomFichier
14             Rename-Item -Path $xmlFilePath.FullName -NewName $newFilePath
15             Write-Output "Le fichier $($xmlFilePath.Name) a été renommé en $nvNomFichier." #On peut suppr cette ligne si on veut éviter l'echo sur le terminal
16         }
17     } else {
18         Write-Output "Pas de vin trouvé $($xmlFilePath.Name)." #On peut suppr cette ligne si on veut éviter l'echo sur le terminal
19     }
20 }
```

Figure 75 : Script de comparaison

Fonctionnement du Script de Suppression :

Le script suppr_non_miss.ps1 supprime tous les fichiers XML qui n'ont pas été renommés avec l'extension .miss, ne conservant ainsi que les fichiers pertinents pour l'analyse ultérieure.

Script suppr_non_miss.ps1

```
$dossierXML = "C:\Users\elhamiou\Documents\xml" #Il faut modifier le chemin par le chemin du dossier contenant les fichiers
2
3 foreach ($file in Get-ChildItem $dossierXML) {
4     if ($file.Extension -ne ".miss") { #Si besoin, changer ".miss" par le rename souhaité, par exemple .old ou autres...
5         Remove-Item -Path $file.FullName -Force
6         Write-Output "Le fichier $($file.Name) a été supprimé." #On peut suppr cette ligne si on veut éviter l'echo sur le terminal
7     }
8 }
```

Figure 76 : Script de suppression

Fonctionnement du Script de Restauration :

Le script rename_origin.ps1 supprime l'extension .miss des fichiers XML renommés par le premier script, ramenant ainsi les noms de fichiers à leur état d'origine.

Script rename_origin.ps1

Figure 77 : Script de Restauration

```
$dossierXML = "C:\Users\elhamiou\Documents\xml" #Il faut modifier le chemin par le chemin du dossier contenant les fichiers
2
3 foreach ($file in Get-ChildItem $dossierXML -Filter "*.miss") {
4     $nvNomFichier = $file.Name -replace '\.miss$', ''
5
6     $newFilePath = Join-Path -Path $file.DirectoryName -ChildPath $nvNomFichier
7
8     Rename-Item -Path $file.FullName -NewName $newFilePath -Force
9     Write-Output "Le fichier $($file.Name) a été renommé en $nvNomFichier."
10 }
```

Pour finir, j'ai fait un fichier readme.txt, explicitant comment utiliser le script.

```
Instructions d'utilisation

Avant de commencer, quelques manip à faire :
1. Trier le fichier excel > supprimer les colonnes et les lignes inutiles
2. Convertir le fichier excel en .csv
3. Se rendre dans le script "script_delta" et modifier la variable "$fichierCSV" > remplacer "C:\Users\elhamiou\Documents\Data_overview_Forvia_cleaned.csv" par le chemin réel de votre fichier csv
4. Se rendre dans chaque script et modifier la variable "$dossierXML" et remplacer "C:\Users\elhamiou\Documents\xml" par le chemin réel de votre dossier contenant les fichiers xml

// EXECUTION DU SCRIPT \\

Détecter et Renommer les fichiers Delta
> Lancer PowerShell > Executer le script : /cheminduscript/script_delta.ps1
Ce script détectera les fichiers delta dans le répertoire spécifié et les renommera.

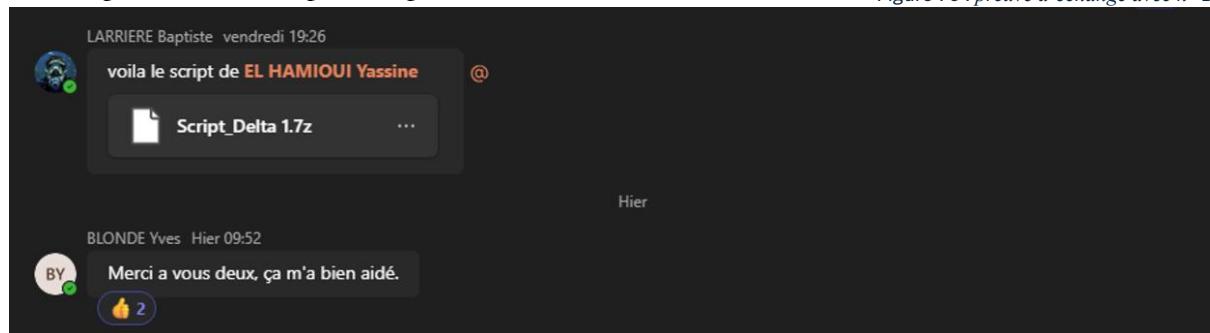
Pour conserver uniquement les fichiers delta :
> Lancer PowerShell > Executer le script : /cheminduscript/suppr_non_miss.ps1
Ce script supprimera tous les fichiers qui ne sont pas des fichiers delta, laissant seulement ceux qui sont pertinents.

Enfin, pour retirer l'extension ".miss" des fichiers d'origine, suivez ces étapes :
> Lancer PowerShell > Executer le script : /cheminduscript/ rename_origin.ps1
Ce script renommera tous les fichiers d'origine en retirant l'extension ".miss".
```

Figure 78 : Fichier readme.txt

Retour positif sur le script de la part du demandeur :

Figure 79 : preuve d'échange avec n+2



9.4 Configuration des camera Raspberry PI EOL (End Of Line)

Pour des raisons de confidentialité, je ne vais pas partager les détails spécifiques ou les logiciels internes utilisés pour configurer les caméras Raspberry Pi EOL chez Forvia. Cependant, je vais fournir une vue d'ensemble du processus et de ma contribution.

Les caméras EOL sont utilisées pour capturer des photos de fin de ligne, générant des preuves de qualité et assurant la traçabilité des produits pour les clients. Ces caméras montées sur Raspberry Pi sont essentielles pour la documentation de la qualité des produits fabriqués.

J'ai configuré une dizaine de caméras EOL, actuellement stockées et prêtes à être utilisées en SPARE. En cas de défaillance d'une caméra en service, elles peuvent être rapidement remplacées.

Processus de Configuration :

1. **Image Raspbian Standardisée :**
 - o Utilisation de l'image "Faur_raspberry_buster_V2.0.1", configurée avec les services nécessaires pour les caméras et leur intégration dans l'usine.
2. **Utilisation du Logiciel de Configuration :**
 - o Un logiciel interne développé par Fernando Pinheiro est utilisé pour configurer le nom de la machine, attribuer l'adresse IP et visualiser le flux de la caméra en temps réel.
3. **Étapes de Configuration :**
 - o Boot de l'image Raspbian sur le Raspberry Pi.
 - o Utilisation du logiciel pour configurer l'adresse IP et le nom de la caméra.
 - o Identification de l'adresse IP actuelle de la caméra à l'aide de Wireshark pour les caméras déjà configurées, facilitant la connexion.

10. Conclusion et Perspectives

Mon stage chez Forvia a été une expérience extrêmement enrichissante et formatrice. J'ai pu travailler sur une variété de projets stimulants qui m'ont permis de développer des compétences techniques et professionnelles précieuses. L'environnement de travail collaboratif et l'accès à des technologies avancées ont grandement contribué à mon apprentissage. Les défis rencontrés m'ont permis de développer des solutions innovantes et d'améliorer mes compétences en résolution de problèmes.

Un aspect particulièrement intéressant de mon stage a été l'application des principes du Juste-à-Temps (JIT) dans l'environnement de production de Forvia. Le JIT est une stratégie de gestion qui vise à améliorer l'efficacité et à réduire les gaspillages en produisant uniquement ce qui est nécessaire, quand cela est nécessaire, et en quantités exactes. Travailler dans un tel environnement m'a permis de comprendre l'importance de la réactivité et de la précision dans les opérations IT.

Chez Forvia, le JIT est essentiel pour maintenir un flux de production fluide et sans interruption. Par exemple, les projets tels que la configuration des caméras EOL pour la traçabilité et les systèmes de monitoring avec CheckMK et Grafana sont cruciaux pour garantir que les processus de production sont surveillés en temps réel, permettant une intervention rapide en cas de problème. Cela réduit les temps d'arrêt et assure une livraison ponctuelle des produits.

La nécessité de configurations rapides et efficaces, comme celle des Raspberry Pi pour les caméras EOL, reflète également les principes du JIT. En ayant des équipements prêts à être utilisés en cas de panne, l'entreprise peut minimiser les interruptions et maintenir la continuité des opérations. De plus, la standardisation des processus, comme la remasterisation des PC via PXE, permet de maintenir une homogénéité et une fiabilité dans les systèmes IT, essentielles pour un environnement de production JIT.

Cette expérience a solidifié mon intérêt pour le domaine de l'IT et m'a préparé pour ma future carrière professionnelle. En travaillant dans un environnement JIT, j'ai appris à être plus réactif, à gérer les priorités efficacement et à développer des solutions qui soutiennent les objectifs de production de l'entreprise. Les compétences et les connaissances acquises pendant ce stage me seront inestimables dans ma carrière future, en particulier dans des secteurs où l'efficacité et la gestion des ressources sont critiques.

11 Annexes

11.1 Captures d'écran et illustrations des projets

Figure 1 : schéma système JIT chez Forvia

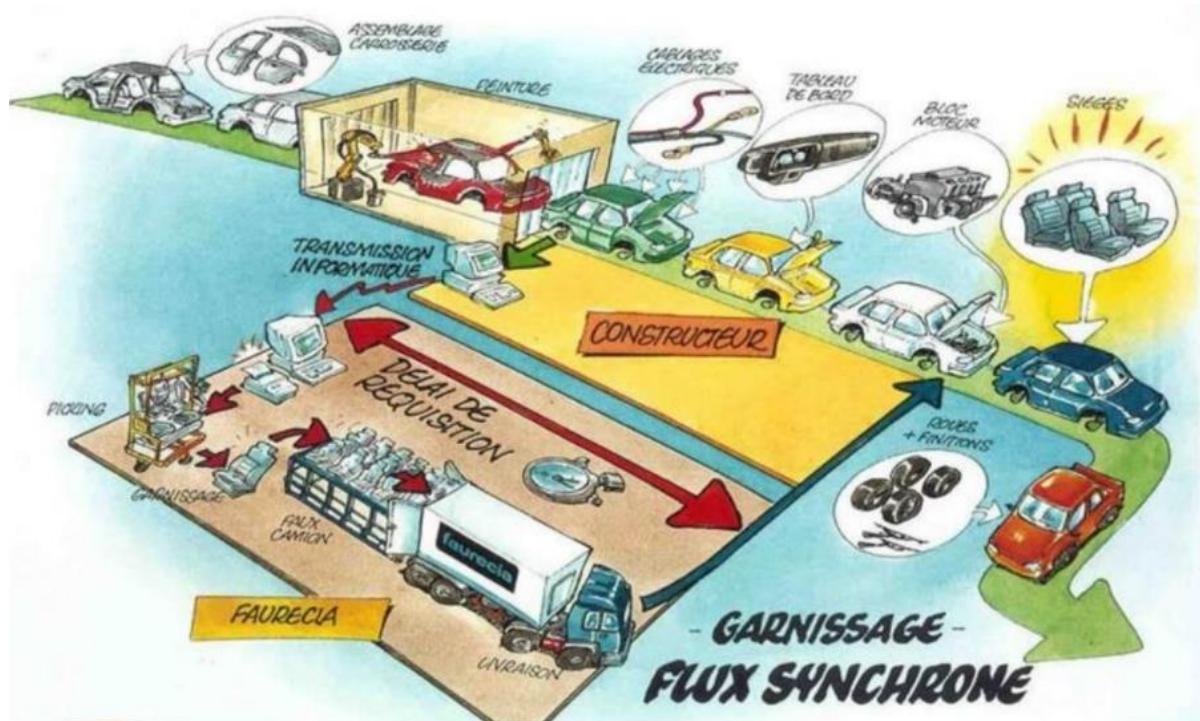


Figure 2: Organigramme Equipe IT

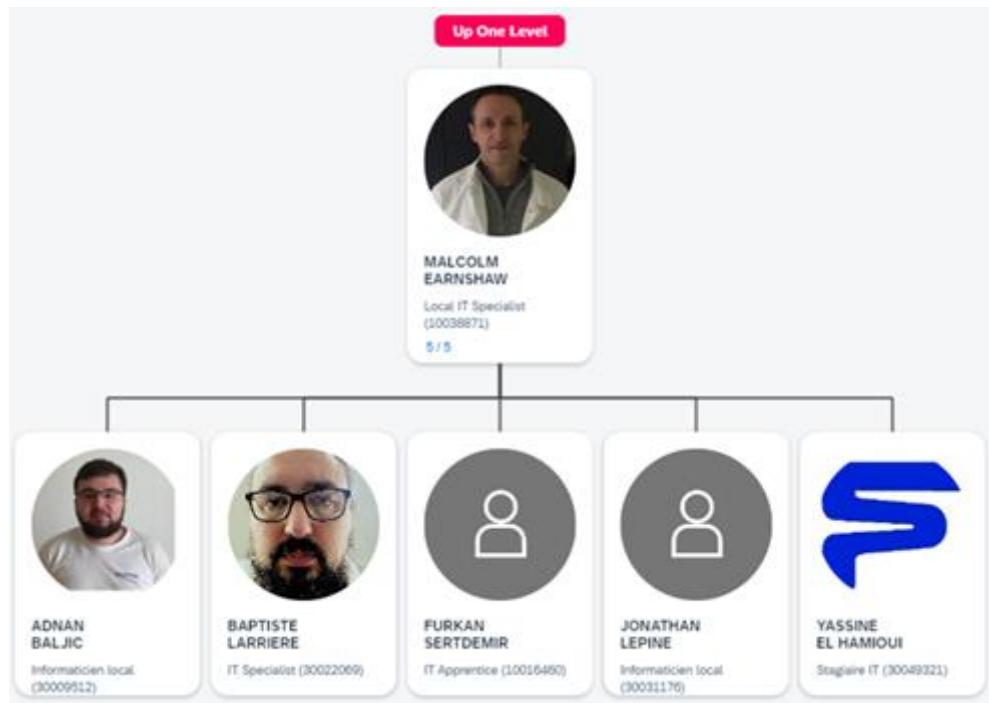


Figure 3 : Logo Notion



Figure 4 : Logo Github



Figure 5 : Logo Visual Studio Code



Figure 6 : Logo Word



Figure 7 : Logo Excel



Yassine EL Hamiou
RAPPORT DE STAGE FORVIA

Figure 8 : Page d'accueil Notion de mon stage chez Forvia

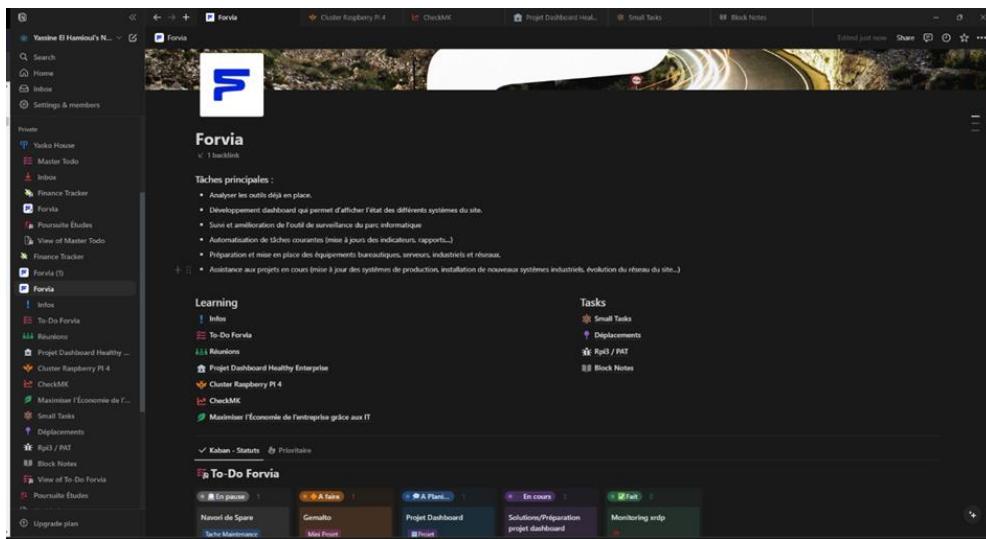


Figure 9 : Suivis d'activité de Stage chez Forvia

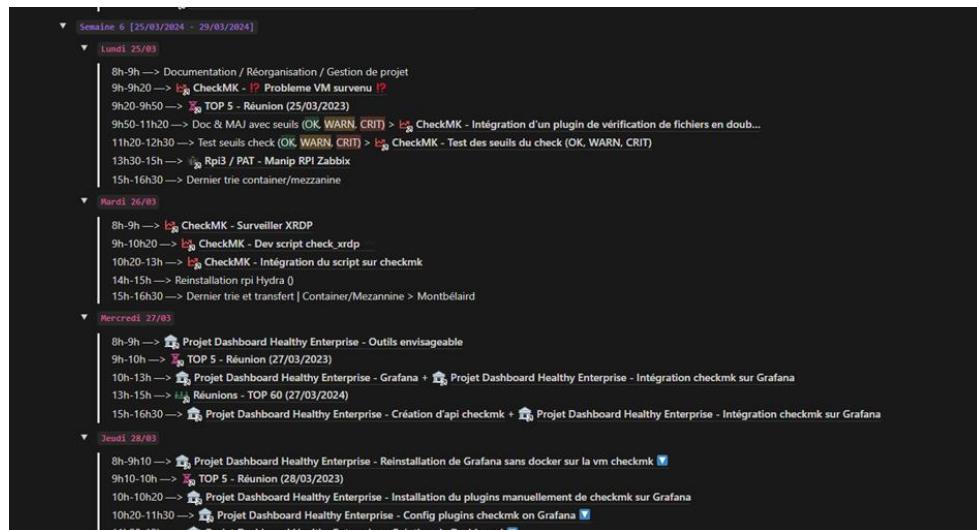


Figure 10 : GitHub de Yassine Stage Forvia

Yassine EL Hamiou
RAPPORT DE STAGE FORVIA

Figure 11 : Screenshot extrait de réunion TOP 5



Figure 12 : Schéma d'installation et de configuration de Hydra

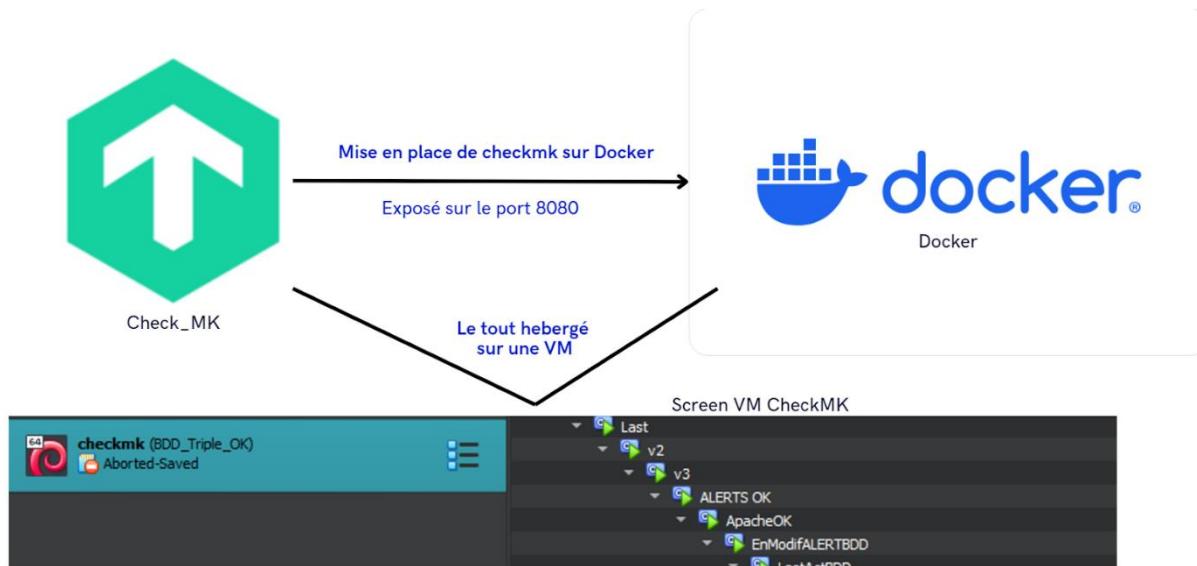


Figure 13 : Screen test de PING/AGENT/SNMP

Figure 14 : Boucle pour surcharger le CPU

