Rapport de stage

Développeur junior

HARO Antoine

MAI 2025 à JUIN 2025

Tutrice de stage : MOURET Lydia

Enseignant référent : PUEL Christophe

Établissement d'accueil : Cegedim Assurances - Green Park 298 all. du Lac, 31670 Labège

Établissement formateur : Institut Limayrac - 50 Rue de Limayrac, 31500 Toulouse

Sommaire

IV. Fiche 4 : Activité 3 : Mise en place de rapports

1. Compétences mises en œuvre
2. Cahier des charges
3. Démarche / mode opératoire
4. Preuve de la réalisation

IV. Fiche 4 : Activité 4: Déploiement de batchs

1. Compétences mises en œuvre
2. Cahier des charges
3. Démarche / mode opératoire
4. Preuve de la réalisation

IV. Fiche 4 : Activité 5: Maquettage de solution pour les appels d’api

1. Compétences mises en œuvre
2. Cahier des charges
3. Démarche / mode opératoire
4. Preuve de la réalisation

V. Retex

5. Remerciements
6. Points positifs
7. Pistes de progrès

1. Présentation de la structure d’accueil

1. Présentation générale

CEGEDIM est une société anonyme (SA) française fondée en 1969, dont le siège social se situe au 129–137 rue d’Aguessau à Boulogne-Billancourt. Spécialisée dans les technologies de l’information, elle propose des solutions pour les secteurs de la santé, des assurances et de la gestion administrative. Initialement créée pour répondre aux besoins informatiques des laboratoires pharmaceutiques, l’entreprise s’est progressivement développée à l’international et a diversifié ses services, notamment dans la dématérialisation, le tiers payant et la gestion de la relation client. Cegedim emploie plusieurs milliers de collaborateurs, est implantée dans plus de 10 pays comme la Belgique, le Maroc ou encore le Royaume-Uni.

Cegedim Assurances est la branche spécialisée du groupe Cegedim dédiée aux solutions logicielles et services pour le secteur de l’assurance de personnes, notamment la santé, la prévoyance, les mutuelles et les courtiers. Elle propose une gamme complète de produits, allant des progiciels de gestion, aux plateformes telles que BEYOND, destinées à répondre aux besoins de digitalisation et d’efficacité des assureurs.

Elle attache une grande importance à la sécurité des données, validée par plusieurs certifications ISO, garantissant la confidentialité et la continuité des services. Implantée dans plusieurs villes françaises dont Boulogne-Billancourt (siège), Nantes, Lyon, Toulouse (Labège) et à l’international (Maroc, Royaume-Uni), Cegedim Assurances joue un rôle clé dans la transformation digitale du secteur de l’assurance santé.

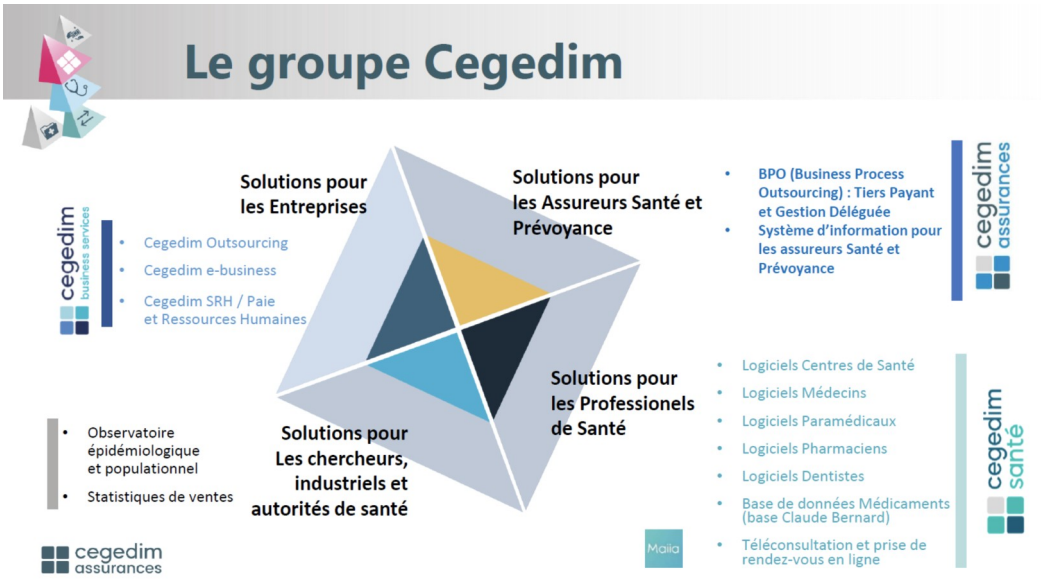
Cegedim Assurances combine technologie, expertise métier et innovation pour offrir aux acteurs de l’assurance des solutions adaptées à leurs enjeux actuels et futurs.

2. Cœur du métier

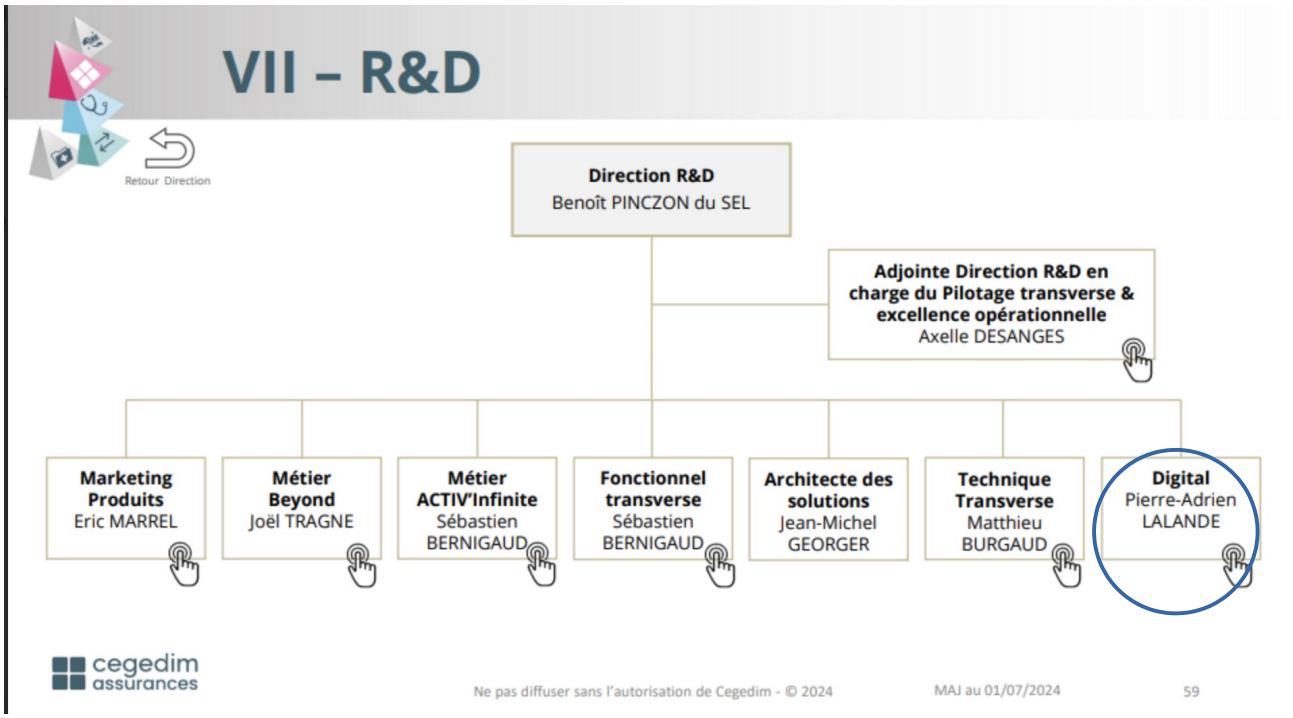
Cegedim Assurances intervient principalement dans le secteur de l’assurance de personnes, notamment la santé, la prévoyance, et les mutuelles. Son activité se concentre sur la fourniture de solutions informatiques et de services dédiés à la digitalisation et à l’optimisation des processus métiers des acteurs de l’assurance. Elle commercialise des progiciels comme Activ’Infinite, leader dans la gestion des contrats santé et prévoyance, ainsi que des plateformes SaaS innovantes telles que Beyond by Cegedim, qui centralisent la gestion des prestations et la relation avec les assurés. L’entreprise propose également des services d’externalisation (BPO) avec iGestion, des solutions spécialisées pour le tiers payant comme iSanté et SP Santé (CETIP). Catherine ABIVEN est actuellement la présidente de Cegedim Assurances.

