



# **VITAM - Documentation d'exploitation**

*Version 0.26.1*

**VITAM**

nov. 22, 2017



<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	But de cette documentation . . . . .	1
1.2	Destinataires de ce document . . . . .	1
<b>2</b>	<b>Rappels</b>	<b>3</b>
2.1	Information concernant les licences . . . . .	3
2.2	Documents de référence . . . . .	3
2.2.1	Documents internes . . . . .	3
2.2.2	Référentiels externes . . . . .	3
2.3	Glossaire . . . . .	3
<b>3</b>	<b>Architecture de la solution logicielle VITAM</b>	<b>5</b>
<b>4</b>	<b>Exploitation globale</b>	<b>7</b>
4.1	Gestion des accès . . . . .	7
4.2	Portails d'administration . . . . .	7
4.2.1	Technique . . . . .	7
4.2.2	Fonctionnel . . . . .	7
4.3	Paramétrage & configuration . . . . .	7
4.3.1	Mise à niveau de la configuration de l'environnement . . . . .	8
4.3.1.1	Mise à jour du nombre de tenants . . . . .	8
4.3.1.2	Mise à jour des paramètres JVM . . . . .	8
4.4	Déploiement / mises à jour . . . . .	8
4.4.1	Mise à jour des certificats . . . . .	8
4.4.2	Mise à jour de la solution vitam . . . . .	8
4.5	Interruption / maintenance . . . . .	9
4.5.1	Procédure d'arrêt . . . . .	9
4.5.2	Procédure de démarrage . . . . .	9
4.5.3	Procédure de statut . . . . .	10
4.5.4	Mise à jour des certificats . . . . .	10
4.6	Batches et traitements . . . . .	10
4.6.1	Curator . . . . .	10
4.6.2	Sécurisation des journaux d'opérations . . . . .	11
4.6.3	Sécurisation des cycles de vie . . . . .	11
<b>5</b>	<b>Suivi de l'état du système</b>	<b>13</b>
5.1	Veille et patchs sécurité . . . . .	13
5.2	Métriques . . . . .	13

5.2.1	Configuration	13
5.2.1.1	Registres	13
5.2.1.2	Reporters	14
5.2.1.3	Fichier de configuration	14
5.2.2	Métier	14
5.2.3	Métriques techniques	14
5.2.3.1	Métriques système critiques	14
5.2.3.2	Indicateurs de SLA	14
5.2.3.3	Indicateurs de performance	14
5.2.4	Visualisation	14
5.2.4.1	Discover	15
5.2.4.2	Visualize	15
5.2.4.3	Dashboards	16
5.3	API de de supervision	17
5.3.1	Détail	17
5.3.1.1	/admin/v1/status	17
5.3.1.2	/admin/v1/version	17
5.3.1.3	/admin/v1/autotest	18
5.4	Logs	19
5.4.1	Changement des règles de log	20
5.5	Audit	21
5.6	Gestion de la capacité	21
5.7	Suivi de l'état de sécurité	21
5.8	Alerting	21
5.8.1	Système	21
5.8.2	Applicatif	21
<b>6</b>	<b>Exploitation des composants de la solution logicielle VITAM</b>	<b>23</b>
6.1	Généralités	23
6.2	Fichiers communs	23
6.2.1	Fichier /vitam/conf/<composant>/sysconfig/java_opts	23
6.2.2	Fichier /vitam/conf/<composant>/logback.xml	24
6.2.3	Fichier /vitam/conf/<composant>/logback-access.xml	24
6.2.4	Fichier /vitam/conf/<composant>/jetty-config.xml	26
6.2.5	Fichier /vitam/conf/<composant>/logbook-client.conf	31
6.2.6	Fichier /vitam/conf/<composant>/server-identity.conf	31
6.2.7	Fichier /vitam/conf/<composant>/antisamy-esapi.xml	31
6.2.8	Fichier /vitam/conf/<composant>/vitam.conf	45
6.2.9	Fichier /vitam/conf/<composant>/vitam.metrics.conf	45
6.2.10	Fichier /vitam/conf/<composant>/java.security	46
6.3	Composants	46
6.3.1	access external	46
6.3.1.1	Présentation	46
6.3.1.2	Configuration / fichiers utiles	46
6.3.1.2.1	Fichier access-external.conf	46
6.3.1.2.2	Fichier access-internal-client.conf	46
6.3.1.2.3	Fichier shiro.ini	47
6.3.1.2.4	Fichier functional-administration-client.conf	47
6.3.1.3	Opérations	47
6.3.2	access-internal	48
6.3.2.1	Présentation du composant	48
6.3.2.2	Configuration / fichiers utiles	48
6.3.2.2.1	Fichier access.conf	48
6.3.2.2.2	Fichier storage-client.conf	48

6.3.2.2.3	Fichier metadata-client.conf	48
6.3.2.3	Opérations	49
6.3.3	Cerebro	49
6.3.3.1	Présentation	49
6.3.3.2	Configuration / fichiers utiles	49
6.3.3.3	Opérations	49
6.3.4	common-plugin	50
6.3.4.1	Présentation du composant	50
6.3.4.2	Classes utiles	50
6.3.4.2.1	Classe Item Status	50
6.3.4.2.2	Classe VitamAutoCloseable	50
6.3.4.2.3	Classe ParameterHelper	50
6.3.4.2.4	Classe VitamParameter	51
6.3.4.2.5	Classe ProcessingException	51
6.3.4.2.6	Classe IOParameter	51
6.3.4.2.7	Classe ProcessingUri	51
6.3.4.2.8	Classe UriPrefix	51
6.3.4.2.9	Classe AbstractWorkerParameters	51
6.3.4.2.10	Classe DefaultWorkerParameters	51
6.3.4.2.11	Classe WorkerParameterName	51
6.3.4.2.12	Classe WorkerParameters	51
6.3.4.2.13	Classe WorkerParametersDeserializer	51
6.3.4.2.14	Classe WorkerParametersFactory	52
6.3.4.2.15	Classe WorkerParametersSerializer	52
6.3.4.2.16	Interface HandlerIO	52
6.3.4.2.17	Classe WorkerAction	52
6.3.4.2.18	Classe HandlerIOImpl	52
6.3.5	consul	52
6.3.5.1	Présentation	52
6.3.5.1.1	Cas serveur	53
6.3.5.1.2	Cas agent	53
6.3.5.2	Configuration / fichiers utiles	53
6.3.5.2.1	Cas des applicatifs monitorés par Consul	53
6.3.5.2.1.1	Fichier /vitam/conf/consul/service-<composant>.json	53
6.3.5.3	Opérations	54
6.3.6	elasticsearch chaîne de log	55
6.3.6.1	Présentation	55
6.3.6.2	Configuration / fichiers utiles	55
6.3.6.2.1	Fichier logging.yml	55
6.3.6.2.2	Fichier elasticsearch.yml	57
6.3.6.2.3	Fichier sysconfig/elasticsearch	59
6.3.6.2.4	Fichier /usr/lib/tmpfiles.d/elasticsearch-data.conf	60
6.3.6.3	Opérations	60
6.3.7	elasticsearch Vitam	61
6.3.7.1	Présentation	61
6.3.7.2	Configuration / fichiers utiles	61
6.3.7.2.1	Fichier logging.yml	61
6.3.7.2.2	Fichier elasticsearch.yml	63
6.3.7.2.3	Fichier sysconfig/elasticsearch	66
6.3.7.2.4	Fichier /usr/lib/tmpfiles.d/elasticsearch-data.conf	67
6.3.7.3	Opérations	67
6.3.8	functional administration	68
6.3.8.1	Présentation	68
6.3.8.2	Configuration / fichiers utiles	68

6.3.8.2.1	Fichier functional-administration.conf	68
6.3.8.3	Opérations	69
6.3.9	ihm-demo	69
6.3.9.1	Présentation	69
6.3.9.2	Configuration / fichiers utiles	70
6.3.9.2.1	Fichier access-external-client.conf	70
6.3.9.2.2	Fichier ihm-demo.conf	70
6.3.9.2.3	Fichier ingest-external-client.conf	70
6.3.9.2.4	Fichier shiro.ini	71
6.3.9.3	Opérations	72
6.3.10	ihm-recette	73
6.3.10.1	Présentation	73
6.3.10.2	Configuration / fichiers utiles	73
6.3.10.2.1	Fichier access-external-client.conf	73
6.3.10.2.2	Fichier ihm-recette.conf	73
6.3.10.2.3	Fichier ihm-recette-client.conf	74
6.3.10.2.4	Fichier ingest-external-client.conf	74
6.3.10.2.5	Fichier functional-administration-client.conf	75
6.3.10.2.6	Fichier shiro.ini	75
6.3.10.2.7	Fichier storage-client.conf	76
6.3.10.2.8	Fichier storage-offer.conf	76
6.3.10.2.9	Fichier tnr.conf	76
6.3.10.3	Opérations	76
6.3.11	ingest-external	77
6.3.11.1	Présentation	77
6.3.11.2	Configuration / fichiers utiles	77
6.3.11.2.1	Fichier ingest-external.conf	78
6.3.11.2.2	Fichier ingest-internal-client.conf	78
6.3.11.3	Opérations	78
6.3.12	ingest-internal	79
6.3.12.1	Présentation	79
6.3.12.2	Configuration / fichiers utiles	79
6.3.12.2.1	Fichier ingest-internal.conf	79
6.3.12.3	Opérations	79
6.3.13	log server	80
6.3.13.1	Présentation	80
6.3.13.2	Configuration / fichiers utiles	80
6.3.13.3	Opérations	80
6.3.14	logbook	81
6.3.14.1	Présentation	81
6.3.14.2	Configuration / fichiers utiles	81
6.3.14.2.1	Fichier logbook.conf	81
6.3.14.3	Opérations	81
6.3.15	metadata	82
6.3.15.1	Présentation	82
6.3.15.2	Configuration / fichiers utiles	82
6.3.15.2.1	Fichier metadata.conf	82
6.3.15.3	Opérations	83
6.3.16	mongoC	83
6.3.16.1	Présentation	83
6.3.16.2	Configuration / fichiers utiles	83
6.3.16.3	Opérations	84
6.3.17	mongoD	84
6.3.17.1	Présentation	84

6.3.17.2	Configuration / fichiers utiles	84
6.3.17.3	Opérations	84
6.3.18	mongoS	85
6.3.18.1	Présentation	85
6.3.18.2	Configuration / fichiers utiles	85
6.3.18.3	Opérations	85
6.3.19	processing	86
6.3.19.1	Présentation	86
6.3.19.2	Configuration / fichiers utiles	86
6.3.19.2.1	Fichier processing.conf	86
6.3.19.2.2	Fichier version.conf	86
6.3.19.2.3	Fichier storage-client.conf	87
6.3.19.3	Opérations	87
6.3.20	Security internal	87
6.3.20.1	Présentation	87
6.3.20.2	Configuration / fichiers utiles	87
6.3.20.2.1	Fichier security-internal.conf	88
6.3.20.3	Opérations	88
6.3.21	siegfried	88
6.3.21.1	Présentation	88
6.3.21.2	Configuration / fichiers utiles	89
6.3.21.3	Opérations	89
6.3.22	storage-engine	89
6.3.22.1	Présentation	89
6.3.22.2	Configuration / fichiers utiles	90
6.3.22.2.1	Fichier driver-location.conf	90
6.3.22.2.2	Fichier driver-mapping.conf	90
6.3.22.2.3	Fichier static-offer.json	90
6.3.22.2.4	Fichier static-strategy.json	90
6.3.22.2.5	Fichier storage-engine.conf	91
6.3.22.3	Opérations	91
6.3.23	offer	91
6.3.23.1	Présentation	91
6.3.23.2	Configuration / fichiers utiles	92
6.3.23.2.1	Fichier default-offer.conf	92
6.3.23.2.2	Fichier default-storage.conf	92
6.3.23.3	Opérations	92
6.3.24	worker	93
6.3.24.1	Présentation	93
6.3.24.2	Configuration / fichiers utiles	93
6.3.24.2.1	Fichier format-identifiers.conf	93
6.3.24.2.2	Fichier functional-administration-client.conf.j2	93
6.3.24.2.3	Fichier metadata-client.conf	93
6.3.24.2.4	Fichier storage-client.conf	94
6.3.24.2.5	Fichier version.conf	94
6.3.24.2.6	Fichier worker.conf	94
6.3.24.3	Opérations	95
6.3.25	workspace	95
6.3.25.1	Présentation	95
6.3.25.2	Configuration / fichiers utiles	95
6.3.25.2.1	Fichier workspace.conf	95
6.3.25.3	Opérations	96

7.1	Prérequis . . . . .	97
7.2	Intégration du certificat client de l'application externe . . . . .	97
7.3	Déploiement des keystores . . . . .	97
7.3.1	Vitam n'est pas encore déployé . . . . .	97
7.3.2	Vitam est déjà déployé . . . . .	97
<b>8</b>	<b>Aide à l'exploitation</b>	<b>99</b>
8.1	Analyse de premier niveau . . . . .	99
8.1.1	Etat par Consul . . . . .	99
8.1.2	Etat par Kibana . . . . .	100
<b>9</b>	<b>Questions Fréquemment Posées</b>	<b>101</b>
9.1	Présentation . . . . .	101
9.2	Retour d'expérience / cas rencontrés . . . . .	101
<b>10</b>	<b>Exploitation par composant</b>	<b>103</b>
10.1	Access . . . . .	103
10.1.1	Introduction . . . . .	103
10.2	Common . . . . .	103
10.2.1	Présentation . . . . .	103
10.2.2	Format Identifiers . . . . .	103
10.2.2.1	Configuration des services d'identification des formats . . . . .	103
10.3	Functional administration . . . . .	104
10.3.1	Présentation . . . . .	104
10.4	Ingest . . . . .	104
10.4.1	Introduction . . . . .	104
10.4.2	ingest-external-exploitation . . . . .	104
10.4.3	ingest-internal-exploitation . . . . .	105
10.5	Logbook . . . . .	105
10.5.1	Présentation . . . . .	105
10.6	Metadata . . . . .	105
10.6.1	Présentation . . . . .	105
10.7	Processing . . . . .	105
10.7.1	Introduction . . . . .	105
10.7.1.1	But de cette documentation . . . . .	105
10.7.2	Processing . . . . .	105
10.7.2.1	Configuration du worker . . . . .	106
10.7.2.2	Supervision du service . . . . .	106
10.8	Storage . . . . .	106
10.8.1	Introduction . . . . .	106
10.8.1.1	But de cette documentation . . . . .	106
10.8.2	Storage Engine . . . . .	106
10.8.2.1	Configuration du moteur de stockage . . . . .	107
10.8.2.2	Configuration du driver de l'offre de stockage par défaut . . . . .	108
10.8.2.3	Supervision du service . . . . .	108
10.8.3	Storage Offer Default . . . . .	108
10.8.3.1	Configuration de l'offre de stockage . . . . .	108
10.8.3.2	Supervision du service . . . . .	109
10.9	Technical administration . . . . .	109
10.9.1	Présentation . . . . .	109
10.10	Worker . . . . .	109
10.10.1	Introduction . . . . .	109
10.10.1.1	But de cette documentation . . . . .	109
10.10.2	Storage Engine . . . . .	109



10.10.2.1 Configuration du worker . . . . .	109
10.10.2.2 Supervision du service . . . . .	110
10.11 Workspace . . . . .	110
10.11.1 Présentation . . . . .	110
<b>11 Annexes</b>	<b>111</b>
<b>Index</b>	<b>117</b>



---

# Introduction

---

## 1.1 But de cette documentation

Ce document a pour but de permettre de fournir à une équipe d'exploitants de VITAM les procédures et informations utiles et nécessaires au bon fonctionnement de la solution logicielle.

## 1.2 Destinataires de ce document

Ce document s'adresse à des exploitants du secteur informatique ayant de bonnes connaissances en environnement Linux.



---

## Rappels

---

### 2.1 Information concernant les licences

La solution logicielle *VITAM* est publiée sous la licence *CeCILL 2.1*<sup>1</sup> ; la documentation associée (comprenant le présent document) est publiée sous *Licence Ouverte V2.0*<sup>2</sup>.

### 2.2 Documents de référence

#### 2.2.1 Documents internes

Tableau 2.1 – Documents de référence VITAM

Nom	Lien
<i>DAT</i>	<a href="http://www.programmevitam.fr/ressources/DocCourante/html/archi">http://www.programmevitam.fr/ressources/DocCourante/html/archi</a>
<i>DIN</i>	<a href="http://www.programmevitam.fr/ressources/DocCourante/html/installation">http://www.programmevitam.fr/ressources/DocCourante/html/installation</a>
<i>DEX</i>	<a href="http://www.programmevitam.fr/ressources/DocCourante/html/exploitation">http://www.programmevitam.fr/ressources/DocCourante/html/exploitation</a>
Release notes	

#### 2.2.2 Référentiels externes

### 2.3 Glossaire

**API** Application Programming Interface

**BDD** Base De Données

**COTS** Component Off The Shelves ; il s’agit d’un composant “sur étagère”, non développé par le projet *VITAM*, mais intégré à partir d’un binaire externe. Par exemple : MongoDB, ElasticSearch.

**DAT** Dossier d’Architecture Technique

**DEX** Dossier d’EXploitation

**DIN** Dossier d’Installation

---

1. [http://www.cecill.info/licences/Licence\\_CeCILL\\_V2.1-fr.html](http://www.cecill.info/licences/Licence_CeCILL_V2.1-fr.html)

2. <https://www.etalab.gouv.fr/wp-content/uploads/2017/04/ETALAB-Licence-Ouverte-v2.0.pdf>

**DNSSEC** *Domain Name System Security Extensions* est un protocole standardisé par l'IETF permettant de résoudre certains problèmes de sécurité liés au protocole DNS. Les spécifications sont publiées dans la RFC 4033 et les suivantes (une version antérieure de DNSSEC n'a eu aucun succès). [Définition DNSSEC](#)<sup>3</sup>

**DUA** Durée d'Utilité Administrative

**IHM** Interface Homme Machine

**JRE** Java Runtime Environment ; il s'agit de la machine virtuelle Java permettant d'y exécuter les programmes compilés pour.

**JVM** Java Virtual Machine ; Cf. [JRE](#)

**MitM** L'attaque de l'homme du milieu (HDM) ou *man-in-the-middle attack* (MITM) est une attaque qui a pour but d'intercepter les communications entre deux parties, sans que ni l'une ni l'autre ne puisse se douter que le canal de communication entre elles a été compromis. Le canal le plus courant est une connexion à Internet de l'internaute lambda. L'attaquant doit d'abord être capable d'observer et d'intercepter les messages d'une victime à l'autre. L'attaque « homme du milieu » est particulièrement applicable dans la méthode d'échange de clés Diffie-Hellman, quand cet échange est utilisé sans authentification. Avec authentification, Diffie-Hellman est en revanche invulnérable aux écoutes du canal, et est d'ailleurs conçu pour cela. [Explication](#)<sup>4</sup>

**NoSQL** Base de données non-basée sur un paradigme classique des bases relationnelles. [Définition NoSQL](#)<sup>5</sup>

**OAIS** *Open Archival Information System*, acronyme anglais pour Systèmes de transfert des informations et données spatiales – Système ouvert d'archivage d'information (SOAI) - Modèle de référence.

**PDMA** Perte de Données Maximale Admissible ; il s'agit du pourcentage de données stockées dans le système qu'il est acceptable de perdre lors d'un incident de production.

**PKI** Une infrastructure à clés publiques (ICP) ou infrastructure de gestion de clés (IGC) ou encore Public Key Infrastructure (PKI), est un ensemble de composants physiques (des ordinateurs, des équipements cryptographiques logiciels ou matériel type HSM ou encore des cartes à puces), de procédures humaines (vérifications, validation) et de logiciels (système et application) en vue de gérer le cycle de vie des certificats numériques ou certificats électroniques. [Définition PKI](#)<sup>6</sup>

**REST** REpresentational State Transfer : type d'architecture d'échanges. Appliqué aux services web, en se basant sur les appels http standard, il permet de fournir des API dites "RESTful" qui présentent un certain nombre d'avantages en termes d'indépendance, d'universalité, de maintenabilité et de gestion de charge. [Définition REST](#)<sup>7</sup>

**RPM** Red Hat Package Manager ; il s'agit du format de packets logiciels nativement utilisé par les distributions CentOS (entre autres)

**SAE** Système d'Archivage Électronique

**SEDA** Standard d'Échange de Données pour l'Archivage

**SIA** Système d'Informations Archivistique

**TNR** Tests de Non-Régression

**VITAM** Valeurs Immatérielles Transférées aux Archives pour Mémoire

---

3. [https://fr.wikipedia.org/wiki/Domain\\_Name\\_System\\_Security\\_Extensions](https://fr.wikipedia.org/wiki/Domain_Name_System_Security_Extensions)

4. [https://fr.wikipedia.org/wiki/Attaque\\_de\\_l'homme\\_du\\_milieu](https://fr.wikipedia.org/wiki/Attaque_de_l'homme_du_milieu)

5. <https://fr.wikipedia.org/wiki/NoSQL>

6. [https://fr.wikipedia.org/wiki/Infrastructure\\_%C3%A0\\_cl%C3%A9s\\_publicques](https://fr.wikipedia.org/wiki/Infrastructure_%C3%A0_cl%C3%A9s_publicques)

7. [https://fr.wikipedia.org/wiki/Representational\\_state\\_transfer](https://fr.wikipedia.org/wiki/Representational_state_transfer)

# Architecture de la solution logicielle VITAM

Le schéma ci-dessous représente une solution *VITAM* :

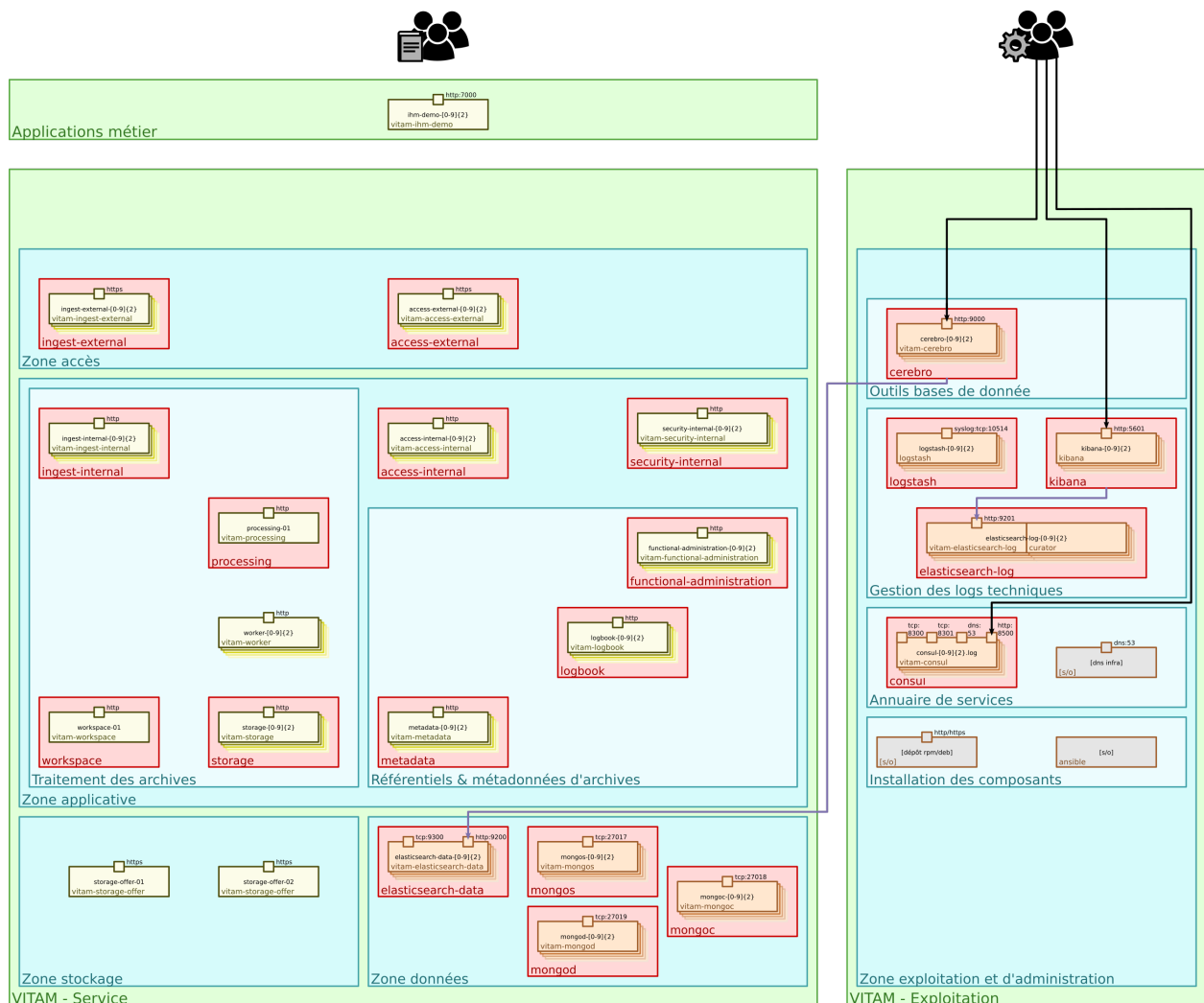


Fig. 3.1 – Vue d'ensemble d'un déploiement VITAM : zones, composants

Voir aussi :

Se référer au *DAT* (et notamment le chapitre dédié à l'architecture technique) pour plus de détails, en particulier concernant les flux entre les composants.



---

# Exploitation globale

---

## 4.1 Gestion des accès

### 1. API

La gestion des accès à l'API se fait via les granted stores (cf. *Configuration / fichiers utiles* (page 46) pour access-external et *Configuration / fichiers utiles* (page 79) pour ingest-external).

### 2. IHM d'admin

Dans cette version, la gestion des utilisateurs se fait par configuration d'un fichier plat (cf. *Opérations* (page 72)).

## 4.2 Portails d'administration

### 4.2.1 Technique

Aucun portail d'administration technique n'est prévu dans cette version de VITAM.

### 4.2.2 Fonctionnel

Le portail d'administration fonctionnel est intégré à l'IHM Démo dans cette version de VITAM (cf. *Présentation* (page 69)).

## 4.3 Paramétrage & configuration

L'étape de paramétrage et la configuration sont essentiellement liées à la mise en place ou la mise à niveau de la solution logicielle *VITAM* (ansible / inventaire).

### Voir aussi :

Plus d'informations, et notamment les paramètres d'installation, sont disponibles dans le *DIN*.

## 4.3.1 Mise à niveau de la configuration de l'environnement

### 4.3.1.1 Mise à jour du nombre de tenants

---

**Note :** se référer au *DIN*

---

### 4.3.1.2 Mise à jour des paramètres JVM

---

**Note :** se référer au *DIN*

---

**Prudence :** limitation technique à ce jour ; il n'est pas possible de définir des variables JVM différentes pour des composants colocalisés sur une même partition.