```

1 #!/bin/bash
2 Load_cpu() {
3     while true; do
4         true
5         echo "Le CPU pleure"
6     done
7 }
8 load_cpu

```

Figure 15 : Test des check avec surcharge

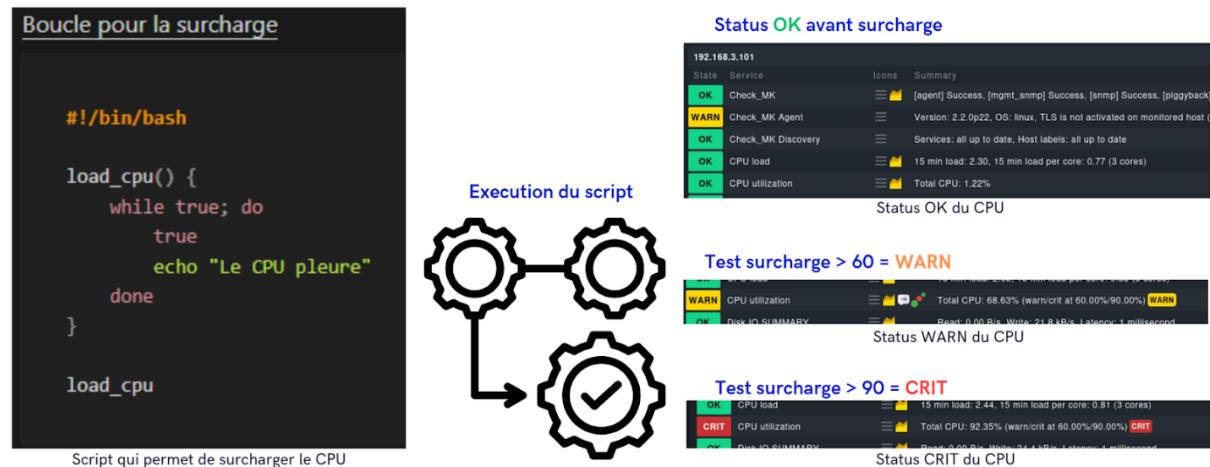


Figure 16 : Screen règle Check HTTP service



Figure 17 : Screen Expect the content

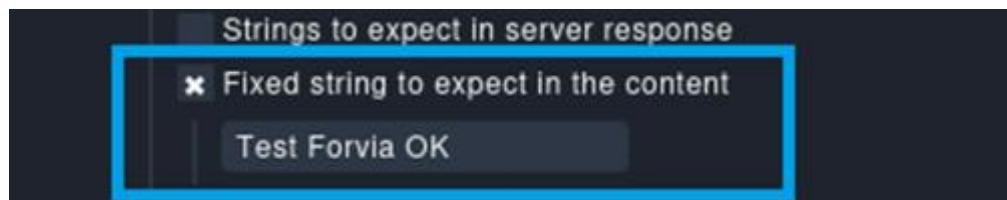


Figure 18 : Screen config de la règle



Figure 19 : Test du check sans le text “Test Forvia OK”



Figure 20 : Test du check avec le text “Test Forvia OK”



Figure 21 : Création de la Base de données

```
root@raspberry:/var/www/html# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 30
Server version: 10.5.23-MariaDB-0+deb11u1 Debian 11

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE suivis_prod_kits;
Query OK, 1 row affected (0,003 sec)

MariaDB [(none)]> CREATE USER 'forvia'@'localhost' IDENTIFIED BY 'toto';
Query OK, 0 rows affected (0,017 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON suivis_prod_kits.* TO 'forvia'@'localhost';
Query OK, 0 rows affected (0,002 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,003 sec)
```

Figure 22 : Simulation de donnée

```
MariaDB [suivis_prod_kits]> SELECT * FROM production_tracking;
+----+-----+-----+-----+
| id | timestamp           | site_name | status |
+----+-----+-----+-----+
| 1  | 2024-02-29 14:24:17 | Siège A   | 0      |
| 2  | 2024-02-29 14:24:17 | Siège A   | 1      |
| 3  | 2024-02-29 14:24:17 | Siège B   | 0      |
| 4  | 2024-02-29 14:24:17 | Siège B   | 1      |
| 5  | 2024-02-29 14:24:17 | Siège A   | 1      |
```

Figure 23 : Requête SQL pour la supervision de la base de données

```

1  SELECT
2    CASE
3      WHEN COUNT(status) = 0 THEN 0
4      WHEN COUNT(status) <= 3 THEN 1
5      ELSE 2
6    END AS alert_level
7  FROM
8    production_tracking
9 WHERE
10   status = 0;

```

Figure 24 : Analyse de la requête SQL & Seuils d'alertes



Figure 25 : Test du check status 0 Hydra (soit OK)

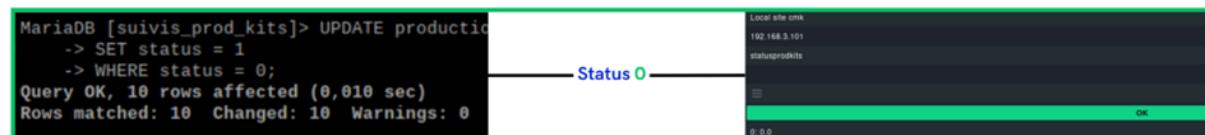


Figure 26 : Test du check status 1 Hydra (soit WARN)

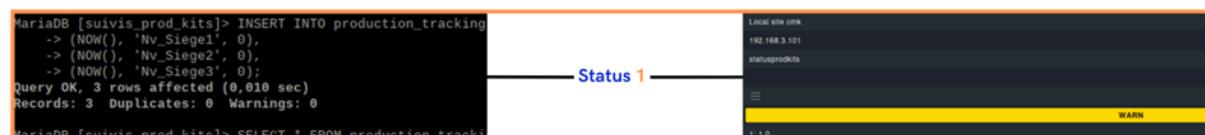


Figure 27 : Test du check status 2 Hydra (soit CRIT)

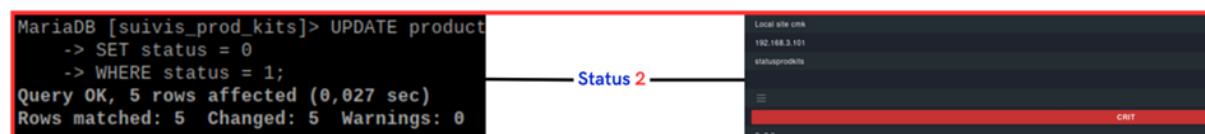


Figure 28 : schéma du script de détection de fichiers double



Figure 29 : file_duplicate.py pt.1 & pt.2

Première partie	Deuxième partie
<pre> 1 import os 2 3 def find_duplicate_files(folder_path): 4 file_checksums = {} 5 6 for root, dirs, files in os.walk(folder_path): 7 for file in files: 8 file_path = os.path.join(root, file) 9 with open(file_path, 'rb') as f: 10 file_content = f.read() 11 checksum = hash(file_content) 12 file_checksums[file_path] = checksum 13 14 duplicate_files = {} 15 for file_path, checksum in file_checksums.items(): 16 if list(file_checksums.values()).count(checksum) > 1: 17 if checksum not in duplicate_files: 18 duplicate_files[checksum] = [] 19 duplicate_files[checksum].append(file_path) 20 21 return duplicate_files </pre>	<pre> 22 23 folder_path = "/home/forvia/double" 24 duplicates = find_duplicate_files(folder_path) 25 26 num_duplicates = sum(len(files) for files in duplicates.values()) 27 if duplicates: 28 for checksum, files in duplicates.items(): 29 print(f"- Doublon (checksum : {checksum}) :") 30 for file_path in files: 31 print(f" - {file_path}") 32 if num_duplicates >= 5: 33 print("Trop de doublons détectés.") 34 exit(2) # Statut CRITICAL 35 else: 36 print("Quelques doublons détectés.") 37 exit(1) # Statut WARNING 38 else: 39 print("Aucun doublon trouvé dans le dossier.") 40 exit(0) # Statut OK </pre>

Figure 30 : Test Script en local & config hydra



Figure 31 : Check Hydra pas de fichier en double