3. Organisation de la structure

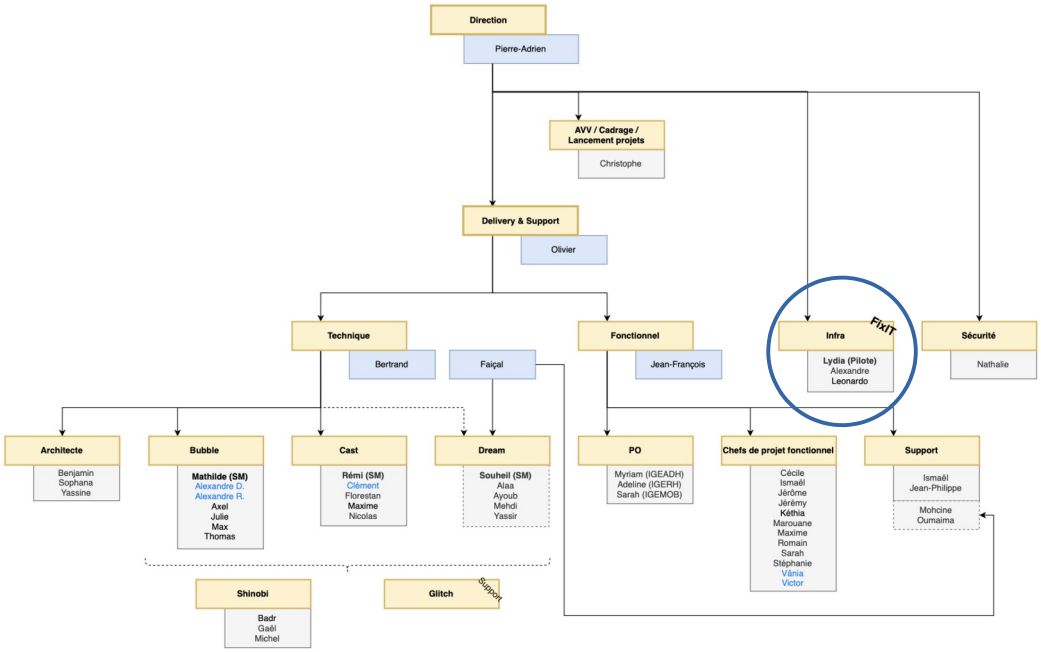
J’ai été durant mon stage intégré au pôle-digital de CEGEDIM Assurances, qui est une branche de CEGEDIM. Voici un schéma explicatif des différentes branches du groupe CEGEDIM:



Voici la place du pole digital au sein de la branche CEGEDIM Assurances :



Voici un organigramme des différentes teams au sein du Pole-digital :



4. Chiffres clés

En 2024, Cegedim Assurances compte environ 1 700 collaborateurs répartis sur 12 sites, situés principalement en France, notamment à Boulogne-Billancourt, Labège, Lyon, et Nantes ainsi qu'à l'international avec des implantations au Royaume-Uni, au Maroc (Rabat) et en Roumanie.

Ces équipes interviennent principalement dans le développement de solutions logicielles et le support aux clients, du secteur de l'assurance de personnes.

Le chiffre d'affaires en 2024 de la branche s'élève à environ 71,5 millions d'euros, représentant près de 11 % du chiffre global du groupe Cegedim (654,5 millions d'euros).

Cegedim Assurances possède 3 data center en France, permettant ainsi un support technique constant et rigoureux. L'entreprise veille aussi à la formation de ses salariés avec une plateforme de e-learning spécialement dédiée, contenant plus de 94 formations telles que des formations obligatoires, comme celle sur la non divulgation des données personnelles (en conformité du RGPD) ou encore des formations de sensibilisation face à l'IA.

2. Présentation du contexte du stage

1. Maitre de stage

Au sein de Cegedim Assurances, à l'intérieur du Pole Digital, Lydia MOURET a été ma maitre de stage durant toute la durée de celui-ci. Lydia fait partie de l'entreprise depuis un peu moins d'un an, en tant qu'Ingénieur DevOps. Evoluant dans la team Fix-It, elle est principalement en charge de la vérification du bon déroulement des batchs ainsi que de leurs déploiements, elle s'occupe aussi des bascules SMTP, de tout ce qui est vérification dans les logs, de l'activation d'options pour les clients, …

2. Positionnement dans l’organisation

Au sein de Cegedim Assurances, j’ai été rattaché au Pole Digital (*voir organigramme*), dans la team Fix-It avec MOURET Lydia comme tutrice. Je m’occupais du bon fonctionnement et du bon déploiement des batchs. Pour ce faire, j’ai dû réaliser plusieurs missions.

3. Mission réalisées

Le sujet de mon stage était la mise en place d’une solution de supervision sur les batchs de production et sur ceux à venir.

Afin de réaliser cette mission, j’ai d’abord fait un recensement des batchs (fiche 4, activité 1), puis j’ai commencé à mettre en place des alertes tout au long de mon stage en les perfectionnant (fiche 4, activité 2). Une fois les alertes en place j’ai mis en marche des tableaux de statistiques des erreurs dans les lignes de logs (fiche 4, activité 3). Tout le long du stage, j’ai participé au déploiement de batchs d’abord en recette puis en production (fiche 4, activité 4). Et enfin durant mon stage, pour les batchs qui faisaient simplement un appel d’api directement sur l’url du site, j’ai commencé à réfléchir à la mise en place d’une solution de surveillance (fiche 4, activité 5).

J’ai aussi réalisé une documentation sur toutes les manipulations que je faisais pour la mise en place d’alertes, afin de faciliter la tâche à tous ceux qui voudraient mettre en place une alerte ou des dashboards ainsi que d’autres annexes.

Un batch (ou script batch) est un fichier texte contenant une suite de commandes destinées à être exécutées automatiquement via le task-scheduler. Il est utilisé pour automatiser des tâches répétitives sans avoir à les saisir manuellement à chaque fois. Un fichier batch permet ainsi de gagner du temps ainsi que de simplifier des opérations complexes. Dans le cadre de mon stage on utilisait par exemple, les batchs pour automatiser l’envoie de mail ou de notifications mobiles lors de remboursement de prestations. Il existe de nombreux batchs qui se distinguent selon leur catégorie (envoie, import, export, purge et taches-recurentes), cette classification rend la maintenance beaucoup plus simple et pratique.

Parfois les batchs employés étaient dans des .bat (ils font un appel d’un fichier executable .exe du répertoire où il se situe pour une url de client). D’autres fois les batchs étaient simplement des .ps1 (script power-shell) qui faisaient des appels d’api sur une url donc il n’y avait aucun fichier de configuration en local sauf le script.

Ces batchs renvoyaient des fichiers de logs qui contenaient l’activité de ces derniers (les lignes étaient ordonnées de manière structurée : heure / date / statut / activité) dans un répertoire de logs (je modifiais le répertoire dans le .config du batch pour qu’il corresponde bien à celui que j’avais créé ou qui était déjà en place).