## 4.4 Déploiement / mises à jour

### 4.4.1 Mise à jour des certificats

Pour mettre à jour les certificats (avant expiration par exemple), il suffit de mettre à jour les stores dans `environments/keystores` Puis de lancer leur redéploiement via cette commande ansible :

```
# Si le mot de passe du vault n'est pas renseigné dans le fichier vault_pass.txt
ansible-playbook ansible-vitam/vitam.yml -i environments/<fichier d'inventaire> --ask-
↪vault-pass --tags update_vitam_certificates
ansible-playbook ansible-vitam-extra/extra.yml -i environments/<fichier d'inventaire> ↵
↪--ask-vault-pass --tags update_vitam_certificates
# Si le mot de passe du vault est renseigné dans le fichier vault_pass.txt
ansible-playbook ansible-vitam/vitam.yml -i environments/<fichier d'inventaire> --
↪vault-password-file vault_pass.txt --tags update_vitam_certificates
ansible-playbook ansible-vitam-extra/extra.yml -i environments/<fichier d'inventaire> ↵
↪--vault-password-file vault_pass.txt --tags update_vitam_certificates
```

### 4.4.2 Mise à jour de la solution vitam

Pour la mise à jour de la solution logicielle *VITAM* (tout comme pour sa première installation), se référer au *DIN*, ainsi qu'à la "release note" associée à toute version.

Ces documents détaillent les pré-requis, la configuration des fichiers pour effectuer un déploiement. Le *DIN* explique également comment valider une montée de version applicative de VITAM.

**Voir aussi :**

Plus d'informations, et notamment les paramètres d'installation, sont disponibles dans le *DIN*.

## 4.5 Interruption / maintenance

### 4.5.1 Procédure d'arrêt

Cette procédure décrit la cinématique d'arrêt de la plate-forme ; les commandes pour chaque composant sont détaillées plus bas dans ce même document.

**Note :** Il est conseillé de n'arrêter les bases de données qu'une fois les composants applicatifs arrêtés.

Tableau 4.1 – Cinématique d'arrêt de VITAM

Composant	Ordre d'arrêt	OK ?
ihm-demo	1	
ingest-external	2	
ingest-internal	3	
access-external	4	
access-internal	5	
worker	6	
processing	7	
workspace	8	
functional-administration	9	
logbook	10	
metadata	11	
storage-engine	12	
storage-offer-xxx	13	
mongos	14	
mongod	15	
mongoc	16	
elasticsearch vitam	17	
kibana	18	
logstash	19	
elasticsearch logs	20	
consul	21	

Un playbook ansible d'arrêt de VITAM est fourni, sous `deployment/ansible-vitam-exploitation` (fichier de `playbook stop_vitam.yml`), pour réaliser de façon automatisée les actions nécessaires.

**Avertissement :** ce script, en l'état, permet un *EMERGENCY BREAK*, autrement dit un arrêt brutal des composants, ne permettant pas de garantir, à l'issue, une cohérence des données.

**Note :** Une confirmation est demandée pour lancer ce script d'arrêt de la solution logicielle VITAM.

### 4.5.2 Procédure de démarrage

Le pré-requis est le bon fonctionnement des partitions hébergeant la solution logicielle *VITAM*.

Tableau 4.2 – Cinématique de démarrage de VITAM

Composant	Ordre	OK ?
consul	1	
elasticsearch logs	2	
logstash	3	
kibana	4	
elasticsearch vitam	5	
mongoc	6	
mongod	7	
mongos	8	
storage-offer-xxx	9	
storage-engine	10	
metadata	11	
logbook	12	
functional-administration	13	
workspace	14	
processing	15	
worker	16	
access-internal	17	
access-external	18	
ingest-internal	19	
ingest-external	20	
ihm-demo	21	

Un playbook ansible de démarrage de VITAM est fourni, sous `deployment/ansible-vitam-exploitation` (fichier de *playbook* `start_vitam.yml`), pour réaliser de façon automatisée les actions nécessaires.

### 4.5.3 Procédure de statut

Un playbook ansible de démarrage de VITAM est fourni, sous `deployment/ansible-vitam-exploitation` (fichier de *playbook* `status_vitam.yml`), pour réaliser de façon automatisée les tests “autotest” intégrés dans la solution logicielle VITAM.

### 4.5.4 Mise à jour des certificats

Les certificats générés par les scripts de création d’une PKI interne à VITAM ont une durée de 10 ans pour les “CA”, 3 ans pour les intermédiaires.

La mise à jour des certificats associés aux composants externes est décrite dans la section *Mise à jour des certificats* (page 8).

## 4.6 Batches et traitements

### 4.6.1 Curator

Il existe des jobs Curator de :

- fermeture d’index
- suppression d’index fermés

Ces jobs sont lancés via `crontab` toutes les nuits.

## 4.6.2 Sécurisation des journaux d'opérations

Le script à appeler pour sécuriser les journaux d'opérations par tenant se trouve dans les machines `logbook` au chemin suivant : `/vitam/script/logbook/launchTraceability.sh`. Il est important de ne lancer ce script que sur une seule instance de `logbook` à la fois.

Exemple de cron pour appeler le script (toutes les heures) :

```
1 * * * * /vitam/script/logbook/launchTraceability.sh
```

## 4.6.3 Sécurisation des cycles de vie

Le script à appeler pour sécuriser les cycles de vie par tenant se trouve dans les machines `logbook` au chemin suivant : `/vitam/script/logbook/launchTraceabilityLFC.sh`. Il est important de ne lancer ce script que sur une seule instance de `logbook` à la fois.

Exemple de cron pour appeler le script (toutes les heures) :

```
1 * * * * /vitam/script/logbook/launchTraceabilityLFC.sh
```



---

## Suivi de l'état du système

---

### 5.1 Veille et patchs sécurité

Les éléments d'infrastructure suivants sont particulièrement sensibles pour la sécurité de la solution logicielle *VITAM* et nécessitent d'être intégrés à la veille sécurité du système :

- Runtime Java (OpenJDK 8)

### 5.2 Métriques

La solution logicielle *VITAM* intègre une solution de monitoring des applications à l'aide de métriques. L'exploitant peut, s'il le souhaite, changer la configuration des remontées de métriques, ou bien utiliser celle par défaut proposée dans *VITAM*.

#### 5.2.1 Configuration

##### 5.2.1.1 Registres

Par défaut, 3 registres de métriques sont créés pour toutes les applications *VITAM* :

- les métriques de Jersey
- les métriques de la JVM (Java Virtual Machine)
- les métriques "métier"

**JERSEY** : Les métriques Jersey correspondent à 3 métriques, des *Timers*, des *Meters*, et des *ExceptionMeters* qui vont être enregistrées pour chaque URI des API Rest de *VITAM*.

- Les *Meters* font office de compteurs. Ils sont incrémentés de 1 chaque fois qu'une URI est requêtée.
- Les *Timers* font office de chronomètres. Ils chronomètrent le temps de réponse d'une URI chaque fois que celle-ci est requêtée.
- Les *ExceptionMeters* font office de compteurs. Ils sont incrémentés de 1 chaque fois qu'une URI soulève une Exception dans le code.

**JVM** : Les métriques JVM correspondent à des *Gauges* qui enregistrent des valeurs de ressources système utilisées par la Java Virtual Machine pour chaque application *VITAM*.

**BUSINESS** : Les métriques métiers correspondent à des métriques de n'importe quel type qui peuvent remonter toute donnée considérée utile dans une application *VITAM*.

### 5.2.1.2 Reporters

Par défaut, 2 reporters de métriques sont disponibles pour les applications VITAM. Les reporters de métriques sont en charge de collecter les valeurs des métriques à des intervalles réguliers.

**LogBack** : le reporter LogBack affiche les valeurs des métriques dans LogBack.

**ELASTICSEARCH** : le reporter Elasticsearch sauvegarde les valeurs des métriques dans une base de données Elasticsearch qui peut être configurée dans le fichier de configuration.

### 5.2.1.3 Fichier de configuration

Le fichier de configuration des métriques est situé dans `/vitam/conf/<service_id>/vitam.metrics.conf`. Ce fichier contient la documentation nécessaire pour configurer correctement les métriques. Une description des clés YAML y est disponible.

## 5.2.2 Métier

Aucun métrique n'a encore été défini à ce stade du projet.

## 5.2.3 Métriques techniques

### 5.2.3.1 Métriques système critiques

Aucun métrique n'a encore défini à ce stade du projet.

### 5.2.3.2 Indicateurs de SLA

Aucun indicateur n'a encore défini à ce stade du projet.

### 5.2.3.3 Indicateurs de performance

Aucun indicateur n'a encore défini à ce stade du projet.

## 5.2.4 Visualisation

Si un reporter de type **ElasticSearch** est configuré, alors les métriques peuvent être visualisées via l'application web Kibana<sup>8</sup>.

L'application Kibana comporte 4 sections qui seront développées :

- **Discover**
- **Visualize**
- **Dashboards**
- **Settings**

---

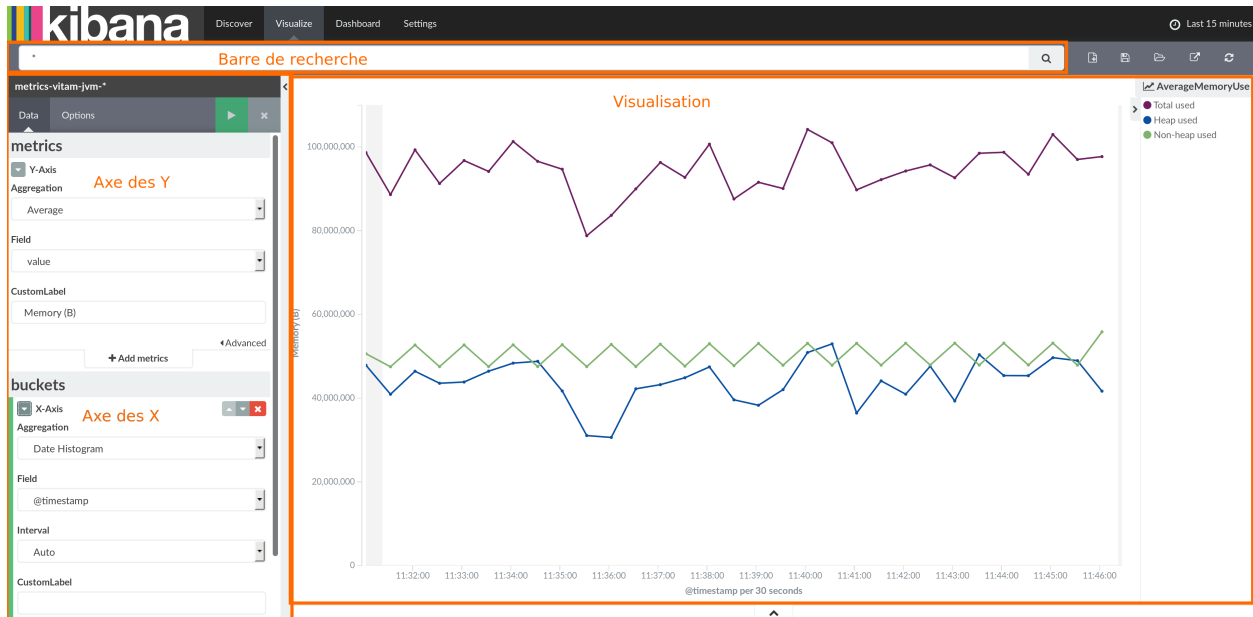
8. <https://www.elastic.co/fr/products/kibana>





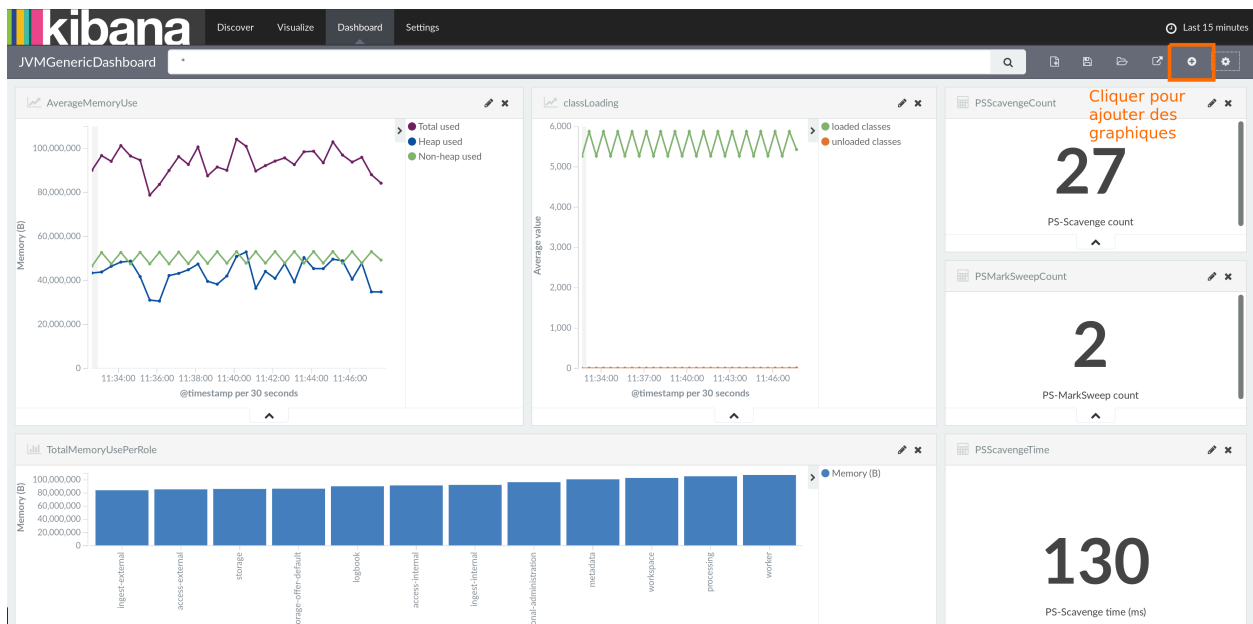
La barre latérale gauche du panneau de visualisation permet de configurer la donnée à représenter. Pour l'axe des Y, il est impératif d'utiliser une agrégation (moyenne, minimum/maximum, écart type...) sur une valeur pour la représenter. En fonction du graphique sélectionné, il est possible de configurer l'axe des X, toujours au moyen d'aggrégations (dates, date range, terme...).

En haut se situe la même barre de recherche que sur la partie Discover, qui permet d'affiner son graphique en effectuant des tris sur sa donnée.



### 5.2.4.3 Dashboards

La section **Dashboard** permet de regrouper plusieurs graphiques pour constituer un dashboard. Pour ce faire il suffit d'importer des graphiques avec le bouton “+” en haut à droite.



## 5.3 API de de supervision

La solution logicielle :term`VITAM` expose en interne de la plate-forme les API REST suivantes sur ses composants :

- `/admin/v1/status` : statut simple, renvoyant un statut de fonctionnement incluant des informations techniques sur l'état actuel du composant. Un exemple d'utilisation typique est l'intégration à un outil de supervision ou à un élément actif tiers (ex : load-balancer, ...) . L'appel doit être peu coûteux.
- `/admin/v1/version` : informations de version, build, commit git ayant servi à builder les différents jar.
- `/admin/v1/autotest` : autotest du composant, lançant un test de présence des différentes ressources requises par le composant et renvoyant un statut d'état de ces ressources.

### 5.3.1 Détail

#### 5.3.1.1 `/admin/v1/status`

L'API de status renvoie un fichier JSON contenant les informations suivantes :

```
{
  "serverIdentity": {
    "Name": "vitam-iaas-app-01",
    "Role": "logbook",
    "PlatformId": 425367
  },
  "status": true,
  "detail": { },
  "componentsVersions": {
    "e2eb99d93a74409b3ebc5224e596953e9b8a178f": 18
  }
}
```

Signification des champs :

- **serverIdentity**
  - Name : hostname du serveur hébergeant le composant (type : texte)
  - Role : Nom du composant (type : texte)
  - PlatformId : ID de l'environnement (type : entier)
- status : Statut du composant (OK/KO) (type : booléen)
- detail : vide dans cette version, sera défini ultérieurement
- **componentsVersions**
  - hash de commit git : nombre de jars avec buildés depuis ce hash

#### 5.3.1.2 `/admin/v1/version`

L'API de version renvoie les informations suivantes :

```
[
  {
    "Scm-tags": "",
    "Scm-commit-id": "e2eb99d93a74409b3ebc5224e596953e9b8a178f",
    "Scm-commit-id-abbrev": "e2eb99d",
    "Maven-version": "0.13.0-SNAPSHOT",
    "Scm-dirty": "false",
```

```

        "Scm-commit-time": "2017-01-11T16:38:14+01",
        "Maven-build-timestamp": "2017-01-11T16:06:09Z",
        "Scm-branch": "origin/master_iteration_13",
        "Build-Jdk": "1.8.0_111",
        "Maven-artefactId": "logbook-rest",
        "Maven-groupId": "fr.gouv.vitam"
    },
    {
        "Scm-tags": "",
        "Scm-commit-id": "e2eb99d93a74409b3ebc5224e596953e9b8a178f",
        "Scm-commit-id-abbrev": "e2eb99d",
        "Maven-version": "0.13.0-SNAPSHOT",
        "Scm-dirty": "false",
        "Scm-commit-time": "2017-01-11T16:38:14+01",
        "Maven-build-timestamp": "2017-01-11T16:06:09Z",
        "Scm-branch": "origin/master_iteration_13",
        "Build-Jdk": "1.8.0_111",
        "Maven-artefactId": "logbook-administration",
        "Maven-groupId": "fr.gouv.vitam"
    },
    ...
    ...
    ...
]

```

Signification des champs :

- Scm-tags : en cours de définition
- Scm-commit-id : hash de commit git à partir duquel le composant à été buildé
- Scm-commit-id-abbrev : hash de commit abrégé
- Maven-version : Version indiquée à maven dans le fichier pom.xml
- Scm-dirty : Etat du repo git au moment du build (si présence de fichiers unstaged => dirty)
- Scm-commit-time : Date du commit git
- Maven-build-timestamp : Date du build par maven
- Scm-branch : Nom de la branche git à partir de laquelle le composant a été buildé
- Build-Jdk : Version de la jdk ayant servi à builder le composant
- Maven-artefactId : Nom du composant
- Maven-groupId : namespace du composant

### 5.3.1.3 /admin/v1/autotest

L'API d'autotest renvoie les informations suivantes :

```

{
    "httpCode": 200,
    "code": "000000",
    "context": "logbook",
    "state": "OK",
    "message": "All services are available",
    "description": "All services are available",
    "errors": [
        {
            "httpCode": 200,
            "code": "1",

```

```

        "context": "LogbookMongoDbAccessImpl",
        "state": "OK",
        "message": "Sub service is available",
        "description": "LogbookMongoDbAccessImpl service is available"
    },
    {
        "httpCode": 200,
        "code": "2",
        "context": "logbook",
        "state": "OK",
        "message": "Internal service is available",
        "description": "vitam-iaas-app-01 service is available"
    }
]
}

```

Signification des champs :

- httpCode : code de retour http
- code : en cours de définition ; futur code retour interne VITAM
- context : Nom du composant
- state : Etat du composant (OK/KO)
- message : Message de statut
- description : Message de description
- **errors**
  - httpCode : code de retour http
  - code : code de retour
  - context : nom du composant
  - state : Etat du composant
  - message : Message sur l'état du composant
  - description : Description sur l'état du composant

## 5.4 Logs

La solution logicielle *VITAM* propose une solution ouverte, au choix de l'exploitant. Ce dernier peut, à l'installation, comme à la mise à jour de la solution logicielle :term`VITAM`, choisir d'utiliser sa propre solution de "regroupement" des logs ou la solution embarquée dans la solution logicielle :term`VITAM`.

Dans le cas de la solution embarquée, celle-ci se décompose en :

- rsyslog déployé sur les machines "applicatives" *VITAM* et les envois applicatifs syslog vers un serveur de centralisation de logs (via facility local0)
- un serveur de centralisation de logs, comprenant :
  - un mono-noeud (au minimum, ou multi-noeuds) Elasticsearch
  - un moteur logstash, parsant les messages VITAM
  - un afficheur de rendu/aggrégation de données Kibana

**Voir aussi :**

Les principes & implémentation du système de gestion de logs inclus dans VITAM sont décrits plus en détail dans le DAT.

### 5.4.1 Changement des règles de log

- Pour les logs fichiers :
  - Définition : fichier `/vitam/conf/<service_id>/logback.xml`
  - Format des logs (encoder) : ne doit pas être changé ;
  - La sévérité peut être changée ;
  - Roulement : le roulement des fichiers défini par défaut dépend du temps, avec une taille globale maximale ; il est défini par la politique `TimeBasedRollingPolicy` de l'appendeur `RollingFileAppender`<sup>10</sup>, avec les paramètres suivants :
    - Nombre total de fichiers conservés : 30 (paramètre `maxHistory`) ;
    - Taille totale des fichiers de logs : 5 Go (paramètre `totalSizeCap`) ;
    - Pattern des fichiers : dans le répertoire de logs de l'application : `<service_id>.%d.log` (%d étant remplacé par `yyyy-MM-dd`) (paramètre `fileNamePattern`).
- Pour les logs syslog :
  - Format des logs (suffixPattern) : ne doit pas être changé ;
  - La sévérité peut être changée ;
  - Les stacktraces sont exclues de l'envoi à la centralisation des logs (paramètre `throwableExcluded` placé à `false`) ; ce paramètre ne doit pas être changé.
- Pour les logs du garbage collector :
  - Niveau de détail : activation des détails et des timestamps (paramètres JVM `-XX:+PrintGCDetails -XX:+PrintGCApplicationStoppedTime`)
  - Roulement : le roulement des fichiers dépend de la taille des fichiers, avec un nombre de fichiers maximal ; il est défini comme suit :
    - Activation du roulement : (paramètre JVM `-XX:+UseGCLogFileRotation`)
    - Nombre total de fichiers conservés : 10 (paramètre JVM `-XX:NumberOfGCLogFiles=10`)
    - Taille unitaire maximale d'un' fichiers de logs : 10 Mo (paramètre JVM `-XX:GCLogFileSize=10M`)
    - Pattern des fichiers : dans le répertoire de logs de l'application (paramètre `-Xloggc:$LOG_FOLDER/gc.log`) pour le fichier courant ; après roulement, les fichiers sont nommés `gc.log.<n>``` (avec ```<n>` le numéro du fichier, sur base 0).
- Pour les logs accès :
  - Définition : fichier `/vitam/conf/<service_id>/logback-access.xml`
  - Format des logs (encoder) : ne doit pas être changé ;
  - Roulement : le roulement des fichiers défini par défaut dépend du temps, avec une taille globale maximale ; il est défini par la politique `TimeBasedRollingPolicy` de l'appendeur `RollingFileAppender`<sup>11</sup>, avec les paramètres suivants :
    - Nombre total de fichiers conservés : 7 (paramètre `maxHistory`) ;
    - Taille totale des fichiers de logs : 14 Go (paramètre `totalSizeCap`) ;
    - Pattern des fichiers : dans le répertoire de logs de l'application : `accesslog-<service_id>.%d.log` (%d étant remplacé par `yyyy-MM-dd`) (paramètre `fileNamePattern`).

---

10. <http://logback.qos.ch/manual/appenders.html#RollingFileAppender>

11. <http://logback.qos.ch/manual/appenders.html#RollingFileAppender>

<p><b>Prudence :</b> La configuration de la durée de rétention des logs accès et/ou leur externalisation devra être ajustée pour respecter les contraintes légales en vigueur pour le système déployé.</p>
--

## 5.5 Audit

## 5.6 Gestion de la capacité

## 5.7 Suivi de l'état de sécurité

Une étude est actuellement en cours pour réaliser ce type de suivi.

## 5.8 Alerting

### 5.8.1 Système

### 5.8.2 Applicatif





## Exploitation des composants de la solution logicielle VITAM

### 6.1 Généralités

Les composants de la solution logicielle *VITAM* sont déployés par un playbook ansible qui :

1. déploie, selon l'inventaire employé, les *packages* nécessaires
2. applique la configuration de chaque composant selon son contexte défini dans l'inventaire

Les composants VITAM sont décrits ci-après.

**Avertissement :** En cas de modification de la configuration, redémarrer le service associé.

### 6.2 Fichiers communs

Les composants de la solution logicielle *VITAM* utilisent un socle de fichiers communs.

