



VITAM - Documentation d'exploitation

Version 0.15.1

VITAM

avr. 26, 2017

1	Introduction	3
1.1	But de cette documentation	3
1.2	Destinataires de ce document	3
2	Rappels	5
2.1	Information concernant les licences	5
2.2	Documents de référence	5
2.2.1	Documents internes	5
2.2.2	Référentiels externes	5
2.3	Glossaire	6
3	Architecture de la solution logicielle VITAM	9
4	Exploitation globale	11
4.1	Gestion des accès	11
4.2	Portails d'administration	11
4.2.1	Technique	11
4.2.2	Fonctionnel	11
4.3	Paramétrage & configuration	11
4.4	Déploiement / mises à jour	12
4.4.1	Mise à jour des certificats	12
4.4.2	Mise à jour de la solution vitam	12
4.5	Interruption / maintenance	12
4.5.1	Procédure d'arrêt	12
4.5.2	Procédure de démarrage	13
4.5.3	Procédure de statut	14
4.5.4	Mise à jour des certificats	14
4.6	Batches et traitements	14
5	Suivi de l'état du système	17
5.1	Veille et patches sécurité	17
5.2	Métriques	17
5.2.1	Configuration	17
5.2.1.1	Registres	17
5.2.1.2	Reporters	18
5.2.1.3	Fichier de configuration	18
5.2.2	Métier	18
5.2.3	Métriques techniques	18

5.2.3.1	Métriques système critiques	18
5.2.3.2	Indicateurs de SLA	18
5.2.3.3	Indicateurs de performance	18
5.2.4	Visualisation	18
5.2.4.1	Discover	19
5.2.4.2	Visualize	19
5.2.4.3	Dashboards	20
5.3	API de de supervision	21
5.3.1	Détail	21
5.3.1.1	/admin/v1/status	21
5.3.1.2	/admin/v1/autotest	21
5.3.1.3	/admin/v1/version	22
5.4	Logs	23
5.4.1	Changement des règles de log	24
5.5	Audit	25
5.6	Gestion de la capacité	25
5.7	Suivi de l'état de sécurité	25
5.8	Alerting	25
5.8.1	Système	25
5.8.2	Applicatif	25
6	Exploitation des composants de la solution logicelle VITAM	27
6.1	Généralités	27
6.2	Fichiers communs	27
6.2.1	Fichier /vitam/conf/<composant>/sysconfig/java_opts	27
6.2.2	Fichier /vitam/conf/<composant>/logback.xml	28
6.2.3	Fichier /vitam/conf/<composant>/logback-access.xml	28
6.2.4	Fichier /vitam/conf/<composant>/jetty-config.xml	29
6.2.5	Fichier /vitam/conf/<composant>/logbook-client.conf	33
6.2.6	Fichier /vitam/conf/<composant>/server-identity.conf	34
6.2.7	Fichier /vitam/conf/<composant>/antisamy-esapi.xml	34
6.2.8	Fichier /vitam/conf/<composant>/vitam.conf	47
6.2.9	Fichier /vitam/conf/<composant>/vitam.metrics.conf	47
6.2.10	Fichier /vitam/conf/<composant>/vitam.metrics.conf	48
6.2.11	Fichier /vitam/conf/<composant>/java.security	49
6.3	Composants	49
6.3.1	access external	49
6.3.1.1	Présentation	49
6.3.1.2	Configuration / fichiers utiles	49
6.3.1.2.1	Fichier access-external.conf	49
6.3.1.2.2	Fichier access-internal-client.conf	49
6.3.1.2.3	Fichier shiro.ini	49
6.3.1.2.4	Fichier functional-administration-client.conf	50
6.3.1.3	Opérations	50
6.3.2	access-internal	51
6.3.2.1	Présentation du composant	51
6.3.2.2	Configuration / fichiers utiles	51
6.3.2.2.1	Fichier access.conf	51
6.3.2.2.2	Fichier storage-client.conf	51
6.3.2.2.3	Fichier metadata-client.conf	51
6.3.2.3	Opérations	51
6.3.3	common-plugin	52
6.3.3.1	Présentation du composant	52
6.3.4	consul	52

6.3.4.1	Présentation	52
6.3.4.1.1	Cas serveur	52
6.3.4.1.2	Cas agent	52
6.3.4.2	Configuration / fichiers utiles	53
6.3.4.2.1	Cas des applicatifs monitorés par Consul	53
6.3.4.2.1.1	Fichier /vitam/conf/consul/service-<composant>.json	53
6.3.4.3	Opérations	54
6.3.5	elasticsearch chaîne de log	54
6.3.5.1	Présentation	54
6.3.5.2	Configuration / fichiers utiles	55
6.3.5.2.1	Fichier logging.yml	55
6.3.5.2.2	Fichier elasticsearch.yml	56
6.3.5.2.3	Fichier sysconfig/elasticsearch	59
6.3.5.2.4	Fichier /usr/lib/tmpfiles.d/elasticsearch-data.conf	60
6.3.5.3	Opérations	60
6.3.6	elasticsearch Vitam	61
6.3.6.1	Présentation	61
6.3.6.2	Configuration / fichiers utiles	61
6.3.6.2.1	Fichier logging.yml	61
6.3.6.2.2	Fichier elasticsearch.yml	63
6.3.6.2.3	Fichier sysconfig/elasticsearch	65
6.3.6.2.4	Fichier /usr/lib/tmpfiles.d/elasticsearch-data.conf	67
6.3.6.3	Opérations	67
6.3.7	functional administration	67
6.3.7.1	Présentation	67
6.3.7.2	Configuration / fichiers utiles	67
6.3.7.2.1	Fichier functional-administration.conf	68
6.3.7.2.2	Fichier logbook-client.conf	68
6.3.7.3	Opérations	68
6.3.8	ihm-demo	69
6.3.8.1	Présentation	69
6.3.8.2	Configuration / fichiers utiles	69
6.3.8.2.1	Fichier access-external-client.conf	69
6.3.8.2.2	Fichier ihm-demo.conf	69
6.3.8.2.3	Fichier ingest-external-client.conf	70
6.3.8.2.4	Fichier shiro.ini	70
6.3.8.3	Opérations	71
6.3.9	ihm-recette	72
6.3.9.1	Présentation	72
6.3.9.2	Configuration / fichiers utiles	72
6.3.9.2.1	Fichier access-external-client.conf	72
6.3.9.2.2	Fichier ihm-recette.conf	73
6.3.9.2.3	Fichier ingest-external-client.conf	73
6.3.9.2.4	Fichier functional-administration-client.conf	74
6.3.9.2.5	Fichier shiro.ini	74
6.3.9.2.6	Fichier soapui.conf	75
6.3.9.3	Opérations	75
6.3.10	ingest-external	76
6.3.10.1	Présentation	76
6.3.10.2	Configuration / fichiers utiles	76
6.3.10.2.1	Fichier ingest-external.conf	76
6.3.10.2.2	Fichier ingest-internal-client.conf	77
6.3.10.2.3	Fichier scan-clamav.sh	77
6.3.10.3	Opérations	78

6.3.11	ingest-internal	78
6.3.11.1	Présentation	78
6.3.11.2	Configuration / fichiers utiles	79
6.3.11.2.1	Fichier <code>ingest-internal.conf</code>	79
6.3.11.3	Opérations	79
6.3.12	log server	79
6.3.12.1	Présentation	79
6.3.12.2	Configuration / fichiers utiles	80
6.3.12.3	Opérations	80
6.3.13	logbook	80
6.3.13.1	Présentation	80
6.3.13.2	Configuration / fichiers utiles	81
6.3.13.2.1	Fichier <code>logbook.conf</code>	81
6.3.13.3	Opérations	81
6.3.14	metadata	82
6.3.14.1	Présentation	82
6.3.14.2	Configuration / fichiers utiles	82
6.3.14.2.1	Fichier <code>metadata.conf</code>	82
6.3.14.3	Opérations	82
6.3.15	mongoC	83
6.3.15.1	Présentation	83
6.3.15.2	Configuration / fichiers utiles	83
6.3.15.3	Opérations	83
6.3.16	mongoclient	84
6.3.16.1	Présentation	84
6.3.16.2	Configuration / fichiers utiles	84
6.3.16.2.1	Paramétrage bases de données à monitorer	84
6.3.16.2.2	Configuration	85
6.3.16.2.2.1	Fichier <code>mongoclient.conf</code>	85
6.3.16.3	Opérations	85
6.3.17	mongoD	86
6.3.17.1	Présentation	86
6.3.17.2	Configuration / fichiers utiles	86
6.3.17.3	Opérations	86
6.3.18	mongoS	87
6.3.18.1	Présentation	87
6.3.18.2	Configuration / fichiers utiles	87
6.3.18.3	Opérations	87
6.3.19	processing	87
6.3.19.1	Présentation	87
6.3.19.2	Configuration / fichiers utiles	88
6.3.19.2.1	Fichier <code>processing.conf</code>	88
6.3.19.2.2	Fichier <code>version.conf</code>	88
6.3.19.2.3	Fichier <code>storage-client.conf</code>	88
6.3.19.3	Opérations	88
6.3.20	siegfried	89
6.3.20.1	Présentation	89
6.3.20.2	Configuration / fichiers utiles	89
6.3.20.3	Opérations	89
6.3.21	storage-engine	90
6.3.21.1	Présentation	90
6.3.21.2	Configuration / fichiers utiles	90
6.3.21.2.1	Fichier <code>driver-location.conf</code>	90
6.3.21.2.2	Fichier <code>driver-mapping.conf</code>	90

