



VITAM - Documentation d'installation

Version 9.0.0

VITAM

oct. 29, 2025

Table des matières

1	Introduction	1
1.1	Objectif de ce document	1
2	Rappels	2
2.1	Information concernant les licences	2
2.2	Documents de référence	2
2.2.1	Documents internes	2
2.2.2	Référentiels externes	3
2.3	Glossaire	3
3	Prérequis à l'installation	6
3.1	Expertises requises	6
3.2	Pré-requis plate-forme	8
3.2.1	Base commune	8
3.2.2	PKI	9
3.2.3	Systèmes d'exploitation	9
3.2.3.1	Déploiement sur environnement AlmaLinux	10
3.2.3.2	Déploiement sur environnement Debian	10
3.2.3.3	Présence d'un agent antiviral	10
3.2.4	Matériel	11
3.2.5	Librairie de cartouches pour offre froide	11
3.3	Questions préparatoires	11
3.4	Récupération de la version	12
3.4.1	Utilisation des dépôts <i>open-source</i>	12
3.4.1.1	<i>Repository</i> pour environnement CentOS	12
3.4.1.1.1	Cas de <i>griffins</i>	12
3.4.1.2	<i>Repository</i> pour environnement Debian	13
3.4.1.2.1	Cas de <i>griffins</i>	13
3.4.2	Utilisation des packages de livraison	13
4	Procédures d'installation / mise à jour	14
4.1	Vérifications préalables	14
4.2	Procédures	14
4.2.1	Cinématique de déploiement	14
4.2.2	Cas particulier d'une installation multi-sites	15
4.2.2.1	Procédure d'installation	15
4.2.2.1.1	<i>vitam_site_name</i>	15

4.2.2.1.2	primary_site	15
4.2.2.1.3	consul_remote_sites	16
4.2.2.1.4	vitam_offers	16
4.2.2.1.5	vitam_strategy	17
4.2.2.1.6	other_strategies	18
4.2.2.1.7	plateforme_secret	19
4.2.2.1.8	consul_encrypt	20
4.2.2.2	Procédure de réinstallation	20
4.2.2.3	Flux entre Storage et Offer	20
4.2.2.3.1	Avant la génération des keystores	21
4.2.2.3.2	Après la génération des keystores	22
4.2.3	Configuration du déploiement	22
4.2.3.1	Fichiers de déploiement	22
4.2.3.2	Informations <i>plate-forme</i>	22
4.2.3.2.1	Inventaire	22
4.2.3.2.2	Fichier main.yml	32
4.2.3.2.3	Fichier vitam_security.yml	35
4.2.3.2.4	Fichier offers_opts.yml	36
4.2.3.2.5	Fichier cots_vars.yml	41
4.2.3.2.6	Fichier tenants_vars.yml	48
4.2.3.3	Déclaration des secrets	53
4.2.3.3.1	vitam	53
4.2.3.3.2	Cas des extras	57
4.2.3.3.3	Commande ansible-vault	58
4.2.3.3.3.1	Générer des fichiers <i>vaultés</i> depuis des fichier en clair	58
4.2.3.3.3.2	Re-chiffrer un fichier <i>vaulté</i> avec un nouveau mot de passe	58
4.2.3.4	La configuration d'ElasticSearch	58
4.2.3.5	Le mapping ElasticSearch pour Unit et ObjectGroup	58
4.2.4	Gestion des certificats	65
4.2.4.1	Cas 1 : Configuration développement / tests	65
4.2.4.1.1	Procédure générale	66
4.2.4.1.2	Génération des CA par les scripts Vitam	66
4.2.4.1.3	Génération des certificats par les scripts Vitam	66
4.2.4.2	Cas 2 : Configuration production	66
4.2.4.2.1	Procédure générale	66
4.2.4.2.2	Génération des certificats	67
4.2.4.2.2.1	Certificats serveurs	67
4.2.4.2.2.2	Certificat clients	67
4.2.4.2.2.3	Certificats d'horodatage	68
4.2.4.2.3	Intégration de certificats existants	68
4.2.4.2.4	Intégration de certificats clients de VITAM	69
4.2.4.2.4.1	Intégration d'une application externe (cliente)	69
4.2.4.2.4.2	Intégration d'un certificat personnel (<i>personae</i>)	69
4.2.4.2.5	Cas des offres objet	69
4.2.4.2.6	Absence d'usage d'un <i>reverse</i>	70
4.2.4.3	Intégration de CA pour une offre <i>Swift</i> ou <i>s3</i>	70
4.2.4.4	Génération des magasins de certificats	70
4.2.5	Paramétrages supplémentaires	70
4.2.5.1	<i>Tuning</i> JVM	70
4.2.5.2	Installation en mode conteneur	71
4.2.5.3	Installation des <i>griffins</i> (greffons de préservation)	71
4.2.5.4	Rétention liée aux logback	72
4.2.5.4.1	Cas des accesslog	72
4.2.5.5	Paramétrage de l'antivirus (ingest-external)	72

4.2.5.5.1	Extra : Avast Business Antivirus for Linux	73
4.2.5.6	Paramétrage des certificats externes (*-externe)	74
4.2.5.7	Placer « hors Vitam » le composant ihm-demo	74
4.2.5.8	Paramétrer le <code>secure_cookie</code> pour ihm-demo	74
4.2.5.9	Paramétrage de la centralisation des logs VITAM	75
4.2.5.9.1	Gestion par VITAM	75
4.2.5.9.2	Redirection des logs sur un SIEM tiers	75
4.2.5.10	Passage des identifiants des référentiels en mode <i>esclave</i>	75
4.2.5.11	Paramétrage du batch de calcul pour l'indexation des règles héritées	76
4.2.5.12	Durées minimales permettant de contrôler les valeurs saisies	76
4.2.5.13	Augmenter la précision sur le nombre de résultats retournés dépassant 10000	77
4.2.5.14	Fichiers complémentaires	78
4.2.5.15	Paramétrage de l'Offre Froide (librairies de cartouches)	101
4.2.5.16	Sécurisation SELinux	105
4.2.5.17	Installation de la stack Prometheus	106
4.2.5.17.1	Playbooks ansible	107
4.2.5.18	Installation de Grafana	107
4.2.5.18.1	Configuration	108
4.2.5.18.2	Configuration spécifique derrière un proxy	108
4.2.5.19	Installation de restic	108
4.2.5.19.1	Configuration	108
4.2.5.19.2	Limitations actuelles	108
4.2.6	Procédure de première installation	109
4.2.6.1	Déploiement	109
4.2.6.1.1	Cas particulier : utilisation de ClamAv en environnement Debian	109
4.2.6.1.2	Fichier de mot de passe des vaults ansible	109
4.2.6.1.3	Mise en place des repositories VITAM (optionnel)	109
4.2.6.1.4	Génération des <code>hostvars</code>	110
4.2.6.1.4.1	Cas 1 : Machines avec une seule interface réseau	110
4.2.6.1.4.2	Cas 2 : Machines avec plusieurs interfaces réseau	110
4.2.6.1.4.3	Vérification de la génération des hostvars	111
4.2.6.1.5	Tests d'infrastructure	111
4.2.6.1.6	Déploiement	111
4.2.7	Éléments <i>extras</i> de l'installation	112
4.2.7.1	Configuration des <i>extras</i>	112
4.2.7.2	Déploiement des <i>extras</i>	114
4.2.7.2.1	ihm-recette	114
4.2.7.2.2	<i>Extras</i> complet	114
5	Procédures de mise à jour de la configuration	116
5.1	Cas d'une modification du nombre de tenants	116
5.2	Cas d'une modification des paramètres JVM	117
5.3	Cas de la mise à jour des <i>griffins</i>	117
6	Post installation	118
6.1	Validation du déploiement	118
6.1.1	Sécurisation du fichier <code>vault_pass.txt</code>	118
6.1.2	Validation manuelle	118
6.1.3	Validation via Consul	118
6.1.4	Post-installation : administration fonctionnelle	119
6.2	Sauvegarde des éléments d'installation	119
6.3	Troubleshooting	119
6.3.1	Erreur au chargement des <i>index template</i> kibana	119
6.3.2	Erreur au chargement des tableaux de bord Kibana	120

6.4	Retour d'expérience / cas rencontrés	120
6.4.1	Crash rsyslog, code killed, signal : BUS	120
6.4.2	Mongo-express ne se connecte pas à la base de données associée	120
6.4.3	Elasticsearch possède des shard non alloués (état « UNASSIGNED »)	120
6.4.4	Elasticsearch possède des shards non initialisés (état « INITIALIZING »)	121
6.4.5	Elasticsearch est dans l'état « <i>read-only</i> »	121
6.4.6	MongoDB semble lent	122
6.4.7	Les shards de MongoDB semblent mal équilibrés	122
6.4.8	L'importation initiale (profil de sécurité, certificats) retourne une erreur	123
6.4.9	Problème d'ingest et/ou d'accès	123
7	Montée de version	124
8	Annexes	125
8.1	Vue d'ensemble de la gestion des certificats	125
8.1.1	Liste des suites cryptographiques & protocoles supportés par VITAM	125
8.1.2	Vue d'ensemble de la gestion des certificats	126
8.1.3	Description de l'arborescence de la PKI	126
8.1.4	Description de l'arborescence du répertoire <code>deployment/environments/certs</code>	128
8.1.5	Description de l'arborescence du répertoire <code>deployment/environments/keystores</code>	129
8.1.6	Fonctionnement des scripts de la PKI	129
8.2	Spécificités des certificats	129
8.2.1	Cas des certificats serveur	130
8.2.1.1	Généralités	130
8.2.1.2	Noms DNS des serveurs https VITAM	130
8.2.2	Cas des certificats client	131
8.2.3	Cas des certificats d'horodatage	131
8.2.4	Cas des certificats des services de stockage objets	131
8.3	Cycle de vie des certificats	131
8.4	Ansible & SSH	133
8.4.1	Authentification du compte utilisateur utilisé pour la connexion SSH	133
8.4.1.1	Par clé SSH avec passphrase	133
8.4.1.2	Par login/mot de passe	133
8.4.1.3	Par clé SSH sans passphrase	133
8.4.2	Authentification des hôtes	133
8.4.3	Elévation de priviléges	133
8.4.3.1	Par sudo avec mot de passe	134
8.4.3.2	Par su	134
8.4.3.3	Par sudo sans mot de passe	134
8.4.3.4	Déjà Root	134
Index		137

CHAPITRE 1

Introduction

1.1 Objectif de ce document

Ce document a pour but de fournir à une équipe d'exploitants de la solution logicielle *VITAM* les procédures et informations utiles et nécessaires pour l'installation de la solution logicielle.

Il s'adresse aux personnes suivantes :

- Les architectes techniques des projets désirant intégrer la solution logicielle *VITAM* ;
- Les exploitants devant installer la solution logicielle *VITAM*.

CHAPITRE 2

Rappels

2.1 Information concernant les licences

La solution logicielle *VITAM* est publiée sous la licence CeCILL 2.1¹; la documentation associée (comprenant le présent document) est publiée sous Licence Ouverte V2.0².

Les clients externes java de solution *VITAM* sont publiés sous la licence CeCILL-C³; la documentation associée (comprenant le présent document) est publiée sous Licence Ouverte V2.0⁴.

2.2 Documents de référence

2.2.1 Documents internes

TABLEAU 1 – Documents de référence VITAM

Nom	Lien
<i>DAT</i>	http://www.programmevitam.fr/ressources/DocCourante/html/archi
<i>DIN</i>	http://www.programmevitam.fr/ressources/DocCourante/html/installation
<i>DEX</i>	http://www.programmevitam.fr/ressources/DocCourante/html/exploitation
<i>DMV</i>	http://www.programmevitam.fr/ressources/DocCourante/html/migration
Release notes	https://github.com/ProgrammeVitam/vitam/releases/latest

1. https://cecill.info/licences/Licence_CeCILL_V2.1-fr.html
2. <https://www.etalab.gouv.fr/wp-content/uploads/2017/04/ETALAB-Licence-Ouverte-v2.0.pdf>
3. https://cecill.info/licences/Licence_CeCILL-C_V1-fr.html
4. <https://www.etalab.gouv.fr/wp-content/uploads/2017/04/ETALAB-Licence-Ouverte-v2.0.pdf>

2.2.2 Référentiels externes

2.3 Glossaire

API *Application Programming Interface*

AU *Archive Unit*, unité archivistique

BDD Base De Données

BDO *Binary DataObject*

CA *Certificate Authority*, autorité de certification

CAS Content Adressable Storage

CCFN Composant Coffre Fort Numérique

CN Common Name

COTS Component Off The shelf; 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.

CRL *Certificate Revocation List*; liste des identifiants des certificats qui ont été révoqués ou invalidés et qui ne sont donc plus dignes de confiance. Cette norme est spécifiée dans les RFC 5280 et RFC 6818.

CRUD *create, read, update, and delete*, s'applique aux opérations dans une base de données MongoDB

DAT Dossier d'Architecture Technique

DC Data Center

DEX Dossier d'EXploitation

DIN Dossier d'INstallation

DIP *Dissemination Information Package*

DMV Documentation de Montées de Version

DNS *Domain Name System*

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 DNSSEC⁵](#)

DSL *Domain Specific Language*, langage dédié pour le requêtage de VITAM

DUA Durée d'Utilité Administrative

E BIOS Méthode d'évaluation des risques en informatique, permettant d'apprécier les risques Sécurité des systèmes d'information (entités et vulnérabilités, méthodes d'attaques et éléments menaçants, éléments essentiels et besoins de sécurité...), de contribuer à leur traitement en spécifiant les exigences de sécurité à mettre en place, de préparer l'ensemble du dossier de sécurité nécessaire à l'acceptation des risques et de fournir les éléments utiles à la communication relative aux risques. Elle est compatible avec les normes ISO 13335 (GMITS), ISO 15408 (critères communs) et ISO 17799

EAD Description archivistique encodée

ELK Suite logicielle *Elasticsearch Logstash Kibana*

FIP *Floating IP*

GOT Groupe d'Objet Technique

IHM Interface Homme Machine

IP *Internet Protocol*

IsaDG Norme générale et internationale de description archivistique

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

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

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

LAN Local Area Network, réseau informatique local, qui relie des ordinateurs dans une zone limitée

LFC LiFe Cycle, cycle de vie

LTS Long-term support, support à long terme : version spécifique d'un logiciel dont le support est assuré pour une période de temps plus longue que la normale.

M2M Machine To Machine

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. [Explication⁶](#)

MoReq Modular Requirements for Records System, recueil d'exigences pour l'organisation de l'archivage, élaboré dans le cadre de l'Union européenne.

NoSQL Base de données non-basée sur un paradigme classique des bases relationnelles. [Définition NoSQL⁷](#)

NTP Network Time Protocol

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.

OOM Aussi appelé *Out-Of-Memory Killer*; mécanisme de la dernière chance incorporé au noyau Linux, en cas de dépassement de la capacité mémoire

OS Operating System, système d'exploitation

OWASP Open Web Application Security Project, communauté en ligne de façon libre et ouverte à tous publiant des recommandations de sécurisation Web et de proposant aux internautes, administrateurs et entreprises des méthodes et outils de référence permettant de contrôler le niveau de sécurisation de ses applications Web

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

PKI Une infrastructure à clés publiques (ICP) ou infrastructure de gestion de clés (IGC) ou encore Public Key Infrastructure (PKI), est un ensemble de composants physiques (des ordinateurs, des équipements cryptographiques logiciels ou matériel type HSM ou encore des cartes à puces), de procédures humaines (vérifications, validation) et de logiciels (système et application) en vue de gérer le cycle de vie des certificats numériques ou certificats électroniques. [Définition PKI⁸](#)

PCA Plan de Continuité d'Activité

PRA Plan de Reprise d'Activité

REST REpresentational State Transfer : type d'architecture d'échanges. Appliqué aux services web, en se basant sur les appels http standard, il permet de fournir des API dites « RESTful » qui présentent un certain nombre d'avantages en termes d'indépendance, d'universalité, de maintenabilité et de gestion de charge. [Définition REST⁹](#)

RGAA Référentiel Général d'Accessibilité pour les Administrations

RGI Référentiel Général d'Interopérabilité

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

SAE Système d'Archivage Électronique

SEDA Standard d'Échange de Données pour l'Archivage

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

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

8. https://fr.wikipedia.org/wiki/Infrastructure_%C3%A0_cl%C3%A9s_publiques

9. https://fr.wikipedia.org/wiki/Representational_state_transfer

SGBD Système de Gestion de Base de Données

SGBDR Système de Gestion de Base de Données Relationnelle

SIA Système d'Informations Archivistique

SIEM *Security Information and Event Management*

SIP *Submission Information Package*

SSH *Secure SHell*

Swift *OpenStack Object Store project*

TLS *Transport Layer Security*

TNA *The National Archives, Pronom*¹⁰

TNR Tests de Non-Régression

TTL *Time To Live*, indique le temps pendant lequel une information doit être conservée, ou le temps pendant lequel une information doit être gardée en cache

UDP *User Datagram Protocol*, protocole de datagramme utilisateur, un des principaux protocoles de télécommunication utilisés par Internet. Il fait partie de la couche transport du modèle OSI

UID *User IDentification*

VITAM Valeurs Immatérielles Transférées aux Archives pour Mémoire

VM *Virtual Machine*

WAF *Web Application Firewall*

WAN *Wide Area Network*, réseau informatique couvrant une grande zone géographique, typiquement à l'échelle d'un pays, d'un continent, ou de la planète entière

10. <https://www.nationalarchives.gov.uk/PRONOM/>

CHAPITRE 3

Prérequis à l'installation

3.1 Expertises requises

Les équipes en charge du déploiement et de l'exploitation de la solution logicielle *VITAM* devront disposer en interne des compétences suivantes :

TABLEAU 1 – Matrice de compétences

Thème	Outil	Description de l'outil	Niveau requis	Niveau de criticité	Exemples de compétences requises
Système	Linux (AlmaLinux 9 ou Debian 12)	Système d'exploitation	3/4 : maîtrise	3/4 : Majeur	Etre à l'aise avec l'arborescence linux / Configurer une interface réseau / Analyse avancée des logs systèmes et réseaux
Configuration	- Git	Suivi des modifications quotidiennes des sources de déploiement VITAM	1/4 : débutant	1/4 : Mineur	Savoir exécuter les commandes de bases (commit, pull, push, etc...)
Configuration	- Git	Adaptation des sources de déploiement VITAM dans le cadre d'une montée de version	2/4 : intermédiaire	1/4 : Mineur	Savoir exécuter les commandes intermédiaires (branche, merge, etc...)
Configuration	Ansible	Gestion de configuration et déploiement automatisé	3/4 : maîtrise	3/4 : Majeur	Adapter les paramètres pour permettre une installation spécifique / Comprendre l'arborescence des rôles et des playbooks
Exploitation	Consul	Outil d'enregistrement des services VITAM	1/4 : débutant	4/4 : critique	Contrôler l'état des services via l'interface consul Eteindre et redémarrer un Consul Agent sur une machine virtuelle
Supervision	Kibana	Interface de visualisation du contenu des bases Elasticsearch	1/4 : débutant	2/4 : significatif	Créer un nouveau dashboard avec des indicateurs spécifiques / Lire et relever les données pertinentes dans un dashboard donné
Supervision	Cerebro	Interface de contrôle des clusters Elasticsearch	1/4 : débutant	2/4 : significatif	Contrôler l'état des clusters Elasticsearch via l'interface cerebro
Base de données	MongoDB	Base de données NoSQL	2/4 : intermédiaire	4/4 : critique	Effectuer une recherche au sein d'une base MongoDB / Sauvegarder et restaurer une base MongoDB (data ou offer) / Augmenter la capacité de stockage d'une base MongoDB
Base de données	Elasticsearch	Moteur de recherche et d'indexation de données distribué	2/4 : intermédiaire	4/4 : critique	Sauvegarder et restaurer une base Elasticsearch (data ou log) / Augmenter la capacité de stockage d'une base Elasticsearch / Effectuer une procédure de maintenance d'un nœud au sein d'un cluster Elasticsearch
Appliquacatif	Applications Java	Composants logiciels Vitam	2/4 : intermédiaire	4/4 : critique	Appeler le point "v1/status" manuellement sur tous les composants VITAM / Arrêter et relancer selectivement les composants VITAM à l'aide d'Ansible (ordre important) / Lancer une procédure d'indisponibilité de VITAM (fermeture des services external, arrêt des timers)

3.1 Expertises requises

- Niveau requis : Qualifie le niveau de compétence attendue par l'exploitant de la solution logicielle Vitam.
- Niveau de criticité : Qualifie le degré d'importance pour le bon fonctionnement de la plateforme.