#### 6.2.1 Fichier `/vitam/conf/<composant>/sysconfig/java_opts`

Ce fichier définit les JVMARGS.

```

1  # *****
2  # Copyright French Prime minister Office/SGMAP/DINSIC/Vitam Program (2015-2019)
3  #
4  # contact.vitam@culture.gouv.fr
5  #
6  # This software is a computer program whose purpose is to implement a digital_
   ↳ archiving back-office system managing
7  # high volumetry securely and efficiently.
8  #
9  # This software is governed by the CeCILL 2.1 license under French law and abiding by_
   ↳ the rules of distribution of free
10 # software. You can use, modify and/ or redistribute the software under the terms of_
   ↳ the CeCILL 2.1 license as
11 # circulated by CEA, CNRS and INRIA at the following URL "http://www.cecill.info".
12 #
13 # As a counterpart to the access to the source code and rights to copy, modify and_
   ↳ redistribute granted by the license,
14 # users are provided only with a limited warranty and the software's author, the_
   ↳ holder of the economic rights, and the

```

```

15 # successive licensors have only limited liability.
16 #
17 # In this respect, the user's attention is drawn to the risks associated with loading,
18 ↪ using, modifying and/or
19 # developing or reproducing the software by the user in light of its specific status,
20 ↪ of free software, that may mean
21 # that it is complicated to manipulate, and that also therefore means that it is
22 ↪ reserved for developers and
23 # experienced professionals having in-depth computer knowledge. Users are therefore
24 ↪ encouraged to load and test the
25 # software's suitability as regards their requirements in conditions enabling the
26 ↪ security of their systems and/or data
27 # to be ensured and, more generally, to use and operate it in the same conditions as
28 ↪ regards security.
29 #
30 # The fact that you are presently reading this means that you have had knowledge of
31 ↪ the CeCILL 2.1 license and that you
32 # accept its terms.
33 #*****
34 JAVA_OPTS="{ { vitam_struct.jvm_opts.gc | default(gc_opts) } } { { vitam_struct.jvm_opts.
35 ↪ memory | default(memory_opts) } } { { vitam_struct.jvm_opts.java | default(java_opts)
36 ↪ } } -Dorg.owasp.esapi.resources={ { vitam_folder_conf } } -Dlogback.configurationFile={
37 ↪ {vitam_folder_conf}}/logback.xml -Dvitam.config.folder={ { vitam_folder_conf } } -
38 ↪ Dvitam.data.folder={ { vitam_folder_data } } -Dvitam.tmp.folder={ { vitam_folder_tmp } }
39 ↪ -Dvitam.log.folder={ { vitam_folder_log } } -Djava.security.properties={ {vitam_folder_
40 ↪ conf}}/java.security"
41 JAVA_ARGS="{ { vitam_folder_conf } }/{ {vitam_struct.vitam_component}}.conf"

```

## 6.2.2 Fichier /vitam/conf/<composant>/logback.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <configuration>
3     <appender name="FILE" class="ch.qos.logback.core.rolling.RollingFileAppender">
4         <rollingPolicy class="ch.qos.logback.core.rolling.
5 ↪ TimeBasedRollingPolicy">
6             <fileNamePattern>{ { vitam_folder_log } }/accesslog-{ { vitam_
7 ↪ struct.vitam_component } }.%d{yyyy-MM-dd}.log</fileNamePattern>
8             <maxHistory>{ { days_to_delete_access_local } }</maxHistory>
9             <totalSizeCap>14GB</totalSizeCap>
10        </rollingPolicy>
11        <encoder>
12            <pattern>%h %l %u %t "%r" %s %b "%i{Referer}" "%i{User-agent}
13 ↪ " %D %i{X-Request-ID} %i{X-Tenant-Id} %i{X-Application-Id}</pattern>
14        </encoder>
15    </appender>
16    <appender-ref ref="FILE" />
17 </configuration>

```

## 6.2.3 Fichier /vitam/conf/<composant>/logback-access.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <configuration>
3
4     <!-- Send debug messages to System.out -->

```

```

5      <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">
6          <!-- By default, encoders are assigned the type ch.qos.logback.
↳ classic.encoder.PatternLayoutEncoder -->
7              <encoder>
8                  <pattern>%d{ISO8601} [%thread] [%X{X-REQUEST-ID}] %-5level
↳ %logger - %replace(%caller{1..2}){'Caller\+1          at |\n',''} : %msg
↳ %rootException%n</pattern>
9                  </encoder>
10             </appender>
11         <!-- <appender name="FILE" class="ch.qos.logback.core.FileAppender" -->
12         <appender name="FILE" class="ch.qos.logback.core.rolling.RollingFileAppender">
13             <rollingPolicy class="ch.qos.logback.core.rolling.
↳ SizeAndTimeBasedRollingPolicy">
14                 <fileNamePattern>{{vitam_folder_log}}/{{ vitam_struct.vitam_
↳ component }}.{{d{yyyy-MM-dd}.%i.log</fileNamePattern>
15                 <maxFileSize>10MB</maxFileSize>
16                 <maxHistory>{{ days_to_delete_logback_logfiles }}</maxHistory>
17                 <totalSizeCap>5GB</totalSizeCap>
18             </rollingPolicy>
19
20             <!-- TODO : replace with rolling file appender -->
21             <!-- <file>{{vitam_folder_log}}/{{ vitam_struct.vitam_component }}.log
↳ </file>
22
23             <append>true</append> -->
24             <encoder>
25                 <pattern>%d{ISO8601} [%thread] [%X{X-REQUEST-ID}] %-5level
↳ %logger - %replace(%caller{1..2}){'Caller\+1          at |\n',''} : %msg %rootException%n
↳ </pattern>
26                 </encoder>
27             </appender>
28
29         <appender name="SYSLOG" class="ch.qos.logback.classic.net.SyslogAppender">
30             <syslogHost>localhost</syslogHost>
31             <facility>{{ vitam_defaults.syslog_facility }}</facility>
32             <suffixPattern>vitam-{{vitam_struct.vitam_component}}: %d{ISO8601} [[
↳ %thread] [%X{X-REQUEST-ID}] %-5level %logger - %replace(%caller{1..2}){'Caller\+1
↳ at |\n',''} : %msg %rootException%n</suffixPattern>
33
34             <!-- By default, the level of the root level is set to TRACE -->
35             <root level="{{ vitam_defaults.services.log_level }}">
36                 <!-- <appender-ref ref="STDOUT" /> -->
37                 <appender-ref ref="FILE" />
38                 <appender-ref ref="SYSLOG" />
39             </root>
40
41             <logger name="org.eclipse.jetty" level="WARN"/>
42             <logger name="fr.gouv.vitam.storage.engine.server.logbook.StorageLogbookMock"
↳ level="INFO"/>
43             <logger name="fr.gouv.vitam.common" level="WARN" />
44             <logger name="fr.gouv.vitam.worker.core.impl.WorkerImpl" level="INFO" />
45 </configuration>

```

## 6.2.4 Fichier /vitam/conf/<composant>/jetty-config.xml

```

1  <?xml version="1.0"?>
2  <!DOCTYPE Configure PUBLIC "-//Jetty//Configure//EN" "http://www.eclipse.org/jetty/
   ↪configure_9_0.dtd">
3
4  <!-- ===== -->
5  <!-- Documentation of this file format can be found at: -->
6  <!-- http://wiki.eclipse.org/Jetty/Reference/jetty.xml_syntax -->
7  <!-- -->
8  <!-- Additional configuration files are available in $JETTY_HOME/etc -->
9  <!-- and can be mixed in. See start.ini file for the default -->
10 <!-- configuration files. -->
11 <!-- -->
12 <!-- For a description of the configuration mechanism, see the -->
13 <!-- output of: -->
14 <!-- java -jar start.jar -? -->
15 <!-- ===== -->
16
17 <!-- ===== -->
18 <!-- Configure a Jetty Server instance with an ID "Server" -->
19 <!-- Other configuration files may also configure the "Server" -->
20 <!-- ID, in which case they are adding configuration to the same -->
21 <!-- instance. If other configuration have a different ID, they -->
22 <!-- will create and configure another instance of Jetty. -->
23 <!-- Consult the javadoc of o.e.j.server.Server for all -->
24 <!-- configuration that may be set here. -->
25 <!-- ===== -->
26 <Configure id="Server" class="org.eclipse.jetty.server.Server">
27
28
29     <!-- ===== -->
30     <!-- Add shared Scheduler instance -->
31     <!-- ===== -->
32     <Call name="addBean">
33         <Arg>
34             <New class="org.eclipse.jetty.util.thread.ScheduledExecutorScheduler"/>
35         </Arg>
36     </Call>
37
38     <!-- ===== -->
39     <!-- Http Configuration. -->
40     <!-- This is a common configuration instance used by all -->
41     <!-- connectors that can carry HTTP semantics (HTTP, HTTPS, SPDY) -->
42     <!-- It configures the non wire protocol aspects of the HTTP -->
43     <!-- semantic. -->
44     <!-- -->
45     <!-- This configuration is only defined here and is used by -->
46     <!-- reference from the jetty-http.xml, jetty-https.xml and -->
47     <!-- jetty-spydy.xml configuration files which instantiate the -->
48     <!-- connectors. -->
49     <!-- -->
50     <!-- Consult the javadoc of o.e.j.server.HttpConfiguration -->
51     <!-- for all configuration that may be set here. -->
52     <!-- ===== -->
53     <New id="httpConfig" class="org.eclipse.jetty.server.HttpConfiguration">
54         <Set name="secureScheme">http</Set>
55         <Set name="securePort">8443</Set>

```

```

56     <Set name="outputBufferSize">32768</Set>
57     <Set name="requestHeaderSize">8192</Set>
58     <Set name="responseHeaderSize">8192</Set>
59     <Set name="sendServerVersion">false</Set>
60     <Set name="sendDateHeader">false</Set>
61     <Set name="headerCacheSize">512</Set>
62
63     <!-- Uncomment to enable handling of X-Forwarded- style headers
64     <Call name="addCustomizer">
65         <Arg><New class="org.eclipse.jetty.server.ForwardedRequestCustomizer"/></
↪Arg>
66         </Call>
67         -->
68     </New>
69
70     <!-- ===== Original Connector ===== -
↪->
71     <!-- <Call name="addConnector">
72         <!-- <Arg>
73         <!-- <New class="org.eclipse.jetty.server.ServerConnector">
74         <!-- <Arg name="server"><Ref refid="Server" /></Arg>
75         <!-- <Arg name="factories">
76         <!-- <Array type="org.eclipse.jetty.server.ConnectionFactory">
77         <!-- <Item>
78         <!-- <New class="org.eclipse.jetty.server.
↪HttpConnectionFactory">
79         <!-- <Arg name="config"><Ref refid="httpConfig" /></
↪Arg>
80         <!-- </New>
81         <!-- </Item>
82         <!-- </Array>
83         <!-- </Arg>
84         <!-- <Set name="port">{{ vitam_struct.port_http }}</Set>
85         <!-- <Set name="idleTimeout">
86         <!-- <Property name="http.timeout" default="{{ vitam_defaults.
↪services.port_http_timeout }}" />
87         <!-- </Set>
88         <!-- </New>
89         <!-- </Arg>
90         <!-- </Call>
91         -->

```

```

92
93
94     <!-- ===== -->
95     <!-- Set the default handler structure for the Server -->
96     <!-- A handler collection is used to pass received requests to -->
97     <!-- both the ContextHandlerCollection, which selects the next -->
98     <!-- handler by context path and virtual host, and the -->
99     <!-- DefaultHandler, which handles any requests not handled by -->
100    <!-- the context handlers. -->
101    <!-- Other handlers may be added to the "Handlers" collection, -->
102    <!-- for example the jetty-requestlog.xml file adds the -->
103    <!-- RequestLogHandler after the default handler -->
104    <!-- ===== -->
105    <Set name="handler">
106        <New id="Handlers" class="org.eclipse.jetty.server.handler.HandlerCollection">
107            <Set name="handlers">
108                <Array type="org.eclipse.jetty.server.Handler">
109                    <Item>
110                        <New id="Contexts" class="org.eclipse.jetty.server.handler.
111↪ContextHandlerCollection"/>
112                    </Item>
113                    <Item>
114                        <New id="DefaultHandler" class="org.eclipse.jetty.server.
115↪handler.DefaultHandler"/>
116                    </Item>
117                </Array>
118            </Set>
119        </New>
120    </Set>
121
122    <Set name="RequestLog">
123        <New id="RequestLogImpl" class="ch.qos.logback.access.jetty.RequestLogImpl
124↪">
125            <Set name="fileName">{{vitam_folder_conf}}/logback-access.xml</Set>
126        </New>
127    </Set>
128    <Ref id="RequestLogImpl">
129        <Call name="start"/>
130    </Ref>
131
132    <!-- ===== -->
133    <!-- extra server options -->
134    <!-- ===== -->
135    <Set name="stopAtShutdown">true</Set>
136    <Set name="stopTimeout">5000</Set>
137    <Set name="dumpAfterStart">false</Set>
138    <Set name="dumpBeforeStop">false</Set>
139
140    {% if vitam_struct.https_enabled==true %}
141    <New id="httpsConfig" class="org.eclipse.jetty.server.HttpConfiguration">
142        <Call name="addCustomizer">
143            <Arg>
144                <New class="org.eclipse.jetty.server.SecureRequestCustomizer" />
145            </Arg>
146        </Call>
147    </New>
148    <New id="sslContextFactory" class="org.eclipse.jetty.util.ssl.
149↪SslContextFactory">

```

```

146     <Set name="KeyStorePath">{{ vitam_folder_conf }}/keystore_{{ vitam_struct.
↪vitam_component }}.jks</Set>
147     <Set name="KeyStorePassword">{{password_keystore}}</Set>
148     <Set name="KeyManagerPassword">{{password_manager_keystore}}</Set>
149     <Set name="TrustStorePath">{{ vitam_folder_conf }}/truststore_{{ vitam_struct.
↪vitam_component }}.jks</Set>
150     <Set name="TrustStorePassword">{{password_truststore}}</Set>
151     <Set name="TrustStoreType">JKS</Set>
152     <Set name="NeedClientAuth">>false</Set>
153     <Set name="WantClientAuth">>true</Set>
154     <Set name="IncludeCipherSuites">
155         <Array type="String">
156             <Item>TLS_ECDHE.*</Item>
157             <Item>TLS_DHE_RSA.*</Item>
158         </Array>
159     </Set>
160     <Set name="IncludeProtocols">
161         <Array type="String">
162             <Item>TLSv1</Item>
163             <Item>TLSv1.1</Item>
164             <Item>TLSv1.2</Item>
165         </Array>
166     </Set>
167     <Set name="ExcludeCipherSuites">
168         <Array type="String">
169             <Item>.*NULL.*</Item>
170             <Item>.*RC4.*</Item>
171             <Item>.*MD5.*</Item>
172             <Item>.*DES.*</Item>
173             <Item>.*DSS.</Item>
174         </Array>
175     </Set>
176     <Set name="UseCipherSuitesOrder">>true</Set>
177     <Set name="RenegotiationAllowed">>true</Set>
178 </New>
179 <New id="sslConnectionFactory" class="org.eclipse.jetty.server.
↪SslConnectionFactory">
180     <Arg name="sslContextFactory">
181         <Ref refid="sslContextFactory" />
182     </Arg>
183     <Arg name="next">http/1.1</Arg>
184 </New>
185 <New id="businessConnector" class="org.eclipse.jetty.server.ServerConnector">
186     <Arg name="server">
187         <Ref refid="Server" />
188     </Arg>
189     <Arg name="factories">
190         <Array type="org.eclipse.jetty.server.ConnectionFactory">
191             <Item>
192                 <Ref refid="sslConnectionFactory" />
193             </Item>
194             <Item>
195                 <New class="org.eclipse.jetty.server.HttpConnectionFactory">
196                     <Arg name="config">
197                         <Ref refid="httpsConfig" />
198                     </Arg>
199                 </New>
200             </Item>

```

```

201     </Array>
202   </Arg>
203   <Set name="host">{{ ip_service }}</Set>
204   <Set name="port">
205     <SystemProperty name="jetty.port" default="{{ vitam_struct.port_https }}" /
↪>
206   </Set>
207   <Set name="name">business</Set>
208 </New>
209
210 {% else %}
211
212 <!-- ===== -->
213 <!-- Connector for API business -->
214 <!-- Attach all ContextHanlder except Admin -->
215 <!-- ===== -->
216
217 <New id="businessConnector" class="org.eclipse.jetty.server.ServerConnector">
218   <Arg name="server"><Ref refid="Server" /></Arg>
219   <Arg name="factories">
220     <Array type="org.eclipse.jetty.server.ConnectionFactory">
221       <Item>
222         <New class="org.eclipse.jetty.server.HttpConnectionFactory">
223           <Arg name="config"><Ref refid="httpConfig" /></Arg>
224         </New>
225       </Item>
226     </Array>
227   </Arg>
228   <Set name="host">{{ ip_service }}</Set>
229   <Set name="port">{{ vitam_struct.port_http }}</Set>
230   <Set name="name">business</Set>
231   <Set name="idleTimeout">
232     <Property name="http.timeout" default="{{ vitam_defaults.services.port_
↪http_timeout }}" />
233   </Set>
234 </New>
235
236 {% endif %}
237
238 <!-- ===== -->
239 <!-- Connector for API Admin -->
240 <!-- Attach all ContextHanlder -->
241 <!-- ===== -->
242
243 <New id="adminConnector" class="org.eclipse.jetty.server.ServerConnector">
244   <Arg name="server"><Ref refid="Server" /></Arg>
245   <Arg name="factories">
246     <Array type="org.eclipse.jetty.server.ConnectionFactory">
247       <Item>
248         <New class="org.eclipse.jetty.server.HttpConnectionFactory">
249           <Arg name="config"><Ref refid="httpConfig" /></Arg>
250         </New>
251       </Item>
252     </Array>
253   </Arg>
254   <Set name="host">{{ ip_admin }}</Set>
255   <Set name="port">{{ vitam_struct.port_admin }}</Set>
256   <Set name="name">admin</Set>

```



```

257     <Set name="idleTimeout">
258         <Property name="http.timeout" default="{ { vitam_defaults.services.port_
↪http_timeout } }"/>
259     </Set>
260 </New>
261
262
263
264
265     <Call name="setConnectors">
266         <Arg>
267             <Array type="org.eclipse.jetty.server.ServerConnector">
268                 <Item>
269                     <Ref refid="businessConnector" />
270                 </Item>
271                 <Item>
272                     <Ref refid="adminConnector" />
273                 </Item>
274             </Array>
275         </Arg>
276     </Call>
277
278 </Configure>

```

## 6.2.5 Fichier /vitam/conf/<composant>/logbook-client.conf

Ce fichier permet de configurer l'appel au composant logbook.

```

1 serverHost: { { vitam.logbook.host } }
2 serverPort: { { vitam.logbook.port_http } }

```

## 6.2.6 Fichier /vitam/conf/<composant>/server-identity.conf

```

1 identityName: { { ansible_nodename } }
2 identityRole: { { vitam_struct.vitam_composant } }
3 identitySiteId: { { vitam_site_id } }

```

## 6.2.7 Fichier /vitam/conf/<composant>/antisamy-esapi.xml

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2
3 <!--
4 W3C rules retrieved from:
5 http://www.w3.org/TR/html401/struct/global.html
6 -->
7
8 <!--
9 Slashdot allowed tags taken from "Reply" page:
10 <b> <i> <p> <br> <a> <ol> <ul> <li> <dl> <dt> <dd> <em> <strong> <tt> <blockquote>
↪<div> <code> <quote>
11 -->
12

```

```

13 <anti-samy-rules xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
14     xsi:noNamespaceSchemaLocation="antisamy.xsd">
15
16     <directives>
17         <directive name="omitXmlDeclaration" value="true"/>
18         <directive name="omitDoctypeDeclaration" value="true"/>
19         <directive name="maxInputSize" value="2000000"/>
20         <directive name="embedStyleSheets" value="false"/>
21     </directives>
22
23
24     <common-regexps>
25
26         <!--
27         From W3C:
28         This attribute assigns a class name or set of class names to an
29         element. Any number of elements may be assigned the same class
30         name or names. Multiple class names must be separated by white
31         space characters.
32         -->
33
34         <regexp name="htmlTitle" value="[a-zA-Z0-9\s-_',:\[\]!\./\\(\)]*" />
35         <!-- force non-empty with a '+' at the end instead of '*' -->
36         <regexp name="onsiteURL" value="([\w\\/\.\.?=&;\#~]+|(\#(\w)+)"/>
37         <!-- z0-9]+[~a-zA-Z0-9-_\.@#$%&;:,\?=/+!]*(\s)*"/>
38         <regexp name="offsiteURL" value="(\s)*((ht|f)tp(s?)://|mailto:)[A-Za-
39         <!--
40
41         Tag.name = a, b, div, body, etc.
42         Tag.action = filter: remove tags, but keep content, validate: keep content as
43         <!-- long as it passes rules, remove: remove tag and contents
44         Attribute.name = id, class, href, align, width, etc.
45         Attribute.onInvalid = what to do when the attribute is invalid, e.g., remove
46         <!-- the tag (removeTag), remove the attribute (removeAttribute), filter the tag
47         <!-- (filterTag)
48         Attribute.description = What rules in English you want to tell the users they
49         <!-- can have for this attribute. Include helpful things so they'll be able to tune
50         <!-- their HTML
51
52         -->
53
54         <!--
55         Some attributes are common to all (or most) HTML tags. There aren't many that
56         <!-- qualify for this. You have to make sure there's no
57         <!-- collisions between any of these attribute names with attribute names of other
58         <!-- tags that are for different purposes.
59         -->
60
61     <common-attributes>
62
63         <attribute name="lang" description="The 'lang' attribute tells the
64         <!-- browser what language the element's attribute values and content are written in">
65             <regexp-list>
66                 <regexp value="[a-zA-Z]{2,20}"/>

```

```

61         </regexp-list>
62     </attribute>
63
64     <attribute name="title" description="The 'title' attribute provides
65     ↳text that shows up in a 'tooltip' when a user hovers their mouse over the element">
66         <regexp-list>
67             <regexp name="htmlTitle"/>
68         </regexp-list>
69     </attribute>
70
71     <attribute name="href" onInvalid="filterTag">
72         <regexp-list>
73             <regexp name="onsiteURL"/>
74             <regexp name="offsiteURL"/>
75         </regexp-list>
76     </attribute>
77
78     <attribute name="align" description="The 'align' attribute of an HTML
79     ↳element is a direction word, like 'left', 'right' or 'center'">
80         <literal-list>
81             <literal value="center"/>
82             <literal value="left"/>
83             <literal value="right"/>
84             <literal value="justify"/>
85             <literal value="char"/>
86         </literal-list>
87     </attribute>
88
89     </common-attributes>
90
91     <!--
92     This requires normal updates as browsers continue to diverge from the W3C and
93     ↳each other. As long as the browser wars continue
94     this is going to continue. I'm not sure war is the right word for what's
95     ↳going on. Doesn't somebody have to win a war after
96     a while?
97     -->
98
99     <global-tag-attributes>
100         <attribute name="title"/>
101         <attribute name="lang"/>
102     </global-tag-attributes>
103
104     <tag-rules>
105
106         <!-- Tags related to JavaScript -->
107
108         <tag name="script" action="remove"/>
109         <tag name="noscript" action="remove"/>
110
111         <!-- Frame & related tags -->
112
113         <tag name="iframe" action="remove"/>
114         <tag name="frameset" action="remove"/>
115         <tag name="frame" action="remove"/>
116         <tag name="noframes" action="remove"/>

```

```

115
116
117     <!-- All reasonable formatting tags -->
118
119     <tag name="p" action="validate">
120         <attribute name="align"/>
121     </tag>
122
123     <tag name="div" action="validate"/>
124     <tag name="i" action="validate"/>
125     <tag name="b" action="validate"/>
126     <tag name="em" action="validate"/>
127     <tag name="blockquote" action="validate"/>
128     <tag name="tt" action="validate"/>
129
130     <tag name="br" action="truncate"/>
131
132     <!-- Custom Slashdot tags, though we're trimming the idea of having a
133     ↪ possible mismatching end tag with the endtag="" attribute -->
134
135     <tag name="quote" action="validate"/>
136     <tag name="ecode" action="validate"/>
137
138     <!-- Anchor and anchor related tags -->
139
140     <tag name="a" action="validate">
141
142         <attribute name="href" onInvalid="filterTag"/>
143         <attribute name="nohref">
144             <literal-list>
145                 <literal value="nohref"/>
146                 <literal value=""/>
147             </literal-list>
148         </attribute>
149         <attribute name="rel">
150             <literal-list>
151                 <literal value="nofollow"/>
152             </literal-list>
153         </attribute>
154     </tag>
155
156     <!-- List tags -->
157
158     <tag name="ul" action="validate"/>
159     <tag name="ol" action="validate"/>
160     <tag name="li" action="validate"/>
161
162 </tag-rules>
163
164
165
166     <!-- No CSS on Slashdot posts -->
167
168     <css-rules>
169     </css-rules>
170
171

```

```

172     <html-entities>
173         <entity name="amp" cdata="&#106;" />
174         <entity name="nbsp" cdata="&#160;" />
175
176         <entity name="iexcl" cdata="&#161;" /> <!--inverted exclamation
177         ↪mark, U+00A1 ISONum -->
178         <entity name="cent" cdata="&#162;" /> <!--cent sign, U+00A2 ISONum
179         ↪-->
180         <entity name="pound" cdata="&#163;" /> <!--pound sign, U+00A3
181         ↪ISONum -->
182         <entity name="curren" cdata="&#164;" /> <!--currency sign, U+00A4
183         ↪ISONum -->
184         <entity name="yen" cdata="&#165;" /> <!--yen sign = yuan sign,
185         ↪U+00A5 ISONum -->
186         <entity name="brvbar" cdata="&#166;" /> <!--broken bar = broken
187         ↪vertical bar, U+00A6 ISONum -->
188         <entity name="sect" cdata="&#167;" /> <!--section sign, U+00A7
189         ↪ISONum -->
190         <entity name="uml" cdata="&#168;" /> <!--diaeresis = spacing
191         ↪diaeresis, U+00A8 ISodia -->
192         <entity name="copy" cdata="&#169;" /> <!--copyright sign, U+00A9
193         ↪ISONum -->
194         <entity name="ordf" cdata="&#170;" /> <!--feminine ordinal
195         ↪indicator, U+00AA ISONum -->
196         <entity name="laquo" cdata="&#171;" /> <!--left-pointing double
197         ↪angle quotation mark = left pointing guillemet, U+00AB ISONum -->
198         <entity name="not" cdata="&#172;" /> <!--not sign, U+00AC ISONum --
199         ↪>
200         <entity name="shy" cdata="&#173;" /> <!--soft hyphen =
201         ↪discretionary hyphen, U+00AD ISONum -->
202         <entity name="reg" cdata="&#174;" /> <!--registered sign =
203         ↪registered trade mark sign, U+00AE ISONum -->
204         <entity name="macr" cdata="&#175;" /> <!--macron = spacing macron
205         ↪= overline = APL overbar, U+00AF ISodia -->
206         <entity name="deg" cdata="&#176;" /> <!--degree sign, U+00B0
207         ↪ISONum -->
208         <entity name="plusmn" cdata="&#177;" /> <!--plus-minus sign = plus-
209         ↪or-minus sign, U+00B1 ISONum -->
210         <entity name="sup2" cdata="&#178;" /> <!--superscript two =
211         ↪superscript digit two = squared, U+00B2 ISONum -->
212         <entity name="sup3" cdata="&#179;" /> <!--superscript three =
213         ↪superscript digit three= cubed, U+00B3 ISONum -->
214         <entity name="acute" cdata="&#180;" /> <!--acute accent = spacing
215         ↪acute, U+00B4 ISodia -->
216         <entity name="micro" cdata="&#181;" /> <!--micro sign, U+00B5
217         ↪ISONum -->
218         <entity name="para" cdata="&#182;" /> <!--pilcrow sign = paragraph
219         ↪sign, U+00B6 ISONum -->
220         <entity name="middot" cdata="&#183;" /> <!--middle dot = Georgian
221         ↪comma = Greek middle dot, U+00B7 ISONum -->
222         <entity name="cedil" cdata="&#184;" /> <!--cedilla = spacing
223         ↪cedilla, U+00B8 ISodia -->
224         <entity name="sup1" cdata="&#185;" /> <!--superscript one =
225         ↪superscript digit one, U+00B9 ISONum -->
226         <entity name="ordm" cdata="&#186;" /> <!--masculine ordinal
227         ↪indicator, U+00BA ISONum -->
228         <entity name="raquo" cdata="&#187;" /> <!--right-pointing double
229         ↪angle quotation mark = right pointing guillemet, U+00BB ISONum -->

```