6.3.21.2.3	Fichier static-offer.json	90
6.3.21.2.4	Fichier static-strategy.json	91
6.3.21.2.5	Fichier storage-engine.conf	91
6.3.21.3	Opérations	91
6.3.22	storage-offer-default	92
6.3.22.1	Présentation	92
6.3.22.2	Configuration / fichiers utiles	92
6.3.22.2.1	Fichier default-offer.conf	92
6.3.22.2.2	Fichier default-storage.conf	92
6.3.22.3	Opérations	92
6.3.23	worker	93
6.3.23.1	Présentation	93
6.3.23.2	Configuration / fichiers utiles	93
6.3.23.2.1	Fichier format-identifiers.conf	93
6.3.23.2.2	Fichier functional-administration-client.conf.j2	93
6.3.23.2.3	Fichier logbook-client.conf	94
6.3.23.2.4	Fichier metadata-client.conf	94
6.3.23.2.5	Fichier storage-client.conf	94
6.3.23.2.6	Fichier version.conf	94
6.3.23.2.7	Fichier worker.conf	94
6.3.23.3	Opérations	95
6.3.24	workspace	95
6.3.24.1	Présentation	95
6.3.24.2	Configuration / fichiers utiles	95
6.3.24.2.1	Fichier workspace.conf	96
6.3.24.3	Opérations	96
7	Aide à l'exploitation	97
7.1	Analyse de premier niveau	97
7.1.1	Etat Consul	97
7.1.2	Etat VITAM via Kibana	98
8	Questions Fréquemment Posées	99
8.1	Présentation	99
8.2	Retour d'expérience / cas rencontrés	99
9	Contacts et support	101
9.1	Contacts	101
10	Exploitation par composant	103
10.1	Access	103
10.2	Common	103
10.3	Functional administration	103
10.4	Ingest	103
10.5	Logbook	103
10.6	Metadata	103
10.7	Processing	103
10.7.1	Introduction	103
10.7.1.1	But de cette documentation	103
10.7.2	Processing	103
10.7.2.1	Configuration du worker	104
10.7.2.2	Supervision du service	104
10.8	Storage	104
10.8.1	Introduction	104
10.8.1.1	But de cette documentation	104

10.8.2	Storage Engine	104
10.8.2.1	Configuration du moteur de stockage	105
10.8.2.2	Configuration du driver de l'offre de stockage par défaut	106
10.8.2.3	Supervision du service	106
10.8.3	Storage Offer Default	106
10.8.3.1	Configuration de l'offre de stockage	106
10.8.3.2	Supervision du service	106
10.9	Technical administration	106
10.10	Worker	106
10.10.1	Introduction	106
10.10.1.1	But de cette documentation	106
10.11	Workspace	107
11	Annexes	109
	Index	115

Prudence : Cette documentation est un travail en cours ; elle est susceptible de changer de manière conséquente.

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

Le logiciel *VITAM* est publié sous la license [CeCILL 2.1](http://www.cecill.info/licences/Licence_CeCILL_V2.1-fr.html)¹ ; la documentation associée (comprenant le présent document) est publiée sous license [CC-BY-SA 3.0](https://creativecommons.org/licenses/by-sa/3.0/fr/legalcode)².

2.2 Documents de référence

2.2.1 Documents internes

Tableau 2.1 – Documents de référence VITAM

Nom	Lien
<i>DAT</i>	(à renseigner)
<i>DIN</i>	(à renseigner)
<i>DEX</i>	(à renseigner)
Release notes	(à renseigner)

2.2.2 Référentiels externes

Référentiel Général d'Interopérabilité [RGI] V1.0 du 12 juin 2009 approuvé par arrêté du Premier ministre du 9 novembre 2009

Règles d'interopérabilité (format, protocoles, encodages, etc.) rentrant dans le champ d'application de l'ordonnance n°2005-1516 du 8 décembre 2005 relative aux échanges électroniques entre les usagers et les autorités administratives et entre les autorités administratives.

<https://references.modernisation.gouv.fr/rgi-interoperabilite>

Référentiel Général de Sécurité [RGS] V2.0 du 13 juin 2014 approuvé par arrêté du Premier ministre du 13 juin 2014

Le RGS précise les règles de sécurité s'imposant aux autorités administratives dans la sécurisation de leur SI et notamment sur les dispositifs de sécurité relatifs aux mécanismes cryptographiques et à l'utilisation de certificats électroniques et contremarques de temps. Le RGS propose également des

1. http://www.cecill.info/licences/Licence_CeCILL_V2.1-fr.html

2. <https://creativecommons.org/licenses/by-sa/3.0/fr/legalcode>

bonnes pratiques en matière de SSI. Le RGS découle de l'application de l'ordonnance n°2005-1516 du 8 décembre 2005 relative aux échanges électroniques entre les usagers et les autorités administratives et entre les autorités administratives.

<https://references.modernisation.gouv.fr/rgs-securite>

Norme OAIS (ISO 14721 :2012 – 1 septembre 2012) Systèmes de transfert des informations et données spatiales – Système ouvert d'archivage d'information (SOAI) - Modèle de référence

Standard d'échange de données pour l'archivage (SEDA) Transfert, communication, élimination, restitution, modification – Version 1.0 – Septembre 2012

Cadre normatif pour les différents échanges d'informations entre les services d'archives publics et leurs partenaires : entités productrices des archives, entités gestionnaires, entités de contrôle des processus, et enfin entités qui utilisent ces archives. Il concerne également les échanges entre plusieurs services d'archives (services publics d'archives, prestataires d'archivage, archivage intermédiaire, archivage définitif).

<http://www.archivesdefrance.culture.gouv.fr/seda/>

2.3 Glossaire

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.

DIN Dossier d'Installation

DEX Dossier d'EXploitation

DAT Dossier d'Architecture Technique

IHM Interface Homme Machine

VITAM Valeurs Immatérielles Transférées aux Archives pour Mémoire

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

Deb Debian ; il s'agit du format de packets logiciels nativement utilisé par les distributions Debian GNU/Linux

API Application Programming Interface

BDD Base De Données

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*

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.

NoSQL Base de données non-basée sur un paradigme classique des bases relationnelles. *Définition*³

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*⁴

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*⁵

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

4. https://fr.wikipedia.org/wiki/Attaque_de_l'homme_du_milieu

5. https://fr.wikipedia.org/wiki/Domain_Name_System_Security_Extensions

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](#)⁶

SIA Système d'Informations Archivistique

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.

TNR Tests de Non-Régression

6. https://fr.wikipedia.org/wiki/Infrastructure_%C3%A0_cl%C3%A9s_publics

Architecture de la solution logicielle VITAM

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

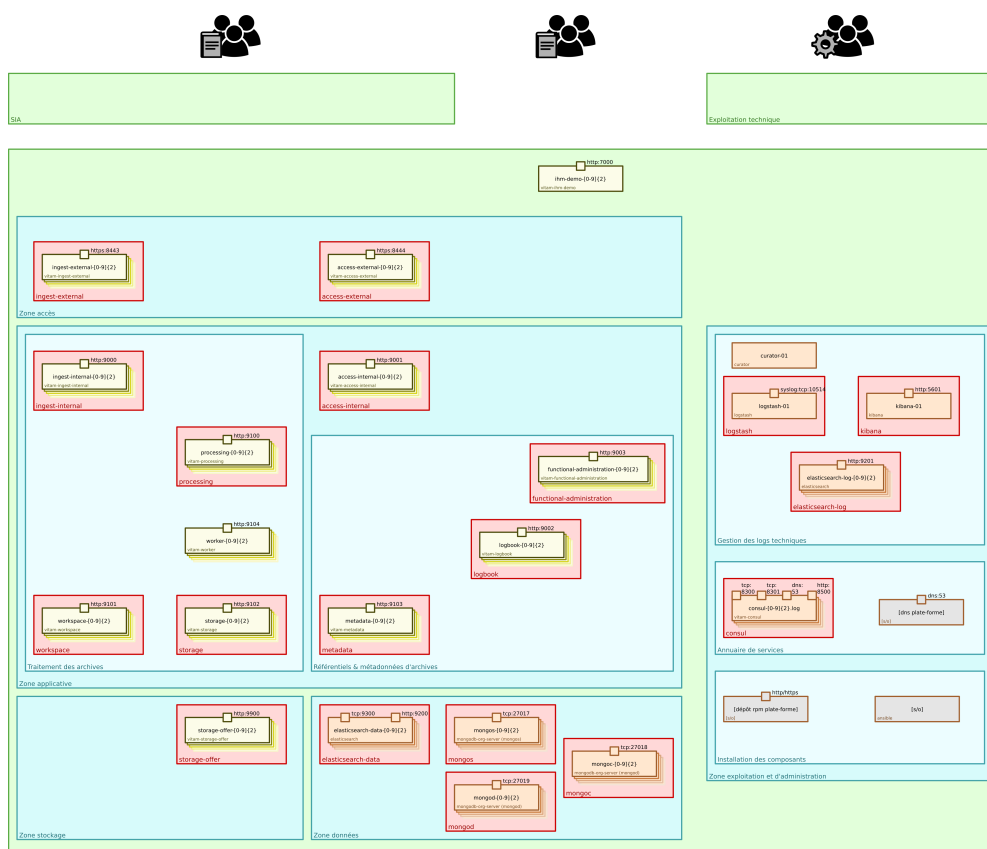


Fig. 3.1 – Vue d’ensemble d’un déploiement VITAM : zones, composants

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. installation et exploitation des composants external).

2. IHM d'admin

Dans cette version, la gestion des utilisateurs se fait par configuration d'un fichier plat (Cf. exploitation du composant ihm-demo).

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.

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 *VITAM* (ansible / inventaire).

Voir aussi :

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

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-rpm/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-rpm/vitam.yml -i environments-rpm/<fichier d
↳ 'inventaire> --ask-vault-pass --tags update_vitam_certificates
ansible-playbook ansible-vitam-rpm-extra/extra.yml -i environments-rpm/<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-rpm/vitam.yml -i environments-rpm/<fichier d
↳ 'inventaire> --vault-password-file vault_pass.txt --tags update_vitam_certificates
ansible-playbook ansible-vitam-rpm-extra/extra.yml -i environments-rpm/<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 *VITAM* (tout comme pour sa première installation), se référer au *DIN*, ainsi qu'à la "release note" associée à toute montée de version.

Ce document détaille les pré-requis, la configuration des fichiers pour effectuer un déploiement.

Enfin, ce document 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	Résultat OK pour passer à l'étape suivante
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 `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.

4.5.2 Procédure de démarrage

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

Tableau 4.2 – Cinématique de démarrage de VITAM

Composant	Ordre	pour passer à l'étape suivante
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 `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 `status_vitam.yml`), pour réaliser de façon automatisée les tests “autotest” intégrés dans 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 sera décrite dans une prochaine version de ce document.