3.2 Pré-requis plate-forme

Les pré-requis suivants sont nécessaires :

3.2.1 Base commune

- Tous les serveurs hébergeant la solution logicielle *VITAM* doivent être synchronisés sur un serveur de temps (protocole *NTP*, pas de *stratum* 10)
- Disposer de la solution de déploiement basée sur ansible

Le déploiement est orchestré depuis un poste ou serveur d'administration ; les pré-requis suivants doivent y être présents :

- packages nécessaires :
 - **ansible** (version **2.9** minimale et conseillée; se référer à la documentation [ansible](#)¹¹ pour la procédure d'installation)
 - **openssh-client** (client SSH utilisé par ansible)
 - **JRE OpenJDK 11** et **openssl** (du fait de la génération de certificats / *stores*, l'utilitaire *keytool* est nécessaire)
- un accès ssh vers un utilisateur d'administration avec élévation de priviléges vers les droits *root*, *vitam*, *vitamdb* (les comptes *vitam* et *vitamdb* sont créés durant le déploiement) sur les serveurs cibles.
- Le compte utilisé sur le serveur d'administration doit avoir confiance dans les serveurs sur lesquels la solution logicielle *VITAM* doit être installée (fichier `~/.ssh/known_hosts` correctement renseigné)

Note : Se référer à la documentation d'[usage](#)¹² pour les procédures de connexion aux machines-cibles depuis le serveur ansible.

Prudence : Les adresses *IP* des machines sur lesquelles la solution logicielle *VITAM* sera installée ne doivent pas changer d'adresse IP au cours du temps. En cas de changement d'adresse IP, la plateforme ne pourra plus fonctionner.

Prudence : Aucune version pré-installée de la JRE OpenJDK ne doit être présente sur les machines cibles où sera installé *VITAM*.

Prudence : La solution *VITAM* ne tolère qu'une très courte désynchronisation de temps entre les machines (par défaut, 10 secondes). La configuration NTP doit être finement monitorée. Idéalement une synchronisation doit être planifiée chaque 5/10 minutes.

11. http://docs.ansible.com/ansible/latest/intro_installation.html

12. http://docs.ansible.com/ansible/latest/intro_getting_started.html

Prudence : Dans le cadre de l'installation des packages « extra », il est nécessaire, pour les partitions hébergeant des conteneurs docker (mongo-express, head), qu'elles aient un accès internet (installation du paquet officiel docker, récupération des images).

Prudence : Dans le cadre de l'installation des packages « extra », il est nécessaire, pour les partitions hébergeant le composant `ihm-recette`, qu'elles aient un accès internet (installation du *repository* et installation du *package git-lfs*; récupération des *TNR* depuis un dépôt git).

Avertissement : Dans le cas d'une installation du composant `vitam-offer` en `filesystem-hash`, il est fortement recommandé d'employer un système de fichiers `xfs` pour le stockage des données. Se référer au [DAT](#) pour connaître la structuration des *filesystems* dans la solution logicielle [VITAM](#). En cas d'utilisation d'un autre type, s'assurer que le filesystem possède/gère bien l'option `user_xattr`.

Avertissement : Dans le cas d'une installation du composant `vitam-offer` en `tape-library`, il est fortement recommandé d'installer au préalable sur les machines cible associées les paquets pour les commandes `mt`, `mtx` et `dd`. Ces composants doivent également apporter le groupe système `tape`. Se reporter également à [prerequisoffrefroide](#).

3.2.2 PKI

La solution logicielle [VITAM](#) nécessite des certificats pour son bon fonctionnement (cf. [DAT](#) pour la liste des secrets et [Vue d'ensemble de la gestion des certificats](#) (page 125) pour une vue d'ensemble de leur usage.) La gestion de ces certificats, par le biais d'une ou plusieurs [PKI](#), est à charge de l'équipe d'exploitation. La mise à disposition des certificats et des chaînes de validation [CA](#), placés dans les répertoires de déploiement adéquats, est un pré-requis à tout déploiement en production de la solution logicielle [VITAM](#).

Voir aussi :

Veuillez vous référer à la section [Vue d'ensemble de la gestion des certificats](#) (page 125) pour la liste des certificats nécessaires au déploiement de la solution VITAM, ainsi que pour leurs répertoires de déploiement.

3.2.3 Systèmes d'exploitation

Seules deux distributions Linux suivantes sont supportées à ce jour :

- AlmaLinux 9
- Debian 12 (bookworm)

SELinux doit être configuré en mode `permissive` ou `disabled`. Toutefois depuis la release R13, la solution logicielle [VITAM](#) prend désormais en charge l'activation de SELinux sur le périmètre du composant worker et des processus associés aux *griffins* (greffons de préservation).

Note : En cas de changement de mode SELinux, redémarrer les machines pour la bonne prise en compte de la modification avant de lancer le déploiement.

Prudence : En cas d'installation initiale, les utilisateurs et groupes systèmes (noms et *UID*) utilisés par VITAM (et listés dans le *DAT*) ne doivent pas être présents sur les serveurs cible. Ces comptes sont créés lors de l'installation de VITAM et gérés par VITAM.

3.2.3.1 Déploiement sur environnement AlmaLinux

- Disposer d'une plate-forme Linux AlmaLinux 9 installée selon la répartition des services souhaités. En particulier, ces serveurs doivent avoir :
 - une configuration de temps synchronisée (ex : en récupérant le temps à un serveur centralisé)
 - Des autorisations de flux conformément aux besoins décrits dans le *DAT*
 - une configuration des serveurs de noms correcte (cette configuration sera surchargée lors de l'installation)
 - un accès à un dépôt (ou son miroir) AlmaLinux 9 (base et extras) et EPEL 9
- Disposer des binaires VITAM : paquets *RPM* de VITAM (vitam-product) ainsi que les paquets d'éditeurs tiers livrés avec VITAM (vitam-external)
- Disposer, si besoin, des binaires pour l'installation des *griffins*

3.2.3.2 Déploiement sur environnement Debian

- Disposer d'une plate-forme Linux Debian « bookworm » installée selon la répartition des services souhaitée. En particulier, ces serveurs doivent avoir :
 - une configuration de temps synchronisée (ex : en récupérant le temps à un serveur centralisé)
 - Des autorisations de flux conformément aux besoins décrits dans le *DAT*
 - une configuration des serveurs de noms correcte (cette configuration sera surchargée lors de l'installation)
 - un accès à un dépôt (ou son miroir) Debian (base et extras) et bookworm-backports
 - un accès internet, car le dépôt docker sera ajouté
- Disposer des binaires VITAM : paquets deb de VITAM (vitam-product) ainsi que les paquets d'éditeurs tiers livrés avec VITAM (vitam-external)
- Disposer, si besoin, des binaires pour l'installation des *griffins*

Avertissement : Pour l'installation des *packages* mongoDB, il est nécessaire de mettre à disposition le *package* libcurl3 présent en stretch uniquement (le *package* libcurl4 sera désinstallé).

Avertissement : Le *package* curl est installé depuis les dépôts stretch.

3.2.3.3 Présence d'un agent antivirus

Dans le cas de partitions sur lesquelles un agent antivirus est déjà configuré (typiquement, *golden image*), il est recommandé de positionner une exception sur l'arborescence /vitam et les sous-arborescences, hormis la partition hébergeant le composant ingest-external (emploi d'un agent antivirus en prérequis des *ingest*; se reporter à *Paramétrage de l'antivirus (ingest-external)* (page 72)).

3.2.4 Matériel

Les prérequis matériel sont définis dans le [DAT](#); à l'heure actuelle, le minimum recommandé pour la solution Vitam est 2 CPUs. Il également est recommandé de prévoir (paramétrage par défaut à l'installation) 512Mo de RAM disponible par composant applicatif [VITAM](#) installé sur chaque machine (hors elasticsearch et mongo).

Concernant l'espace disque, à l'heure actuelle, aucun pré-requis n'a été défini; cependant, sont à prévoir par la suite des espaces de stockage conséquents pour les composants suivants :

- offer
- solution de centralisation des logs (*cluster* elasticsearch de log)
- workspace
- worker (temporairement, lors du traitement de chaque fichier à traiter)
- *cluster* elasticsearch et mongodb des données [VITAM](#)

L'arborescence associée sur les partitions associées est : /vitam/data/<composant>

3.2.5 Librairie de cartouches pour offre froide

Des prérequis sont à réunir pour utiliser l'offre froide de stockage « tape-library » définie dans le [DAT](#).

- La librairie de cartouches doit être opérationnelle et chargée en cartouches.
- La librairie et les lecteurs doivent déjà être configurés sur la machine devant supporter une instance de ce composant. La commande `lsscsi -g` peut permettre de vérifier si des périphériques sont détectés.
- Le dossier `/vitam/data/offer/` doit correspondre à une seule partition de système de fichiers (i.e. tout le contenu du dossier `/vitam/data/offer` doit appartenir au même point de montage). Le système de fichiers doit supporter les opérations de atomiques (type atomic rename / move) et la création de liens symboliques (ex. XFS, EXT4...)

3.3 Questions préparatoires

La solution logicielle [VITAM](#) permet de répondre à différents besoins.

Afin d'y répondre de la façon la plus adéquate et afin de configurer correctement le déploiement [VITAM](#), il est nécessaire de se poser en amont les questions suivantes :

- **Questions techniques :**

- Topologie de déploiement et dimensionnement de l'environnement ?
- Espace de stockage (volumétrie métier cible, technologies d'offres de stockage, nombre d'offres, etc.) ?
- Sécurisation des flux http (réécriture des clés publiques des services versants, sécurisation des flux d'accès aux offres, etc.) ?

- **Questions liées au métier :**

- Nombre de tenants souhaités (hormis les tenant 0 et 1 qui font respectivement office de tenant « blanc » et de tenant d'administration) ?
- Niveau de classification (la plate-forme est-elle « Secret Défense » ?)
- Modalités d'indexation des règles de gestion des unités archivistiques (autrement dit, sur quels tenant le recalculation des inheritedRules doit-il être fait complètement / partiellement) ?
- Greffons de préservations (*griffins*) nécessaires ?
- Fréquence de calcul de l'état des fonds symboliques souhaitée ?
- Définition des habilitations (profil de sécurité, contextes applicatifs, ...) ?

- Modalités de gestion des données de référence (maître/esclave) pour chaque tenant ?

Par la suite, les réponses apportées vous permettront de configurer le déploiement par la définition des paramètres ansible.

3.4 Récupération de la version

3.4.1 Utilisation des dépôts *open-source*

Les scripts de déploiement de la solution logicielle **VITAM** sont disponibles dans le dépôt github **VITAM**¹³, dans le répertoire `deployment`.

Les binaires de la solution logicielle **VITAM** sont disponibles sur des dépôts **VITAM** publics indiqués ci-dessous par type de *package*; ces dépôts doivent être correctement configurés sur la plate-forme cible avant toute installation.

3.4.1.1 *Repository* pour environnement CentOS

Sur les partitions cibles, configurer le fichier `/etc/yum.repos.d/vitam-repositories.repo` (remplacer `<branche_vitam>` par le nom de la branche de support à installer) comme suit

```
[programmavitam-vitam-rpm-release-product]
name=programmavitam-vitam-rpm-release-product
baseurl=http://download.programmavitam.fr/vitam_repository/<vitam_version>/rpm/vitam-
    ↪product/
gpgcheck=0
repo_gpgcheck=0
enabled=1

[programmavitam-vitam-rpm-release-external]
name=programmavitam-vitam-rpm-release-external
baseurl=http://download.programmavitam.fr/vitam_repository/<vitam_version>/rpm/vitam-
    ↪external/
gpgcheck=0
repo_gpgcheck=0
enabled=1
```

Note : remplacer `<vitam_version>` par la version à déployer.

3.4.1.1.1 Cas de *griffins*

Un dépôt supplémentaire est à paramétriser pour pouvoir dérouler l'installation des *griffins*

```
[programmavitam-vitam-griffins]
name=programmavitam-vitam-griffins
baseurl=http://download.programmavitam.fr/vitam_griffins/<version_griffins>/rpm/
gpgcheck=0
repo_gpgcheck=0
enabled=1
```

13. <https://github.com/ProgrammeVitam/vitam>

Note : remplacer <version_griffins> par la version à déployer.

3.4.1.2 Repository pour environnement Debian

Sur les partitions cibles, configurer le fichier /etc/apt/sources.list.d/vitam-repositories.list comme suit

```
deb [trusted=yes] http://download.programmavitam.fr/vitam_repository/<vitam_version>/  
↳deb/vitam_product/ ./  
deb [trusted=yes] http://download.programmavitam.fr/vitam_repository/<vitam_version>/  
↳deb/vitam-external/ ./
```

Note : remplacer <vitam_version> par la version à déployer.

3.4.1.2.1 Cas de *griffins*

Un dépôt supplémentaire est à paramétrer pour pouvoir dérouler l'installation des *griffins*

```
deb [trusted=yes] http://download.programmavitam.fr/vitam_griffins/<version_griffins>/  
↳deb/ ./
```

Note : remplacer <version_griffins> par la version à déployer.

3.4.2 Utilisation des packages de livraison

Le package global -FULL d'installation contient les livrables binaires (dépôts AlmaLinux, Debian, Maven & Documentation)

Sur la machine « ansible » dédiée au déploiement de la solution logicielle *VITAM*, décompresser le package -DEPLOYMENT (au format tar.gz).

Sur le repository « VITAM », récupérer également depuis le fichier d'extension tar.gz les binaires d'installation (-RPM pour AlmaLinux ; -DEB pour Debian) et les faire prendre en compte par le repository.

Sur le repository « *griffins* », récupérer également depuis le fichier d'extension tar.gz les binaires d'installation (-RPM pour AlmaLinux ; -DEB pour Debian) et les faire prendre en compte par le repository.

CHAPITRE 4

Procédures d'installation / mise à jour

4.1 Vérifications préalables

Tous les serveurs cibles doivent avoir accès aux dépôts de binaires contenant les paquets de la solution logicielle *VITAM* et des composants externes requis pour l'installation. Les autres éléments d'installation (playbook ansible, ...) doivent être disponibles sur la machine ansible orchestrant le déploiement de la solution.

4.2 Procédures

4.2.1 Cinématique de déploiement

La cinématique de déploiement d'un site *VITAM* est représentée dans le schéma suivant :

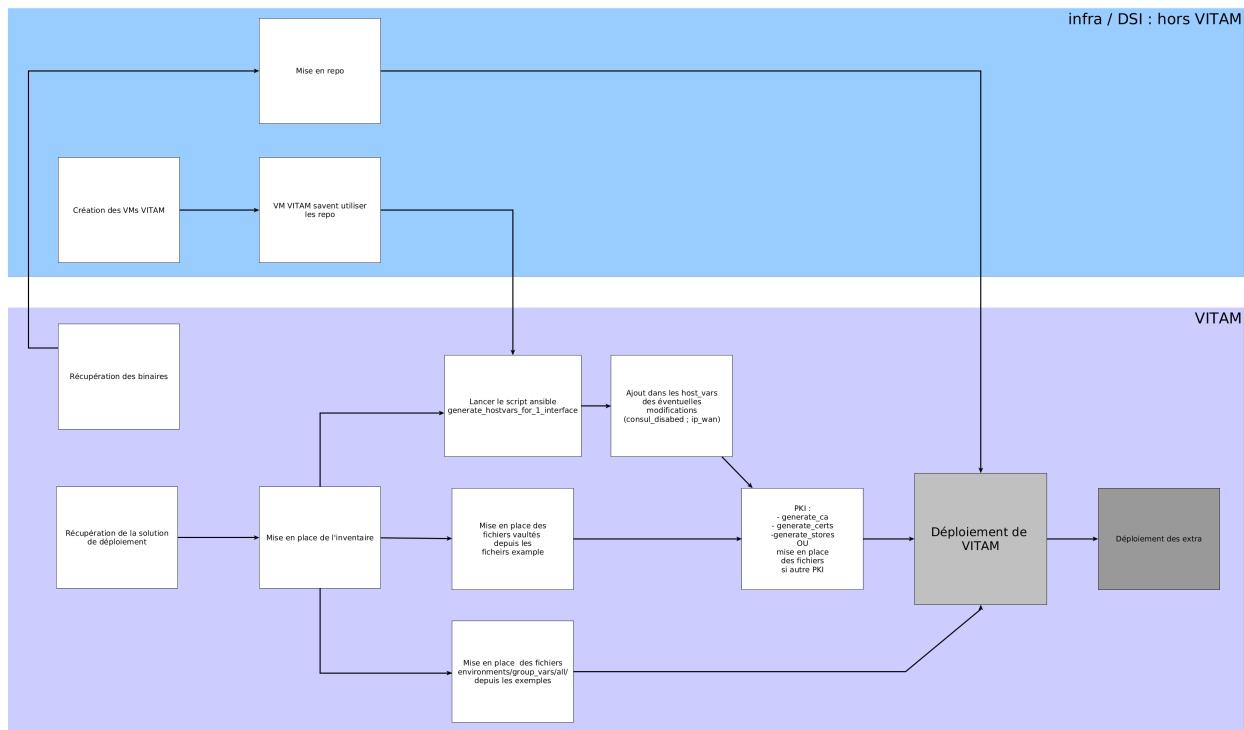


FIG. 1 – Cinématique de déploiement

4.2.2 Cas particulier d'une installation multi-sites

4.2.2.1 Procédure d'installation

Dans le cadre d'une installation multi-sites, il est nécessaire de déployer la solution logicielle **VITAM** sur le site secondaire dans un premier temps, puis déployer le site *production*.