```

203         <entity name="frac14" cdata="&#188;"/> <!--vulgar fraction one_
↳quarter = fraction one quarter, U+00BC ISOnum -->
204         <entity name="frac12" cdata="&#189;"/> <!--vulgar fraction one_
↳half = fraction one half, U+00BD ISOnum -->
205         <entity name="frac34" cdata="&#190;"/> <!--vulgar fraction three_
↳quarters = fraction three quarters, U+00BE ISOnum -->
206         <entity name="iquest" cdata="&#191;"/> <!--inverted question mark_
↳= turned question mark, U+00BF ISOnum -->
207         <entity name="Agrave" cdata="&#192;"/> <!--latin capital letter A_
↳with grave = latin capital letter A grave, U+00C0 ISolat1 -->
208         <entity name="Aacute" cdata="&#193;"/> <!--latin capital letter A_
↳with acute, U+00C1 ISolat1 -->
209         <entity name="Acirc" cdata="&#194;"/> <!--latin capital letter A_
↳with circumflex, U+00C2 ISolat1 -->
210         <entity name="Atilde" cdata="&#195;"/> <!--latin capital letter A_
↳with tilde, U+00C3 ISolat1 -->
211         <entity name="Auml" cdata="&#196;"/> <!--latin capital letter A_
↳with diaeresis, U+00C4 ISolat1 -->
212         <entity name="Aring" cdata="&#197;"/> <!--latin capital letter A_
↳with ring above = latin capital letter A ring, U+00C5 ISolat1 -->
213         <entity name="AElig" cdata="&#198;"/> <!--latin capital letter AE_
↳= latin capital ligature AE, U+00C6 ISolat1 -->
214         <entity name="Ccedil" cdata="&#199;"/> <!--latin capital letter C_
↳with cedilla, U+00C7 ISolat1 -->
215         <entity name="Egrave" cdata="&#200;"/> <!--latin capital letter E_
↳with grave, U+00C8 ISolat1 -->
216         <entity name="Eacute" cdata="&#201;"/> <!--latin capital letter E_
↳with acute, U+00C9 ISolat1 -->
217         <entity name="Ecirc" cdata="&#202;"/> <!--latin capital letter E_
↳with circumflex, U+00CA ISolat1 -->
218         <entity name="Euml" cdata="&#203;"/> <!--latin capital letter E_
↳with diaeresis, U+00CB ISolat1 -->
219         <entity name="Igrave" cdata="&#204;"/> <!--latin capital letter I_
↳with grave, U+00CC ISolat1 -->
220         <entity name="Iacute" cdata="&#205;"/> <!--latin capital letter I_
↳with acute, U+00CD ISolat1 -->
221         <entity name="Icirc" cdata="&#206;"/> <!--latin capital letter I_
↳with circumflex, U+00CE ISolat1 -->
222         <entity name="Iuml" cdata="&#207;"/> <!--latin capital letter I_
↳with diaeresis, U+00CF ISolat1 -->
223         <entity name="ETH" cdata="&#208;"/> <!--latin capital letter ETH, _
↳U+00D0 ISolat1 -->
224         <entity name="Ntilde" cdata="&#209;"/> <!--latin capital letter N_
↳with tilde, U+00D1 ISolat1 -->
225         <entity name="Ograve" cdata="&#210;"/> <!--latin capital letter O_
↳with grave, U+00D2 ISolat1 -->
226         <entity name="Oacute" cdata="&#211;"/> <!--latin capital letter O_
↳with acute, U+00D3 ISolat1 -->
227         <entity name="Ocirc" cdata="&#212;"/> <!--latin capital letter O_
↳with circumflex, U+00D4 ISolat1 -->
228         <entity name="Otilde" cdata="&#213;"/> <!--latin capital letter O_
↳with tilde, U+00D5 ISolat1 -->
229         <entity name="Ouml" cdata="&#214;"/> <!--latin capital letter O_
↳with diaeresis, U+00D6 ISolat1 -->
230         <entity name="times" cdata="&#215;"/> <!--multiplication sign, _
↳U+00D7 ISOnum -->
231         <entity name="Oslash" cdata="&#216;"/> <!--latin capital letter O_
↳with stroke = latin capital letter O slash, U+00D8 ISolat1 -->

```

```

232      <entity name="Ugrave" cdata="&#217;"/> <!--latin capital letter U_
↳with grave, U+00D9 ISolat1 -->
233      <entity name="Uacute" cdata="&#218;"/> <!--latin capital letter U_
↳with acute, U+00DA ISolat1 -->
234      <entity name="Ucirc" cdata="&#219;"/> <!--latin capital letter U_
↳with circumflex, U+00DB ISolat1 -->
235      <entity name="Uuml" cdata="&#220;"/> <!--latin capital letter U_
↳with diaeresis, U+00DC ISolat1 -->
236      <entity name="Yacute" cdata="&#221;"/> <!--latin capital letter Y_
↳with acute, U+00DD ISolat1 -->
237      <entity name="THORN" cdata="&#222;"/> <!--latin capital letter_
↳THORN, U+00DE ISolat1 -->
238      <entity name="szlig" cdata="&#223;"/> <!--latin small letter_
↳sharp s = ess-zed, U+00DF ISolat1 -->
239      <entity name="agrave" cdata="&#224;"/> <!--latin small letter a_
↳with grave = latin small letter a grave, U+00E0 ISolat1 -->
240      <entity name="aacute" cdata="&#225;"/> <!--latin small letter a_
↳with acute, U+00E1 ISolat1 -->
241      <entity name="acirc" cdata="&#226;"/> <!--latin small letter a_
↳with circumflex, U+00E2 ISolat1 -->
242      <entity name="atilde" cdata="&#227;"/> <!--latin small letter a_
↳with tilde, U+00E3 ISolat1 -->
243      <entity name="auml" cdata="&#228;"/> <!--latin small letter a_
↳with diaeresis, U+00E4 ISolat1 -->
244      <entity name="aring" cdata="&#229;"/> <!--latin small letter a_
↳with ring above = latin small letter a ring, U+00E5 ISolat1 -->
245      <entity name="aelig" cdata="&#230;"/> <!--latin small letter ae =
↳latin small ligature ae, U+00E6 ISolat1 -->
246      <entity name="ccedil" cdata="&#231;"/> <!--latin small letter c_
↳with cedilla, U+00E7 ISolat1 -->
247      <entity name="egrave" cdata="&#232;"/> <!--latin small letter e_
↳with grave, U+00E8 ISolat1 -->
248      <entity name="eacute" cdata="&#233;"/> <!--latin small letter e_
↳with acute, U+00E9 ISolat1 -->
249      <entity name="ecirc" cdata="&#234;"/> <!--latin small letter e_
↳with circumflex, U+00EA ISolat1 -->
250      <entity name="euml" cdata="&#235;"/> <!--latin small letter e_
↳with diaeresis, U+00EB ISolat1 -->
251      <entity name="igrave" cdata="&#236;"/> <!--latin small letter i_
↳with grave, U+00EC ISolat1 -->
252      <entity name="iacute" cdata="&#237;"/> <!--latin small letter i_
↳with acute, U+00ED ISolat1 -->
253      <entity name="icirc" cdata="&#238;"/> <!--latin small letter i_
↳with circumflex, U+00EE ISolat1 -->
254      <entity name="iuml" cdata="&#239;"/> <!--latin small letter i_
↳with diaeresis, U+00EF ISolat1 -->
255      <entity name="eth" cdata="&#240;"/> <!--latin small letter eth,
↳U+00F0 ISolat1 -->
256      <entity name="ntilde" cdata="&#241;"/> <!--latin small letter n_
↳with tilde, U+00F1 ISolat1 -->
257      <entity name="ograve" cdata="&#242;"/> <!--latin small letter o_
↳with grave, U+00F2 ISolat1 -->
258      <entity name="oacute" cdata="&#243;"/> <!--latin small letter o_
↳with acute, U+00F3 ISolat1 -->
259      <entity name="ocirc" cdata="&#244;"/> <!--latin small letter o_
↳with circumflex, U+00F4 ISolat1 -->
260      <entity name="otilde" cdata="&#245;"/> <!--latin small letter o_
↳with tilde, U+00F5 ISolat1 -->

```

```

261         <entity name="ouml" cdata="&#246;"/> <!--latin small letter o
↳with diaeresis, U+00F6 ISolat1 -->
262         <entity name="divide" cdata="&#247;"/> <!--division sign, U+00F7
↳ISOnum -->
263         <entity name="oslash" cdata="&#248;"/> <!--latin small letter o
↳with stroke, = latin small letter o slash, U+00F8 ISolat1 -->
264         <entity name="ugrave" cdata="&#249;"/> <!--latin small letter u
↳with grave, U+00F9 ISolat1 -->
265         <entity name="uacute" cdata="&#250;"/> <!--latin small letter u
↳with acute, U+00FA ISolat1 -->
266         <entity name="ucirc" cdata="&#251;"/> <!--latin small letter u
↳with circumflex, U+00FB ISolat1 -->
267         <entity name="uuml" cdata="&#252;"/> <!--latin small letter u
↳with diaeresis, U+00FC ISolat1 -->
268         <entity name="yacute" cdata="&#253;"/> <!--latin small letter y
↳with acute, U+00FD ISolat1 -->
269         <entity name="thorn" cdata="&#254;"/> <!--latin small letter
↳thorn, U+00FE ISolat1 -->
270         <entity name="yuml" cdata="&#255;"/> <!--latin small letter y
↳with diaeresis, U+00FF ISolat1 -->
271
272         <entity name="fnof" cdata="&#402;"/> <!--latin small f with hook
↳= function = florin, U+0192 ISOtech -->
273
274         <!-- Greek -->
275         <entity name="Alpha" cdata="&#913;"/> <!--greek capital letter
↳alpha, U+0391 -->
276         <entity name="Beta" cdata="&#914;"/> <!--greek capital letter
↳beta, U+0392 -->
277         <entity name="Gamma" cdata="&#915;"/> <!--greek capital letter
↳gamma, U+0393 ISOgrk3 -->
278         <entity name="Delta" cdata="&#916;"/> <!--greek capital letter
↳delta, U+0394 ISOgrk3 -->
279         <entity name="Epsilon" cdata="&#917;"/> <!--greek capital letter
↳epsilon, U+0395 -->
280         <entity name="Zeta" cdata="&#918;"/> <!--greek capital letter
↳zeta, U+0396 -->
281         <entity name="Eta" cdata="&#919;"/> <!--greek capital letter eta,
↳U+0397 -->
282         <entity name="Theta" cdata="&#920;"/> <!--greek capital letter
↳theta, U+0398 ISOgrk3 -->
283         <entity name="Iota" cdata="&#921;"/> <!--greek capital letter
↳iota, U+0399 -->
284         <entity name="Kappa" cdata="&#922;"/> <!--greek capital letter
↳kappa, U+039A -->
285         <entity name="Lambda" cdata="&#923;"/> <!--greek capital letter
↳lambda, U+039B ISOgrk3 -->
286         <entity name="Mu" cdata="&#924;"/> <!--greek capital letter mu,
↳U+039C -->
287         <entity name="Nu" cdata="&#925;"/> <!--greek capital letter nu,
↳U+039D -->
288         <entity name="Xi" cdata="&#926;"/> <!--greek capital letter xi,
↳U+039E ISOgrk3 -->
289         <entity name="Omicron" cdata="&#927;"/> <!--greek capital letter
↳omicron, U+039F -->
290         <entity name="Pi" cdata="&#928;"/> <!--greek capital letter pi,
↳U+03A0 ISOgrk3 -->
291         <entity name="Rho" cdata="&#929;"/> <!--greek capital letter rho,
↳U+03A1 -->

```



```

292      <!-- there is no Sigmaf, and no U+03A2 character either -->
293      <entity name="Sigma" cdata="&#931;" /> <!--greek capital letter_
↳sigma, U+03A3 ISOgrk3 -->
294      <entity name="Tau" cdata="&#932;" /> <!--greek capital letter tau,
↳U+03A4 -->
295      <entity name="Upsilon" cdata="&#933;" /> <!--greek capital letter_
↳upsilon, U+03A5 ISOgrk3 -->
296      <entity name="Phi" cdata="&#934;" /> <!--greek capital letter phi,
↳U+03A6 ISOgrk3 -->
297      <entity name="Chi" cdata="&#935;" /> <!--greek capital letter chi,
↳U+03A7 -->
298      <entity name="Psi" cdata="&#936;" /> <!--greek capital letter psi,
↳U+03A8 ISOgrk3 -->
299      <entity name="Omega" cdata="&#937;" /> <!--greek capital letter_
↳omega, U+03A9 ISOgrk3 -->
300
301      <entity name="alpha" cdata="&#945;" /> <!--greek small letter_
↳alpha, U+03B1 ISOgrk3 -->
302      <entity name="beta" cdata="&#946;" /> <!--greek small letter beta,
↳U+03B2 ISOgrk3 -->
303      <entity name="gamma" cdata="&#947;" /> <!--greek small letter_
↳gamma, U+03B3 ISOgrk3 -->
304      <entity name="delta" cdata="&#948;" /> <!--greek small letter_
↳delta, U+03B4 ISOgrk3 -->
305      <entity name="epsilon" cdata="&#949;" /> <!--greek small letter_
↳epsilon, U+03B5 ISOgrk3 -->
306      <entity name="zeta" cdata="&#950;" /> <!--greek small letter zeta,
↳U+03B6 ISOgrk3 -->
307      <entity name="eta" cdata="&#951;" /> <!--greek small letter eta,
↳U+03B7 ISOgrk3 -->
308      <entity name="theta" cdata="&#952;" /> <!--greek small letter_
↳theta, U+03B8 ISOgrk3 -->
309      <entity name="iota" cdata="&#953;" /> <!--greek small letter iota,
↳U+03B9 ISOgrk3 -->
310      <entity name="kappa" cdata="&#954;" /> <!--greek small letter_
↳kappa, U+03BA ISOgrk3 -->
311      <entity name="lambda" cdata="&#955;" /> <!--greek small letter_
↳lambda, U+03BB ISOgrk3 -->
312      <entity name="mu" cdata="&#956;" /> <!--greek small letter mu,
↳U+03BC ISOgrk3 -->
313      <entity name="nu" cdata="&#957;" /> <!--greek small letter nu,
↳U+03BD ISOgrk3 -->
314      <entity name="xi" cdata="&#958;" /> <!--greek small letter xi,
↳U+03BE ISOgrk3 -->
315      <entity name="omicron" cdata="&#959;" /> <!--greek small letter_
↳omicron, U+03BF NEW -->
316      <entity name="pi" cdata="&#960;" /> <!--greek small letter pi,
↳U+03C0 ISOgrk3 -->
317      <entity name="rho" cdata="&#961;" /> <!--greek small letter rho,
↳U+03C1 ISOgrk3 -->
318      <entity name="sigmaf" cdata="&#962;" /> <!--greek small letter_
↳final sigma, U+03C2 ISOgrk3 -->
319      <entity name="sigma" cdata="&#963;" /> <!--greek small letter_
↳sigma, U+03C3 ISOgrk3 -->
320      <entity name="tau" cdata="&#964;" /> <!--greek small letter tau,
↳U+03C4 ISOgrk3 -->
321      <entity name="upsilon" cdata="&#965;" /> <!--greek small letter_
↳upsilon, U+03C5 ISOgrk3 -->

```

```

322         <entity name="phi" cdata="&#966;"/> <!--greek small letter phi,
↪U+03C6 ISOgrk3 -->
323         <entity name="chi" cdata="&#967;"/> <!--greek small letter chi,
↪U+03C7 ISOgrk3 -->
324         <entity name="psi" cdata="&#968;"/> <!--greek small letter psi,
↪U+03C8 ISOgrk3 -->
325         <entity name="omega" cdata="&#969;"/> <!--greek small letter
↪omega, U+03C9 ISOgrk3 -->
326         <entity name="thetasym" cdata="&#977;"/> <!--greek small letter
↪theta symbol, U+03D1 NEW -->
327         <entity name="upsih" cdata="&#978;"/> <!--greek upsilon with hook
↪symbol, U+03D2 NEW -->
328         <entity name="piv" cdata="&#982;"/> <!--greek pi symbol, U+03D6
↪ISOgrk3 -->
329
330         <!-- General Punctuation -->
331         <entity name="bull" cdata="&#8226;"/> <!--bullet = black small
↪circle, U+2022 ISOpub -->
332         <!-- bullet is NOT the same as bullet operator, U+2219 -->
333         <entity name="hellip" cdata="&#8230;"/> <!--horizontal ellipsis =
↪three dot leader, U+2026 ISOpub -->
334         <entity name="prime" cdata="&#8242;"/> <!--prime = minutes = feet,
↪ U+2032 ISOTECH -->
335         <entity name="Prime" cdata="&#8243;"/> <!--double prime = seconds
↪= inches, U+2033 ISOTECH -->
336         <entity name="oline" cdata="&#8254;"/> <!--overline = spacing
↪overscore, U+203E NEW -->
337         <entity name="frasl" cdata="&#8260;"/> <!--fraction slash, U+2044
↪NEW -->
338
339         <!-- Letterlike Symbols -->
340         <entity name="weierp" cdata="&#8472;"/> <!--script capital P =
↪power set = Weierstrass p, U+2118 ISOamso -->
341         <entity name="image" cdata="&#8465;"/> <!--blackletter capital I
↪= imaginary part, U+2111 ISOamso -->
342         <entity name="real" cdata="&#8476;"/> <!--blackletter capital R =
↪real part symbol, U+211C ISOamso -->
343         <entity name="trade" cdata="&#8482;"/> <!--trade mark sign,
↪U+2122 ISOnum -->
344         <entity name="alefsym" cdata="&#8501;"/> <!--alef symbol = first
↪transfinite cardinal, U+2135 NEW -->
345         <!-- alef symbol is NOT the same as hebrew letter alef,
346         U+05D0 although the same glyph could be used to depict both
↪characters -->
347
348         <!-- Arrows -->
349         <entity name="larr" cdata="&#8592;"/> <!--leftwards arrow, U+2190
↪ISOnum -->
350         <entity name="uarr" cdata="&#8593;"/> <!--upwards arrow, U+2191
↪ISOnum-->
351         <entity name="rarr" cdata="&#8594;"/> <!--rightwards arrow,
↪U+2192 ISOnum -->
352         <entity name="darr" cdata="&#8595;"/> <!--downwards arrow, U+2193
↪ISOnum -->
353         <entity name="harr" cdata="&#8596;"/> <!--left right arrow,
↪U+2194 ISOamsa -->
354         <entity name="crarr" cdata="&#8629;"/> <!--downwards arrow with
↪corner leftwards

```

```

355                                     = carriage return, U+21B5 NEW -->
356         <entity name="lArr" cdata="&#8656;"/> <!--leftwards double arrow,
↪ U+21D0 ISOtech -->
357
358         <!-- ISO 10646 does not say that lArr is the same as the 'is implied
↪ by' arrow
359         but also does not have any other character for that function. So ?
↪ lArr can
360         be used for 'is implied by' as ISOtech suggests -->
361
362         <entity name="uArr" cdata="&#8657;"/> <!--upwards double arrow,
↪ U+21D1 ISOamsa -->
363         <entity name="rArr" cdata="&#8658;"/> <!--rightwards double arrow,
↪ U+21D2 ISOtech -->
364
365         <!-- ISO 10646 does not say this is the 'implies' character but does
↪ not have
366         another character with this function so ?
367         rArr can be used for 'implies' as ISOtech suggests -->
368
369         <entity name="dArr" cdata="&#8659;"/> <!--downwards double arrow,
↪ U+21D3 ISOamsa -->
370         <entity name="hArr" cdata="&#8660;"/> <!--left right double arrow,
↪ U+21D4 ISOamsa -->
371
372         <!-- Mathematical Operators -->
373         <entity name="forall" cdata="&#8704;"/> <!--for all, U+2200
↪ ISOtech -->
374         <entity name="part" cdata="&#8706;"/> <!--partial differential,
↪ U+2202 ISOtech -->
375         <entity name="exist" cdata="&#8707;"/> <!--there exists, U+2203
↪ ISOtech -->
376         <entity name="empty" cdata="&#8709;"/> <!--empty set = null set =
↪ diameter, U+2205 ISOamsa -->
377         <entity name="nabla" cdata="&#8711;"/> <!--nabla = backward
↪ difference, U+2207 ISOtech -->
378         <entity name="isin" cdata="&#8712;"/> <!--element of, U+2208
↪ ISOtech -->
379         <entity name="notin" cdata="&#8713;"/> <!--not an element of,
↪ U+2209 ISOtech -->
380         <entity name="ni" cdata="&#8715;"/> <!--contains as member,
↪ U+220B ISOtech -->
381
382         <!-- should there be a more memorable name than 'ni'? -->
383         <entity name="prod" cdata="&#8719;"/> <!--n-ary product = product
↪ sign, U+220F ISOamsb -->
384
385         <!-- prod is NOT the same character as U+03A0 'greek capital letter pi
↪ ' though
386         the same glyph might be used for both -->
387
388         <entity name="sum" cdata="&#8721;"/> <!--n-ary sumation, U+2211
↪ ISOamsb -->
389
390         <!-- sum is NOT the same character as U+03A3 'greek capital letter
↪ sigma'
391         though the same glyph might be used for both -->
392

```

```

393      <entity name="minus" cdata="&#8722;"/> <!--minus sign, U+2212_
↪ISOtech -->
394      <entity name="lowast" cdata="&#8727;"/> <!--asterisk operator,
↪U+2217 ISOtech -->
395      <entity name="radic" cdata="&#8730;"/> <!--square root = radical_
↪sign, U+221A ISOtech -->
396      <entity name="prop" cdata="&#8733;"/> <!--proportional to, U+221D_
↪ISOtech -->
397      <entity name="infin" cdata="&#8734;"/> <!--infinity, U+221E_
↪ISOtech -->
398      <entity name="ang" cdata="&#8736;"/> <!--angle, U+2220 ISOamso -->
399      <entity name="and" cdata="&#8743;"/> <!--logical and = wedge,
↪U+2227 ISOtech -->
400      <entity name="or" cdata="&#8744;"/> <!--logical or = vee, U+2228_
↪ISOtech -->
401      <entity name="cap" cdata="&#8745;"/> <!--intersection = cap,
↪U+2229 ISOtech -->
402      <entity name="cup" cdata="&#8746;"/> <!--union = cup, U+222A_
↪ISOtech -->
403      <entity name="int" cdata="&#8747;"/> <!--integral, U+222B ISOtech_
↪-->
404      <entity name="there4" cdata="&#8756;"/> <!--therefore, U+2234_
↪ISOtech -->
405      <entity name="sim" cdata="&#8764;"/> <!--tilde operator = varies_
↪with = similar to, U+223C ISOtech -->
406
407      <!-- tilde operator is NOT the same character as the tilde, U+007E,
408           although the same glyph might be used to represent both -->
409
410      <entity name="cong" cdata="&#8773;"/> <!--approximately equal to,
↪U+2245 ISOtech -->
411      <entity name="asymp" cdata="&#8776;"/> <!--almost equal to =
↪asymptotic to, U+2248 ISOamsr -->
412      <entity name="ne" cdata="&#8800;"/> <!--not equal to, U+2260_
↪ISOtech -->
413      <entity name="equiv" cdata="&#8801;"/> <!--identical to, U+2261_
↪ISOtech -->
414      <entity name="le" cdata="&#8804;"/> <!--less-than or equal to,
↪U+2264 ISOtech -->
415      <entity name="ge" cdata="&#8805;"/> <!--greater-than or equal to,
↪U+2265 ISOtech -->
416      <entity name="sub" cdata="&#8834;"/> <!--subset of, U+2282_
↪ISOtech -->
417      <entity name="sup" cdata="&#8835;"/> <!--superset of, U+2283_
↪ISOtech -->
418
419      <!-- note that nsup, 'not a superset of, U+2283' is not covered by_
↪the Symbol
420           font encoding and is not included. Should it be, for symmetry?
421           It is in ISOamsn -->
422
423      <entity name="nsup" cdata="&#8836;"/> <!--not a subset of, U+2284_
↪ISOamsn -->
424      <entity name="sube" cdata="&#8838;"/> <!--subset of or equal to,
↪U+2286 ISOtech -->
425      <entity name="supe" cdata="&#8839;"/> <!--superset of or equal to,
↪U+2287 ISOtech -->
426      <entity name="oplus" cdata="&#8853;"/> <!--circled plus = direct_
↪sum, U+2295 ISOamsb -->

```

```

427         <entity name="otimes" cdata="&#8855;"/> <!--circled times =_
↳vector product, U+2297 ISOamsb -->
428         <entity name="perp" cdata="&#8869;"/> <!--up tack = orthogonal to_
↳= perpendicular, U+22A5 ISOTech -->
429         <entity name="sdot" cdata="&#8901;"/> <!--dot operator, U+22C5_
↳ISOamsb -->
430         <!-- dot operator is NOT the same character as U+00B7 middle dot -->
431
432         <!-- Miscellaneous Technical -->
433         <entity name="lceil" cdata="&#8968;"/> <!--left ceiling = apl_
↳upstile, U+2308 ISOamsc -->
434         <entity name="rceil" cdata="&#8969;"/> <!--right ceiling, U+2309_
↳ISOamsc -->
435         <entity name="lfloor" cdata="&#8970;"/> <!--left floor = apl_
↳downstile, U+230A ISOamsc -->
436         <entity name="rfloor" cdata="&#8971;"/> <!--right floor, U+230B_
↳ISOamsc -->
437         <entity name="lang" cdata="&#9001;"/> <!--left-pointing angle_
↳bracket = bra, U+2329 ISOTech -->
438         <!-- lang is NOT the same character as U+003C 'less than'
439         or U+2039 'single left-pointing angle quotation mark' -->
440         <entity name="rang" cdata="&#9002;"/> <!--right-pointing angle_
↳bracket = ket, U+232A ISOTech -->
441         <!-- rang is NOT the same character as U+003E 'greater than' or_
↳U+203A 'single right-pointing angle quotation mark' -->
442
443         <!-- Geometric Shapes -->
444         <entity name="loz" cdata="&#9674;"/> <!--lozenge, U+25CA ISOpub --
↳>
445
446         <!-- Miscellaneous Symbols -->
447         <entity name="spades" cdata="&#9824;"/> <!--black spade suit,_
↳U+2660 ISOpub -->
448         <!-- black here seems to mean filled as opposed to hollow -->
449         <entity name="clubs" cdata="&#9827;"/> <!--black club suit =_
↳shamrock, U+2663 ISOpub -->
450         <entity name="hearts" cdata="&#9829;"/> <!--black heart suit =_
↳valentine, U+2665 ISOpub -->
451         <entity name="diams" cdata="&#9830;"/> <!--black diamond suit,_
↳U+2666 ISOpub -->
452
453         <entity name="quot" cdata="&#34;"/> <!--quotation mark = API_
↳quote, U+0022 ISOnum -->
454         <!-- Latin Extended-A -->
455         <entity name="OElig" cdata="&#338;"/> <!--latin capital ligature_
↳OE, U+0152 ISolat2 -->
456         <entity name="oelig" cdata="&#339;"/> <!--latin small ligature_
↳oe, U+0153 ISolat2 -->
457         <!-- ligature is a misnomer, this is a separate character in some_
↳languages -->
458         <entity name="Scaron" cdata="&#352;"/> <!--latin capital letter_
↳S with caron, U+0160 ISolat2 -->
459         <entity name="scaron" cdata="&#353;"/> <!--latin small letter s_
↳with caron, U+0161 ISolat2 -->
460         <entity name="Yuml" cdata="&#376;"/> <!--latin capital letter Y_
↳with diaeresis, U+0178 ISolat2 -->
461
462         <!-- Spacing Modifier Letters -->

```

