|  |  |
| --- | --- |
| Carrefour Système d'Information France.  DOSI | |
| Version : | draft |
| État : | En cours |
| Dernière modification : | 24/05/2017 |
|  |  |

| Identification du document | |
| --- | --- |
| Référence : | DAT.doc |
| Date d’édition : | 24/05/2017 |
| Date de création : | 30/03/2017 |
| Auteurs : | Ali DHAHRI (Phenix RUN) |
| Chemin d’accès : |  |
| Modèle : |  |

TABLE DES MATIERES

1 - Administration du document : 3

1.1 - Liste des acronymes utilisés 4

1.2 - Documents de référence 4

2 - Introduction 5

2.1 - Audience 5

2.2 - Objet et périmètre du document 5

2.3 - Les Environnements 5

3 - Architecture technique du Socle Technique : 6

3.1 - Rappel de l’architecture logique générale 6

3.2 - Rappel de l’architecture physique générale 8

3.3.2 - Topologie d’instanciation des composants logiciels et applicatifs 10

3.3.3 - Récapitulation des types de nœuds 12

3.4 - Architecture technique par environnement 14

3.4.1 - Répartition des nœuds physiques 14

4 - Sécurité 18

# Administration du document :

|  |  |  |
| --- | --- | --- |
| **Rédigé par** | **Vérifié par** | **Validé par** |
| Ferihane MESSAOUDI |  |  |

| Gestion des versions | | | |
| --- | --- | --- | --- |
| Version | Date | Nature de la modification | Paragraphe |
| 0.1 | 24/05/2017 | Initialisation |  |
|  | 16/08/2017 |  |  |

|  |  |  |
| --- | --- | --- |
| **Liste des destinataires** | **Exemplaire pour** | |
| **Application** | **Information** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Liste des acronymes utilisés

|  |  |  |
| --- | --- | --- |
| **Terme** | **Type** | **Description** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Documents de référence

Le présent document s’appuie sur le besoin exprimé par l’équipe PHENIX RUN (Tech & Funct) et l’équipe DOSI Legacy

# Introduction

## Audience

Ce document est à destination de toute personne ayant un rôle dans la définition ou la mise en place de l’architecture de MONITORING OMNICANAL. Le but est de fournir les informations nécessaires pour la construction de ce socle.

## Objet et périmètre du document

Ce document a pour objet:

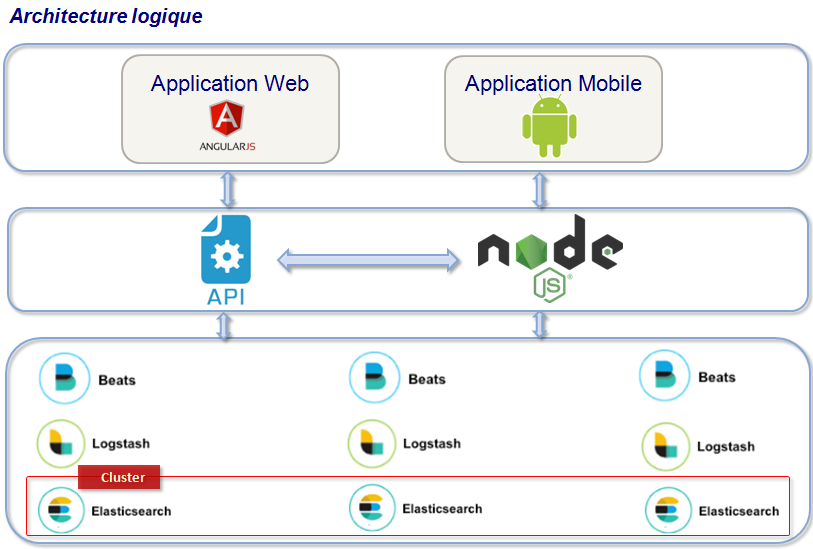
* De définir l’architecture logique générale
* De définir l’architecture physique générale
* Détailler les deux applications web et mobile