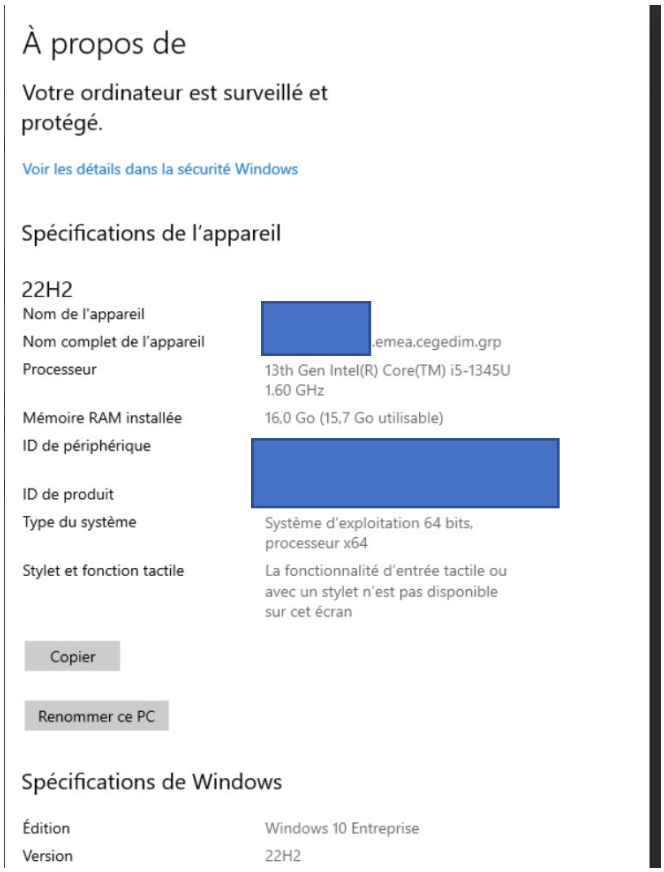
J’ai dû déployer des batchs, les paramétrer pour qu’ils marchent bien et qu’ils correspondent bien aux exigences du chefs de projects / clients. J’ai dû aussi en désactiver et en supprimer.

Chaque changement de statut d’un batch devait être remonté au chef de projet en charge du client pour être sûr (des points étaient organisés régulièrement pour statuer certains batchs, pour connaître leurs fonctionnements etc. avec les différents développeurs et chefs de projets).

3. Environnement technique

1. Ressources matérielles

Pour les ressources matérielles l'entreprise m'a fourni un ordinateur portable dont les caractéristiques techniques sont les suivantes :



Elle m'a également fourni un casque audio ainsi qu'un clavier. Quand j'étais au bureau j'allais sur les bureaux non occupés (vacances, congés longue durée, etc.) et je pouvais ainsi bénéficier de 2 écrans supplémentaires.

Je n'ai actuellement pas de schéma réseau (je n'y ai pas eu accès durant mon stage, j'ai fait une demande complémentaire j'en aurais peut-être un ultérieurement)

2. Ressources logicielles

Pour réaliser ce qui m'était demandé durant mon stage j'ai utilisé différents logiciels qui sont les suivants :

Grafana :

Grafana, est un logiciel (ici installé sur les différents serveurs BDD ou d'ordo) qui permet de surveiller en temps réel l'état et les performances des serveurs grâce à des tableaux de bord

dynamiques (avec des graphiques principalement). Il permet ainsi d'afficher des métriques essentielles telles que l'utilisation du CPU, de la mémoire, du disque ou encore l'activité réseau.

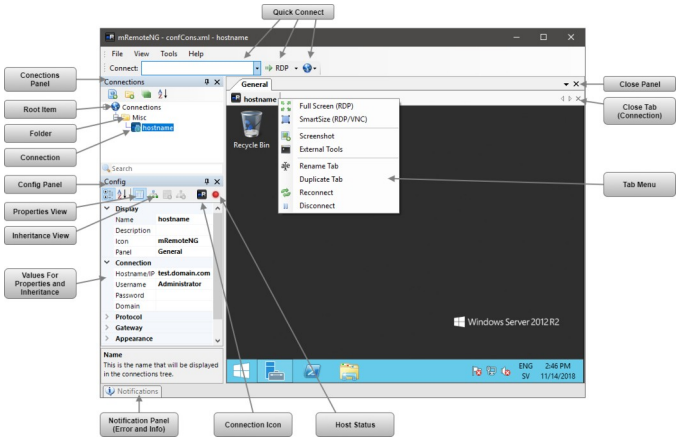
Il permet aussi de conserver un historique de ces données, ce qui facilite l'analyse des performances passées et l'identification des causes d'éventuels incidents.

Exemple de rendu (photo non contractuelle)

mRemoteNG :

mRemoteNG est un logiciel open source permettant de gérer et d'organiser de multiples connexions à distance dans une interface centralisée. Il est principalement utilisé pour accéder rapidement à différents serveurs ou desktop (en l'occurrence d'abord au Bastion via Ivanti puis aux différents serveurs).

Grâce à son interface en onglets, mRemoteNG offre une gestion claire et efficace de plusieurs sessions ouvertes simultanément, ce qui facilite l'accès à plusieurs machines au même moment. L'outil permet également de stocker les informations de connexion de manière sécurisée, avec la possibilité d'organiser les connexions en dossiers hiérarchiques.



Confluence :

Confluence est un logiciel collaboratif développé par Atlassian, conçu pour faciliter la création, le partage et la gestion de la documentation au sein d'une équipe ou d'une organisation. Il permet de centraliser les connaissances sous forme de pages organisées dans des espaces de travail, ce qui le rend idéal pour la rédaction de procédures internes, de comptes rendus, de cahiers des charges ou de notes de projet. Grâce à son éditeur riche et intuitif, les utilisateurs peuvent facilement créer du contenu structuré, insérer des tableaux, des images, des liens ou encore intégrer d'autres outils Atlassian comme Jira. Confluence favorise également la collaboration en temps réel, en permettant à plusieurs personnes d'éditer une même page, de laisser des commentaires ou d'assigner des tâches. Il offre des fonctionnalités de gestion des droits d'accès, d'historique de modifications et de recherche avancée, ce qui en fait un véritable wiki d'entreprise.

Zoom :

Zoom est une messagerie, intégrée à la plateforme de visioconférence du même nom. C'est un outil de communication professionnelle en temps réel qui va bien au-delà des simples appels vidéo. Elle permet notamment aux collaborateurs d'échanger par messages instantanés, de partager des fichiers, de créer des canaux de discussion (j'ai créé un canal zoom pour recevoir les alertes Splunk directement dans ce canal) et de maintenir une communication fluide au sein de l'entreprise.

Son principal atout réside dans sa capacité à unifier la communication, en facilitant la transition entre un simple message, un appel vocal ou une réunion vidéo, sans avoir à changer de plateforme.

Elle permet aussi une historisation des discussions, d'organiser les conversations par fils thématiques, de définir des statuts de disponibilité, et de recevoir des notifications adaptées au contexte de travail. Pour les collaborateurs à l'étranger ou en télétravail, cela améliore grandement la réactivité, la coordination et la productivité entre les équipes.

Pour moi Zoom est la messagerie par excellence : sur les canaux il y a de nombreuses features très pratiques comme l'attribution d'un mail à un canal permettant ainsi une communication centralisée. On peut ajouter des onglets ou juste en cliquant dessus ça nous renvoie vers une url, ce qui permet d'avoir tout à portée de mains. Il y a pleins d'autres options disponibles

comme la création de mind-map ou encore des schémas. Cette messagerie rempli tous les besoins nécessaires à une bonne collaboration entre les équipes.

