



République Tunisienne

Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique

École Supérieure Privée d'ingénierie et de technologie

TEK-UP



RAPPORT DE PROJET DE FIN D'ÉTUDES

Cycle de Formation des Ingénieurs en Informatique

Réalisé par

Nader Ghabri

Analyse des données de différents modules d'un CRM d'une société de gestion de l'eau potable

Encadrant professionnel : **Madame Ellob Nada**

Ingénieur Informatique

Encadrant académique : **Monsieur Gasri Yassine**

Maître Assistant

J'autorise l'étudiant à faire le dépôt de son rapport de stage en vue d'une soutenance.

Encadrant professionnel, **Mme. Nada Ellob**

Signature et cachet

J'autorise l'étudiant à faire le dépôt de son rapport de stage en vue d'une soutenance.

Encadrant académique, **M. Yassine Gasri**

Signature

Dédicace

Je dédie ce travail à :

Mes chers parents, ma soeur, mon frère , mes amis, à mes enseignants et à toute ma famille,
sans leurs encouragements et sacrifices, rien de cela n'aurait été possible.

Toute ma reconnaissance...

Remerciements

Il est particulièrement agréable, avant de présenter cette oeuvre, d'exprimer toute ma gratitude envers les personnes qui de près ou de loin m'ont apporté leur aide inestimable lors de la réalisation de ce projet.

Je tiens tout particulièrement à remercier mes tuteurs de stage dans la société

SIGA ,

Mme . Nada Ellob

J'adresse, mes sincères remerciements à mon encadrant académique,

M. Yassine Gasri,

Je tiens aussi à remercier tous les membres du jury pour avoir bien voulu examiner et juger ce travail.

Merci.

Table des matières

Introduction générale	1
1 Cadre du projet	3
1.1 Cadre général du projet	4
1.1.1 Présentation de l'organisme d'accueil	4
1.2 Étude de l'existant	6
1.2.1 Description de l'existant	6
1.2.2 Critique de l'existant	7
1.2.3 Solution proposée	7
1.3 Méthodologie de conduite adoptée	9
1.3.1 Les intervenants dans Scrum	9
1.3.2 Les artefacts Scrum	10
1.4 Conclusion	11
2 Le sprint 0 : Phase de préparation	12
2.1 Description du domaine métier	13
2.1.1 Description des processus de la gestion des demandes et des abonnements	13
2.1.2 Description des processus de la gestion des relèves et de la facturation	13
2.1.3 Définition des termes techniques	13
2.2 Analyse des besoins	15
2.2.1 Présentation des acteurs	15
2.2.2 Spécification des besoins fonctionnels	15
2.2.3 Spécification des besoins non fonctionnels	16
2.3 Méthodologie de travail	17
2.3.1 Approche Bottom-Up de Ralph Kimball	17
2.3.2 Approche Top-Down de Bill Inmon	17
2.3.3 Approche adoptée	18
2.4 Planification de travail	18
2.4.1 Répartition des releases	18
2.4.2 Planification des sprints	19
2.5 Environnement de travail	20
2.5.1 Environnement matériel	20

2.5.2	Environnement logiciel	20
3	Release 1 : Collecte des Données et Création de Tableaux de Bord Initiaux	22
3.1	Sprint 1 :Collecte des Données	23
3.1.1	Objectifs du sprint 1	23
3.1.2	Backlog du sprint 1	23
3.1.3	Mettre en place l'infrastructure ELK Stack	24
3.1.4	Creation de pipelines	28
3.1.5	Integration de données	33
3.1.6	Tests et Validations de données	35
3.2	Sprint 2 : création de Tableaux de bord initiaux	36
3.2.1	Objectifs du sprint 2	36
3.2.2	Analyse des Besoins Utilisateur	36
3.2.3	Création des Nouveaux Tableaux de Bord sur Kibana	36
3.2.4	Sélection des Visualisations	37
3.2.5	Tests et Validation	37
4	Release 2 : Suivi des abonnements, des demandes,des relèves et des factures	38
4.1	sprint 3 :Suivi des Abonnements et des demandes	39
4.1.1	Objectifs du sprint 3	39
4.1.2	Backlog du sprint 3	39
4.1.3	Choix des indicateurs de performance	39
4.1.4	Les Tableaux de bord Abonnements et Demandes	40
4.2	sprint 4 :Suivi des Relèves et des Factures	45
4.2.1	Objectifs du sprint 4	45
4.2.2	Backlog du sprint 4	45
4.2.3	Choix des indicateurs de performance	45
4.2.4	Les Tableaux de bord relèves et factures	46
5	Release 3 : Suivi des fraudes et anomalies , Personnalisation des Tableaux de bord	53
5.1	Sprint 5 :Suivi des fraudes et anomalies	54
5.1.1	Objectifs du sprint 5	54
5.1.2	Backlog du sprint 5	54
5.1.3	Les anomalies relèves et les fraudes de factures	54
5.1.4	Choix des indicateurs de performance	55

5.1.5	Les Tableaux de bord Anomalies et Fraudes	55
5.2	Sprint 6 : Personnalisation des tableaux de bord	60
5.2.1	Objectifs du sprint 6	60
5.2.2	Backlog du sprint 6	60
5.2.3	Creation des Roles et des compte utilisateur	60
Conclusion générale		64

Table des figures

1.1	SIGA	4
1.2	Solutions SIGA	5
1.3	Organigramme SIGA	6
1.4	Cycle de vie Scrum	10
2.1	Indicateurs de performance	16
2.2	Approche Bottom-Up et Top-Down	18
2.3	Planification des sprints	19
2.4	Elasticsearch	20
2.5	Logstash	20
2.6	Kibana	21
2.7	Architecture ELK Stack	21
2.8	Oracle	21
3.1	Fichier configuration elasticsearch	24
3.2	cluster elasticsearch	25
3.3	Démarrage d'elasticsearch	25
3.4	Etat d'elasticsearch	26
3.5	Fichier configuration Logstash	27
3.6	Demarrage kibana	28
3.7	Accès à l'Interface kibana	28
3.8	Input Abonnement"	30
3.9	Input Relève"	30
3.10	Input Facture"	31
3.11	Output Abonnement"	31
3.12	Output Relève"	31
3.13	Output Facture"	31
3.14	Pipeline redondance Clinet"	32
3.15	Pipeline Anomalie et fraude Relève"	33
3.16	Pipeline fraude Facture	33
3.17	integration de données abonnement , relève et facture	34
3.18	integration de données redondance , anomalie et fraude	34

3.19	Index Management	35
3.20	creation dashboard kibana"	37
3.21	creation Visualisation kibana"	37
4.1	Accée au Dashbord Kibana	40
4.2	Visualisation	41
4.3	Dashbord Abonnement	42
4.4	Dashbord Abonnement Valide"	42
4.5	Dashbord Abonnement Resilie"	43
4.6	Dashbord Solde avance sur consommation"	43
4.7	Dashbord Demande"	44
4.8	Dashbord General	44
4.9	Accée au Dashbord Kibana	47
4.10	Visualisation nombre de relèves par date	47
4.11	Dashbord Relève	48
4.12	Dashbord Relève Valide	49
4.13	Dashbord Relève calculé	49
4.14	Dashbord consommation Total	50
4.15	Dashbord Facture	50
4.16	Dashbord Facture calculé	51
4.17	Dashbord Solde TTC	51
5.1	Visualisation fraude facture par secteur	56
5.2	Dashbord Anomalie de surconsommation	57
5.3	Dashbord Anomalie de consommation	57
5.4	Dashbord Anomalie de problème d'estimation	58
5.5	Dashbord Fraude Facture	59
5.6	Role "read_abonnements"	61
5.7	User "Chef d'exploitation"	62

Liste des tableaux

1.1	Points forts et points faibles de Tableau	8
1.2	Points forts et points faibles de Power Bi	8
1.3	Points forts et points faibles de ELK Stack	9
2.1	Répartition des releases	19
3.1	Backlog du Sprint 1	23
4.1	Backlog du Sprint 3	39
4.2	Backlog du Sprint 4	45
5.1	Backlog du Sprint 5	54
5.2	Backlog du Sprint 6	60

Introduction générale

L'eau, essentielle à l'écosystème, est utilisée par l'homme pour divers usages. Elle est au centre de l'attention des gouvernements et des utilisateurs. Pour répondre aux besoins fondamentaux et assurer la continuité des processus vitaux, il est nécessaire de fournir de manière fiable une quantité suffisante d'eau de qualité. Ainsi, l'économie de cette ressource limitée est une exigence majeure de la vie sur terre. Afin d'améliorer la gestion et la commercialisation de l'eau, les entreprises de distribution d'eau stockent de plus en plus de données sur leurs clients.

Les données sont considérées comme le "pétrole" de chaque organisation. Elles sont au cœur de la vie de chaque entreprise et font partie de ses actifs stratégiques. Exploiter efficacement la gigantesque quantité de données dispersées, non structurées et hétérogènes est essentiel pour une gestion optimale de l'eau afin de préserver cette ressource naturelle précieuse. Dans cette optique, la Business Intelligence est un outil indispensable pour analyser efficacement les données.

C'est dans ce contexte que s'inscrit notre stage de fin d'études au sein de SIGA " Analyse des données de différents modules d'un CRM d'une société de gestion de l'eau potable". Notre objectif est de créer une solution décisionnelle pour une entreprise de distribution d'eau afin d'aider les décideurs à mieux gérer les abonnements, les relevés et la facturation.

Le présent rapport sera organisé de la manière suivante :

- Le premier chapitre est intitulé «Cade du projet». Il présente l'organisme d'accueil, la méthode de conduite du projet adoptée ainsi que l'analyse et la critique de l'existant.
- Le second chapitre est intitulé «Phase de préparation». Il résume le sprint zéro correspondant a la présentation d'une analyse et une spécification des besoins, fonctionnels et non fonctionnels, réalisés à partir de l'identification des acteurs de systèeme. Il expose aussi la répartition des releases et l'architecture de l'application.
- Le troisième chapitre portera sur la Release 1 : « Colletion des données et creation de tableaux de bord initiaux», qui a consisté en la réalisation d'un Sprint 1 portant sur la collection des données. D'un Sprint 2, portant sur la creation de tableaux de bord intiaux.
- Dans un quatrième chapitre, nous présenterons la Release 2, « Suivi des abonnemnets , demandes , relèves et factures », qui comprend le Sprint 3, portant sur le suivi des abonnements et demandes , et le Sprint 4, portant sur le suivi des relèves et factures.
- Enfin, dans un cinquième et dernier chapitre, nous présenterons la Release 3, « Suivi des frudes et anomalies etPersonnalistion des tableaux de bord », comprenant le Sprint 5, portant sur le suivi des fraudes et anomalies, ainsi que le Sprint 6, portant sur la personnalistion des tableaux

de bord.

CADRE DU PROJET

Plan

1	Cadre général du projet	4
2	Étude de l'existant	6
3	Méthodologie de conduite adoptée	9
4	Conclusion	11

Introduction

Afin de faciliter la compréhension de notre projet, nous débuterons en définissant notre organisme d'accueil. Ensuite, nous procéderons à une étude de l'existant, suivie d'une description de la solution proposée. Enfin, nous présenterons la méthodologie de travail mise en place. L'objectif est de garantir une compréhension approfondie de notre projet.

1.1 Cadre général du projet

Dans le cadre de notre Diplôme d'ingénieur en Data Sciences à l'École Supérieure Privée d'ingénierie et de technologie TEK-UP, nous devons conclure nos études par un projet de fin d'études. Ce projet, d'une durée de quatre mois, a été réalisé au sein de la société SIGA (Système Informatique et Gestion Automatisée). Notre objectif principal était d'analyser les données provenant de différents modules d'un CRM d'une société de gestion de l'eau potable.

1.1.1 Présentation de l'organisme d'accueil

1.1.1.1 SIGA

SIGA, acronyme de Système Informatique et Gestion Automatisée, est une entreprise spécialisée dans l'ingénierie logicielle. Elle exerce ses activités auprès d'une clientèle variée, principalement constituée de grandes entreprises dans différents secteurs tels que les banques, les assurances, les administrations, les transports, les télécommunications, les organismes sociaux, les secteurs de la santé et l'industrie, pour n'en citer que quelques-uns.



FIGURE 1.1 : SIGA

1.1.1.2 Activité

SIGA se positionne principalement sur deux domaines d'activités stratégiques :

- L'édition des progiciels de gestion performants et évolutifs.
- La conception et la réalisation de systèmes informatiques spécifiques.

SIGA propose une gamme variée de solutions logicielles intégrées :

- **SIGA'ERP** : Intégrant les différents domaines d'une entreprise, SIGA rassemble ces derniers au sein d'une même base de données relationnelle.
- **SIGA'GTC** :SIGA assure la gestion technique et l'analyse de la situation des caisses de sécurité sociale.
- **SIGA'AM** :Facilite les démarches et procédures d'assurance maladie.
- **SIGA'CRM** :Assure la gestion commerciale et la gestion des relations clients.
- **SIGA'GMAO** :Gère la maintenance assistée par ordinateur.
- **SIGA'GPAO** :Garantit la production assistée par ordinateur.
- **SIGA'GED** :Numérise et archive les documents.
- **Business Intelligence** :Assure l'aide à la prise de décision.
- **Workflow** :Gère le travail collaboratif.



FIGURE 1.2 : Solutions SIGA

Grâce à une politique de croissance réfléchie, SIGA gère actuellement environ 150 projets. Elle a

étendu ses activités à des pays de l'Orient tels que l'Arabie Saoudite et les Émirats arabes unis, ainsi qu'à des pays africains tels que le Mali, la Côte d'Ivoire, le Tchad et la Mauritanie.

1.1.1.3 Organigramme

Nous effectuons notre stage au sein de la direction technique, plus spécifiquement au sein de la division Recherche et Développement des services.

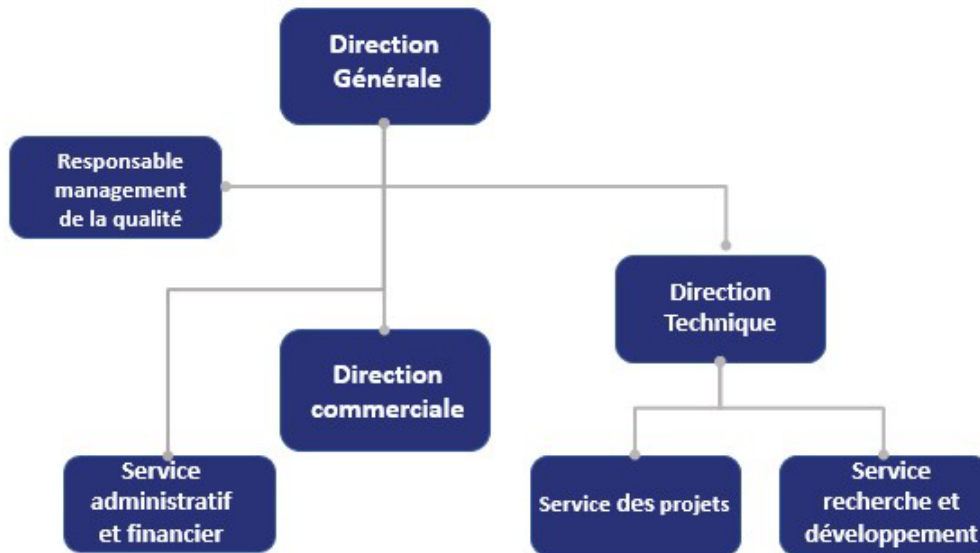


FIGURE 1.3 : Organigramme SIGA

1.2 Étude de l'existant

Afin de bien appréhender la problématique de notre cible, il est essentiel de comprendre la nature de son système actuel et, le cas échéant, d'identifier les éventuelles lacunes à corriger.

1.2.1 Description de l'existant

La société d'eau dispose actuellement d'un système de gestion commerciale et de relation client appelé SIGA'CRM. Ce système fournit les outils nécessaires pour une gestion commerciale globale, flexible et adaptée aux spécificités des activités. Ses fonctionnalités incluent la gestion de la facturation, de la relation client, des relevés, des abonnements et du recouvrement. SIGA'CRM garantit :

- Le contrôle d'accès et la sécurité des données.
- offre une administration et un paramétrage faciles, permettant de se conformer aux réglementations du pays ainsi qu'aux règles internes de l'entreprise..
- La conservation de l'information historique.

1.2.2 Critique de l'existant

En pratique, ce système ne parvient pas à répondre pleinement aux attentes des décideurs. En effet, l'entreprise fait face à des pertes commerciales résultant d'un déséquilibre dans le rendement du travail entre les différentes structures, ainsi que d'une insatisfaction des clients.

Actuellement, la société ne dispose pas des moyens concrets pour superviser son activité, suivre les réponses aux demandes des clients et détecter les anomalies pouvant survenir dans le processus de relevé. En raison de ces facteurs, les dirigeants de la société de distribution d'eau ont besoin d'un dispositif capable de leur permettre de suivre leurs services, de contrôler le flux de travail dans les différentes zones, structures et secteurs, de détecter les sources d'anomalies liées aux relevés et de superviser l'évolution de leur activité commerciale.

1.2.3 Solution proposée

En combinant SIGA'CRM avec un système décisionnel orienté vers les objectifs commerciaux, nous assurerons la flexibilité du processus de prise de décision et une vision globale des performances de l'entreprise. Notre solution consiste donc à mettre en place un système décisionnel permettant de suivre les éléments suivants :

- les demandes abonnements.
- les abonnements courants.
- les relèves.
- les anomalies des relèves.
- la gestion de facturation.

1.2.3.1 Le choix de la solution

La comparaison entre les solutions de gestion et d'analyse de données est une étape indispensable pour bien réaliser notre projet, elle permet d'examiner les caractéristiques, avantages et inconvénients des solutions existantes. Pour cela nous avons choisi d'analyser trois applications :

- Tableau.
- Power Bi.
- ELK stack.
- **Tableau :**

Tableau est une suite de logiciels de visualisation de données qui permet aux entreprises et aux professionnels de créer des tableaux de bord interactifs, des rapports et des visualisations pour

analyser et présenter leurs données. Voici quelques points clés à connaître sur Tableau :

Points forts	Points faibles
<ul style="list-style-type: none">— Facilité d'utilisation.— Intégration de données.— Adaptabilité.	<ul style="list-style-type: none">— Coût.— Courbe d'apprentissage.— Licence basée sur l'utilisateur

TABLEAU 1.1 : Points forts et points faibles de Tableau

- **Power Bi :**

Power BI est une suite de logiciels de business intelligence (BI) développée par Microsoft. Elle offre un ensemble d'outils pour l'analyse, la visualisation et le partage de données afin d'aider les entreprises à prendre des décisions éclairées. Voici quelques points clés à connaître sur Power BI :

Points forts	Points faibles
<ul style="list-style-type: none">— Intégration avec Microsoft.— Intégration de données.— Partage et collaboration.— Visualisations interactives.	<ul style="list-style-type: none">— Coût.— Courbe d'apprentissage.— Limitations dans la personnalisation.

TABLEAU 1.2 : Points forts et points faibles de Power Bi

- **ELK Stack :**

L'ELK Stack est un ensemble d'outils open source utilisés pour l'analyse, la recherche et la visualisation de données en temps réel. Le nom ELK est un acronyme pour Elasticsearch, Logstash et Kibana, les trois principaux composants de cette pile. Voici quelques points clés à connaître sur ELK stack :

Points forts	Points faibles
<ul style="list-style-type: none">— Open Source.— Extensibilité.— Traitement en temps réel.— Collecte de données diverses.— Sécurité et conformité.— Partage et collaboration.— Performance.— Adaptabilité.	<ul style="list-style-type: none">— Complexité initiale.— Maintenance.

TABLEAU 1.3 : Points forts et points faibles de ELK Stack

Après l'analyse de trois solutions , nous avons conclu que Chacune de ces solutions a ses propres avantages et inconvénients, et la meilleure option dépendra des objectifs et des contraintes du projet. Et au vu les avantages et les inconvénients de ELK stack, il sera la solution retenue dans notre projet .