Il faut paramétriser correctement un certain nombre de variables ansible pour chaque site :

4.2.2.1.1 vitam_site_name

Fichier : deployment/environments/hosts.<environnement>

Cette variable sert à définir le nom du site. Elle doit être différente sur chaque site.

4.2.2.1.2 primary_site

Fichier : deployment/environments/hosts.<environnement>

Cette variable sert à définir si le site est primaire ou non. Sur VITAM installé en mode multi site, un seul des sites doit avoir la valeur *primary_site* à true. Sur les sites secondaires (*primary_site* : false), certains composants ne seront pas démarrés et apparaîtront donc en orange sur l'**IHM** de consul. Certains timers systemd seront en revanche démarrés pour mettre en place la reconstruction au fil de l'eau, par exemple.

4.2.2.1.3 consul_remote_sites

Fichier : deployment/environments/group_vars/all/main.yml

Cette variable sert à référencer la liste des *Consul Server* des sites distants, à celui que l'on configure.

Exemple de configuration pour une installation avec 3 sites.

Site 1 :

```
consul_remote_sites:
  - dc2:
      wan: ["dc2-host-1", "dc2-host-2", "dc2-host-3"]
  - dc3:
      wan: ["dc3-host-1", "dc3-host-2", "dc3-host-3"]
```

Site 2 :

```
consul_remote_sites:
  - dc1:
      wan: ["dc1-host-1", "dc1-host-2", "dc1-host-3"]
  - dc3:
      wan: ["dc3-host-1", "dc3-host-2", "dc3-host-3"]
```

Site 3 :

```
consul_remote_sites:
  - dc1:
      wan: ["dc1-host-1", "dc1-host-2", "dc1-host-3"]
  - dc2:
      wan: ["dc2-host-1", "dc2-host-2", "dc2-host-3"]
```

Il faut également prévoir de déclarer, lors de l'installation de chaque site distant, la variable `ip_wan` pour les partitions hébergeant les serveurs Consul (groupe `ansible hosts_consul_server`) et les offres de stockage (groupe `ansible hosts_storage_offer_default`, considérées distantes par le site primaire). Ces ajouts sont à faire dans `environments/host_vars/<nom partition>`.

Exemple :

```
ip_service : 172.17.0.10 ip_admin : 172.19.0.10 ip_wan : 10.2.64.3
```

Ainsi, à l'usage, le composant `storage` va appeler les services `offer`. Si le service est « hors domaine » (déclaration explicite `<service>.<datacenter distant>.service.<domaine consul>`), un échange d'information entre « datacenters » Consul est réalisé et la valeur de `ip_wan` est fournie pour l'appel au service distant.

4.2.2.1.4 vitam_offers

Fichier : deployment/environments/group_vars/all/offer_opts.yml

Cette variable référence toutes les offres disponibles sur la totalité des sites VITAM. Sur les sites secondaires, il suffit de référencer les offres disponible localement.

Exemple :

```
vitam_offers:
  offer-fs-1:
    provider: filesystem-hash
  offer-fs-2:
    provider: filesystem-hash
```

(suite sur la page suivante)

(suite de la page précédente)

```
offer-fs-3:
  provider: filesystem-hash
```

4.2.2.1.5 vitam_strategy

Fichier : deployment/environments/group_vars/all/offer_opts.yml

Cette variable référence la stratégie de stockage de plateforme *default* sur le site courant.

Si l'offre se situe sur un site distant, il est nécessaire de préciser le nom du site, via la variable *vitam_site_name*, sur lequel elle se trouve comme dans l'exemple ci-dessous.

Il est fortement conseillé de prendre comme offre référente une des offres locale au site. Les sites secondaires doivent uniquement écrire sur leur(s) offre(s) locale(s).

Exemple pour le site 1 (site primaire) :

```
vitam_strategy:
  - name: offer-fs-1
    referent: true
    rank: 0
  - name: offer-fs-2
    referent: false
    distant: true
    vitam_site_name: site2
    rank: 1
  - name: offer-fs-3
    referent: false
    distant: true
    vitam_site_name: site3
    rank: 2
# Optional params for each offers in vitam_strategy. If not set, the default values
# are applied.
#   referent: false           # true / false (default), only one per site must be
#                             referent
#   status: ACTIVE            # ACTIVE (default) / INACTIVE
#   vitam_site_name: distant-dc2 # default is the value of vitam_site_name defined
#                             in your local inventory file, should be specified with the vitam_site_name defined
#                             for the distant offer
#   distant: false            # true / false (default). If set to true, it will
#                             not check if the provider for this offer is correctly set
#   id: idoffre               # OPTIONAL, but IF ACTIVATED, MUST BE UNIQUE & SAME
#                             if on another site
#   asyncRead: false           # true / false (default). Should be set to true for
#                             tape offer only
#   rank: 0                   # Integer that indicates in ascending order the
#                             priority of the offer in the strategy
```

Exemple pour le site 2 (site secondaire) :

```
vitam_strategy:
  - name: offer-fs-2
    referent: true
```

Exemple pour le site 3 (site secondaire) :

```
vitam_strategy:
  - name: offer-fs-3
    referent: true
```

4.2.2.1.6 other_strategies

Fichier : deployment/environments/group_vars/all/offer_opts.yml

Cette variable référence les stratégies de stockage additionnelles sur le site courant. **Elles ne sont déclarées et utilisées que dans le cas du multi-stratégies.** Si l'offre se situe sur un site distant, il est nécessaire de préciser le nom du site sur lequel elle se trouve comme dans l'exemple ci-dessous. Les sites secondaires doivent uniquement écrire sur leur(s) offre(s) locale(s).

Les offres correspondant à l'exemple other_strategies sont les suivantes :

```
vitam_offers:
  offer-fs-1:
    provider: filesystem-hash
  offer-fs-2:
    provider: filesystem-hash
  offer-fs-3:
    provider: filesystem-hash
  offer-s3-1:
    provider: amazon-s3-v2
  offer-s3-2:
    provider: amazon-s3-v2
  offer-s3-3:
    provider: amazon-s3-v2
```

Exemple pour le site 1 (site primaire) :

```
other_strategies:
  metadata:
    - name: offer-fs-1
      referent: true
      rank: 0
    - name: offer-fs-2
      referent: false
      distant: true
      vitam_site_name: site2
      rank: 1
    - name: offer-fs-3
      referent: false
      distant: true
      vitam_site_name: site3
      rank: 2
    - name: offer-s3-1
      referent: false
      rank: 3
    - name: offer-s3-2
      referent: false
      distant: true
      vitam_site_name: site2
      rank: 4
    - name: offer-s3-3
      referent: false
```

(suite sur la page suivante)

(suite de la page précédente)

```

distant: true
vitam_site_name: site3
rank: 5
binary:
  - name: offer-s3-1
    referent: false
    rank: 0
  - name: offer-s3-2
    referent: false
    distant: true
    vitam_site_name: site2
    rank: 1
  - name: offer-s3-3
    referent: false
    distant: true
    vitam_site_name: site3
    rank: 2

```

Exemple pour le site 2 (site secondaire) :

```

other_strategies:
  metadata:
    - name: offer-fs-2
      referent: true
      rank: 0
    - name: offer-s3-2
      referent: false
      rank: 1
  binary:
    - name: offer-s3-2
      referent: false
      rank: 0

```

Exemple pour le site 3 (site secondaire) :

```

other_strategies:
  metadata:
    - name: offer-fs-3
      referent: true
      rank: 0
    - name: offer-s3-3
      referent: false
      rank: 1
  binary:
    - name: offer-s3-3
      referent: false
      rank: 0

```

4.2.2.1.7 plateforme_secret

Fichier : deployment/environments/group_vars/all/main/vault-vitam.yml

Cette variable stocke le *secret de plateforme* qui doit être commun à tous les composants de la solution logicielle **VITAM** de tous les sites. La valeur doit donc être identique pour chaque site.

4.2.2.1.8 consul_encrypt

Fichier : deployment/environments/group_vars/all/main/vault-vitam.yml

Cette variable stocke le *secret de plateforme* qui doit être commun à tous les *Consul* de tous les sites. La valeur doit donc être identique pour chaque site.

4.2.2.2 Procédure de réinstallation

En prérequis, il est nécessaire d'attendre que tous les *workflows* et reconstructions (sites secondaires) en cours soient terminés.

Ensuite :

- Arrêter vitam sur le site primaire.
- Arrêter les sites secondaires.
- Redéployer vitam sur les sites secondaires.
- Redéployer vitam sur le site primaire

4.2.2.3 Flux entre Storage et Offer

Dans le cas **d'appel en https entre les composants Storage et Offer**, il faut modifier deployment/environments/group_vars/all/advanced/vitam_vars.yml et indiquer https_enabled: true dans storageofferdefault.

Il convient également également d'ajouter :

- **Sur le site primaire**
 - Dans le truststore de Storage : la *CA* ayant signé le certificat de l'Offer du site secondaire
- **Sur le site secondaire**
 - Dans le truststore de Offer : la *CA* ayant signé le certificat du Storage du site primaire
 - Dans le grantedstore de Offer : le certificat du storage du site primaire

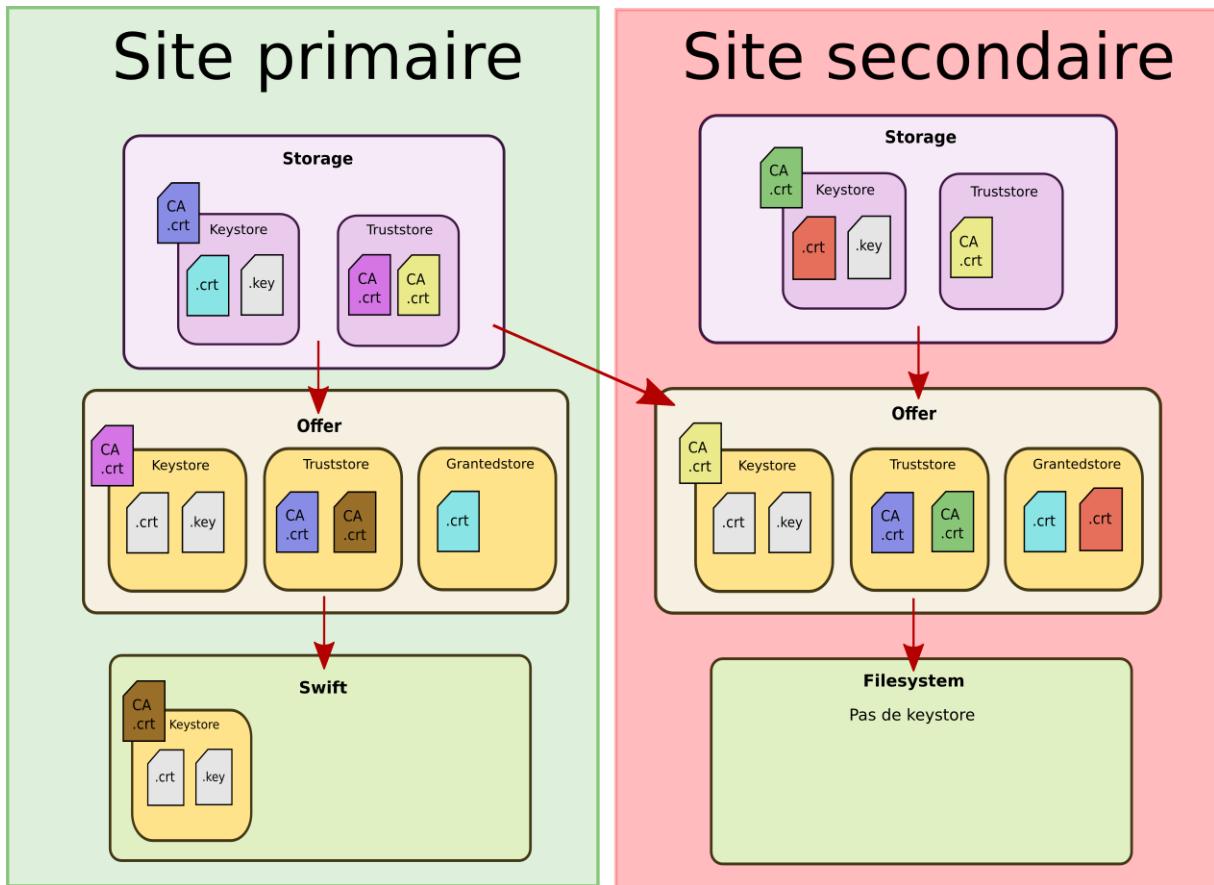


FIG. 2 – Vue détaillée des certificats entre le storage et l'offre en multi-site

Il est possible de procéder de 2 manières différentes :

4.2.2.3.1 Avant la génération des keystores

Avertissement : Pour toutes les copies de certificats indiquées ci-dessous, il est important de ne jamais les écraser, il faut donc renommer les fichiers si nécessaire.

Déposer les **CA** du client storage du site 1 environments/certs/client-storage/ca/* dans le client storage du site 2 environments/certs/client-storage/ca/.

Déposer le certificat du client storage du site 1 environments/certs/client-storage/clients/storage/*.crt dans le client storage du site 2 environments/certs/client-storage/clients/storage/.

Déposer les **CA** du serveur offer du site 2 environments/certs/server/ca/* dans le répertoire des **CA** serveur du site 1 environments/certs/server/ca/

4.2.2.3.2 Après la génération des keystores

Via le script deployment/generate_stores.sh, il convient donc d'ajouter les *CA* et certificats indiqués sur le schéma ci-dessus.

Ajout d'un certificat : keytool -import -keystore -file <certificat.crt> -alias <alias_certificat>

Ajout d'une *CA* : keytool -import -trustcacerts -keystore -file <ca.crt> -alias <alias_certificat>

4.2.3 Configuration du déploiement

Voir aussi :

L'architecture de la solution logicielle, les éléments de dimensionnement ainsi que les principes de déploiement sont définis dans le *DAT*.

4.2.3.1 Fichiers de déploiement

Les fichiers de déploiement sont disponibles dans la version *VITAM* livrée, dans le sous-répertoire deployment/. Concernant l'installation, ils se déclinent en 2 parties :

- les playbooks ansible de déploiement, présents dans le sous-répertoire ansible-vitam/, qui est indépendant de l'environnement à déployer ; ces fichiers ne sont normalement pas à modifier pour réaliser une installation.
- l'arborescence d'inventaire ; des fichiers d'exemples sont disponibles dans le sous-répertoire environments/. Cette arborescence est valable pour le déploiement d'un environnement, et doit être dupliquée lors de l'installation d'environnements ultérieurs. Les fichiers contenus dans cette arborescence doivent être adaptés avant le déploiement, comme expliqué dans les paragraphes suivants.

4.2.3.2 Informations plate-forme

4.2.3.2.1 Inventaire

Pour configurer le déploiement, il est nécessaire de créer, dans le répertoire environments/, un nouveau fichier d'inventaire (par la suite, ce fichier sera communément appelé hosts.<environnement>). Ce fichier devra se conformer à la structure présente dans le fichier hosts.example (et notamment respecter scrupuleusement l'arborescence des groupes *ansible*). Les commentaires dans ce fichier fournissent les explications permettant l'adaptation à l'environnement cible :

```
1 # Group definition ; DO NOT MODIFY
2 [hosts]
3
4 # Group definition ; DO NOT MODIFY
5 [hosts:children]
6 vitam
7 reverse
8 hosts_dev_tools
9 ldap
10 hosts_vitamui
11 ###### Tests environments specifics #####
12
13
```

(suite sur la page suivante)

(suite de la page précédente)

```

14 # EXTRA : Front reverse-proxy (test environments ONLY) ; add machine name after
15 [reverse]
16 # optional : if this machine is different from VITAM machines, you can specify ↵
17 # another become user
18 # Example
19 # vm-reverse.vitam ansible_ssh_user=reverse_user
20
21 [ldap] # Extra : OpenLDAP server
22 # LDAP server !!! NOT FOR PRODUCTION !!! Test only
23
24
25 [library]
26 # TODO: Put here servers where this service will be deployed : library
27
28
29 [hosts_dev_tools]
30 # TODO: Put here servers where this service will be deployed : mongo-express, ↵
31 # elasticsearch-head
32 # /!\ WARNING !!! NOT FOR PRODUCTION
33
34 [elasticsearch:children] # EXTRA : elasticsearch
35 hosts_elasticsearch_data
36 hosts_elasticsearch_log
37
38 ##### VITAM services #####
39
40 # Group definition ; DO NOT MODIFY
41 [vitam:children]
42 zone_external
43 zone_access
44 zone_applicative
45 zone_storage
46 zone_data
47 zone_admin
48 library
49
50 ##### Zone externe
51 [zone_external:children]
52 hosts_ihm_demo
53 hosts_ihm_recette
54
55 [hosts_ihm_demo]
56 # TODO: Put here servers where this service will be deployed : ihm-demo. If you use ↵
57 # vitam-ui or your own frontend, it is recommended to leave this group blank
58 # If you don't need consul for ihm-demo, you can set this var after each hostname :
59 # consul_disabled=true
60 # DEPRECATED / We'll soon be removed. Please consider using vitam-ui or your own ↵
61 # front-end
62 # /!\ WARNING !!! NOT recommended for PRODUCTION
63
64 [hosts_ihm_recette]
65 # TODO: Put here servers where this service will be deployed : ihm-recette (extra ↵
66 # feature)
67 # DEPRECATED / We'll soon be removed.