4.6 Batches et traitements

Il existe également des jobs Curator de :

- fermeture d’index
- suppression d’index fermés

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

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
```

Suivi de l'état du système

5.1 Veille et patchs sécurité

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

- Runtime Java (OpenJDK 8)

5.2 Métriques

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 Kibana⁷.

L'application Kibana comporte 4 sections qui seront développées :

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

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

Néanmoins si vous souhaitez travailler avec Kibana, il est judicieux de consulter la documentation officielle. Celle-ci n'ayant pour but qu'une présentation sommaire de l'outil.

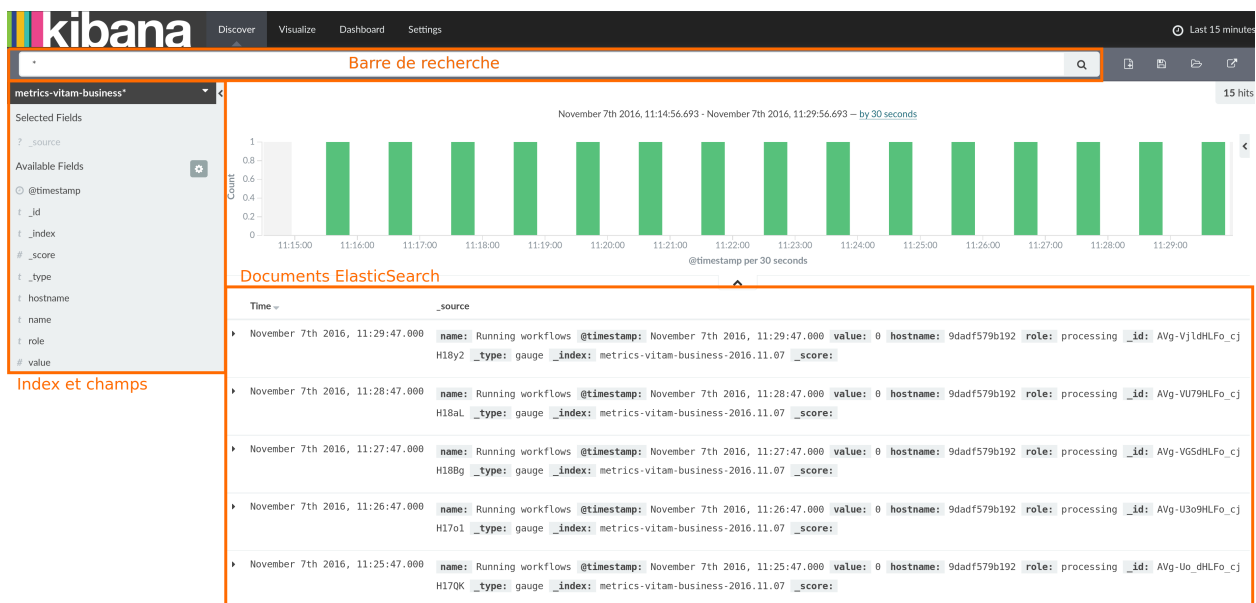
Voir aussi :

[Documentation officielle de Kibana](#)⁸

5.2.4.1 Discover

La section **Discover** permet de consulter rapidement les données présentes dans un index d'ElasticSearch. Pour cela il suffit de sélectionner un index dans la barre latérale gauche, de choisir les champs que l'on souhaite consulter (optionnel) et les données apparaissent triées par ordre chronologique décroissant.

Il est possible d'effectuer des recherches poussées sur les documents, comme des expressions régulières, grâce à la barre de recherche en haut de la page. Une fois la recherche exécutée, il peut être utile de la sauvegarder afin de la réutiliser pour des visualisations.



5.2.4.2 Visualize

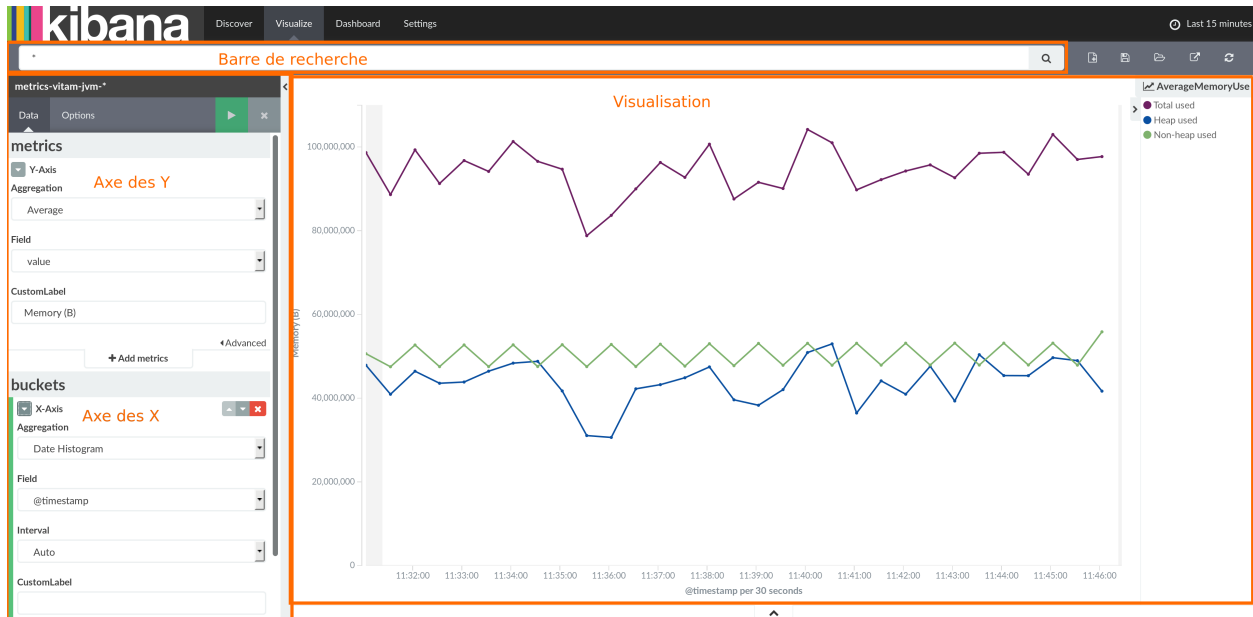
La section **Visualize** permet de consulter les données présentes dans ElasticSearch à travers différents graphiques statistiques. Les graphiques disponibles sont :

- **Area chart** : utile pour un regroupement de séries chronologiques dans lequel le total des séries est plus important que la différence entre plusieurs séries.
- **Data table** : un tableau de données classique.
- **Line chart** : graphique pour des séries temporelles. Très utile pour comparer deux séries entre elles.
- **Markdown widget** : utile pour insérer des informations sur un dashboard Kibana.
- **Metric** : représentation d'une aggrégation de données sous la forme d'un seul nombre.
- **Pie chart** : un diagramme circulaire classique.
- **Tile map** : représentation de coordonnées géographiques sur une carte.
- **Vertical bar chart** : un histogramme classique.

8. <https://www.elastic.co/guide/en/kibana/current/index.html>

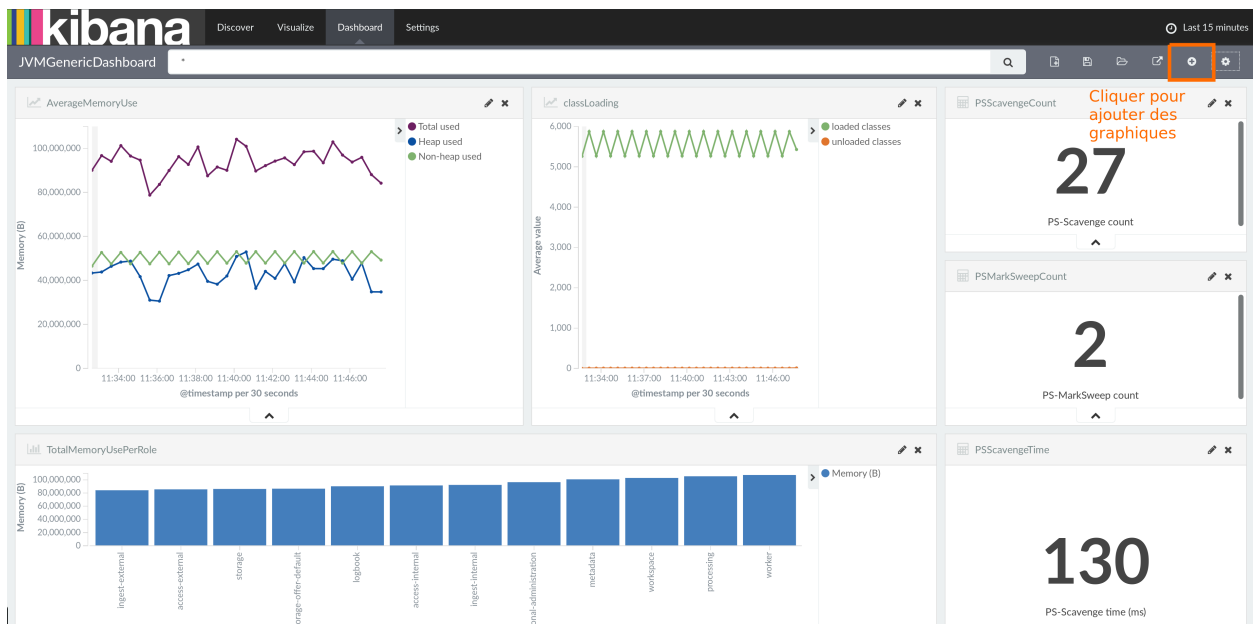
La barre latérale gauche du panneau de visualisation permet de configurer la donnée à représenter. Pour l'axe des Y, il est impératif d'utiliser une agrégation (moyenne, minimum/maximum, écart type...) sur une valeur pour la représenter. En fonction du graphique sélectionné, il est possible de configurer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

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/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

5.3.1.3 /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

5.4 Logs

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 VITAM, choisir d'utiliser sa propre solution de "regroupement" des logs ou la solution embarquée dans la solution 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 Elasticsearch
 - un moteur logstash, parsant les messages Vitam
 - un afficheur de rendu/aggrégation de données Kibana

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`⁹, 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`¹⁰, 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`).

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

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

<p>Prudence : 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 VITAM sont déployés par un playbook ansible qui :

1. déploie, selon l'inventaire employé, les packages rpm 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 *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="{ { gc_opts } } { { memory_opts } } { { java_opts } } -Dorg.owasp.esapi.
35   ↳ resources={ { vitam_folder_conf } } -Dlogback.configurationFile={ { vitam_folder_conf } } /
36   ↳ logback.xml -Dvitam.config.folder={ { vitam_folder_conf } } -Dvitam.data.folder={ {
37   ↳ vitam_folder_data } } -Dvitam.tmp.folder={ { vitam_folder_tmp } } -Dvitam.log.folder={
38   ↳ { vitam_folder_log } } -Djava.security.properties={ { vitam_folder_conf } } / java.security"
39 JAVA_ARGS="{ { java_args } }"