1.3 Méthodologie de conduite adoptée

SIGA a adopté Scrum, une méthodologie agile qui préconise une approche itérative de la gestion de projet. Scrum assure la satisfaction du client en tenant compte des évolutions possibles du projet en cours de route.

Apparue en 1998, la méthodologie Scrum établit un cadre de travail agile efficace, principalement destiné aux entreprises de structure simple. Bien qu'initialement conçue pour le développement de projets informatiques, Scrum peut également s'appliquer à d'autres types de projets. Cette approche pragmatique consiste à décomposer le projet en unités appelées "Sprints", chacun ayant un objectif bien défini et une durée maximale d'un mois. Le client est étroitement impliqué dans le suivi de l'avancement du projet, ce qui permet d'ajuster le projet en fonction des besoins au fur et à mesure.

1.3.1 Les intervenants dans Scrum

Dans la méthodologie Scrum, il y a plusieurs rôles ou intervenants clés qui jouent des rôles spécifiques dans le processus de développement. Voici les principaux :

- **Product Owner :** Le propriétaire du produit est responsable de définir les besoins et les

fonctionnalités du produit à développer. Il priorise également le backlog du produit (la liste des tâches à réaliser) en fonction de la valeur commerciale.

- **Scrum Master** : Le maître Scrum est responsable de veiller à ce que l'équipe suive correctement les pratiques Scrum. Il aide à éliminer les obstacles qui pourraient empêcher l'équipe d'avancer et facilite les réunions Scrum.
- **Équipe de développement** : C'est l'équipe qui réalise le travail. Elle est composée de développeurs, de testeurs et d'autres experts nécessaires pour créer le produit. L'équipe est auto-organisée et responsable de la livraison des fonctionnalités à la fin de chaque sprint.

1.3.2 Les artefacts Scrum

les artefacts sont des éléments concrets et tangibles qui aident à organiser et à suivre le travail tout au long du projet. Voici les principaux artefacts Scrum :

- **Product Backlog** : C'est une liste ordonnée de toutes les fonctionnalités, les améliorations, les corrections de bogues et les tâches nécessaires pour créer et maintenir le produit. Le Product Owner est responsable de la gestion et de la priorisation du Product Backlog.
- **Sprint Backlog** : Il s'agit de la liste des éléments sélectionnés à partir du Product Backlog pour un sprint spécifique. Ces éléments sont définis et détaillés par l'équipe de développement pendant la réunion de planification de sprint.
- **Incrément** : C'est la somme de toutes les fonctionnalités et les améliorations réalisées pendant un sprint donné. À la fin de chaque sprint, l'équipe de développement doit livrer un incrément potentiellement livrable du produit.

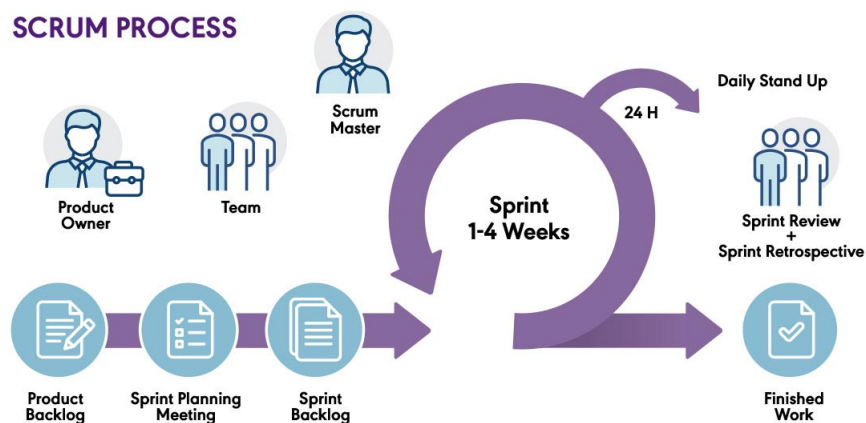


FIGURE 1.4 : Cycle de vie Scrum

1.4 Conclusion

Dans ce chapitre, nous avons exposé l'activité de SIGA. Nous avons effectué une analyse de l'existant en identifiant ses insuffisances. De plus, nous avons formulé notre propre solution pour remédier à ces points faibles. Enfin, nous avons introduit l'approche que nous allons adopter pour la gestion de notre projet. Et nous passons maintenant à la phase de préparation.

LE SPRINT 0 : PHASE DE PRÉPARATION

Plan

1	Description du domaine métier	13
2	Analyse des besoins	15
3	Méthodologie de travail	17
4	Planification de travail	18
5	Environnement de travail	20

Introduction

Pour démarrer notre projet de manière solide, nous avons initié la phase de préparation, également appelée sprint 0. Dans cette phase, nous avons travaillé à clarifier la vision globale de notre projet en passant par l'analyse des besoins, la planification du projet, l'étude des données sources et la conception globale.

2.1 Description du domaine métier

Au cours de cette section nous allons présenter les processus des différents métiers :

2.1.1 Description des processus de la gestion des demandes et des abonnements

Pour bénéficier des différents services de distribution d'eau, le client doit soumettre une demande au centre d'exploitation situé dans sa zone. Le processus de traitement de la demande et de l'abonnement implique plusieurs étapes de saisie et de vérification des données du client. Une fois que le contrat est finalisé, les agents de la société entament les travaux de branchement et confirment la validation du contrat pour le nouvel abonnement.

2.1.2 Description des processus de la gestion des relèves et de la facturation

En ce qui concerne les relevés normales, les factures sont cyclique. Cependant, pour les autres types de relevés, ils sont hors du cycle de facturation. Le processus débute par l'extraction des indicateurs des compteurs par les agents de la société. Après la validation des données par le chef du service commercial, celles-ci sont transmises à l'équipe de saisie qui les enregistre. Le chef du service commercial informe ensuite le chef de la division, qui prend en charge le calcul et la validation des factures. Enfin, le chef du service exploitation génère les factures.

2.1.3 Définition des termes techniques

- **Centre d'exploitation** : Chaque zone est composée d'un groupe de structures. Chacune de ces structures est constituée d'un ensemble de secteurs, et chaque secteur est composé de centres d'exploitation. Ces centres sont responsables de l'entretien, de la surveillance et de la maintenance de la distribution d'eau potable.
- **Type de demandes** :
 - Résiliation : Ce type implique la cessation de l'abonnement et le débranchement du compteur associé .

- Nouveau branchement : Ce type de demande permet l'obtention de trois types de contrats. En effet, un contrat normal implique un branchement simple, un contrat social forfaitaire permet de fixer un prix d'avance pour les clients nécessiteux et un contrat d'extension assure l'extension d'un branchement.
- Mutation : C'est une demande pour changer le compteur d'eau.
- Réabonnement : Lorsqu'un abonnement est suspendu, le client peut renouveler l'abonnement.
- Transfert entre les centres : Si un client souhaite changer le centre d'exploitation pour cause de déménagement, etc.

- **Type de relèves :**

- Relève normale : est une relève obtenue au cours du cycle facturation .
- Relève déposée : est une relève déposée avec le compteur lors d'une résiliation de l'abonnement.
- Releve posée : est une relève posée lors d'une résiliation de l'abonnement sans déposer le compteur.
- Releve de vérification : est une relève obtenue hors du cycle de facturation suivant une demande du client.

- **Etat du compteur :**

- Actif : Etat normal de comptage.
- Coupé : Compteur arrêté pour cause de pénalité, etc.
- Déposé : Un compteur déposé avec la relève lors de la résiliation.

- **Type de facturation :**

- Redressement : correspond à une facture envoyée si le client a dépassé le délai de paiement d'une facture.
- Mutation : correspond à une facture générée lors du changement d'un compteur d'eau.
- Débit Intercalaire : correspond à une facture payée sur plusieurs tranches.
- Pénalité : correspond à une facture générée si le client n'a pas payé plusieurs factures consécutives.
- Migration : correspond à une facture générée si le client souhaite changer d'un compte à un autre.

2.2 Analyse des besoins

2.2.1 Présentation des acteurs

- **Chef d'exploitation** : C'est le responsable du suivi des demandes clients et des abonnements.
- **Chef de division** : C'est le responsable chargé de suivre les relèves, les états des compteurs et la détection des anomalies.
- **Chef du service commercial** : C'est le responsable du processus de facturation et de paiement.
- **L'équipe de développement** : C'est l'équipe chargée de maintenir la solution informatique.

2.2.2 Spécification des besoins fonctionnels

Au sein de l'entreprise de distribution d'eau, les responsables ont la nécessité de visualiser les données afin de surveiller les demandes, les abonnements, les relevés et la facturation. Notre but est en effet de développer un système décisionnel qui garantit les capacités suivantes :

- Collecte de données en temps réel : Besoin de collecter, agréger et indexer en temps réel les données provenant de différentes sources, comme des bases de données.
- Garantir une analyse suivant les indicateurs de performance extraits suite aux réunions réalisées avec le directeur du produit.
- Création de tableaux de bord interactifs : Besoin de créer des tableaux de bord visuels interactifs et personnalisables pour surveiller les indicateurs clés de performance
- Reporting personnalisé : Possibilité de générer des rapports avec des filtres, des regroupements et des options de visualisation personnalisables.
- Exploration de données : Capacité à explorer les données de manière interactive et rapide pour découvrir des informations non évidentes et des relations cachées.

Métier	Indicateur	
Suivi de demandes et abonnements	Demande par :	<ul style="list-style-type: none"> • Année, mois • Département, zone, structure • Type de client • Type de demande • Tarification • Validité
	Abonnement par :	<ul style="list-style-type: none"> • Etat abonnement • Validité abonnement • Validité demande • Année, Mois • Type et titre de client • Tranche d'âge et Genre client • Département, zone, structure
	Solde Client par :	<ul style="list-style-type: none"> • Année, mois • Département, zone, structure • Type client • Etat abonnement
Suivi de la relève	Consommation de l'eau par :	<ul style="list-style-type: none"> • Année, mois • Zone, structure, secteur • Type, Titre de client • Tarification • Age et diamètre de compteur
	Relève par :	<ul style="list-style-type: none"> • Type d'anomalie • Zone, structure, secteur • Etat abonnement • Validité abonnement et demande • Type relève • Année, mois • Type de client
Suivi de la facturation	Evolution de chiffre d'affaire par :	<ul style="list-style-type: none"> • Année, mois • Type et titre de client • Tarification • Zone, structure, secteur
	Facturation par :	<ul style="list-style-type: none"> • Etat (impayé, annulé) • Localisation • Année, mois • Type de facture et client • Zone, structure, secteur
	Volume facturé	<ul style="list-style-type: none"> • Age compteur • Année, mois • Zone, structure, secteur

FIGURE 2.1 : Indicateurs de performance

2.2.3 Spécification des besoins non fonctionnels

- La performance : Temps de réponse rapide pour les requêtes et les visualisations. Capacité à traiter de grandes quantités de données en temps réel.

- **Fiabilité** : Disponibilité élevée pour assurer un accès continu aux données. Tolérance aux pannes pour éviter la perte de données critique.
- **Facilité d'utilisation** : Interface utilisateur conviviale pour les utilisateurs finaux.
- **Sécurité** : Protection des données sensibles. Gestion des droits d'accès pour les utilisateurs et les groupes.

2.3 Méthodologie de travail

Avant d'établir notre plan de travail, il est nécessaire de déterminer l'approche la mieux adaptée à notre projet.

Ralph Kimball et Bill Inmon sont deux figures influentes dans le domaine de la modélisation des données et de la conception de data warehouses. Ils ont développé deux approches principales pour la création de data warehouses : l'approche Bottom-Up de Ralph Kimball et l'approche Top-Down de Bill Inmon. Ces approches diffèrent principalement dans leur philosophie de conception et leur séquence de développement.

2.3.1 Approche Bottom-Up de Ralph Kimball

- **Philosophie** :

L'approche Bottom-Up, développée par Ralph Kimball, met l'accent sur la rapidité de mise en œuvre et la flexibilité. Elle favorise la création de petites structures de données appelées "data marts" dédiées à des domaines spécifiques de l'entreprise. Chaque data mart est conçu de manière indépendante pour répondre aux besoins métier immédiats.

- **Avantages** :

L'approche Bottom-Up permet une mise en œuvre rapide, une flexibilité accrue pour répondre rapidement aux besoins changeants, et elle est souvent plus adaptée aux entreprises de taille moyenne ou aux départements spécifiques.

2.3.2 Approche Top-Down de Bill Inmon

- **Philosophie** :

L'approche Top-Down, développée par Bill Inmon, est axée sur la création d'un data warehouse d'entreprise unique et centralisé. Elle privilégie la cohérence et la qualité des données. Selon Inmon, il ne devrait y avoir qu'une seule source de vérité pour toutes les données de l'entreprise.

- **Avantages** :

L'approche Top-Down offre une vue unique et cohérente des données de l'entreprise, favorise la

qualité des données et est adaptée aux grandes entreprises où la conformité aux normes et la cohérence des données sont cruciales.

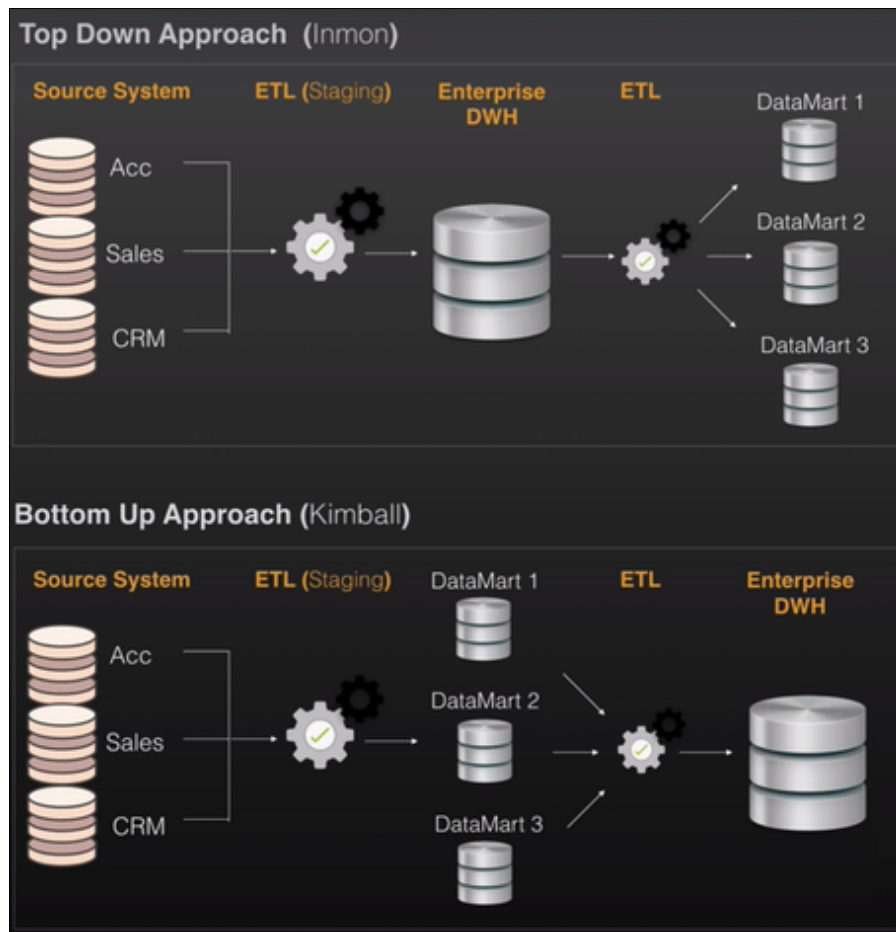


FIGURE 2.2 : Approche Bottom-Up et Top-Down

2.3.3 Approche adoptée

L'approche qui correspond le plus à l'élaboration de notre solution est l'approche Kimball "Bottom-Up" vu qu'elle traite chaque processus métier en entier.

2.4 Planification de travail

2.4.1 Répartition des releases

Release ID	Nom du Sprint

Release ID	Nom du Sprint
1	<ul style="list-style-type: none"> • Sprint 1 :Collecte des Données. • Sprint 2 :Création de Tableaux de Bord Initiaux.
2	<ul style="list-style-type: none"> • sprint 3 :Suivi des abonnements et des demandes. • Sprint 4 :Suivi des relèves et factures
3	<ul style="list-style-type: none"> • Sprint 5 : Suivi des fraudes et Anomalies. • Sprint 6 : Personnalisation des Tableaux de Bord.

TABLEAU 2.1 : Répartition des releases

2.4.2 Planification des sprints

La figure 2.3 représente le diagramme de Gantt illustrant la répartition du travail tout au long de la période de stage.

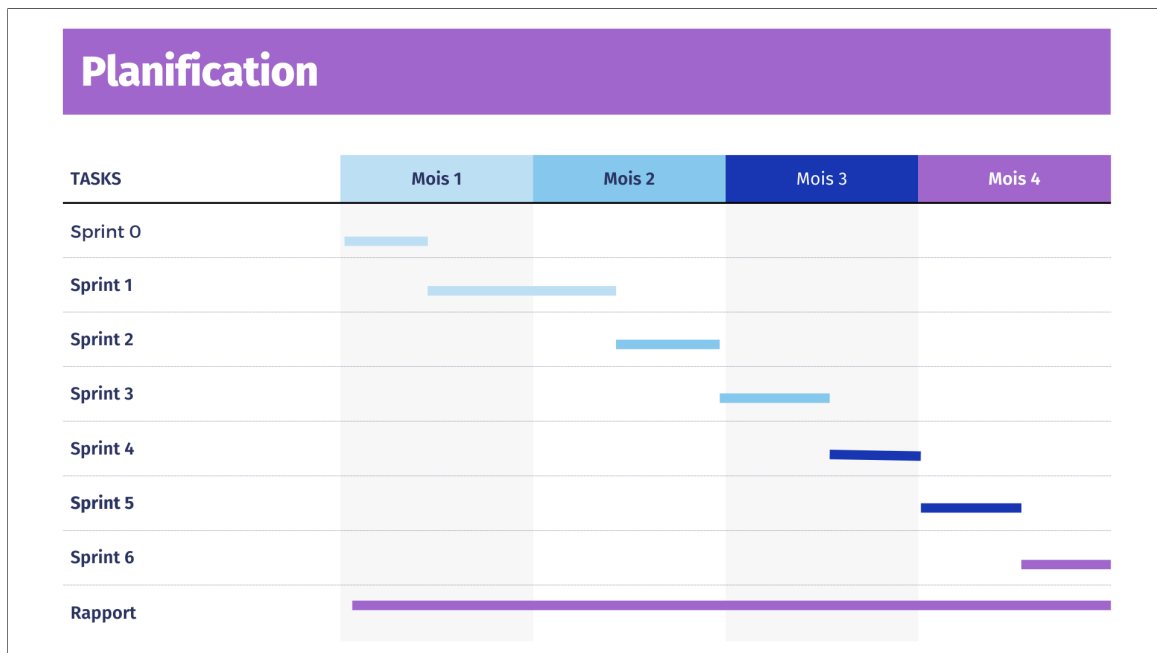


FIGURE 2.3 : Planification des sprints

2.5 Environnement de travail

2.5.1 Environnement matériel

un ordinateur Dell

— Processeur Intel Core i3-7th Gen

— 8,00 Go de RAM

2.5.2 Environnement logiciel

2.5.2.1 Architecture ELK Stack

L'ELK Stack est une suite d'outils open source conçue pour la collecte, la gestion et l'analyse des données . L'acronyme "ELK" signifie Elasticsearch, Logstash et Kibana. Voici une brève description de chacun de ces composants et de leur rôle dans l'architecture ELK Stack :

- **Elasticsearch (Entrepôt de données)** : Elasticsearch est le moteur de recherche distribué qui sert de système de stockage central pour les données . Il est conçu pour indexer rapidement de grandes quantités de données non structurées et permet une recherche et une analyse en temps réel. Elasticsearch utilise une structure de type NoSQL pour stocker les données de manière optimisée pour la recherche.