```

(suite sur la page suivante)

(suite de la page précédente)

```
66 # /!\ WARNING !!! NOT FOR PRODUCTION
67
68
69 ##### Zone access
70
71 # Group definition ; DO NOT MODIFY
72 [zone_access:children]
73 hosts_ingest_external
74 hosts_access_external
75 hosts_collect_external
76
77 [hosts_ingest_external]
78 # TODO: Put here servers where this service will be deployed : ingest-external
79
80
81 [hosts_access_external]
82 # TODO: Put here servers where this service will be deployed : access-external
83
84
85 [hosts_collect_external]
86 # TODO: Put here servers where this service will be deployed : collect-external
87
88
89 ##### Zone applicative
90
91 # Group definition ; DO NOT MODIFY
92 [zone_applicative:children]
93 hosts_ingest_internal
94 hosts_processing
95 hosts_batch_report
96 hosts_worker
97 hosts_access_internal
98 hosts_metadata
99 hosts_functional_administration
100 hosts_scheduler
101 hosts_logbook
102 hosts_workspace
103 hosts_storage_engine
104 hosts_security_internal
105 hosts_collect_internal
106 hosts_metadata_collect
107 hosts_workspace_collect
108
109
110 [hosts_security_internal]
111 # TODO: Put here servers where this service will be deployed : security-internal
112
113
114 [hosts_logbook]
115 # TODO: Put here servers where this service will be deployed : logbook
116
117
118 [hosts_workspace]
119 # TODO: Put the server where this service will be deployed : workspace
120 # WARNING: put only ONE server for this service, not more !
121
122
```

(suite sur la page suivante)

(suite de la page précédente)

```

123 [hosts_ingest_internal]
124 # TODO: Put here servers where this service will be deployed : ingest-internal
125
126
127 [hosts_access_internal]
128 # TODO: Put here servers where this service will be deployed : access-internal
129
130
131 [hosts_metadata]
132 # TODO: Put here servers where this service will be deployed : metadata
133
134
135 [hosts_functional_administration]
136 # TODO: Put here servers where this service will be deployed : functional-
137 →administration
138
139 [hosts_scheduler]
140 # TODO: Put here servers where this service will be deployed : scheduler
141 # Optional parameter after each host : vitam_scheduler_thread_count=<integer> ; This_
142 →is the number of threads that are available for concurrent execution of jobs. ;_
143 →default is 3 thread
144
145 [hosts_processing]
146 # TODO: Put the server where this service will be deployed : processing
147 # WARNING: put only one server for this service, not more !
148
149 [hosts_storage_engine]
150 # TODO: Put here servers where this service will be deployed : storage-engine
151
152
153 [hosts_batch_report]
154 # TODO: Put here servers where this service will be deployed : batch-report
155
156
157 [hosts_worker]
158 # TODO: Put here servers where this service will be deployed : worker
159 # Optional parameter after each host : vitam_worker_capacity=<integer> ; please refer_
160 →to your infrastructure for defining this number ; default is ansible_processor_
161 →vcpus value (cpu number in /proc/cpuinfo file)
162
163 [hosts_collect_internal]
164 # TODO: Put here servers where this service will be deployed : collect_internal
165
166 [hosts_metadata_collect]
167 # TODO: Put here servers where this service will be deployed : metadata_collect
168
169
170 [hosts_workspace_collect]
171 # TODO: Put the server where this service will be deployed : workspace_collect
172 # WARNING: put only ONE server for this service, not more !
173
174

```

(suite sur la page suivante)

(suite de la page précédente)

```

175
176 ##### Zone storage
177
178 [zone_storage:children] # DO NOT MODIFY
179 hosts_storage_offer_default
180 hosts_mongodb_offer
181
182 [hosts_storage_offer_default]
183 # TODO: Put here servers where this service will be deployed : storage-offer-default
184 # LIMIT : only 1 offer per machine
185 # LIMIT and 1 machine per offer when filesystem or filesystem-hash provider
186 # Possibility to declare multiple machines with same provider only when provider is ↳
187 ↳ s3 or swift.
188 # Mandatory param for each offer is offer_conf and points to offer_opts.yml & vault-
189 ↳ vitam.yml (with same tree)
190 # Optional parameter: restic_enabled=true (only 1 per offer_conf) available for ↳
191 ↳ providers filesystem*, openstack-swift-v3 & amazon-s3-v2
192 # for swift
193 # hostname-offre-1.vitam offer_conf=offer-swift-1 restic_enabled=true
194 # hostname-offre-2.vitam offer_conf=offer-swift-1
195 # for filesystem
196 # hostname-offre-2.vitam offer_conf=offer-fs-1 restic_enabled=true
197 # for s3
198 # hostname-offre-3.vitam offer_conf=offer-s3-1 restic_enabled=true
199 # hostname-offre-4.vitam offer_conf=offer-s3-1
200
201
202 [hosts_mongodb_offer:children]
203 hosts_mongos_offer
204 hosts_mongoc_offer
205 hosts_mongod_offer
206
207 [hosts_mongos_offer]
208 # WARNING : DO NOT COLLOCATE WITH [hosts_mongos_data]
209 # TODO: put here servers where this service will be deployed : mongos cluster for ↳
210 ↳ storage offers
211 # Mandatory params
212 # - mongo_cluster_name=<offer_name> ; name of the cluster (should exist on vitam_
213 ↳ strategy configuration in offer_opts.yml)
214 # The recommended practice is to install the mongos instance on the same servers as ↳
215 ↳ the mongoc instances
216 # Example
217 # vitam-mongo-swift-offer-01 mongo_cluster_name=offer-swift-1
218 # vitam-mongo-swift-offer-02 mongo_cluster_name=offer-swift-1
219 # vitam-mongo-fs-offer-01 mongo_cluster_name=offer-fs-1
220 # vitam-mongo-fs-offer-02 mongo_cluster_name=offer-fs-1
221 # vitam-mongo-s3-offer-01 mongo_cluster_name=offer-s3-1
222 # vitam-mongo-s3-offer-02 mongo_cluster_name=offer-s3-1
223
224 [hosts_mongoc_offer]
225 # WARNING : DO NOT COLLOCATE WITH [hosts_mongoc_data]
226 # TODO: put here servers where this service will be deployed : mongoc cluster for ↳
227 ↳ storage offers
228 # Mandatory params
229 # - mongo_cluster_name=<offer_name> ; name of the cluster (should exist on vitam_
230 ↳ strategy configuration in offer_opts.yml)

```

(suite sur la page suivante)

(suite de la page précédente)

```

224 # Optional params
225 #   - mongo_rs_bootstrap=true ; mandatory for 1 node, some init commands will be_
226 #     ↪executed on it
227 #   The recommended practice is to install the mongoc instance on the same servers as_
228 #     ↪the mongos instances
229 # Recommended practice in production: use 3 instances
230 # IMPORTANT : Updating cluster configuration is NOT supported. Do NOT add/remove a_
231 #     ↪host to an existing replica set.
232 # Example :
233 # vitam-mongo-swift-offer-01      mongo_cluster_name=offer-swift-1      mongo_rs_
234 #     ↪bootstrap=true
235 # vitam-mongo-swift-offer-02      mongo_cluster_name=offer-swift-1
236 # vitam-swift-offer               mongo_cluster_name=offer-swift-1
237 # vitam-mongo-fs-offer-01        mongo_cluster_name=offer-fs-1      mongo_rs_
238 #     ↪bootstrap=true
239 # vitam-mongo-fs-offer-02        mongo_cluster_name=offer-fs-1
240 # vitam-fs-offer                 mongo_cluster_name=offer-fs-1
241 # vitam-mongo-s3-offer-01        mongo_cluster_name=offer-s3-1      mongo_rs_
242 #     ↪bootstrap=true
243 # vitam-mongo-s3-offer-02        mongo_cluster_name=offer-s3-1
244 # vitam-s3-offer                 mongo_cluster_name=offer-s3-1
245
246 [hosts_mongod_offer]
247 # WARNING : DO NOT COLLOCATE WITH [hosts_mongod_data]
248 # TODO: put here servers where this service will be deployed : mongod cluster for_
249 #     ↪storage offers
250 # Mandatory params
251 #   - mongo_cluster_name=<offer_name> ; name of the cluster (should exist on vitam_
252 #     ↪strategy configuration in offer_opts.yml)
253 #   - mongo_shard_id=x ; increment by 1 from 0 to n to create multiple shards
254 # Optional params
255 #   - mongo_rs_bootstrap=true (default: false); mandatory for 1 node of the shard,_
256 #     ↪some init commands will be executed on it
257 #   - mongo_arbiter=true (default: false); the node will be only an arbiter, it will_
258 #     ↪not store data ; do not add this parameter on a mongo_rs_bootstrap node, maximum 1_
259 #     ↪node per shard
260 #   - mongod_memory=x (default: unset); this will force the wiredtiger cache size to x_
261 #     ↪(unit is GB)
262 #   - is_small=true (default: false); this will force the priority for this server to_
263 #     ↪be lower when electing master ; hardware can be downgraded for this machine
264 # Recommended practice in production: use 3 instances per shard
265 # IMPORTANT : Updating cluster configuration is NOT supported. Do NOT add/remove a_
266 #     ↪host to an existing replica set, update shard id, arbiter mode or PSSmin_
267 #     ↪configuration.
268 # Example :
269 # vitam-mongo-swift-offer-01      mongo_cluster_name=offer-swift-1      mongo_shard_id=0
270 #     ↪mongo_rs_bootstrap=true
271 # vitam-mongo-swift-offer-02      mongo_cluster_name=offer-swift-1      mongo_shard_id=0
272 # vitam-swift-offer               mongo_cluster_name=offer-swift-1      mongo_shard_id=0
273 #     ↪mongo_arbiter=true
274 # vitam-mongo-fs-offer-01        mongo_cluster_name=offer-fs-1      mongo_shard_id=0
275 #     ↪mongo_rs_bootstrap=true
276 # vitam-mongo-fs-offer-02        mongo_cluster_name=offer-fs-1      mongo_shard_id=0
277 # vitam-fs-offer                 mongo_cluster_name=offer-fs-1      mongo_shard_id=0
278 #     ↪mongo_arbiter=true
279 # vitam-mongo-s3-offer-01        mongo_cluster_name=offer-s3-1      mongo_shard_id=0
280 #     ↪mongo_rs_bootstrap=true

```

(suite sur la page suivante)

(suite de la page précédente)

```

262 # vitam-mongo-s3-offer-02      mongo_cluster_name=offer-s3-1      mongo_shard_id=0 ↵
263 ↵is_small=true # PSSmin, this machine needs less hardware
264 # vitam-s3-offer               mongo_cluster_name=offer-s3-1      mongo_shard_id=0 ↵
265 ↵mongo_arbiter=true
266
267 ##### Zone data
268
269 # Group definition ; DO NOT MODIFY
270 [zone_data:children]
271 hosts_elasticsearch_data
272 hosts_mongodb_data
273
274 [hosts_elasticsearch_data]
275 # TODO: Put here servers where this service will be deployed : elasticsearch-data
276 ↵cluster
277 # 2 params available for huge environments (parameter to be declared after each
278 ↵server) :
279 #     is_data=true/false
280 #     is_master=true/false
281 #     for site/room balancing : is_balancing=<whatever> so replica can be applied on
282 ↵all sites/rooms ; default is vitam_site_name
283 #     other options are not handled yet
284 # defaults are set to true, if undefined. If defined, at least one server MUST be is_
285 ↵data=true
286 # Examples :
287 # server1 is_master=true is_data=false
288 # server2 is_master=false is_data=true
289 # More explanation here : https://www.elastic.co/guide/en/elasticsearch/reference/5.6/
290 ↵modules-node.html
291
292
293 # Group definition ; DO NOT MODIFY
294 [hosts_mongodb_data:children]
295 hosts_mongos_data
296 hosts_mongoc_data
297 hosts_mongod_data
298
299 [hosts_mongos_data]
300 # WARNING : DO NOT COLLOCATE WITH [hosts_mongos_offer]
301 # TODO: Put here servers where this service will be deployed : mongos_data cluster
302 # Mandatory params
303 #     - mongo_cluster_name=mongo-data ; "mongo-data" is mandatory
304 #     The recommended practice is to install the mongos instance on the same servers as
305 ↵the mongoc instances
306 # Example :
307 # vitam-mdbs-01    mongo_cluster_name=mongo-data
308 # vitam-mdbs-02    mongo_cluster_name=mongo-data
309 # vitam-mdbs-03    mongo_cluster_name=mongo-data
310
311
312 [hosts_mongoc_data]
313 # WARNING : DO NOT COLLOCATE WITH [hosts_mongoc_offer]
314 # TODO: Put here servers where this service will be deployed : mongoc_data cluster
315 # Mandatory params
316 #     - mongo_cluster_name=mongo-data ; "mongo-data" is mandatory
317 # Optional params

```

(suite sur la page suivante)

(suite de la page précédente)

```

311 # - mongo_rs_bootstrap=true ; mandatory for 1 node, some init commands will be_
312 # executed on it
313 # The recommended practice is to install the mongoc instance on the same servers as_
314 # the mongos instances
315 # Recommended practice in production: use 3 instances
316 # IMPORTANT : Updating cluster configuration is NOT supported. Do NOT add/remove a_
317 # host to an existing replica set.
318 # Example :
319 # vitam-mdbs-01 mongo_cluster_name=mongo-data mongo_rs_bootstrap=true
320 # vitam-mdbs-02 mongo_cluster_name=mongo-data
321 # vitam-mdbs-03 mongo_cluster_name=mongo-data

322 [hosts_mongod_data]
323 # WARNING : DO NOT COLLOCATE WITH [hosts_mongod_offer]
324 # TODO: Put here servers where this service will be deployed : mongod_data cluster
325 # Each replica_set should have an odd number of members (2n + 1)
326 # Reminder: For Vitam, one mongodb shard is using one replica_set
327 # Mandatory params
328 # - mongo_cluster_name=mongo-data ; "mongo-data" is mandatory
329 # - mongo_shard_id=x ; increment by 1 from 0 to n to create multiple shards
330 # Optional params
331 # - mongo_rs_bootstrap=true (default: false); mandatory for 1 node of the shard,_
332 # some init commands will be executed on it
333 # - mongo_arbiter=true (default: false); the node will be only an arbiter, it will_
334 # not store data ; do not add this parameter on a mongo_rs_bootstrap node, maximum 1_
335 # node per shard
336 # - mongod_memory=x (default: unset); this will force the wiredtiger cache size to x_
337 # (unit is GB) ; can be usefull when colocalization with elasticsearch
338 # - is_small=true (default: false); this will force the priority for this server to_
339 # be lower when electing master ; hardware can be downgraded for this machine
340 # Recommended practice in production: use 3 instances per shard
341 # IMPORTANT : Updating cluster configuration is NOT supported. Do NOT add/remove a_
342 # host to an existing replica set, update shard id, arbiter mode or PSSmin_
343 # configuration.
344 # Example:
345 # vitam-mdbd-01 mongo_cluster_name=mongo-data mongo_shard_id=0 mongo_rs_
346 # bootstrap=true
347 # vitam-mdbd-02 mongo_cluster_name=mongo-data mongo_shard_id=0
348 # vitam-mdbd-03 mongo_cluster_name=mongo-data mongo_shard_id=0 is_small=true #
349 # PSSmin, this machine needs less hardware
350 # vitam-mdbd-04 mongo_cluster_name=mongo-data mongo_shard_id=1 mongo_rs_
351 # bootstrap=true
352 # vitam-mdbd-05 mongo_cluster_name=mongo-data mongo_shard_id=1
353 # vitam-mdbd-06 mongo_cluster_name=mongo-data mongo_shard_id=1 mongo_arbiter=true

354 ###### Zone admin

355 # Group definition ; DO NOT MODIFY
356 [zone_admin:children]
357 hosts_cerebro
358 hosts_consul_server
359 hosts_kibana_data
360 log_servers
361 hosts_elasticsearch_log
362 prometheus

```

(suite sur la page suivante)

(suite de la page précédente)

```

355 hosts_grafana
356
357 [hosts_cerebro]
358 # TODO: Put here servers where this service will be deployed : vitam-elasticsearch-
359 # ↵cerebro
360 # !\ WARNING !!! NOT recommended for PRODUCTION
361
362 [hosts_consul_server]
363 # TODO: Put here servers where this service will be deployed : consul
364 # Recommended practice in production: use 3 instances
365
366 [hosts_kibana_data]
367 # TODO: Put here servers where this service will be deployed : kibana (for data_
368 # ↵cluster)
369 # WARNING : DEPRECATED / We'll soon be removed.
370 # !\ WARNING !!! NOT FOR PRODUCTION
371
372 [log_servers:children]
373 hosts_kibana_log
374 hosts_logstash
375
376 [hosts_kibana_log]
377 # TODO: Put here servers where this service will be deployed : kibana (for log_
378 # ↵cluster)
379
380 [hosts_logstash]
381 # TODO: Put here servers where this service will be deployed : logstash
382 # IF you connect VITAM to external SIEM, DO NOT FILL THE SECTION
383
384
385 [hosts_elasticsearch_log]
386 # TODO: Put here servers where this service will be deployed : elasticsearch-log_
387 # ↵cluster
388 # IF you connect VITAM to external SIEM, DO NOT FILL THE SECTION
389
390 #####
391 ##### Extra VITAM applications #####
392 [prometheus:children]
393 hosts_prometheus
394 hosts_alertmanager
395
396 [hosts_prometheus]
397 # TODO: Put here server where this service will be deployed : prometheus server
398
399 [hosts_alertmanager]
400 # TODO: Put here servers where this service will be deployed : alertmanager
401
402
403 [hosts_grafana]
404 # TODO: Put here servers where this service will be deployed : grafana-server
405
406
407

```

(suite sur la page suivante)

(suite de la page précédente)

```

408 ##########
409 # ZONE VITAMUI
410 #####
411 [hosts_vitamui]
412 # optional: To deploy exporters on VitamUI
413
414
415 [hosts_vitamui:children]
416 hosts_vitamui_mongod
417
418 [hosts_vitamui_mongod]
419 # optional: To deploy mongodb-exporter on VitamUI
420
421
422 ##### Global vars #####
423
424 [hosts:vars]
425
426 # =====
427 # VITAM
428 # =====
429
430 # Declare user for ansible on target machines
431 ansible_ssh_user=
432 # Can target user become as root ? ; true is required by VITAM (usage of a sudoer is mandatory)
433 ansible_become=true
434 # How can ansible switch to root ?
435 # See https://docs.ansible.com/ansible/latest/user_guide/become.html
436
437 # Related to Consul ; apply in a table your DNS server(s)
438 # Example : dns_servers=["8.8.8.8","8.8.4.4"]
439 # If no dns recursors are available, leave this value empty.
440 dns_servers=
441
442 # Define local Consul datacenter name
443 # CAUTION !!! Only alphanumeric characters when using s3 as offer backend !!!
444 vitam_site_name=prod-dc1
445
446 # On offer, value is the prefix for all container's names. If upgrading from R8, you MUST UNCOMMENT this parameter AS IS !!!
447 #vitam_prefix_offer=""
448
449 # check whether on primary site (true) or secondary (false)
450 primary_site=true
451
452 # =====
453 # EXTRA
454 # =====
455
456 ### vitam-itest repository ###
457 vitam_tests_branch=master
458 vitam_tests_gitrepo_protocol=
459 vitam_tests_gitrepo_baseurl=
460 vitam_tests_gitrepo_url=
461
462 # Used when VITAM is behind a reverse proxy (provides configuration for reverse proxy, && displayed in header page)

```

(suite sur la page suivante)

(suite de la page précédente)

```

463 vitam_reverse_external_dns=
464 # For reverse proxy use
465 reverse_proxy_port=443
466 vitam_reverse_external_protocol=https
467 # http_proxy env var to use ; has to be declared even if empty
468 http_proxy_environnement=

```

Pour chaque type de *host*, indiquer le(s) serveur(s) défini(s), pour chaque fonction. Une colocalisation de composants est possible (Cf. le paragraphe idoine du [DAT](#))

Note : Concernant le groupe *hosts_consul_server*, il est nécessaire de déclarer au minimum 3 machines.