```
root@raspberry:/home/forvia# rm double/*
root@raspberry:/home/forvia# ls double
root@raspberry:/home/forvia# cat double/*
cat: 'double/*': Aucun fichier ou dossier de ce type
root@raspberry:/home/forvia#
```

Figure 32 : Check Hydra entre 1 & 4 doublons

```
root@raspberry:/home/forvia# ls double
fichier1.txt fichier2.txt fichier3.txt
root@raspberry:/home/forvia# cat double/*
Toto
Toto
Toto
Toto
root@raspberry:/home/forvia#
```

Figure 33 : Check Hydra 5 ou plus de fichiers en double

```
root@raspberry:/home/forvia# ls double
fichier1.txt fichier2.txt fichier3.txt fichier4.txt fichier5.txt
root@raspberry:/home/forvia# cat double/*
Toto
Toto
Toto
Toto
Toto
root@raspberry:/home/forvia#
```

Figure 34 : Hydra test connexion host UPS

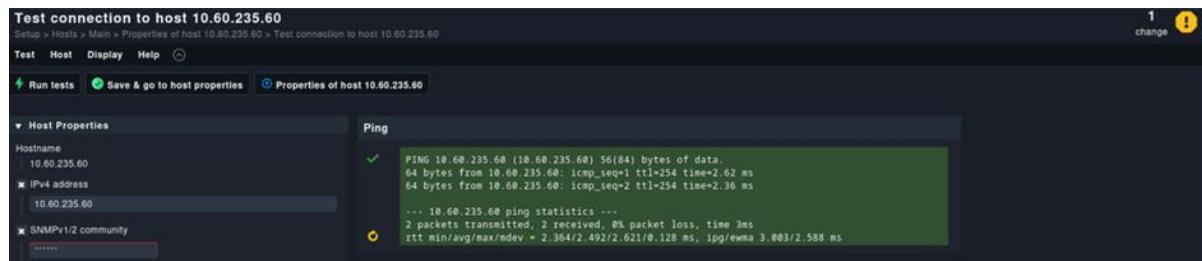


Figure 35 : Configuration SNMP Hydra et UPS

Conf SNMPv3 depuis WEB UPS

Schneider Electric UPS Network Management Card 3 Smart-UPS Application

Home Status Control Configuration Tests Logs

Configure SNMPv3 Access Control

Access Control

Enable

User Name: adminforvia

NMS IP/Host Name: 10.60.235.51

Apply Cancel

Knowledge Base | Schneider Electric Product Center | Schneider Electric Downloads

Intégration UPS sur Hydra

Monitoring agents

- Checkmk agent / API integrations: No API integrations, no Checkmk agent
- SNMP: SNMP v2 or v3
- SNMP credentials: Credentials for SNMPv3 without authentication and privacy (noAuthN...)

Security Level: No authentication, no privacy

Security name:

10.60.235.60

Check_MK Discovery

WARN

Unmonitored services: 8 services: 1. apc_symmetra_temp, 1. apc_symmetra_output, 1. apc_symmetra_input, 1. interfaces, 1. apc_symmetra_load, 1. apc_invre_system_events, 1. apc_invre_system

1. apc_invre_system

Unmonitored service: update: Update

Unmonitored service: apc_symmetra_temp: Temperature Battery

Unmonitored service: apc_symmetra_dcinput: Phase Output

Unmonitored service: apc_symmetra_dcoutput: Primary/Output

Unmonitored service: http: Info: SNMP info

Unmonitored service: apc_symmetra_input: Phase Input

Unmonitored service: apc_symmetra_load: Load Test

Unmonitored service: apc_symmetra_test: Self Test

Unmonitored service: apc_invre_system: Events, System events

Figure 36 : Intégration de l'UPS sans erreurs

Local site cmk, 10.60.235.60	State	Service	Icons	Summary	Age	Checked	Perf-O-Meter
OK	Check_MK			[snmp] Success, [piggyback] Success (but no data found for this host), execution time 0.7 sec	8.17 s	8.17 s	710 ms
OK	Check_MK			Services: all up to date, Host labels: all up to date	114 s	114 s	
OK	Discovery			Battery status: normal, Output status: on line (calibration invalid), Capacity: 100%, Time remaining: 14 hours 23 minutes	7.18 s	7.18 s	
OK	APC Symmetra status				7.18 s	7.18 s	
OK	Interface 2			[vmac0], (up), MAC: 28:29:86:30:25:B3, Speed: unknown	7.18 s	7.18 s	
OK	Phase Input			Voltage: 239.0 V	7.18 s	7.18 s	239 V
OK	Phase Output			Voltage: 230.0 V, Current: 0.0 A, Load: 0%	7.18 s	7.18 s	0%
OK	Self Test			Result of self test: OK, Date of last test: 03/18/2024	7.18 s	7.18 s	
OK	SNMP Info			APC Web/SNMP Management Card (MB:v4.2.9 PF:v1.1.0.16 PN:apc_hw21_aos_1.1.0.1.0.1.0.16 bin AF1:v1.1.0.16 AN1:apc_hw21_su_1.1.0.16.bin MN:AP9640 HR:5 SN:ZA2010550704 MD:03/05/2020) (Embedded PowerNet SNMP Agent SW v2.2 compatible), FRETUPS0017, IDF17, Unknown	7.18 s	7.18 s	
OK	System events			No service events	7.18 s	7.18 s	
OK	Temperature				7.18 s	7.18 s	22 °C
OK	Battery			22.0 °C	7.18 s	7.18 s	
OK	Uptime			Up since Mar 19 2024 09:15:42, Uptime: 1 hour 26 minutes	7.18 s	7.18 s	86 m

Figure 37 : Supervision de XRD P RPI

[En rapport avec les screens de supervision en salle de stock qui offre la disponibilité d'une connexion bureau à distance \(Hydra / Zabbix\).](#)



Status xrdp
root@raspberrypi:/home/forvia\$ systemctl status xrdp
● xrdp.service - xrdp daemon
 Loaded: loaded (/lib/systemd/system/xrdp.service; enabled; vendor preset: enabled)
 Active: active (running) since Mon 2024-03-25 12:19:01 CET; 1min 0s ago
 Docs: man:xrdp(8)
 Main PID: 3512 (xrdp)
 Tasks: 1 (limit: 4609)
 Memory: 286K
 CPU: 28ms
 CGroup: /system.slice/xrdp.service
 └─ 3512 /usr/sbin/xrdp

status xrdp

Figure 38 : Schéma de l'intégration à Hydra

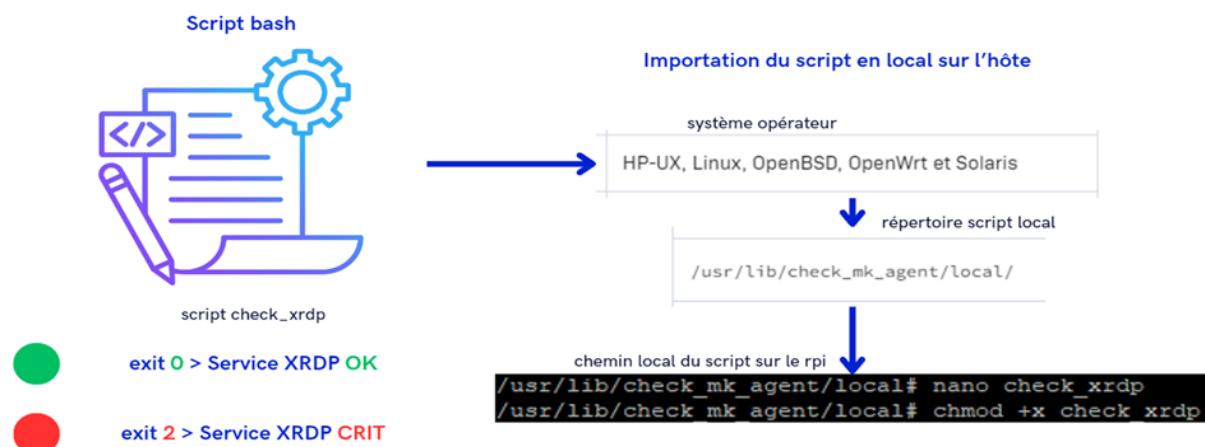


Figure 39 : script check_xrdp

```
1  #!/bin/bash
2
3  if systemctl is-active --quiet xrdp; then
4      # Service XRD P en cours d'exécution, renvoie un code OK
5      echo "0 XRD P_Check - XRD P est en cours d'exécution."
6      exit 0
7  else
8      # Service XRD P non en cours d'exécution, renvoie un code CRIT
9      echo "2 XRD P_Check - XRD P n'est pas en cours d'exécution."
10     exit 2
11 fi
```

Figure 40 : Service discovery xrdp_check



Figure 41 : Check service XRDП UP

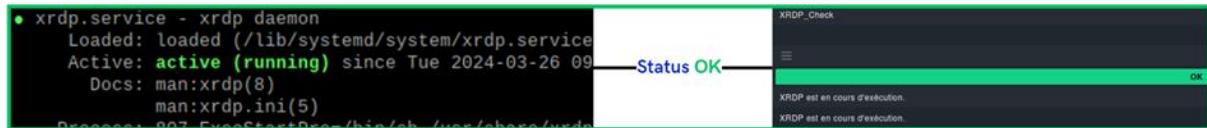


Figure 42 : Check service XRDП Down

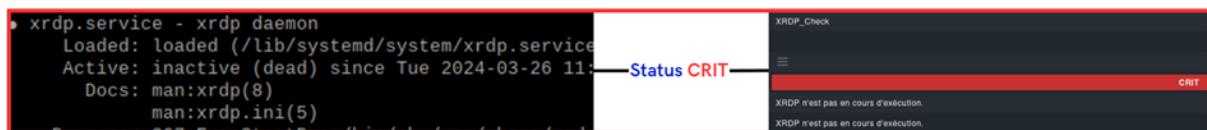


Figure 43 : Service discovery ssh_check

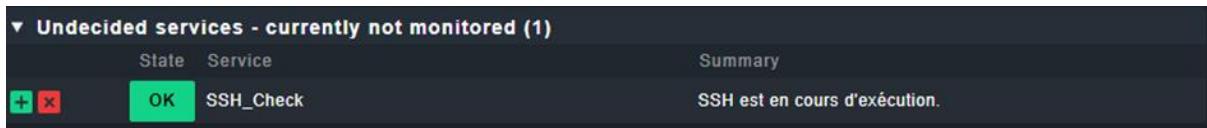


Figure 44 : schéma Dashboard orienté IT

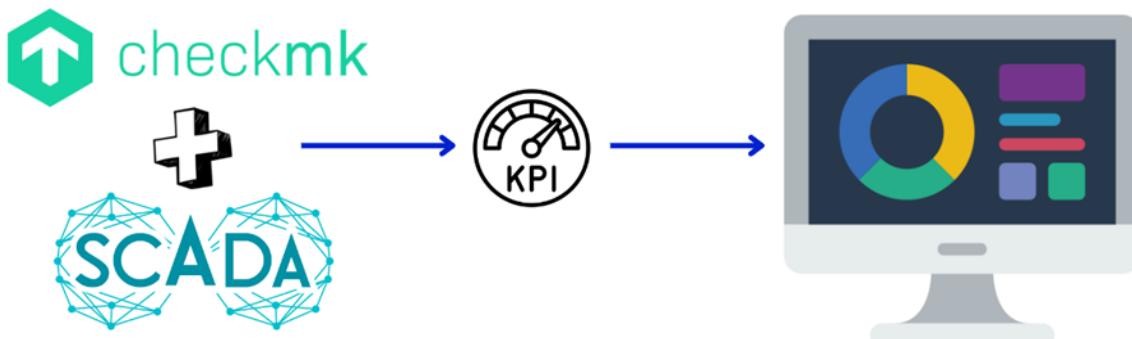


Figure 45 : Structure de 2 solutions envisageable

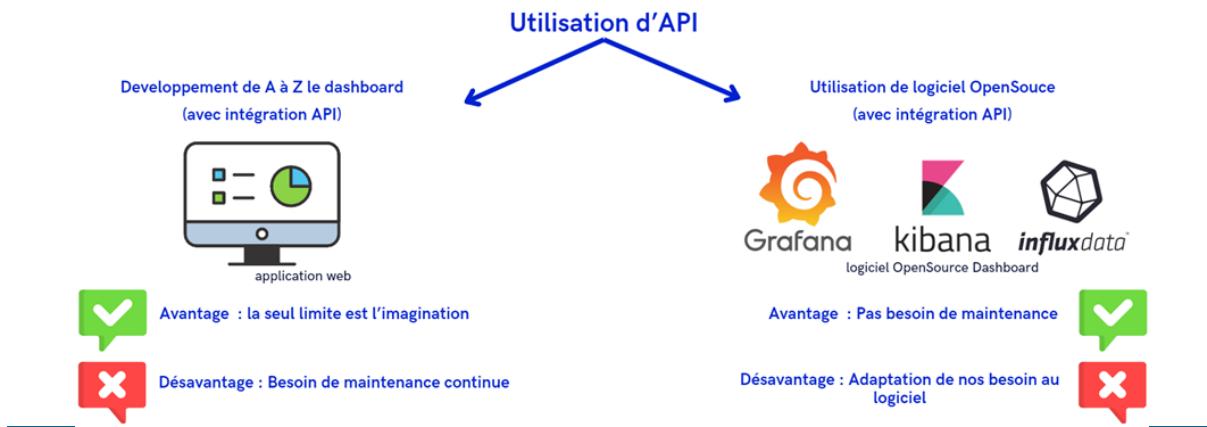


Figure 46 : Commandes d'installation de Grafana