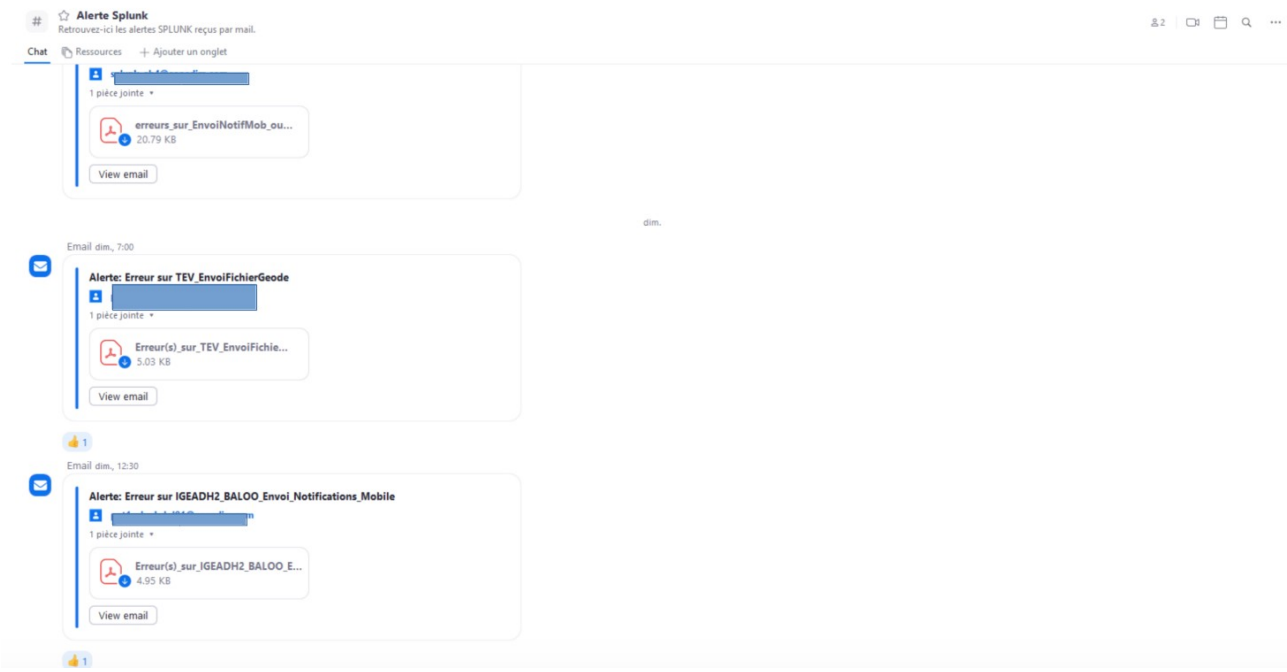


Photo soumise aux restrictions des droits d'auteurs.

Splunk :

Splunk est une plateforme logicielle utilisée pour la collecte, l'analyse et la visualisation des données machine générées par les systèmes informatiques, les applications, les serveurs et les équipements réseau. Elle permet d'exploiter en temps réel de vastes volumes de données, notamment les logs, les métriques, les événements et les traces, afin de surveiller la performance, détecter des anomalies, analyser des incidents de sécurité ou encore répondre à des exigences de conformité.

Au cœur de son architecture se trouve le Splunk Forwarder, un composant léger installé sur les machines sources. Ce forwarder agit comme un agent chargé de collecter et d'envoyer les données (principalement les logs système, applicatifs ou personnalisés) vers un serveur Splunk (appelé indexeur), où elles sont stockées, indexées puis visualisées via des tableaux de bord dynamiques.

Le forwarder surveille des fichiers de log définis (dans le cadre de mon stage on avait un fichier .conf à remplir sur git où il fallait rentrer les chemins des fichiers de logs pour pouvoir ensuite y accéder via Splunk), et les envoie en continu ou selon une planification.

Une fois indexées sur git, les données sont facilement consultables via le langage de requête SPL (Search Processing Language) de Splunk, qui permet de rechercher, filtrer, corréler et visualiser les informations.

Splunk permet l'envoie d'alerte sur un mail (donc un canal zoom), sur un téléphone en notif mobile (avec l'app), faire tourner un script ou encore faire un curl d'une url.

Grâce à ce mécanisme de collecte distribuée, Splunk a été le logiciel parfait pour mettre en place les alertes dont nous avions besoins, quotidiennement.

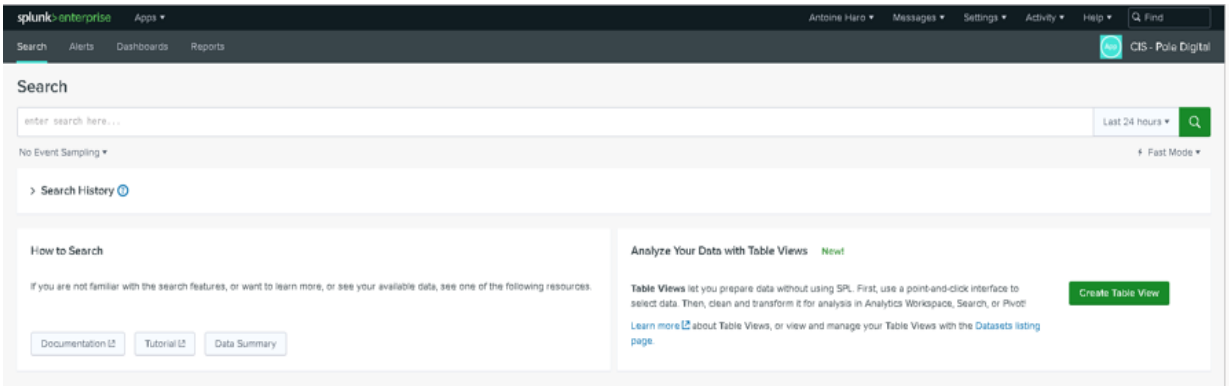


Photo soumise aux restrictions des droits d'auteurs.

Park-digital :

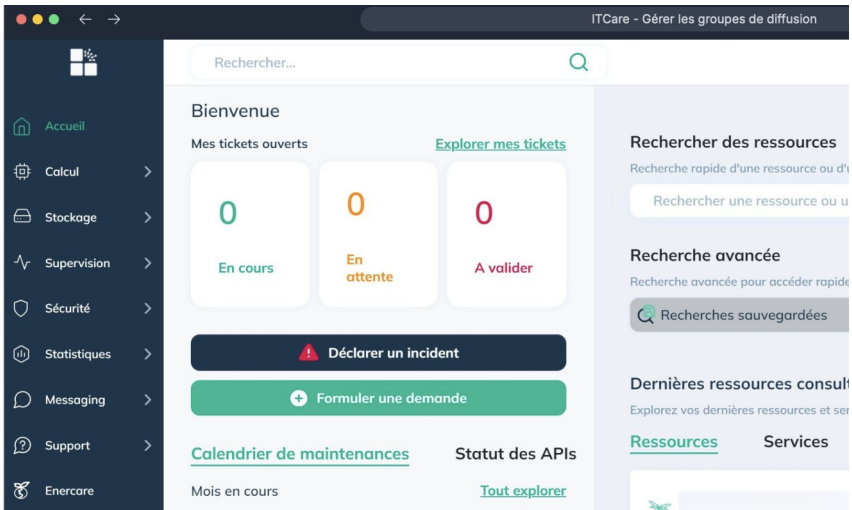
Park-Digital est un outil développé par des membres de CEGEDIM Assurances, accessible depuis un moteur de recherche, qui permet de trouver les serveurs qu'occupe un client ou son service via le nom du client ou son url. Facilitant ainsi la localisation des services lors de déploiement ou de maintenance.