## Les Environnements

Ce document décrit la déclinaison de l’architecture technique vers les environnements PROD et PREPROD déployé en Cloud

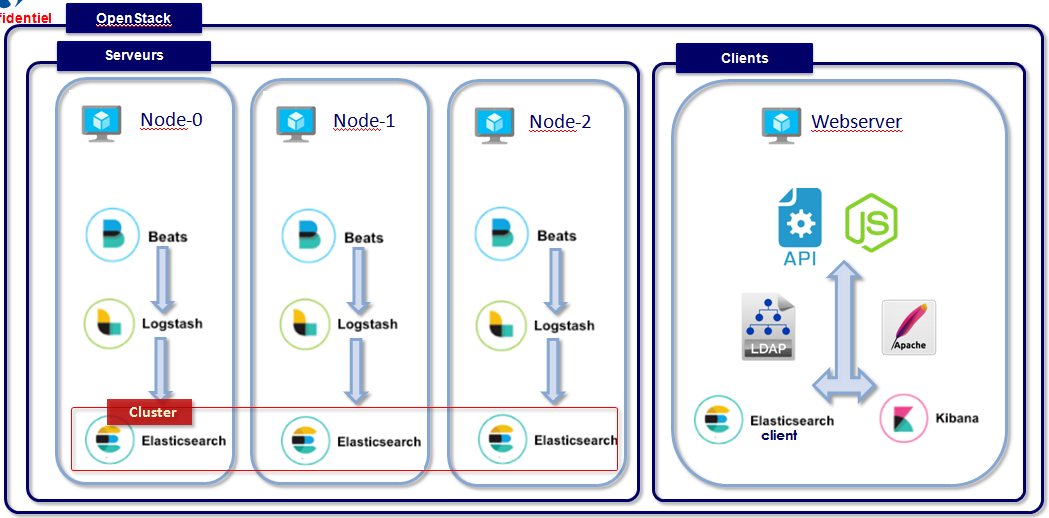
# Architecture technique générale :

## Rappel de l’architecture logique générale :



|  |  |  |
| --- | --- | --- |
| **Composant** | **Rôle** | **Version** |
| Filebeat | Outil de collecte et de transfert des logs vers logstash | v5.3.2 |
| Logstash | Outil de transformation des logs au format JSON | v5.3.2 |
| Elasticsearch | Moteur de recherche à partir des logs indexés | v5.3.2 |
| Serveur NodeJs | Serveur gérant les requêtes à partir des applications web et mobile | V6.10.2 |

## Rappel de l’architecture physique générale :



Cette architecture est composée de :

* 1 cluster contenant 3 serveurs

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom de la machine** | **Adresse IP** | **Composants** | |
| Node-0 | 192.168.199.20 | Filebeat | v5.3.2 |
| Node-1 | 192.168.199.21 | Logstash | v5.3.2 |
| Node-2 | 192.168.199.22 | Elasticsearch | v5.3.2 |

* Une machine cliente comportant :

|  |  |  |
| --- | --- | --- |
| **Composant** | **Rôle** | **Version** |
| API REST | Webservice assurant la connexion entre Elasticsearch et le serveur NodeJs |  |
| Serveur NodeJs | Gestion des requêtes Elasticsearch et renvoi des réponses | 6.10.2 |
| Elasticsearch client | Affichage des résultats renvoyés par l’API REST | 5.3.2 |
| Serveur Apache2 | Redirection vers le port 8080 du protocole https | 2.4.18 |
| Serveur LDAP | Authentification |  |

## 

## Rappel des exigences non fonctionnelles :

(A compléter ultérieurement)

## Modèle de l’architecture technique

Le modèle de l’architecture technique consiste à identifier les rôles (ou types) logiques des nœuds sans entrer dans l’aspect physique des nœuds.

A noter que lors de l’instanciation physique :

* Un nœud physique peut porter un ou plusieurs rôles logiques.
* Un rôle peut être instancié à plat dans un système hôte d’un nœud (virtuel ou physique) ou en tant que nœud virtuel ou en tant que zône d’exécution (ex : Zoning, Container, …), mais peut être aussi en tant que nœud physique.