```

463         <entity name="circ" cdata="&#710;" /> <!--modifier letter,
↳circumflex accent, U+02C6 ISOpub -->
464         <entity name="tilde" cdata="&#732;" /> <!--small tilde, U+02DC
↳ISODia -->
465
466         <!-- General Punctuation -->
467         <entity name="ensp" cdata="&#8194;" /> <!--en space, U+2002 ISOpub
↳-->
468         <entity name="emsp" cdata="&#8195;" /> <!--em space, U+2003 ISOpub
↳-->
469         <entity name="thinsp" cdata="&#8201;" /> <!--thin space, U+2009
↳ISOpub -->
470         <entity name="zwnj" cdata="&#8204;" /> <!--zero width non-joiner,
↳U+200C NEW RFC 2070 -->
471         <entity name="zwj" cdata="&#8205;" /> <!--zero width joiner,
↳U+200D NEW RFC 2070 -->
472         <entity name="lrm" cdata="&#8206;" /> <!--left-to-right mark,
↳U+200E NEW RFC 2070 -->
473         <entity name="rlm" cdata="&#8207;" /> <!--right-to-left mark,
↳U+200F NEW RFC 2070 -->
474         <entity name="ndash" cdata="&#8211;" /> <!--en dash, U+2013 ISOpub
↳-->
475         <entity name="mdash" cdata="&#8212;" /> <!--em dash, U+2014 ISOpub
↳-->
476         <entity name="lsquo" cdata="&#8216;" /> <!--left single quotation
↳mark, U+2018 ISOnum -->
477         <entity name="rsquo" cdata="&#8217;" /> <!--right single quotation
↳mark, U+2019 ISOnum -->
478         <entity name="sbquo" cdata="&#8218;" /> <!--single low-9 quotation
↳mark, U+201A NEW -->
479         <entity name="ldquo" cdata="&#8220;" /> <!--left double quotation
↳mark, U+201C ISOnum -->
480         <entity name="rdquo" cdata="&#8221;" /> <!--right double quotation
↳mark, U+201D ISOnum -->
481         <entity name="bdquo" cdata="&#8222;" /> <!--double low-9 quotation
↳mark, U+201E NEW -->
482         <entity name="dagger" cdata="&#8224;" /> <!--dagger, U+2020 ISOpub
↳-->
483         <entity name="Dagger" cdata="&#8225;" /> <!--double dagger, U+2021
↳ISOpub -->
484         <entity name="permil" cdata="&#8240;" /> <!--per mille sign,
↳U+2030 ISotech -->
485         <entity name="lsaquo" cdata="&#8249;" /> <!--single left-pointing
↳angle quotation mark, U+2039 ISO proposed -->
486         <!-- lsaquo is proposed but not yet ISO standardized -->
487         <entity name="rsaquo" cdata="&#8250;" /> <!--single right-pointing
↳angle quotation mark, U+203A ISO proposed -->
488         <!-- rsaquo is proposed but not yet ISO standardized -->
489         <entity name="euro" cdata="&#8364;" /> <!--euro sign, U+20AC NEW -
↳-->
490     </html-entities>
491
492 </anti-samy-rules>

```

## 6.2.8 Fichier /vitam/conf/<composant>/vitam.conf

```

1 secret : {{plateforme_secret}}
2 filterActivation : {{ vitam_struct.secret_platform }}
3 {% if inventory_hostname in groups['hosts-processing'] %}
4 distributeurBatchSize: 800
5 workerBulkSize: 16
6 {% endif %}
7 intervalDelayCheckIdle : 5000
8 maxDelayUnusedConnection : 5000
9 delayValidationAfterInactivity : 2500
10

```

Ce fichier permet de définir le secret de plate-forme.

## 6.2.9 Fichier /vitam/conf/<composant>/vitam.metrics.conf

```

1 # Fichier de configuration des métriques
2 #
3 # Les différents clés disponibles pour ce fichier de configuration sont les_
4 ↪ suivantes :
5 #
6 # metricsJersey: true / false           Active ou non les métriques Jersey
7 # metricsJVM: true / false             Active ou non les métriques JVM
8 #
9 # metricReporter: ELASTICSEARCH | LOGBACK | NONE           défini le_
10 ↪ type de reporter
11 # metricReporterInterval: int > 0           défini 1
12 ↪ 'interval entre chaque reporting
13 # metricReporterIntervalUnit: TimeUnit (ex: SECONDS, MINUTES...) défini le_
14 ↪ type d'interval
15 #
16 # Si le reporter est de type LOGBACK, la clé suivante est configurable:
17 # metricLogLevel: DEBUG | INFO | WARN | ERROR ...           défini le_
18 ↪ niveau de log Logback
19 #
20 # Si le reporter est de type ELASTICSEARCH, la clé suivante est obligatoire :
21 #
22 # (un tableau avec les différentes adresses des bases ElasticSearch)
23 # metricReporterHosts:
24 #     - 127.0.0.1:9201
25 #     - 0.0.0.0:80
26 #     - 8.8.8.8:22
27
28 {% if (groups['hosts-logstash'] | length) > 0 %}
29 metricsJersey: true
30 metricsJVM: true
31
32 metricReporter: ELASTICSEARCH
33 metricReporterHosts:
34 {% for host in groups['hosts-elasticsearch-log'] %}
35     - "{{ hostvars[host]['ip_admin'] }}:{{ elasticsearch.log.port_http }}"
36 {% endfor %}
37
38 metricLogLevel: DEBUG
39 metricReporterInterval: 1
40 metricReporterIntervalUnit: MINUTES

```

```
35 {% endif %}
```

## 6.2.10 Fichier /vitam/conf/<composant>/java.security

```
1 # Use Bouncy Castle Provider when it is available
2 security.provider.9=org.bouncycastle.jce.provider.BouncyCastleProvider
3
4 # Override the default list of Centos 7 that disable Elliptic Curved Based Algorithms
5 jdk.tls.disabledAlgorithms="SSLv3, RC4, MD5withRSA, DH keySize < 768,RSA keySize <
  ↪2048"
```

## 6.3 Composants

### 6.3.1 access external

#### 6.3.1.1 Présentation

Access-external est le composant d'interface entre *VITAM* et un *SIA* client, permettant de réaliser des recherches sur les objets archivés et les journaux. Il permet également quelques fonctions d'administration, en particulier les chargements des référentiels.

Rôle :

- Exposer les API publiques du système
- Sécuriser l'accès aux API de VITAM

#### 6.3.1.2 Configuration / fichiers utiles

Les fichiers de configuration sont gérés par les procédures d'installation ou de mise à niveau de l'environnement *VITAM*. Se référer au *DIN*.

Les fichiers de configuration sont définis sous /vitam/conf/access-external.

##### 6.3.1.2.1 Fichier access-external.conf

```
authentication: true
jettyConfig: jetty-config.xml
tenants: [ "{{ vitam_tenant_ids | join(' ', '') }}" ]
tenantFilter : true
```

##### 6.3.1.2.2 Fichier access-internal-client.conf

```
serverHost: {{vitam.accessinternal.host}}
serverPort: {{vitam.accessinternal.port_http}}
```



### 6.3.1.2.3 Fichier shiro.ini

```
[main]

x509 = fr.gouv.vitam.common.auth.web.filter.X509AuthenticationFilter

x509.useHeader = {{vitam_struct.https_enabled}}

x509credentialsMatcher = fr.gouv.vitam.common.auth.core.authc.
↳X509CredentialsSha256Matcher

x509Realm = fr.gouv.vitam.common.auth.core.realm.X509KeystoreFileRealm
x509Realm.grantedKeyStoreName = {{vitam_folder_conf}}/grantedstore_{{ vitam_struct.
↳vitam_component }}.jks
x509Realm.grantedKeyStorePassphrase = {{password_grantedstore}}
x509Realm.trustedKeyStoreName = {{vitam_folder_conf}}/truststore_{{ vitam_struct.
↳vitam_component }}.jks
x509Realm.trustedKeyStorePassphrase = {{password_truststore}}
x509Realm.credentialsMatcher = $x509credentialsMatcher
securityManager.realm = $x509Realm
securityManager.subjectDAO.sessionStorageEvaluator.sessionStorageEnabled = false
[urls]
/access-external/v1/**= x509
/admin-external/v1/**= x509
```

### 6.3.1.2.4 Fichier functional-administration-client.conf

```
serverHost: {{vitam.functional_administration.host}}
serverPort: {{vitam.functional_administration.port_http}}
```

### 6.3.1.3 Opérations

- Démarrage du service

En tant qu'utilisateur root : `systemctl start vitam-access-external`

- Arrêt du service

En tant qu'utilisateur root : `systemctl stop vitam-access-external`

- Sauvegarde du service

Ce service ne nécessite pas de sauvegarde particulière.

- Supervision du service

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:<port>/access-external/v1/status

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:<port admin>/admin/v1/status

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

- cas des batches

N/A

## 6.3.2 access-internal

### 6.3.2.1 Présentation du composant

Access-internal est le composant *VITAM*, permettant de réaliser des recherches et consultations sur les objets archivés et les journaux. Il permet également de modifier les informations d'un *ArchiveUnit*.

Rôle :

- Permettre l'accès aux données du système VITAM

Fonction :

- Exposition des fonctions de recherche d'archives offertes par metadata ;
- Exposition des fonctions de parcours de journaux offertes par logbook.

### 6.3.2.2 Configuration / fichiers utiles

Les fichiers de configuration sont gérés par les procédures d'installation ou de mise à niveau de l'environnement *VITAM*. Se référer au *DIN*.

Les fichiers de configuration sont définis sous `/vitam/conf/access`.

#### 6.3.2.2.1 Fichier `access.conf`

Ce fichier permet de définir l'URL d'accès au metadata server.

```
urlMetaData: {{vitam.metadata | client_url}}
urlWorkspace: {{vitam.workspace | client_url}}
urlProcessing: {{vitam.processing | client_url}}
jettyConfig: jetty-config.xml
tenants: [ "{{ vitam_tenant_ids | join(' ', ' ') }}" ]
```

#### 6.3.2.2.2 Fichier `storage-client.conf`

Ce fichier permet de définir l'accès au storage-engine.

```
serverHost: {{ vitam.storageengine.host }}
serverPort: {{ vitam.storageengine.port_http }}
```

#### 6.3.2.2.3 Fichier `metadata-client.conf`

Ce fichier permet de définir l'accès au storage-engine.

```
serverHost: {{ vitam.metadata.host }}
serverPort: {{ vitam.metadata.port_http }}
```

### 6.3.2.3 Opérations

- Démarrage du service

En tant qu'utilisateur root : `systemctl start vitam-access-internal`

- Arrêt du service

En tant qu'utilisateur root : `systemctl stop vitam-access-internal`

- Sauvegarde du service

Ce service ne nécessite pas de sauvegarde particulière.

- Supervision du service

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:<port>/access/v1/status

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:<port admin>/admin/v1/status

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

- cas des batches

N/A

## 6.3.3 Cerebro

### 6.3.3.1 Présentation

Cerebro est un utilitaire de supervision de l'état d'un cluster ElasticSearch.

### 6.3.3.2 Configuration / fichiers utiles

Le fichiers de configuration est `/vitam/conf/cerebro/application.conf`.

### 6.3.3.3 Opérations

- Démarrage du service

Les commandes suivantes sont à passer sur les différentes machines hébergeant le composant vitam-elasticsearch-cerebro.

En tant qu'utilisateur root : `systemctl start vitam-elasticsearch-cerebro`

- Arrêt du service

Les commandes suivantes sont à passer sur les différentes machines constituant le composant vitam-elasticsearch-cerebro.

En tant qu'utilisateur root : `systemctl stop vitam-elasticsearch-cerebro`

- Sauvegarde du service

N/A

- Supervision du service

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:9000/cerebro

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

N/A

- cas des batches

N/A

## 6.3.4 common-plugin

### 6.3.4.1 Présentation du composant

common-plugin est le composant permettant de réaliser des plugins sans appel à des package privé . Rôle :

- l'objet de ce common-plugin n'est pas que de fournir des interfaces à implémenter mais aussi les classes d'implémentations imposées par Vitam pour réaliser des plugins.

Fonction :

- Exposition interfaces à implémenter et les classes d'implémentations pour réaliser des plugins .

### 6.3.4.2 Classes utiles

L'Objectif de Plugin Common est d'inclure tous les classes utiles afin de créer un plugin à partir de ce package .

Les classes de model sont définis sous /vitam/common/model.

#### 6.3.4.2.1 Classe Item Status

Ce classe permet de retourner le statut d'un Item.

#### 6.3.4.2.2 Classe VitamAutoCloseable

Le mot clé try-with-resources garantit que chaque ressource sera fermée lorsqu'elle n'est plus utilisée. Une ressource et un objet qui implémente l'interface VitamAutoCloseable. Il est donc possible d'utiliser une instance de ces interfaces avec le mot clé try-with-resources.

Les classes de common parameter sont définis sous /vitam/common/parameter.

#### 6.3.4.2.3 Classe ParameterHelper

Ce classe permet de faire un check sur les paramètres et avoir le tenant parameter de session vitam .

#### **6.3.4.2.4 Classe VitamParameter**

Cet interface permet d'aider à créer des nouveaux paramètres liés au classes .

Les classes de common exception sont définis sous /vitam/processing/common/exception.

#### **6.3.4.2.5 Classe ProcessingException**

Ce classe est le classe père de tous les Vitam Processing Exception .

Les classes de model common processing sont définis sous /vitam/processing/common/model.

#### **6.3.4.2.6 Classe IOParameter**

Ce class permet de définir les paramètres Input et Output pour une action et une step .

#### **6.3.4.2.7 Classe ProcessingUri**

Ce classe permet de formater le processing URI .

#### **6.3.4.2.8 Classe UriPrefix**

C'est le Handler IO

Les classes des paramètres common sont définis sous /vitam/processing/common/parameter.

#### **6.3.4.2.9 Classe AbstractWorkerParameters**

C'est une implémentation abstraite de tous les paramètres de workers .

#### **6.3.4.2.10 Classe DefaultWorkerParameters**

Ce classe permet de définir les paramètres par défaut d'un worker.

#### **6.3.4.2.11 Classe WorkerParameterName**

Ce classe inclut une énumération avec tous les noms des paramètres d'un worker .

#### **6.3.4.2.12 Classe WorkerParameters**

Ce classe permet de définir les paramètres de worker.

#### **6.3.4.2.13 Classe WorkerParametersDeserializer**

Ce classe permet de définir les paramètres d'un worker deserializer.

#### 6.3.4.2.14 Classe `WorkerParametersFactory`

Ce classe permet de définir les paramètres d'un worker Factory.

#### 6.3.4.2.15 Classe `WorkerParametersSerializer`

Ce classe permet de définir les paramètres de Worker Serializer.

Les classes de model sont définis sous `/vitam/worker/common`.

#### 6.3.4.2.16 Interface `HandlerIO`

Cet interface permet de définir les paramètres in et out de tous les Handlers.

Les classes de l'api sont définis sous `/vitam/worker/core/api`.

#### 6.3.4.2.17 Classe `WorkerAction`

C'est l'interface contrat de tous les actions Handler event. Un action Handler doit implémenter cette interface .

Les classes de l'implémentation sont définis sous `/vitam/worker/core/impl`.

#### 6.3.4.2.18 Classe `HandlerIOImpl`

Ce classe définit les paramètres in et out d'un Handler

---

How to use : Pour créer un Plugin :

- extends Abstract Class Action Handler
- implementer l'interface `VitamAutoCloseable` pour garantir qu'une ressource sera fermée lorsqu'elle n'est plus utilisée.
- Un constructeur par défaut
- **redéfinir la méthode execute de l'Action Handler :**
  - Paramètre `WorkerParameters` et `Handler IO`
  - type de retour `Item Status`
  - throws `Processing Exception`
- **faire l'override de méthode `CheckMandatoryIOParameter`**
  - Paramètre `Handler IO`
  - throws `Processing Exception`

### 6.3.5 consul

#### 6.3.5.1 Présentation

Consul est un DNS applicatif.

### 6.3.5.1.1 Cas serveur

Le serveur Consul fédère les agents dans leurs requêtes “DNS-like” et permet de rebondir sur un DNS externe, s’il ne permet pas de lui-même, de faire la résolution.

### 6.3.5.1.2 Cas agent

L’agent Consul annonce aux serveurs les services qu’il permet de porter et “checke” régulièrement l’état de ces services.

## 6.3.5.2 Configuration / fichiers utiles

Les fichiers de configuration sont gérés par les procédures d’installation ou de mise à niveau de l’environnement *VITAM*. Se référer au *DIN*.

### 6.3.5.2.1 Cas des applicatifs monitorés par Consul

Pour chaque composant *VITAM* nécessitant une supervision de la part de Consul, un fichier est installé sur l’agent de la machine sous `vitam/conf/consul/service-<composant>.json` et est basé sur ce squelette :

#### 6.3.5.2.1.1 Fichier `/vitam/conf/consul/service-<composant>.json`

```

1 {
2   "service": {
3     "name": "{{ vitam_struct.vitam_component }}",
4     "address": "{{ ip_service }}",
5     {% if vitam_struct.https_enabled==true %}
6     "port": {{ vitam_struct.port_https }},
7     {% else %}
8     "port": {{ vitam_struct.port_http }},
9     {% endif %}
10    "enableTagOverride": false,
11    "checks": [
12      {
13        "name": "{{ vitam_struct.vitam_component }}: business service check",
14        {% if vitam_struct.https_enabled==true %}
15        "notes": "HTTPS port opened",
16        "tcp": "{{ ip_service }}:{{ vitam_struct.port_https }}",
17        {% else %}
18        "notes": "HTTP port opened",
19        "tcp": "{{ ip_service }}:{{ vitam_struct.port_http }}",
20        {% endif %}
21        "interval": "1s"
22      },
23      {
24        "name": "{{ vitam_struct.vitam_component }} : admin service check",
25        "notes": "Status admin : /admin/v1/status",
26        "http": "http://{{ ip_admin }}:{{ vitam_struct.port_admin }}/admin/v1/status",
27        "interval": "1s"
28      }
29    {% if (vitam_struct.vitam_component == 'worker') or (vitam_struct.vitam_component ==
    → 'ingest-external') %}

```

```

30     ,
31     {
32         "name": "Siegfried check",
33         "notes": "Is siegfried running ?",
34         "tcp": "localhost:{{ siegfried.port }}",
35         "interval": "1s"
36     }
37 {% endif %}
38 {% if vitam_struct.antivirus is defined %}
39     ,
40     {
41         "name": "Antivirus check",
42         "notes": "Is {{ vitam_struct.antivirus }} running ?",
43         "args": ["{{ vitam_folder_conf }}/scan-{{ vitam_struct.antivirus }}.sh", "{{ _
44 →vitam_folder_conf }}/scan-{{ vitam_struct.antivirus }}.sh"],
45         "interval": "30s",
46         "timeout": "5s"
47     }
48 {% endif %}
49 ]
50 }

```

### 6.3.5.3 Opérations

- Démarrage du service

En tant qu'utilisateur root : `systemctl start vitam-consul`

- Arrêt du service

En tant qu'utilisateur root : `systemctl stop vitam-consul`

**Avvertissement :** en cas de redémarrage du cluster serveur consul, il faut procéder à un arrêt/relance par serveur avant de passer au suivant.

- Sauvegarde du service

Ce service ne nécessite pas de sauvegarde particulière.

- Logs

Les logs applicatifs sont envoyés par rsyslog à la solution de centralisation des logs ; il est néanmoins possible d'en virionner une représentation par la commande :

```
journalctl --unit vitam-consul
```

- Supervision du service

Consul possède une IHM permettant de superviser l'ensemble des services qu'il couvre.

`http(s) ://<adresse> :<port>/ui`

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes



- cas des batches

N/A

## 6.3.6 elasticsearch chaîne de log

### 6.3.6.1 Présentation

Elasticsearch-log est une instance de la base d'indexation elasticsearch stockant les informations suivantes :

- les logs des applications VITAM ;
- les logs des applications du sous-système de centralisation des logs ;
- les métriques applicatives.

### 6.3.6.2 Configuration / fichiers utiles

Se reporter au *DIN*, qui configure le cluster ElastciSearch.

Les fichiers de configuration sous sous /vitam/conf/elasticsearch-log.

#### 6.3.6.2.1 Fichier logging.yml

```
# you can override this using by setting a system property, for example -Des.logger.
↪level=DEBUG
es.logger.level: INFO
rootLogger: ${es.logger.level}, console, file
logger:
  # log action execution errors for easier debugging
  action: DEBUG

  # deprecation logging, turn to DEBUG to see them
  deprecation: INFO, deprecation_log_file

  # reduce the logging for aws, too much is logged under the default INFO
  com.amazonaws: WARN
  # aws will try to do some sketchy JMX stuff, but its not needed.
  com.amazonaws.jmx.SdkMBeanRegistrySupport: ERROR
  com.amazonaws.metrics.AwsSdkMetrics: ERROR

  org.apache.http: INFO

  # gateway
  #gateway: DEBUG
  #index.gateway: DEBUG

  # peer shard recovery
  #indices.recovery: DEBUG

  # discovery
  #discovery: TRACE

  index.search.slowlog: TRACE, index_search_slow_log_file
  index.indexing.slowlog: TRACE, index_indexing_slow_log_file
```

```

additivity:
  index.search.slowlog: false
  index.indexing.slowlog: false
  deprecation: false

appender:
  console:
    type: console
    layout:
      type: consolePattern
      conversionPattern: "[%d{ISO8601}][%-5p][%-25c] %m%n"

  file:
    type: dailyRollingFile
    file: ${path.logs}/${cluster.name}.log
    datePattern: "'. 'yyyy-MM-dd"
    layout:
      type: pattern
      conversionPattern: "[%d{ISO8601}][%-5p][%-25c] %.10000m%n"

  syslog:
    type: org.apache.log4j.net.SyslogAppender
    syslogHost: localhost:514
    layout:
      type: pattern
      conversionPattern: "{ ${ composant.cluster_name }}: [%d{ISO8601}][%-5p][%-25c] %.
↪10000m%n"

  # Use the following log4j-extras RollingFileAppender to enable gzip compression of
↪log files.
  # For more information see https://logging.apache.org/log4j/extras/apidocs/org/
↪apache/log4j/rolling/RollingFileAppender.html
  #file:
  #type: extrasRollingFile
  #file: ${path.logs}/${cluster.name}.log
  #rollingPolicy: timeBased
  #rollingPolicy.FileNamePattern: ${path.logs}/${cluster.name}.log.%d{yyyy-MM-dd}.gz
  #layout:
  #type: pattern
  #conversionPattern: "[%d{ISO8601}][%-5p][%-25c] %m%n"

deprecation_log_file:
  type: dailyRollingFile
  file: ${path.logs}/${cluster.name}_deprecation.log
  datePattern: "'. 'yyyy-MM-dd"
  layout:
    type: pattern
    conversionPattern: "[%d{ISO8601}][%-5p][%-25c] %m%n"

index_search_slow_log_file:
  type: dailyRollingFile
  file: ${path.logs}/${cluster.name}_index_search_slowlog.log
  datePattern: "'. 'yyyy-MM-dd"
  layout:
    type: pattern
    conversionPattern: "[%d{ISO8601}][%-5p][%-25c] %m%n"

index_indexing_slow_log_file:

```

```

type: dailyRollingFile
file: ${path.logs}/${cluster.name}_index_indexing_slowlog.log
datePattern: "'. 'yyy-MM-dd"
layout:
  type: pattern
  conversionPattern: "[%d{ISO8601}][%-5p][%-25c] %m%n"

```

### 6.3.6.2.2 Fichier elasticsearch.yml

```

# ===== Elasticsearch Configuration =====
#
# NOTE: Elasticsearch comes with reasonable defaults for most settings.
#       Before you set out to tweak and tune the configuration, make sure you
#       understand what are you trying to accomplish and the consequences.
#
# The primary way of configuring a node is via this file. This template lists
# the most important settings you may want to configure for a production cluster.
#
# Please see the documentation for further information on configuration options:
# <http://www.elastic.co/guide/en/elasticsearch/reference/current/setup-configuration.
↪html>
#
# ----- Cluster -----
#
# Use a descriptive name for your cluster:
#
cluster.name: {{ composant.cluster_name }}
#
# ----- Node -----
#
# Use a descriptive name for the node:
#
node.name: {{ inventory_hostname }}
#
# Add custom attributes to the node:
#
node.rack: r1
#
# ----- Paths -----
#
# Path to directory where to store the data (separate multiple locations by comma):
#
path.data: {{ elasticsearch_data_dir }}
#
# Path to log files:
#
path.logs: {{ elasticsearch_log_dir }}
#
# ----- Memory -----
#
# Lock the memory on startup:
#
bootstrap.mlockall: true
#
# Make sure that the 'ES_HEAP_SIZE' environment variable is set to about half the_
↪memory

```

```

# available on the system and that the owner of the process is allowed to use this_
↪limit.
#
# Elasticsearch performs poorly when the system is swapping the memory.
#
# ----- Network -----
#
# Set the bind address to a specific IP (IPv4 or IPv6):
#
# Note : if installing to localhost, notably a docker container, we need to bind_
↪larger than localhost
{% if inventory_hostname == "localhost" %}
network.host: 0.0.0.0
http.cors.enabled: true
http.cors.allow-origin: "*"
{% else %}
# KWA TODO: Check it again (ansible_hostname VS inventory_hostname VS ip_service)
network.host: {{ ip_admin }}
{% endif %}
# Set a custom port for HTTP:
#
http.port: {{ composant.port_http }}
#network.port: {{ composant.port_tcp }}
transport.tcp.port: {{ composant.port_tcp }}

# For more information, see the documentation at:
# <http://www.elastic.co/guide/en/elasticsearch/reference/current/modules-network.
↪html>
#
# ----- Discovery -----
#
# Pass an initial list of hosts to perform discovery when new node is started:
# The default list of hosts is ["127.0.0.1", "[::1]"]
#
discovery.zen.ping.unicast.hosts: [ {% for host in groups['hosts-elasticsearch-log']
↪%} "{{ hostvars[host]['ip_admin'] }}" {% if not loop.last %},{% endif %} {% endfor %} ]
#
# Prevent the "split brain" by configuring the majority of nodes (total number of_
↪nodes / 2 + 1):
#
# discovery.zen.minimum_master_nodes: 3
#
# For more information, see the documentation at:
# <http://www.elastic.co/guide/en/elasticsearch/reference/current/modules-discovery.
↪html>
#
# ----- Gateway -----
#
# Block initial recovery after a full cluster restart until N nodes are started:
#
# gateway.recover_after_nodes: 3
#
# For more information, see the documentation at:
# <http://www.elastic.co/guide/en/elasticsearch/reference/current/modules-gateway.
↪html>
#
# ----- Various -----
#

```

```
# Disable starting multiple nodes on a single system:
#
# node.max_local_storage_nodes: 1
#
# Require explicit names when deleting indices:
#
# action.destructive_requires_name: true
```

### 6.3.6.2.3 Fichier sysconfig/elasticsearch

```
#####
# Elasticsearch
#####

# Elasticsearch home directory
#ES_HOME=/usr/share/elasticsearch

# Elasticsearch configuration directory
CONF_DIR={{ vitam_defaults.folder.root_path }}/conf/{{ composant.cluster_name }}

# Elasticsearch data directory
#DATA_DIR={{ vitam_defaults.folder.root_path }}/data/{{ composant.cluster_name }}

# Elasticsearch logs directory
#LOG_DIR={{ vitam_defaults.folder.root_path }}/log/{{ composant.cluster_name }}

# Elasticsearch PID directory
#PID_DIR=/var/run/{{ composant.cluster_name }}

# Heap size defaults to 256m min, 1g max
# Set ES_HEAP_SIZE to 50% of available RAM, but no more than 31g
#ES_JAVA_OPTS=

# Heap new generation
#ES_HEAP_NEWSIZE=

# Maximum direct memory
#ES_DIRECT_SIZE=

# Additional Java OPTS
ES_JAVA_OPTS="-Xms{{ elasticsearch_memory }} -Xmx{{ elasticsearch_memory }} -XX:
↪+UseGCLogFileRotation -XX:NumberOfGCLogFiles=10 -XX:GCLogFileSize=10M -XX:
↪+PrintGCDetails -XX:+PrintGCApplicationStoppedTime"

# Configure restart on package upgrade (true, every other setting will lead to not_
↪restarting)
#RESTART_ON_UPGRADE=true

# Path to the GC log file
#ES_GC_LOG_FILE={{ vitam_defaults.folder.root_path }}/log/{{ composant.cluster_name }}
↪/gc.log

#####
# Elasticsearch service
#####
```

```

# SysV init.d
#
# When executing the init script, this user will be used to run the elasticsearch_
↳service.
# The default value is 'elasticsearch' and is declared in the init.d file.
# Note that this setting is only used by the init script. If changed, make sure that
# the configured user can read and write into the data, work, plugins and log_
↳directories.
# For systemd service, the user is usually configured in file /usr/lib/systemd/system/
↳elasticsearch.service