```

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

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <configuration>
3
4
5      <appender name="FILE" class="ch.qos.logback.core.rolling.RollingFileAppender">
6          <rollingPolicy class="ch.qos.logback.core.rolling.
7   ↳ TimeBasedRollingPolicy">
8              <fileNamePattern>{ { vitam_folder_log } } / accesslog- { { vitam_
9   ↳ component } } . %d { yyyy-MM-dd } . log </fileNamePattern>
10             <maxHistory> { { days_to_delete_access_local } } </maxHistory>
11             <totalSizeCap>14GB</totalSizeCap>
12         </rollingPolicy>
13         <encoder>
14             <pattern>%h %l %u %t "%r" %s %b "%i{Referer}" "%i{User-agent}
15   ↳ " %D %i{X-Request-ID} %i{X-TenantID} %i{X-Application-ID} </pattern>
16         </encoder>
17     </appender>
18
19     <appender-ref ref="FILE" />
20 </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_component }}.%d
↳ {yyyy-MM-dd}.%i.log</fileNamePattern>
15             <maxFileSize>10MB</maxFileSize>
16             <maxHistory>{{ days_to_delete_local }}</maxHistory>
17             <totalSizeCap>5GB</totalSizeCap>
18         </rollingPolicy>
19
20         <!-- TODO : replace with rolling file appender -->
21         <!-- <file>{{vitam_folder_log}}/{{ vitam_component }}.log</file>
22         <append>true</append> -->
23         <encoder>
24             <pattern>%d{ISO8601} [[%thread]] [%X{X-REQUEST-ID}] %-5level
↳ %logger - %replace(%caller{1..2}){'Caller\+1          at |\n',''} : %msg %rootException%n
↳ </pattern>
25         </encoder>
26     </appender>
27
28     <appender name="SYSLOG" class="ch.qos.logback.classic.net.SyslogAppender">
29         <syslogHost>localhost</syslogHost>
30         <facility>{{syslog_facility}}</facility>
31         <suffixPattern>vitam-{{vitam_component}}: %d{ISO8601} [[%thread]] [%X
↳ {X-REQUEST-ID}] %-5level %logger - %replace(%caller{1..2}){'Caller\+1          at |\n',''
↳ '} : %msg %rootException%n</suffixPattern>
32     </appender>
33     <!-- By default, the level of the root level is set to TRACE -->
34     <root level="{{ log_level }}">
35         <!-- <appender-ref ref="STDOUT" /> -->
36         <appender-ref ref="FILE" />
37         <appender-ref ref="SYSLOG" />
38     </root>
39
40     <logger name="org.eclipse.jetty" level="WARN"/>
41     <logger name="fr.gouv.vitam.storage.engine.server.logbook.StorageLogbookMock"
↳ level="INFO"/>
42     <logger name="fr.gouv.vitam.common" level="WARN" />
43 </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-spy.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"/></
66     <Arg>
67     </Call>
68     -->
69     </New>
70     <Call name="addConnector">
71         <Arg>
72             <New class="org.eclipse.jetty.server.ServerConnector">
73                 <Arg name="server"><Ref refid="Server" /></Arg>
74                 <Arg name="factories">
75                     <Array type="org.eclipse.jetty.server.ConnectionFactory">
76                         <Item>
77                             <New class="org.eclipse.jetty.server.HttpConnectionFactory
78                             <Arg name="config"><Ref refid="httpConfig" /></Arg>
79                             </New>
80                         </Item>
81                     </Array>
82                 </Arg>
83                 <Set name="port">{{port_http}}</Set>
84                 <Set name="idleTimeout">
85                     <Property name="http.timeout" default="{{port_http_timeout}}"/>
86                 </Set>
87             </New>
88         </Arg>
89     </Call>
90
91     <!-- ===== -->
92     <!-- Set the default handler structure for the Server -->
93     <!-- A handler collection is used to pass received requests to -->
94     <!-- both the ContextHandlerCollection, which selects the next -->
95     <!-- handler by context path and virtual host, and the -->
96     <!-- DefaultHandler, which handles any requests not handled by -->
97     <!-- the context handlers. -->
98     <!-- Other handlers may be added to the "Handlers" collection, -->
99     <!-- for example the jetty-requestlog.xml file adds the -->
100    <!-- RequestLogHandler after the default handler -->
101    <!-- ===== -->
102    <Set name="handler">
103        <New id="Handlers" class="org.eclipse.jetty.server.handler.HandlerCollection">
104            <Set name="handlers">
105                <Array type="org.eclipse.jetty.server.Handler">
106                    <Item>
107                        <New id="Contexts" class="org.eclipse.jetty.server.handler.
108    <ContextHandlerCollection"/>
109                    </Item>
110                    <Item>
111                        <New id="DefaultHandler" class="org.eclipse.jetty.server.
112    <handler.DefaultHandler"/>
113                    </Item>
114                </Array>
115            </Set>
116        </New>
117    </Set>

```

```

115
116     <Set name="RequestLog">
117         <New id="RequestLogImpl" class="ch.qos.logback.access.jetty.RequestLogImpl
118     ↪ ">
119             <Set name="fileName">{{vitam_folder_conf}}/logback-access.xml</Set>
120             </New>
121         </Set>
122         <Ref id="RequestLogImpl">
123             <Call name="start"/>
124         </Ref>
125
126     <!-- ===== -->
127     <!-- extra server options -->
128     <!-- ===== -->
129     <Set name="stopAtShutdown">true</Set>
130     <Set name="stopTimeout">5000</Set>
131     <Set name="dumpAfterStart">false</Set>
132     <Set name="dumpBeforeStop">false</Set>
133     {% if tls_active %}
134     <New id="httpsConfig" class="org.eclipse.jetty.server.HttpConfiguration">
135         <Call name="addCustomizer">
136             <Arg>
137                 <New class="org.eclipse.jetty.server.SecureRequestCustomizer" />
138             </Arg>
139         </Call>
140     </New>
141     <New id="sslContextFactory" class="org.eclipse.jetty.util.ssl.
142     ↪ SslContextFactory">
143         <Set name="KeyStorePath">{{vitam_folder_conf}}/keystore_{{vitam_component}}.
144     ↪ jks</Set>
145         <Set name="KeyStorePassword">{{password_keystore}}</Set>
146         <Set name="KeyManagerPassword">{{password_manager_keystore}}</Set>
147         <Set name="TrustStorePath">{{vitam_folder_conf}}/truststore_{{vitam_component}}
148     ↪ .jks</Set>
149         <Set name="TrustStorePassword">{{password_truststore}}</Set>
150         <Set name="TrustStoreType">JKS</Set>
151         <Set name="NeedClientAuth">false</Set>
152         <Set name="WantClientAuth">true</Set>
153         <Set name="IncludeCipherSuites">
154             <Array type="String">
155                 <Item>TLS_ECDHE.*</Item>
156                 <Item>TLS_DHE_RSA.*</Item>
157             </Array>
158         </Set>
159         <Set name="IncludeProtocols">
160             <Array type="String">
161                 <Item>TLSv1</Item>
162                 <Item>TLSv1.1</Item>
163                 <Item>TLSv1.2</Item>
164             </Array>
165         </Set>
166         <Set name="ExcludeCipherSuites">
167             <Array type="String">
168                 <Item>.*NULL.*</Item>
169                 <Item>.*RC4.*</Item>
170                 <Item>.*MD5.*</Item>
171                 <Item>.*DES.*</Item>
172                 <Item>.*DSS.*</Item>

```



```

169     </Array>
170   </Set>
171   <Set name="UseCipherSuitesOrder">true</Set>
172   <Set name="RenegotiationAllowed">true</Set>
173 </New>
174 <New id="sslConnectionFactory" class="org.eclipse.jetty.server.
↪SslConnectionFactory">
175   <Arg name="sslContextFactory">
176     <Ref refid="sslContextFactory" />
177   </Arg>
178   <Arg name="next">http/1.1</Arg>
179 </New>
180 <New id="sslConnector" class="org.eclipse.jetty.server.ServerConnector">
181   <Arg name="server">
182     <Ref refid="Server" />
183   </Arg>
184   <Arg name="factories">
185     <Array type="org.eclipse.jetty.server.ConnectionFactory">
186       <Item>
187         <Ref refid="sslConnectionFactory" />
188       </Item>
189       <Item>
190         <New class="org.eclipse.jetty.server.HttpConnectionFactory">
191           <Arg name="config">
192             <Ref refid="httpsConfig" />
193           </Arg>
194         </New>
195       </Item>
196     </Array>
197   </Arg>
198   <Set name="port">
199     <SystemProperty name="jetty.port" default="{{ port_https }}" />
200   </Set>
201 </New>
202 <Call name="setConnectors">
203   <Arg>
204     <Array type="org.eclipse.jetty.server.ServerConnector">
205       <!-- <Item>
206         <Ref refid="connector" />
207       </Item> -->
208       <Item>
209         <Ref refid="sslConnector" />
210       </Item>
211     </Array>
212   </Arg>
213 </Call>
214 {% endif %}
215 </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}}

```

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

```

1 identityName: {{ansible_nodename}}
2 identityRole: {{vitam_component}}
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>
11 ↪ <div> <code> <quote>
12 -->
13
14 <anti-samy-rules xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
15     xsi:noNamespaceSchemaLocation="antisamy.xsd">
16
17     <directives>
18         <directive name="omitXmlDeclaration" value="true"/>
19         <directive name="omitDoctypeDeclaration" value="true"/>
20         <directive name="maxInputSize" value="2000000"/>
21         <directive name="embedStyleSheets" value="false"/>
22     </directives>
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         <regexp name="offsiteURL" value="(\s)*((ht|f)tp(s?)://|mailto:)[A-Za-
38 ↪ z0-9]+[~a-zA-Z0-9-_\.\@#\$%&;:;,\?=/\+!]*(\s)*"/>
39
40     </common-regexps>
41
42     <!--
43     Tag.name = a, b, div, body, etc.
44     Tag.action = filter: remove tags, but keep content, validate: keep content as
45 ↪ long as it passes rules, remove: remove tag and contents
46     Attribute.name = id, class, href, align, width, etc.

```