ItCare :

ItCare est un logiciel qui permet de faire des tickets, semblables au logiciel GLPI. Le logiciel permet donc de faire des demandes aux autres services. Il m'a été particulièrement utile pour demander les push des commits que je faisais sur le git du Pole-Digital, aux services qui avaient plus de droit que moi pour faire cela. On pouvait sélectionner lors de la création de ticket : le service visé, le pôle visé et également le serveur sur lequel la modif devait être faite. Ensuite on rédigeait le message et on envoyait. Selon le service, le traitement était plus ou moins rapide.

Les tickets avaient alors 3 statuts : « En cours », « En attente » et « à valider ». Ce dernier étant obligatoire et devait être fait par l'émetteur pour clore le ticket (avec un commentaire obligatoirement).

Ce logiciel est intéressant car il permet une historisation des tickets, et une très bonne supervision de ceux-ci ; chaque tickets est attribué à un service, et chaque personne du service est directement joignable via Zoom (voir explication logiciel).



Par exemple si un ticket est bloqué sur un statut « En attente » on voit le nom et prénom de la personne en charge et on peut alors lui envoyer un message si besoin, pour demander des informations ou autre. L'historisation permet de garder une trace des tickets déjà faits (avec toutes les informations pas juste le numéro).

Jira :

Jira est un logiciel de gestion de projet développé par Atlassian (même créateur que Confluence donc il a des similitudes au niveau de l'interface), largement utilisé dans les environnements agiles, notamment par les équipes de développement logiciel, mais également adaptable à d'autres secteurs.

Jira permet de suivre, planifier et organiser le travail sous forme de tickets, appelés "issues", qui peuvent représenter des tâches, des bugs, des fonctionnalités à développer ou toute autre unité de travail. L'une des fonctionnalités clés de Jira est son système d'attribution de tâches, qui permet d'assigner chaque ticket à un membre spécifique de l'équipe. Cette attribution s'accompagne généralement d'un statut et peut être intégrée à un workflow personnalisé ou plusieurs collaborateurs peuvent se voir attribuer une même tache.

Ainsi, chaque collaborateur connaît précisément ses responsabilités et les échéances associées, ce qui améliore la clarté, la traçabilité et la répartition du travail. Jira offre une visibilité en temps réel sur l'avancement des projets. L'assignation des tâches peut aussi être automatisée ou conditionnée selon des règles définies, facilitant la gestion des priorités, la répartition de la charge de travail et le respect des délais.

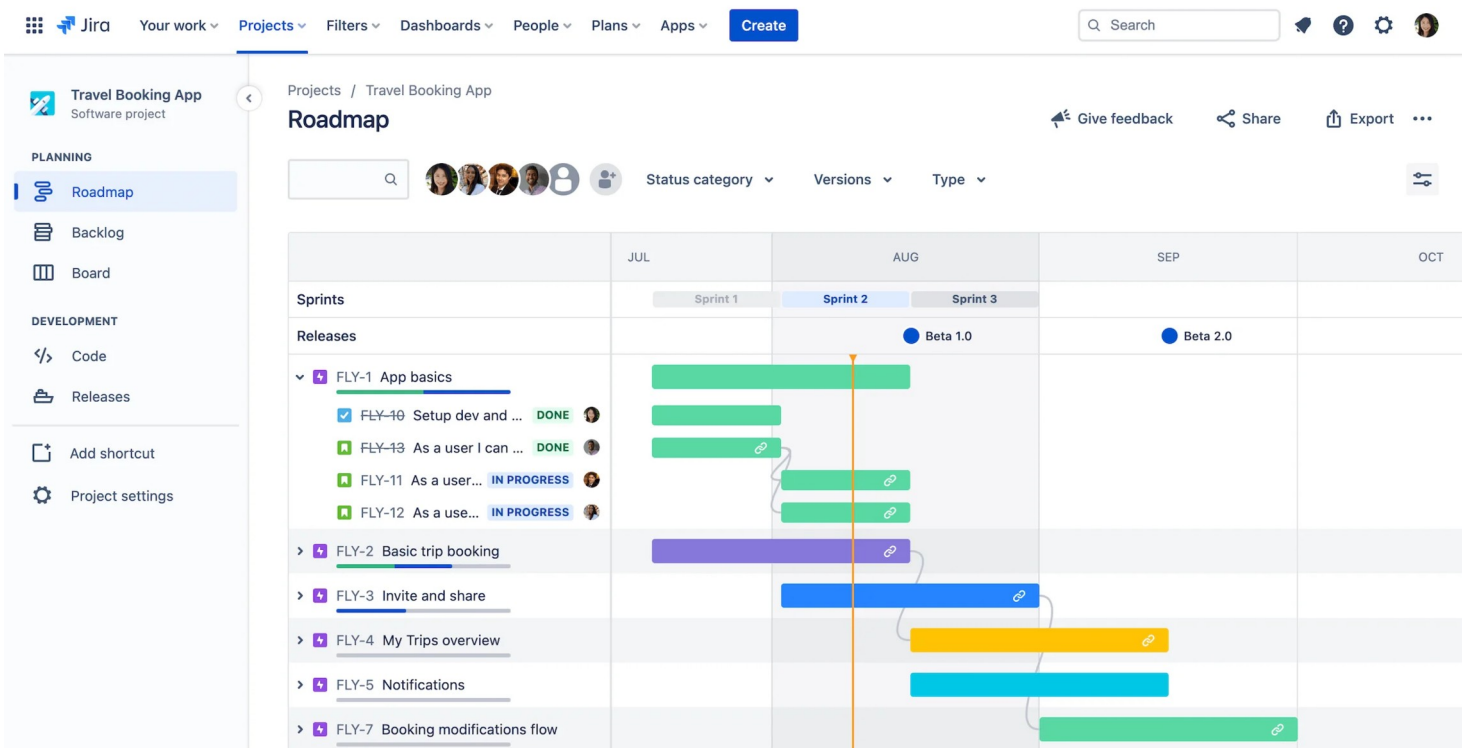


Photo non contractuelle de l'interface JIRA

Activité 1 : Recensement des batchs

Compétences mises en œuvre

Gérer le patrimoine informatique :

- Mettre en place et vérifier les niveaux d'habilitation associés à un service
- Vérifier les conditions de la continuité d'un service informatique
- Recenser et identifier les ressources numériques

Cahier des charges

On a besoin que le tableau excel qui recense les batchs de production soit régulièrement mis à jour. Souvent après des déploiements, des suppressions ou des désactivations.

Ce tableau contient : **le serveur** sur lequel se situe le batch, **le nom** du batch, **sa catégorie** (envoie, import, export, purge, tache récurrente), **sa date de création**, **nom de l'admin** qui l'a déployé, **sa date d'exécution**, **son répertoire d'exécution**, **son répertoire de logs** ainsi que **son état** (désactivé, inactif, prêt).

Démarche / mode opératoire

Pour ce faire j'ai donc commencé par regarder dans le task-scheduler les batchs déjà en place sur les 2 serveurs administrateurs, et j'ai mis à jour tous les statuts, j'ai vérifié les fichiers de logs pour savoir si ça correspondait bien, j'ai aussi ajouté les batchs qui ne figuraient pas dans le tableau.

A chaque déploiement, suppression ou encore changement de statuts, répertoire, etc. je mettais le fichier à jour pour éviter les confusions.