FIGURE 2.4 : Elasticsearch

- **Logstash (ETL)** : Logstash est l'outil d'ingestion de données qui collecte, normalise et transforme les données à partir de différentes sources avant de les envoyer à Elasticsearch. Il prend en charge une variété de formats de données et peut également effectuer des opérations de filtrage et de transformation sur les données en transit.



FIGURE 2.5 : Logstash

- **Kibana** : Kibana est une interface utilisateur web qui permet de visualiser et d'analyser les

données stockées dans Elasticsearch. Il offre une gamme d'outils de création de tableaux de bord interactifs, de graphiques et de visualisations pour explorer et comprendre les données. Kibana facilite également la création de requêtes et de filtres pour extraire des informations spécifiques.



FIGURE 2.6 : Kibana

Voici l'architecture complète de l'ELK stack :

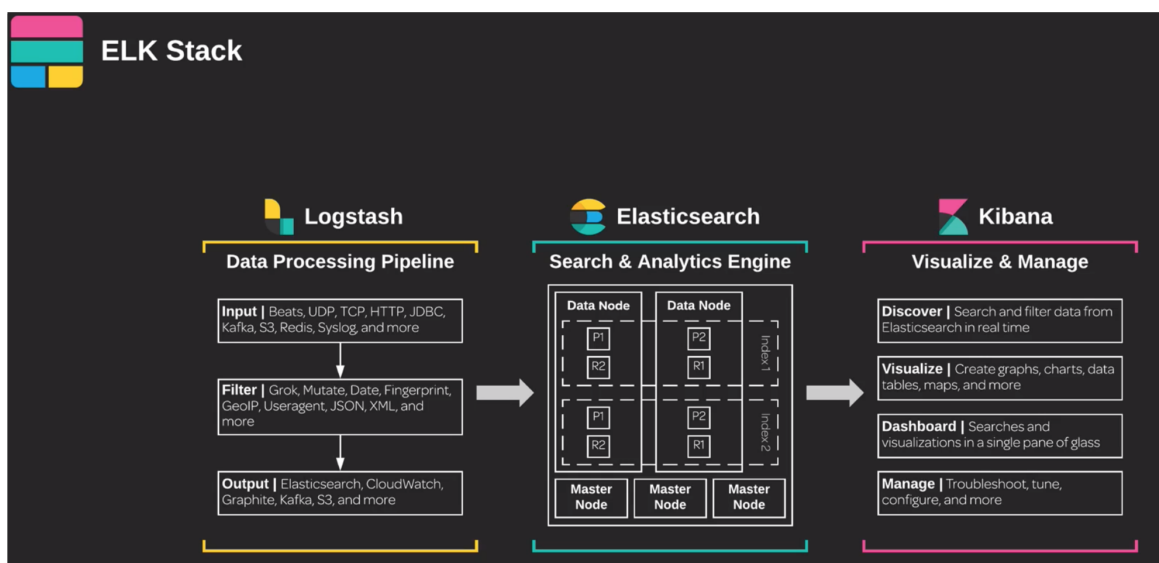


FIGURE 2.7 : Architecture ELK Stack

2.5.2.2 Source de données initiale

La société SIGA utilise Oracle comme base de données où elle stocke les données nécessaires.



FIGURE 2.8 : Oracle

Conclusion

La phase de préparation est une phase très importante dans le cycle de vie d'un projet, car elle offre une vue plus claire du système et des principales caractéristiques avec une Description détaillée du domaine métier. Par conséquent, nous avons introduit la méthodologie de travail et la planification des releases pour passer à la phase de réalisation.

RELEASE 1 : COLLECTE DES DONNÉES ET CRÉATION DE TABLEAUX DE BORD INITIAUX

Plan

1	Sprint 1 :Collecte des Données	23
2	Sprint 2 : création de Tableaux de bord initiaux	36

Introduction

Au cours de ce chapitre, nous allons présenter les différentes étapes de réalisation du premier sprint "Collecte des Données", et du deuxième sprint "Création de Tableaux de Bord Initiaux".

3.1 Sprint 1 :Collecte des Données

Dans cette section nous allons présenter les différents étapes de la réalisation du premier sprint.

3.1.1 Objectifs du sprint 1

L'objectif du premier sprint est :

- Mettre en place l'infrastructure ELK Stack.
- Mettre en place les pipelines de collecte des données.
- Intégrer les données des abonnements ,des relèves et des factures dans ELK Stack.
- Effectuer une première validation des données.

3.1.2 Backlog du sprint 1

Id	Fonctionnalités	Priorité	Estimation (Jour)
1	En tant que équipe de développement je veux insataller les outils elasticsearch, logstash et kibana afin collecter , traiter et analyser les données	1	7
2	En tant que équipe de développement je veux creer des piplines logstash afin de collecter, transformer et charger les donnees dans elasticsearch	2	15
3	En tant que équipe de développement je veux Intégrer les données des abonnements ,des relèves et des factures dans elasticsearch	3	4
4	En tant que équipe de développement je veux Effectuer des tests et validation des données	4	4

TABLEAU 3.1 : Backlog du Sprint 1

3.1.3 Mettre en place l'infrastructure ELK Stack

3.1.3.1 mise en place d'Elasticsearch

- **Installation et Configuration :**

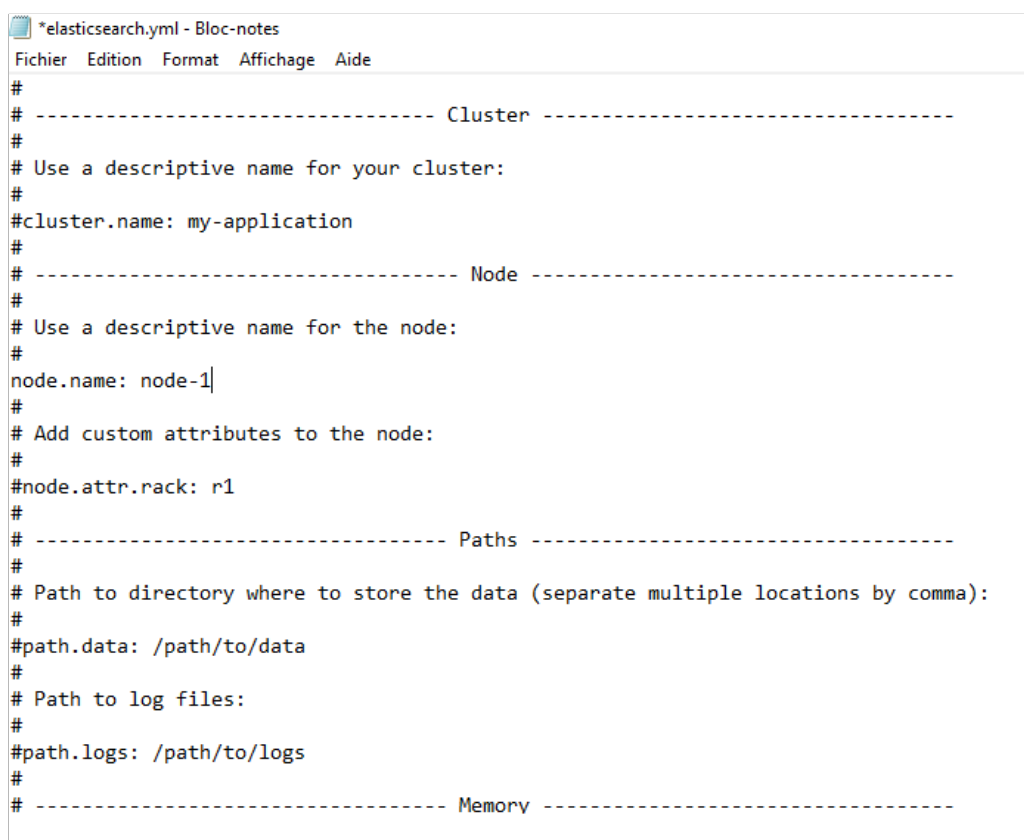
- Installation :

Avant de commencer l'installation, on doit assurer que notre système répond aux exigences d'Elasticsearch . Aussi Vérifier la compatibilité avec la version d'Elasticsearch que nous souhaitons installer.

Télécharger la version elasticsearch appropriée en fonction de notre système d'exploitation. Une fois le fichier d'installation téléchargé, extrayons l'archive dans le répertoire de projet.

- Configuration :

Avant de démarrer Elasticsearch, nous devons configurer certains paramètres. Le fichier de configuration principal d'Elasticsearch se trouve généralement dans le répertoire "config". Nous pouvons modifier des paramètres tels que le numéro de port, la mémoire allouée, la configuration du cluster, etc., en fonction de nos besoins.

A screenshot of a text editor window titled "*elasticsearch.yml - Bloc-notes". The window shows the default configuration for Elasticsearch. The menu bar includes "Fichier", "Edition", "Format", "Affichage", and "Aide". The content of the file is as follows:

```
#
# ----- Cluster -----
#
# Use a descriptive name for your cluster:
#
#cluster.name: my-application
#
# ----- Node -----
#
# Use a descriptive name for the node:
#
node.name: node-1
#
# Add custom attributes to the node:
#
#node.attr.rack: r1
#
# ----- Paths -----
#
# Path to directory where to store the data (separate multiple locations by comma):
#
#path.data: /path/to/data
#
# Path to log files:
#
#path.logs: /path/to/logs
#
# ----- Memory -----
```

FIGURE 3.1 : Fichier configuration elasticsearch

Dans notre cas , nous avons ajouté des noeuds a notre cluster afin d'assurer la scalabilité, la haute disponibilité, la tolérance aux pannes et l'optimisation des performances de notre

système de recherche et d'analyse de données. La capacité d'évoluer horizontalement est l'une des caractéristiques clés d'Elasticsearch, ce qui en fait une solution puissante pour gérer des volumes importants de données en croissance constante.

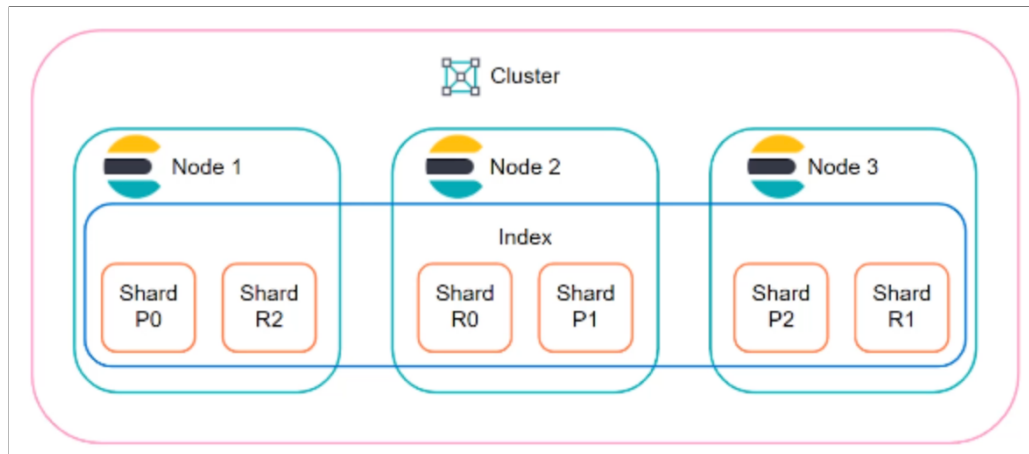


FIGURE 3.2 : cluster elasticsearch

- Démarrage et Vérification de l'état :

-Démarrage :

Pour démarrer Elasticsearch, on exécute le script d'exécution approprié en fonction de notre système d'exploitation. Sur Windows, nous pouvons exécuter 'bin/elasticsearch.bat'

```

C:\Windows\System32\cmd.exe - elasticsearch.bat
Microsoft Windows [version 10.0.19045.3324]
(c) Microsoft Corporation. Tous droits réservés.

C:\ELK\elasticsearch\bin>elasticsearch.bat
[2023-09-06T22:47:37,831][INFO ][o.e.n.Node               ] [DESKTOP-H5T406P] version[8.6.1], pid[2508], build[zip/180c9
830da956993e59e2cd70eb32b5e383ea42c/2023-01-24T21:35:11.506992272Z], OS[Windows 10/10.0/amd64], JVM[Oracle Corporation/O
penJDK 64-Bit Server VM/19.0.1/19.0.1+10-21]
[2023-09-06T22:47:38,065][INFO ][o.e.n.Node               ] [DESKTOP-H5T406P] JVM home [C:\ELK\elasticsearch\jdk], using
bundled JDK [true]
[2023-09-06T22:47:38,068][INFO ][o.e.n.Node               ] [DESKTOP-H5T406P] JVM arguments [-Des.networkaddress.cache.t
tl=60, -Des.networkaddress.cache.negative.ttl=10, -Djava.security.manager=allow, -XX:+AlwaysPreTouch, -Xss1m, -Djava.awt
.headless=true, -Dfile.encoding=UTF-8, -Djna.nosys=true, -XX:-OmitStackTraceInFastThrow, -Dio.netty.noUnsafe=true, -Dio.
netty.noKeySetOptimization=true, -Dio.netty.recycler.maxCapacityPerThread=0, -Dlog4j.shutdownHookEnabled=false, -Dlog4j2
.disable.jmx=true, -Dlog4j2.formatMsgNoLookups=true, -Djava.locale.providers=SPI,COMPAT, --add-opens=java.base/java.io=A
LL=UNNAMED, -XX:+UseG1GC, -Djava.io.tmpdir=C:\Users\del\AppData\Local\Temp\elasticsearch, -XX:+HeapDumpOnOutOfMemoryErr
or, -XX:+ExitOnOutOfMemoryError, -XX:HeapDumpPath=data, -XX:ErrorFile=logs/hs_err_pid%p.log, -Xlog:gc*,gc+age=trace,safe
point:file=logs/gc.log:utctime,pid,tags:filecount=32,filesize=64m, -Xms4046m, -Xmx4046m, -XX:MaxDirectMemorySize=2121269
248, -XX:G1HeapRegionSize=4m, -XX:InitiatingHeapOccupancyPercent=30, -XX:G1ReservePercent=15, -Des.distribution.type=zip
, --module-path=C:\ELK\elasticsearch\lib, --add-modules=jdk.net, -Djdk.module.main=org.elasticsearch.server]
[2023-09-06T22:48:05,336][INFO ][c.a.c.i.j.JacksonVersion ] [DESKTOP-H5T406P] Package versions: jackson-annotations=2.13
.2, jackson-core=2.13.2, jackson-databind=2.13.2.2, jackson-dataformat-xml=2.13.2, jackson-datatype-jsr310=2.13.2, azure
-core=1.27.0, Troubleshooting version conflicts: https://aka.ms/azsdk/java/dependency/troubleshoot
[2023-09-06T22:48:27,785][INFO ][o.e.p.PluginsService     ] [DESKTOP-H5T406P] loaded module [aggregations]
[2023-09-06T22:48:27,788][INFO ][o.e.p.PluginsService     ] [DESKTOP-H5T406P] loaded module [aggs-matrix-stats]
[2023-09-06T22:48:27,790][INFO ][o.e.p.PluginsService     ] [DESKTOP-H5T406P] loaded module [analysis-common]
[2023-09-06T22:48:27,794][INFO ][o.e.p.PluginsService     ] [DESKTOP-H5T406P] loaded module [apm]
[2023-09-06T22:48:27,795][INFO ][o.e.p.PluginsService     ] [DESKTOP-H5T406P] loaded module [constant-keyword]
[2023-09-06T22:48:27,797][INFO ][o.e.p.PluginsService     ] [DESKTOP-H5T406P] loaded module [data-streams]
[2023-09-06T22:48:27,799][INFO ][o.e.p.PluginsService     ] [DESKTOP-H5T406P] loaded module [frozen-indices]
[2023-09-06T22:48:27,800][INFO ][o.e.p.PluginsService     ] [DESKTOP-H5T406P] loaded module [ingest-attachment]

```

FIGURE 3.3 : Démarrage d'elasticsearch

-Vérification de l'état :

Pour vérifier si Elasticsearch est en cours d'exécution en accédant à `http://localhost:9200` dans un navigateur web. Nous devons voir des informations sur la version d'Elasticsearch et son état.

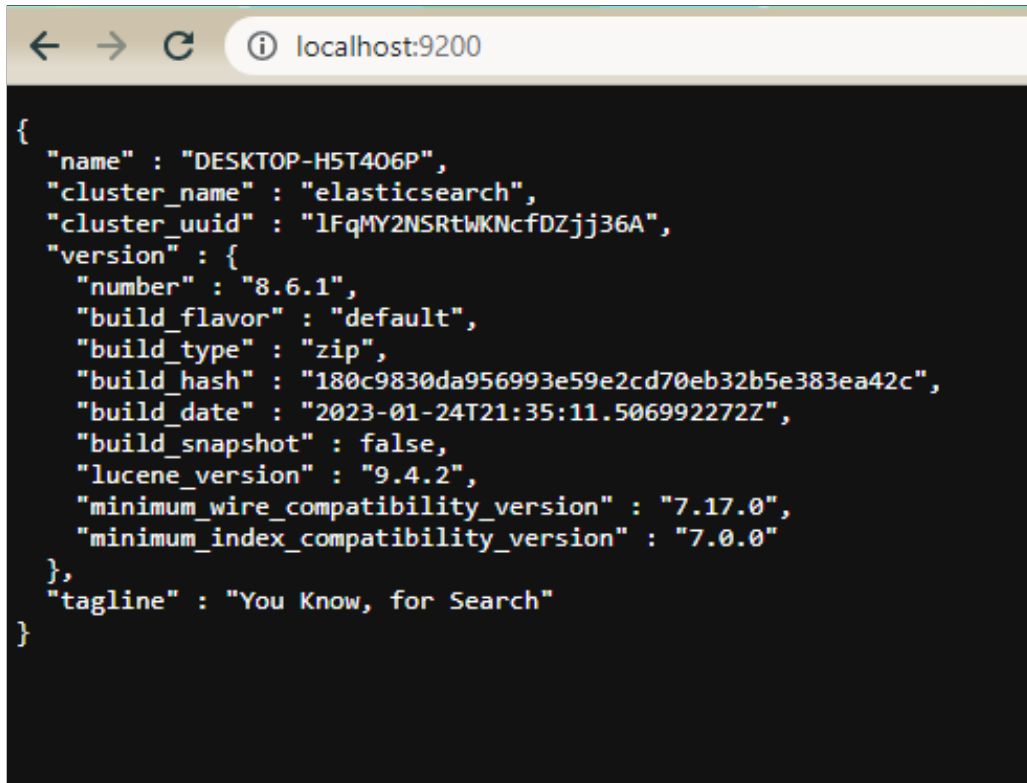


FIGURE 3.4 : Etat d'elasticsearch

- **les problèmes rencontrés et résolu :**

- Problème version Java .
- Problème mémoire .
- Problème de configuration des noeuds du cluster .