```

1 apt-get install -y apt-transport-https software-properties-common wget
2 mkdir -p /etc/apt/keyrings/
3 wget -q -O - https://apt.grafana.com/gpg.key | gpg --dearmor | sudo tee /etc/apt/keyrings/grafana.gpg > /dev/null
4 echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://apt.grafana.com stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list
5 echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://apt.grafana.com beta main" | sudo tee -a /etc/apt/sources.list.d/grafana.list
6 apt-get install grafana-enterprise -y

```

Figure 47 : Installation CheckMK plugins sur Grafana

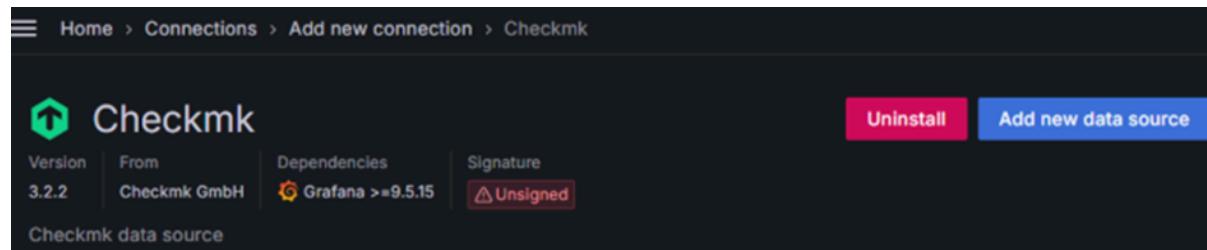


Figure 48 : Configuration de CheckMK plugins sur Grafana

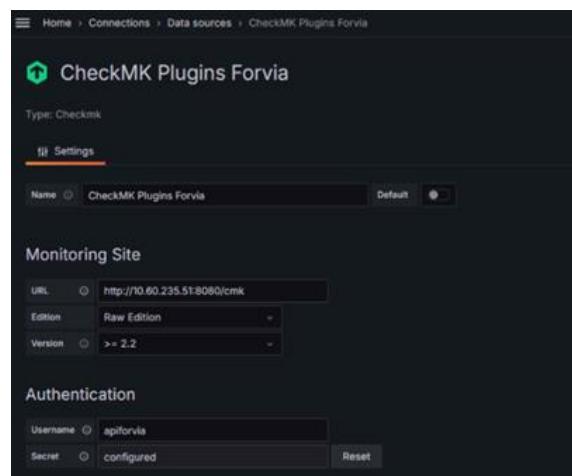


Figure 49 : Configuration et ajout au Dashboard

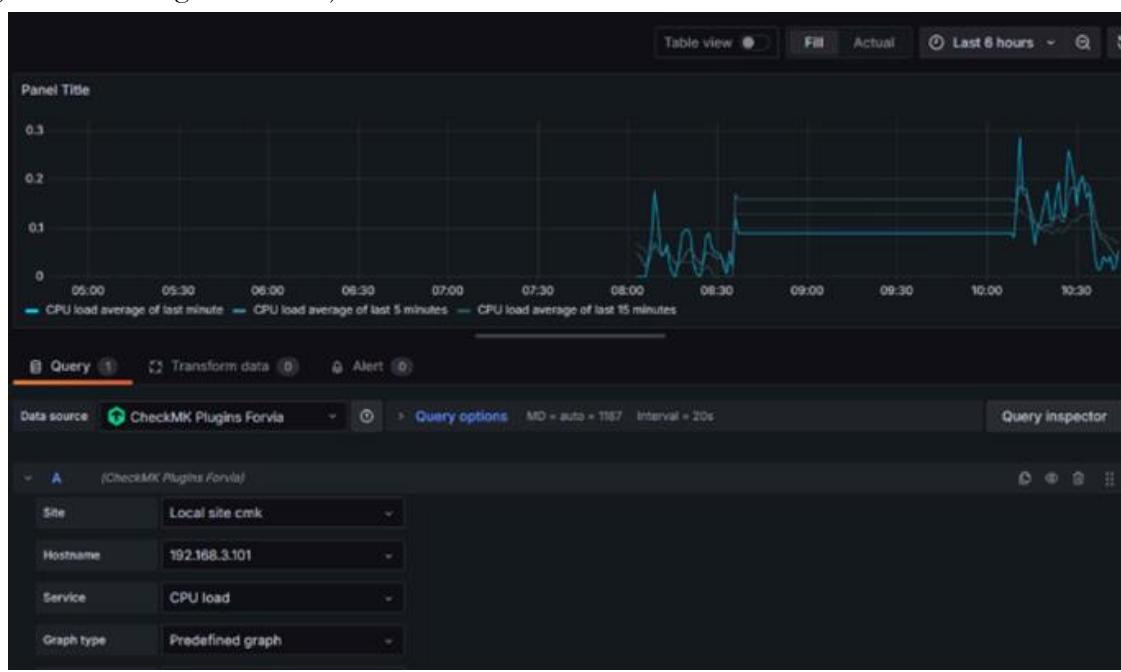


Figure 50 : Série d'étapes d'exploration de donnée json API

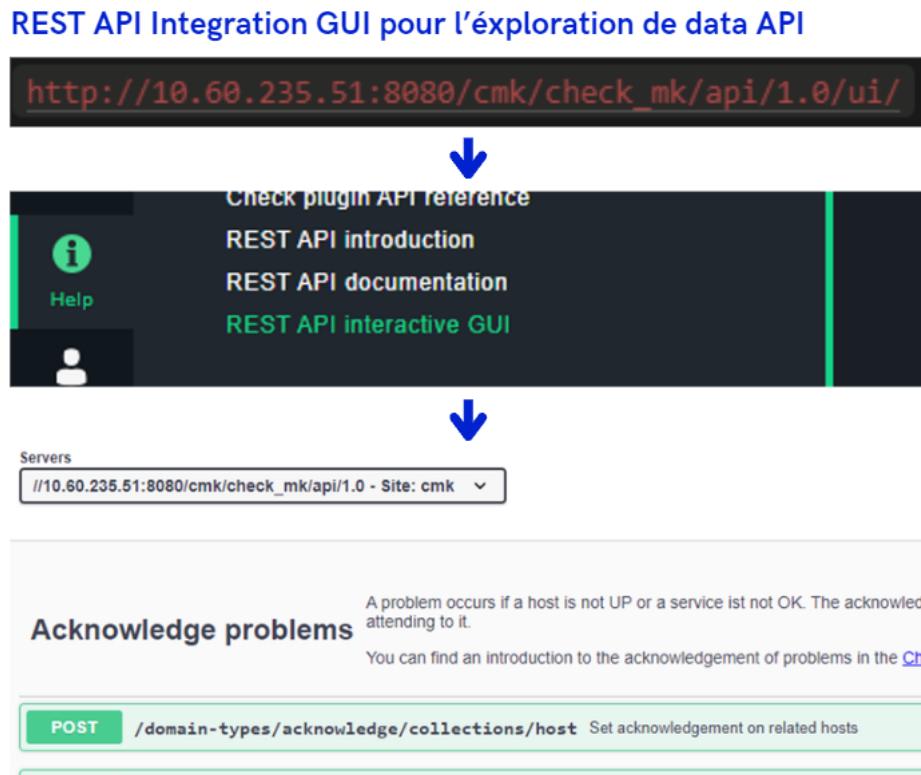


Figure 51 : Série d'étapes configuration JSON API sur le dashboard

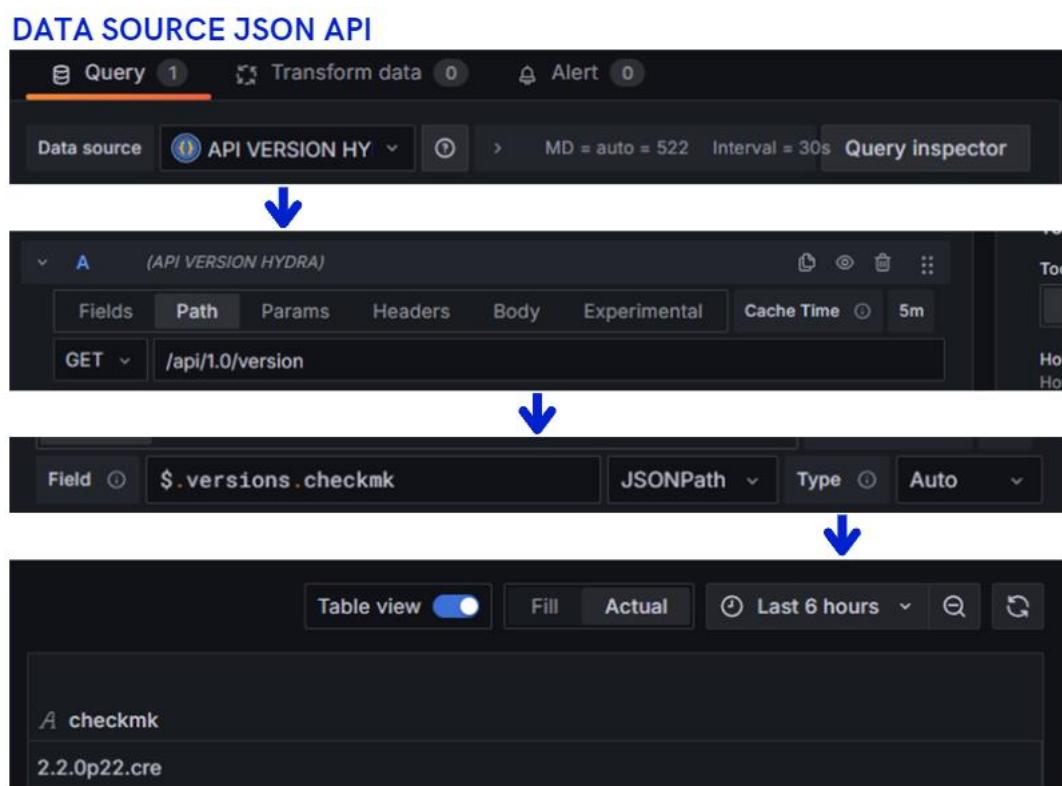


Figure 52 : Analyse JSON