# Note: useless for VITAM, as the startup is managed by systemd
ES_USER={{ vitam_defaults.users.vitamdb }}
ES_GROUP={{ vitam_defaults.users.group }}

# The number of seconds to wait before checking if Elasticsearch started successfully_
↳as a daemon process
ES_STARTUP_SLEEP_TIME=5

#####
# System properties
#####

# Specifies the maximum file descriptor number that can be opened by this process
# When using Systemd, this setting is ignored and the LimitNOFILE defined in
# /usr/lib/systemd/system/elasticsearch.service takes precedence
#MAX_OPEN_FILES=65536

# The maximum number of bytes of memory that may be locked into RAM
# Set to "unlimited" if you use the 'bootstrap.memory_lock: true' option
# in elasticsearch.yml (ES_HEAP_SIZE must also be set).
# When using Systemd, the LimitMEMLOCK property must be set
# in /usr/lib/systemd/system/elasticsearch.service
#MAX_LOCKED_MEMORY=unlimited

# Maximum number of VMA (Virtual Memory Areas) a process can own
# When using Systemd, this setting is ignored and the 'vm.max_map_count'
# property is set at boot time in /usr/lib/sysctl.d/elasticsearch.conf
#MAX_MAP_COUNT=262144

```

#### 6.3.6.2 Fichier /usr/lib/tmpfiles.d/elasticsearch-data.conf

```

d    /var/run/{{ composant.cluster_name }}    0755 {{ vitam_defaults.users.vitamdb }} {
↳{ vitam_defaults.users.group }} - -

```

#### 6.3.6.3 Opérations

- Démarrage du service

Les commandes suivantes sont à passer sur les différentes machines constituant le cluster Elasticsearch.

En tant qu'utilisateur root : `systemctl start vitam-elasticsearch-log`

- Arrêt du service

Les commandes suivantes sont à passer sur les différentes machines constituant le cluster Elasticsearch.

En tant qu'utilisateur root : `systemctl stop vitam-elasticsearch-log`

- Sauvegarde du service

Dans cette version du système, seule une sauvegarde à froid du service est supportée (par la sauvegarde des fichiers de données présents dans `/vitam/data`)

- Supervision du service

Contrôler le retour HTTP 200 sur l'URL `<protocole web https ou https>://<host>:<port>/`

- Exports

N/A

- gestion de la capacité

N/A

- Réouverture d'un index fermé

Les index sont fermés par action récurrente de Curator ; il est néanmoins possible de rouvrir un index fermé par la commande suivante :

```
curl -XPOST '<adresseIP>:<port>/<index_fermé>/_open'
```

Référence <sup>12</sup>

- actions récurrentes
- cas des batches

N/A

## 6.3.7 elasticsearch Vitam

### 6.3.7.1 Présentation

Elasticsearch-log est une instance de la base d'indexation elasticsearch stockant les informations relatives aux archives hébergées dans VITAM. Elle participe dans ce sens à l'indexation et la recherche des données contenues dans MongoDB.

### 6.3.7.2 Configuration / fichiers utiles

Se reporter au *DIN*, qui configure le cluster ElasticSearch.

Les fichiers de configuration sous `/vitam/conf/elasticsearch-data`.

#### 6.3.7.2.1 Fichier `logging.yml`

```
# you can override this using by setting a system property, for example -Des.logger.
↪ level=DEBUG
es.logger.level: INFO
rootLogger: ${es.logger.level}, console, file
logger:
  # log action execution errors for easier debugging
  action: DEBUG
```

12. <https://www.elastic.co/guide/en/elasticsearch/reference/2.4/indices-open-close.html>

```

# deprecation logging, turn to DEBUG to see them
deprecation: INFO, deprecation_log_file

# reduce the logging for aws, too much is logged under the default INFO
com.amazonaws: WARN
# aws will try to do some sketchy JMX stuff, but its not needed.
com.amazonaws.jmx.SdkMBeanRegistrySupport: ERROR
com.amazonaws.metrics.AwsSdkMetrics: ERROR

org.apache.http: INFO

# gateway
#gateway: DEBUG
#index.gateway: DEBUG

# peer shard recovery
#indices.recovery: DEBUG

# discovery
#discovery: TRACE

index.search.slowlog: TRACE, index_search_slow_log_file
index.indexing.slowlog: TRACE, index_indexing_slow_log_file

additivity:
  index.search.slowlog: false
  index.indexing.slowlog: false
  deprecation: false

appender:
  console:
    type: console
    layout:
      type: consolePattern
      conversionPattern: "[%d{ISO8601}][%-5p][%-25c] %m%n"

  file:
    type: dailyRollingFile
    file: ${path.logs}/${cluster.name}.log
    datePattern: "'.'yyyy-MM-dd"
    layout:
      type: pattern
      conversionPattern: "[%d{ISO8601}][%-5p][%-25c] %.10000m%n"

  syslog:
    type: org.apache.log4j.net.SyslogAppender
    syslogHost: localhost:514
    layout:
      type: pattern
      conversionPattern: "{ compositant.cluster_name }: [%d{ISO8601}][%-5p][%-25c] %.
↪10000m%n"

# Use the following log4j-extras RollingFileAppender to enable gzip compression of
↪log files.
# For more information see https://logging.apache.org/log4j/extras/apidocs/org/
↪apache/log4j/rolling/RollingFileAppender.html
#file:
#type: extrasRollingFile

```



```

#file: ${path.logs}/${cluster.name}.log
#rollingPolicy: timeBased
#rollingPolicy.FileNamePattern: ${path.logs}/${cluster.name}.log.%d{yyyy-MM-dd}.gz
#layout:
  #type: pattern
  #conversionPattern: "[%d{ISO8601}][%-5p][%-25c] %m%n"

deprecation_log_file:
  type: dailyRollingFile
  file: ${path.logs}/${cluster.name}_deprecation.log
  datePattern: "'. 'yyyy-MM-dd"
  layout:
    type: pattern
    conversionPattern: "[%d{ISO8601}][%-5p][%-25c] %m%n"

index_search_slow_log_file:
  type: dailyRollingFile
  file: ${path.logs}/${cluster.name}_index_search_slowlog.log
  datePattern: "'. 'yyyy-MM-dd"
  layout:
    type: pattern
    conversionPattern: "[%d{ISO8601}][%-5p][%-25c] %m%n"

index_indexing_slow_log_file:
  type: dailyRollingFile
  file: ${path.logs}/${cluster.name}_index_indexing_slowlog.log
  datePattern: "'. 'yyyy-MM-dd"
  layout:
    type: pattern
    conversionPattern: "[%d{ISO8601}][%-5p][%-25c] %m%n"

```

### 6.3.7.2.2 Fichier elasticsearch.yml

```

# ===== Elasticsearch Configuration =====
#
# NOTE: Elasticsearch comes with reasonable defaults for most settings.
#       Before you set out to tweak and tune the configuration, make sure you
#       understand what are you trying to accomplish and the consequences.
#
# The primary way of configuring a node is via this file. This template lists
# the most important settings you may want to configure for a production cluster.
#
# Please see the documentation for further information on configuration options:
# <http://www.elastic.co/guide/en/elasticsearch/reference/current/setup-configuration.
↪html>
#
# ----- Cluster -----
#
# Use a descriptive name for your cluster:
#
cluster.name: {{ composant.cluster_name }}
#
# ----- Node -----
#
# Use a descriptive name for the node:
#

```

```
node.name: {{inventory_hostname}}
#
# Add custom attributes to the node:
#
# node.rack: r1
#
# ----- Paths -----
#
# Path to directory where to store the data (separate multiple locations by comma):
#
path.data: {{ elasticsearch_data_dir }}
#
# Path to log files:
#
path.logs: {{ elasticsearch_log_dir }}
#
# ----- Memory -----
#
# Lock the memory on startup:
#
bootstrap.mlockall: true
#
# Make sure that the 'ES_HEAP_SIZE' environment variable is set to about half the
↪memory
# available on the system and that the owner of the process is allowed to use this
↪limit.
#
# Elasticsearch performs poorly when the system is swapping the memory.
#
# ----- Network -----
#
# Set the bind address to a specific IP (IPv4 or IPv6):
#
# Note : if installing to localhost, notably a docker container, we need to bind
↪larger than localhost
{% if inventory_hostname == "localhost" %}
network.host: 0.0.0.0
http.cors.enabled: true
http.cors.allow-origin: "*"
{% else %}
# KWA TODO: Check it again (ansible_hostname VS inventory_hostname VS ip_service) VS
↪0.0.0.0
network.host: 0.0.0.0
{% endif %}
#
# Set a custom port for HTTP:
#
http.port: {{ composant.port_http }}
network.port: {{ composant.port_tcp }}
transport.tcp.port: {{ composant.port_tcp }}
#
# For more information, see the documentation at:
# <http://www.elastic.co/guide/en/elasticsearch/reference/current/modules-network.
↪html>
#
# ----- Discovery -----
#
# Pass an initial list of hosts to perform discovery when new node is started:
```

```

# The default list of hosts is ["127.0.0.1", ":::1"]
#
discovery.zen.ping.unicast.hosts: [ {% for host in groups['hosts-elasticsearch-data']
↪%}"{{ hostvars[host]['ip_admin'] }}" {% if not loop.last %},{% endif %}{% endfor %} ]
#
# Prevent the "split brain" by configuring the majority of nodes (total number of ↪
↪nodes / 2 + 1):
#
# discovery.zen.minimum_master_nodes: 3
#
# For more information, see the documentation at:
# <http://www.elastic.co/guide/en/elasticsearch/reference/current/modules-discovery.
↪html>
#
# ----- Gateway -----
#
# Block initial recovery after a full cluster restart until N nodes are started:
#
# gateway.recover_after_nodes: 3
#
# For more information, see the documentation at:
# <http://www.elastic.co/guide/en/elasticsearch/reference/current/modules-gateway.
↪html>
#
# ----- Various -----
#
# Disable starting multiple nodes on a single system:
#
# node.max_local_storage_nodes: 1
#
# Require explicit names when deleting indices:
#
# action.destructive_requires_name: true
{% if groups['hosts-elasticsearch-data']|length == 1 %}
index.number_of_replicas: 0
{% else %}
index.number_of_replicas: 2
{% endif %}#

index.refresh_interval: 10s
indices.memory.index_buffer_size: 30%
index.translog.flush_threshold_ops: 50000
refresh_interval_in_millis: 30000

# Threadpools configuration
threadpool:
  search:
    size: {{ ansible_processor_cores * ansible_processor_threads_per_core * 2 }}
    queue_size: 5000
  bulk:
    size: {{ ansible_processor_cores * ansible_processor_threads_per_core * 2 }}
    queue_size: 5000
  refresh:
    size: {{ ansible_processor_cores * ansible_processor_threads_per_core * 2 }}
    keep_alive: 2m

```

## 6.3.7.2.3 Fichier sysconfig/elasticsearch

```
#####
# Elasticsearch
#####

# Elasticsearch home directory
#ES_HOME=/usr/share/elasticsearch

# Elasticsearch configuration directory
CONF_DIR={{ vitam_defaults.folder.root_path }}/conf/{{ composant.cluster_name }}

# Elasticsearch data directory
#DATA_DIR={{ vitam_defaults.folder.root_path }}/data/{{ composant.cluster_name }}

# Elasticsearch logs directory
#LOG_DIR={{ vitam_defaults.folder.root_path }}/log/{{ composant.cluster_name }}

# Elasticsearch PID directory
#PID_DIR=/var/run/{{ composant.cluster_name }}

# Heap size defaults to 256m min, 1g max
# Set ES_HEAP_SIZE to 50% of available RAM, but no more than 31g
#ES_JAVA_OPTS=

# Heap new generation
#ES_HEAP_NEWSIZE=

# Maximum direct memory
#ES_DIRECT_SIZE=

# Additional Java OPTS
ES_JAVA_OPTS="-Xms{{ elasticsearch_memory }} -Xmx{{ elasticsearch_memory }} -XX:
↪+UseGCLogFileRotation -XX:NumberOfGCLogFiles=10 -XX:GCLogFileSize=10M -XX:
↪+PrintGCDetails -XX:+PrintGCApplicationStoppedTime"

# Configure restart on package upgrade (true, every other setting will lead to not_
↪restarting)
#RESTART_ON_UPGRADE=true

# Path to the GC log file
#ES_GC_LOG_FILE={{ vitam_defaults.folder.root_path }}/log/{{ composant.cluster_name }}
↪gc.log

#####
# Elasticsearch service
#####

# SysV init.d
#
# When executing the init script, this user will be used to run the elasticsearch_
↪service.
# The default value is 'elasticsearch' and is declared in the init.d file.
# Note that this setting is only used by the init script. If changed, make sure that
# the configured user can read and write into the data, work, plugins and log_
↪directories.
# For systemd service, the user is usually configured in file /usr/lib/systemd/system/
↪elasticsearch.service
```

```
# Note: useless for VITAM, as the startup is managed by systemd
ES_USER={{ vitam_defaults.users.vitamdb }}
ES_GROUP={{ vitam_defaults.users.group }}

# The number of seconds to wait before checking if Elasticsearch started successfully
↳as a daemon process
ES_STARTUP_SLEEP_TIME=5

#####
# System properties
#####

# Specifies the maximum file descriptor number that can be opened by this process
# When using Systemd, this setting is ignored and the LimitNOFILE defined in
# /usr/lib/systemd/system/elasticsearch.service takes precedence
#MAX_OPEN_FILES=65536

# The maximum number of bytes of memory that may be locked into RAM
# Set to "unlimited" if you use the 'bootstrap.memory_lock: true' option
# in elasticsearch.yml (ES_HEAP_SIZE must also be set).
# When using Systemd, the LimitMEMLOCK property must be set
# in /usr/lib/systemd/system/elasticsearch.service
#MAX_LOCKED_MEMORY=unlimited

# Maximum number of VMA (Virtual Memory Areas) a process can own
# When using Systemd, this setting is ignored and the 'vm.max_map_count'
# property is set at boot time in /usr/lib/sysctl.d/elasticsearch.conf
#MAX_MAP_COUNT=262144
```

#### 6.3.7.2.4 Fichier /usr/lib/tmpfiles.d/elasticsearch-data.conf

```
d    /var/run/{{ composant.cluster_name }}    0755 {{ vitam_defaults.users.vitamdb }} {
↳{{ vitam_defaults.users.group }} - -
```

#### 6.3.7.3 Opérations

- Démarrage du service

Les commandes suivantes sont à passer sur les différentes machines constituant le cluster Elasticsearch.

En tant qu'utilisateur root : `systemctl start vitam-elasticsearch-data`

- Arrêt du service

Les commandes suivantes sont à passer sur les différentes machines constituant le cluster Elasticsearch.

En tant qu'utilisateur root : `systemctl stop vitam-elasticsearch-data`

- Sauvegarde du service

Dans cette version du système, seule une sauvegarde à froid du service est supportée (par la sauvegarde des fichiers de données présents dans /vitam/data)

- Supervision du service

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:<port>/

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes
- cas des batches

N/A

## 6.3.8 functional administration

### 6.3.8.1 Présentation

Rôle :

- Gérer les référentiels métier de la plate-forme

Fonctions :

- Gestion du référentiel des formats (PRONOM)
- Gestion des règles de gestion des archives

### 6.3.8.2 Configuration / fichiers utiles

Les fichiers de configuration sont gérés par les procédures d'installation ou de mise à niveau de l'environnement *VITAM*. Se référer au *DIN*.

Les fichiers de configuration sont définis sous `/vitam/conf/functional-administration`.

#### 6.3.8.2.1 Fichier `functional-administration.conf`

ce fichier permet de définir l'URL d'accès au access server.

```
# Configuration MongoDB
mongoDbNodes:
{% for host in groups['mongos'] %}
- dbHost: {{hostvars[host]['ip_service']}}
  dbPort: {{ mongodb.mongos_port }}
{% endfor %}
dbName: masterdata
dbAuthentication: {{ mongodb.mongo_authentication }}
dbUserName: {{mongoFunctionalAdminUser}}
dbPassword: {{mongoFunctionalAdminPassword}}

jettyConfig: jetty-config.xml
tenants: [ "{{ vitam_tenant_ids | join(' ', '') }}" ]
workspaceUrl: {{vitam.workspace | client_url}}
processingUrl: {{vitam.processing | client_url}}

# ElasticSearch
clusterName: {{ vitam_struct.cluster_name }}
elasticsearchNodes:
{% for host in groups['hosts-elasticsearch-data'] %}
- hostName: {{hostvars[host]['ip_service']}}
```

```

    tcpPort: {{ elasticsearch.data.port_tcp }}
  {% endfor %}

# ExternalId configuration
listEnableExternalIdentifiers:
  0:
    - INGEST_CONTRACT
    - ACCESS_CONTRACT
  1:
    - INGEST_CONTRACT
    - ACCESS_CONTRACT
    - PROFILE
    - SECURITY_PROFILE

```

### 6.3.8.3 Opérations

- Démarrage du service

En tant qu'utilisateur root : `systemctl start vitam-functional-administration`

- Arrêt du service

En tant qu'utilisateur root : `systemctl stop vitam-functional-administration`

- Sauvegarde du service

Ce service ne nécessite pas de sauvegarde particulière.

- Supervision du service

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:<port>/functional-administration/v1/status

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:<port admin>/admin/v1/status

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

- cas des batches

N/A

## 6.3.9 ihm-demo

### 6.3.9.1 Présentation

Cette IHM a été développée pour des fins de tests de VITAM.

Rôle :

- Permettre une utilisation basique de VITAM, notamment sans SIA

Fonctions :

- Représentation des arborescences et des graphes

- Formulaires dynamiques
- Suivi des opérations
- Gestion des référentiels

### 6.3.9.2 Configuration / fichiers utiles

Les fichiers de configuration sont gérés par les procédures d'installation ou de mise à niveau de l'environnement *VITAM*. Se référer au *DIN*.

Les fichiers de configuration sont définis sous `/vitam/conf/ihm-demo`.

#### 6.3.9.2.1 Fichier `access-external-client.conf`

ce fichier permet de définir l'URL d'accès au access server.

```
serverHost: {{ vitam.accessexternal.host }}
serverPort: {{ vitam.accessexternal.port_https }}
secure: true
sslConfiguration :
  keystore :
    - keyPath: {{vitam_folder_conf}}/keystore_{{ vitam_struct.vitam_component }}.p12
      keyPassword: {{keystores.client_external.ihm_demo}}
  truststore :
    - keyPath: {{vitam_folder_conf}}/truststore_{{ vitam_struct.vitam_component }}.jks
      keyPassword: {{truststores.client_external}}
hostnameVerification: true
```

#### 6.3.9.2.2 Fichier `ihm-demo.conf`

```
serverHost: {{ ip_service }}
port: {{ vitam_struct.port_http }}

baseUrl: "/{{ vitam_struct.baseuri }}"
staticContent: {{ vitam_struct.static_content }}
staticContentV2: {{ vitam_struct.static_content_v2 }}
baseUri: /{{vitam_struct.baseuri}}
baseUriV2: /{{vitam_struct.baseuri_v2}}

jettyConfig: jetty-config.xml
tenants: [ "{{ vitam_tenant_ids | join('"', '"') }}" ]
secure: true
sipDirectory: {{vitam_folder_data}}
```

**tenants** : liste des tenants disponibles sur l'ihm-demo.

#### 6.3.9.2.3 Fichier `ingest-external-client.conf`

```
serverHost: {{ vitam.ingestexternal.host }}
serverPort: {{ vitam.ingestexternal.port_https }}
secure: true
sslConfiguration :
```



```

keystore :
- keyPath: {{vitam_folder_conf}}/keystore_{{ vitam_struct.vitam_component }}.p12
  keyPassword: {{keystores.client_external.ihm_demo}}
truststore :
- keyPath: {{vitam_folder_conf}}/truststore_{{ vitam_struct.vitam_component }}.jks
  keyPassword: {{truststores.client_external}}
hostnameVerification: true

```

#### 6.3.9.2.4 Fichier shiro.ini

```

# =====
# Shiro INI configuration
# =====

[main]
# Objects and their properties are defined here,
# Such as the securityManager, Realms and anything
# else needed to build the SecurityManager

# credentialsMatcher
sha256Matcher = org.apache.shiro.authc.credential.Sha256CredentialsMatcher
iniRealm.credentialsMatcher = $sha256Matcher

# Cache Manager
builtInCacheManager = org.apache.shiro.cache.MemoryConstrainedCacheManager

# Security Manager
securityManager.cacheManager = $builtInCacheManager

sessionManager = org.apache.shiro.web.session.mgt.DefaultWebSessionManager
securityManager.sessionMode=native
securityManager.sessionManager.globalSessionTimeout = 60000
securityManager.sessionManager = $sessionManager

# Notice how we didn't define the class for the FormAuthenticationFilter ('authc') -
↳ it is instantiated and available already:
authc.loginUrl = /#!/login

[users]
# The 'users' section is for simple deployments
# when you only need a small number of statically-defined
# set of User accounts.
#username = password

{% for item in vitam_users %}
{{item.login}}={{item.password|hash('sha256')}} , {{item.role}}
{% endfor %}

[roles]
admin = *
user = messages:*, archivesearch:*, logbook:*, ingest:*, archiveupdate:*, archiveunit:
↳ *, ingests:read, admin:formats:read, admin:rules:read, admin:accession-register:
↳ read, logbookunitlifecycles:*, logbookobjectslifecycles:*, clear:delete, check:read,
↳ traceability:content:read, accesscontracts:read, profiles:read, contracts:read,
↳ contexts:read
guest = archivesearch:*, archiveunit:*, units:*, unit:*, admin:accession-register:
↳ read, accesscontracts:read

```

```
[urls]
# make sure the end-user is authenticated.  If not, redirect to the 'authc.loginUrl'
↪above,
# and after successful authentication, redirect them back to the original account
↪page they
# were trying to view:
/v1/api/login = anon
/v1/api/logout = logout
/v1/api/messages/logbook = anon
/v1/api/tenants = anon
/** = authc
```

### 6.3.9.3 Opérations

- Démarrage du service

En tant qu'utilisateur root : `systemctl start vitam-ihm-demo`

- Arrêt du service

En tant qu'utilisateur root : `systemctl stop vitam-ihm-demo`

- Sauvegarde du service

Ce service ne nécessite pas de sauvegarde particulière.

- Supervision du service

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:<port>/ihm-demo/v1/status

- Gestion des utilisateurs

Les utilisateurs sont actuellement gérés via le fichier `shiro.ini`, dans la section `[users]`.

- Créer un utilisateur

Lancer la commande shell suivante pour générer le mot de passe :

```
echo -n <motdepasse> | sha256sum
```

Copier le résultat.

Ensuite, éditer le fichier `/vitam/conf/ihm-demo/shiro.ini` et ajouter, dans la section `[users]`, la ligne suivante :

```
<login de l'utilisateur>=<résultat de la commande de génération de mot de
↪passe précédente>
```

Pour terminer, relancer le service `vitam-ihm-demo` par la commande :

```
systemctl restart vitam-ihm-demo
```

- Supprimer un utilisateur

Dans la section `[users]`, enlever la ligne correspondant à l'utilisateur à supprimer. Pour terminer, relancer le service `vitam-ihm-demo` par la commande :

```
systemctl restart vitam-ihm-demo
```

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes
- cas des batches

N/A

## 6.3.10 ihm-recette

### 6.3.10.1 Présentation

Cette IHM a été développée pour des fins de validation de VITAM. Elle permet de réaliser des tests de non-régression, mais également des actions sur le contenu des bases de données.

**Danger :** Cette IHM ne doit PAS être déployée dans un environnement de production !

### 6.3.10.2 Configuration / fichiers utiles

Les fichiers de configuration sont gérés par les procédures d'installation ou de mise à niveau de l'environnement *VITAM*. Se référer au *DIN*.

Les fichiers de configuration sont définis sous `/vitam/conf/ihm-recette`.

#### 6.3.10.2.1 Fichier `access-external-client.conf`

ce fichier permet de définir l'URL d'accès au access server.