L’identification des rôles logiques consiste à :

* Regrouper les instances de composant par rôle ou suivant leur vocation (ou motivation, raison d’être, fonction)
* Définir le mode de duplication possible : mono instance ou multi instance
* Définir le mode de fonctionnement en cas de multi instance : active/passive ou active/active avec ou sans réplication de contexte (cluster)

### Légende

TODO

Matrice de flux

### Topologie d’instanciation des composants logiciels et applicatifs

Détailler les deux applications web et mobile (maque les maquttes)

Spécifier la zone front et back par rapport au projet et non pas de manière générale

La topologique d’instanciation des composants est subdivisée en 3 niveaux de zones :

* Les zones « **FRONTEND**», qui devraient héberger les composants en interaction avec les systèmes externes notamment pour l’***ingestion des données (INGESTION)*** et l’***exposition des services de consommation des données (DATA SERVICES)***. De manière générale, les composants en « FRONTEND » devraient être compatibles à l’instanciation multiple (en ***multi instance*** et en ***multi nœud***, au moins sur 2 nœuds) afin de répondre aux contraintes de haute disponibilité et de dimensionnement (dimensionnement horizontale).
* Les zones « **BACKEND** », qui devraient héberger les composants de ***traitement (STREAM PROCESSING)*** et de ***stockage des données (OPTIMIZED DATA STORAGE)***. De même que les composants en « FRONTEND », les composants en « BACKEND » devraient être compatibles à l’instanciation multiple (en ***multi instance*** et en ***multi nœud***, au moins sur 2 nœuds) afin de répondre aux contraintes de haute disponibilité et de dimensionnement (dimensionnement horizontale).
* Les zones « **ADMIN** », qui devraient héberger les composants d’***administration de la plateforme « Online Platform » (ONLINE PLATFORM MANAGEMENT)*** et d’***administration des données « Online Data » (ONLINE DATA MANAGEMENT)***. Les composants d’administration n’ont pas de contrainte forte en termes de disponibilité mais leur instanciation multiple réduira la durée d’indisponibilité des fonctions d’administration.

Les zones « FRONTEND » et « BACKEND » de la plateforme « Online » interagissent avec la plateforme « Offline ».

#### Accès aux plusieurs instances

Dans un « Data Center (DC) », l’accès aux composants en « FRONTEND » et « ADMIN » doit tenir en compte les multiples instances déployées sur les nœuds de type « FRONT-ING-NODE », « FRONT-SVC-NODE » et éventuellement sur les nœuds de type « PLAT-ADMIN-NODE » et « DATA-ADMIN-NODE » si ceux-ci sont en multi nœuds.

L’infrastructure peut disposer d’un **mécanisme de commutation de type « L4/L4 »** pour faire :

* De « **Load Balancing** » vers les différentes instances en active/active.
* De « **Switching** » des instances en active/passive

Ce mécanisme de commutation de type « L4/L7 » peut être offert par un composant physique (Appliance, ex : ALTEON, BIG-IP) ou par un composant applicatif (ex : HA-PROXY, APACHE HTTPD, KEEP-ALIVED). Par simplification de nommage, ce composant est considéré comme un « **LOAD BALANCER** ».

A compléter

## Architecture technique par environnement

L’architecture technique par environnement consiste à décliner le modèle d’architecture technique sur l’environnement cible en terme de nœud (physique ou virtuel).

### Répartition des nœuds physiques

TODO

# Sécurité

La sécurité de la plateforme traite les thématiques suivantes :

* Identification et authentification
* Clés SSH
* Certifications SSL
* Authentification LDAP

Schématiser les interactions entre les tokens et les clés SSH lors de l’authentification

Chaque thématique est étudiée sur plusieurs niveaux :

* Niveau applicatif
* Niveau logiciel
* Niveau système
* Niveau infrastructure