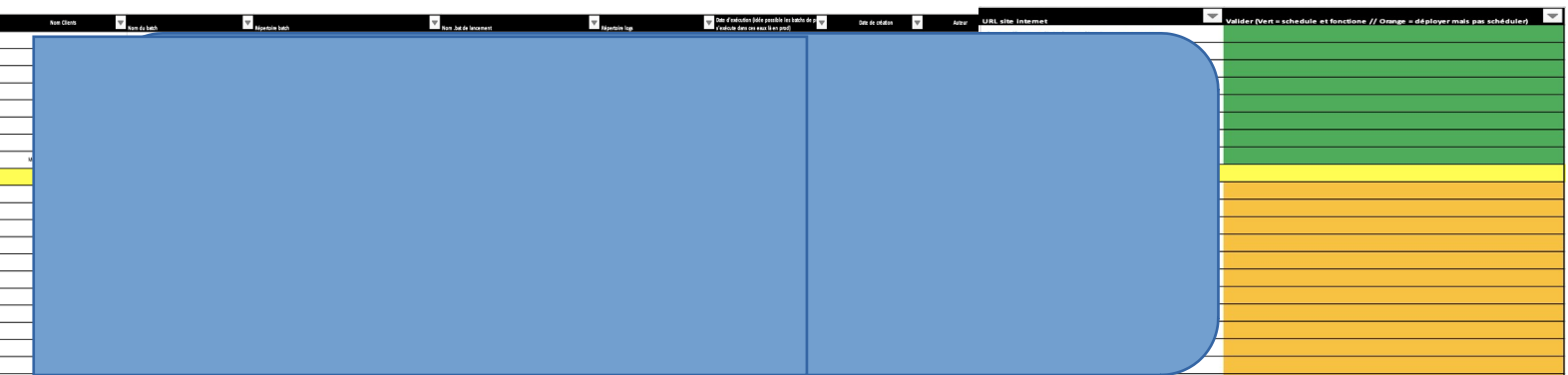
Le tableau excel était disponible via le portail cegedim sous format .csv où étaient partagés différentes documentations. Il était ainsi accessible et modifiable par celui qui y avait accès (je gardais une save locale que je préférais push à chaque fois).

Cette activité m'a donc suivi jusqu'au dernier jour du stage.

Preuve de la réalisation

Voici un exemple d'en-tête d'un de mes fichiers .xls que j'avais fait pour le déploiement de plusieurs batchs de purge en recette client en vue d'une mep.

Les batchs de purge permettaient la purge des historiques des actions utilisateurs ou encore la purge d'adhérents directement (contrat résilié depuis tant de temps, …).



Activité 2 : Mise en place d’alertes

Compétences mises en œuvre

Gérer le patrimoine informatique :
- Recenser et identifier les ressources numériques
- Mettre en place et vérifier les niveaux d'habilitation associés à un service
- Vérifier les conditions de la continuité d'un service informatique
- Gérer des sauvegardes

Cahier des charges

On a besoin de savoir rapidement si les batchs sont en erreurs, s’il y a eu un dysfonctionnement ou autre qui aurait pu engendrer des problèmes sur celui-ci et on veut avoir toutes ces informations centralisées à un endroit et qu’on puisse y avoir accès : à la ligne d’erreur, sur quel serveur l’erreur s’est produite, la date, ainsi que le batch en erreur accompagné du nom du client.

Démarche / mode opératoire

A mon arrivée, j’ai eu un briefing sur les logiciels de l’entreprise et plus particulièrement sur celui qui allait m’être utile : Splunk (voir ressources logicielles pour la présentation). Ce logiciel m’a été essentiel car il permet d’avoir accès aux fichiers de logs via la plateforme, et de pouvoir ainsi lire les fichiers directement sur le logiciel et les trier selon mes besoins, sans avoir à aller sur chaque serveur, dans chaque fichier de logs, faire ctrl + f « erreur ». Et surtout ce logiciel me permet de faire des alertes ; une alerte permet le déclenchement d’une action, je me suis adapté pour que concrètement l’alerte m’envoie un mail lorsque le serveur splunk joue ma requête qui est en langage SPL (une requête préparée et enregistrée) à une heure définie (par moi). Dès lors, il a un résultat : il envoie un mail sur le groupe Zoom que j’ai créé avec un .pdf téléchargeable ou ouvrable avec un simple clic dessus. Celui-ci contient : le serveur où se situent les fichiers de logs du batch, la date, le nom du batch, le client et surtout la ligne de logs en erreur complète.

Je vais donc tout le long de mon stage perfectionner ces requêtes et les modifier quotidiennement pour obtenir une version simple et facile, pour que tout le monde puisse prendre exemple. J’arrive donc à la fin de mon stage avec des alertes perfectionnées. On a un git commun au pôle digital qui permet d’administrer le serveur splunk pour collectiviser nos actions pour qu’elles soient publiques (alertes, dashboards, déploiement chemin de logs pour que splunk les lise, …).

La première étape consiste à enregistrer l’alerte sur un fichier de conf sur git (savedsearch.conf) qui permet une sauvegarde de toutes les alertes, elles seront publiques à tout le pôle-digital et seront consultables via Splunk ou git.

Mon alerte consiste à regarder sur tous les chemins de logs disponibles dans son git, ensuite elle regarde s’il y a une erreur. Si le mot/statu « error » apparaît alors elle regarde dans un fichier .csv qui contient le chemin de logs et le client pour savoir s’il y a une correspondance avec le chemin de logs. S’il y a une correspondance alors elle envoie un mail (défini dans l’alerte) contenant le fameux pdf où il y a la nom du serveur, le chemin de logs, le nom du batch et le client. Je fais remonter le nom du client grâce au fichier .csv sur lequel il y a la correspondance avec le chemin de logs et dès lors, il renvoie la ligne associée.

Au départ mes prototypes étaient déployés directement sur Splunk sauf qu'une problématique est apparue : quand je faisais mes alertes via l'interface graphique mes erreurs étaient privées (lors de mon départ mon compte allait être supprimé donc les alertes disparaitraient) et quand je les passais en public elles se supprimaient, car à chaque déploiement .git ça supprimait toutes celles qui n'étaient pas enregistrés sur le fichiers de configuration git. J'ai donc déployé mes alertes sur git et je demandais des deploy régulièrement pour ainsi modifier et corriger mes alertes.

Preuve de la réalisation

```
index=portail_digital "error"
| eval fichier=replace(source, ".*/", "")
| lookup chemin_logs_APV.csv chemin_logs AS fichier OUTPUT nom_du_batch
| where isnotnull(nom_du_batch)
| table host fichier nom_du_batch _raw
```

Voici la requête finale que joue le forwarder Splunk à une heure définie. Explication ligne par ligne :

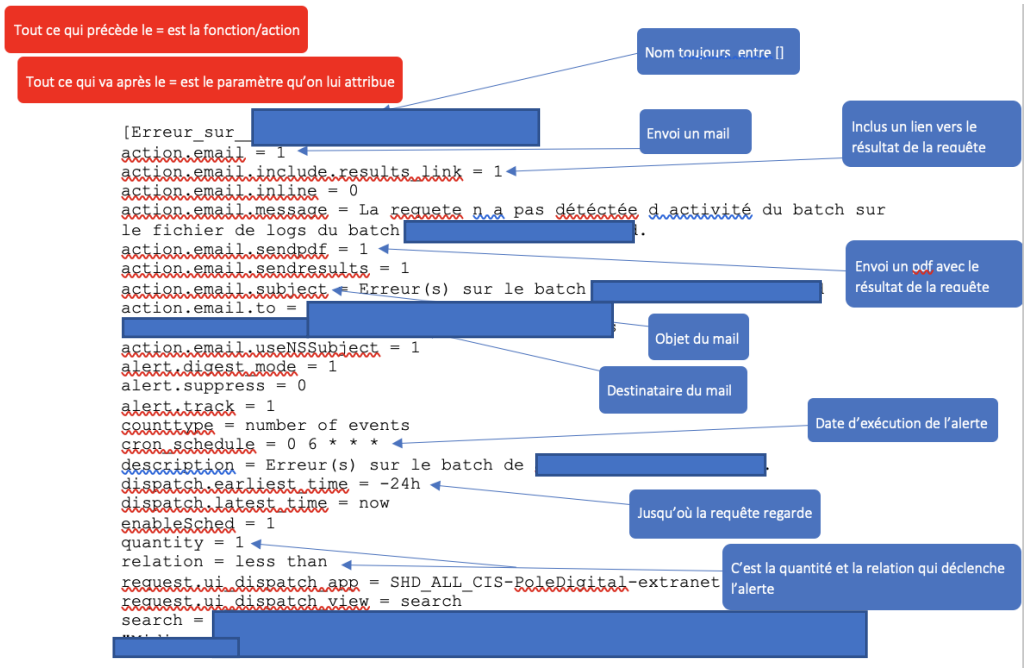
L1 = cherche dans tout le portail_digital le mot « error »

L2 = elle va regarder la correspondance avec les fichiers .csv en simplifiant le source (pour éviter les erreurs)

L3 = elle regarde les fichiers .csv et regarde s'il y a une correspondance avec la colonne chemin_logs et elle sort aussi par la même occasion la colonne nom_du_batch (j'ai fait ça pour simplifier la création du .pdf

L4 = elle regarde bien que nom du batch existe bien et est bien associé au chemin de logs

L5 = elle trie ça sous forme de tableau où apparaît : le serveur (host), le chemin de logs (fichier), le nom du batch (nom_du_batch) ainsi que la ligne en erreur (_raw).