```

{
  "site": "cmk",
  "group": "",
  "rest_api": {
    "revision": "0"
  },
  "versions": {
    "apache": [2, 4, 52],
    "checkmk": "2.2.0p22.cre",
    "python": "3.11.5 (main, Nov 30 2023, 14:57:54) [GCC 13.2.0]",
    "mod_wsgi": [4, 9, 4],
    "wsgi": [1, 0]
  }
},

```

Figure 53 : Dashboard Prototype

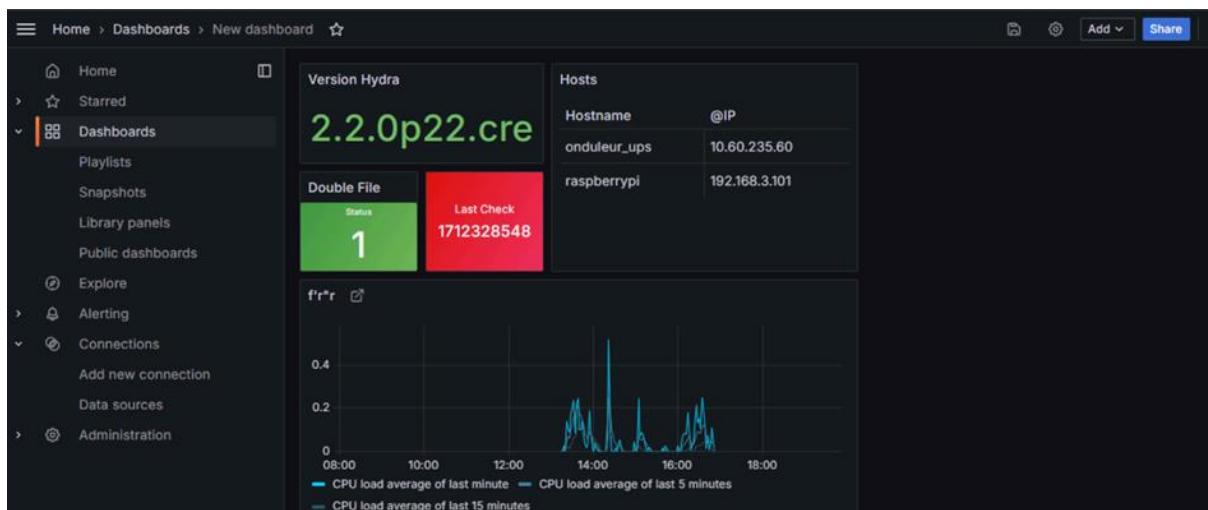


Figure 54 : Structure du cluster

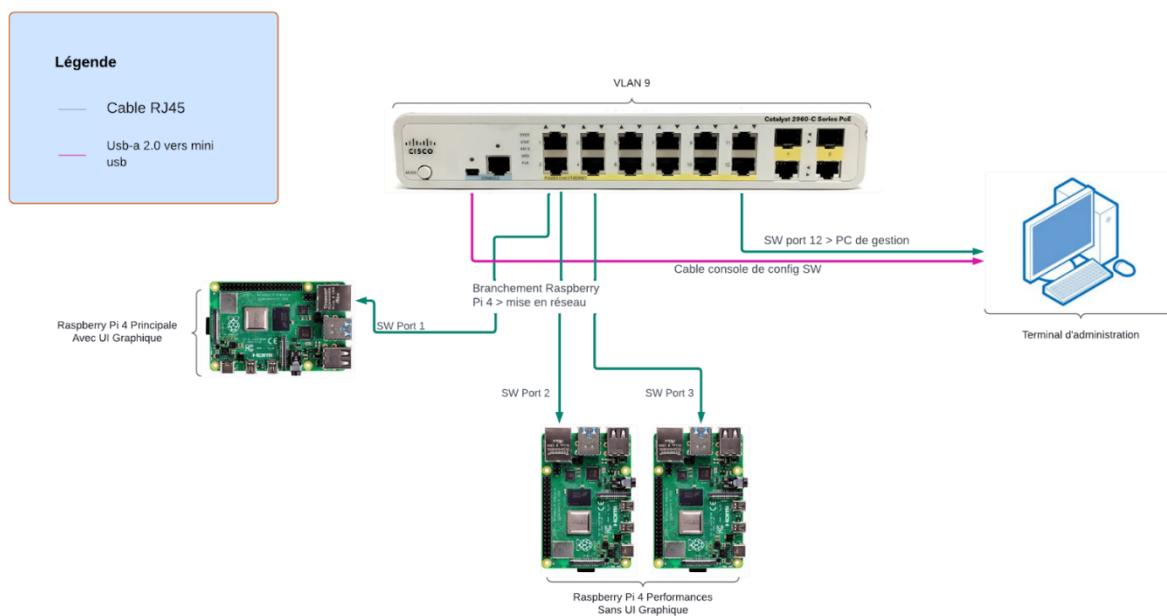


Figure 55 : Schéma cluster

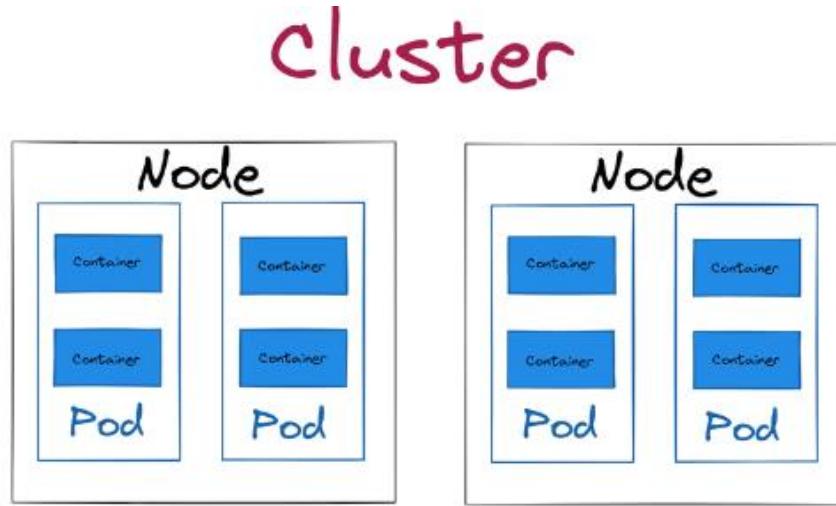


Figure 56 : Configuration du switch

Configuration du Switch

```

fralsswt0054#conf t
Enter configuration commands, one per line. End with CNTL/Z.
fralsswt0054(config)#int Fas
fralsswt0054(config)#int FastEthernet0/1
fralsswt0054(config-if)#swi
fralsswt0054(config-if)#switchport mode access
fralsswt0054(config-if)#switchport access vlan 9
fralsswt0054(config-if)#int FastEthernet0/2
fralsswt0054(config-if)#switchport mode access
fralsswt0054(config-if)#switchport access vlan 9
fralsswt0054(config-if)#int FastEthernet0/3
fralsswt0054(config-if)#switchport mode access
fralsswt0054(config-if)#switchport access vlan 9
fralsswt0054(config-if)#int FastEthernet0/4
fralsswt0054(config-if)#switchport mode access
fralsswt0054(config-if)#switchport access vlan 9
fralsswt0054(config-if)#int FastEthernet0/12
fralsswt0054(config-if)#switchport mode access
fralsswt0054(config-if)#switchport access vlan 9
fralsswt0054(config-if)#end
fralsswt0054#wr mem
Building configuration...
[OK]
  
```

Status des interfaces du Switch

Port	Name	Status	Vlan	Duplex	Speed	Type
Fa0/1	users	connected	9	a-full	a-100	10/100BaseTX
Fa0/2	users	connected	9	a-full	a-100	10/100BaseTX
Fa0/3	users	connected	9	a-full	a-100	10/100BaseTX
Fa0/4	users	notconnect	9	auto	auto	10/100BaseTX
Fa0/5	users	notconnect	11	auto	auto	10/100BaseTX
Fa0/6	users	notconnect	11	auto	auto	10/100BaseTX
Fa0/7	users	notconnect	11	auto	auto	10/100BaseTX
Fa0/8	users	notconnect	11	auto	auto	10/100BaseTX
Fa0/9		notconnect	1	auto	auto	10/100BaseTX
Fa0/10		notconnect	1	auto	auto	10/100BaseTX
Fa0/11		notconnect	1	auto	auto	10/100BaseTX
Fa0/12		notconnect	9	auto	auto	10/100BaseTX
Glo/1	UPLINK	notconnect	1	auto	auto	Not Present
Glo/2	UPLINK	notconnect	1	auto	auto	Not Present

Les 3 RPI 4 sont donc connecté sur le interfaces
Fa0/1 , Fa0/2 et Fa0/3

Figure 57 : Docker to Docker Swarm

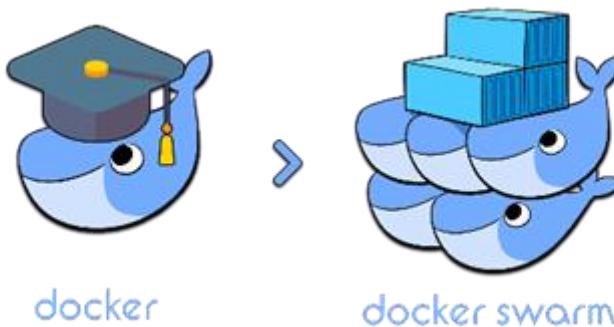


Figure 58 : Schéma de clusterisation

Installation et configuration de Docker swarm —

Génération du token depuis le maître

```
root@forviamain:/home/fvmain$ sudo docker swarm init --advertise-addr 10.0.0.1
Swarm initialized: current node (s8uuy55h7n9v0xqakyhlhiz6k) is now a manager.

To add a worker to this swarm, run the following command:

  docker swarm join --token SWMTKN-1-3tvjmoamhdmrjp24030cczgpgkg2m0exne9c866214jny49nqi-8q60uzlirflid9am0ghuozs8xl 10.0.0.1:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.
```

```
root@forviamain:/home/fvmain/Cluster# docker --version
Docker version 26.0.2, build 3c863ff
root@forviamain:/home/fvmain/Cluster# [REDACTED]
```

version docker

Intégration des au maître en tant que worker

```
root@forviaperf1:/home/fvperf1$ docker swarm join --token SWMTKN-1-3tvjmoamhdmrjp24030cczgpgkg2m0exne9c866214jny49nqi-8q60uzlirflid9am0ghuozs8xl 10.0.0.1:2377
This node joined a swarm as a worker.
root@forviaperf1:/home/fvperf1# [REDACTED]
```

Screenshot de la génération de token docker swarm



Status des noeuds du cluster

ID	HOSTNAME	STATUS	AVAILABILITY	MANAGER STATUS	ENGINE VERSION
s8uuy55h7n9v0xqakyhlhiz6k *	forviamain	Ready	Active	Leader	26.0.2
synfcsf9a7cywe842rsohvst9	forviaperf1	Ready	Active		26.0.2
2t70zq05q2bvf2iybbpk7eg08	forviaperf2	Ready	Active		26.0.2

Status des noeuds du cluster

Figure 59 : Haute disponibilité



Figure 60 : Extraits de code du fichier docker-compose.yml

```
3   services:
4     nginx-master:
5       image: nginx:latest
6       ports:
7         - "80:80"
17    nginx-perf1:
18       image: nginx:latest
19       ports:
20         - "8081:80"
28    nginx-perf2:
29       image: nginx:latest
30       ports:
31         - "8082:80"
```

Figure 61 : extrait de code du fichier load balancer

```
3   http {
4     upstream backend {
5       server nginx-master;
6       server nginx-perf1;
7       server nginx-perf2;
8     }
```

Figure 62 : extrait de code du fichier default.conf

```
1   server {
2     listen 80;
3     server_name _;
4
5     location / {
6       root /usr/share/nginx/html;
7       index index.html;
8     }
9 }
```

Figure 63 : extrait de code du fichier index.html