```

45     Attribute.onInvalid = what to do when the attribute is invalid, e.g., remove
↳ the tag (removeTag), remove the attribute (removeAttribute), filter the tag
↳ (filterTag)
46     Attribute.description = What rules in English you want to tell the users they
↳ can have for this attribute. Include helpful things so they'll be able to tune
↳ their HTML
47
48     -->
49
50     <!--
51     Some attributes are common to all (or most) HTML tags. There aren't many that
↳ qualify for this. You have to make sure there's no
52     collisions between any of these attribute names with attribute names of other
↳ tags that are for different purposes.
53     -->
54
55     <common-attributes>
56
57
58         <attribute name="lang" description="The 'lang' attribute tells the
↳ browser what language the element's attribute values and content are written in">
59             <regexp-list>
60                 <regexp value="[a-zA-Z]{2,20}" />
61             </regexp-list>
62         </attribute>
63
64         <attribute name="title" description="The 'title' attribute provides
↳ text that shows up in a 'tooltip' when a user hovers their mouse over the element">
65             <regexp-list>
66                 <regexp name="htmlTitle" />
67             </regexp-list>
68         </attribute>
69
70         <attribute name="href" onInvalid="filterTag">
71             <regexp-list>
72                 <regexp name="onsiteURL" />
73                 <regexp name="offsiteURL" />
74             </regexp-list>
75         </attribute>
76
77         <attribute name="align" description="The 'align' attribute of an HTML
↳ element is a direction word, like 'left', 'right' or 'center'">
78             <literal-list>
79                 <literal value="center" />
80                 <literal value="left" />
81                 <literal value="right" />
82                 <literal value="justify" />
83                 <literal value="char" />
84             </literal-list>
85         </attribute>
86
87     </common-attributes>
88
89
90     <!--
91     This requires normal updates as browsers continue to diverge from the W3C and
↳ each other. As long as the browser wars continue
92     this is going to continue. I'm not sure war is the right word for what's
↳ going on. Doesn't somebody have to win a war after

```

```

93     a while?
94     -->
95
96     <global-tag-attributes>
97         <attribute name="title"/>
98         <attribute name="lang"/>
99     </global-tag-attributes>
100
101
102     <tag-rules>
103
104         <!-- Tags related to JavaScript -->
105
106         <tag name="script" action="remove"/>
107         <tag name="noscript" action="remove"/>
108
109         <!-- Frame & related tags -->
110
111         <tag name="iframe" action="remove"/>
112         <tag name="frameset" action="remove"/>
113         <tag name="frame" action="remove"/>
114         <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="&"/>
174     <entity name="nbsp" cdata="&#160;"/>
175
176     <entity name="iexcl" cdata="&#161;"/> <!--inverted exclamation_
↪mark, U+00A1 ISOnum -->
177     <entity name="cent" cdata="&#162;"/> <!--cent sign, U+00A2 ISOnum_
↪-->
178     <entity name="pound" cdata="&#163;"/> <!--pound sign, U+00A3_
↪ISOnum -->
179     <entity name="curren" cdata="&#164;"/> <!--currency sign, U+00A4_
↪ISOnum -->
180     <entity name="yen" cdata="&#165;"/> <!--yen sign = yuan sign,_
↪U+00A5 ISOnum -->
181     <entity name="brvbar" cdata="&#166;"/> <!--broken bar = broken_
↪vertical bar, U+00A6 ISOnum -->
182     <entity name="sect" cdata="&#167;"/> <!--section sign, U+00A7_
↪ISOnum -->
183     <entity name="uml" cdata="&#168;"/> <!--diaeresis = spacing_
↪diaeresis, U+00A8 ISodia -->
184     <entity name="copy" cdata="&#169;"/> <!--copyright sign, U+00A9_
↪ISOnum -->
185     <entity name="ordf" cdata="&#170;"/> <!--feminine ordinal_
↪indicator, U+00AA ISOnum -->
186     <entity name="laquo" cdata="&#171;"/> <!--left-pointing double_
↪angle quotation mark = left pointing guillemet, U+00AB ISOnum -->
187     <entity name="not" cdata="&#172;"/> <!--not sign, U+00AC ISOnum --
↪>
188     <entity name="shy" cdata="&#173;"/> <!--soft hyphen =_
↪discretionary hyphen, U+00AD ISOnum -->
189     <entity name="reg" cdata="&#174;"/> <!--registered sign =_
↪registered trade mark sign, U+00AE ISOnum -->
190     <entity name="macr" cdata="&#175;"/> <!--macron = spacing macron_
↪= overline = APL overbar, U+00AF ISodia -->
191     <entity name="deg" cdata="&#176;"/> <!--degree sign, U+00B0_
↪ISOnum -->

```

```

192      <entity name="plusmn" cdata="&#177;"/> <!--plus-minus sign = plus-
↳or-minus sign, U+00B1 ISOnum -->
193      <entity name="sup2" cdata="&#178;"/> <!--superscript two =
↳superscript digit two = squared, U+00B2 ISOnum -->
194      <entity name="sup3" cdata="&#179;"/> <!--superscript three =
↳superscript digit three= cubed, U+00B3 ISOnum -->
195      <entity name="acute" cdata="&#180;"/> <!--acute accent = spacing
↳acute, U+00B4 ISodia -->
196      <entity name="micro" cdata="&#181;"/> <!--micro sign, U+00B5
↳ISOnum -->
197      <entity name="para" cdata="&#182;"/> <!--pilcrow sign = paragraph
↳sign, U+00B6 ISOnum -->
198      <entity name="middot" cdata="&#183;"/> <!--middle dot = Georgian
↳comma = Greek middle dot, U+00B7 ISOnum -->
199      <entity name="cedil" cdata="&#184;"/> <!--cedilla = spacing
↳cedilla, U+00B8 ISodia -->
200      <entity name="sup1" cdata="&#185;"/> <!--superscript one =
↳superscript digit one, U+00B9 ISOnum -->
201      <entity name="ordm" cdata="&#186;"/> <!--masculine ordinal
↳indicator, U+00BA ISOnum -->
202      <entity name="raquo" cdata="&#187;"/> <!--right-pointing double
↳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 ISOftech -->
335         <entity name="Prime" cdata="&#8243;"/> <!--double prime = seconds_
↳ = inches, U+2033 ISOftech -->
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 ISOamso -->
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
395      <entity name="lowast" cdata="&#8727;"/> <!--asterisk operator,
↳U+2217 ISOtech -->
396      <entity name="radic" cdata="&#8730;"/> <!--square root = radical,
↳sign, U+221A ISOtech -->
397      <entity name="prop" cdata="&#8733;"/> <!--proportional to, U+221D,
↳ISOtech -->
398      <entity name="infin" cdata="&#8734;"/> <!--infinity, U+221E,
↳ISOtech -->
399      <entity name="ang" cdata="&#8736;"/> <!--angle, U+2220 ISOamso -->
400      <entity name="and" cdata="&#8743;"/> <!--logical and = wedge,
↳U+2227 ISOtech -->
401      <entity name="or" cdata="&#8744;"/> <!--logical or = vee, U+2228,
↳ISOtech -->
402      <entity name="cap" cdata="&#8745;"/> <!--intersection = cap,
↳U+2229 ISOtech -->
403      <entity name="cup" cdata="&#8746;"/> <!--union = cup, U+222A,
↳ISOtech -->
404      <entity name="int" cdata="&#8747;"/> <!--integral, U+222B ISOtech,
↳-->
405      <entity name="there4" cdata="&#8756;"/> <!--therefore, U+2234,
↳ISOtech -->
406      <entity name="sim" cdata="&#8764;"/> <!--tilde operator = varies,
↳with = similar to, U+223C ISOtech -->
407
408      <!-- tilde operator is NOT the same character as the tilde, U+007E,
409          although the same glyph might be used to represent both -->
410
411      <entity name="cong" cdata="&#8773;"/> <!--approximately equal to,
↳U+2245 ISOtech -->
412      <entity name="asymp" cdata="&#8776;"/> <!--almost equal to =,
↳asymptotic to, U+2248 ISOamsr -->
413      <entity name="ne" cdata="&#8800;"/> <!--not equal to, U+2260,
↳ISOtech -->
414      <entity name="equiv" cdata="&#8801;"/> <!--identical to, U+2261,
↳ISOtech -->
415      <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 = APL
↳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 : {{secret_platform}}

```

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
↳suivantes :
4 #
5 # metricsJersey: true / false           Active ou non les métriques Jersey
6 # metricsJVM: true / false             Active ou non les métriques JVM
7 #
8 # metricReporter: ELASTICSEARCH | LOGBACK | NONE           défini le
↳type de reporter
9 # metricReporterInterval: int > 0           défini l
↳'interval entre chaque reporting
10 # metricReporterIntervalUnit: TimeUnit (ex: SECONDS, MINUTES...)   défini le
↳type d'interval
11 #
12 # Si le reporter est de type LOGBACK, la clé suivante est configurable:
13 # metricLogLevel: DEBUG | INFO | WARN | ERROR ...           défini le
↳niveau de log Logback
14 #
15 # Si le reporter est de type ELASTICSEARCH, la clé suivante est obligatoire :
16 #
17 # (un tableau avec les différentes adresses des bases ElasticSearch)
18 # metricReporterHosts:
19 #     - 127.0.0.1:9201
20 #     - 0.0.0.0:80
21 #     - 8.8.8.8:22
22
23 {% if (groups['hosts-log-server'] | length) > 0 %}
24 metricsJersey: true

```



```

25 metricsJVM: true
26
27 metricReporter: ELASTICSEARCH
28 metricReporterHosts:
29 {% for host in groups['hosts-elasticsearch-log'] %}
30   - "{{ host }}:{{ elasticsearch_log_http_port }}"
31 {% endfor %}
32 metricLogLevel: DEBUG
33 metricReporterInterval: 1
34 metricReporterIntervalUnit: MINUTES
35 {% endif %}

```

6.2.10 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 l
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-log-server'] | length) > 0 %}
29 metricsJersey: true
30 metricsJVM: true
31
32 metricReporter: ELASTICSEARCH
33 metricReporterHosts:
34 {% for host in groups['hosts-elasticsearch-log'] %}
35   - "{{ host }}:{{ elasticsearch_log_http_port }}"
36 {% endfor %}
37 metricLogLevel: DEBUG
38 metricReporterInterval: 1
39 metricReporterIntervalUnit: MINUTES
40 {% endif %}

```


6.2.11 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(' ', ' ') }" ]

```

6.3.1.2.2 Fichier access-internal-client.conf

```

serverHost: {{vitam_accessinternal_host}}
serverPort: {{vitam_accessinternal_port}}

```

6.3.1.2.3 Fichier shiro.ini

```

[main]

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

```

```
x509credentialsMatcher = fr.gouv.vitam.common.auth.core.authc.  
↳X509CredentialsSha256Matcher  
  
x509Realm = fr.gouv.vitam.common.auth.core.realm.X509KeystoreFileRealm  
x509Realm.grantedKeyStoreName = {{vitam_folder_conf}}/grantedstore_access-external.jks  
x509Realm.grantedKeyStorePassphrase = {{password_grantedstore}}  
x509Realm.trustedKeyStoreName = {{vitam_folder_conf}}/truststore_access-external.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}}
```

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/v1/status

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

logrotate / purge des logs si occupation importante ?