Et voici l'alerte au format .git. La version finale a été raccourcie et simplifiée.

Activité 3 : Mise en place de rapports

Compétences mises en œuvre

Gérer le patrimoine informatique :
- Recenser et identifier les ressources numériques
- Mettre en place et vérifier les niveaux d'habilitation associés à un service
- Vérifier les conditions de la continuité d'un service informatique
- Gérer des sauvegardes

Cahier des charges

On souhaitait avoir une estimation rapide du nombre de lignes d'erreurs sur les fichiers de logs, pour ainsi statuer les problèmes. La meilleure façon de voir rapidement le nombre d'erreurs c'est de les présenter via des graphiques.

Démarche / mode opératoire

Ainsi, je me suis lancé dans la création de tableau dynamiques : je jouais des requêtes sur Splunk qui me permettaient de faire des calculs, et après, via l'interface, je choisissais une visualisation sous forme de graphique (diagramme bâton, courbe, etc.).

Une fois que j'avais mon graphique je l'enregistrais sur un dashboard Splunk que j'avais créé au préalable sur l'interface. Ensuite je lui attribuais un nom et une description, et je les triais par sources.

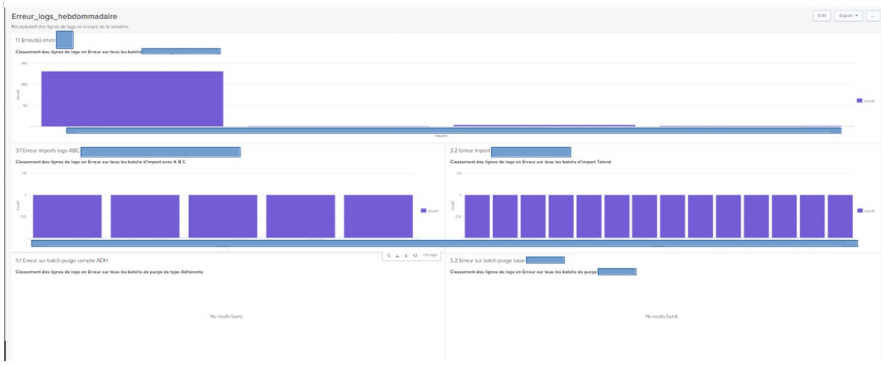
J'ai dû faire face au même problème de déploiement, du coup j'ai dû créer un répertoire spécial sur git et y enregistrer mon code pour éviter qu'il ne se fasse supprimer et que mes dashboards disparaissent. J'ai créé 2 dashboards : un qui regarde les erreurs sur la semaine passée (je l'ai fait s'envoyer tous les lundis sur le canal Zoom sous format de .pdf) et un qui regarde les dernières 24h, sachant que les deux sont consultables directement via l'interface en ligne.

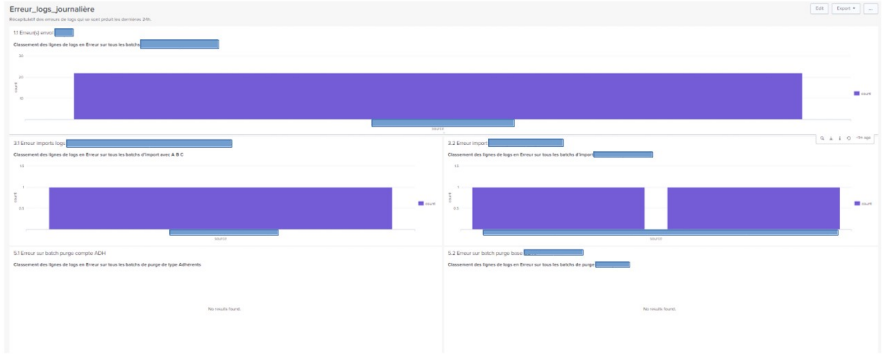
Preuve de la réalisation

Voici une partie du code .xml d'un de mes dashboards : la requête étant recouverte du cadre bleu (une requête correspond à un graphique) on peut voir la balise stats qui permet de faire des calculs.

```
<dashboard version="1.1" theme="light">
  <label>Erreur_logs_hebdommadaire</label>
  <description>Récapitulatif des lignes de logs en erreurs de la semaine.</description>
  <row>
    <panel>
      <title>1.1 Erreur(s) envoi APV</title>
      <chart>
        <title>Classement des lignes de logs en Erreur sur tous les batchs [redacted]</title>
        <search>
          <query>index=por[redacted]ery>
          <earliest>-7d@h</earliest>
          <latest>now</latest>
          <sampleRatio>1</sampleRatio>
        </search>
        <option name="charting.axisLabelsX.majorLabelStyle.overflowMode">ellipsisNone</option>
        <option name="charting.axisLabelsX.majorLabelStyle.rotation">0</option>
        <option name="charting.axisTitleX.visibility">visible</option>
        <option name="charting.axisTitleY.visibility">visible</option>
        <option name="charting.axisTitleY2.visibility">visible</option>
        <option name="charting.axisX.abbreviation">none</option>
        <option name="charting.axisX.scale">linear</option>
        <option name="charting.axisY.abbreviation">none</option>
        <option name="charting.axisY.scale">linear</option>
        <option name="charting.axisY2.abbreviation">none</option>
        <option name="charting.axisY2.enabled">0</option>
        <option name="charting.axisY2.scale">inherit</option>
        <option name="charting.chart">column</option>
        <option name="charting.chart.bubbleMaximumSize">50</option>
        <option name="charting.chart.bubbleMinimumSize">10</option>
        <option name="charting.chart.bubbleSizeBy">area</option>
        <option name="charting.chart.nullValueMode">gaps</option>
        <option name="charting.chart.showDataLabels">none</option>
        <option name="charting.chart.sliceCollapsingThreshold">0.01</option>
        <option name="charting.chart.stackMode">default</option>
        <option name="charting.chart.style">shiny</option>
        <option name="charting.drilldown">none</option>
        <option name="charting.layout.splitSeries">0</option>
        <option name="charting.layout.splitSeries.allowIndependentYRanges">0</option>
        <option name="charting.legend.labelStyle.overflowMode">ellipsisMiddle</option>
        <option name="charting.legend.mode">standard</option>
        <option name="charting.legend.placement">right</option>
        <option name="charting.lineWidth">2</option>
        <option name="refresh.display">progressbar</option>
        <option name="trellis.enabled">0</option>
        <option name="trellis.scales.shared">1</option>
        <option name="trellis.size">medium</option>
      </chart>
    </panel>
  </row>
  <row>
    <panel>
      <title>3.1 Erreur imports logs A[redacted]</title>
      <chart>
        <title>Classement des lignes [redacted]e>
        <search>
          <query>index=portail_digital source=[redacted]</que
          <earliest>-7d@h</earliest>
          <latest>now</latest>
          <sampleRatio>1</sampleRatio>
        </search>
        <option name="charting.axisLabelsX.majorLabelStyle.overflowMode">ellipsisNone</option>
        <option name="charting.axisLabelsX.majorLabelStyle.rotation">0</option>
        <option name="charting.axisTitleX.visibility">visible</option>
        <option name="charting.axisTitleY.visibility">visible</option>
        <option name="charting.axisTitleY2.visibility">visible</option>
        <option name="charting.axisX.abbreviation">none</option>
        <option name="charting.axisX.scale">linear</option>
        <option name="charting.axisY.abbreviation">none</option>
        <option name="charting.axisY.scale">linear</option>
```