```
5   <title>Cluster Forvia</title>
7   </head>
8   <body>
9   <h1>Redondance Cluster Forvia</h1>
```

Figure 64 : Quelle solution d'économie prendre ?



Figure 65 : Sensibiliser les utilisateurs

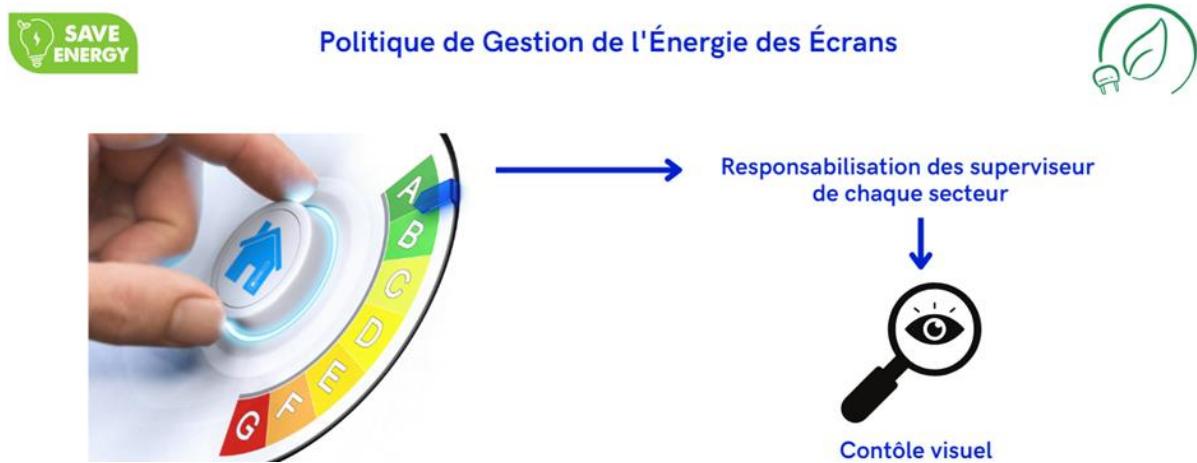


Figure 66 : Système de gestion d'énergie

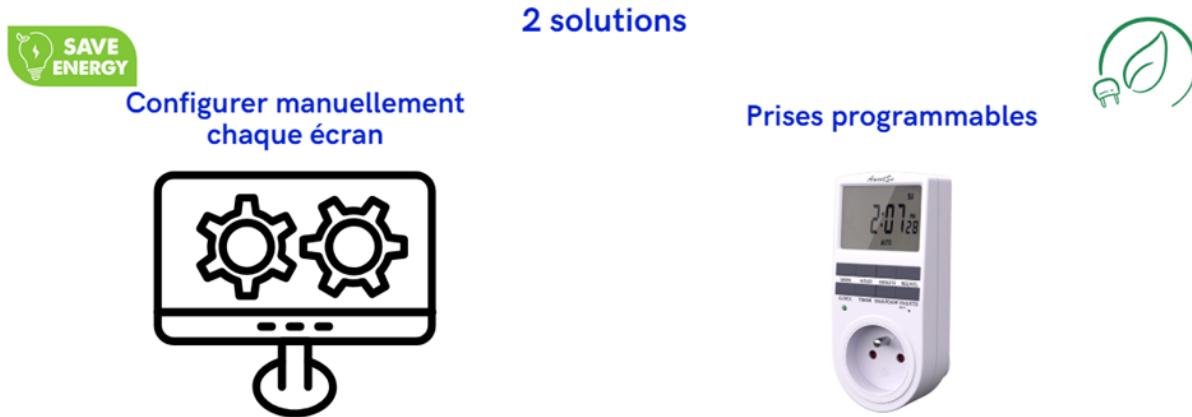


Figure 67 : Prise Programmable Digitale



Figure 68 : Écran choisi d'exemple

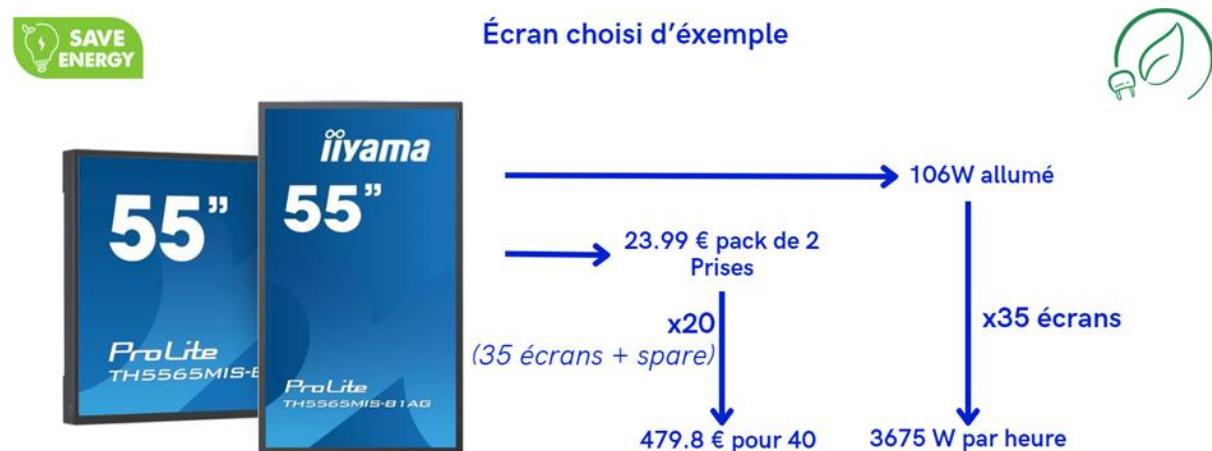


Figure 69 : Graphique d'estimation consommation en kWh

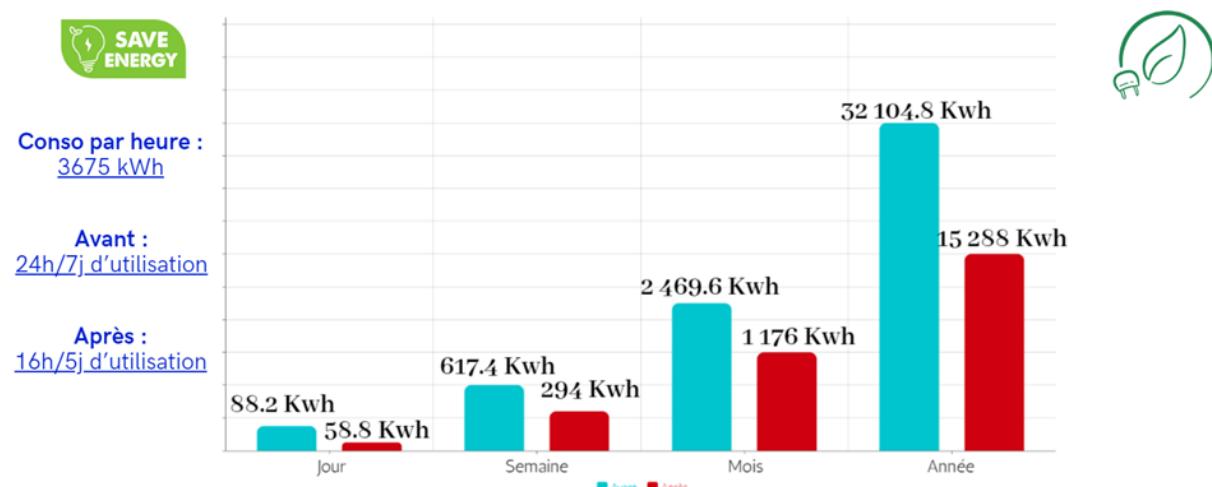


Figure 70 : Graphique d'estimation des coûts

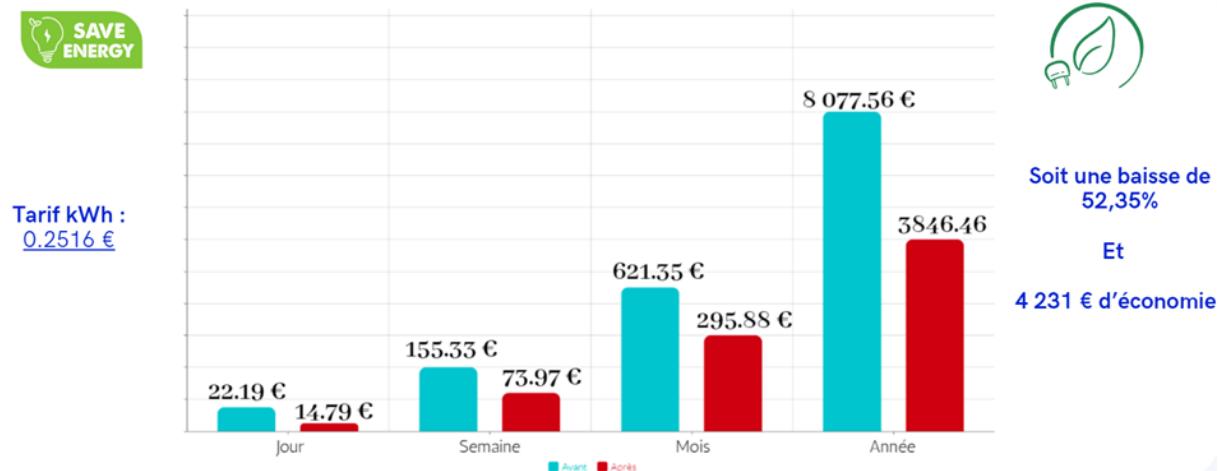


Figure 71 : Configuration d'imprimante Zebra

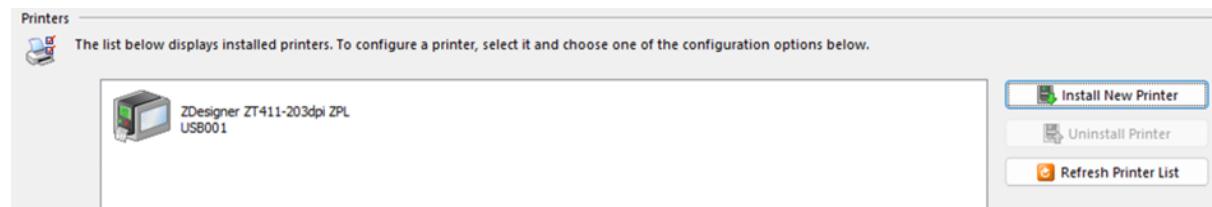


Figure 72 : Cartographie des imprimantes

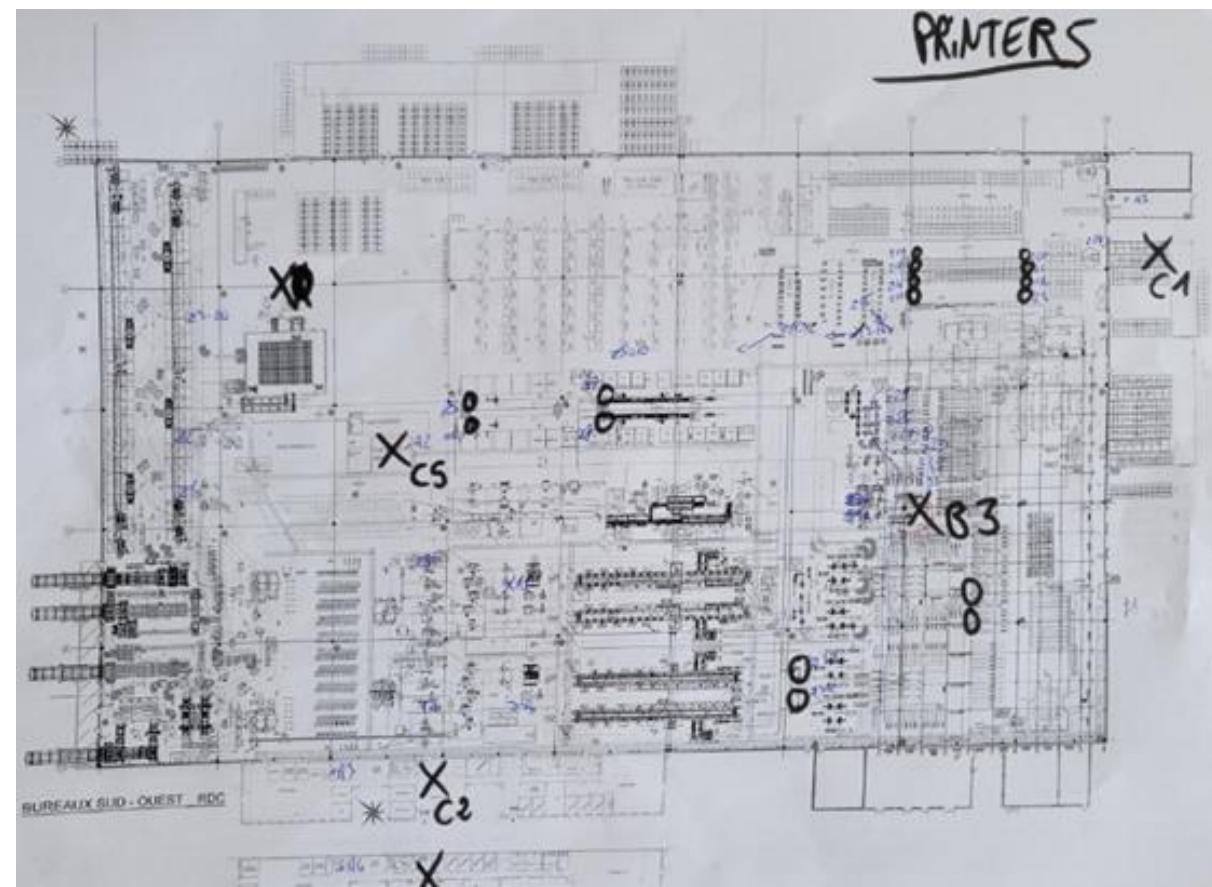


Figure 73 : Note des @IP et Carto détaillé

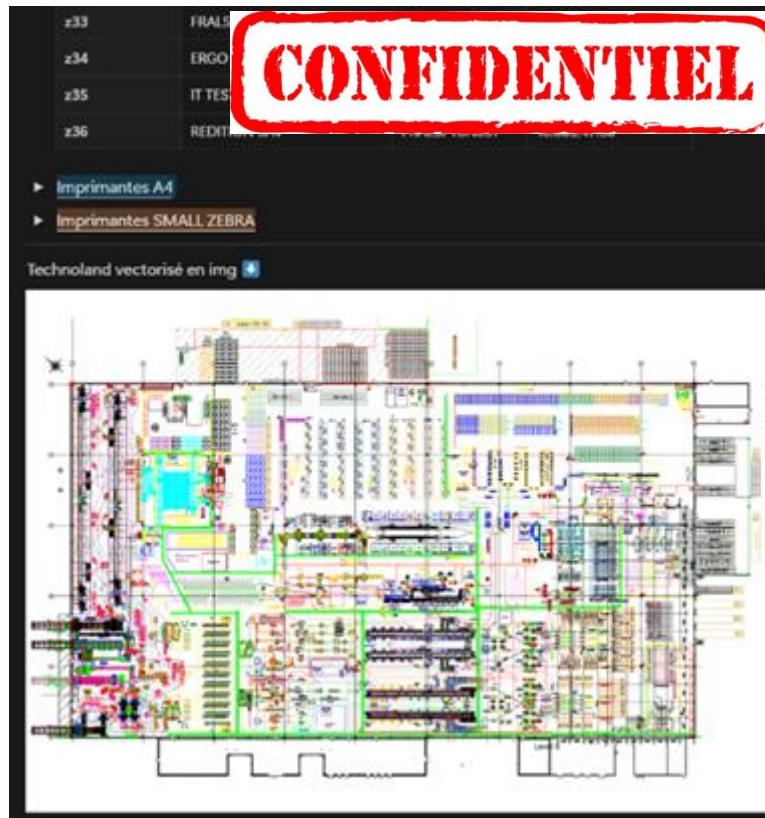


Figure 74 : Contact Zebra



Figure 75 : Script de comparaison

```

1 $fichierCSV = "C:\Users\elhamiou\Documents\Data_overview_Forvia_cleaned.csv" #Modifier avec le chemin du bon fichier .csv
2
3 $dossierXML = "C:\Users\elhamiou\Documents\xml" #Il faut modifier par le chemin du dossier contenant les fichiers
4 $listeVIN = @(Get-Content $fichierCSV | Select-Object -Skip 1)
5
6 foreach ($xmlFilePath in Get-ChildItem $dossierXML -Filter *.xml) {
7     $ContentXML = [xml](Get-Content $xmlFilePath.FullName)
8     $idVIN = $ContentXML.SelectSingleNode("//VIN")
9     if ($idVIN -ne $null) {
10         $vin = $idVIN.InnerText
11         if ($vin -notin $listeVIN) {
12             $nvNomFichier = $xmlFilePath.Name -replace '\.xml$', '.miss' #Ici on peut changer .miss par le rename souhaité, par exemple .old ou autres...
13             $newFilePath = Join-Path -Path $dossierXML -ChildPath $nvNomFichier
14             Rename-Item -Path $xmlFilePath.FullName -NewName $newFilePath
15             Write-Output "Le fichier $($xmlFilePath.Name) a été renommé en $nvNomFichier." #On peut suppr cette ligne si on veut éviter l'echo sur le terminal
16         }
17     } else {
18         Write-Output "Pas de vin trouvé $($xmlFilePath.Name)." #On peut suppr cette ligne si on veut éviter l'echo sur le terminal
19     }
20 }
```

Figure 76 : Script de suppression

```

1 $dossierXML = "C:\Users\elhamiou\Documents\xml" #Il faut modifier le chemin par le chemin du dossier contenant les fichiers
2
3 foreach ($file in Get-ChildItem $dossierXML) {
4     if ($file.Extension -ne ".miss") { #Si besoin, changer ".miss" par le rename souhaité, par exemple .old ou autres...
5         Remove-Item -Path $file.FullName -Force
6         Write-Output "Le fichier $($file.Name) a été supprimé." #On peut suppr cette ligne si on veut éviter l'echo sur le terminal
7     }
8 }
```

Figure 77 : Script de Restauration

```

1 $dossierXML = "C:\Users\elhamiou\Documents\xml" #Il faut modifier le chemin par le chemin du dossier contenant les fichiers
2
3 foreach ($file in Get-ChildItem $dossierXML -Filter "*.miss") {
4     $nvNomFichier = $file.Name -replace '\.miss$', ''
5
6     $newFilePath = Join-Path -Path $file.DirectoryName -ChildPath $nvNomFichier
7
8     Rename-Item -Path $file.FullName -NewName $newFilePath -Force
9     Write-Output "Le fichier $($file.Name) a été renommé en $nvNomFichier."
10 }
```

Figure 78 : Fichier readme.txt