```
serverHost: {{ vitam.accessexternal.host }}
serverPort: {{ vitam.accessexternal.port_https }}
secure: true
sslConfiguration :
  keystore :
    - keyPath: {{vitam_folder_conf}}/keystore_{{ vitam_struct.vitam_component }}.p12
      keyPassword: {{keystores.client_external.ihm_recette}}
  truststore :
    - keyPath: {{vitam_folder_conf}}/truststore_{{ vitam_struct.vitam_component }}.jks
      keyPassword: {{truststores.client_external}}
hostnameVerification: false
```

#### 6.3.10.2.2 Fichier `ihm-recette.conf`

```
serverHost: {{ ip_service }}
port: {{ vitam_struct.port_https }}

baseUrl: "/{{ vitam_struct.baseuri }}"
staticContent: "{{ vitam_struct.static_content }}"

jettyConfig: jetty-config.xml
tenants: [ "{{ vitam_tenant_ids | join(' ', ' ') }}" ]
```

```
secure: true
secureMode: {{vitam_struct.secure_mode}}
sipDirectory: {{ vitam_folder_data }}/test-data
performanceReportDirectory: {{ vitam_folder_data }}/report/performance

testSystemSipDirectory: {{ vitam_folder_data }}/test-data/system
testSystemReportDirectory: {{ vitam_folder_data }}/report/system

# Configuration MongoDB
mongoDbNodes:
{% for server in groups['mongos'] %}
- dbHost: {{hostvars[server]['ip_service']}}
  dbPort: {{ mongodb.mongos_port }}
{% endfor %}
# Actually need this field for compatibility
dbName: admin
# @integ: parametrize it !
masterdataDbName: masterdata
logbookDbName: logbook
metadataDbName: metadata
dbAuthentication: {{ mongodb.mongo_authentication }}
dbUserName: {{mongoAdminUser}}
dbPassword: {{mongoAdminPassword}}

# ElasticSearch
clusterName: {{ vitam_struct.cluster_name }}
elasticsearchNodes:
{% for server in groups['hosts-elasticsearch-data'] %}
- hostName: {{hostvars[server]['ip_service']}}
  tcpPort: {{ elasticsearch.data.port_tcp }}
{% endfor %}
```

#### 6.3.10.2.3 Fichier ihm-recette-client.conf

```
serverHost: {{ vitam_struct.host }}
serverPort: {{ vitam_struct.port_http }}
```

#### 6.3.10.2.4 Fichier ingest-external-client.conf

```
serverHost: {{ vitam.ingestexternal.host }}
serverPort: {{ vitam.ingestexternal.port_https }}
secure: true
sslConfiguration :
  keystore :
    - keyPath: {{vitam_folder_conf}}/keystore_{{ vitam_struct.vitam_component }}.p12
      keyPassword: {{ keystores.client_external.ihm_recette }}
  truststore :
    - keyPath: {{vitam_folder_conf}}/truststore_{{ vitam_struct.vitam_component }}.jks
      keyPassword: {{ truststores.client_external }}
hostnameVerification: false
```

### 6.3.10.2.5 Fichier functional-administration-client.conf

```
serverHost: {{ vitam.functional_administration.host }}
serverPort: {{ vitam.functional_administration.port_http }}
```

### 6.3.10.2.6 Fichier shiro.ini

```
[main]

{% if vitam_struct.secure_mode == 'x509' %}
x509 = fr.gouv.vitam.common.auth.web.filter.X509AuthenticationFilter

x509.useHeader = {{vitam_ssl_user_header}}

x509credentialsMatcher = fr.gouv.vitam.common.auth.core.authc.
↳X509CredentialsSha256Matcher

x509Realm = fr.gouv.vitam.common.auth.core.realm.X509KeystoreFileRealm
x509Realm.grantedKeyStoreName = {{vitam_folder_conf}}/grantedstore_ihm-recette.jks
x509Realm.grantedKeyStorePassphrase = {{password_grantedstore}}
x509Realm.trustedKeyStoreName = {{vitam_folder_conf}}/truststore_ihm-recette.jks
x509Realm.trustedKeyStorePassphrase = {{password_truststore}}
x509Realm.credentialsMatcher = $x509credentialsMatcher
securityManager.realm = $x509Realm
securityManager.subjectDAO.sessionStorageEvaluator.sessionStorageEnabled = false
[urls]
/v1/api/** = x509

{% else %}
# Objects and their properties are defined here,
# Such as the securityManager, Realms and anything
# else needed to build the SecurityManager
# credentialsMatcher
sha256Matcher = org.apache.shiro.authc.credential.Sha256CredentialsMatcher
iniRealm.credentialsMatcher = $sha256Matcher
# Cache Manager
builtInCacheManager = org.apache.shiro.cache.MemoryConstrainedCacheManager
# Security Manager
securityManager.cacheManager = $builtInCacheManager
sessionManager = org.apache.shiro.web.session.mgt.DefaultWebSessionManager
securityManager.sessionMode=native
securityManager.sessionManager.globalSessionTimeout = 60000
securityManager.sessionManager = $sessionManager
# Notice how we didn't define the class for the FormAuthenticationFilter ('authc') -
↳it is instantiated and available already:
authc.loginUrl = /#!/login
[users]
# The 'users' section is for simple deployments
# when you only need a small number of statically-defined
# set of User accounts.
#username = password
{% for item in vitam_users %}
{% if item.role == "admin" %}
{{item.login}}={{item.password|hash('sha256')}}
{% endif %}
{% endfor %}
```

```
[roles]
# The 'roles' section is for simple deployments
# when you only need a small number of statically-defined
# roles.
[urls]
# make sure the end-user is authenticated.  If not, redirect to the 'authc.loginUrl'
↪above,
# and after successful authentication, redirect them back to the original account
↪page they
# were trying to view:
/v1/api/login = anon
/v1/api/logout = logout
/v1/api/securemode = anon
/** = authc

{% endif %}
```

#### 6.3.10.2.7 Fichier `storage-client.conf`

```
serverHost: {{ vitam.storageengine.host }}
serverPort: {{ vitam.storageengine.port_http }}
```

#### 6.3.10.2.8 Fichier `storage-offer.conf`

```
strategy_name=[{% for item in groups['hosts-storage-offer-default'] %}"{{ item }}"{%
↪if not loop.last %},{% endif %}{% endfor %}]
```

#### 6.3.10.2.9 Fichier `tnr.conf`

```
urlWorkspace: {{vitam.workspace | client_url}}
tenantsTest: [ "0" ]
```

### 6.3.10.3 Opérations

- Démarrage du service

En tant qu'utilisateur root : `systemctl start vitam-ihm-recette`

- Arrêt du service

En tant qu'utilisateur root : `systemctl stop vitam-ihm-recette`

- Sauvegarde du service

Ce service ne nécessite pas de sauvegarde particulière.

- Supervision du service

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:<port admin>/admin/v1/status

- Gestion des utilisateurs

Les utilisateurs sont actuellement gérés via le fichier `shiro.ini`, dans la section `[users]`.

- Créer un utilisateur

Lancer la commande shell suivante pour générer le mot de passe :

```
echo -n <motdepasse> | sha256sum
```

Copier le résultat.

Ensuite, éditer le fichier `/vitam/conf/ihm-recette/shiro.ini` et ajouter, dans la section `[users]`, la ligne suivante :

```
<login de l'utilisateur>=<résultat de la commande de génération de mot de_
↳passe précédente>
```

Pour terminer, relancer le service `vitam-ihm-recette` par la commande :

```
systemctl restart vitam-ihm-recette
```

- Supprimer un utilisateur

Dans la section `[users]`, enlever la ligne correspondant à l'utilisateur à supprimer. Pour terminer, relancer le service `vitam-ihm-recette` par la commande :

```
systemctl restart vitam-ihm-recette
```

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes
- cas des batches

N/A

## 6.3.11 ingest-external

### 6.3.11.1 Présentation

Ingest-external est le composant d'interface entre *VITAM* et un *SIA* client, permettant de réaliser des entrées d'archives dans *VITAM*.

Rôle :

- Exposer les API publiques du système
- Sécuriser l'accès aux API de VITAM

### 6.3.11.2 Configuration / fichiers utiles

Les fichiers de configuration sont gérés par les procédures d'installation ou de mise à niveau de l'environnement *VITAM*. Se référer au *DIN*.

Les fichiers de configuration sont définis sous `/vitam/conf/ingest-external`.

#### 6.3.11.2.1 Fichier ingest-external.conf

```
path: {{vitam_folder_data}}
jettyConfig: jetty-config.xml
tenants: [ "{{ vitam_tenant_ids | join(' ', '') }}" ]
authentication: true
tenantFilter : true
antiVirusScriptName: scan-{{ vitam_struct.antivirus }}.sh
timeoutScanDelay: 60000
```

Ce fichier contient un appel au shell d'antivirus (par défaut, ClamAV) ; se reporter au *DIN*.

#### 6.3.11.2.2 Fichier ingest-internal-client.conf

```
serverHost: {{ vitam_ingestinternal.host }}
serverPort: {{ vitam_ingestinternal.port_http }}
```

#### 6.3.11.3 Opérations

- Démarrage du service

En tant qu'utilisateur root : `systemctl start vitam-ingest-external`

- Arrêt du service

En tant qu'utilisateur root : `systemctl stop vitam-ingest-external`

- Sauvegarde du service

Ce service ne nécessite pas de sauvegarde particulière.

- Supervision du service

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:<port>/ingest-external/v1/status

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:<port admin>/admin/v1/status

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

- cas des batches

N/A



## 6.3.12 ingest-internal

### 6.3.12.1 Présentation

Rôle :

- Permettre l'entrée d'une archive SEDA dans le SAE

Fonctions :

- Upload HTTP de fichiers au format SEDA
- Sas de validation antivirus des fichiers entrants
- Persistance du SEDA dans workspace
- Lancement des workflows de traitements liés à l'entrée dans processing

### 6.3.12.2 Configuration / fichiers utiles

Les fichiers de configuration sont gérés par les procédures d'installation ou de mise à niveau de l'environnement *VITAM*. Se référer au *DIN*.

Les fichiers de configuration sont définis sous `/vitam/conf/ingest-internal`.

#### 6.3.12.2.1 Fichier `ingest-internal.conf`

```
workspaceUrl: {{vitam.workspace | client_url}}
processingUrl: {{vitam.processing | client_url}}
jettyConfig: jetty-config.xml
tenants: [ "{{ vitam_tenant_ids | join(' ', ' ') }}" ]
```

### 6.3.12.3 Opérations

- Démarrage du service

En tant qu'utilisateur root : `systemctl start vitam-ingest-internal`

- Arrêt du service

En tant qu'utilisateur root : `systemctl stop vitam-ingest-internal`

- Sauvegarde du service

Ce service ne nécessite pas de sauvegarde particulière.

- Supervision du service

Contrôler le retour HTTP 200 sur l'URL `<protocole web https ou https>://<host>:<port>/ingest-internal/v1/status`

Contrôler le retour HTTP 200 sur l'URL `<protocole web https ou https>://<host>:<port admin>/admin/v1/status`

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

- cas des batches

N/A

## 6.3.13 log server

### 6.3.13.1 Présentation

Ce composant représente en réalité l'ensemble des 3 composants suivants :

- Kibana, pour la présentation des dashboards de logs et de métriques ;
- Logstash, pour l'analyse et la centralisation des logs ;
- Curator, pour la maintenance des index elasticsearch de log.

### 6.3.13.2 Configuration / fichiers utiles

L'ansibleerie se charge du paramétrage de ces composants.

### 6.3.13.3 Opérations

- Démarrage du service

En tant qu'utilisateur root :

Pré-requis : le cluster elasticsearch associé est déjà démarré.

```
systemctl start logstash
systemctl start kibana
```

- Arrêt du service

En tant qu'utilisateur root :

```
systemctl stop kibana
systemctl stop logstash
```

Post-requis : le cluster elasticsearch associé est arrêté.

- Sauvegarde du service

Ce service ne nécessite pas de sauvegarde particulière.

- Supervision du service

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:<port>/app/kibana

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

batch Curator, actuellement purgeant les données de plus de XX jours (selon ce qui a été défini dans l'inventaire de ansible) dans Elasticsearch de logs.

- cas des batches

Curator

## 6.3.14 logbook

### 6.3.14.1 Présentation

Rôle :

- Gérer les journaux métiers à fort besoin d'intégrité et potentiellement à valeur probante : journal du cycle de vie, journal métier (SAE/opérations + écritures)

Fonctions :

- Appel uniquement à partir de l'application

### 6.3.14.2 Configuration / fichiers utiles

Les fichiers de configuration sont gérés par les procédures d'installation ou de mise à niveau de l'environnement *VITAM*. Se référer au *DIN*.

Les fichiers de configuration sont définis sous `/vitam/conf/logbook`.

#### 6.3.14.2.1 Fichier `logbook.conf`

```
# Configuration MongoDB
mongoDbNodes:
{% for server in groups['mongos'] %}
- dbHost: {{hostvars[server]['ip_service']}}
  dbPort: {{ mongodb.mongos_port }}
{% endfor %}
dbName: logbook
dbAuthentication: {{ mongodb.mongo_authentication }}
dbUserName: {{mongoLogbookUser}}
dbPassword: {{mongoLogbookPassword}}
jettyConfig: jetty-config.xml
tenants: [ "{{{ vitam_tenant_ids | join(' ', '') }}" ]
p12LogbookPassword: {{keystores.timestamping.secure_logbook}}
p12LogbookFile: keystore_secure-logbook.p12
workspaceUrl: {{ vitam.workspace | client_url }}
processingUrl: {{ vitam.processing | client_url }}

# ElasticSearch
clusterName: {{ vitam_struct.cluster_name }}
elasticsearchNodes:
{% for server in groups['hosts-elasticsearch-data'] %}
- hostName: {{hostvars[server]['ip_service']}}
  tcpPort: {{ elasticsearch.data.port_tcp }}
{% endfor %}
```

### 6.3.14.3 Opérations

- Démarrage du service

En tant qu'utilisateur root : `systemctl start vitam-logbook`

- Arrêt du service

En tant qu'utilisateur root : `systemctl stop vitam-logbook`

- Sauvegarde du service

Ce service ne nécessite pas de sauvegarde particulière.

- Supervision du service

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:<port>/logbook/v1/status

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:<port admin>/admin/v1/status

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

- cas des batches

N/A

## 6.3.15 metadata

### 6.3.15.1 Présentation

Rôle :

- Stocker de manière requêtable et rapide les métadonnées des objets (également stockées mais de manière pérenne dans l'offre de stockage)

Fonctions :

- Fournit une API agréant une technologie de base de données et un moteur d'indexation
- Fournit un cache des requêtes pour optimisation

### 6.3.15.2 Configuration / fichiers utiles

#### 6.3.15.2.1 Fichier metadata.conf

```
# Configuration MongoDB
mongoDbNodes:
{% for server in groups['mongos'] %}
- dbHost: {{hostvars[server]['ip_service']}}
  dbPort: {{ mongodb.mongos_port }}
{% endfor %}
dbName: metadata
dbAuthentication: {{ mongodb.mongo_authentication }}
dbUserName: {{mongoMetadataUser}}
dbPassword: {{mongoMetadataPassword}}

jettyConfig: jetty-config.xml
tenants: [ "{{ vitam_tenant_ids | join(' ', ' ') }}" ]

# Elasticsearch
clusterName: {{ vitam_struct.cluster_name }}
```

```

elasticsearchNodes:
{% for server in groups['hosts-elasticsearch-data'] %}
- hostName: {{hostvars[server]['ip_service']}}
  tcpPort: {{ elasticsearch.data.port_tcp }}
{% endfor %}

```

### 6.3.15.3 Opérations

- Démarrage du service

En tant qu'utilisateur root : `systemctl start vitam-metadata`

- Arrêt du service

En tant qu'utilisateur root : `systemctl stop vitam-metadata`

- Sauvegarde du service

Ce service ne nécessite pas de sauvegarde particulière.

- Supervision du service

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:<port>/metadata/v1/status

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:<port admin>/admin/v1/status

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

- cas des batches

N/A

## 6.3.16 mongoC

### 6.3.16.1 Présentation

Replicaset mongoDB servant à stocker la configuration MongoDB (clés de sharding, shards, ...) lors de l'utilisation de MondoDB en mode sharding.

### 6.3.16.2 Configuration / fichiers utiles

Pour le moment, aucun fichier à paramétrer.

### 6.3.16.3 Opérations

- Démarrage du service

En tant qu'utilisateur root : `systemctl start vitam-mongoc`

- Arrêt du service

En tant qu'utilisateur root : `systemctl stop vitam-mongoc`

- Sauvegarde du service

Il est recommandé d'effectuer des sauvegardes régulières des données.

Pour cela, la procédure à suivre est :

1. Arrêt du service
2. Lancement d'un backup (à définir)
3. Démarrage du service

- Supervision du service
- Exports
- gestion de la capacité

N/A

- actions récurrentes
- cas des batches

Cas de l'export tous les soirs/matins ?

## 6.3.17 mongoD

### 6.3.17.1 Présentation

Replicaset MongoDB stockant les données métier de Vitam.

### 6.3.17.2 Configuration / fichiers utiles

Pour le moment, aucun fichier à paramétrer.

### 6.3.17.3 Opérations

- Démarrage du service

En tant qu'utilisateur root : `systemctl start vitam-mongod`

- Arrêt du service

En tant qu'utilisateur root : `systemctl stop vitam-mongod`

- Sauvegarde du service

Il est recommandé d'effectuer des sauvegardes régulières des données.

Pour cela, la procédure à suivre est :

1. Arrêt du service

2. Lancement d'un backup (à définir)
  3. Démarrage du service
- Supervision du service

Via mongo-express ?

- Exports
- gestion de la capacité

N/A

- actions récurrentes
- cas des batches

Cas de l'export tous les soirs/matins ?

## 6.3.18 mongoS

### 6.3.18.1 Présentation

Point d'accès frontal à la base de données MongoDB de Vitam. Redirige sur le bon shard en fonction de la clé de sharding positionnée sur la collection.

### 6.3.18.2 Configuration / fichiers utiles

Pour le moment, aucun fichier à paramétrer.

### 6.3.18.3 Opérations

- Démarrage du service

En tant qu'utilisateur root : `systemctl start vitam-mongos`

- Arrêt du service

En tant qu'utilisateur root : `systemctl stop vitam-mongos`

- Sauvegarde du service

Il est recommandé d'effectuer des sauvegardes régulières des données.

Pour cela, la procédure à suivre est :

1. Arrêt du service
  2. Lancement d'un backup (à définir)
  3. Démarrage du service
- Supervision du service

Via mongo-express ?

- Exports
- gestion de la capacité

N/A

- actions récurrentes
- cas des batches

Cas de l'export tous les soirs/matins ?

## 6.3.19 processing

### 6.3.19.1 Présentation

Rôle :

- Exécution massive de processus métiers complexes
- Utilisé notamment lors du versement et de la préservation

Fonctions :

- Découpage en micro tâches de processus métier (en fonction d'un référentiel)
- Supervision de l'état d'exécution de chaque « job »
- Reprise sur incident
- Traçabilité de l'ensemble des actions effectuées

### 6.3.19.2 Configuration / fichiers utiles

Les fichiers de configuration sont gérés par les procédures d'installation ou de mise à niveau de l'environnement *VITAM*. Se référer au *DIN*.

Les fichiers de configuration sont définis sous `/vitam/conf/processing`.

#### 6.3.19.2.1 Fichier `processing.conf`

```
urlMetadata: {{ vitam_struct | client_url }}
urlWorkspace: {{ vitam.workspace | client_url }}
jettyConfig: jetty-config.xml
tenants: [ "{{ vitam_tenant_ids | join(' ', '') }}" ]
workflowRefreshPeriod: 1
processingCleanerPeriod: 1
```

#### 6.3.19.2.2 Fichier `version.conf`

```
binaryDataObjectVersions:
- BinaryMaster
- Dissemination
- Thumbnail
- TextContent
physicalDataObjectVersions:
- PhysicalMaster
- Dissemination
```



### 6.3.19.2.3 Fichier `storage-client.conf`

```
serverHost: {{ vitam.storageengine.host }}
serverPort: {{ vitam.storageengine.port_http }}
```

### 6.3.19.3 Opérations

- Démarrage du service

En tant qu'utilisateur `root` : `systemctl start vitam-processing`

- Arrêt du service

En tant qu'utilisateur `root` : `systemctl stop vitam-processing`

- Sauvegarde du service

Ce service ne nécessite pas de sauvegarde particulière.

- Supervision du service

Contrôler le retour HTTP 200 sur l'URL `<protocole web https ou https>://<host>:<port>/processing/v1/status`

Contrôler le retour HTTP 200 sur l'URL `<protocole web https ou https>://<host>:<port admin>/admin/v1/status`

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

- cas des batches

N/A

## 6.3.20 Security internal

### 6.3.20.1 Présentation

Rôle :

- Service d'authentification par certificats

Fonctions :

- Authentification par certificats
- Authentification personae

### 6.3.20.2 Configuration / fichiers utiles

Les fichiers de configuration sont gérés par les procédures d'installation ou de mise à niveau de l'environnement *VITAM*. Se référer au *DIN*.

Les fichiers de configuration sont définis sous `/vitam/conf/security-internal`.

### 6.3.20.2.1 Fichier `security-internal.conf`

ce fichier permet de définir l'URL d'accès au access server.

```
# Configuration MongoDB
mongoDbNodes:
{% for host in groups['mongos'] %}
- dbHost: {{hostvars[host]['ip_service']}}
  dbPort: {{ mongodb.mongos_port }}
{% endfor %}
dbName: identity
dbAuthentication: {{ mongodb.mongo_authentication }}
dbUserName: {{mongoSecurityInternalUser}}
dbPassword: {{mongoSecurityInternalPassword}}

jettyConfig: jetty-config.xml
tenants: [ "{{{ vitam_tenant_ids | join(' ', '') }}" ]
```

### 6.3.20.3 Opérations

- Démarrage du service

En tant qu'utilisateur root : `systemctl start vitam-security-internal`

- Arrêt du service

En tant qu'utilisateur root : `systemctl stop vitam-security-internal`

- Sauvegarde du service

Ce service ne nécessite pas de sauvegarde particulière.

- Supervision du service

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:<port>/vitam-security-internal/v1/status

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:<port> admin>/admin/v1/status

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

- cas des batches

N/A

## 6.3.21 siegfried

### 6.3.21.1 Présentation

Siegfried est un outil permettant la détection de format d'un fichier.

### 6.3.21.2 Configuration / fichiers utiles

Les fichiers de configuration sont gérés par les procédures d'installation ou de mise à niveau de l'environnement *VITAM*. Se référer au *DIN*.

### 6.3.21.3 Opérations

- Démarrage du service

En tant qu'utilisateur root : `systemctl start vitam-siegfried`

- Arrêt du service

En tant qu'utilisateur root : `systemctl stop vitam-siegfried`

**Avertissement :** ne pas oublier que cela peut perturber le comportement de certains composants Vitam (ingest-external et worker).

- Sauvegarde du service

Ce service ne nécessite pas de sauvegarde particulière.

- Logs

Les logs applicatifs sont envoyés par rsyslog à la solution de centralisation des logs ; il est néanmoins possible d'en visionner une représentation par la commande :

```
journalctl --unit vitam-siegfried
```

- Supervision du service

N/A

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

- cas des batches

N/A

## 6.3.22 storage-engine

### 6.3.22.1 Présentation

Rôle :

- Stockage des données (Méta Données, Objets Numériques et journaux SAE et de l'archive)

Fonctions :

- Utilisation de stratégie de stockage (abstraction par rapport aux offres de stockage sous-jacentes)
- Gestion des différentes offres de stockage

### 6.3.22.2 Configuration / fichiers utiles

Les fichiers de configuration sont gérés par les procédures d'installation ou de mise à niveau de l'environnement *VITAM*. Se référer au *DIN*.

Les fichiers de configuration sont définis sous `/vitam/conf/storage-engine`.

#### 6.3.22.2.1 Fichier driver-location.conf

```
driverLocation: {{vitam_folder_lib}}
```

#### 6.3.22.2 Fichier driver-mapping.conf

```
driverMappingPath: {{vitam_folder_data}}/  
delimiter: ;
```

#### 6.3.22.2.3 Fichier static-offer.json

```
[
{% for item in groups['hosts-storage-offer-default'] %}
{
    "id" : "{{ item }}",
    {% if vitam.storageofferdefault.https_enabled==true %}
    "baseUrl" : "https://{{hostvars[item]['ip_service']}}:{{ vitam.
↪storageofferdefault.port_https }}",
    {% else %}
    "baseUrl" : "http://{{hostvars[item]['ip_service']}}:{{ vitam.storageofferdefault.
↪port_http }}",
    {% endif %}
    "parameters" : {
        {% if vitam.storageofferdefault.https_enabled==true %}
        "keyStore-keyPath": "{{vitam_folder_conf}}/keystore_storage.p12",
        "keyStore-keyPassword": "{{keystores.client_storage.storage}}",
        "trustStore-keyPath": "{{vitam_folder_conf}}/truststore_storage.jks",
        "trustStore-keyPassword": "{{truststores.client_storage}}"
        {% endif %}
    }
}
{% if not loop.last %},
{% endif %}
{% endfor %}
]
```

#### 6.3.22.2.4 Fichier static-strategy.json

```
{
  "id" : "default",
  "hot" : {
    "copy" : { { groups['hosts-storage-offer-default']|length } },
    "offers" : [
```

```

        {% for item in groups['hosts-storage-offer-default'] %}
        {"id" : "{{ item }}" }{% if not loop.last %},{% endif %}
        {% endfor %}
    ]
}
}

```

#### 6.3.22.2.5 Fichier `storage-engine.conf`

```

urlWorkspace: {{ vitam.workspace | client_url }}
timeoutMsPerKB: 100
jettyConfig: jetty-config.xml
tenants: [ "{{ vitam_tenant_ids | join(' ', ' ') }}" ]
zippingDirecorty: {{ vitam_folder_data }}/storage_archives
loggingDirectory: {{ vitam_folder_log }}

```

#### 6.3.22.3 Opérations

- Démarrage du service

En tant qu'utilisateur root : `systemctl start vitam-storage`

- Arrêt du service

En tant qu'utilisateur root : `systemctl stop vitam-storage`

- Sauvegarde du service

Ce service ne nécessite pas de sauvegarde particulière.

- Supervision du service

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:<port>/storage/v1/status

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:<port admin>/admin/v1/status

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

- cas des batches

N/A

### 6.3.23 offer

#### 6.3.23.1 Présentation

Ce composant est une déclinaison des offres de stockage sur FileSystem et CEPH.

Rôle :

- Fournir une offre de stockage par défaut permettant la persistance des objets sur un système de fichier local

Fonctions :

- Offre de stockage fournie par défaut
- Stockage simple des objets numériques sur un système de fichiers local

### 6.3.23.2 Configuration / fichiers utiles

Les fichiers de configuration sont gérés par les procédures d'installation ou de mise à niveau de l'environnement *VITAM*. Se référer au *DIN*.

Les fichiers de configuration sont définis sous `/vitam/conf/offer`.

#### 6.3.23.2.1 Fichier default-offer.conf

```
contextPath: /
# Smile : TODO : remove storagePath from this file
storagePath: {{vitam_folder_data}}
jettyConfig: jetty-config.xml
authentication : {{ vitam_struct.https_enabled }}
tenants: [ "{{ vitam_tenant_ids | join(' ', ' ') }}" ]
```

#### 6.3.23.2.2 Fichier default-storage.conf

```
provider: {{ vitam_offers[offer_conf]["provider"] }}
{% if vitam_offers[offer_conf]["provider"] != "openstack-swift" %}
storagePath: {{vitam_folder_data}}
{% endif %}
keystoneEndPoint: {{ vitam_offers[offer_conf]["keystone_auth_url"] | default("") }}
swiftUid: {{ vitam_offers[offer_conf]["swift_uid"] | default("") }}
swiftSubUser: {{ vitam_offers[offer_conf]["swift_subuser"] | default("") }}
credential: {{ vitam_offers[offer_conf]["swift_password"] | default("") }}
cephMode: {{ vitam_offers[offer_conf]["ceph_mode"] | default(true) }}
```

### 6.3.23.3 Opérations

- Démarrage du service

En tant qu'utilisateur root : `systemctl start vitam-offer`

- Arrêt du service

En tant qu'utilisateur root : `systemctl stop vitam-offer`

- Sauvegarde du service

Ce service ne nécessite pas de sauvegarde particulière.

- Supervision du service

Contrôler le retour HTTP 200 sur l'URL `<protocole web https ou https>://<host>:<port>/offer/v1/status`

Contrôler le retour HTTP 200 sur l'URL `<protocole web https ou https>://<host>:<port admin>/admin/v1/status`

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes
- cas des batches

N/A

## 6.3.24 worker

### 6.3.24.1 Présentation

Ce composant permet de réaliser l'ensemble des traitements sur les archives.

Rôle :

- Fournir un moyen d'exécuter les traitements sur les archives, piloté par le composant processing.

### 6.3.24.2 Configuration / fichiers utiles

Les fichiers de configuration sont gérés par les procédures d'installation ou de mise à niveau de l'environnement *VITAM*. Se référer au *DIN*.

Les fichiers de configuration sont définis sous `/vitam/conf/worker`.

#### 6.3.24.2.1 Fichier `format-identifiers.conf`

Ce fichier permet de définir l'URL d'accès à Siegfried.

```
siegfried-local:
  type: SIEGFRIED
  client: http
  host: localhost
  port: {{ siegfried.port }}
  rootPath: {{ vitam_folder_tmp }}/
  versionPath: {{ vitam_folder_data }}/version/folder
```

#### 6.3.24.2.2 Fichier `functional-administration-client.conf.j2`

Ce fichier permet de définir l'accès à functional-administration.

```
serverHost: {{ vitam_functional_administration.host }}
serverPort: {{ vitam_functional_administration.port_http }}
```

#### 6.3.24.2.3 Fichier `metadata-client.conf`

Ce fichier permet de définir l'accès au metadata.

```
serverHost: {{ vitam.metadata.host }}
serverPort: {{ vitam.metadata.port_http }}
```

#### 6.3.24.2.4 Fichier `storage-client.conf`

Ce fichier permet de définir l'accès au storage.

```
serverHost: {{ vitam.storageengine.host }}
serverPort: {{ vitam.storageengine.port_http }}
```

#### 6.3.24.2.5 Fichier `version.conf`

```
binaryDataObjectVersions:
- BinaryMaster
- Dissemination
- Thumbnail
- TextContent
physicalDataObjectVersions:
- PhysicalMaster
- Dissemination
```

#### 6.3.24.2.6 Fichier `worker.conf`

Ce fichier permet de définir le paramétrage du composant worker.