Voici le rendu des 2 dashboards via l'interface Splunk :





Activité 4 : Déploiement de batchs

Compétences mises en œuvre

Mettre à disposition des utilisateurs un service informatique :
- Réaliser les tests d'intégration et d'acceptation d'un service
- Déployer un service
- Accompagner les utilisateurs dans la mise en place d'un service

Cahier des charges

On a besoin qu'un client ait un batch d'envoi, purge ou autre, car il vient de le commander.

Démarche / mode opératoire

A partir d'un mois de stage, j'ai commencé à déployer des batchs. On déploie le batch d'abord en recette interne, puis en recette client et si tout va bien on le met en production.

Afin de déployer un batch, j'allais récupérer en premier l'artefact git du batch que le client souhaitait. Puis je le déplaçais sur le serveur d'ordonnancement où j'avais au préalable créé un répertoire (en recette interne, puis recette client, puis en production) où installer l'artefact. Je créais par la même occasion un répertoire de logs pour celui-ci (en recette interne, puis recette client, puis en production). Ensuite je dézippais l'artéfact dans le répertoire que j'avais créé, je créais un .bat qui spécifiait l'environnement sur lequel il devait tourner, je précisais l'exécutable qu'il devait lancer ainsi que le domaine du client. Avant la mep j'activais aussi l'option testmode = true, pour que quand on lance le batch celui-ci n'apporte pas les modifications (on a un résultat dans les logs comme s'il le faisait vraiment mais cela ne supprime rien) cela permettait de voir directement si le fichier de logs contenait des erreurs ou pas, et c'était vraiment très pratique car en fonction on pouvait ajuster les paramètres. Avant la création du bat j'enregistrait le chemin du répertoire de logs dans les fichiers de configuration du batch, pour que quand celui-ci se lance, les logs atterrissent au bon endroit. Pour finir j'allais sur le task-scheduler, et je planifiais une nouvelle tâche en fonction de la volonté du moment d'exécution du client. Par exemple les batchs de purge s'exécutaient 1 fois par mois car ils regardaient loin derrière (jusqu'à 36 mois avant).

Preuve de la réalisation

Les batchs sont des automatisations de choses confidentielles (envoie de mail, envoie de notifications mobile de remboursement, purge des comptes adhérents, ...). Je ne peux donc pas montrer de « réelles » preuves de la réalisation. Voici cependant un schéma explicatif de la création des .bat lors de la mise en recette client d'un batch :

Activité 5 : maquettage de solution pour les appels d’api

Compétences mises en œuvres
Travailler en mode projet :
- Analyser les objectifs et les modalités d'organisation d'un projet

Mettre à disposition des utilisateurs un service informatique :
- Réaliser les tests d'intégration et d'acceptation d'un service

Cahier des charges

On a besoin de savoir si les batchs qui font simplement des appels d’api via une url tournent bien, car dans les logs on ne peut pas mettre en place de la surveillance dessus.

Démarche / mode opératoire

En milieu de stage j’ai donc réfléchis à une maquette d’un script power-shell qui regarderait le .ps1 (script du batch). On rajouterait un code erreur dans le script du batch et ce code erreur serait envoyé selon si l’appel d’api avait réussi ou non. Si le code = 1 alors c’est réussi et ça envoie un message dans un fichier de logs et s’il renvoie = 0 alors ça envoie un message d’échec dans le fichier de logs.

Sachant que si le fichier de logs n’existe pas il se crée dans le répertoire donné.

Dans un premier temps j’ai donc étudié comment faire un appel d’url. Puis j’ai regardé comment je pouvais observer si un processus était réussi ou non. Et enfin j’ai mis en relation le script de test (qui imitait celui du batch) avec le script qui supervise.

A la fin j’obtenais un fichier de logs avec : la date, le statut (error ou info) et j’avais l’information qui me permettait de savoir si l’appel de l’url par le premier script avait réussi. Je faisais des tests sur l’url de google ou sur des url faussées pour voir si le script marchait en erreur ou pas. J’ai réussi à le faire fonctionner. Ce n’était qu’une phase de tests.

Preuve de la réalisation

Je n’ai pas réussi à conserver des preuves des scripts .ps1. J’ai perdu les scripts lors de leur envoi par mail.

8. Retour d’expérience :

Remerciements

Je tiens à exprimer ma plus profonde gratitude à Lydia pour avoir été ma tutrice pendant ces sept semaines de stage dans le domaine de l’informatique et plus particulièrement sur le fonctionnement des batch. Son expertise, sa disponibilité et sa patience ont été des éléments clés de mon expérience de stage. Je lui témoigne toute ma reconnaissance pour tout ce qu’elle a pu m’apprendre.

Je tiens également à remercier chaleureusement Nathalie qui s'occupe de la sécurité au sein du pôle-digital. On a pu travailler ensemble à de nombreuses reprises dans le cadre de déploiement de batch de purge notamment (en rapport avec le RGPD).

Je tenais aussi à remercier Stéphane qui m’a énormément apporté sur le logiciel Splunk. Il m’a appris énormément sur son utilisation ainsi que sur l’emploi de git (pour la sauvegarde de mon travail). Sa disponibilité et sa capacité à transmettre des connaissances m’ont permis d’apprendre beaucoup de choses.

Points positifs

Ce stage m’a énormément apporté, tant sur le plan technique que sur le plan humain. Il m’a permis de découvrir en profondeur l’univers des batchs, leur rôle essentiel dans le traitement automatisé des données, ainsi que les enjeux techniques qu’ils impliquent. J’ai ainsi pu acquérir une meilleure compréhension de leur fonctionnement, de leur planification ainsi que de leur déploiement.

Au cours de cette expérience, j’ai eu l’opportunité de collaborer avec des collègues passionnés et expérimentés. Leurs conseils et leur disponibilité m’ont grandement aidé lors de mon stage. J’ai ainsi pu progresser rapidement au sein de l’entreprise. Grâce à eux, j’ai pu me familiariser avec des logiciels comme Splunk et approfondir mes connaissances sur des logiciels de collaboration tels que Jira et Confluence (pour la transmission de connaissances).

Travailler dans le secteur des assurances m’a également permis de réfléchir à mes aspirations professionnelles. Cette immersion m’a conforté dans mon envie d’évoluer dans le domaine de la santé, toujours dans un contexte informatique. Je souhaite ainsi mettre mes compétences techniques au service d’un secteur porteur de sens, qui contribue directement au bien-être des individus.

Piste de progrès

En ce qui concerne les progrès, j’ai appris à faire plus attention surtout lors de manipulation en production, et notamment à éviter de lancer un chemin absolu avec un exécutable à la fin.

J’ai aussi appris à être plus rigoureux, particulièrement lors de la mise en recette ou en production d’un logiciel car s’il manque un paramètre ou un autre élément, le batch va mal fonctionner.

Je me suis également rendu compte de l’importance de faire attention à l’orthographe lors de l’utilisation de logiciels. En effet, s’il y a une erreur d’orthographe, une requête ne se jouera pas. J’ai ainsi appris à être plus vigilent.