Avertissement : Il n'est pas possible de colocaliser les clusters MongoDB *data* et *offer*.

Avertissement : Il n'est pas possible de colocaliser *kibana-data* et *kibana-log*.

Note : Pour les composants considérés par l'exploitant comme étant « hors [VITAM](#) » (typiquement, le composant ihm-demo), il est possible de désactiver la création du service Consul associé. Pour cela, après chaque hostname impliqué, il faut rajouter la directive suivante : `consul_disabled=true`.

Prudence : Concernant la valeur de `vitam_site_name`, seuls les caractères alphanumériques et le tiret (« - ») sont autorisés (regexp : `[A-Za-z0-9-]`).

Note : Il est possible de multi-instancier le composant « storage-offer-default » dans le cas d'un *provider* de type objet (s3, swift). Il faut ajouter `offer_conf=<le nom>`.

4.2.3.2.2 Fichier `main.yml`

La configuration des principaux paramètres est réalisée dans le fichier `reperoire_inventory/group_vars/all/main/main.yml`, comme suit :

```

1 -----
2
3 # TENANTS
4 # List of active tenants
5 vitam_tenant_ids: [0,1,2,3,4,5,6,7,8,9]
6 # For functional-administration, manage master/slave tenant configuration
7 # http://www.programmavitam.fr/ressources/DocCourante/html/installation/installation/
8 #→21-addons.html#passage-des-identifiants-des-referentiels-en-mode-esclave
9 vitam_tenants_usage_external:
10   - name: 0
     identifiers:

```

(suite sur la page suivante)

(suite de la page précédente)

```

11      - INGEST_CONTRACT
12      - ACCESS_CONTRACT
13      - MANAGEMENT_CONTRACT
14      - ARCHIVE_UNIT_PROFILE
15  - name: 1
16  identifiers:
17      - INGEST_CONTRACT
18      - ACCESS_CONTRACT
19      - MANAGEMENT_CONTRACT
20      - PROFILE
21      - SECURITY_PROFILE
22      - CONTEXT
23
24  # GRIFFINS
25  # Vitam griffins required to launch preservation scenario
26  # Example:
27  # vitam_griffins: ["vitam-imagemagick-griffin", "vitam-libreoffice-griffin", "vitam-
28  ↪jhave-griffin", "vitam-odfvalidator-griffin", "vitam-siegfried-griffin", "vitam-
29  ↪tesseract-griffin", "vitam-verapdf-griffin", "vitam-ffmpeg-griffin"]
30  vitam_griffins: []
31
32  # CONSUL
33  consul:
34      network: "ip_admin" # Which network to use for consul communications ? ip_admin or ↪
35  ↪ip_service ?
36  consul_remote_sites:
37  # wan contains the wan addresses of the consul server instances of the external ↪
38  ↪vitam sites
39  # Exemple, if our local dc is dc1, we will need to set dc2 & dc3 wan conf:
40  #   - dc2:
41  #     wan: ["10.10.10.10", "1.1.1.1"]
42  #   - dc3:
43  #     wan: ["10.10.10.11", "1.1.1.1"]
44
45  # LOGGING
46  # vitam_defaults:
47  #   access_retention_days: 365 # Number of days for accesslog retention
48  #   access_total_size_cap: "5GB" # total acceptable size
49  #   logback_max_file_size: "10MB"
50  #   logback_total_size_cap:
51  #     file:
52  #       history_days: 365
53  #       totalsize: "5GB"
54  #     security:
55  #       history_days: 365
56  #       totalsize: "5GB"
57
58  # ELASTICSEARCH
59  # 'number_of_shards': number of shards per index, every ES shard is stored as a ↪
60  ↪lucene index
61  # 'number_of_replicas': number of additional copies of primary shards
62  # Total number of shards: number_of_shards * (1 primary + M number_of_replicas)
63  # CAUTION: The total number of shards should be lower than or equal to the number of ↪
64  ↪elasticsearch-data instances in the cluster
65  # More details in groups_vars/all/advanced/tenants_vars.yml file
66  vitam_elasticsearch_tenant_indexation:
67      default_config:

```

(suite sur la page suivante)

(suite de la page précédente)

```

62      # Default settings for masterdata collections (1 index per collection)
63      masterdata:
64          number_of_shards: 1
65          number_of_replicas: 2
66      # Default settings for unit indexes (1 index per tenant)
67      unit:
68          number_of_shards: 1
69          number_of_replicas: 2
70      # Default settings for object group indexes (1 index per tenant)
71      objectgroup:
72          number_of_shards: 1
73          number_of_replicas: 2
74      # Default settings for logbook operation indexes (1 index per tenant)
75      logbookoperation:
76          number_of_shards: 1
77          number_of_replicas: 2
78      # Default settings for collect_unit indexes
79      collect_unit:
80          number_of_shards: 1
81          number_of_replicas: 2
82      # Default settings for collect_objectgroup indexes
83      collect_objectgroup:
84          number_of_shards: 1
85          number_of_replicas: 2
86
87      collect_grouped_tenants:
88      - name: 'all'
89          # Group all tenants for collect's indexes (collect_unit & collect_objectgroup)
90          tenants: "{{ vitam_tenant_ids | join(',') }}"
91
92      elasticsearch:
93          log:
94              index_templates:
95                  default:
96                      shards: 1
97                      replica: 1
98          data:
99              index_templates:
100                 default:
101                     shards: 1
102                     replica: 2
103
104      curator:
105          indices:
106              vitam:
107                  close: 30
108                  delete: 365
109              access:
110                  close: 30
111                  delete: 180
112              system:
113                  close: 7
114                  delete: 30

```

Une attention particulière doit être portée à la configuration du nombre de shards et de replicas dans le paramètre vitam_elasticsearch_tenant_indexation.default_config.

Voir aussi :

Se référer au chapitre « Gestion des indexes Elasticsearch dans un contexte massivement multi-tenants » du [DEX](#) pour plus d'informations sur cette fonctionnalité.

Avertissement : Attention, en cas de modification de la distribution des tenants, une procédure de réindexation de la base elasticsearch-data est nécessaire. Cette procédure est à la charge de l'exploitation et nécessite un arrêt de service sur la plateforme. La durée d'exécution de cette réindexation dépend de la quantité de données à traiter.

Voir aussi :

Se référer au chapitre « Réindexation » du [DEX](#) pour plus d'informations.

4.2.3.2.3 Fichier vitam_security.yml

La configuration des droits d'accès à VITAM est réalisée dans le fichier `!reperatoire_inventory!“group_vars/all/advanced/vitam_security.yml”`, comme suit :

```

1   ---
2
3   hide_passwords_during_deploy: true
4
5   ### Admin context name and tenants ####
6   admin_context_name: "admin-context"
7   admin_context_tenants: "{{ vitam_tenant_ids }}"
8
9   # Indicate context certificates relative paths under {{ inventory_dir }}/certs/client-
10  # ↪external/clients
11  # vitam-admin-int is mandatory for internal use (PRONOM upload)
12  admin_context_certs:
13    - "{{ 'collect-external/collect-external.crt' if groups['hosts_collect_external'] | "
14      # ↪default([]) | length > 0 else '' }}"
15    - "{{ 'ihm-demo/ihm-demo.crt' if groups['hosts_ihm_demo'] | default([]) | length > "
16      # ↪0 else '' }}"
17    - "{{ 'ihm-recette/ihm-recette.crt' if groups['hosts_ihm_recette'] | default([]) | "
18      # ↪length > 0 else '' }}"
19    - "vitam-admin-int/vitam-admin-int.crt"
20
21  # Indicate here all the personal certificates relative paths under {{ inventory_dir }}/
22  # ↪/certs/client-vitam-users/clients
23  admin_personal_certs: []
24
25  # Admin security profile name
26  admin_security_profile: "admin-security-profile"
27
28  admin_basic_auth_user: "adminUser"
29
30  # SELinux state, can be: enforcing, permissive, disabled
31  selinux_state: "disabled"
32  # SELinux Policy, can be: targeted, minimum, mls
33  selinux_policy: "targeted"
34  # If needed, reboot the VM to enable SELinux
35  selinux_reboot: True
36  # Relabel the entire filesystem ?
37  selinux_relabel: False

```

Note : Pour la directive `admin_context_certs` concernant l'intégration de certificats *SIA* au déploiement, se reporter à la section *Intégration d'une application externe (cliente)* (page 69).

Note : Pour la directive `admin_personal_certs` concernant l'intégration de certificats personnels (*personae*) au déploiement, se reporter à la section *Intégration d'un certificat personnel (personae)* (page 69).

4.2.3.2.4 Fichier `offers_opts.yml`

La déclaration de configuration des offres de stockage associées se fait dans le fichier `lrepository_inventory/group_vars/all/main/offers_opts.yml` :

```

1  # This is the default vitam strategy ('default'). It is mandatory and must
2  # define a referent offer.
3  # This list of offers will be ordered by the property rank. It has to be
4  # completed if more offers are necessary
5  # The property rank indicates the rank of the offer in the strategy. The
6  # ranking is done in ASC order and should be different for all declared
7  # offers
8  vitam_strategy:
9    - name: offer-fs-1
10      referent: true
11      rank: 0
12
13    # Optional params for each offers in vitam_strategy. If not set, the default
14    # values are applied.
15    #   referent: false           # true / false (default), only one per
16    # site must be referent
17    #   status: ACTIVE          # ACTIVE (default) / INACTIVE
18    #   vitam_site_name: distant-dc2 # default is the value of vitam_site_name
19    # defined in your local inventory file, should be specified with the vitam_
20    # site_name defined for the distant offer
21    #   distant: false          # true / false (default). If set to true,
22    # it will not check if the provider for this offer is correctly set
23    #   id: idoffre             # OPTIONAL, but IF ACTIVATED, MUST BE
24    # UNIQUE & SAME if on another site
25    #   asyncRead: false         # true / false (default). Should be set to
26    # true for tape offer only
27    #   rank: 0                 # Integer that indicates in ascending
28    # order the priority of the offer in the strategy
29
30  # Example for tape offer:
31  # Tape offer mustn't be referent (referent: false) and should be configured
32  # as asynchrone read (asyncRead: true)
33  #   - name: offer-tape-1
34  #     referent: false
35  #     asyncRead: true
36  #     rank: 0
37
38  # Example distant offer:
39  #   - name: distant
40  #     referent: false
41  #     vitam_site_name: distant-dc2

```

(suite sur la page suivante)

(suite de la page précédente)

```

29      # distant: true # Only add this parameter when distant offer (not on same ↵
30      # platform)
31      # rank: 1
32
33      # WARNING : multi-strategy is a BETA functionality
34      # More strategies can be added but are optional
35      # Strategy name must only use [a-z][a-z0-9-]* pattern
36      # Any strategy must contain at least one offer
37      # This list of offers is ordered. It can and has to be completed if more ↵
38      # offers are necessary
39      # Every strategy can define at most one referent offer.
40      # other_strategies:
41      #   metadata:
42      #     - name: offer-fs-1
43      #       referent: true
44      #       rank: 0
45      #     - name: offer-fs-2
46      #       referent: false
47      #       rank: 1
48      #   binary:
49      #     - name: offer-fs-2
50      #       referent: false
51      #       rank: 0
52      #     - name: offer-s3-1
53      #       referent: false
54      #       rank: 1
55
56      # DON'T forget to add associated passwords in vault-vitam.yml with same tree ↵
57      # when using provider openstack-swift*
58      # ATTENTION !!! Each offer has to have a distinct name, except for clusters ↵
59      # binding a same physical storage
60      # WARNING : for offer names, please only use [a-z][a-z0-9-]* pattern
61      vitam_offers:
62      offer-fs-1:
63          # param can be filesystem-hash (recommended) or filesystem (not ↵
64          # recommended)
65          provider: filesystem-hash
66          ### Optional parameters
67          # Offer log compaction
68          offer_log_compaction:
69              ## Expiration, here offer logs 21 days old will be compacted
70              expiration_value: 21
71                  ## Choose one of "MILLENNIA", "HALF_DAYS", "MILLIS", "FOREVER", "MICROS"
72                  #, "CENTURIES", "DECADES", "YEARS", "DAYS", "SECONDS", "HOURS", "MONTHS",
73                  #, "WEEKS", "NANOS", "MINUTES", "ERAS"
74              expiration_unit: "DAYS"
75                  ## Compaction bulk size here 10 000 offers logs (at most) will be ↵
76                  # compacted (Expected value between 1 000 and 200 000)
77                  compaction_size: 10000
78          # Batch processing thread pool size
79          maxBatchThreadPoolSize: 32
80          # Batch metadata computation timeout in seconds
81          batchMetadataComputationTimeout: 600
82          # Clean up non-rewritable / non-overridable objects on write errors.
83          # True (default) for fast auto-recovery in most failure cases, but may ↵
84          # cause data loses in rare cases of concurrent (re-)writes by other threads. ↵
85          # Use at your own risk.

```

(suite sur la page suivante)

(suite de la page précédente)

```

76  cleanupObjectsOnWriteError: true
77  #####
78  #####
79  offer-swift-1:
80    # provider : openstack-swift for v1 or openstack-swift-v3 for v3
81    provider: openstack-swift-v3
82    # swiftKeystoneAuthUrl : URL de connexion à keystone
83    swiftKeystoneAuthUrl: https://openstack-hostname:port/auth/1.0
84    # swiftDomain : domaine OpenStack dans lequel l'utilisateur est enregistré
85    swiftDomain: domaine
86    # swiftUser: has to be set in vault-vitam.yml (encrypted) with same structure => DO NOT COMMENT OUT
87    # swiftPassword: has to be set in vault-vitam.yml (encrypted) with same structure => DO NOT COMMENT OUT
88    # swiftProjectName : nom du projet openstack
89    swiftProjectName: monTenant
90    ### Optional parameters
91    # swiftUrl: optional variable to force the swift URL
92    # swiftUrl: https://swift-hostname:port/swift/v1
93    #SSL TrustStore
94    swiftTrustStore: /chemin_vers_mon_fichier/monSwiftTrustStore.jks
95    #Max connection (concurrent connections), per route, to keep in pool (if a pooling ConnectionManager is used) (optional, 200 by default)
96    swiftMaxConnectionsPerRoute: 200
97    #Max total connection (concurrent connections) to keep in pool (if a pooling ConnectionManager is used) (optional, 1000 by default)
98    swiftMaxConnections: 1000
99    #Max time (in milliseconds) for waiting to establish connection (optional, 200000 by default)
100   swiftConnectionTimeout: 200000
101   #Max time (in milliseconds) waiting for a data from the server (socket) (optional, 60000 by default)
102   swiftReadTimeout: 60000
103   # Disable keep-alive. Optional, defaults to false.
104   swiftDisableKeepAlive: false
105   #Default number of (re)tries on errors
106   swiftNbRetries: 3
107   #Time (in seconds) to renew a token before expiration occurs (blocking) (optional, 60 by default)
108   swiftHardRenewTokenDelayBeforeExpireTime: 60
109   #Time (in seconds) to renew a token before expiration occurs (optional, 300 by default)
110   swiftSoftRenewTokenDelayBeforeExpireTime: 300
111   # Offer log compaction
112   offer_log_compaction:
113     ## Expiration, here offer logs 21 days old will be compacted
114     expiration_value: 21
115     ## Choose one of "MILLENNIA", "HALF_DAYS", "MILLIS", "FOREVER", "MICROS", "CENTURIES", "DECADES", "YEARS", "DAYS", "SECONDS", "HOURS", "MONTHS", "WEEKS", "NANOS", "MINUTES", "ERAS"
116     expiration_unit: "DAYS"
117     ## Compaction bulk size here 10 000 offers logs (at most) will be compacted (Expected value between 1 000 and 200 000)
118     compaction_size: 10000
119     # Batch processing thread pool size
      maxBatchThreadPoolSize: 32

```

(suite sur la page suivante)

(suite de la page précédente)

```

120  # Batch metadata computation timeout in seconds
121  batchMetadataComputationTimeout: 600
122  # Clean up non-rewritable / non-overridable objects on write errors.
123  # True (default) for fast auto-recovery in most failure cases, but may
124  #cause data loses in rare cases of concurrent (re-)writes by other threads.
125  #Use at your own risk.
126  cleanupObjectsOnWriteError: true
127  # Enable / Disable use of vitam custom headers for offer requests
128  enableCustomHeaders: false
129  # List of vitam custom headers used by offer requests
130  #customHeaders:
131  #    # - key: 'Cookie'
132  #        #     value: 'Origin=vitam'
133  #####
134  #####
135  offer-s3-1:
136  # provider : can only be amazon-s3-v2 for Amazon SDK S3 V2
137  provider: 'amazon-s3-v2'
138  # s3Endpoint : URL of connection to S3
139  s3Endpoint: http://172.17.0.2:6007
140  ### Optional parameters
141  # s3RegionName (optional): Region name (default value us-east-1)
142  s3RegionName: us-west-1
143  # s3PathStyleAccessEnabled (optional): 'true' to access bucket in "path-
144  #style", else "virtual-hosted-style" (true by default)
145  s3PathStyleAccessEnabled: true
146  # s3MaxConnections (optional): Max total connection (concurrent_
147  #connections) (50 by default)
148  s3MaxConnections: 1000
149  # s3ConnectionTimeout (optional): Max time (in milliseconds) for waiting_
150  #to establish connection (10000 by default)
151  s3ConnectionTimeout: 200000
152  # s3SocketTimeout (optional): Max time (in milliseconds) for reading_
153  #from a connected socket (50000 by default)
154  s3SocketTimeout: 50000
155  # s3RequestTimeout (optional): Max time (in milliseconds) for a request_
156  #(0 by default, disabled)
157  s3RequestTimeout: 0
158  # s3ClientExecutionTimeout (optional): Max time (in milliseconds) for a_
159  #request by java client (0 by default, disabled)
160  s3ClientExecutionTimeout: 0
161  # Disable multipart upload of large objects (legacy mode / not_
162  #recommended, only for S3 servers without multipart upload support)
163  s3DisableMultipartUpload: false
164  # Max upload size for single object upload size in MB (min: 5 MB, max: 5_
165  #GB, default: 5 GB)
166  s3MaxUploadPartSizeMB: 5_120
167  # Nb retries for S3 multipart upload cleanup
168  s3MultiPartCleanNbRetries: 3
169  # Wait delay for S3 multipart upload cleanup (in milliseconds)
170  s3MultiPartCleanWaitingTimeInMilliseconds: 10_000
171  # Ignore S3 HTTPS certificate hostname validation (defaults to false,_
172  #not for production environments)
173  s3IgnoreCertificateHostnameValidation: false
174  # Offer log compaction
175  offer_log_compaction:
176  ## Expiration, here offer logs 21 days old will be compacted