3.1.3.2 mise en place de Logstash

- **Installation et Configuration :**

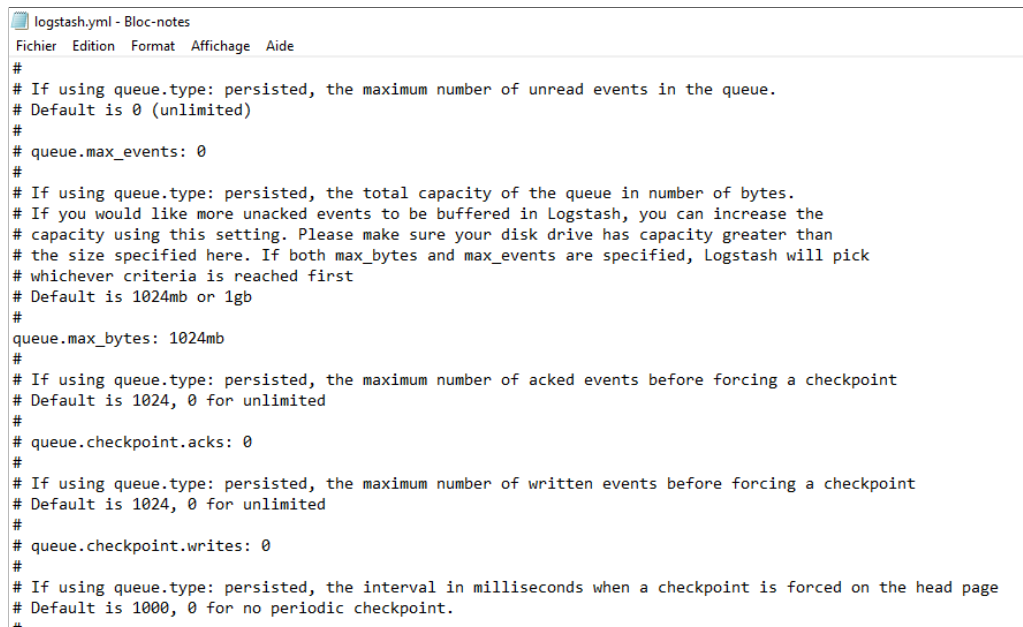
- Installation :

Pour l'installation de Logstash, il faut assurer que notre système répond aux prérequis système. Logstash est généralement compatible avec les systèmes d'exploitation Linux, Windows et macOS. Nous aurons également besoin de Java, car Logstash est écrit en Java.

Télécharger la dernière version de Logstash compatible avec notre système d'exploitation. Une fois le fichier d'installation téléchargé, extrayons l'archive dans le répertoire de projet.

- Configuration :

Logstash est livré avec un fichier de configuration par défaut situé dans le répertoire "config". nous pouvons personnaliser ce fichier de configuration en fonction de nos besoins. La configuration de Logstash est essentielle car elle définit comment Logstash collecte, transforme et transfère les données.



```

logstash.yml - Bloc-notes
Fichier  Edition  Format  Affichage  Aide
#
# If using queue.type: persisted, the maximum number of unread events in the queue.
# Default is 0 (unlimited)
#
# queue.max_events: 0
#
# If using queue.type: persisted, the total capacity of the queue in number of bytes.
# If you would like more unacked events to be buffered in Logstash, you can increase the
# capacity using this setting. Please make sure your disk drive has capacity greater than
# the size specified here. If both max_bytes and max_events are specified, Logstash will pick
# whichever criteria is reached first
# Default is 1024mb or 1gb
#
# queue.max_bytes: 1024mb
#
# If using queue.type: persisted, the maximum number of acked events before forcing a checkpoint
# Default is 1024, 0 for unlimited
#
# queue.checkpoint.acks: 0
#
# If using queue.type: persisted, the maximum number of written events before forcing a checkpoint
# Default is 1024, 0 for unlimited
#
# queue.checkpoint.writes: 0
#
# If using queue.type: persisted, the interval in milliseconds when a checkpoint is forced on the head page
# Default is 1000, 0 for no periodic checkpoint.
#

```

FIGURE 3.5 : Fichier configuration Logstash

- **Démarrage**

-Démarrage :

Pour démarrer Logstash, on accède au répertoire où nous avons extrait Logstash, et exécutons la commande de démarrage dans une fenêtre d'invite de commande. Nous pouvons utiliser la commande :

```
'logstash.bat -f syslog.conf'
```

3.1.3.3 mise en place de Kibana

- **Installation et Configuration :**

- Installation :

Pour l'installation de Kibana, il faut assurer que notre système répond aux prérequis système. Kibana est généralement compatible avec les systèmes d'exploitation Linux, Windows et macOS. Nous aurons également besoin de Java, car Kibana est écrit en JavaScript et nécessite Node.js. Télécharger la dernière version de Kibana compatible avec notre système d'exploitation. Une fois le fichier d'installation téléchargé, extrayons l'archive dans le répertoire de projet.

- Configuration :

Kibana est livré avec un fichier de configuration par défaut situé dans le répertoire "config". Vous pouvez personnaliser ce fichier de configuration en fonction de vos besoins. La configuration de Kibana inclut des paramètres tels que l'URL d'Elasticsearch, le port d'écoute de Kibana, la sécurité, etc.

- **Démarrage et Accès à l'Interface Web**

-Démarrage :

Pour démarrer Kibana, on accède au répertoire où nous avons extrait Logstash, et exécutons la commande de démarrage dans une fenêtre d'invite de commande. Nous pouvons utiliser la commande :

'kibana.bat'



FIGURE 3.6 : Demarrage kibana

- Accès à l'Interface Web :

Une fois Kibana démarré, nous pouvons accéder à son interface web en ouvrant un navigateur web et en accédant à l'URL par défaut :

<http://localhost:5601>.

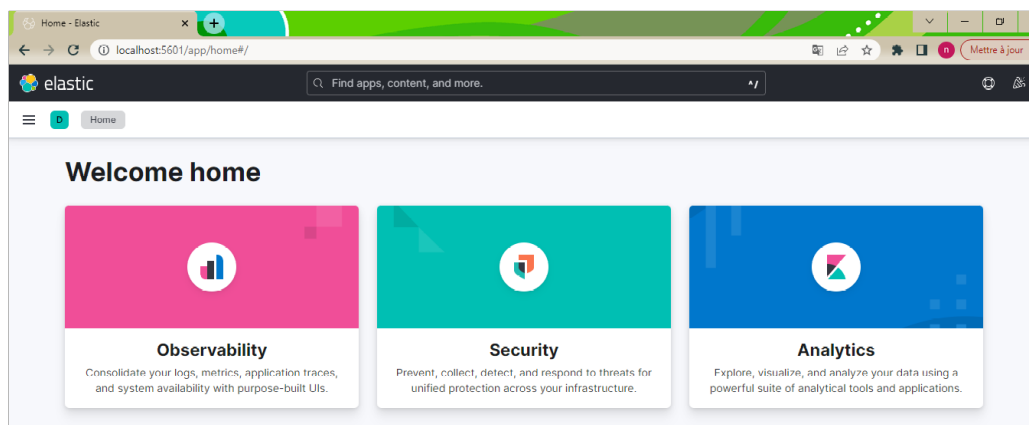


FIGURE 3.7 : Accès à l'Interface kibana

3.1.4 Création de pipelines

La création de pipelines en Logstash est essentielle pour le traitement, la transformation et l'acheminement des données depuis différentes sources vers une destination, généralement Elasticsearch ou une autre sortie. Les pipelines sont composés de plusieurs étapes de traitement qui permettent de nettoyer, d'enrichir et de structurer les données pour une utilisation ultérieure.

le fichier de configuration de pipeline doit contenir les trois composants suivants :

- Les Entrées (Inputs) : sont les sources de données (inputs) à partir desquelles Logstash doit collecter des données..

- Les filtres (Filters) : sont utilisés pour traiter et transformer les données collectées avant de les envoyer à la sortie.
- La Sortie (Output) : c'est où nous souhaitons envoyer les données. La sortie typique est Elasticsearch, mais vous pouvez également utiliser d'autres destinations comme des fichiers, des bases de données, etc.

Dans notre projet , la source de données est une base de données Oracle et la destination est elasticsearch ,donc on doit configurer le fichier de configuration pipeline comme suit :

- **Entrée (JDBC) :**

- jdbc_driver_library : le chemin vers le fichier JAR du pilote JDBC Oracle.
- jdbc_driver_class : la classe du pilote JDBC Oracle.
- jdbc_connection_string : la chaîne de connexion à la base de données Oracle .
- jdbc_user et jdbc_password : les informations d'identification de l'utilisateur qui a accès à la base de données.
- statement : requête SQL pour extraire les données de la base de données.

- **Filtre :**

- **Sortie (Elasticsearch) :**

- hosts : l'adresse et le port d'Elasticsearch où Logstash doit envoyer les données.
- index : le nom de l'index Elasticsearch dans lequel vous souhaitez indexer les données.

3.1.4.1 Pipeline abonnements, relèves et factures

- **Configuration Des Entrées :**

Pour bien indexer les données dans elasticsearch , il faut écrire une requête SQL correcte qui satisfait nos besoins. Pour cela , la Choix des tables dimensions est nécessaire afin de faire une jointure avec nos tables faits (abonnement , relèves et factures).

Les tables dimensions :

- dbo.Zone.
- dbo.Structure.
- dbo.Secteur.
- dbo.Tour .

Après la définition des tables dimensions , on passe a la choix de types de jointure SQL qui correspond a nos besoins. commençons par introduire la jointure SQL :


Une jointure SQL est une opération qui combine des lignes de deux tables ou plus en fonction d'une colonne commune entre elles. Elle est largement utilisée pour interroger et récupérer des

données à partir de plusieurs tables dans une base de données relationnelle. Il existe plusieurs types de jointures SQL, notamment :

- **INNER JOIN** : Cette jointure retourne uniquement les lignes qui ont des correspondances dans les deux tables.
- **LEFT JOIN** : Cette jointure retourne toutes les lignes de la table de gauche (table1) et les lignes correspondantes de la table de droite (table2).
- **RIGHT JOIN** : C'est l'opposé du LEFT JOIN. Il retourne toutes les lignes de la table de droite et les lignes correspondantes de la table de gauche.
- **FULL JOIN** : Cette jointure retourne toutes les lignes de toutes les tables, en incluant les lignes correspondantes et les lignes sans correspondance.

Dans notre projet et selon nos données sources et nos besoins , nous avons choisi "LEFT JOIN" comme type de jointure .

- **Input Abonnement :**



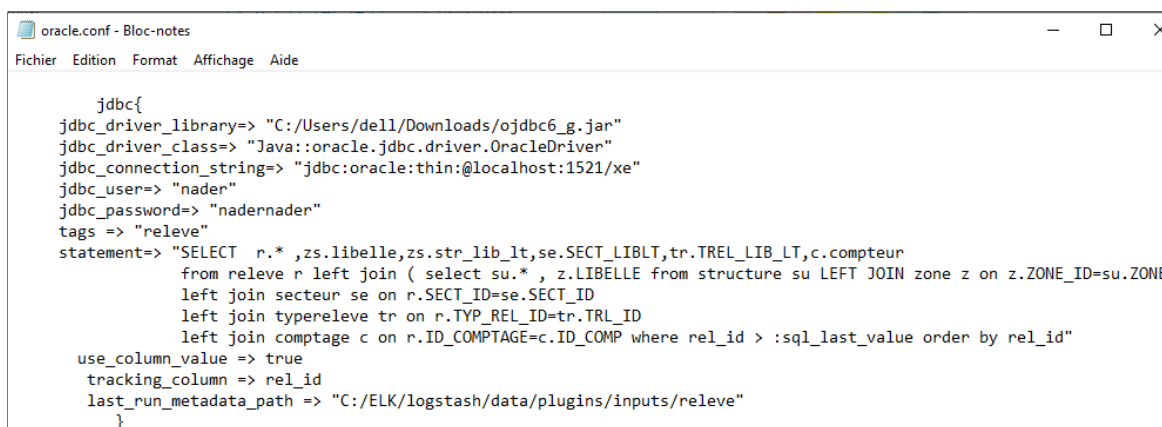
```

input{
  jdbc{
    jdbc_driver_library=> "C:/Users/dell/Downloads/ojdbc6_g.jar"
    jdbc_driver_class=> "Java::oracle.jdbc.driver.OracleDriver"
    jdbc_connection_string=> "jdbc:oracle:thin:@localhost:1521/xe"
    jdbc_user=> "nader"
    jdbc_password=> "nadernader"
    tags => "abonnement"
    statement=> "SELECT a.*,zs.libelle,zs.str_lib_lt,se.SECT_LIBLT
                from abonnement a left join ( select su.*, z.LIBELLE from structure su LEFT JOIN zone z on z.ZONE_ID=su.
                left join secteur se on a.SECT_ID=se.SECT_ID where abn_id > :sql_last_value order by abn_id"
    use_column_value => true
    tracking_column => abn_id
    last_run_metadata_path => "C:/ELK/logstash/data/plugins/inputs/abonnement"
  }
}

```

FIGURE 3.8 : Input Abonnement"

- **Input Relève :**



```

  jdbc{
    jdbc_driver_library=> "C:/Users/dell/Downloads/ojdbc6_g.jar"
    jdbc_driver_class=> "Java::oracle.jdbc.driver.OracleDriver"
    jdbc_connection_string=> "jdbc:oracle:thin:@localhost:1521/xe"
    jdbc_user=> "nader"
    jdbc_password=> "nadernader"
    tags => "releve"
    statement=> "SELECT r.*,zs.libelle,zs.str_lib_lt,se.SECT_LIBLT,tr.TREL_LIB_LT,c.compteur
                from releve r left join ( select su.*, z.LIBELLE from structure su LEFT JOIN zone z on z.ZONE_ID=su.ZONE
                left join secteur se on r.SECT_ID=se.SECT_ID
                left join typereleve tr on r.TYP_REL_ID=tr.TRL_ID
                left join comptage c on r.ID_COMPAGE=c.ID_COMP where rel_id > :sql_last_value order by rel_id"
    use_column_value => true
    tracking_column => rel_id
    last_run_metadata_path => "C:/ELK/logstash/data/plugins/inputs/releve"
  }
}

```

FIGURE 3.9 : Input Relève"

- **Input Facture :**

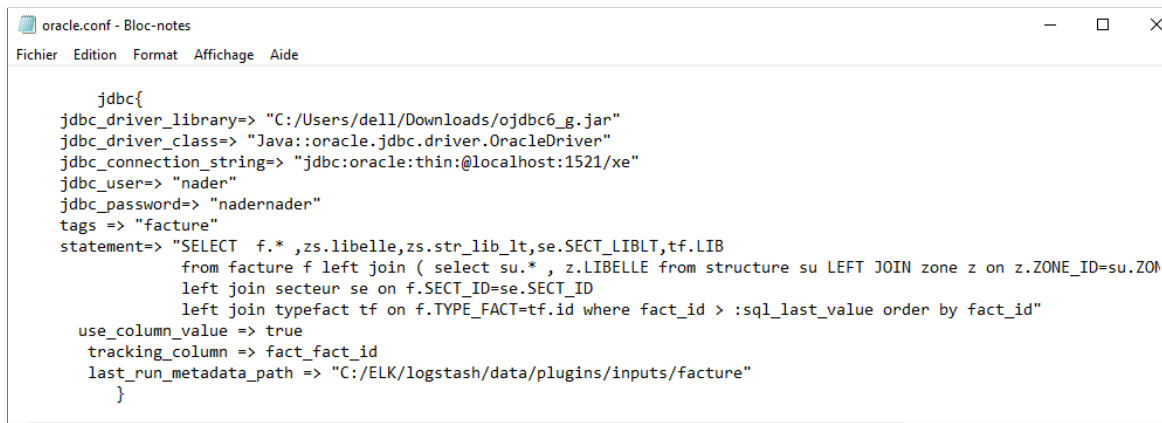


FIGURE 3.10 : Input Facture"

- **Configuration Des Sorties :** Notre sortie est Elasticsearch , il suffit d'indiquer l'adresse et le port d'Elasticsearch et spécifiez le nom de l'index Elasticsearch dans lequel vous souhaitez indexer les données.

- hosts : "localhost :9200"

- index : "abnanalyse1", "fctanalyse" , "rlvanalyse"

- **Output Abonnement :**

```

elasticsearch {
  hosts=> "localhost:9200"
  index=> "abnanalyse1"
}

```

FIGURE 3.11 : Output Abonnement"

- **Output Relève :**

```

elasticsearch {
  hosts=> "localhost:9200"
  index=> "rlvanalyse"
}

```

FIGURE 3.12 : Output Relève"

- **Output Facture :**

```

elasticsearch {
  hosts=> "localhost:9200"
  index=> "fctanalyse"
}

```

FIGURE 3.13 : Output Facture"

3.1.4.2 Pipeline redondance, anomalie et fraude

- **Redondance Clinet :**

dans cette etape, nous devons extraire en utilisant la requête SQL convenable les clients ayant plus que un abonnements au meme adresse et les charger dans un index elasticsearch .

```
input {
  jdbc {
    jdbc_driver_library=> "C:/Users/dell/Downloads/ojdbc6_g.jar"
    jdbc_driver_class=> "Java:oracle.jdbc.driver.OracleDriver"
    jdbc_connection_string=> "jdbc:oracle:thin:@localhost:1521/xe"
    jdbc_user=> "nader"
    jdbc_password=> "nadernader"
    statement=> "select cli_id ,abn_adresse,libelle,str_lib_lt,SECT_LIBLT,tour_id,TYPCLT_LIBLT,PFACT_LIBLT,COUNT(*)
FROM (SELECT a.*,zs.libelle,zs.str_lib_lt,se.SECT_LIBLT
from abonnement a left join ( select su.*, z.LIBELLE from structure su LEFT JOIN zone z on z.ZONE_ID=su.ZONE_ID) zs on a.str_id=zs.str_id
left join secteur se on a.SECT_ID=se.SECT_ID) where abn_vld= 1
GROUP BY cli_id ,abn_adresse,libelle,str_lib_lt,SECT_LIBLT,tour_id,TYPCLT_LIBLT,PFACT_LIBLT HAVING count(*) > 1;"
  }
}
output{
  elasticsearch{
    hosts=> "localhost:9200"
    index=> "abnredondance"
  }
  stdout {}
}
```

FIGURE 3.14 : Pipeline redondance Clinet"

- **Anomalie et fraude Relève :**

Les anomalies de relèves ont été traitées selon les trois types d'anomalies :

- Anomalie de comptage : lorsque le compteur n'est pas accessible.
- Fraude : lorsque la valeur de la période et la valeur de la consommation ne sont pas logiquement significatives par exemple (période > 0 et une consommation nulle)
- Anomalie de type Surconsommation : lorsqu'une consommation d'un seul relève est supérieure au triple de la moyenne de consommation dans le secteur.