```
# Configuration processing
# HERE MUST BE MY (WORKER) current configuration
registerServerHost: {{ ip_service }}
registerServerPort: {{ vitam_struct.port_http }}
# Configuration handler
processingUrl: {{vitam.processing | client_url}}
urlMetadata: {{vitam.metadata | client_url}}
urlWorkspace: {{vitam.workspace | client_url}}
# Configuration jetty
jettyConfig: jetty-config.xml
tenants: [ "{{ vitam_tenant_ids | join(' ', '') }}" ]
#Configuration parallele
capacity: {{vitam_worker_capacity}}
{% if vitam_worker_workerFamily is defined %}
workerFamily: {{vitam_worker_workerFamily}}
{% endif %}
```

Paramètres obligatoires :

- **processingUrl** : URL de connexion au composant Vitam processing
- **urlMetadata** : URL de connexion au composant VITAM metadata
- **urlWorkspace** : URL de connexion au composant VITAM workspace
- **registerServerHost** : host ou le worker déployé
- **registerServerPort** : port ou le worker déployé
- **jettyConfig** : le fichier config jetty associé au service du worker

Paramètres optionnels :



- **workerFamily** : la famille dont le worker appartient en fonction de tâche exécutée
- **capacity** : capacité du worker en mode parallèle de tâche (par défaut à 1 dans l'ansible, si non définie)

### 6.3.24.3 Opérations

- Démarrage du service

En tant qu'utilisateur root : `systemctl start vitam-workspace`

- Arrêt du service

En tant qu'utilisateur root : `systemctl stop vitam-workspace`

- Sauvegarde du service

Ce service ne nécessite pas de sauvegarde particulière.

- Supervision du service

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:<port>/workspace/v1/status

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:<port admin>/admin/v1/status

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

- cas des batches

N/A

## 6.3.25 workspace

### 6.3.25.1 Présentation

Rôle :

- Fourniture d'un espace pour l'échange de fichiers (et faire un appel par pointeur lors des appels entre composants) entre les différents composants de Vitam

Fonctions :

- Utilisation du moteur de stockage dans un mode minimal (Opérations CREATE, READ, DELETE sur 1 seule offre de stockage)

### 6.3.25.2 Configuration / fichiers utiles

Les fichiers de configuration sont gérés par les procédures d'installation ou de mise à niveau de l'environnement *VITAM*. Se référer au *DIN*.

Les fichiers de configuration sont définis sous `/vitam/conf/workspace`.

#### 6.3.25.2.1 Fichier `workspace.conf`

```
storagePath: {{vitam_folder_data}}
jettyConfig: jetty-config.xml
tenants: [ "{{ vitam_tenant_ids | join(' ', '') }}" ]
provider: filesystem
```

### 6.3.25.3 Opérations

- Démarrage du service

En tant qu'utilisateur root : `systemctl start vitam-workspace`

- Arrêt du service

En tant qu'utilisateur root : `systemctl stop vitam-workspace`

- Sauvegarde du service

Ce service ne nécessite pas de sauvegarde particulière.

- Supervision du service

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:<port>/workspace/v1/status

Contrôler le retour HTTP 200 sur l'URL <protocole web https ou https>://<host>:<port admin>/admin/v1/status

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

- cas des batches

N/A

---

## Intégration d'une application externe dans Vitam

---

### 7.1 Prérequis

L'application externe devra être en mesure de requêter les composants ingest-external & access-external sur leur port de service par protocole HTTPS. Il faut donc une ouverture de flux réseau pour le protocole TCP ou TLS, selon l'infrastructure en place, sur les ports de service de ces deux composants Vitam.

Il est nécessaire de créer un certificat TLS client pour l'application externe, le certificat public devra bien entendu être intégré dans le granted-store des composants ingest-external & access-external. La chaîne de certification complète de ce certificat devra aussi être ajoutée dans le truststore de ces composants.

### 7.2 Intégration du certificat client de l'application externe

Vitam fournit une arborescence de répertoire permettant d'ajouter automatiquement le certificat client ainsi que la chaîne de certification dans les bons keystores.

- Déposer les certificats de chaîne de certification dans *environments/certs/client-external/ca*
- Déposer le certificat client de l'application dans *environments/certs/client-external/clients/external*

<b>Avertissement :</b> Bien vérifier qu'aucun certificat ou CA non souhaité ne soit présent dans <i>environments/certs/*</i>
--

#### Voir aussi :

Ensuite, si ce n'est pas déjà fait, il reste à positionner les autres certificats, puis générer les keystores. Pour cela, se référer au [DIN](#), chapitre concernant la gestion des certificats.

### 7.3 Déploiement des keystores

#### 7.3.1 Vitam n'est pas encore déployé

Déployer Vitam en suivant la procédure indiquée dans le [DIN](#).

#### 7.3.2 Vitam est déjà déployé

Suivre la procédure de la section *Mise à jour des certificats* (page 8).



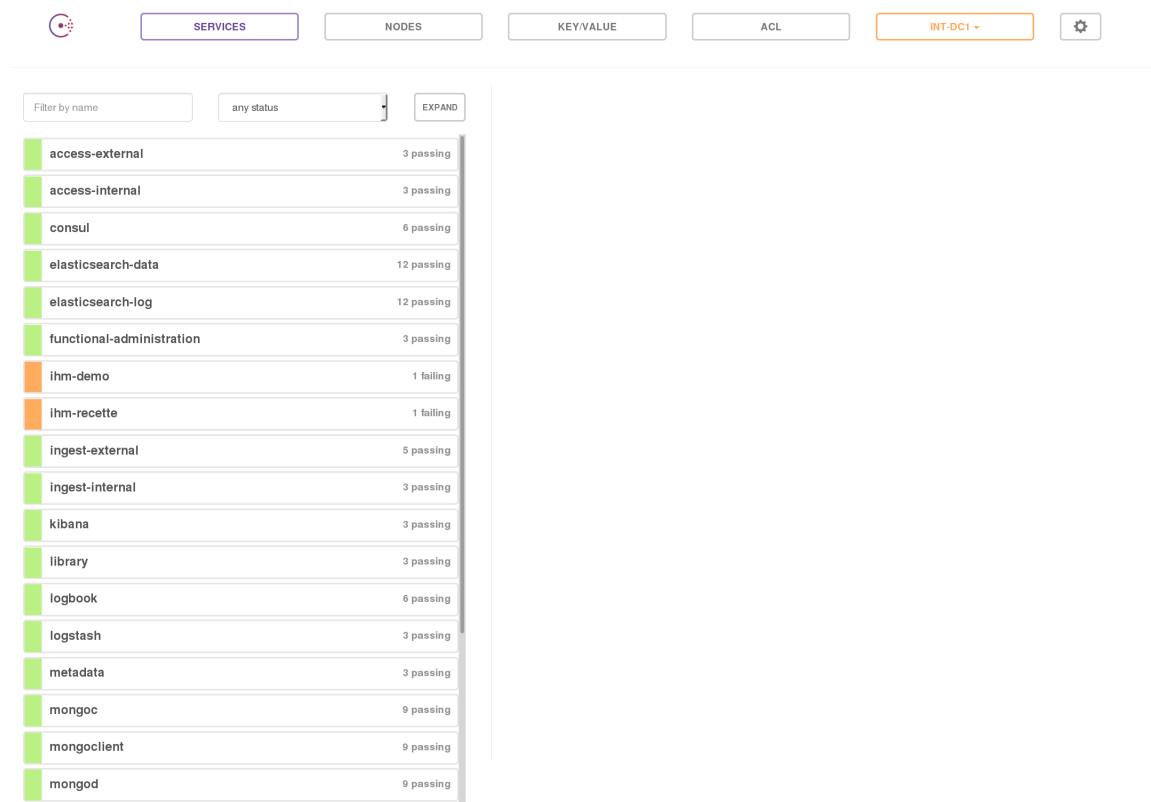
## Aide à l'exploitation



### 8.1 Analyse de premier niveau

Cette section a pour but de présenter les premiers outils à utiliser pour réaliser une analyse de premier niveau, en cas de problème avec la solution logicielle *VITAM*.

#### 8.1.1 Etat par Consul

Se connecter à l'IHM de Consul et recenser les états des composants de la solution logicielle *VITAM*.



	SERVICES	NODES	KEY/VALUE	ACL	INT-DC1	
Filter by name	any status	EXPAND				
access-external	3 passing					
access-internal	3 passing					
consul	6 passing					
elasticsearch-data	12 passing					
elasticsearch-log	12 passing					
functional-administration	3 passing					
ihm-demo	1 failing					
ihm-recette	1 failing					
ingest-external	5 passing					
ingest-internal	3 passing					
kibana	3 passing					
library	3 passing					
logbook	6 passing					
logstash	3 passing					
metadata	3 passing					
mongoc	9 passing					
mongoclient	9 passing					
mongod	9 passing					

A l'heure actuelle, tous les composants, excepté ihm-demo (couleur orange), doivent avoir un statut de couleur verte. Si ce n'est pas le cas :

1. seul un composant est KO, alors redémarrer le composant incriminé
2. si plusieurs services sont KO, suivre la procédure de redémarrage de VITAM
3. si tous les “check-DNS” (visible dans le détail des checks de chaque service) sont KO, s’assurer que, sur les machines hébergeant VITAM, le fichier `/etc/resolv.conf` contient, en début de fichier, la ligne :  
`nameserver 127.0.0.1.`

### **8.1.2 Etat par Kibana**

Se connecter à Kibana, aller dans “Dashboards”. Cliquer sur le bouton “Load Saved Dashboard” et sélectionner “Composants VITAM”. Eventuellement, changer la résolution (en haut à droite, par défaut, réglé sur les 15 dernières minutes).

Sur “pie-logback-error-level”, cliquer sur la section de camembert d’intérêt (ERROR) et regarder, en bas de page, les éventuelles erreurs remontées dans Kibana.

---

## Questions Fréquemment Posées

---

### 9.1 Présentation

Cette section a vocation à répertorier les différents problèmes rencontrés et apporter la solution la plus appropriée ; elle est amenée à être régulièrement mise à jour pour répertorier les problèmes rencontrés.

### 9.2 Retour d'expérience / cas rencontrés

**Mongo-express ne se connecte pas à la base de données associée** Si mongoDB a été redémarré, il faut également redémarrer mongo-express.

**Elasticsearch possède des shard non alloués (état “UNASSIGNED”)** Lors de la perte d'un noeud d'un cluster elasticseach, puis du retour de ce noeud, certains shards d'elasticseach peuvent rester dans l'état UNASSIGNED ; dans ce cas, cerebro affiche les shards correspondant en gris (au-dessus des noeuds) dans la vue “cluster”, et l'état du cluster passe en “yellow”. Il est possible d'avoir plus d'informations sur la cause du problème via une requête POST sur l'API `elasticsearch/_cluster/reroute?explain`. Si la cause de l'échec de l'assignation automatique a été résolue, il est possible de relancer les assignations automatiques en échec via une requête POST sur l'API `_cluster/reroute?retry_failed`. Dans le cas où l'assignation automatique ne fonctionne pas, il est nécessaire de faire l'assignation à la main pour chaque shard incriminé (requête POST sur `_cluster/reroute`):

```
{
  "commands": [
    {
      "allocate": {
        "index": "topbeat-2016.11.22",
        "shard": 3,
        "node": "vitam-iaas-dblog-01.int"
      }
    }
  ]
}
```

Cependant, un shard primaire ne peut être réalloué de cette manière (il y a risque de perte de données). Si le défaut d'allocation provient effectivement de la perte puis de la récupération d'un noeud, et que TOUS les noeuds du cluster sont de nouveaux opérationnels et dans le cluster, alors il est possible de forcer la réallocation sans perte.

```
{
  "commands": [
```

```
{
  "allocate": {
    "index": "topbeat-2016.11.22",
    "shard": 3,
    "node": "vitam-iaas-dblog-01.int",
    "allow_primary": "true"
  }
}
]
```

Sur tous ces sujets, Cf. la [documentation officielle](#) <sup>13</sup>.

**Elasticsearch possède des shards non initialisés (état “INITIALIZING”)** Tout d’abord, il peut être difficile d’identifier les shards en questions dans cerebro ; une requête HTTP GET sur l’API `_cat/shards` permet d’avoir une liste plus compréhensible. Un shard non initialisé correspond à un shard en cours de démarrage (Cf. [une ancienne page de documentation](#) <sup>14</sup>. Si les shards non initialisés sont présents sur un seul noeud, il peut être utile de redémarrer le noeud en cause. Sinon, une investigation plus poussée doit être menée.

---

13. <https://www.elastic.co/guide/en/elasticsearch/reference/current/cluster-reroute.html>

14. <https://www.elastic.co/guide/en/elasticsearch/reference/1.4/states.html>



---

## Exploitation par composant

---

Les sections qui suivent donnent une description plus fine pour l'exploitation des services VITAM.

### 10.1 Access

#### 10.1.1 Introduction

Cette documentation permet de paramétrer le port, le host ... à l'environnement de dev, d'intégration et de production pour le module access client et access-rest —————

```
#Configuration access-client serverHost : localhost serverPort : 8189
#Configuration access-rest urlMetaData : http://localhost:8088
#Configuration du serveur jetty jettyConfig : access-jetty.xml
```

### 10.2 Common

#### 10.2.1 Présentation

#### 10.2.2 Format Identifiers

Les services d'identification de formats peuvent être déployés sur tous les serveurs applicatifs vitam.

##### 10.2.2.1 Configuration des services d'identification des formats

Dans **/vitam/conf** du serveur applicatif où sont déployés les services d'identification de formats, il faut un fichier **format-identifiers.conf**. C'est un fichier YAML de configuration des services d'identification de format. Il possède les configurations des services que l'on souhaite déployer sur le serveur.

Le code suivant contient un exemple de toutes les configurations possibles :

```
siegfried-local:
type: SIEGFRIED
client: http
host: localhost
port: 55800
rootPath: /root/path
versionPath: /root/path/version/folder
```

```
createVersionPath: false
mock:
type: MOCK
```

- **Le service Mock :**
  - identifié par *mock*
  - *type* : le type de service déployé : *MOCK*
- **Le service Siegfried :**
  - identifié par *siegfried-local*
  - *type* : le type de service déployé : *SIEGFRIED*
  - *client* : type de client (pour le moment seul *http* existe).
  - *host* : le host du serveur siegfried déployé (devrait être le host du serveur courant)
  - *port* : le port du serveur siegfried déployé
  - *rootPath* : la racine sur laquelle le service Siegfried doit résoudre les fichiers à tester (ex : *"/data"*)
  - *versionPath* : le chemin vers un dossier vide pour renvoyer la version (Doit posséder des droits en lecture)
  - *createVersionPath* : Si *false* le dossier doit pré-exister sur le serveur sur lequel tourne Siegfried. Sinon, le client siegfried tente de créer automatiquement le dossier en local.

**NOTE : Chaque serveur est en charge de décrire la configuration nécessaire**

## 10.3 Functional administration

### 10.3.1 Présentation

## 10.4 Ingest

### 10.4.1 Introduction

Ce document présente les configs pour utiliser des modules différents de ingest. Deux configurations à voir pour ingest-external et ingest-internal

### 10.4.2 ingest-external-exploitation

Ce document spécifie la configurations (fichiers de config) pour lancer le services de ingest-external.

#### 1. Serveur ingest-external :

Pour lancer le serveur ingest-external, deux fichier config suivant sont nécessaires : - ingest-external.conf : préciser le répertoire temporaire ou les fichiers temporaires stockés et la configuration du serveur jetty. - jetty-config.xml : contenant le information pour lancer le serveur jetty de ingest-external. Ce fichier de jetty précise aussi la configuration TLS du mode SSL du serveur : les keystores et password pour les load, les algorithmes de chiffrement supportés ... - scan-clamav.sh : le script de scan pour détecter les virus qui sera appelé par les service ingest-external. -shiro.ini : la configuration de shiro permettant de filtrer des requêtes côté serveur. Il décrit différents paramètres : le marcher ; le filter et les keystores utilisé pour le filtre (truststore.jks et granted\_certs.jks) - le répertoire de keystore tls : ce répertoire contient différents fichiers de keystore permettant de lancer le serveur ingest-external en mode SSL.

- **truststore.jks** : contenant les CA qui attribue des certificats

- **keystore.jks** : contenant la clé privé et le certificat du serveur
- **granted\_certs.jks** : contenant les certificat des clients qui auront une authentification

#### 1. Client ingest-external :

Pour la création d'un lien ingest-external, nous avons besoin aussi le fichier de configuration ingest-external-client.conf qui précise le serveur host et la porte du serveur ou le client se connectent pour les requêtes.

Tous ces fichiers de configuration seront mis dans le répertoire /vitam/config. Les exemples de ces fichiers se trouvent dans les répertoires de src/test/resources correspondants.

### 10.4.3 ingest-internal-exploitation

Ce document spécifie la configurations (fichiers de config) pour lancer le services de ingest-internal.

1. Serveur ingest-internal : Pour lancer le serveur ingest-internal, deux fichier config suivant sont nécessaires - ingest-internal.conf : préciser les urls de services Processing & Workspace, et la configuration du serveur jetty. - jetty-config.xml : contenant le information pour lancer le serveur jetty de ingest-internal.

1.Client ingest-internal : Pour la création d'un lien ingest-internal, nous avons besoin aussi le fichier de configuration ingest-internal-client.conf qui précise le serveur host et la porte du serveur ou le client se connecte pour les requêtes.

Tous ces fichiers de configuration seront mis dans le répertoire /vitam/config. Les exemples de ces fichiers se trouvent dans les répertoires de src/test/resources correspondants.

## 10.5 Logbook

### 10.5.1 Présentation

## 10.6 Metadata

### 10.6.1 Présentation

## 10.7 Processing

### 10.7.1 Introduction

#### 10.7.1.1 But de cette documentation

Le but de cette documentation est d'expliquer la configuration et l'exploitation de ce module.

### 10.7.2 Processing

Nom de l'image docker : **processing**

Dans cette image est déployé le module processing

### 10.7.2.1 Configuration du worker

Dans `/vitam/conf` :

1. **processing.conf** : Fichier Yaml de configuration du server *processing*. Il possède une propriété :
  - **jettyConfig** : emplacement du fichier de configuration XML *jetty* (exemple `jetty-config.xml`)
  - **urlWorkspace** : URL d'accès au service distant *workspace* (exemple `http://localhost:8088`)
  - **urlMetadata** : URL d'accès au service distant *metadata* (exemple `http://localhost:8088`)
2. **logbook-client.conf** : Fichier de configuration du client qui communique avec le **logbook**. Il contient les propriétés suivantes :
  - **serverHost** : host distant du service logbook
  - **serverPort** : port distant du service logbook
3. **server-identity.conf** : identification du serveur
4. **logback.xml** : configuration des logs

### 10.7.2.2 Supervision du service

Contrôler le retour HTTP 200 et identité du serveur (cf *server-identity.conf*) sur l'URL `<protocole web https ou https>://<host>:<port>/processing/v1/status`

## 10.8 Storage

### 10.8.1 Introduction

#### 10.8.1.1 But de cette documentation

Le but de cette documentation est d'expliquer la configuration et l'exploitation des modules :

- **storage-engine**
- **storage-offer-default**

### 10.8.2 Storage Engine

Nom de l'image docker : **storage-engine**

Dans cette image sont déployés :

- le moteur de stockage (**storage-engine**)
- l'implémentation du driver correspondant à l'offre de stockage par défaut (**storage-offer-default**)

### 10.8.2.1 Configuration du moteur de stockage

Dans `/vitam/conf` :

1. **storage-engine.conf** : Fichier Yaml de configuration du server *storage-engine*. Il possède une propriété :
  - **urlWorkspace** : URL d'accès au service distant *workspace* (exemple <http://localhost:8088>)
2. **driver-location.conf** : Fichier Yaml de configuration du DriverManager, Il permet de définir l'emplacement où sont stockés les fichiers JAR contenant les implémentations des différents drivers pour les différentes offres. Il possède une seule propriété :
  - **driverLocation** : emplacement des jars (chemin absolu de préférence)
3. **driver-mapping.conf** : Fichier Yaml de configuration du DriverManager (persistance de l'association driver / offre). Pour le moment, ce fichier de configuration contient le chemin d'accès aux fichiers qui définissent le mapping driver<->offre, plus tard il évoluera sans doute pour prendre en compte des données en base et donc contenir la configuration d'accès à la base. Il contient deux propriétés :
  - **driverMappingPath** : Définit l'emplacement des fichiers de persistance (au jourd'hui on a 1 seul driver/offre, donc 1 seul fichier de persistance sera présent). La propriété doit finir par "/"
  - **delimiter** : Définit le "délimiteur" (CSV style) des fichiers.
4. **static-offer.json** : Contient la description de l'offre 'default' au format JSON (un jour sera sans doute dans une base de données). En PJ un exemple de ce fichier. La propriété `baseUrl` et `parameters` nécessitent d'être templaté. Et la propriété `parameters` doit contenir `keystore`, `trustore` et leur mot de passe que le storage driver va utiliser pour la vérification de l'authentification. Il s'agit de l'URL d'accès à l'offre de stockage 'default'. Exemple :

```
{
  "id" : "default",
  "baseUrl" : "https://localhost:8088",
  "parameters" : {
    "user" : "bob"
    "keyStore-keyPath": "src/test/resources/storage-test/tls/client/client.p12",
    "keyStore-keyPassword": "vitam2016",
    "trustStore-keyPath": "src/test/resources/storage-test/tls/server/truststore.jks",
    "trustStore-keyPassword": "tazerty"
  }
}
```

To remove TLS support :

- change "https" to "http" in `baseUrl`\*\*

```
{
  "id" : "default",
  "baseUrl" : "http://localhost:8088",
  "parameters" : {
    "user" : "bob"
  }
}
```

- change `storage-default-offer.json` to disable authentication

```
jettyConfig: jetty-config-nossl.xml
authentication : false
```

- change the `jetty-config-nossl.xml` of the offer (CAS Manager) to not include any TLS configuration
5. **static-strategy.json** : Contient les informations de la stratégie de stockage (1 seule pour le moment). Ce fichier n'est pas à modifier.

```
{
  "id" : "default",
  "hot" : {
    "copy" : 1,
    "offers" : [
      { "id" : "default" }
    ]
  }
}
```

6. **server-identity.conf** : identification du serveur
7. **logback.xml** : configuration des logs

### 10.8.2.2 Configuration du driver de l'offre de stockage par défaut

Dans `/vitam/data` :

1. **fr.gouv.vitam.storage.offers.workspace.driver.DriverImpl** : Il s'agit du fichier de persistance. Il contient l'identifiant de l'offre associée au driver (plus tard potentiellement DES offres associées) : `"default"`. Il DOIT être placé dans le répertoire défini dans le fichier `driver-mapping.conf`.

Dans `/vitam/lib` :

1. **storage-driver-default.jar** : Il s'agit d'un jar contenant l'implémentation du Driver vitam pour l'offre `"storage-offer-default"`. Ce jar DOIT être placé dans le dossier défini dans la propriété `driverLocation` du fichier `driver-location.conf`. Par défaut il est chargé en tant que dépendance du projet.

### 10.8.2.3 Supervision du service

Contrôler le retour HTTP 200 et identité du serveur (cf `server-identity.conf`) sur l'URL `<protocole web https ou https>://<host>:<port>/storage/v1/status`

## 10.8.3 Storage Offer Default

Nom de l'image docker : **storage-offer-default**

Dans cette image est déployée l'offre de stockage par défaut utilisant le workspace.

### 10.8.3.1 Configuration de l'offre de stockage

1. **default-storage.conf** : Fichier Yaml de configuration du service. Contient les propriétés suivantes :
  - **contextPath** : context path du server (mettre / par défaut)
  - **storagePath** : chemin sur le filesystem sur lequel sont stockés les objects (`/vitam/data`).
2. **server-identity.conf** : identification du serveur
3. **logback.xml** : configuration des logs

### 10.8.3.2 Supervision du service

Contrôler le retour HTTP 200 et identité du serveur (cf *server-identity.conf*) sur l'URL <protocole web https ou https>://<host>:<port>/offer/v1/status

## 10.9 Technical administration

### 10.9.1 Présentation

## 10.10 Worker

### 10.10.1 Introduction

#### 10.10.1.1 But de cette documentation

Le but de cette documentation est d'expliquer la configuration et l'exploitation de ce module :

- **worker**

#### 10.10.2 Storage Engine

Nom de l'image docker : **worker**

Dans cette image est déployé le module worker

##### 10.10.2.1 Configuration du worker

Dans */vitam/conf* :

1. **worker.conf** : Fichier Yaml de configuration du server *worker*. Il possède une propriété :
  - **jettyConfig** : emplacement du fichier de configuration XML *jetty* (exemple *jetty-config.xml*)
  - **registerServerHost** : le nom d'hôte du serveur courant auquel le client worker chargé par le processing va se connecter (Exemple : *localhost*)
  - **registerServerPort** : le port du serveur courant auquel le client worker chargé par le processing va se connecter (Exemple : 8082)
  - **processingUrl** : URL d'accès au service distant *processing* (exemple <http://localhost:8088>)
  - **urlWorkspace** : URL d'accès au service distant *workspace* (exemple <http://localhost:8088>)
  - **urlMetadata** : URL d'accès au service distant *metadata* (exemple <http://localhost:8088>)
2. **version.conf** : Fichier contenant la liste des version valides pour les SEDA. Il contient deux listes : une pour "binaryDataObjectVersions", une pour "physicalDataObjectVersions".
3. **logbook-client.conf** : Fichier de configuration du client qui communique avec le **logbook**. Il contient les propriétés suivantes :
  - **serverHost** : host distant du service logbook
  - **serverPort** : port distant du service logbook
4. **storage-client.conf** : Fichier de configuration du client qui communique avec le **storage-engine**. Il contient les propriétés suivantes :

- **serverHost** : host distant du service storage-engine
  - **serverPort** : port distant du service storage-engine
5. **server-identity.conf** : identification du serveur
  6. **logback.xml** : configuration des logs

#### 10.10.2.2 Supervision du service

Contrôler le retour HTTP 200 et identité du serveur (cf *server-identity.conf*) sur l'URL <protocole web https ou https>://<host>:<port>/worker/v1/status

## 10.11 Workspace

### 10.11.1 Présentation



---

**Annexes**

---



3.1	Vue d'ensemble d'un déploiement VITAM : zones, composants . . . . .	5
-----	---	---



2.1	Documents de référence VITAM . . . . .	3
4.1	Cinématique d'arrêt de VITAM . . . . .	9
4.2	Cinématique de démarrage de VITAM . . . . .	10



## A

API, [3](#)

## B

BDD, [3](#)

## C

COTS, [3](#)

## D

DAT, [3](#)

DEX, [3](#)

DIN, [3](#)

DNSSEC, [4](#)

DUA, [4](#)

## I

IHM, [4](#)

## J

JRE, [4](#)

JVM, [4](#)

## M

MitM, [4](#)

## N

NoSQL, [4](#)

## O

OAIS, [4](#)

## P

PDMA, [4](#)

PKI, [4](#)

## R

REST, [4](#)

RPM, [4](#)

## S

SAE, [4](#)

SEDA, [4](#)

SIA, [4](#)

## T

TNR, [4](#)

## V

VITAM, [4](#)