- 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_baseurl}}
urlWorkspace: {{vitam_workspace_baseurl}}
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}}
```

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}}
```

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/v1/status

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

logrotate / purge des logs si occupation importante ?

- cas des batches

N/A

6.3.3 common-plugin

6.3.3.1 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 consul

6.3.4.1 Présentation

Consul est un DNS applicatif.

6.3.4.1.1 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.

6.3.4.1.2 Cas agent

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

6.3.4.2 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*.

6.3.4.2.1 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` et est basé sur ce squelette :

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

```

1 {
2   "service": {
3     "name": "{{ vitam_component }}",
4     {% if tls_active==true %}
5       "port": {{ port_https }},
6     {% else %}
7       "port": {{ port_http }},
8     {% endif %}
9     "enableTagOverride": false,
10    "checks": [
11      {
12        {% if tls_active==true %}
13          "id": "https tcp check",
14          "notes": "HTTPS port opened",
15          "tcp": "localhost:{{ port_https }}",
16        {% else %}
17          "id": "admin status",
18          "notes": "Status admin : /admin/v1/status",
19          "http": "http://localhost:{{ port_http }}/admin/v1/status",
20        {% endif %}
21        "interval": "1s"
22      }
23    {% if vitam_component == 'worker' %}
24      ,
25      {
26        "id": "siegfried check",
27        "notes": "Is siegfried running ?",
28        "tcp": "localhost:{{ vitam_siegfried_port }}",
29        "interval": "1s"
30      }
31    {% elif vitam_component == 'ingest-external' %}
32      ,
33      {
34        "id": "siegfried check",
35        "notes": "Is siegfried running ?",
36        "tcp": "localhost:{{ vitam_siegfried_port }}",
37        "interval": "1s"
38      }
39    {% if installation_clamav == "true" %}
40      ,
41      {
42        "id": "ClamAV check",
43        "notes": "Is ClamAV daemon running ?",

```

```
44     "tcp": "localhost:{{ clamav_port }}",
45     "interval": "1s"
46   }
47 {% endif %}
48 {% endif %}
49   ]
50   }
51 }
```

6.3.4.3 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`

Avertissement : 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 visionner 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

logrotate / purge des logs si occupation importante ?

- cas des batches

N/A

6.3.5 elasticsearch chaîne de log

6.3.5.1 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.

6.3.5.2 Configuration / fichiers utiles

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

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

6.3.5.2.1 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}, file, syslog
logger:
  # log action execution errors for easier debugging
  action: INFO

  # 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: "${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"

```

6.3.5.2.2 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: {{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
{% else %}
network.host: {{inventory_hostname}}
{% endif %}#
# Set a custom port for HTTP:
#

```

```
http.port: {{port_http}}
network.port: {{port_tcp}}
transport.tcp.port: {{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: [ "{{ groups['hosts-elasticsearch-log']|join(' ', "
↪') }}" ]
#
# 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-log']|length == 1 %}
index.number_of_replicas: 0
{% else %}
index.number_of_replicas: 1
{% endif %}#

index.refresh_interval: 10s
indices.memory.index_buffer_size: 30%
index.translog.flush_threshold_ops: 50000
refresh_interval_in_millis: 30000
threadpool.search.queue_size: 12000
```

6.3.5.2.3 Fichier sysconfig/elasticsearch

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

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

# Elasticsearch configuration directory
CONF_DIR=/vitam/conf/elasticsearch-{{groupe}}

# Elasticsearch data directory
#DATA_DIR=/vitam/data/elasticsearch-{{groupe}}

# Elasticsearch logs directory
#LOG_DIR=/vitam/log/elasticsearch-{{groupe}}

# Elasticsearch PID directory
#PID_DIR=/var/run/elasticsearch-{{groupe}}

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

# Heap new generation
#ES_HEAP_NEWSIZE=

# Maximum direct memory
#ES_DIRECT_SIZE=

# Additional Java OPTS
ES_JAVA_OPTS="-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/log/elasticsearch-{{groupe}}/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=vitamdb
ES_GROUP=vitam

# 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

```

6.3.5.2.4 Fichier /usr/lib/tmpfiles.d/elasticsearch-data.conf

```
d    /var/run/elasticsearch-{{groupe}}    0755 vitamdb vitam - -
```

6.3.5.3 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 ¹¹

- actions récurrentes

logrotate / purge des logs si occupation importante ?

- cas des batches

N/A

6.3.6 elasticsearch Vitam

6.3.6.1 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.

6.3.6.2 Configuration / fichiers utiles

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

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

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}, file, syslog
logger:
  # log action execution errors for easier debugging
  action: INFO

  # 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
```

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

```

# 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: "{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"

```

6.3.6.2 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: {{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
{% else %}
network.host: {{ansible_hostname}}
{% endif %}
#
# Set a custom port for HTTP:
#
http.port: {{port_http}}
network.port: {{port_tcp}}
transport.tcp.port: {{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: [ "{{ groups['hosts-elasticsearch-data']|join(' ', "
↪') }}" ]
#
# 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
threadpool.search.queue_size: 12000
```

6.3.6.2.3 Fichier sysconfig/elasticsearch

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

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

# Elasticsearch configuration directory
CONF_DIR=/vitam/conf/elasticsearch-{{groupe}}

# Elasticsearch data directory
#DATA_DIR=/vitam/data/elasticsearch-{{groupe}}

# Elasticsearch logs directory
#LOG_DIR=/vitam/log/elasticsearch-{{groupe}}

# Elasticsearch PID directory
#PID_DIR=/var/run/elasticsearch-{{groupe}}

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

# Heap new generation
#ES_HEAP_NEWSIZE=

# Maximum direct memory
#ES_DIRECT_SIZE=
```

```

# Additional Java OPTS
ES_JAVA_OPTS="-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/log/elasticsearch-{{groupe}}/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=vitamdb
ES_GROUP=vitam

# 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

```

6.3.6.2.4 Fichier `/usr/lib/tmpfiles.d/elasticsearch-data.conf`

```
d    /var/run/elasticsearch-{{groupe}}    0755 vitamdb vitam - -
```

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-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

logrotate / purge des logs si occupation importante ?

- cas des batches

N/A

6.3.7 functional administration

6.3.7.1 Présentation

Rôle :

- Gérer les référentiels métier de la plate-forme

Fonctions :

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

6.3.7.2 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`.

6.3.7.2.1 Fichier `functional-administration.conf`

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

```
# Configuration MongoDB
mongoDbNodes:
{% for server in groups['mongos'] %}
- dbHost: {{server}}
  dbPort: {{vitam_mongodb_port}}
{% endfor %}
dbName: masterdata
dbAuthentication: {{mongo_authentication}}
dbUserName: {{mongoFunctionalAdminUser}}
dbPassword: {{mongoFunctionalAdminPassword}}

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

# Elasticsearch
clusterName: {{cluster_name}}
elasticsearchNodes:
{% for server in groups['hosts-elasticsearch-data'] %}
- hostName: {{server}}
  tcpPort: {{elasticsearch_data_tcp_port}}
{% endfor %}
```

6.3.7.2.2 Fichier `logbook-client.conf`

```
serverHost: {{vitam_logbook_host}}
serverPort: {{vitam_logbook_port}}
```

6.3.7.3 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>/ihm-demo/v1/status

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

logrotate / purge des logs si occupation importante ?

- cas des batches

N/A

6.3.8 ihm-demo

6.3.8.1 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

6.3.8.2 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/ihm-demo`.

6.3.8.2.1 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_component }}.p12
      keyPassword: {{keystores.client_external.ihm_demo}}
  truststore :
    - keyPath: {{vitam_folder_conf}}/truststore_{{ vitam_component }}.jks
      keyPassword: {{truststores.client_external}}
hostnameVerification: true
```

6.3.8.2.2 Fichier `ihm-demo.conf`

```
serverHost: {{vitam_ihm_demo_host}}
port: {{vitam_ihm_demo_port}}

baseUrl: "/ihm-demo"
staticContent: {{vitam_ihm_demo_static_content}}

jettyConfig: jetty-config.xml
```

```
tenants: [ "{{ vitam_tenant_ids | join(' ', '') }}" ]
secure: true
sipDirectory: {{ vitam_folder_data }}
```

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

6.3.8.2.3 Fichier ingest-external-client.conf

```
serverHost: {{ vitam_ingestexternal_host }}
serverPort: {{ vitam_ingestexternal_port_https }}
secure: true
sslConfiguration :
  keystore :
    - keyPath: {{ vitam_folder_conf }}/keystore_{{ vitam_component }}.p12
      keyPassword: {{ keystores.client_external.ihm_demo }}
  truststore :
    - keyPath: {{ vitam_folder_conf }}/truststore_{{ vitam_component }}.jks
      keyPassword: {{ truststores.client_external }}
hostnameVerification: true
```

6.3.8.2.4 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')}}
{% 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/messages/logbook = anon
/v1/api/tenants/ = anon
/** = authc
```

6.3.8.3 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

logrotate / purge des logs si occupation importante ?

- cas des batches

N/A

6.3.9 ihm-recette

6.3.9.1 Présentation

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

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

6.3.9.2 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-recette`.