```

(suite sur la page suivante)

(suite de la page précédente)

```

165      expiration_value: 21
166      ## Choose one of "MILLENNIA", "HALF_DAYS", "MILLIS", "FOREVER", "MICROS
167      ↵", "CENTURIES", "DECADES", "YEARS", "DAYS", "SECONDS", "HOURS", "MONTHS",
168      ↵"WEEKS", "NANOS", "MINUTES", "ERAS"
169      expiration_unit: "DAYS"
170      ## Compaction bulk size here 10 000 offers logs (at most) will be
171      ↵compacted (Expected value between 1 000 and 200 000)
172      compaction_size: 10000
173      # Batch processing thread pool size
174      maxBatchThreadPoolSize: 32
175      # Batch metadata computation timeout in seconds
176      batchMetadataComputationTimeout: 600
177      # Clean up non-rewritable / non-overridable objects on write errors.
178      # True (default) for fast auto-recovery in most failure cases, but may
179      ↵cause data loses in rare cases of concurrent (re-)writes by other threads.
180      ↵Use at your own risk.
181      cleanupObjectsOnWriteError: true
182      #####
183      ↵#####
184      offer-tape-1:
185          provider: tape-library
186          # tapeLibraryConfiguration:
187          #
188          # ...
189          # topology:
190          #
191          # ...
192          # tapeLibraries:
193          #
194          # ...
195          # Offer log compaction
196          offer_log_compaction:
197              ## Expiration, here offer logs 21 days old will be compacted
198              expiration_value: 21
199              ## Choose one of "MILLENNIA", "HALF_DAYS", "MILLIS", "FOREVER", "MICROS
200              ↵", "CENTURIES", "DECADES", "YEARS", "DAYS", "SECONDS", "HOURS", "MONTHS",
201              ↵"WEEKS", "NANOS", "MINUTES", "ERAS"
202              expiration_unit: "DAYS"
203              ## Compaction bulk size here 10 000 offers logs (at most) will be
204              ↵compacted (Expected value between 1 000 and 200 000)
205              compaction_size: 10000
206              # Batch processing thread pool size
207              maxBatchThreadPoolSize: 32
208              # Batch metadata computation timeout in seconds
209              batchMetadataComputationTimeout: 600
210              # Clean up non-rewritable / non-overridable objects on write errors.
211              # True (default) for fast auto-recovery in most failure cases, but may
212              ↵cause data loses in rare cases of concurrent (re-)writes by other threads.
213              ↵Use at your own risk.
214              cleanupObjectsOnWriteError: true
215              #####
216              ↵#####
217              # WARNING: Swift V1 is deprecated
218              # example_swift_v1:
219              #
220              # provider: openstack-swift
221              # swiftKeystoneAuthUrl: https://keystone/auth/1.0
222              # swiftDomain: domain
223              # swiftUser: has to be set in vault-vitam.yml (encrypted) with same
224              ↵structure => DO NOT COMMENT OUT
225              # swiftPassword: has to be set in vault-vitam.yml (encrypted) with same
226              ↵structure => DO NOT COMMENT OUT

```

(suite sur la page suivante)

(suite de la page précédente)

```

209 # THIS PART IS ONLY FOR CLEANING (and mandatory for this use case)
210 # swiftProjectId: related to OS_PROJECT_ID
211 # swiftRegionName: related to OS_REGION_NAME
212 # swiftInterface: related to OS_INTERFACE
213 # example_swift_v3:
214 # provider: openstack-swift-v3
215 # swiftKeystoneAuthUrl: https://keystone/v3
216 # swiftDomain: domaine
217 # swiftUser: has to be set in vault-vitam.yml (encrypted) with same_
218 # structure => DO NOT COMMENT OUT
219 # swiftPassword: has to be set in vault-vitam.yml (encrypted) with same_
220 # structure => DO NOT COMMENT OUT
221 # swiftProjectName: monTenant
222 # projectName: monTenant
223 # THIS PART IS ONLY FOR CLEANING (and mandatory for this use case)
224 # swiftProjectId: related to OS_PROJECT_ID
225 # swiftRegionName: related to OS_REGION_NAME
226 # swiftInterface: related to OS_INTERFACE
227
228 # swiftTrustStore: /chemin-vers-mon_fichier/monSwiftTrustStore.jks
229 # swiftMaxConnectionsPerRoute: 200
230 # swiftMaxConnections: 1000
231 # swiftConnectionTimeout: 200000
232 # swiftReadTimeout: 60000
233 # swiftDisableKeepAlive: false
234 # Time (in seconds) to renew a token before expiration occurs
235 # swiftHardRenewTokenDelayBeforeExpireTime: 60
236 # swiftSoftRenewTokenDelayBeforeExpireTime: 300
237 # enableCustomHeaders: false
238 # customHeaders:
#   - key: 'Cookie'
#     value: 'Origin=vitam'
```

Se référer aux commentaires dans le fichier pour le renseigner correctement.

Note : Dans le cas d'un déploiement multi-sites, dans la section `vitam_strategy`, la directive `vitam_site_name` définit pour l'offre associée le nom du datacenter Consul. Par défaut, si non définie, c'est la valeur de la variable `vitam_site_name` définie dans l'inventaire qui est prise en compte.

Avertissement : La cohérence entre l'inventaire et la section `vitam_strategy` (et `other_strategies` si multi-stratégies) est critique pour le bon déploiement et fonctionnement de la solution logicielle VITAM. En particulier, la liste d'offres de `vitam_strategy` doit correspondre *exactement* aux noms d'offres déclarés dans l'inventaire (ou les inventaires de chaque datacenter, en cas de fonctionnement multi-site).

Avertissement : Ne pas oublier, en cas de connexion à un keystone en https, de répercuter dans la [PKI](#) la clé publique de la [CA](#) du keystone.

4.2.3.2.5 Fichier `cots_vars.yml`

La configuration s'effectue dans le fichier `|repertoire_inventory|“group_vars/all/advanced/cots_vars.yml“` :

```

1  ---
2
3 consul:
4   retry_interval: 10 # in seconds
5   check_interval: 10 # in seconds
6   check_timeout: 5 # in seconds
7   log_level: WARN # Available log_level are: TRACE, DEBUG, INFO, WARN or ERR
8   at_boot: true
9
10 # Please uncomment and fill values if you want to connect VITAM to external SIEM
11 # external_siem:
12 #   host:
13 #   port:
14
15 elasticsearch:
16   log:
17     host: "elasticsearch-log.service.{{ consul_domain }}"
18     port_http: "9201"
19     at_boot: true
20     groupe: "log"
21     baseuri: "elasticsearch-log"
22     cluster_name: "elasticsearch-log"
23     consul_check_http: 10 # in seconds
24     consul_check_tcp: 10 # in seconds
25     action_log_level: error
26     https_enabled: false
27     indices_fielddata_cache_size: '30%' # related to https://www.elastic.co/guide/en/elasticsearch/reference/8.13/modules-fielddata.html
28     indices_breaker_fielddata_limit: '40%' # related to https://www.elastic.co/guide/en/elasticsearch/reference/8.13/circuit-breaker.html
29     #fielddata-circuit-breaker
30       dynamic_timeout: 30s
31       # log configuration
32     log_appenders:
33       root:
34         log_level: "info"
35       rolling:
36         max_log_file_size: "10MB"
37         max_total_log_size: "2GB"
38       deprecation_rolling:
39         max_log_file_size: "10MB"
40         max_files: "20"
41         log_level: "warn"
42       index_search_slowlog_rolling:
43         log_level: "warn"
44       index_indexing_slowlog_rolling:
45         log_level: "warn"
46       # By default, is commented. Should be uncommented if ansible computes badly vCPUs number ; values are associated vCPUs numbers ; please adapt to your configuration
47         # thread_pool:
48           # index:
49             # size: 2
50             # get:
51               # size: 2

```

(suite sur la page suivante)

(suite de la page précédente)

```

51      #     search:
52      #         size: 2
53      #     write:
54      #         size: 2
55      #     warmer:
56      #         max: 2
57
58 data:
59     host: "elasticsearch-data.service.{{ consul_domain }}"
60     # default is 0.1 (10%) and should be quite enough in most cases
61     #index_buffer_size_ratio: "0.15"
62     port_http: "9200"
63     groupe: "data"
64     baseuri: "elasticsearch-data"
65     cluster_name: "elasticsearch-data"
66     consul_check_http: 10 # in seconds
67     consul_check_tcp: 10 # in seconds
68     action_log_level: debug
69     https_enabled: false
70     indices_fielddata_cache_size: '30%' # related to https://www.elastic.co/guide/en/elasticsearch/reference/8.13/modules-fielddata.html
71     ↵ indices_breaker_fielddata_limit: '40%' # related to https://www.elastic.co/guide/en/elasticsearch/reference/8.13/circuit-breaker.html
72     ↵ #fielddata-circuit-breaker
73     dynamic_timeout: 30s
74     # log configuration
75     log_appenders:
76         root:
77             log_level: "info"
78         rolling:
79             max_log_file_size: "10MB"
80             max_total_log_size: "5GB"
81             deprecation_rolling:
82                 max_log_file_size: "10MB"
83                 max_files: "20"
84                 log_level: "warn"
85             index_search_slowlog_rolling:
86                 log_level: "warn"
87             index_indexing_slowlog_rolling:
88                 log_level: "warn"
89
90             # By default, is commented. Should be uncommented if ansible
91             computes badly vCPUs number ; values are associated vCPUs numbers ; please adapt to your configuration
92             # thread_pool:
93             #     index:
94             #         size: 2
95             #     get:
96             #         size: 2
97             #     search:
98             #         size: 2
99             #     write:
100            #         size: 2
101            #     warmer:
102            #         max: 2
103
104 mongodb:
105     mongos_port: 27017
106     mongoc_port: 27018

```

(suite sur la page suivante)

(suite de la page précédente)

```

103  mongod_port: 27019
104  mongo_authentication: "true"
105  check_consul: 10 # in seconds
106  drop_info_log: false # Drop mongo (I)nformational log, for Verbosity
107  ↵Level of 0
108    # logs configuration
109    logrotate: enabled # or disabled
110    ↵history_days: 30 # How many days to store logs if logrotate is set to
111    ↵'enabled'
112
113  logstash:
114    host: "logstash.service.{{ consul_domain }}"
115    port: 10514
116    rest_port: 20514
117    at_boot: true
118    check_consul: 10 # in seconds
119    ## logstash xms & xmx in Megabytes
120    # jvm_xms: 256 # default to memory_size/8
121    # jvm_xmx: 1024 # default to memory_size/4
122    # workers_number: 4 # default to cores*threads
123    log_appenders:
124      rolling:
125        max_log_file_size: "10MB"
126        max_total_log_size: "2GB"
127      json_rolling:
128        max_log_file_size: "10MB"
129        max_total_log_size: "2GB"
130
131
132  filebeat:
133    at_boot: true
134
135  # Prometheus params
136  prometheus:
137    metrics_path: /admin/v1/metrics
138    check_consul: 10 # in seconds
139    prometheus_config_file_target_directory: # Set path where "prometheus.yml"
140    ↵" file will be generated. Example: /tmp/
141    server:
142      port: 9090
143      at_boot: true
144      tsdb_retention_time: "15d"
145      tsdb_retention_size: "5GB"
146    node_exporter:
147      enabled: true
148      port: 9101
149      at_boot: true
150      metrics_path: /metrics
151      log_level: "warn"
152      logrotate: enabled # or disabled
153      history_days: 30 # How many days to store logs if logrotate is set to
154      ↵'enabled'
155      consul_exporter:
156        enabled: true
157        port: 9107
158        at_boot: true
159        metrics_path: /metrics
160        elasticsearch_exporter:

```

(suite sur la page suivante)

(suite de la page précédente)

```

156     enabled: true
157     port: 9114
158     at_boot: true
159     metrics_path: /metrics
160     log_level: "warn"
161     logrotate: enabled # or disabled
162     history_days: 30 # How many days to store logs if logrotate is set
163     ↵to 'enabled'
164     alertmanager:
165       api_port: 9093
166       cluster_port: 9094
167       at_boot: true
168       #receivers: # https://grafana.com/blog/2020/02/25/step-by-step-guide-
169       ↵to-setting-up-prometheus-alertmanager-with-slack-pagerduty-and-gmail/
170       #- name: "slack_alert"
171       # slack_configs:
172       #   - api_url: "https://hooks.slack.com/services/xxxxxxxxx/
173       ↵xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
174       #   channel: '#your_alert_channel'
175       #   send_resolved: true
176     blackbox_exporter:
177       enabled: true
178       port: 9115
179       at_boot: true
180       logrotate: enabled # or disabled
181       history_days: 30 # How many days to store logs if logrotate is set
182     ↵to 'enabled'
183     targets:
184       ## List all the targeted URLs that must be controled with
185     ↵blackbox
186       - "{{ vitam_reverse_external_protocol | default('https') }}://{{{
187       ↵vitam_reverse_external_dns }}}:{{ reverse_proxy_port | default(443) }},
188       ↵reverse"
189     mongodb_exporter:
190       enabled: true
191       port_mongoc: 9216
192       port_mongod: 9217
193       at_boot: true
194
195     grafana:
196       check_consul: 10 # in seconds
197       http_port: 3000
198       at_boot: true
199
200     # Curator units: days
201     curator:
202       at_boot: true
203       indices:
204         metricbeat:
205           close: 5
206           delete: 10
207         packetbeat:
208           close: 5
209           delete: 10
210
211     kibana:
212       header_value: "reporting"

```

(suite sur la page suivante)

(suite de la page précédente)

```

206 import_delay: 10
207 import_retries: 10
208 # logs configuration
209 logrotate: enabled # or disabled
210 history_days: 30 # How many days to store logs if logrotate is set to
211 ↵ 'enabled'
212 log:
213     baseuri: "kibana_log"
214     api_call_timeout: 120
215     groupe: "log"
216     port: 5601
217     at_boot: true
218     default_index_pattern: "logstash-vitam*"
219     check_consul: 10 # in seconds
220     # default shards & replica
221     shards: 1
222     replica: 1
223     # pour index logstash-*
224     metrics:
225         shards: 1
226         replica: 1
227         # pour index metricbeat-*
228         metricbeat:
229             shards: 3 # must be a factor of 30
230             replica: 1
231 data:
232     baseuri: "kibana_data"
233     # OMA : bugdette : api_call_timeout is used for retries ; should_
234 ↵ create a separate variable rather than this one
235     api_call_timeout: 120
236     groupe: "data"
237     port: 5601
238     default_index_pattern: "logbookoperation_*"
239     check_consul: 10 # in seconds
240     # index template for .kibana
241     shards: 1
242     replica: 1
243
244 syslog:
245     # value can be syslog-nginx, rsyslog or filebeat (default)
246     name: "filebeat"
247
248 # filebeat:
249 # Default values are under ansible-vitam/roles/filebeat/defaults/main.yml
250
251 cerebro:
252     baseuri: "cerebro"
253     port: 9000
254     check_consul: 10 # in seconds
255     # logs configuration
256     logrotate: enabled # or disabled
257     history_days: 30 # How many days to store logs if logrotate is set to
258     ↵ 'enabled'
259
260 siegfried:
261     port: 19000
262     consul_check: 10 # in seconds

```

(suite sur la page suivante)

(suite de la page précédente)

```

260
261 clamav:
262     port: 3310
263     # logs configuration
264     logrotate: enabled # or disabled
265     history_days: 30 # How many days to store logs if logrotate is set to
266     ↳ 'enabled'
267     freshclam:
268         # frequency freshclam for database update per day (from 0 to 24 - 24_
269         ↳ meaning hourly check)
270         db_update_periodicity: 1
271         private_mirror_address:
272         use_proxy: "no"
273
274
275     ## Avast Business Antivirus for Linux
276     ## if undefined, the following default values are applied.
277     # avast:
278     #     # logs configuration
279     #     logrotate: enabled # or disabled
280     #     history_days: 30 # How many days to store logs if logrotate is set to
281     #     ↳ 'enabled'
282     #     manage_repository: true
283     #     repository:
284     #         state: present
285     #         # For RedHat family
286     #         baseurl: https://repo.avcdn.net/linux-av/rpm/el9/release
287     #         gpgcheck: no
288     #         proxy: _none_
289     #         # For Debian family
290     #         baseurl: 'deb https://repo.avcdn.net/linux-av/deb debian-bookworm_
291     #             release'
292     #         vps_repository: http://linux-av.u.avcdn.net/linux-av/avast/x86_64
293     #         ## List of sha256 hash of excluded files from antivirus. Useful for_
294     #             ↳ test environments.
295     #         whitelist:
296     #             - xxxxxx
297     #             - yyyyyy
298
299 mongo_express:
300     baseuri: "mongo-express"
301
302 ldap_authentication:
303     ldap_protocol: "ldap"
304     ldap_server: "{% if groups['ldap']|length > 0 %}{{ groups['ldap']|first }}"
305     ↳ {% endif %}"
306     ldap_port: "389"
307     ldap_base: "dc=programmavitam,dc=fr"
308     ldap_login: "cn=Manager,dc=programmavitam,dc=fr"
309     uid_field: "uid"
310     ldap_userDn_Template: "uid={0},ou=people,dc=programmavitam,dc=fr"
311     ldap_group_request: "(&(objectClass=groupOfNames)(member={0}))"
312     ldap_admin_group: "cn=admin,ou=groups,dc=programmavitam,dc=fr"
313     ldap_user_group: "cn=user,ou=groups,dc=programmavitam,dc=fr"
314     ldap_guest_group: "cn=guest,ou=groups,dc=programmavitam,dc=fr"
315
316     # Backup tool on storage-offer
317     restic:

```

(suite sur la page suivante)

(suite de la page précédente)

```

311 snapshot_retention: 30 # number of snapshots to keep
312 # default run backup at 23:00 everydays
313 cron:
314     minute: '00'
315     hour: '23'
316     day: '*'
317     month: '*'
318     weekday: '*'
319     # [hosts_storage_offer_default] must be able to connect to the listed
320     ↵databases below to properly backup.
321     backup:
322         # mongo-offer
323         - name: "{{ offer_conf }}"
324             type: mongodb
325             host: "{{ offer_conf }}-mongos.service.{{ consul_domain }}:{{
326             ↵mongodb.mongodb_port }}"
327             user: "{{ mongodb[offer_conf].admin.user }}"
328             password: "{{ mongodb[offer_conf].admin.password }}"
329             # # mongo-data (only if mono-sharded cluster)
330             # - name: mongo-data
331             #     type: mongodb
332             #     host: "mongo-data-mongos.service.{{ consul_domain }}:{{ mongodb.
333             ↵mongodb.mongodb_port }}"
334             #     user: "{{ mongodb['mongo-data'].admin.user }}"
335             #     password: "{{ mongodb['mongo-data'].admin.password }}"
336             # # mongo-vitamui (only if vitamui is deployed)
337             # - name: mongo-vitamui
338             #     type: mongodb
339             #     host: mongo-vitamui-mongod.service.{{ consul_domain }}:{{
340             ↵mongodb.mongodb_port }}
341             #     # Add the following params on environments/group_vars/all/main/
342             ↵vault-vitam.yml
343             #     # They can be found under vitamui's deployment sources on
344             ↵environments/group_vars/all/vault-mongodb.yml
345             #     user: "{{ mongodb['mongo-vitamui'].admin.user }}"
346             #     password: "{{ mongodb['mongo-vitamui'].admin.password }}"

```

Dans le cas du choix du *COTS* d'envoi des messages syslog dans logstash, il est possible de choisir entre filebeat, syslog-ng et rsyslog. Il faut alors modifier la valeur de la directive `syslog.name`; la valeur par défaut est filebeat.