Selon ces types des anomalies, on configure notre fichier configuration pour charger les données nécessaires dans elasticsearch.

```

input{
  jdbc{
    jdbc_driver_library=> "C:/Users/dell/Downloads/ojdbc6.g.jar"
    jdbc_driver_class=> "Java:oracle.jdbc.driver.OracleDriver"
    jdbc_connection_string=> "jdbc:oracle:thin:@localhost:1521/xs"
    jdbc_user=> "nader"
    jdbc_password=> "nader"
    tags => "fraude"
    statement=> "select * from (SELECT r.*,zs.libelle,zs.str_lib_lt,se.SECT_LIBLT,tr.TREL_LIB_LT,c.compteur
    from releve r left join ( select su.*, z.LIBELLE from structure su LEFT JOIN zone z on z.ZONE_ID=su.ZONE_ID) zs on r.str_id=zs.str_id
    left join secteur se on r.SECT_ID=se.SECT_ID
    left join typereleve tr on r.TYP_REL_ID=tr.TRL_ID
    left join comptage c on r.ID_COMPTAGE=c.ID_COMP) where REL_NBR_JR=0 AND REL_CONSOM_CALCUL<>0
  UNION
  select * from (SELECT r.*,zs.libelle,zs.str_lib_lt,se.SECT_LIBLT,tr.TREL_LIB_LT,c.compteur
    from releve r left join ( select su.*, z.LIBELLE from structure su LEFT JOIN zone z on z.ZONE_ID=su.ZONE_ID) zs on r.str_id=zs.str_id
    left join secteur se on r.SECT_ID=se.SECT_ID
    left join typereleve tr on r.TYP_REL_ID=tr.TRL_ID
    left join comptage c on r.ID_COMPTAGE=c.ID_COMP) where REL_NBR_JR<>0 AND REL_CONSOM_CALCUL=0"
  }
  jdbc{
    jdbc_driver_library=> "C:/Users/dell/Downloads/ojdbc6.g.jar"
    jdbc_driver_class=> "Java:oracle.jdbc.driver.OracleDriver"
    jdbc_connection_string=> "jdbc:oracle:thin:@localhost:1521/xs"
    jdbc_user=> "nader"
    jdbc_password=> "nader"
    tags => "anomalie"
    statement=> "select * from (SELECT r.*,zs.libelle,zs.str_lib_lt,se.SECT_LIBLT,tr.TREL_LIB_LT,c.compteur
    from releve r left join ( select su.*, z.LIBELLE from structure su LEFT JOIN zone z on z.ZONE_ID=su.ZONE_ID) zs on r.str_id=zs.str_id
    left join secteur se on r.SECT_ID=se.SECT_ID
    left join typereleve tr on r.TYP_REL_ID=tr.TRL_ID
    left join comptage c on r.ID_COMPTAGE=c.ID_COMP) where compteur LIKE 'COMPTEUR INACCESSIBLE'"
  }
  jdbc{
    jdbc_driver_library=> "C:/Users/dell/Downloads/ojdbc6.g.jar"
    jdbc_driver_class=> "Java:oracle.jdbc.driver.OracleDriver"
    jdbc_connection_string=> "jdbc:oracle:thin:@localhost:1521/xs"
    jdbc_user=> "nader"
    jdbc_password=> "nader"
    tags => "anomalie"
    statement=> "select * from (SELECT r.*,zs.libelle,zs.str_lib_lt,se.SECT_LIBLT,se.moy_cons,tr.TREL_LIB_LT,c.compteur
    from releve r left join ( select su.*, z.LIBELLE from structure su LEFT JOIN zone z on z.ZONE_ID=su.ZONE_ID) zs on r.str_id=zs.str_id
    left join secteur se on r.SECT_ID=se.SECT_ID
    left join typereleve tr on r.TYP_REL_ID=tr.TRL_ID
    left join comptage c on r.ID_COMPTAGE=c.ID_COMP) where REL_CONSOM_CALCUL > (3*moy_cons)"
  }
}

```

FIGURE 3.15 : Pipeline Anomalie et fraude Relève"

- **Fraude facturation :**

nous devons extraire en utilisant la requête SQL convenable les factures ayant la date de traitement different de la date de facture et les charger dans un index elasticsearch .

```

Fichier Edition Format Affichage Aide
input{
  jdbc{
    jdbc_driver_library=> "C:/Users/dell/Downloads/ojdbc6.g.jar"
    jdbc_driver_class=> "Java:oracle.jdbc.driver.OracleDriver"
    jdbc_connection_string=> "jdbc:oracle:thin:@localhost:1521/xs"
    jdbc_user=> "nader"
    jdbc_password=> "nader"
    statement=> "select * from (SELECT f.*,zs.libelle,zs.str_lib_lt,se.SECT_LIBLT,tf.LIB
    from facture f left join ( select su.*, z.LIBELLE from structure su LEFT JOIN zone z on z.ZONE_ID=su.ZONE_ID) zs on f.str_id=zs.str_id
    left join secteur se on f.SECT_ID=se.SECT_ID
    left join typefact tf on f.TYPE_FACT=tf.id) where (EXTRACT(month from FACT_DATE)-FACT_PERIODE) <>0"
  }
}
filter {
}
output{
  elasticsearch{
    hosts=> "localhost:9200"
    index=> "fraudefact"
  }
  stdout {}
}

```

FIGURE 3.16 : Pipeline fraude Facture

3.1.5 Integration de données

Une fois que nous avons configuré les pipelines : l'input, les filtres et l'output, nous pouvons démarrer Logstash.

Pour démarrer Logstash avec notre fichier de configuration, on ouvre une invite de commande (terminal) et on exécute la commande de démarrage . A ce moment , Logstash commencera à exécuter le traitement des données en fonction de notre configuration. Nous verrons les logs dans la console indiquant que Logstash est en cours d'exécution.

Pendant l'exécution, des informations sur ce que fait Logstash qui nous aideront à résoudre les problèmes et les erreurs sont affichées sur la console.

- **integration de données abonnement , relève et facture :**

Après l'exécution du commande de démarrage avec le fichier de configuration de pipline abonnement , relève et facture , logstash transfer les données vers elasticsearch et les indexe dans les index indiquer dans le fichier configuartion .

"logstash.bat -f oracle.conf"

```

C:\Windows\System32\cmd.exe - logstash.bat -f oracle.conf
get` option to avoid potential schema conflicts (if your data is ECS compliant or non-conflicting, feel free to ignore this message)
[2023-09-07T23:26:57,352][INFO ][logstash.javapipeline ][main] Pipeline started {"pipeline.id"=>"main"}
[2023-09-07T23:26:57,385][INFO ][logstash.agent      ] Pipelines running {:count=>1, :running_pipelines=>[:main], :non_running_pipelines=>[]}
[2023-09-07T23:27:06,688][INFO ][logstash.inputs.jdbc ][main][35b2c70039ba13821a87137ce49f839d233c0f248e7f2e081652305a6c5f3f48] (9.275966s) SE
LECT a.* ,zs.libelle,zs.str_lib_lt,se.SECT_LIBLT
FROM abonnement a LEFT JOIN ( SELECT su.* , z.LIBELLE from structure su LEFT JOIN zone z on z.ZONE_ID=su.ZONE_ID) zs on a.str_id
=zs.str_id
      LEFT JOIN secteur se on a.SECT_ID=se.SECT_ID
[2023-09-07T23:27:06,697][INFO ][logstash.inputs.jdbc ][main][89cd5a95f4bcdde16ce51222b00af97dadb3114601596bbe81c9a9660cd67b] (9.293096s) SE
LECT f.* ,zs.libelle,zs.str_lib_lt,se.SECT_LIBLT,tf.LIB
FROM facture f LEFT JOIN ( SELECT su.* , z.LIBELLE from structure su LEFT JOIN zone z on z.ZONE_ID=su.ZONE_ID) zs on f.str_id=zs
.str_id
      LEFT JOIN secteur se on f.SECT_ID=se.SECT_ID
      LEFT JOIN typefact tf on f.TYPE_FACT=tf.id
[2023-09-07T23:27:06,729][INFO ][logstash.inputs.jdbc ][main][cf92f25c60a5d884e418f5698855180f77775714b298044973d99e4620c9f55c] (9.324899s) SE
LECT r.* ,zs.libelle,zs.str_lib_lt,se.SECT_LIBLT,trl.TREL_LT,c.compteur
FROM releve r LEFT JOIN ( SELECT su.* , z.LIBELLE from structure su LEFT JOIN zone z on z.ZONE_ID=su.ZONE_ID) zs on r.str_id=zs
.str_id
      LEFT JOIN secteur se on r.SECT_ID=se.SECT_ID
      LEFT JOIN typerel tr on r.TYP_REL_ID=tr.TREL_ID
      LEFT JOIN comptage c on r.ID_COMPTAGE=c.ID_COMP
C:\ELK\logstash\vendor\bundle\jruby/2.6.0/gems/manticore-0.9.1-java/lib/manticore/client.rb:284: warning: already initialized constant Manticore::
Client::HttpPost
C:\ELK\logstash\vendor\bundle\jruby/2.6.0/gems/manticore-0.9.1-java/lib/manticore/client.rb:284: warning: already initialized constant Manticore::
Client::HttpPost
C:\ELK\logstash\vendor\bundle\jruby/2.6.0/gems/manticore-0.9.1-java/lib/manticore/client.rb:536: warning: already initialized constant Manticore::
Client::StringEntity

```

FIGURE 3.17 : integration de données abonnement , relève et facture

- **integration de données redondance , anomalie et fraude :**

Comme l'integration de données abonnement , relève et facture , il suffit d'exécuter la commande de démarrage logstash avec les fichiers de configurations et les données seront transferer vers elasticsearch et seront stockés dans les index.

```

C:\Windows\System32\cmd.exe
{
  "str_lib_lt" => "DAR-NAIM",
  "str_id" => 99,
  "clt_id" => 773131,
  "sect_id" => 1661,
  "comm_err" => nil,
  "@timestamp" => 2023-09-07T22:08:21.938425500Z,
  "fact_mnt_ttc" => -633,
  "tour_id" => 37092,
  "code_facture" => 0,
  "flagannulle" => 0.0,
  "date_gen" => nil,
  "fact_mnt" => -1970,
  "fact_periode" => 9,
  "fact_id" => 2605222,
  "frcp_mnt_calc" => nil,
  "abn_id" => 15679205,
  "type_fact" => 4,
  "fact_code_grp" => nil,
  "fact_date" => 2021-10-22T22:00:00.000Z,
  "fat_gte_cons" => 0,
  "frcp_id" => 2587843,
  "calc_vld" => 1.0,
  "libelle" => "NOUAKCHOTT : ZONE EST",
  "sect_liblt" => "TENS H20 H22 H23...H31",
  "@version" => "1",
  "fact_vld" => 1.0,
  "fact_annee" => 2021
}
[2023-09-08T00:08:23,545][INFO ][logstash.javapipeline ][main] Pipeline terminated {"pipeline.id"=>"main"}
[2023-09-08T00:08:24,028][INFO ][logstash.pipelinesregistry] Removed pipeline from registry successfully {:pipeline_id=>

```

FIGURE 3.18 : integration de données redondance , anomalie et fraude

- **Les problèmes rencontrés et résolus :**

- Problèmes de configuration mémoire.
- Problèmes dans les requêtes SQL.

3.1.6 Tests et Validations de données

Après la transfert de données vers elasticsearch , on accède a l'interface web de kibana où nous pouvons trouver nos données et verfier s'ils besoin de modification ou de re-transférer autre fois.

Pour accéder a l'interface web de kibana , il suffit just ecrire dans le navigateur " localhost :5601".

Dans la fenêtre management sur Kibana , on accède dans "index management" où nous pouvons trouver tous les index de notre cluster elasticsearch.

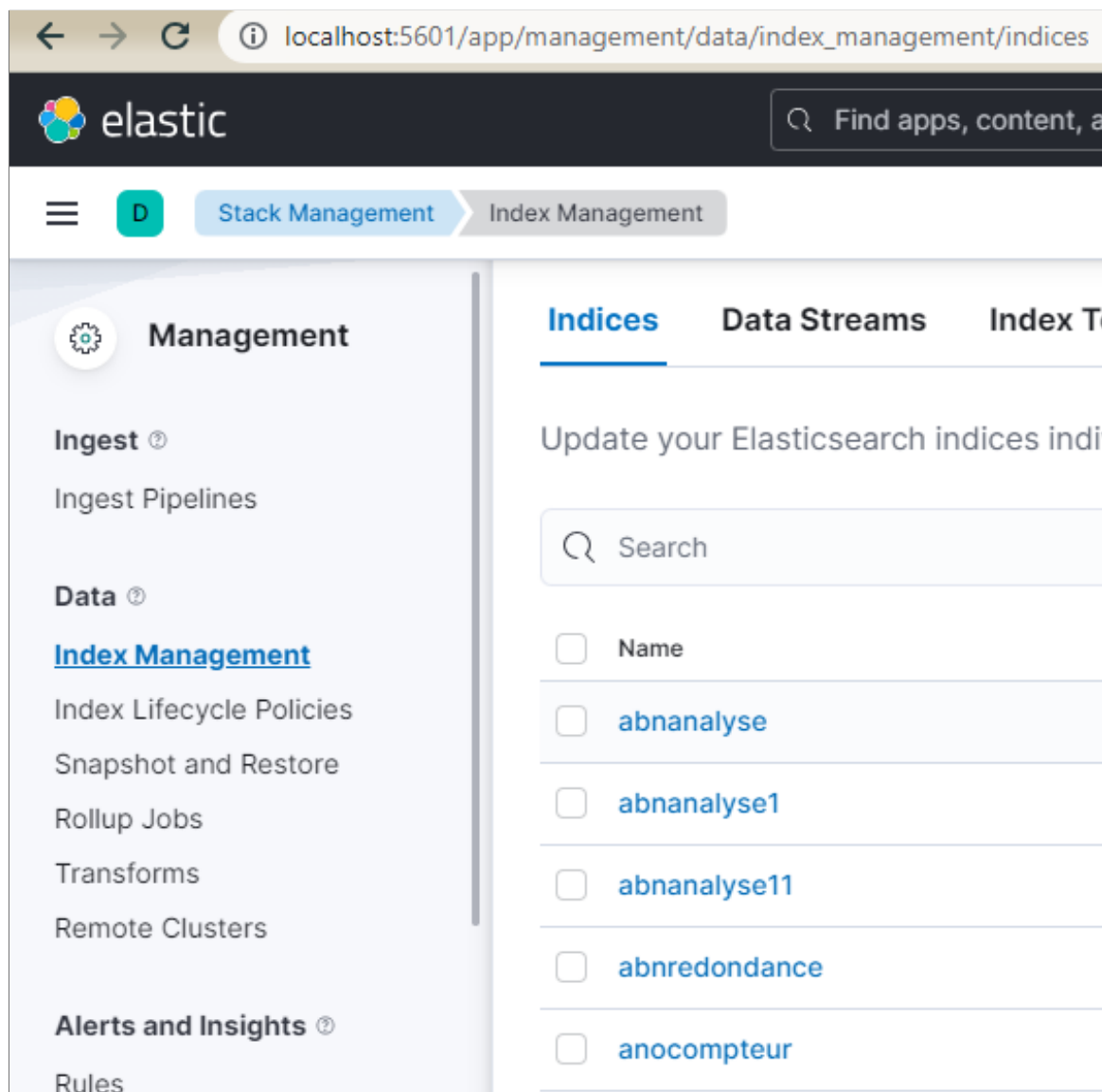


FIGURE 3.19 : Index Management

3.2 Sprint 2 : création de Tableaux de bord initiaux

Dans cette section nous allons présenter les différentes étapes permettant la réalisation du sprint "Création de tableaux de bord initiaux".

3.2.1 Objectifs du sprint 2

L'objectif du deuxième sprint est de créer des tableaux de bord initiaux pour les utilisateurs prochainement dans la visualisation et l'analyse de données .

3.2.2 Analyse des Besoins Utilisateur

Avant de commencer la création des tableaux de bord, il faut tout d'abord comprendre les besoins en matière de visualisation des données et Identifier les principales métriques, indicateurs que l'utilisateur souhaite voir sur les tableaux de bord.

Dans notre projet , les tableaux de bord à créer sont :

- tableau de bord Abonnement
- tableau de bord Abonnement validée
- tableau de bord Abonnement résilié
- tableau de bord relèvement
- tableau de bord relèvement validée
- tableau de bord relèvement calculée
- tableau de bord facture
- tableau de bord facture calculée
- tableau de bord facture validée
- tableau de bord facture erronée
- tableau de bord demande
- tableau de bord consommation
- tableau de bord montant TTC
- les tableaux de bord d'anomalie et fraude

3.2.3 Création des Nouveaux Tableaux de Bord sur Kibana

Une fois connecté à Kibana, cliquons sur "Dashboard" dans le menu puis sur le bouton "Create Dashboard" pour créer un nouveau tableau de bord.

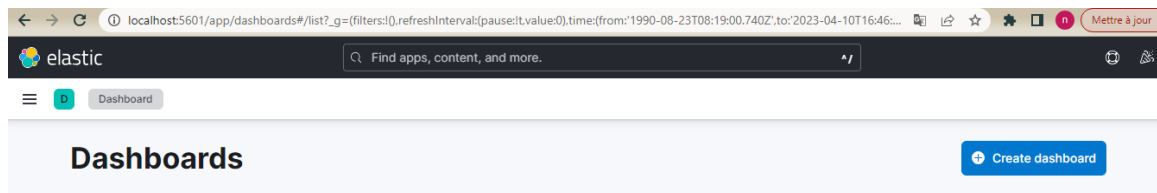


FIGURE 3.20 : creation dashboard kibana"

3.2.4 Sélection des Visualisations

Après la creation des nouveaux tableaux de bord , il faut qu'on Choisi les types de visualisations appropriés en fonction des données que nous collectons et selon les besoins d'utilisateur.

Pour ajouter des visualisations à notre tableau de bord, Cliquons sur le bouton "Add" en haut du tableau de bord, ensuite , on sélectionne le type de visualisation que nous souhaitons ajouter

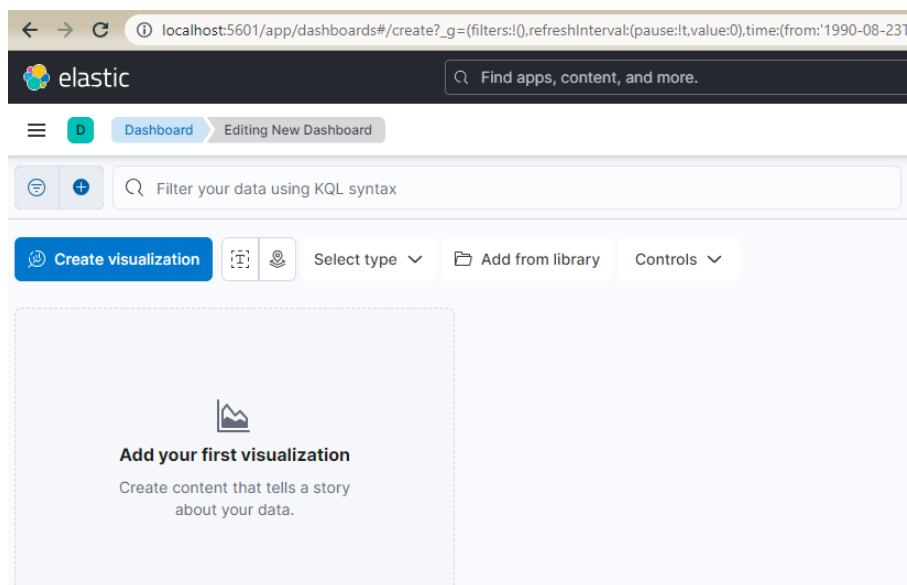


FIGURE 3.21 : creation Visualisation kibana"

3.2.5 Tests et Validation

Choisissons un index Elasticsearch à partir duquel nous souhaitons . Tester les tableaux de bord pour nous assurer qu'ils affichent correctement les données et qu'ils peuvent répondre aux besoins des utilisateurs.

Conclusion

Au cours de ce chapitre, nous avons présenté la réalisation de la première release "collecte des données et création de tableaux de bord initiaux". Pour ce faire, nous avons passé par la réalisation des deux premiers sprints.

RELEASE 2 : SUIVI DES ABONNEMENTS, DES DEMANDES, DES RELÈVES ET DES FACTURES

Plan

1	sprint 3 :Suivi des Abonnements et des demandes	39
2	sprint 4 :Suivi des Relèves et des Factures	45

Introduction

Dans ce chapitre, nous allons présenter les différentes étapes de réalisation du troisième sprint "Suivi des abonnements et des demandes", et du quatrième sprint "Suivi des relèves et factures".

4.1 sprint 3 : Suivi des Abonnements et des demandes

4.1.1 Objectifs du sprint 3

L'objectif de ce sprint est de prendre en charge notre premier sujet d'analyse, qui consiste à suivre les abonnements et les demandes. Nous avons réparti les tâches de la manière suivante :

4.1.2 Backlog du sprint 3

Id	Fonctionnalités	Priorité	Estimation (Jour)
1	En tant que Chef d'exploitation je veux consulter l'état de tout les abonnements	1	2
2	En tant que Chef d'exploitation je veux consulter l'état de tout les abonnements valides	2	2
3	En tant que Chef d'exploitation je veux consulter l'état de tout les abonnements résiliés	3	2
4	En tant que Chef d'exploitation je veux consulter l'état de l'avance sur consommation	4	2
5	En tant que Chef d'exploitation je veux consulter l'état des demandes	5	2
6	En tant que Chef d'exploitation je veux consulter l'état de tout les redondances abonnements	6	2
7	En tant que Chef d'exploitation je veux que tout les tableaux de bord soient liés	7	2

TABLEAU 4.1 : Backlog du Sprint 3

4.1.3 Choix des indicateurs de performance

Les indicateurs de performance adoptés pour le suivi des demandes et des abonnements sont :

- Nombre de demandes par zone , structure , secteur et tour aussi par type client et date .
- Nombre des abonnements par zone , structure , secteur et tour aussi par type client et date .
- Pourcentage des abonnements en cours de traitement.
- Pourcentage des abonnements par type de client et type facturation.
- Solde avance sur consommation pour les abonnements par zone , structure , secteur et tour aussi par type client et date .
- Nombre des abonnements valides par zone , structure , secteur et tour aussi par type client et date .
- Pourcentage des abonnements valides par type de client et type facturation.
- Nombre des abonnements résilié par zone , structure , secteur et tour aussi par type client et date .
- Pourcentage des abonnements résilié par type de client et type facturation.