```

Instructions d'utilisation

Avant de commencer, quelques manip à faire :
1. Trier le fichier excel > supprimer les colonnes et les lignes inutiles
2. Convertir le fichier excel en .csv
3. Se rendre dans le script "script_delta" et modifier la variable "$fichierCSV" > remplacer "C:\Users\elhamiou\Documents\Data_overview_Forvia_cleaned.csv" par le chemin réel de votre fichier csv
4. Se rendre dans chaque script et modifier la variable "$dossierXML" et remplacer "C:\Users\elhamiou\Documents\xml" par le chemin réel de votre dossier contenant les fichiers xml

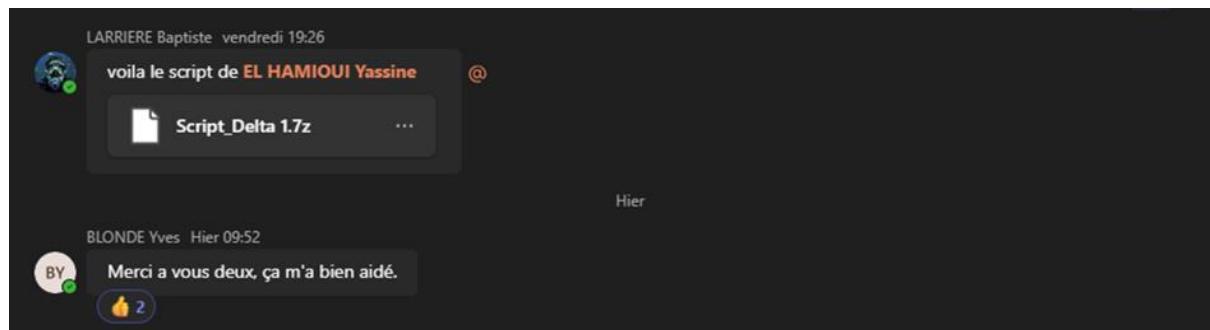
// EXECUTION DU SCRIPT \\

DéTECTER et Renommer les fichiers Delta
> Lancer PowerShell > Executer le script : /cheminduscript/script_delta.ps1
Ce script détectera les fichiers delta dans le répertoire spécifié et les renommera.

Pour conserver uniquement les fichiers delta :
> Lancer PowerShell > Executer le script : /cheminduscript/suppr_non_miss.ps1
Ce script supprimera tous les fichiers qui ne sont pas des fichiers delta, laissant seulement ceux qui sont pertinents.

Enfin, pour retirer l'extension ".miss" des fichiers d'origine, suivez ces étapes :
> Lancer PowerShell > Executer le script : /cheminduscript/renome_origin.ps1
Ce script renommera tous les fichiers d'origine en retirant l'extension ".miss".
```

Figure 79 : preuve d'échange avec n+2



11.2 Documents techniques et procédures créées

11.2.1 Procédure de vérification du statut des sièges dans IJCore (2 pages)

Procédure de Vérification du Statut des Sièges dans IJCore

Procédure de Vérification du Statut des Sièges

1. Accéder à l'Outil

1. Naviguez vers l'URL: <http://CONFIDENTIEL/manualmanagementcontainer/main>
2. Connexion : Entrez vos identifiants personnels (nom d'utilisateur et mot de passe).

2. Utiliser les Filtres

1. Accéder aux Filtres : Sur la page Container Management, cliquez sur l'onglet Filters situé en haut à gauche du tableau > Un formulaire va s'ouvrir.

Container management

Filters Appuyez ici pour configurer les filtres

3. Configurer les Filtres

1. Supprimer les Filtres puis Ajouter un Nouveau Filtre : Cliquez sur le bouton + pour ajouter un nouveau filtre.

Filtres

Filter	Condition	Value	Ajouter un filtre
Container date created	Greater Than Or Equal	05/30/2024 12:00:00 AM	
Container date created	Less Than Or Equal	05/30/2024 11:59:59 PM	

Supprimer les 2 filters par default

2. Configurer le Nouveau Filtre :

- o Type de Filtre : choisissez « JIT Order »
- o Condition : « Contains »
- o Value : entrez le numéro de « séquence interne du module »

Filtres

Filter	Condition	Value
JIT order	Contains	892431450682

remplacer ce n° par la référence du siège

Appliquer le filtre

Page 1

Procédure de Vérification du Statut des Sièges dans IJCore

7. Visualisation des Détails du Conteneur

- Application du Filtre > Accéder aux Détails :** Cliquez sur les trois petits points situés à gauche du tableau à gauche du conteneur concerné.

Actions	Configuration	Container Internal Seq.	Container Internal Seq. Ctr.	Status	Proces Status	Created	Closed	Matched	Total Items	Container ID
...	Container_SCX	1567	2	20000	100000	27/May/2024, 08:16:17	27/May/2024, 08:16:20	-	3	251368

Showing 1 to 1 of 1 rows

Un menu déroulant s'ouvrira, affichant les détails du conteneur sélectionné.

8. Vérification du Statut du Conteneur

Consulter la Colonne "Status" : Dans le menu des détails, repérez la colonne intitulée Status.

Actions	Configuration	Container Internal Seq.	Container Internal Seq. Ctr.	Status	Proces Status	Created	Closed	Matched	Total Items	Container ID
...	Container_SCX	1567	2	20000	100000	27/May/2024, 08:16:17	27/May/2024, 08:16:20	-	3	251368

Showing 1 to 3 of 3 rows

Container Details								
Actions	Customer Seq.	Assembly Seq.	Engine Seq.	Position	Is Missing	Is External	Is Swap	Status
...	11575	115200	11575	1	No	No	No	10000
...	11575	115200	11575	1	No	No	No	10000
...	11575	115200	11575	1	No	No	No	10000

N° status du siège

9. Interprétation du Statut

- Se référer au Tableau Annexe :** Pour comprendre la signification du statut, référez-vous au tableau annexé en bas de cette page.
 - Exemple :** Dans notre cas, le statut affiché est « 10000 », ce qui signifie « Module order is assigned » soit Module non OK.

Tableau Annexe : États des Statuts

Numéro de Statut	Signification
10000	Module non OK
20000	Module OK
30000	Module clos

11.2.2 Procédure mise en place d'un cluster avec Haute disponibilité (4 pages)

Procédure mise en place d'un cluster avec HA

Guide de Configuration d'un Cluster NGINX en Haute Disponibilité sur Raspberry Pi

Ce guide fournit une procédure détaillée pour configurer un cluster NGINX en haute disponibilité sur Raspberry Pi. Nous utiliserons Docker Swarm pour orchestrer les services et assurer la redondance et l'équilibrage de charge.

Pré-requis

1. Plusieurs Raspberry Pi (au moins trois pour un cluster Swarm efficace)
2. Docker et Docker Swarm installés sur chaque Raspberry Pi
3. Accès SSH à chaque Raspberry Pi
4. Un réseau local (LAN) pour interconnecter les Raspberry Pi

Étapes de Configuration

1. Installation de Docker et Docker Swarm

1.1. Installer Docker

Sur chaque Raspberry Pi, exécutez les commandes suivantes pour installer Docker:

```
curl -fsSL https://get.docker.com -o get-docker.sh
sh get-docker.sh
sudo usermod -aG docker $(whoami)
```

Déconnectez-vous et reconnectez-vous pour appliquer les changements de groupe.

1.2. Initialiser Docker Swarm

Sur le Raspberry Pi principal (le manager), initialisez le cluster Docker Swarm:

```
sudo docker swarm init --advertise-addr <IP_ADDRESS>

forviamain@raspberrypi:~ $ sudo docker swarm init --advertise-addr 10.0.0.1
Swarm initialized: current node (8c3e7rzuyeygmyow3ambts2v7q) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-18b05b6jvwtkyxrk0no5js012xb2gr0xwkckbh9s1vs1bac4su-bd3t8rmm72964epeya76p0d7f 1
0.0.0.1:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.
```

Notez la commande fournie pour ajouter des nœuds workers au cluster. Sur chaque nœud worker, exécutez cette commande pour rejoindre le cluster.

Par exemple, dans ce cas : `docker swarm join --token SWMTKN-1-18b05b6jvwtkyxrk0no5js012xb2gr0xwkckbh9s1vs1bac4su-bd3t8rmm72964epeya76p0d7f 10.0.0.1:2377` à rentrer dans la commande des worker

Page 1

2. Configuration du Cluster

2.1. Créer un Réseau Overlay

Sur le manager, créez un réseau overlay pour la communication entre les services Docker:

```
sudo docker network create --driver overlay --subnet=10.0.9.0/24 my_nginx_stack_my_network
```

2.2. Créer un Fichier docker-compose.yml

Créez un fichier `docker-compose.yml` pour définir les services NGINX et le load balancer.

```
version: '3.7'

services:
  load_balancer:
    image: nginx:latest
    ports:
      - "80:80"
    networks:
      - my_network
    deploy:
      replicas: 1
      placement:
        constraints: [node.role == manager]
    configs:
      - source: nginx_load_balancer_config
        target: /etc/nginx/nginx.conf
  nginx-master:
    image: nginx:latest
    networks:
      - my_network
    deploy:
      replicas: 1
  nginx-perf1:
    image: nginx:latest
    networks:
      - my_network
    deploy:
      replicas: 1
  nginx-perf2:
    image: nginx:latest
    networks:
      - my_network
    deploy:
      replicas: 1
  my_network:
    external: true
    configs:
      - nginx_load_balancer_config;
```

Page 2

Procédure mise en place d'un cluster avec HA

file: ./nginx_load_balancer.conf

2.3. Créer le Fichier de Configuration NGINX pour le Load Balancer

Créez un fichier `nginx_load_balancer.conf` pour configurer le load balancer NGINX:

```
events {}  
http {  
    upstream backend {  
        server nginx-master:80;  
        server nginx-perf1:80;  
        server nginx-perf2:80;  
    }  
    server {  
        listen 80;  
        location / {  
            proxy_pass http://backend;  
            proxy_set_header Host $host;  
            proxy_set_header X-Real-IP $remote_addr;  
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
            proxy_set_header X-Forwarded-Proto $scheme;  
        }  
    }  
}
```

3. Déploiement du Cluster

3.1. Déployer la Stack Docker

Sur le manager, déployez la stack Docker:

`sudo docker stack deploy -c docker-compose.yml my_nginx_stack`

```
root@forviamain:~/cluster# sudo docker stack deploy -c docker-compose.yml my_nginx_stack  
Since --detach=false was not specified, tasks will be created in the background.  
In a future release, --detach=false will become the default.  
Creating network my_nginx_stack_default  
Creating network my_nginx_stack_my_network  
Creating service my_nginx_stack_nginx-master  
Creating service my_nginx_stack_nginx-perf1  
Creating service my_nginx_stack_nginx-perf2  
Creating service my_nginx_stack_load_balancer  
root@forviamain:~/cluster#
```

4. Configuration Réseau et Pare-feu

4.1. Ouvrir les Ports Nécessaires

Sur chaque Raspberry Pi, assurez-vous que les ports 80, 8080, 8081, et 8082 sont ouverts:

```
sudo ufw allow 80/tcp  
sudo ufw allow 8080/tcp  
sudo ufw allow 8081/tcp  
sudo ufw allow 8082/tcp  
sudo ufw reload
```

Page 3

Procédure mise en place d'un cluster avec HA

5. Vérification et Dépannage

5.1. Vérifier les Services

Vérifiez que les services NGINX sont en cours d'exécution:

```
sudo docker service ls
```

```
root@forviamain:~/cluster# sudo docker service ls
ID           NAME          MODE      REPLICAS  IMAGE          PORTS
k2mbke6mtc22 my_nginx_stack_load_balancer replicated 0/1    nginx:latest *:80->80/tcp
99ue58pty43b my_nginx_stack_nginx-master   replicated 1/1    nginx:latest *:8080->80/tcp
vdr7fqnm6vy8 my_nginx_stack_nginx-perfl   replicated 0/1    nginx:latest *:8081->80/tcp
m4ziu43ta52c my_nginx_stack_nginx-perf2   replicated 0/1    nginx:latest *:8082->80/tcp
root@forviamain:~/cluster#
```

5.2. Tester l'Accès

Depuis votre PC, essayez d'accéder à l'IP du manager ou du load balancer:

```
curl -I http://<mainforviaIP>
curl -I http://<mainforviaIP>:8080
curl -I http://<mainforviaIP>:8081
curl -I http://<mainforviaIP>:8082
```

5.3. Vérifier les Journaux

Vérifiez les journaux des services pour détecter d'éventuelles erreurs:

```
sudo docker service logs my_nginx_stack_load_balancer
sudo docker service logs my_nginx_stack_nginx-master
sudo docker service logs my_nginx_stack_nginx-perfl
sudo docker service logs my_nginx_stack_nginx-perf2
```

6. Assurer la Haute Disponibilité

Pour assurer une haute disponibilité, vous pouvez configurer plusieurs instances du load balancer et utiliser un DNS round-robin ou une solution de basculement pour distribuer les requêtes. Vous pouvez également utiliser des volumes Docker pour persister les configurations et les logs.

Conclusion

En suivant ce guide, vous avez configuré un cluster NGINX en haute disponibilité sur des Raspberry Pi en utilisant Docker Swarm. Vous avez également configuré un équilibrage de charge et assuré que les services sont accessibles et redondants. Pour toute assistance supplémentaire, veuillez consulter la documentation officielle de Docker et NGINX.

11.2.3 Procédure de configuration des EOL CAMERA (5 pages)

Procédure pour la configuration des EOL CAMERA

Guide de Configuration des RPI camera EOL

Ce guide fournit une procédure détaillée pour configurer les CAMERA EOL

Processus de Configuration

1. Image Raspbian Standardisée

- Utiliser l'image "Faur_raspberry_buster_V2.0.1", configurée avec les services nécessaires pour les caméras et leur intégration dans l'usine.

2. Utilisation du Logiciel de Configuration

- Un logiciel interne développé par Fernando Pinheiro est utilisé pour configurer le nom de la machine, attribuer l'adresse IP et visualiser le flux de la caméra en temps réel.

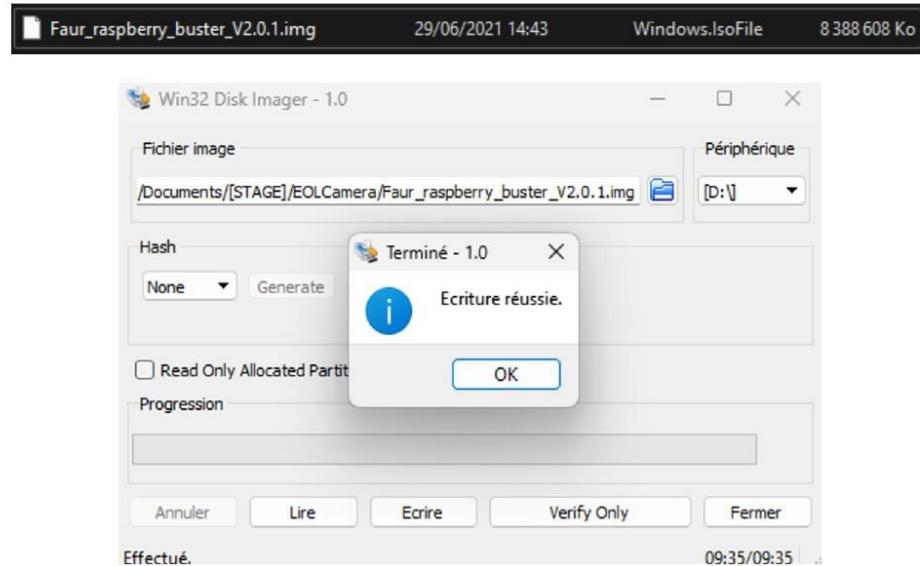
3. Étapes de Configuration

- Booster l'image Raspbian sur le Raspberry Pi.
- Utiliser le logiciel pour configurer l'adresse IP et le nom de la caméra.
- Identifier l'adresse IP actuelle de la caméra à l'aide de Wireshark pour les caméras déjà configurées, facilitant la connexion.

Préparation des Outils

Boot de l'Image :

Image Faur_raspberry_buster_v2.0.1.img avec Win32 Disk Imager.



Page 1

Procédure pour la configuration des EOL CAMERA

Matériel Nécessaire :

Un injecteur POE, deux câbles RJ45, et le Raspberry Pi équipé de la caméra et du POE hat.



Utilisation du Logiciel Interne de Configuration de Caméra

PiCameraConfiguration.exe 11/06/2024 08:54 Application 272 Ko

Détection du Raspberry Pi Camera

1. Détection Automatique :

Utilisez l'onglet "Device" dans le logiciel pour permettre la détection des dispositifs déjà configurés pour le déploiement de la caméra EOL.

NO.	MAC	IP	HOSTNAME
1	b8:27:eb:66:cd:bd	10.59.203.212	FRMONRSP0032
[Empty rows]			

Selected device: 10.59.203.212

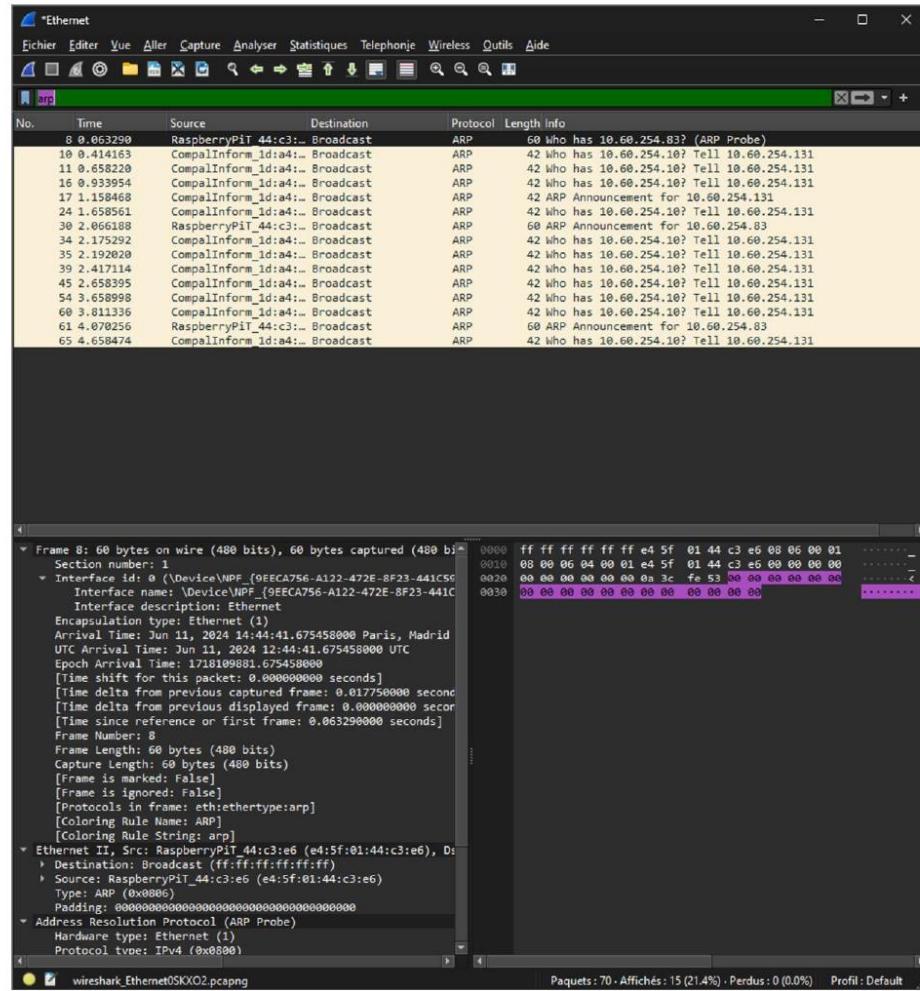
Page 2

Procédure pour la configuration des EOL CAMERA

Il est également possible d'attribuer manuellement une adresse IP spécifique. Cette fonctionnalité est disponible uniquement pour le Raspberry Pi avec la configuration caméra appliquée.

2. Détection Manuelle (Wireshark) :

Si la caméra n'est pas détectée automatiquement, utilisez Wireshark pour trouver l'adresse IP du Raspberry Pi caméra.



Page 3

Procédure pour la configuration des EOL CAMERA

Test de Communication

1. Une fois l'adresse IP de la caméra trouvée, allez dans la partie "Configuration".
2. Cochez "Allow connection" et appuyez sur "Test communication". Si le statut est vert, la connexion est établie, si rouge, la connexion à échoué.



Configuration Finale

1. Définition des Paramètres :

- o Définissez les paramètres pour le déploiement de la caméra EOL : Nom, Adresse IP et paramètres spécifiques pour utiliser le Raspberry Pi en tant que caméra EOL.



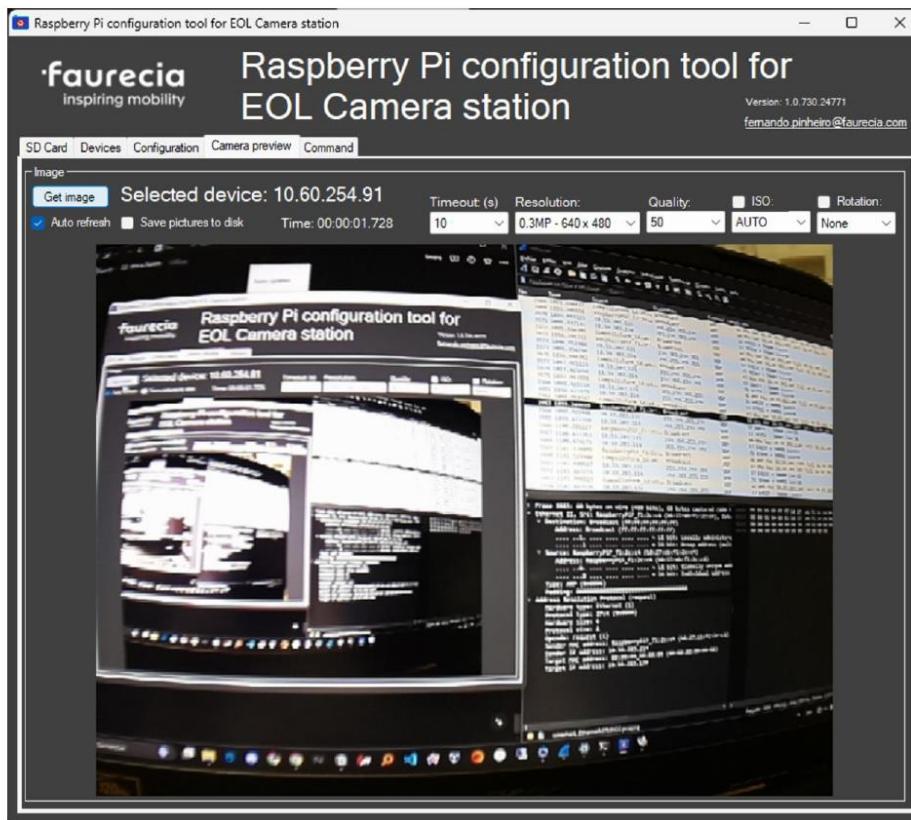
Page 4

Procédure pour la configuration des EOL CAMERA

2. Redémarrage et Prévisualisation :

Une fois la configuration terminée, redémarrez le système.

Pour la prévisualisation, aller sur la page "Camera Preview", cocher "Auto refresh" et appuyez sur "Get image" pour visualiser le flux de la caméra.



Cette procédure assure une configuration correcte et efficace de la caméra sur le Raspberry Pi, garantissant ainsi son intégration harmonieuse dans le système existant.

12 Bibliographie

12.1 Sources documentaires et références utilisées

DOCUMENTATION DOCKER

- URL : <https://docs.docker.com>
-

DOCUMENTATION CHECKMK

- URL : <https://docs.checkmk.com>
-

DOCUMENTATION GRAFANA

- URL : <https://grafana.com/docs>
-

DOCUMENTATION PYTHON

- URL : <https://docs.python.org/3/>
-

DOCUMENTATION NGINX

- URL : <https://nginx.org/en/docs/>
-

DOCUMENTATION RASPBERRY PI

- URL : <https://www.raspberrypi.org/documentation/>
-

DOCUMENTATION POWERSHELL

- URL : <https://docs.microsoft.com/fr-fr/powershell/>
-

DOCUMENTATION OFFICIELLE DE MICROSOFT SUR AUTOLOGON

- URL : <https://docs.microsoft.com/fr-fr/windows-server/administration/windows-commands/autologon>
-

INFORMATION SUR LES SYSTEMES SCADA

- URL : <https://www.scada.org>

GUIDE DE L'UTILISATEUR WIRESHARK

- URL : https://www.wireshark.org/docs/wsug_html_chunked/

SITE OFFICIEL DE FORVIA

- URL : <https://www.forvia.com>

DOCUMENTATION RASPBIAN OS

- URL : <https://www.raspberrypi.org/documentation/raspbian/>

MANUEL ITIL FOUNDATION

- URL : <https://www.axelos.com/best-practice-solutions/itil>

12.2 Acronyme

IT (INFORMATION TECHNOLOGY) : Le domaine de la technologie qui concerne l'utilisation de systèmes informatiques pour stocker, récupérer, transmettre et manipuler des données.

JIT (JUST IN TIME) : Une méthode de gestion de la production qui vise à réduire les stocks et les temps d'attente en produisant uniquement ce qui est nécessaire, quand cela est nécessaire, et en quantités exactes.

VM (VIRTUAL MACHINE) : Un environnement informatique émulé qui fonctionne comme un ordinateur virtuel, permettant l'exécution de systèmes d'exploitation et d'applications distincts sur un même matériel physique.

RPI (RASPBERRY PI) : Une série de petits ordinateurs monocartes à faible coût, largement utilisés pour l'éducation, les projets de bricolage et les applications industrielles.

API (APPLICATION PROGRAMMING INTERFACE) : Un ensemble de définitions et de protocoles qui permettent à différents logiciels de communiquer entre eux.

CSV (COMMA-SEPARATED VALUES) : Un format de fichier utilisé pour stocker des données tabulaires, dans lequel chaque ligne correspond à un enregistrement et chaque champ est séparé par une virgule.

EOL (END OF LINE) : Désigne les systèmes ou processus à la fin de la chaîne de production, souvent utilisés pour la vérification finale et la traçabilité des produits.

PXE (PREBOOT EXECUTION ENVIRONMENT) : Un environnement utilisé pour démarrer un ordinateur via le réseau avant le démarrage du système d'exploitation local, souvent utilisé pour installer des systèmes d'exploitation à distance.

R&D (RESEARCH AND DEVELOPMENT) : La division ou les activités d'une entreprise consacrées à la recherche de nouvelles connaissances et au développement de nouveaux produits ou procédés.

SCADA (SUPERVISORY CONTROL AND DATA ACQUISITION) : Un système de contrôle industriel qui utilise des ordinateurs, des réseaux de communication et des interfaces utilisateur graphiques pour superviser et contrôler les équipements et les processus industriels.

SNMP (SIMPLE NETWORK MANAGEMENT PROTOCOL) : Un protocole de gestion de réseau utilisé pour surveiller et gérer les périphériques réseau.

SSH (SECURE SHELL) : Un protocole réseau cryptographique pour sécuriser les communications sur un réseau non sécurisé, souvent utilisé pour l'accès à distance à des systèmes informatiques.

VIN (VEHICLE IDENTIFICATION NUMBER) : Un numéro unique attribué à chaque véhicule par le constructeur, utilisé pour identifier et tracer les véhicules.