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_component }}.p12
      keyPassword: {{keystores.client_external.ihm_recette}}
  truststore :
    - keyPath: {{vitam_folder_conf}}/truststore_{{ vitam_component }}.jks
      keyPassword: {{truststores.client_external}}
hostnameVerification: false
```


6.3.9.2.2 Fichier ihm-recette.conf

```
serverHost: {{vitam_ihm_recette_host}}
port: {{vitam_ihm_recette_port}}

baseUrl: "{{ vitam_ihm_recette_baseurl }}"
staticContent: "{{ vitam_ihm_recette_static_content }}"

jettyConfig: jetty-config.xml
tenants: [ "{{ vitam_tenant_ids | join(' ', ' ') }}" ]
secure: true
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: {{server}}
  dbPort: {{vitam_mongodb_port}}
{% endfor %}
# Actually need this field for compatibility
dbName: admin
# @integ: parametrize it !
masterdataDbName: masterdata
logbookDbName: logbook
metadataDbName: metadata
dbAuthentication: {{mongo_authentication}}
dbUserName: {{mongoAdminUser}}
dbPassword: {{mongoAdminPassword}}

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

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_component }}.p12
      keyPassword: {{keystores.client_external.ihm_recette}}
  truststore :
    - keyPath: {{vitam_folder_conf}}/truststore_{{ vitam_component }}.jks
      keyPassword: {{truststores.client_external}}
hostnameVerification: false
```

6.3.9.2.4 Fichier functional-administration-client.conf

```
serverHost: {{vitam_functional_administration_host}}
serverPort: {{vitam_functional_administration_port}}
```

6.3.9.2.5 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')}}
{% 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
/** = authc
```

6.3.9.2.6 Fichier soapui.conf

```
ingestProtocol: https
ingestPort: {{vitam_ingestexternal_port_https}}
ingestHost: {{vitam_ingestexternal_host}}

accessExternalProtocol: https
accessExternalPort: {{vitam_accessexternal_port_https}}
accessExternalHost: {{vitam_accessexternal_host}}

certfile: {{vitam_folder_conf}}/keystore_{{ vitam_component }}.p12
certPass: {{ keystores.client_external.ihm_recette }}
trustFile: {{vitam_folder_conf}}/truststore_{{ vitam_component }}.jks
trustPass: {{ truststores.client_external }}

dataDir: {{vitam_folder_data}}/
reportingDir: {{vitam_folder_app}}/
configDir: {{vitam_folder_data}}/
soapUiExecutable: {{ soapui_folder_app }}/bin/testrunner.sh
```

6.3.9.3 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/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

logrotate / purge des logs si occupation importante ?

- cas des batches

N/A

6.3.10 ingest-external

6.3.10.1 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

6.3.10.2 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/ingest-external`.

6.3.10.2.1 Fichier `ingest-external.conf`

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

6.3.10.2.2 Fichier ingest-internal-client.conf

```
serverHost: {{vitam_ingestinternal_host}}
serverPort: {{vitam_ingestinternal_port}}
```

6.3.10.2.3 Fichier scan-clamav.sh

```
#!/bin/sh

#####
# Role:
# Scan un single file using clamav anti-virus
#####
# Args:
# - file to scan
#####
# Return:
# - 0: scan OK - no virus
RET_NOTVIRUS=0
# - 1: virus found and corrected
RET_VIRUS_FOUND_FIXED=1
# - 2: virus found but not corrected
RET_VIRUS_FOUND_NOTFIXED=2
# - 3: Fatal scan not performed
RET_FAILURE=3
# stdout : names of virus found (1 per line) if virus found ;
#          failure description if failure
# stderr : full output of clamav
#####

# Default return code : scan NOK
RET=3
OUTPUT_DIR=$(mktemp -d)
if [ $# -ne 1 ]; then # Argument number must be one
    echo "ERROR : $# parameter(s) provided, only one parameter is needed"
else # one argument, let's go
    if [ ! -f "$1" ]; then # if the file wich will be scan is existing, keep going
        echo "ERROR : \"$1\" doesn't exist"
    else
        clamscan -z --config-file=/etc/clamd.d/scan.conf "$1" 1> ${OUTPUT_
→DIR}/stdout 2> ${OUTPUT_DIR}/stderr # scanning the file and store the output OUTPUT
RET=$? # return code of clamscan

        # Always output clamscan outputs to our own stderr
        (>&2 cat ${OUTPUT_DIR}/stdout ${OUTPUT_DIR}/stderr)

        if [ ${RET} -eq ${RET_VIRUS_FOUND_FIXED} ] ; then
            RET=2 # if virus found clamscan return 1; the script must_
→return 2

            (>&1 cat ${OUTPUT_DIR}/stdout | grep `basename ${1}` | cut -
→d ' ' -f 2) # sending the list of virus to our own stdout
            elif [ ${RET} -eq 2 ] ; then
                RET=3 # if scan not performed clamscan return 2; the script_
→must return 3

                (>&1 cat ${OUTPUT_DIR}/stdout | grep `basename ${1}` | cut -
→d ' ' -f 2-) # sending the failure reason to our own stdout
```

```
        fi
        if [ -f "${OUTPUT_DIR}/stdout" ]
        then
            rm ${OUTPUT_DIR}/stdout
        fi
        if [ -f "${OUTPUT_DIR}/stderr" ]
        then
            rm ${OUTPUT_DIR}/stderr
        fi
    fi
fi
rmdir ${OUTPUT_DIR}
exit ${RET}
```

6.3.10.3 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/v1/status

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

logrotate / purge des logs si occupation importante ?

- cas des batches

N/A

6.3.11 ingest-internal

6.3.11.1 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

6.3.11.2 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-internal`.

6.3.11.2.1 Fichier `ingest-internal.conf`

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

6.3.11.3 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/v1/status`

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

logrotate / purge des logs si occupation importante ?

- cas des batches

N/A

6.3.12 log server

6.3.12.1 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.

6.3.12.2 Configuration / fichiers utiles

L'ansible se charge du paramétrage de ces composants.

6.3.12.3 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

6.3.13 logbook

6.3.13.1 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

6.3.13.2 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`.

6.3.13.2.1 Fichier `logbook.conf`

```
# Configuration MongoDB
mongoDbNodes:
{% for server in groups['mongos'] %}
- dbHost: {{server}}
  dbPort: {{vitam_mongodb_port}}
{% endfor %}
dbName: logbook
dbAuthentication: {{mongo_authentication}}
dbUserName: {{mongoLogbookUser}}
dbPassword: {{mongoLogbookPassword}}
jettyConfig: jetty-config.xml
tenants: [ "{{ vitam_tenant_ids | join(' ', ' ') }}" ]
p12LogbookPassword: {{keystores.timestamping.logbook}}
p12LogbookFile: keystore_logbook.p12
workspaceUrl: {{vitam_workspace_baseurl}}
```

```
# Elasticsearch
clusterName: {{cluster_name}}
elasticsearchNodes:
{% for server in groups['hosts-elasticsearch-data'] %}
- hostName: {{server}}
  tcpPort: {{elasticsearch_data_tcp_port}}
{% endfor %}
```

6.3.13.3 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/v1/status`

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

logrotate / purge des logs si occupation importante ?

- cas des batches

N/A

6.3.14 metadata

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

6.3.14.2 Configuration / fichiers utiles

6.3.14.2.1 Fichier `metadata.conf`

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

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

# Elasticsearch
clusterName: {{cluster_name}}
elasticsearchNodes:
{% for server in groups['hosts-elasticsearch-data'] %}
- hostName: {{server}}
  tcpPort: {{elasticsearch_data_tcp_port}}
{% endfor %}
```

6.3.14.3 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/v1/status

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

logrotate / purge des logs si occupation importante ?

- cas des batches

N/A

6.3.15 mongoC

6.3.15.1 Présentation

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

6.3.15.2 Configuration / fichiers utiles

Pour le moment, aucun fichier à paramétrer.

6.3.15.3 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

Dans cette version du système, la supervision du service peut être réalisée via mongoclient.

- Exports
- gestion de la capacité

N/A

- actions récurrentes

logrotate / purge des logs si occupation importante ?

- cas des batches

Cas de l'export tous les soirs/matins ?

6.3.16 mongoclient

6.3.16.1 Présentation

MongoClient est un outil permettant de gérer les bases de données MongoDB (v3.2 et supérieures).

les fonctionnalités principales sont :

1. Supervision des bases de données
2. Gestions des utilisateurs
3. Gestion graphiques des bases de données

C'est un outil sous license MIT.

Avertissement : Mongoclient est un outil dont l'usage doit être réservé aux administrateurs techniques de Vitam

Les sources sont disponibles _ici : <https://github.com/rsercano/mongoclient> .

6.3.16.2 Configuration / fichiers utiles

6.3.16.2.1 Paramétrage bases de données à monitorer

Pour connecter mongoclient à une base de données mongodb, voici la procédure à suivre sur l'IHM :

- Connect
- **Create new**
 - **Onglet Connection :**
 - Connection name : vitam
 - Hostname : <hostname d'un serveur hébergeant un mongos>
 - Port : 27017
 - DB Name : metadata (ou une autre base renseignée dans le vault)
 - **Onglet Authentication :**
 - Authentication type : Standard
 - User : Voir <mongoAdminUser> dans le vault
 - Password : Voir <mongoAdminPassword> dans le vault
 - Authentication DB : admin
- Save changes
- Sélectionner la connection

- Connect Now

Il est ensuite possible de voir les statistiques de mongodb Il est aussi possible de changer de base de données :

- More
 - Switch Database
 - Sélectionner une autre base
 - Connect now

6.3.16.2.2 Configuration

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/mongoclient`.

6.3.16.2.2.1 Fichier `mongoclient.conf`

```
# Mongo URL for storing Mongoclient configuration.
# example: MONGO_URL='mongodb://user:password@host:port/databasename'
MONGO_URL="{{ mongoclient_mongo_url }}"

# The root URL calling Mongoclient
# example: ROOT_URL='http://example.com'
ROOT_URL="{{ mongoclient_root_url }}"

# Port the application will listen on
# example: PORT='3000'
PORT="{{ mongoclient_port }}"
{% if mongoclient_mail_url is defined %}

# SMTP URL for email configuration.
# example: MAIL_URL='smtp://user:password@mailhost:port/'
MAIL_URL="{{ mongoclient_mail_url }}"
{% endif %}
{% if mongoclient_user is defined %}

#Enable authentication when true
MONGOCLIENT_AUTH=true
MONGOCLIENT_USERNAME="{{mongoclient_user}}"
MONGOCLIENT_PASSWORD="{{mongoclient_pwd}}"
{% endif %}
```

6.3.16.3 Opérations

- Démarrage du service

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

- Arrêt du service

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

- 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>/mongoclient/healthcheck

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

logrotate / purge des logs si occupation importante ?

- cas des batches

N/A

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

logrotate / purge des logs si occupation importante ?

- 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

logrotate / purge des logs si occupation importante ?

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

6.3.19.2.2 Fichier `version.conf`

```
PhysicalMaster
BinaryMaster
Dissemination
Thumbnail
TextContent
```

6.3.19.2.3 Fichier `storage-client.conf`