4.1.4 Les Tableaux de bord Abonnements et Demandes

Dans cette partie , nous accédons via un navigateur web en utilisant l'URL "localhost :5601" a l'interface web Kibana .

- **Accée au tableau de bord initial :** Dans le menu kibana nous cliquons sur "dashbord" , une liste des tableaux de bord s'affichera et nous accedons dans notre tableau de bord intial pour les abonnements et demande .

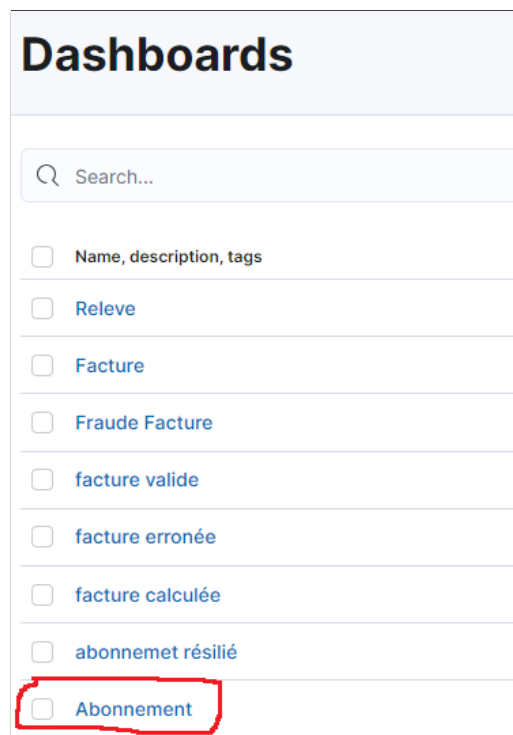


FIGURE 4.1 : Accée au Dashbord Kibana

- **Ajout de Visualisations :**

Un tableau de bord Kibana est composé de visualisations.

- **Configuration des Visualisations :**

Une fois que nous avons ajouté une visualisation, nous devons la configurer pour afficher les données souhaitées. Et cela, avec la choix de l'index Elasticsearch où nous stockons les données d'abonnement et demande.

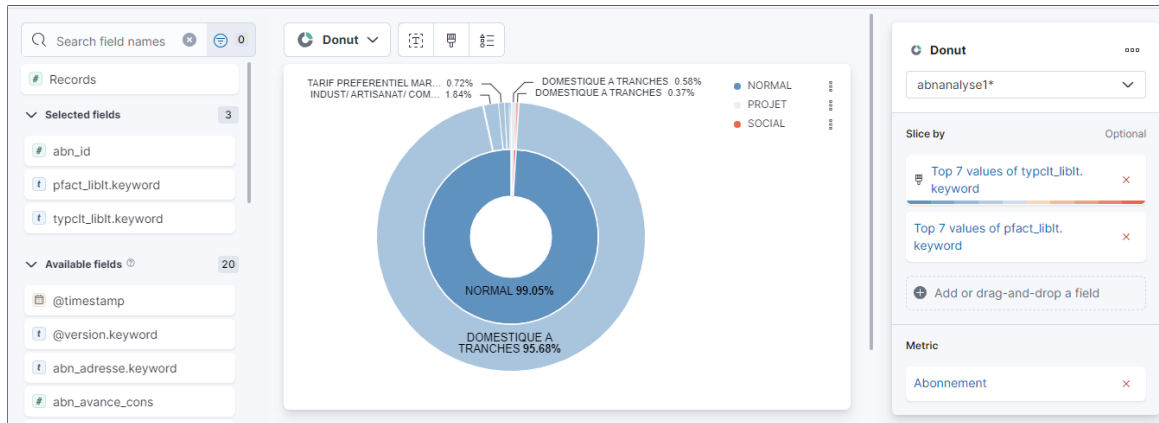


FIGURE 4.2 : Visualisation

- **Ajout et Placement des Visualisations :**

Après la création de visualisations, nous devons les ajouter à notre tableau de bord, les déplacer et redimensionner pour créer la mise en page souhaitée.

4.1.4.1 Tableau de bord Abonnement

Dans cette partie nous allons présenter le tableau de bord des abonnements selon les indicateurs de performance demandés. Pour cela nous citons les dimensions que nous avons pu dégager :

- **Dimension Client :** La dimension client permet de fournir des informations sur la clientèle. Cette dimension aide énormément à mieux comprendre les différents profils des prospects et à déterminer des critères de segmentation.
- **Dimension date :** Cette dimension présente l'axe primordial de chaque analyse BI qui n'est autre que la dimension temporelle.
- **Dimension tarification :** La dimension Tarification présente un axe d'analyse crucial. Elle concerne la catégorie de tarification des services offerts.
- **Dimension zone :** C'est une sous dimension du secteur ayant la plus grande hiérarchie et qui présente les zones géographiques du pays.
- **Dimension structure :** Pour chaque zone géographique, les sociétés d'eau possèdent un ensemble

de structures, qui sont des divisions bien délimitées pour assurer la performance du service de distribution des eaux.

- **Dimension secteur** : Les structures sont divisées en secteurs. Un secteur est caractérisé par une moyenne de consommation pour évaluer le besoin en eau.
- **Dimension tour** : Les secteurs sont divisées en tours.

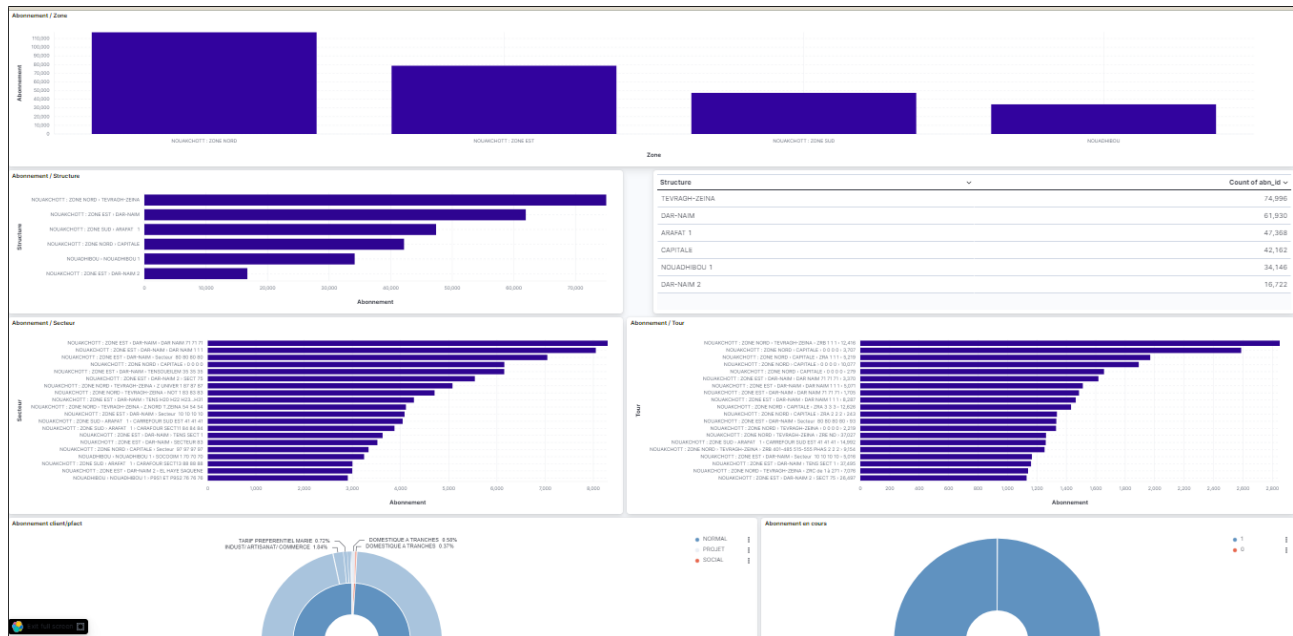


FIGURE 4.3 : Dashboard Abonnement

4.1.4.2 Tableau de bord Abonnement valide

comme l'étape dernière nous allons présenter le tableau de bord des abonnements valides selon Les indicateurs de performance demandées.

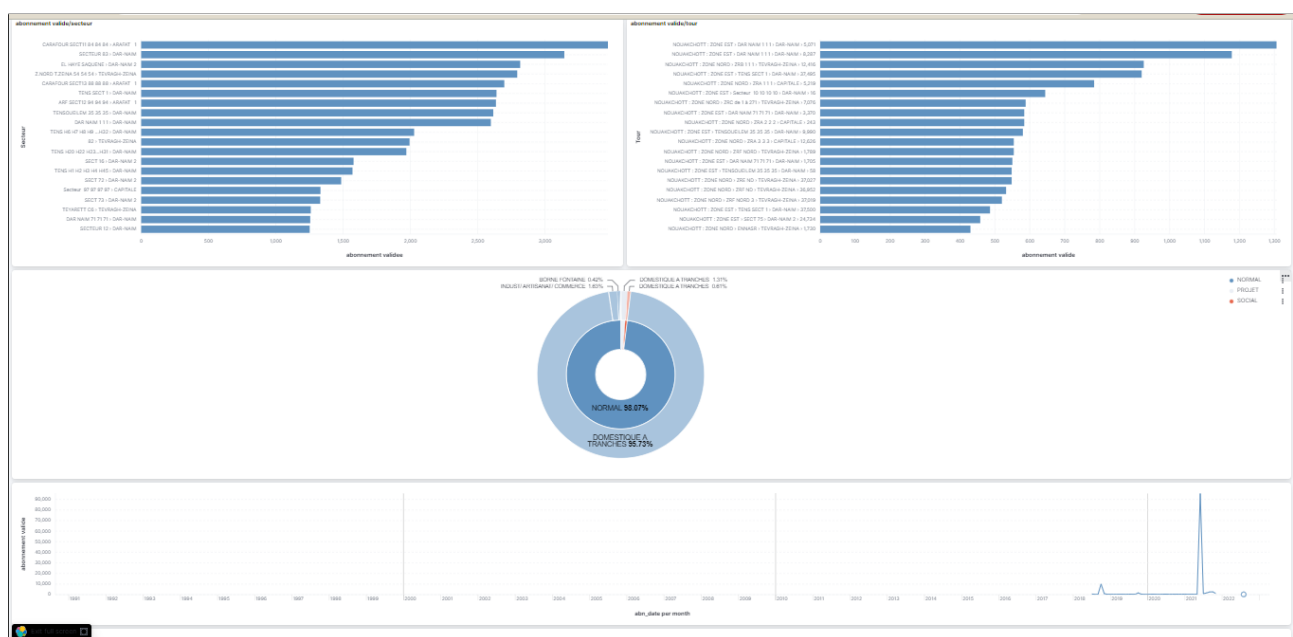


FIGURE 4.4 : Dashboard Abonnement Valide"

4.1.4.3 Tableau de bord Abonnement résilie

comme l'étape dernière nous allons présenter le tableau de bord des abonnements résiliés selon Les indicateurs de performance demandées.



FIGURE 4.5 : Dashbord Abonnement Resilie"

4.1.4.4 Tableau de bord Solde avance sur consommation

comme l'étape dernière nous allons présenter le tableau de bord de solde avance sur consommation selon Les indicateurs de performance demandées.



FIGURE 4.6 : Dashbord Solde avance sur consommation"

4.1.4.5 Tableau de bord Demande

comme l'étape dernière nous allons présenter le tableau de bord des Demandes selon Les indicateurs de performance demandées.

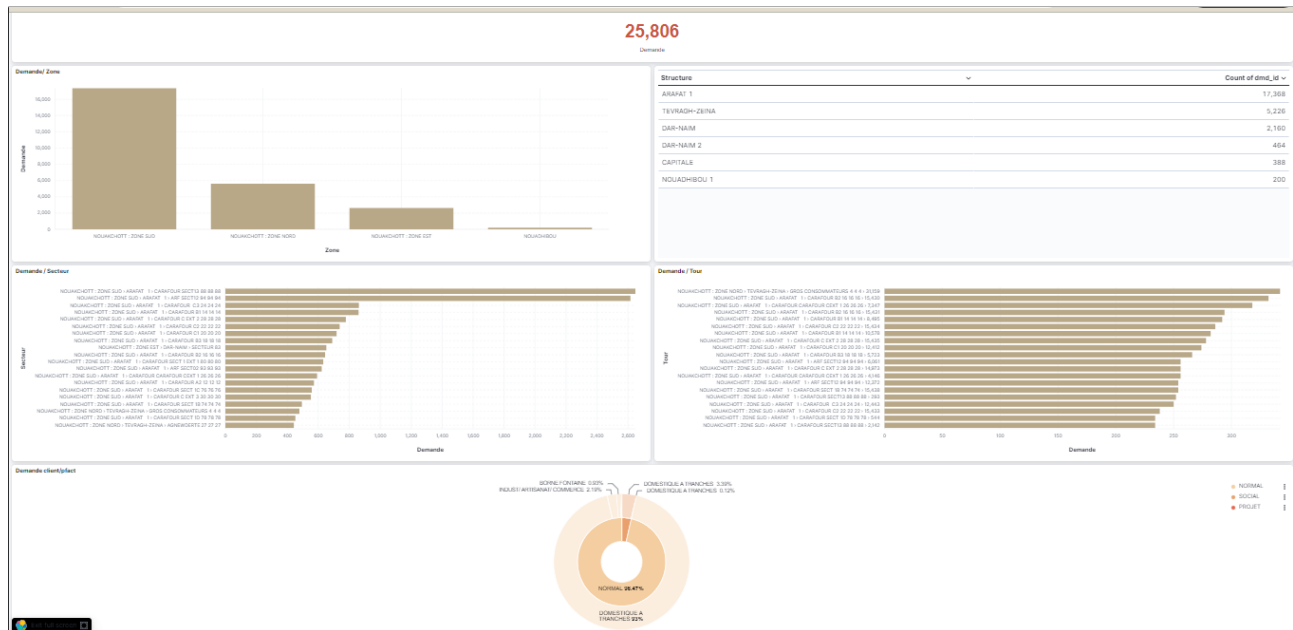


FIGURE 4.7 : Dashbord Demande"

4.1.4.6 Tableau de bord Generale

comme l'étape dernière nous allons présenter le tableau de bord générale qui contient tout les tableaux de bord précédant en utilisant le technique "Drilldown" de kibana .

- DrillDown :

Le "technique de drilldown" dans Kibana fait référence à la capacité de Kibana à permettre aux utilisateurs de naviguer plus en profondeur dans les données en passant d'une visualisation de haut niveau à des détails plus spécifiques.

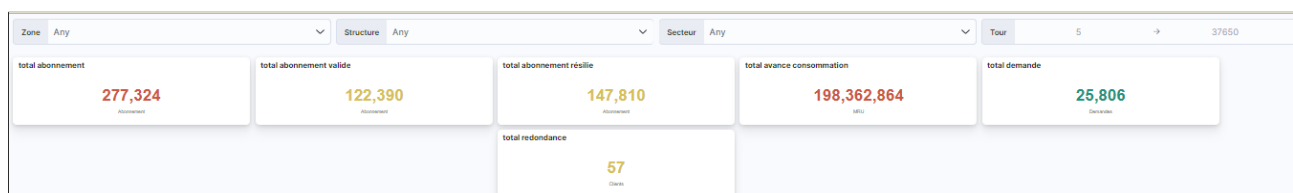


FIGURE 4.8 : Dashbord General

4.2 sprint 4 : Suivi des Relèves et des Factures

4.2.1 Objectifs du sprint 4

L'objectif de ce sprint est de prendre en charge l'analyse, qui consiste à suivre les relèves et les factures. Nous avons réparti les tâches de la manière suivante :

4.2.2 Backlog du sprint 4

Id	Fonctionnalités	Priorité	Estimation (Jour)
1	En tant que Chef de division je veux consulter l'état de tout les relèves	1	2
2	En tant que Chef de division je veux consulter l'état de tout les relèves calculées	2	2
3	En tant que Chef de division je veux consulter l'état de tout les relèves valides	3	2
4	En tant que Chef d'exploitation je veux consulter l'état de consommation totale	4	2
5	En tant que Chef du service commercial je veux consulter l'état des factures	5	2
6	En tant que Chef du service commercial je veux consulter l'état des factures valides	6	2
7	En tant que Chef du service commercial je veux consulter l'état des factures calculées	7	2
8	En tant que Chef du service commercial je veux consulter l'état des factures erronés	8	2
9	En tant que Chef du service commercial je veux consulter l'état de solde total TTC	9	2

TABLEAU 4.2 : Backlog du Sprint 4

4.2.3 Choix des indicateurs de performance

Les indicateurs de performance adoptés pour le suivi des demandes et des abonnements sont :

- Nombre de relèves par zone , structure , secteur et tour aussi par type client et date .

- Nombre des factures par zone , structure , secteur et tour aussi par type client et date .
- Pourcentage des factures par type facture.
- Pourcentage des relèves par type de client et type facturation.
- Solde total pour les factures par zone , structure , secteur et tour aussi par type client et date

.

- Pourcentage de solde total par type facture.
- Nombre des factures valides par zone , structure , secteur et tour aussi par type client et date

.

• Nombre des factures calculées par zone , structure , secteur et tour aussi par type client et date .

- Nombre des factures erronés par zone , structure , secteur et tour aussi par type client et date

.

- Nombre des relèves valides par zone , structure , secteur et tour aussi par type client et date .

- Nombre des relèves calculées par zone , structure , secteur et tour aussi par type client et date

.

- Consommation totale par zone , structure , secteur et tour aussi par type client et date .

- Pourcentage de consommation par type de client et type compteur.

4.2.4 Les Tableaux de bord relèves et factures

Dans cette partie , nous accédons via un navigateur web en utilisant l'URL "localhost :5601" a l'interface web Kibana .

- **Accée au tableau de bord initial :** Dans le menu kibana nous cliquons sur "dashbord" , une liste des tableaux de bord s'affichera et nous accedons dans notre tableau de bord intial pour les relèves et factures.

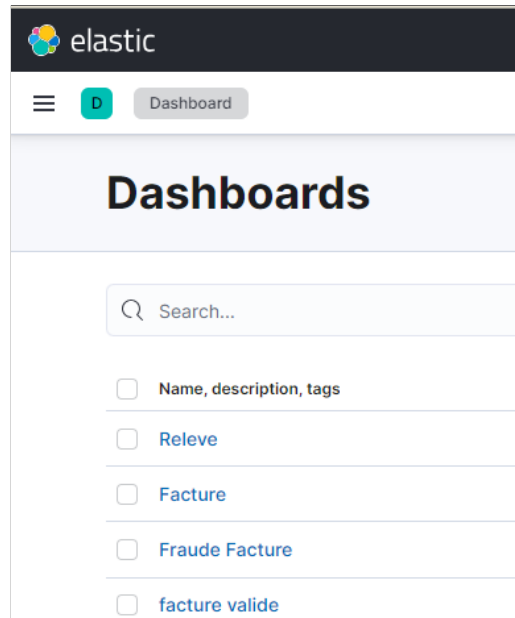


FIGURE 4.9 : Accée au Dashbord Kibana

- **Ajout de Visualisations :**

Un tableau de bord Kibana est composé de visualisations.

- **Configuration des Visualisations :**

Une fois que nous avons ajouté une visualisation, nous devons la configurer pour afficher les données souhaitées. Et cela , avec la choix de l'index Elasticsearch où nous stockons les données des relèves et factures.