Note : si vous décommentez et renseignez les valeurs dans le bloc `external_siem`, les messages seront envoyés (par `syslog` ou `syslog-ng`, selon votre choix de déploiement) dans un *SIEM* externe à la solution logicielle *VITAM*, aux valeurs indiquées dans le bloc ; il n'est alors pas nécessaire de renseigner de partitions pour les groupes `ansible [hosts_logstash]` et `[hosts_elasticsearch_log]`.

4.2.3.2.6 Fichier `tenants_vars.yml`

Le fichier `repertoire_inventory/group_vars/all/advanced/tenants_vars.yml` permet de gérer les configurations spécifiques associés aux tenants de la plateforme (liste des tenants, regroupement de tenants, configuration du nombre de shards et replicas, etc...).

```

1  ##### tenants #####
2  # List of dead / removed tenants that should never be reused / present in
3  ↵vitam_tenant_ids
4  vitam_removed_tenants: [ ]
5  # Administration tenant
6  vitam_tenant_admin: 1
7
8  ####
9  # Elasticsearch tenant indexation
10 # =====
11 #
12 # Elastic search index configuration settings :
13 # - 'number_of_shards' : number of shards per index. Every ES shard is
14 # ↵stored as a lucene index.
15 # - 'number_of_replicas' : number of additional copies of primary shards
16 # The total number of shards : number_of_shards * (1 primary + M number_of_
17 # ↵replicas)
18 #
19 # CAUTION : The total number of shards should be lower than or equal to the
20 # ↵number of elasticsearch-data instances in the cluster
21 #
22 # Default settings should be okay for most use cases.
23 # For more data-intensive workloads or deployments with high number of
24 # ↵tenants, custom tenant and/or collection configuration might be specified.
25 #
26 # Tenant list may be specified as :
27 # - A specific tenant : eg.
28 # ↵'1'
29 # - A tenant range : eg.
30 # ↵'10-19'
31 # - A comma-separated combination of specific tenants & tenant ranges : eg.
32 # ↵'1, 5, 10-19, 50-59'
33 #
34 # Masterdata collections (accesscontract, filerules...) are indexed as
35 # ↵single elasticsearch indexes :
36 # - Index name format : {collection}_{date_time_of_creation}. e.g.
37 # ↵accesscontract_20200415_042011
38 # - Index alias name : {collection}. e.g. accesscontract
39 #
40 # Metadata collections (unit & objectgroup), and logbook operation
41 # ↵collections are stored on a per-tenant index basis :
42 # - Index name : {collection}_{tenant}_{date_time_of_creation}. e.g.
43 # ↵unit_1_20200517_025041
44 # - Index alias name : {collection}_{tenant}. e.g. unit_1
45 #
46 # Very small tenants (1-100K entries) may be grouped in a "tenant group",
47 # ↵and hence, stored in a single elasticsearch index.
48 # This allows reducing the number of indexes & shards that the elasticsearch
49 # ↵cluster need to manage :
50 # - Index name : {collection}_{tenant_group_name}_{date_time_of_
51 # ↵creation}. e.g. logbookoperation_grp5_20200517_025041
52 # - Index alias name : {collection}_{tenant_group_name}. e.g.
53 # ↵logbookoperation_grp5
54 #
55 # Tenant list can be wide ranges (eg: 100-199), and may contain non-existing
56 # ↵(yet) tenants. i.e. tenant lists might be wider than 'vitam_tenant_ids'
57 # ↵section

```

(suite sur la page suivante)

(suite de la page précédente)

```

40 # This allows specifying predefined tenant families (whether normal tenants, ranges, or tenant groups) to which tenants can be added in the future.
41 # However, tenant lists may not intersect (i.e. a single tenant cannot belong to 2 configuration sections).
42 #
43 # Sizing recommendations :
44 #   - 1 shard per 5-10M records for small documents (eg. masterdata collections)
45 #   - 1 shard per 1-2M records for larger documents (eg. metadata & logbook collections)
46 #   - As a general rule, shard size should not exceed 30GB per shard
47 #   - A single ES node should not handle > 200 shards (be it a primary or a replica)
48 #   - It is recommended to start small and add more shards when needed (resharding requires a re-indexation operation)
49 #
50 # Custom mappingFile & customSearch
51 # -----
52 #
53 # metadata & metadata_collect collections (for unit & objectgroup)
54 #   - 'mappingFile' : (Optional) The name of the associated mapping. Must be placed under 'environments/files/elasticsearch-mappings/'
55 #   - 'customSearch' : (Optional) Mandatory if you modify the default mappingFile or provided a dedicated one.
56 #
57 # Example:
58 #   mappingFile: dedicated-unit-es-mapping.json
59 #   customSearch:
60 #     - fieldPath: Title
61 #       types:
62 #         - Strict
63 #         - Minimal
64 #
65 #
66 # /!\ IMPORTANT :
67 # Changing the configuration of an existing tenant requires re-indexation of the tenants and/or tenant groups
68 #
69 # Please refer to the documentation for more details.
70 #
71 #####
72 vitam_elasticsearch_tenant_indexation:
73 #####
74 # Default masterdata collection indexation settings (default_config section) apply for all master data collections
75 # Custom settings can be defined for the following masterdata collections:
76 #   - accesscontract
77 #   - accessionregisterdetail
78 #   - accessionregistersummary
79 #   - accessionregistersymbolic
80 #   - agencies
81 #   - archiveunitprofile
82 #   - context
83 #   - fileformat
84 #   - filerules
85 #   - griffin

```

(suite sur la page suivante)

(suite de la page précédente)

```

87      # - ingestcontract
88      # - managementcontract
89      # - ontology
90      # - preservationscenario
91      # - profile
92      # - securityprofile
93      # - schema
94      #####
95 masterdata:
96     # {collection}:
97     #   number_of_shards: 1
98     #   number_of_replicas: 2
99     # ...
100
101    #####
102    # Custom index settings for regular tenants.
103    #####
104 dedicated_tenants:
105    # - tenants: '1, 3, 11-20'
106    #   unit:
107    #     number_of_shards: 4
108    #     number_of_replicas: 0
109    #   objectgroup:
110    #     number_of_shards: 5
111    #     number_of_replicas: 0
112    #   logbookoperation:
113    #     number_of_shards: 3
114    #     number_of_replicas: 0
115    # ...
116
117    #####
118    # Custom index settings for grouped tenants.
119    # Group name must meet the following criteria:
120    # - alphanumeric characters
121    # - lowercase only
122    # - not start with a number
123    # - be less than 64 characters long.
124    # - NO special characters - / _ | ...
125    #####
126 grouped_tenants:
127    # - name: 'grp1'
128    #   tenants: '5-10'
129    #   unit:
130    #     number_of_shards: 5
131    #     number_of_replicas: 0
132    #   objectgroup:
133    #     number_of_shards: 6
134    #     number_of_replicas: 0
135    #   logbookoperation:
136    #     number_of_shards: 7
137    #     number_of_replicas: 0
138    # ...
139
140 extendedConfiguration:
141   default:
142     eliminationReportExtraFields: [ ]
143     ingestReportUnitExtraFields: ['PersistentIdentifier']

```

(suite sur la page suivante)

(suite de la page précédente)

```

144     ingestReportObjectExtraFields: ['PersistentIdentifier']
145     objectGroupBlackListedFields: [ 'Filename' ]
146
147     custom:
148       # The 'eliminationReportExtraFields' configuration option specifies the
149       # metadata keys that should be included in the report when performing an
150       # elimination.
151       # It determines which additional metadata fields should be retained and
152       # displayed in the elimination report.
153       # You can include any of the following extra fields: "#id", "#version", "
154       #unitups", "#originating_agency", "#approximate_creation_date",
155       #approximate_update_date", "FilePlanPosition", "SystemId",
156       #OriginatingSystemId", "ArchivalAgencyArchiveUnitIdentifier",
157       #OriginatingAgencyArchiveUnitIdentifier",_
158       #TransferringAgencyArchiveUnitIdentifier"
159
160       #
161       # The 'ingestReportUnitExtraFields' configuration option specifies the
162       # metadata keys that should be included in the ATR when performing an ingest.
163       # It determines which additional metadata fields should be retained and
164       # displayed in the ATR report.
165       # You can include any of the following extra fields: "FilePlanPosition",
166       #OriginatingSystemId", "ArchivalAgencyArchiveUnitIdentifier",
167       #OriginatingAgencyArchiveUnitIdentifier",_
168       #TransferringAgencyArchiveUnitIdentifier", "PersistentIdentifier"
169
170       #
171       # The 'ingestReportObjectExtraFields' configuration option specifies the
172       # metadata keys that should be included in the ATR when performing an ingest.
173       # It determines which additional metadata fields should be retained and
174       # displayed in the ATR report.
175       # You can include any of the following extra fields:
176       #PersistentIdentifier", "PhysicalId"
177
178       #
179       # The 'objectGroupBlackListedFields' configuration option specifies the
180       # fields that should not be reported by access-external.
181
182       #
183       # Example for tenant 0 :
184       # 0:
185       #    eliminationReportExtraFields: ["#id", "FilePlanPosition", "SystemId"]
186       #    ingestReportUnitExtraFields: [ "FilePlanPosition",
187       #      "OriginatingSystemId", "ArchivalAgencyArchiveUnitIdentifier",
188       #      "OriginatingAgencyArchiveUnitIdentifier",
189       #      "TransferringAgencyArchiveUnitIdentifier", "PersistentIdentifier" ]
190       #    ingestReportObjectExtraFields: [ "PersistentIdentifier" ]
191       #    objectGroupBlackListedFields: [ 'Filename' ]

```

Se référer aux commentaires dans le fichier pour le renseigner correctement.

Voir aussi :

Se référer au chapitre « Gestion des indexes Elasticsearch dans un contexte massivement multi-tenants » du [DEX](#) pour plus d'informations sur cette fonctionnalité.

Avertissement : Attention, en cas de modification de la distribution des tenants, une procédure de réindexation de la base `elasticsearch-data` est nécessaire. Cette procédure est à la charge de l'exploitation et nécessite un arrêt de service sur la plateforme. La durée d'exécution de cette réindexation dépend de la quantité de données à traiter.

Voir aussi :

Se référer au chapitre « Réindexation » du [DEX](#) pour plus d'informations.

4.2.3.3 Déclaration des secrets

Avertissement : L'ensemble des mots de passe fournis ci-après le sont par défaut et doivent être changés !

4.2.3.3.1 vitam

Avertissement : Cette section décrit des fichiers contenant des données sensibles. Il est important d'implémenter une politique de mot de passe robuste conforme à ce que l'ANSSI préconise. Par exemple : ne pas utiliser le même mot de passe pour chaque service, renouveler régulièrement son mot de passe, utiliser des majuscules, minuscules, chiffres et caractères spéciaux (Se référer à la documentation ANSSI <https://www.ssi.gouv.fr/guide/mot-de-passe>). En cas d'usage d'un fichier de mot de passe (*vault-password-file*), il faut renseigner ce mot de passe comme contenu du fichier et ne pas oublier de sécuriser ou supprimer ce fichier à l'issue de l'installation.

Les secrets utilisés par la solution logicielle (en-dehors des certificats qui sont abordés dans une section ultérieure) sont définis dans des fichiers chiffrés par `ansible-vault`.

Important : Tous les vault présents dans l'arborescence d'inventaire doivent être tous protégés par le même mot de passe !

La première étape consiste à changer les mots de passe de tous les vaults présents dans l'arborescence de déploiement (le mot de passe par défaut est contenu dans le fichier `vault_pass.txt`) à l'aide de la commande `ansible-vault rekey <fichier vault>`.

Voici la liste des vaults pour lesquels il est nécessaire de modifier le mot de passe :

- environments/group_vars/all/main/vault-vitam.yml
- environments/group_vars/all/main/vault-keystores.yml
- environments/group_vars/all/main/vault-extra.yml
- environments/certs/vault-certs.yml

2 vaults sont principalement utilisés dans le déploiement d'une version :

Avertissement : Leur contenu est donc à modifier avant tout déploiement.

- Le fichier `repertoire_inventory/group_vars/all/main/vault-vitam.yml` contient les secrets généraux :

```

1  ---
2  # Vitam platform secret key
3  # Note: It has to be the same on all sites
4  plateforme_secret: change_it_vitamsecret
5
6  # The consul key must be 16-bytes, Base64 encoded: https://www.consul.io/docs/
7  #agent/encryption.html
8  # You can generate it with the "consul keygen" command
9  # Or you can use this script: deployment/pki/scripts/generate_consul_key.sh
  # Note: It has to be the same on all sites

```

(suite sur la page suivante)

(suite de la page précédente)

```
10 consul_encrypt: Biz14ohqN4HtvZmrXp3N4A==  
11  
12 mongodb:  
13     mongo-data:  
14         passphrase: changeitkM4L6zBgK527tWBb  
15         admin:  
16             user: vitamdb-admin  
17             password: change_it_1MpG22m2MywvKW5E  
18         localadmin:  
19             user: vitamdb-localadmin  
20             password: change_it_HycFEVD74g397iRe  
21         system:  
22             user: vitamdb-system  
23             password: change_it_HycFEVD74g397iRe  
24     metadata:  
25         user: metadata  
26         password: change_it_37b97KVaDV8YbCwt  
27     logbook:  
28         user: logbook  
29         password: change_it_jVi6q8eX4H1Ce8UC  
30     report:  
31         user: report  
32         password: change_it_jb7TASZbU6n85t8L  
33     functionalAdmin:  
34         user: functional-admin  
35         password: change_it_9eA2zMCL6tm6KF1e  
36     securityInternal:  
37         user: security-internal  
38         password: change_it_m39XvRQWixyDX566  
39     scheduler:  
40         user: scheduler  
41         password: change_it_Q8WEdxhXXOe2NEhp  
42     collect:  
43         user: collect  
44         password: change_it_m39XvRQWixyDX566  
45     metadataCollect:  
46         user: metadata-collect  
47         password: change_it_37b97KVaDV8YbCwt  
48 offer-fs-1:  
49     passphrase: changeitmB5rnk1M5TY61PqZ  
50     admin:  
51         user: vitamdb-admin  
52         password: change_it_FLkM5emt63N73EcN  
53     localadmin:  
54         user: vitamdb-localadmin  
55         password: change_it_QeH8q4e16ah4QKXS  
56     system:  
57         user: vitamdb-system  
58         password: change_it_HycFEVD74g397iRe  
59     offer:  
60         user: offer  
61         password: change_it_pQi1T1yT9LAF8au8  
62 offer-fs-2:  
63     passphrase: changeiteSY1By57qZr4MX2s  
64     admin:  
65         user: vitamdb-admin  
66         password: change_it_84aTMFZ7h8e2NgMe
```

(suite sur la page suivante)

(suite de la page précédente)

```

67 localadmin:
68   user: vitamdb-localadmin
69   password: change_it_Am1B37tGY1w5VfvX
70 system:
71   user: vitamdb-system
72   password: change_it_HycFEVD74g397iRe
73 offer:
74   user: offer
75   password: change_it_mLDYds957sNQ53mA
76 offer-tape-1:
77   passphrase: changeitmB5rnk1M5TY61PqZ
78 admin:
79   user: vitamdb-admin
80   password: change_it_FLkM5emt63N73EcN
81 localadmin:
82   user: vitamdb-localadmin
83   password: change_it_QeH8q4e16ah4QKXS
84 system:
85   user: vitamdb-system
86   password: change_it_HycFEVD74g397iRe
87 offer:
88   user: offer
89   password: change_it_pQi1T1yT9LAF8au8
90 offer-swift-1:
91   passphrase: changeitgYvt42M2pKL6Zx3T
92 admin:
93   user: vitamdb-admin
94   password: change_it_e21hLp51WNa4sJFS
95 localadmin:
96   user: vitamdb-localadmin
97   password: change_it_QB8857SJrGrQh2yu
98 system:
99   user: vitamdb-system
100  password: change_it_HycFEVD74g397iRe
101 offer:
102   user: offer
103   password: change_it_AWJg2Bp3s69P6nMe
104 offer-s3-1:
105   passphrase: changeituF1jVdR9NqdTG625
106 admin:
107   user: vitamdb-admin
108   password: change_it_5b7cSWcS5M1NF4kv
109 localadmin:
110   user: vitamdb-localadmin
111   password: change_it_S9jE24rxHwUZP6y5
112 system:
113   user: vitamdb-system
114   password: change_it_HycFEVD74g397iRe
115 offer:
116   user: offer
117   password: change_it_TuTB1i2k7iQW3zL2
118 offer-tape-1:
119   passphrase: changeituF1jghT9NqdTG625
120 admin:
121   user: vitamdb-admin
122   password: change_it_5b7cSWcab91NF4kv
123 localadmin:

```

(suite sur la page suivante)

(suite de la page précédente)

```

124     user: vitamdb-localadmin
125     password: change_it_S9jE24rxHwUZP5a6
126 system:
127     user: vitamdb-system
128     password: change_it_HycFEVD74g397iRe
129 offer:
130     user: offer
131     password: change_it_TuTB1i2k7iQW3c2a
132
133 vitam_users:
134     - vitam_aadmin:
135         login: aadmin
136         password: change_it_z5MP7GC4qnR8nL9t
137         role: admin
138     - vitam_uuser:
139         login: uuser
140         password: change_it_w94Q3jPAT2aJYm8b
141         role: user
142     - vitam_gguest:
143         login: gguest
144         password: change_it_E5v7Tr4h6tYaQG2W
145         role: guest
146     - techadmin:
147         login: techadmin
148         password: change_it_K29E1uHcPZ8zXji8
149         role: admin
150
151 ldap_authentication:
152     ldap_pwd: "change_it_t69Rn5NdUv39EYkC"
153
154 admin_basic_auth_password: change_it_5Yn74JgXwbQ9KdP8
155
156 vitam_offers:
157     offer-swift-1:
158         swiftUser: swift_user
159         swiftPassword: password_change_m44j57aYeRPnPQ2
160     offer-s3-1:
161         s3AccessKey: accessKey_change_grLS8372Uga5EJSx
162         s3SecretKey: secretKey_change_p97es2m2CHXPJA1m

```

Prudence : Seuls les caractères alphanumériques sont valides pour les directives passphrase.