```
serverHost: {{vitam_storageengine_host}}
serverPort: {{vitam_storageengine_port}}
```

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/v1/status`

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

logrotate / purge des logs si occupation importante ?

- cas des batches

N/A

6.3.20 siegfried

6.3.20.1 Présentation

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

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*.

6.3.20.3 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

logrotate / purge des logs si occupation importante ?

- cas des batches

N/A

6.3.21 storage-engine

6.3.21.1 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

6.3.21.2 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*.

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

6.3.21.2.1 Fichier `driver-location.conf`

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

6.3.21.2.2 Fichier `driver-mapping.conf`

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

6.3.21.2.3 Fichier `static-offer.json`

```
[  
{% for item in groups['hosts-storage-offer-default'] %}  
{  
  "id" : "{{item}}",  
  "baseUrl" : "http://{{item}}:{{vitam_storageofferdefault_port_https}}",  
  "parameters" : {  
    "keyStore-keyPath": "{{vitam_folder_conf}}/keystore_storage.pl2",  
    "keyStore-keyPassword": "{{keystores.client_storage.storage}}",  
    "trustStore-keyPath": "{{vitam_folder_conf}}/truststore_storage.jks",  
    "trustStore-keyPassword": "{{truststores.client_storage}}"  
  }  
}  
}  
{% if not loop.last %},  
{% endif %}  
{% endfor %}  
]
```

6.3.21.2.4 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 %}
    ]
  }
}
```

6.3.21.2.5 Fichier storage-engine.conf

```
urlWorkspace: {{vitam_workspace_baseurl}}
timeoutMsPerKB: 100
jettyConfig: jetty-config.xml
tenants: [ "{{ vitam_tenant_ids | join(' ', ' ') }}" ]
```

6.3.21.3 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/v1/status

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

logrotate / purge des logs si occupation importante ?

- cas des batches

N/A

6.3.22 storage-offer-default

6.3.22.1 Présentation

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

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

6.3.22.2 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-offer-default`.

6.3.22.2.1 Fichier default-offer.conf

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

6.3.22.2.2 Fichier default-storage.conf

```
{% if vitam_provider_offer != "openstack-swift" %}storagePath: {{vitam_folder_data}}
{% endif %}
provider: {{vitam_provider_offer}}
keystoneEndPoint: {{vitam_keystone_auth_url}}
swiftUid: {{ vitam_swift_uid }}
swiftSubUser: {{vitam_swift_subuser }}
credential: {{ vitam_keystone_passwd }}
cephMode: {{ vitam_ceph_mode }}
```

6.3.22.3 Opérations

- Démarrage du service

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

- Arrêt du service

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

- 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-offer-default/v1/status

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

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

logrotate / purge des logs si occupation importante ?

- cas des batches

N/A

6.3.23 worker

6.3.23.1 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.

6.3.23.2 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/worker.

6.3.23.2.1 Fichier format-identifiers.conf

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

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

6.3.23.2.2 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 }}
```

6.3.23.2.3 Fichier logbook-client.conf

Ce fichier permet de définir l'accès au logbook.

```
serverHost: {{vitam_logbook_host}}
serverPort: {{vitam_logbook_port}}
```

6.3.23.2.4 Fichier metadata-client.conf

Ce fichier permet de définir l'accès au metadata.

```
serverHost: {{vitam_metadata_host}}
serverPort: {{vitam_metadata_port}}
```

6.3.23.2.5 Fichier storage-client.conf

Ce fichier permet de définir l'accès au storage.

```
serverHost: {{vitam_storageengine_host}}
serverPort: {{vitam_storageengine_port}}
```

6.3.23.2.6 Fichier version.conf

```
PhysicalMaster
BinaryMaster
Dissemination
Thumbnail
TextContent
```

6.3.23.2.7 Fichier worker.conf

Ce fichier permet de définir le paramétrage du composant worker.

```
# Configuration processing
# HERE MUST BE MY (WORKER) current configuration
registerServerHost: {{inventory_hostname}}
registerServerPort: {{vitam_worker_port}}
# Configuration handler
processingUrl: {{vitam_processing_baseurl}}
urlMetadata: {{vitam_metadata_baseurl}}
urlWorkspace: {{vitam_workspace_baseurl}}
# 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 compsoant 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 appartenant en fonction de tache exécutée **** capacity **** : capacité du worker en mode parallèle de tache (par défaut à 1 dans l'ansiblerie, si non définie)

6.3.23.3 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/v1/status

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

logrotate / purge des logs si occupation importante ?

- cas des batches

N/A

6.3.24 workspace

6.3.24.1 Présentation

Rôle :

- Fourniture d'un espace pour l'échange des 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)

6.3.24.2 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/workspace`.

6.3.24.2.1 Fichier workspace.conf

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

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/v1/status

- Exports

N/A

- gestion de la capacité

N/A

- actions récurrentes

logrotate / purge des logs si occupation importante ?

- cas des batches

N/A

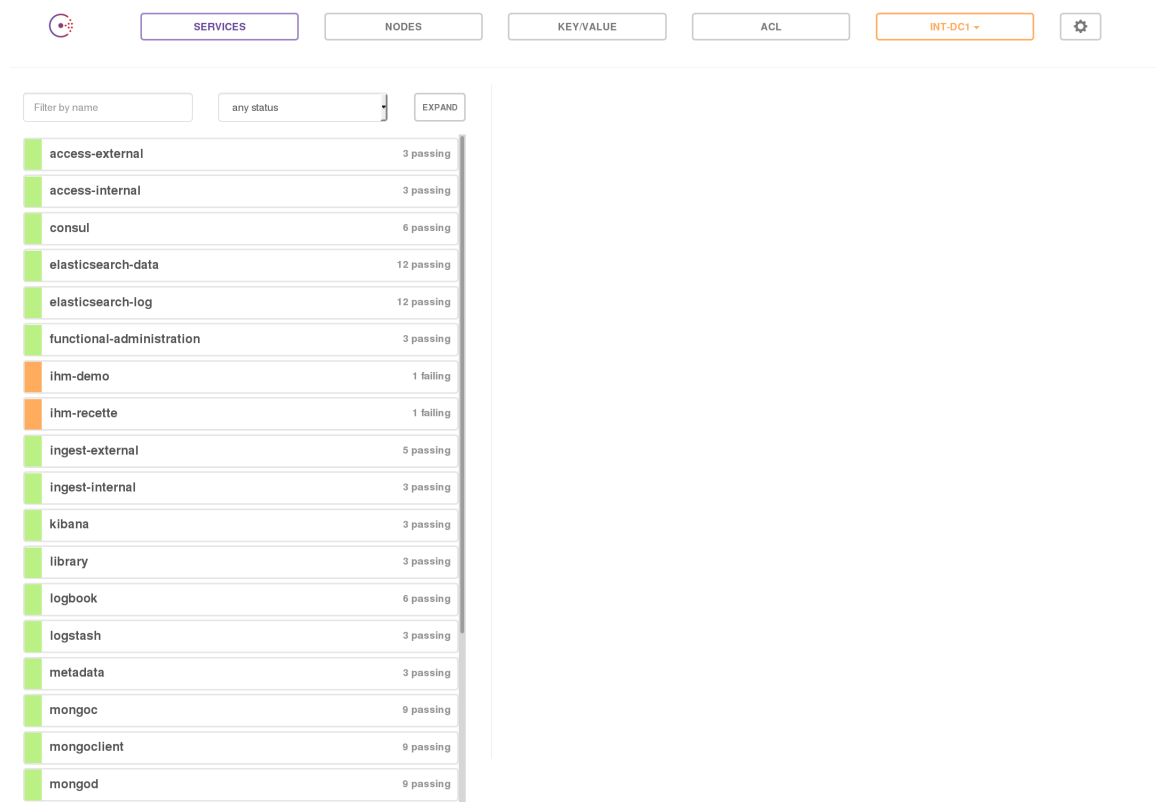
Aide à l'exploitation





















7.1 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 VITAM.

7.1.1 Etat Consul

Se connecter à l'IHM de Consul et recenser les états des composants 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 et ihm-recette (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.`

7.1.2 Etat VITAM via 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 das Kibana.

Questions Fréquemment Posées

8.1 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.

8.2 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 elasticsearch, puis du retour de ce noeud, certains shards d'elasticsearch peuvent rester dans l'état UNASSIGNED ; dans ce cas, le plugin Kopf 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](#) ¹².

Elasticsearch possède des shards non initialisés (état “INITIALIZING”) Tout d’abord, il peut être difficile d’identifier les shards en questions dans le plugin Kopf ; 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](#) ¹³). 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.

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

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

Contacts et support

L'équipe d'exploitation prend en charge les tickets des utilisateurs finaux et procède à la première analyse (support niveau 1). Les équipes du projet VITAM assurent le support niveau 2 et 3.

9.1 Contacts

En cas de problème, il convient d'ouvrir un ticket à l'URL suivante : <https://support.programmevitam.fr>

Dans ce ticket, il est nécessaire d'explicitier le contexte (comportement observé vs comportement attendu), tant fonctionnel que technique (copies écran et code d'erreur sont utiles). Si possible et applicable, fournir également le jeu de tests, déterminer le niveau de criticité du problème, son taux de reproduction et un lien avec l'US concerné, si applicable.

Avertissement : Le support ne prend en charge l'appel qu'à la fourniture d'un numéro de ticket valide.

Suite à l'ouverture du ticket, le programme VITAM étudie et qualifie le problème rencontré :

- Bloquant, qui empêche toute action de recette
- Majeur, qui bloque une action de recette mais permet d'en continuer d'autres
- Mineur, à corriger après la recette

Exploitation par composant

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

10.1 Access

10.2 Common

10.3 Functional administration

10.4 Ingest

10.5 Logbook

10.6 Metadata

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" : "http://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"
  }
}
```

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

10.10 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**

10.11 Workspace

Annexes

3.1	Vue d'ensemble d'un déploiement VITAM : zones, composants	9
-----	---	---

2.1	Documents de référence VITAM	5
4.1	Cinématique d'arrêt de VITAM	13
4.2	Cinématique de démarrage de VITAM	14

A

API, [6](#)

B

BDD, [6](#)

C

COTS, [6](#)

D

DAT, [6](#)

Deb, [6](#)

DEX, [6](#)

DIN, [6](#)

DNSSEC, [6](#)

I

IHM, [6](#)

J

JRE, [6](#)

JVM, [6](#)

M

MitM, [6](#)

N

NoSQL, [6](#)

O

OAIS, [7](#)

P

PDMA, [6](#)

PKI, [7](#)

R

RPM, [6](#)

S

SIA, [7](#)

T

TNR, [7](#)

V

VITAM, [6](#)