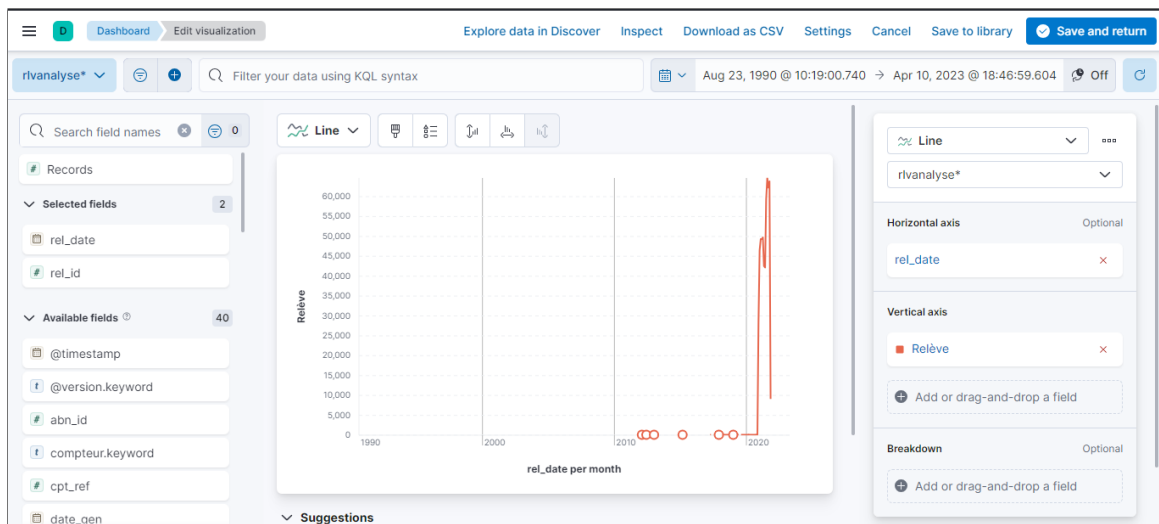


FIGURE 4.10 : Visualisation nombre de relèves par date

- **Ajout et Placement des Visualisations :**

Après la creation de visualisations , nous devons les ajouter a notre tableau de bord ,les déplacer et redimensionner pour créer la mise en page souhaitée

4.2.4.1 Tableau de bord Relève

Dans cette partie nous allons présenter le tableau de bord des relèves selon Les indicateurs de performance demandées . Pour cela nous citons les dimensions que nous avons pu dégager :

- **Dimension Client** : La dimension client permet de fournir des informations sur la clientèle. Cette dimension aide énormément à mieux comprendre les différents profils des prospects et à déterminer des critères de segmentation.
- **Dimension Compteur** : Pour commencer, nous avons extrait les données relatives au type de compteur .
- **Dimension date** : Cette dimension présente l'axe primordial de chaque analyse BI qui n'est autre que la dimension temporelle.
- **Dimension tarification** : La dimension Tarification présente un axe d'analyse crucial. Elle concerne la catégorie de tarification des services offerts.
- **Dimension zone** : C'est une sous dimension du secteur ayant la plus grande hierarchie et qui présente les zones géographiques du pays.
- **Dimension structure** : Pour chaque zone géographique, les sociétés d'eau possèdent un ensemble de structures, qui sont des divisions bien délimitées pour assurer la performance du service de distribution des eaux.
- **Dimension secteur** : Les structures sont divisées en secteurs. Un secteur est caractérisé par une moyenne de consommation pour évaluer le besoin en eau.
- **Dimension tour** : Les secteurs sont divisées en tours.

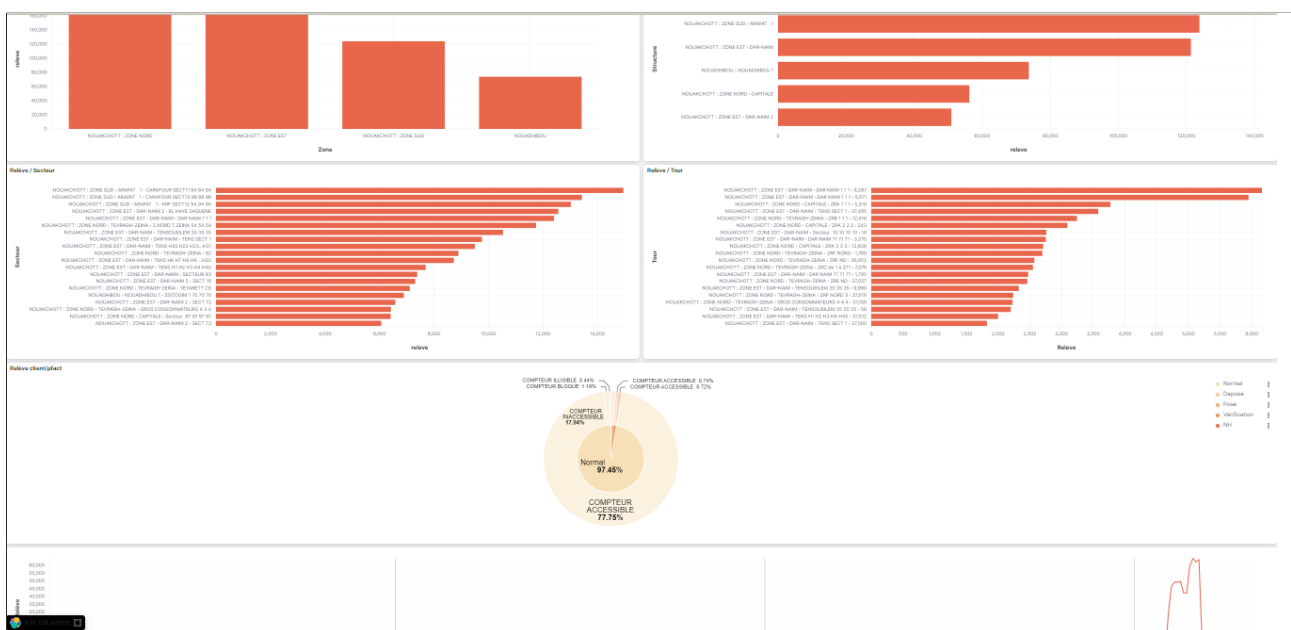


FIGURE 4.11 : Dashbord Relève

4.2.4.4 Tableau de bord consommation total

Nous allons présenter ici le tableau de bord de consommation total selon Les indicateurs de performance demandées.

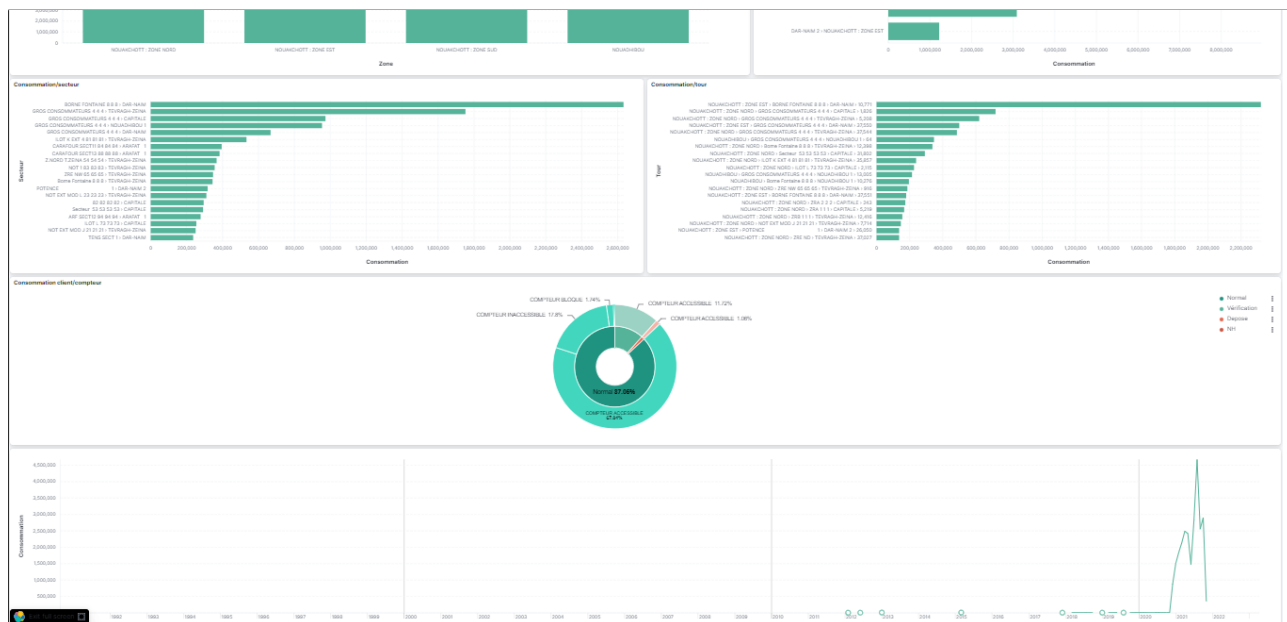


FIGURE 4.14 : Dashbord consommation Total

4.2.4.5 Tableau de bord Facture

Dans cette partie nous allons présenter le tableau de bord des factures selon Les indicateurs de performance demandées .

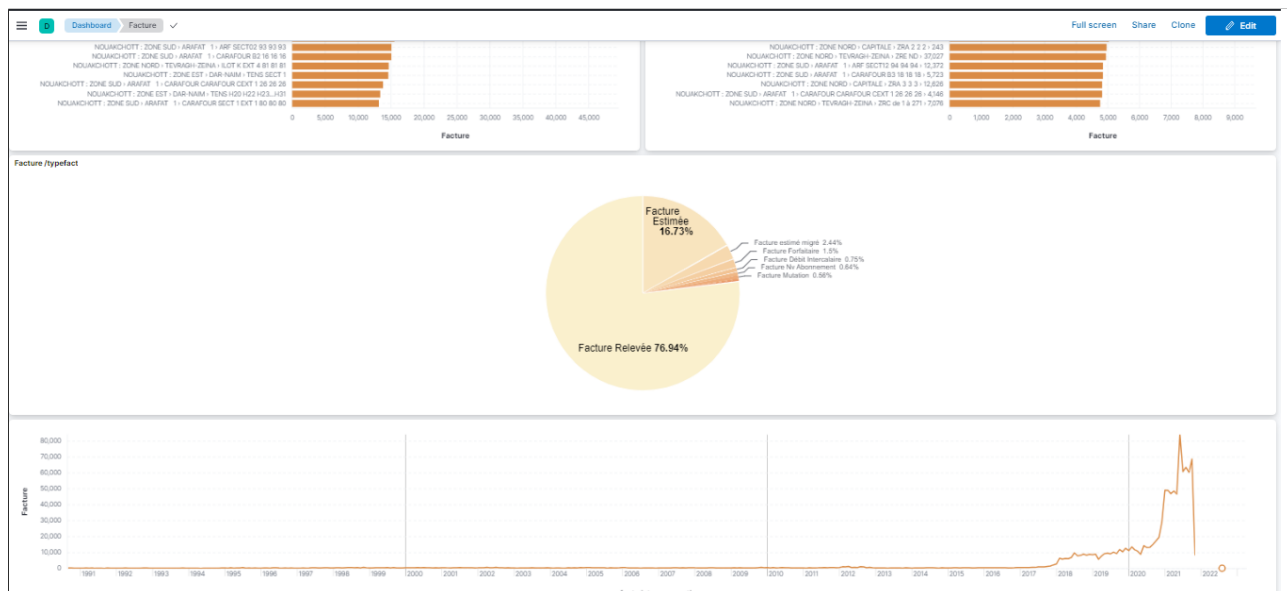


FIGURE 4.15 : Dashbord Facture

et factures".

RELEASE 3 : SUIVI DES FRAUDES ET ANOMALIES , PERSONNALISATION DES TABLEAUX DE BORD

Plan

1	Sprint 5 :Suivi des fraudes et anomalies	54
2	Sprint 6 : Personalisation des tableaux de bord	60

Introduction

Dans ce chapitre, nous allons présenter la réalisation du Sprint 5 "Suivi des fraudes et anomalies", et sprint 6 "Personnalisation des tableaux de bord". Nous allons commencer par l'identification des objectifs du Sprint, le backlog du sprint, la spécification des besoins et la réalisation.

5.1 Sprint 5 :Suivi des fraudes et anomalies

Dans cette section nous allons présenter les différents étapes de la réalisation du sprint "Suivi des fraudes et anomalies".

5.1.1 Objectifs du sprint 5

L'objectif de ce sprint est de prendre en charge l'analyse, qui consiste à suivre les anomalies relèves et les fraudes factures. Nous avons réparti les tâches de la manière suivante :

5.1.2 Backlog du sprint 5

Id	Fonctionnalités	Priorité	Estimation (Jour)
1	En tant que Chef de division je veux consulter les Surconsommations	1	2
2	En tant que Chef de division je veux consulter les consommations non logique	2	2
3	En tant que Chef de division je veux consulter les problèmes d'estiamations	3	2
4	En tant que Chef du service commercial je veux consulter les Fraudes factures	4	2

TABLEAU 5.1 : Backlog du Sprint 5

5.1.3 Les anomalies relèves et les fraudes de factures

Les anomalies de relèves ont été traitées selon les trois types d'anomalies :

- Anomalie d'estiamation : elle est lié a l'état de compteur lorsqu'il n'est pas accessible.
- Anomalie consommation non logique : lorsque la valeur de la période et la valeur de la consommation ne sont pas logiquement significatives par exemple (période > 0 et une consommation nulle).

- Anomalie surconsommation : lorsqu'une consommation d'un seul relève est supérieure au triple de la moyenne de consommation dans le secteur.

le fraude de facture ont été traitées selon les dates de periode de facture et dates factures .

- fraude date : lorsque la date periode de facture est différente a la date de facture .

5.1.4 Choix des indicateurs de performance

Les indicateurs de performance adoptés pour le suivi des anomalies et des des fraudes sont :

- Nombre de relèves ont de surconsommation par zone , structure , secteur et tour aussi par type client et date .
- Nombre de relèves ont de consommation non logique par zone , structure , secteur et tour aussi par type client et date .
- Nombre de relèves ont de problème d'estimation par zone , structure , secteur et tour aussi par type client et date .
- Pourcentage de relèves ont de surconsommation par type client et type compteur.
- Pourcentage de relèves ont de consommation non logique par type client et type compteur.
- Pourcentage de relèves ont de problème d'estimation par type client et type compteur.
- Nombre de factures ont des fraude par zone , structure , secteur et tour aussi par type client et date .
- Pourcentage de factures ont des fraudes par type facture.

5.1.5 Les Tableaux de bord Anomalies et Fraudes

Dans cette partie , nous accédons via un navigateur web en utilisant l'URL "localhost :5601" a l'interface web Kibana .

- **Accée au tableau de bord initial :** Dans le menu kibana nous cliquons sur "dashbord" , une liste des tableaux de bord s'affichera et nous accedons dans notre tableau de bord intial pour les anomalies et fraudes .

- **Ajout de Visualisations :**

Un tableau de bord Kibana est composé de visualisations.

- **Configuration des Visualisations :**

Une fois que nous avons ajouté une visualisation, nous devons la configurer pour afficher les données souhaitées. Et cela , avec la choix de l'index Elasticsearch où nous stockons les données d'anomalie et fraude.

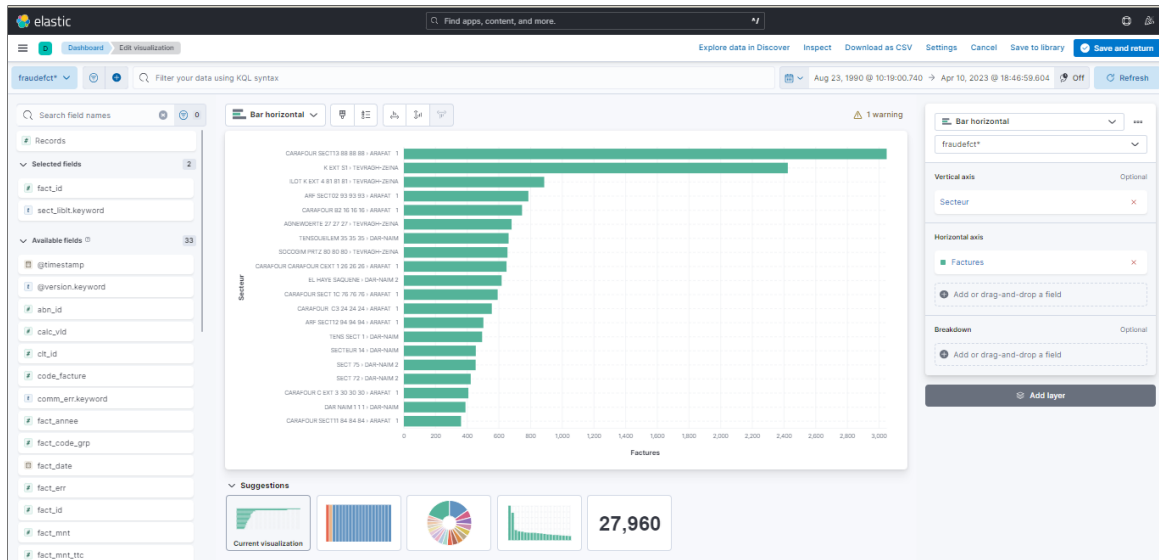


FIGURE 5.1 : Visualisation fraude facture par secteur

• Ajout et Placement des Visualisations :

Après la creation de visualisations , nous devons les ajouter a notre tableau de bord ,les déplacer et redimensionner pour créer la mise en page souhaitée

5.1.5.1 Tableau de bord Anomalie

Dans cette partie nous allons présenter le tableau de bord des anomalies relèves selon Les indicateurs de performance demandées . Pour cela nous citons les dimensions que nous avons pu dégager :

- **Dimension Client** : La dimension client permet de fournir des informations sur la clientèle. Cette dimension aide énormément à mieux comprendre les différents profils des prospects et à déterminer des critères de segmentation.
- **Dimension date** : Cette dimension présente l'axe primordial de chaque analyse BI qui n'est autre que la dimension temporelle.
- **Dimension Compteur** : Pour commencer, nous avons extrait les données relatives au type de compteur .
- **Dimension zone** : C'est une sous dimension du secteur ayant la plus grande hierarchie et qui présente les zones géographiques du pays.
- **Dimension structure** : Pour chaque zone géographique, les sociétés d'eau possèdent un ensemble de structures, qui sont des divisions bien délimitées pour assurer la performance du service de distribution des eaux.
- **Dimension secteur** : Les structures sont divisées en secteurs. Un secteur est caractérisé par une moyenne de consommation pour évaluer le besoin en eau.

- **Dimension tour** : Les secteurs sont divisées en tours.

5.1.5.2 Tableau de bord Anomalies surconsommation

Dans cette etape nous allons présenter le tableau de bord des anomalies surconsommations selon Les indicateurs de performance demandées.

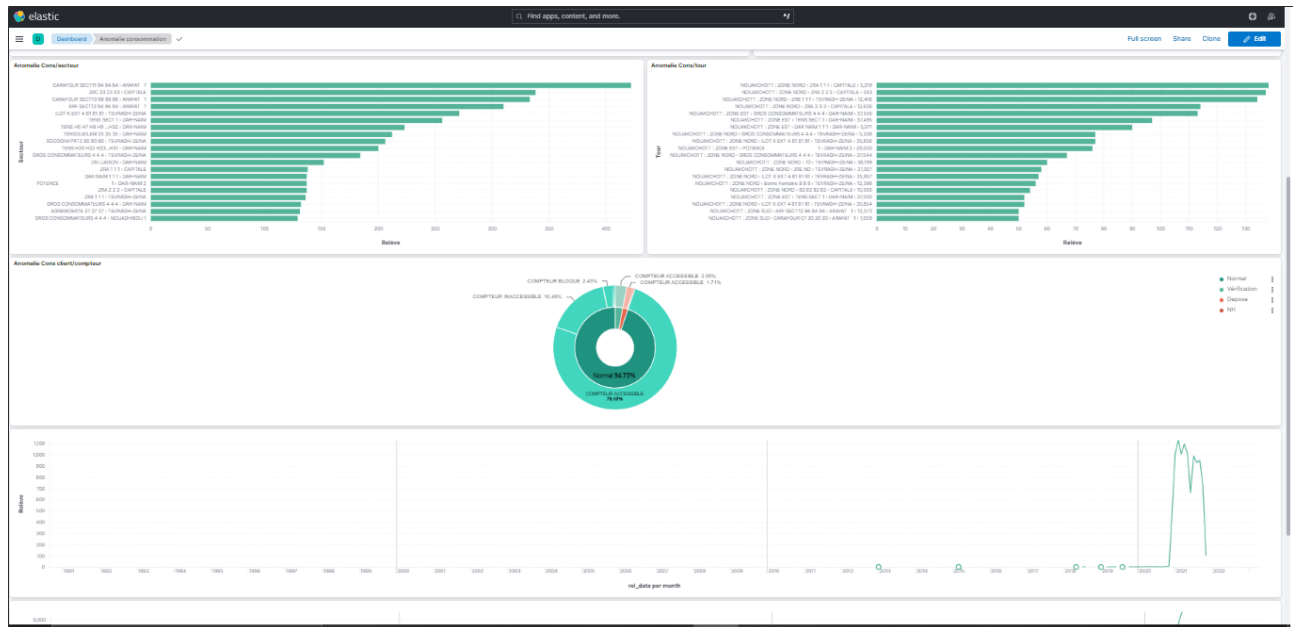


FIGURE 5.2 : Dashbord Anomalie de surconsommation

5.1.5.3 Tableau de bord Anomalies de consommation non logique

Dans cette etape nous allons présenter le tableau de bord des anomalies de consommation selon Les indicateurs de performance demandées.

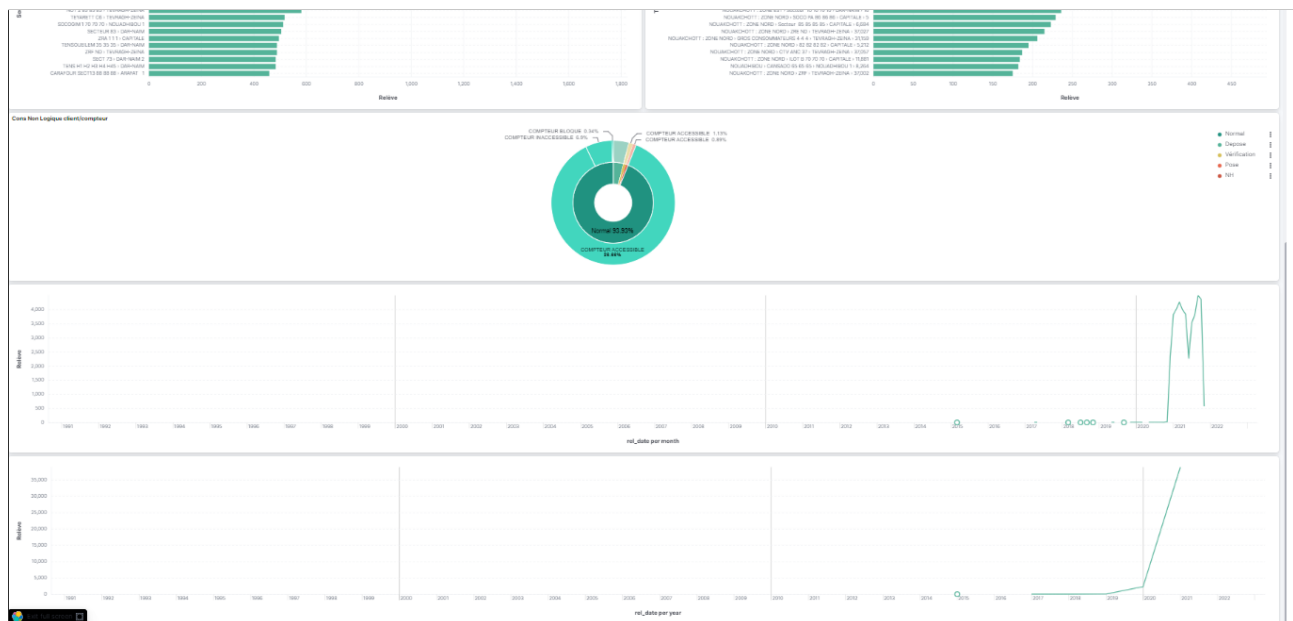


FIGURE 5.3 : Dashbord Anomalie de consommation

5.1.5.4 Tableau de bord Anomalies problème d'estimation

Dans cette etape d nous allons présenter le tableau de bord des anomalies de problème d'estimation selon Les indicateurs de performance demandées.



FIGURE 5.4 : Dashbord Anomalie de problème d'estimation

5.1.5.5 Tableau de bord Fraude

Dans cette partie nous allons présenter le tableau de bord des fraudes de facture selon Les indicateurs de performance demandées . Pour cela nous citons les dimensions que nous avons pu dégager :

- **Dimension Client :** La dimension client permet de fournir des informations sur la clientèle. Cette dimension aide énormément à mieux comprendre les différents profils des prospects et à déterminer des critères de segmentation.
- **Dimension date :** Cette dimension présente l'axe primordial de chaque analyse BI qui n'est autre que la dimension temporelle.
- **Dimension Facture :** Pour commencer, nous avons extrait les données relatives au type de facture .
- **Dimension zone :** C'est une sous dimension du secteur ayant la plus grande hierarchie et qui présente les zones géographiques du pays.
- **Dimension structure :** Pour chaque zone géographique, les sociétés d'eau possèdent un ensemble de structures, qui sont des divisions bien délimitées pour assurer la performance du service de distribution des eaux.

- **Dimension secteur** : Les structures sont divisées en secteurs. Un secteur est caractérisé par une moyenne de consommation pour évaluer le besoin en eau.
- **Dimension tour** : Les secteurs sont divisées en tours.

5.1.5.6 Tableau de bord Fraude de facture

Dans cette etape nous allons présenter le tableau de bord des fraudes factures selon Les indicateurs de performance demandées.

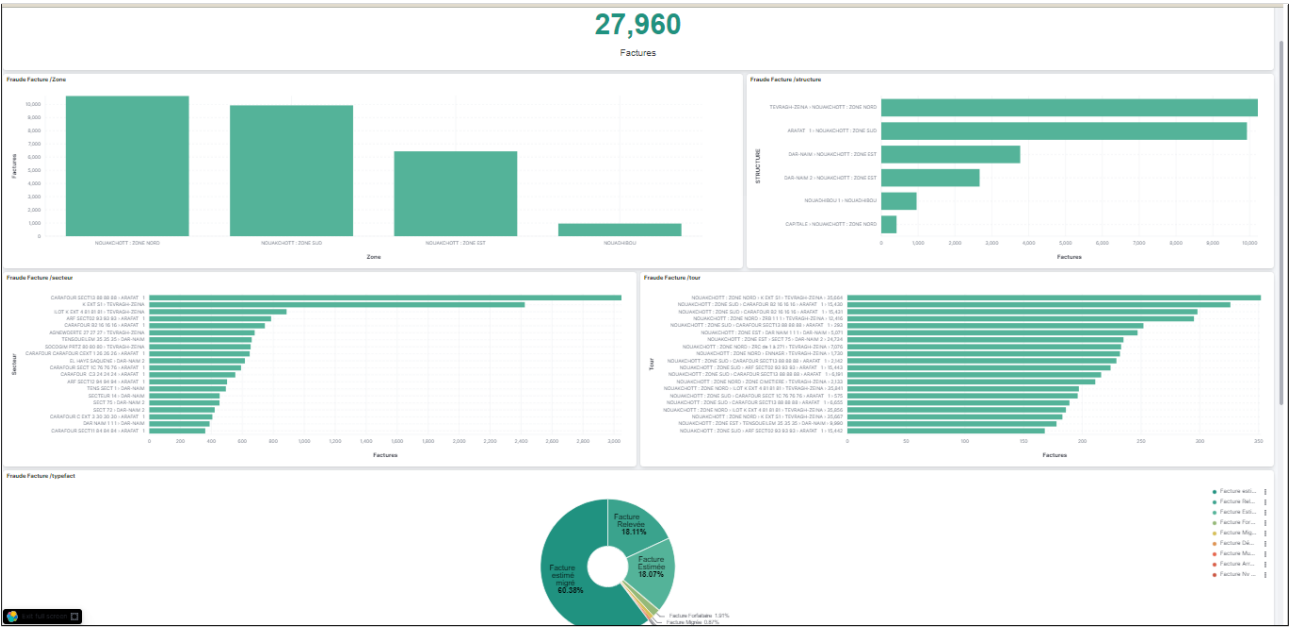


FIGURE 5.5 : Dashbord Fraude Facture

5.2 Sprint 6 : Personnalisation des tableaux de bord

Dans cette section nous allons présenter les différents étapes de la réalisation du sprint "Suivi des fraudes et anomalies".

5.2.1 Objectifs du sprint 6

L'objectif de ce sprint est de prendre en charge personnaliser les tableaux de bord . Nous avons réparti les tâches de la manière suivante :

5.2.2 Backlog du sprint 6

Id	Fonctionnalités	Priorité	Estimation (Jour)
1	En tant que Chef de division je veux avoir mon compte personnel afin de consulter just les dashbord lié a mes taches	1	3
2	En tant que Chef du service commercial je veux avoir mon compte personnel afin de consulter just les dashbord lié a mes taches	2	3
3	En tant que Chef d'Chef d'exploitation je veux avoir mon compte personnel afin de consulter just les dashbord lié a mes taches	3	3

TABLEAU 5.2 : Backlog du Sprint 6

5.2.3 Creation des Roles et des compte utilisateur

La création de rôles et d'utilisateurs dans Kibana est essentielle pour la gestion des accès et des autorisations des utilisateurs à votre cluster Elasticsearch et à vos tableaux de bord Kibana.

- **Création de Rôles dans Kibana :**

- Accéder à Kibana.
- Accéder à la Gestion des Rôles : Dans le menu de gauche, cliquons sur "Management" puis sur "Security" . Sous l'onglet "Roles" , nous pouvons créer, modifier ou supprimer des rôles.
- Créer un Nouveau Rôle : Cliquons sur "Create role" et Donnons un nom significatif à votre rôle.

- Définir les Privilèges : Dans la section "Privileges" , nous pouvons spécifier les actions autorisées pour ce rôle. Par exemple, nous pouvons autoriser la lecture ou l'écriture sur des index Elasticsearch spécifiques, la création de tableaux de bord, etc.
- **Création d'Utilisateurs dans Kibana :**
 - Accéder à la Gestion des Utilisateurs : Dans le même menu de gauche, cliquons sur "Security" , puis sur l'onglet "Users" .
 - Créez un Nouvel Utilisateur :Cliquons sur "Create user" et remplissons les informations de l'utilisateur, y compris son nom d'utilisateur, son mot de passe et les rôles auxquels il sera associé.
 - Attribuer des Rôles : nous pouvons associer un ou plusieurs rôles à un utilisateur. Ces rôles détermineront les autorisations de l'utilisateur.
 - Sauvegarder : Cliquons sur "Create" (Créer) pour enregistrer l'utilisateur.

5.2.3.1 Ceaation des roles et compte utilisateur "Chef d'exploitation"

- **Création des Rôles :**
 - Creer un role pour lire l'index qui contient les données des abonnement et on lui donne un nom "read_abonnements"

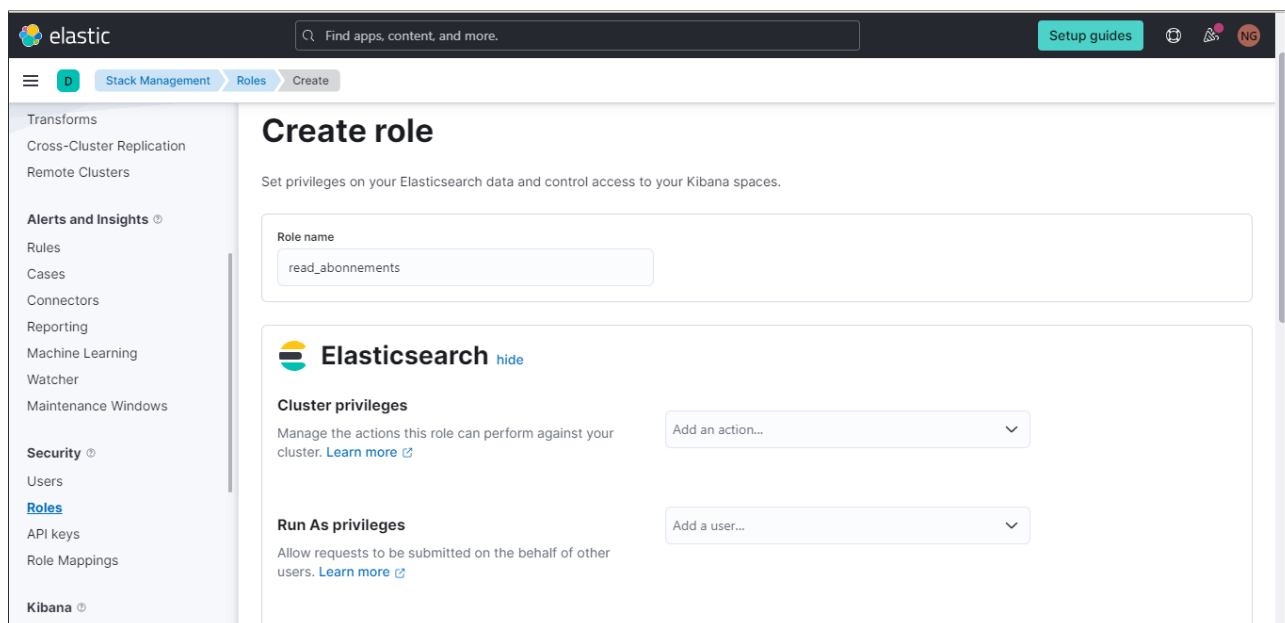


FIGURE 5.6 : Role "read_abonnements"

- Creer un role pour lire l'index qui contient les données des demandes et on lui donne un nom "read_demandes".
- **Création de compte utilisateur Chef d'exploitation :** Dans cette etape , nous allons creer un compte utilisateur pour le chef d'exploitation , on lui attribuaait les roles de lire les index

d'abonnements et des demandes qui lui permet de consulter les dashbord liés a ces index.

The screenshot shows the 'Create user' page in the Elastic Stack Management console. The page is divided into two main sections: 'Profile' and 'Password'. The 'Profile' section contains fields for 'Username' (filled with 'Chef d'exploitation'), 'Full name' (filled with 'Mohammed Ali'), and 'Email address' (filled with 'mohammedali@gmail.com'). The 'Password' section contains fields for 'Password' and 'Confirm password', both filled with masked characters. A note below the password fields states 'Password must be at least 6 characters.' The left sidebar shows the navigation menu with options like 'Stack Management', 'Users', 'Create', 'Transforms', 'Alerts and Insights', 'Security', and 'Kibana'.

FIGURE 5.7 : User "Chef d'exploitation"

5.2.3.2 Ceaion des roles et compte utilisateur "Chef du service commercial"

Comme deja vu pour la creation des roles pour le chef d'exploitation , nous allons faire de meme pour la creation compte chef service commercial.

- **Création des Rôles :**

- Créer un role pour lire l'index qui contient les données des Factures et on lui donne un nom "read_facture"
- Créer un role pour lire l'index qui contient les données des fraudes factures et on lui donne un nom "read_Fraude".

- **Création de compte utilisateur chef service commercial :** Dans cette etape , nous allons creer un compte utilisateur pour le chef service commercial , on lui attribuaait les roles de lire les index des factures et des fraudes qui lui permet de consulter les dashbord liés a ces index.

5.2.3.3 Ceaion des roles et compte utilisateur "Chef de division"

Comme deja vu pour la creation des roles pour les deux autre comptes utilisateurs , nous allons faire de meme pour la creation compte Chef de division.

- **Création des Rôles :**

- Créer un role pour lire l'index qui contient les données des relèves et on lui donne un nom "read_relève"
- Créer un role pour lire l'index qui contient les données des anomalies de relève et on lui donne

un nom "read_anomalie".

- **Création de compte utilisateur chef service commercial** : Dans cette etape , nous allons creer un compte utilisateur pour le Chef de division , on lui attribuait les roles de lire les index des factures et des fraudes qui lui permet de consulter les dashbord liés a ces index.

Conclusion

Au cours de ce chapitre, nous avons présenté la réalisation de la troisième et la dernière Release " Suivi des fraudes et anomalies , Personnalisation des Tableaux de bord". Pour ce faire, nous avons passé par l'analyse, la conception et la réalisation du sprint " Suivi des fraudes et anomalies " et par l'étude et la réalisation du sprint " Personnalisation des Tableaux de bord"

Conclusion générale

Ce rapport présente notre projet de fin d'étude au sein de Tek-Up University et marque l'aboutissement de notre parcours d'enseignement supérieur.

Le projet que nous avons entrepris, axé sur la mise en place d'une solution décisionnelle basée sur ELK Stack pour la société SIGA, a été une entreprise passionnante et enrichissante. Au cours de ce parcours, nous avons accompli plusieurs étapes importantes et atteint des objectifs significatifs.

Tout d'abord, nous avons analysé en profondeur les besoins et les lacunes de la société de distribution d'eau, en identifiant les défis auxquels elle était confrontée dans la gestion des abonnements, des demandes, des relèves et de la facturation. Cette phase d'analyse nous a permis de définir clairement les objectifs de notre projet.

Ensuite, grâce à l'utilisation de l'ELK Stack, nous avons mis en place une infrastructure robuste pour collecter, stocker et visualiser les données de manière efficace. La création de tableaux de bord interactifs et informatifs a été un aspect essentiel de notre solution, permettant aux décideurs de suivre en temps réel les performances de l'entreprise et de prendre des décisions éclairées.

Nous avons également accordé une grande importance à la sécurité des données et à la gestion des accès en mettant en place des rôles et des utilisateurs spécifiques dans Kibana, assurant ainsi la confidentialité et l'intégrité des informations sensibles.

Au fil des sprints, nous avons surmonté divers défis techniques et organisationnels, en adaptant la méthodologie Scrum à notre contexte et en assurant une collaboration efficace au sein de l'équipe.

En fin de compte, notre solution décisionnelle basée sur ELK Stack offre à la société de gestion de l'eau une meilleure visibilité, une prise de décision plus rapide et une gestion améliorée de ses opérations commerciales.

Pour l'avenir, nous envisageons d'élargir encore les fonctionnalités de la solution, d'introduire des fonctionnalités avancées telles que l'analyse prédictive dans le domaine de la distribution d'eau, et de continuer à travailler en étroite collaboration avec nos partenaires pour optimiser l'utilisation de cette puissante plateforme.

Ce projet a été une expérience formatrice pour notre équipe, nous permettant d'acquérir des compétences techniques avancées, de renforcer notre compréhension des enjeux liés à la gestion de l'eau, et de contribuer de manière significative à l'amélioration des opérations de notre client.

Nous sommes reconnaissants de l'opportunité qui nous a été offerte de mener à bien ce projet, et nous attendons avec impatience de continuer à explorer de nouvelles opportunités et à repousser les limites de l'innovation technologique.

Résumé

Le présent rapport synthétise le travail effectué dans le cadre du projet de fin d'études pour l'obtention du diplôme national d'ingénieur en informatique au sein de l'entreprise SIGA. L'objectif de ce travail est axé sur la mise en place d'une solution décisionnelle basée sur ELK Stack pour la société SIGA. Cette solution permette aux décideurs de suivre en temps réel les performances de l'entreprise et de prendre des décisions éclairées.

Mots clés : société de gestion de l'eau potable, ELK Stack, solution décisionnelle

Abstract

Keywords :