Avertissement : Le paramétrage du mode d'authentifications des utilisateurs à l'*IHM* démo est géré au niveau du fichier deployment/environments/group_vars/all/advanced/vitam_vars.yml. Plusieurs modes d'authentifications sont proposés au niveau de la section authentication_realms. Dans le cas d'une authentification se basant sur le mécanisme iniRealm (configuration shiro par défaut), les mots de passe déclarés dans la section vitam_users devront s'appuyer sur une politique de mot de passe robuste, comme indiqué en début de chapitre. Il est par ailleurs possible de choisir un mode d'authentification s'appuyant sur un annuaire LDAP externe (ldapRealm dans la section authentication_realms).

Note : Dans le cadre d'une installation avec au moins une offre *swift*, il faut déclarer, dans la section vitam_offers,

le nom de chaque offre et le mot de passe de connexion *swift* associé, défini dans le fichier `offers_opts.yml`. L'exemple ci-dessus présente la déclaration du mot de passe pour l'offre *offer-swift-1*.

Note : Dans le cadre d'une installation avec au moins une offre *s3*, il faut déclarer, dans la section `vitam_offers`, le nom de chaque offre et l'access key secret *s3* associé, défini dans le fichier `offers_opts.yml`. L'exemple ci-dessus présente la déclaration du mot de passe pour l'offre *s3 offer-s3-1*.

- Le fichier `repertoire_inventory/group_vars/all/main/vault-keystores.yml` contient les mots de passe des magasins de certificats utilisés dans VITAM :

```

1 # NO underscore ALLOWED IN VALUES
2 keystores:
3   server:
4     offer: changeit817NR75vWsZtgAgJ
5     access_external: changeitMZFD2YM4279miiu
6     ingest_external: changeita2C74cQhy84BLWCr
7     ihm_recette: changeit4FWYVK1347mxjGfe
8     ihm_demo: changeit6kQ16eyDY7QPS9fy
9     collect_external: changeit6kQ16eyDYAoPS9fy
10    client_external:
11      ihm_demo: changeitGT38hhTiA32x1PLy
12      gatling: changeit2sBC5ac7NfGF9Qj7
13      ihm_recette: changeitdAZ9Eq65UhDZd9p4
14      reverse: changeite5XTzb5yVPcEX464
15      vitam_admin_int: changeitz6xZe5gDu7nhDZd9
16      collect_external: changeitz6xZe5gDu7nhDZA12
17    client_storage:
18      storage: changeit647D7LWiM6qYMnm
19    timestamping:
20      secure_logbook: changeitMn9Skuyx87VYU62U
21      secure_storage: changeite5gDu9Skuy84BLW9
22  truststores:
23    server: changeitxNe4JLfn528PVHj7
24    client_external: changeitJ2eS93DcPH1v4jAp
25    client_storage: changeitHpSCa31aG8ttB87S
26  grantedstores:
27    client_external: changeitLL22HkmDCA2e2vj7
28    client_storage: changeitR3wwp5C8KQS76Vcu

```

Avertissement : Il convient de sécuriser votre environnement en définissant des mots de passe forts.

4.2.3.3.2 Cas des extras

- Le fichier `repertoire_inventory/group_vars/all/main/vault-extra.yml` contient les mots de passe des magasins de certificats utilisés dans VITAM :

```

1 # Example for git lfs ; uncomment & use if needed
2 #vitam_gitlab_itest_login: "account"
3 #vitam_gitlab_itest_password: "change_it_4DU42JVf2x2xmPBs"

```

Note : Le playbook vitam.yml comprend des étapes avec la mention no_log afin de ne pas afficher en clair des étapes comme les mots de passe des certificats. En cas d'erreur, il est possible de retirer la ligne dans le fichier pour une analyse plus fine d'un éventuel problème sur une de ces étapes.

4.2.3.3.3 Commande ansible-vault

Certains fichiers présents sous `repertoire_inventory/group_vars/all` commençant par **vault-** doivent être protégés (chiffrés) avec l'utilitaire ansible-vault.

Note : Ne pas oublier de mettre en conformité le fichier `vault_pass.txt`

4.2.3.3.1 Générer des fichiers **vaultés** depuis des fichiers en clair

Exemple du fichier `vault-cots.yml`

```
cp vault-cots.yml/plain vault-cots.yml  
ansible-vault encrypt vault-cots.yml
```

4.2.3.3.2 Re-chiffrer un fichier **vaulté** avec un nouveau mot de passe

Exemple du fichier `vault-cots.yml`

```
ansible-vault rekey vault-cots.yml
```

4.2.3.4 La configuration d'ElasticSearch

Les paramètres de configuration d'ElasticSearch sont configurables à travers le fichier de configuration, accessible plus spécifiquement dans le chemin suivant :

```
deployment/ansible-vitam/roles/vitam/files/elasticsearch-settings/  
elasticsearch-configuration.json
```

4.2.3.5 Le mapping ElasticSearch pour Unit et ObjectGroup

Les mappings des indexes elasticsearch pour les collections Unit et ObjectGroup sont configurables de l'extérieur, plus spécifiquement dans le dossier `environments/files/elasticsearch-mappings/`, ce dossier contient :

- `unit-es-mapping.json`
- `og-es-mapping.json`

Exemple du fichier mapping de la collection ObjectGroup :

```
1 {  
2     "dynamic_templates": [  
3         {  
4             "object": {  
5                 "match_mapping_type": "object",
```

(suite sur la page suivante)

(suite de la page précédente)

```
6     "mapping": {
7         "type": "object"
8     }
9 }
10 },
11 {
12     "all_string": {
13         "match": "*",
14         "mapping": {
15             "type": "text"
16         }
17     }
18 }
19 ],
20 "properties": {
21     "FileInfo": {
22         "properties": {
23             "CreatingApplicationName": {
24                 "type": "text"
25             },
26             "CreatingApplicationVersion": {
27                 "type": "text"
28             },
29             "CreatingOs": {
30                 "type": "text"
31             },
32             "CreatingOsVersion": {
33                 "type": "text"
34             },
35             "DateCreatedByApplication": {
36                 "type": "date",
37                 "format": "strict_date_optional_time"
38             },
39             "Filename": {
40                 "type": "text"
41             },
42             "LastModified": {
43                 "type": "date",
44                 "format": "strict_date_optional_time"
45             }
46         }
47     },
48     "Metadata": {
49         "properties": {
50             "Text": {
51                 "type": "object"
52             },
53             "Document": {
54                 "type": "object"
55             },
56             "Image": {
57                 "type": "object"
58             },
59             "Audio": {
60                 "type": "object"
61             },
62             "Video": {
63                 "type": "object"
64             }
65         }
66     }
67 }
```

(suite sur la page suivante)

(suite de la page précédente)

```

63         "type": "object"
64     }
65   }
66 },
67 "OtherMetadata": {
68   "type": "object",
69   "properties": {
70     "RawMetadata": {
71       "type": "object"
72     }
73   }
74 },
75 "_profil": {
76   "type": "keyword"
77 },
78 "_qualifiers": {
79   "properties": {
80     "_nbc": {
81       "type": "long"
82     },
83     "qualifier": {
84       "type": "keyword"
85     },
86     "versions": {
87       "type": "nested",
88       "properties": {
89         "Compressed": {
90           "type": "text"
91         },
92         "DataObjectGroupId": {
93           "type": "keyword"
94         },
95         "DataObjectVersion": {
96           "type": "keyword"
97         },
98         "DataObjectProfile": {
99           "type": "keyword"
100        },
101        "DataObjectSystemId": {
102          "type": "keyword"
103        },
104        "DataObjectGroupSystemId": {
105          "type": "keyword"
106        },
107        "_opi": {
108          "type": "keyword"
109        },
110        "FileInfo": {
111          "properties": {
112            "CreatingApplicationName": {
113              "type": "text"
114            },
115            "CreatingApplicationVersion": {
116              "type": "text"
117            },
118            "CreatingOs": {
119              "type": "text"

```

(suite sur la page suivante)

(suite de la page précédente)

```
120 },
121 "CreatingOsVersion": {
122     "type": "text"
123 },
124 "DateCreatedByApplication": {
125     "type": "date",
126     "format": "strict_date_optional_time"
127 },
128 "Filename": {
129     "type": "text"
130 },
131 "LastModified": {
132     "type": "date",
133     "format": "strict_date_optional_time"
134 }
135 }
136 },
137 "FormatIdentification": {
138     "properties": {
139         "FormatId": {
140             "type": "keyword"
141         },
142         "FormatLiteral": {
143             "type": "keyword"
144         },
145         "MimeType": {
146             "type": "keyword"
147         },
148         "Encoding": {
149             "type": "keyword"
150         }
151     }
152 },
153 "MessageDigest": {
154     "type": "keyword"
155 },
156 "Algorithm": {
157     "type": "keyword"
158 },
159 "PhysicalDimensions": {
160     "properties": {
161         "Diameter": {
162             "properties": {
163                 "unit": {
164                     "type": "keyword"
165                 },
166                 "dValue": {
167                     "type": "double"
168                 }
169             }
170         },
171         "Height": {
172             "properties": {
173                 "unit": {
174                     "type": "keyword"
175                 },
176                 "dValue": {
177                     "type": "double"
178                 }
179             }
180         }
181     }
182 }
```

(suite sur la page suivante)

(suite de la page précédente)

```
177         "type": "double"
178     }
179   }
180 },
181 "Depth": {
182   "properties": {
183     "unit": {
184       "type": "keyword"
185     },
186     "dValue": {
187       "type": "double"
188     }
189   }
190 },
191 "Shape": {
192   "type": "keyword"
193 },
194 "Thickness": {
195   "properties": {
196     "unit": {
197       "type": "keyword"
198     },
199     "dValue": {
200       "type": "double"
201     }
202   }
203 },
204 "Length": {
205   "properties": {
206     "unit": {
207       "type": "keyword"
208     },
209     "dValue": {
210       "type": "double"
211     }
212   }
213 },
214 "NumberOfPage": {
215   "type": "long"
216 },
217 "Weight": {
218   "properties": {
219     "unit": {
220       "type": "keyword"
221     },
222     "dValue": {
223       "type": "double"
224     }
225   }
226 },
227 "Width": {
228   "properties": {
229     "unit": {
230       "type": "keyword"
231     },
232     "dValue": {
233       "type": "double"
```

(suite sur la page suivante)

(suite de la page précédente)

```

234         }
235     }
236   }
237 }
238 },
239 "PhysicalId": {
240   "type": "keyword"
241 },
242 "Size": {
243   "type": "long"
244 },
245 "Uri": {
246   "type": "keyword"
247 },
248 "_id": {
249   "type": "keyword"
250 },
251 "_storage": {
252   "properties": {
253     "_nbc": {
254       "type": "long"
255     },
256     "offerIds": {
257       "type": "keyword"
258     },
259     "strategyId": {
260       "type": "keyword"
261     }
262   }
263 },
264 "PersistentIdentifier": {
265   "properties": {
266     "PersistentIdentifierType": {
267       "type": "keyword"
268     },
269     "PersistentIdentifierOrigin": {
270       "type": "keyword"
271     },
272     "PersistentIdentifierReference": {
273       "type": "keyword"
274     },
275     "PersistentIdentifierContent": {
276       "type": "keyword"
277     }
278   }
279 },
280 "DataObjectUse": {
281   "type": "keyword"
282 },
283 "DataObjectNumber": {
284   "type": "long"
285 },
286 },
287 }
288 },
289 "_v": {
290

```

(suite sur la page suivante)

(suite de la page précédente)

```

291     "type": "long"
292   },
293   "_av": {
294     "type": "long"
295   },
296   "_nbc": {
297     "type": "long"
298   },
299   "_ops": {
300     "type": "keyword"
301   },
302   "_opi": {
303     "type": "keyword"
304   },
305   "_batchId": {
306     "type": "keyword"
307   },
308   "_sp": {
309     "type": "keyword"
310   },
311   "_sps": {
312     "type": "keyword"
313   },
314   "_tenant": {
315     "type": "long"
316   },
317   "_up": {
318     "type": "keyword"
319   },
320   "_us": {
321     "type": "keyword"
322   },
323   "_storage": {
324     "properties": {
325       "_nbc": {
326         "type": "long"
327       },
328       "offerIds": {
329         "type": "keyword"
330       },
331       "strategyId": {
332         "type": "keyword"
333       }
334     }
335   },
336   "_glpd": {
337     "enabled": false
338   },
339   "_acd": {
340     "type": "date",
341     "format": "strict_date_optional_time"
342   },
343   "_aud": {
344     "type": "date",
345     "format": "strict_date_optional_time"
346   },
347   "_errors": {

```

(suite sur la page suivante)

(suite de la page précédente)

```

348 "properties": {
349   "evId": {
350     "type": "keyword"
351   },
352   "obId": {
353     "type": "keyword"
354   },
355   "evTypeProc": {
356     "type": "keyword"
357   },
358   "outDetail": {
359     "type": "keyword"
360   },
361   "outMessg": {
362     "type": "text"
363   },
364   "evDetData": {
365     "type": "keyword"
366   }
367 }
368 }
369 }
370 }

```

Note : Le paramétrage de ce mapping se fait sur les composants metadata, metadata_collect et le composant extra ihm-recette.

Prudence : En cas de changement du mapping, il faut veiller à ce que cette mise à jour soit en accord avec l'Ontologie de [VITAM](#). Le mapping est pris en compte lors de la première création des indexes. Pour une nouvelle installation de [VITAM](#), les mappings seront automatiquement pris en compte.

Note : Une modification de ces mappings après installation peut-être faite mais nécessitera de rejouer les playbooks metadata et metadata_collect avant de réindexer : ansible-vitam/services/vitam/metadata.yml ansible-vitam/services/vitam/metadata_collect.yml ansible-vitam-exploitation/reindex_es_data.yml

4.2.4 Gestion des certificats

Une vue d'ensemble de la gestion des certificats est présentée [dans l'annexe dédiée](#) (page 125).

4.2.4.1 Cas 1 : Configuration développement / tests

Pour des usages de développement ou de tests hors production, il est possible d'utiliser la [PKI](#) fournie avec la solution logicielle [VITAM](#).

4.2.4.1.1 Procédure générale

Danger : La *PKI* fournie avec la solution logicielle *VITAM* doit être utilisée UNIQUEMENT pour faire des tests, et ne doit par conséquent surtout pas être utilisée en environnement de production ! De plus il n'est pas possible de l'utiliser pour générer les certificats d'une autre application qui serait cliente de VITAM.

La *PKI* de la solution logicielle *VITAM* est une suite de scripts qui vont générer dans l'ordre ci-dessous :

- Les autorités de certification (*CA*)
- Les certificats (clients, serveurs, de *timestamping*) à partir des *CA*
- Les *keystores*, en important les certificats et *CA* nécessaires pour chacun des *keystores*

4.2.4.1.2 Génération des CA par les scripts Vitam

Il faut faire la génération des autorités de certification (*CA*) par le script décrit ci-dessous.

Dans le répertoire de déploiement, lancer le script :

```
pki/scripts/generate_ca.sh
```

Ce script génère sous *pki/ca* les autorités de certification *root* et intermédiaires pour générer des certificats clients, serveurs, et de timestamping. Les mots de passe des clés privées des autorités de certification sont stockés dans le vault ansible environments/certs/vault-ca.yml

Avertissement : Il est impératif de noter les dates de création et de fin de validité des CA. En cas d'utilisation de la PKI fournie, la CA root a une durée de validité de 10 ans ; la CA intermédiaire a une durée de 3 ans.

4.2.4.1.3 Génération des certificats par les scripts Vitam

Le fichier d'inventaire de déploiement environments/<fichier d'inventaire> (cf. *Informations plate-forme* (page 22)) doit être correctement renseigné pour indiquer les serveurs associés à chaque service. En prérequis les *CA* doivent être présentes.

Puis, dans le répertoire de déploiement, lancer le script :

```
pki/scripts/generate_certs.sh <fichier d'inventaire>
```

Ce script génère sous environments/certs les certificats (format crt & key) nécessaires pour un bon fonctionnement dans VITAM. Les mots de passe des clés privées des certificats sont stockés dans le vault ansible environments/certs/vault-certs.yml.

Prudence : Les certificats générés à l'issue ont une durée de validité de 3 ans.

4.2.4.2 Cas 2 : Configuration production

4.2.4.2.1 Procédure générale

La procédure suivante s'applique lorsqu'une *PKI* est déjà disponible pour fournir les certificats nécessaires.

Les étapes d'intégration des certificats à la solution *Vitam* sont les suivantes :

- Générer les certificats avec les bons *key usage* par type de certificat
- Déposer les certificats et les autorités de certifications correspondantes dans les bons répertoires.
- Renseigner les mots de passe des clés privées des certificats dans le vault ansible environments/certs/vault-certs.yml
- Utiliser le script VITAM permettant de générer les différents *keystores*.

Note : Rappel pré-requis : vous devez disposer d'une ou plusieurs *PKI* pour tout déploiement en production de la solution logicielle *VITAM*.

4.2.4.2.2 Génération des certificats

En conformité avec le document RGSV2 de l'ANSSI, il est recommandé de générer des certificats avec les caractéristiques suivantes :

4.2.4.2.2.1 Certificats serveurs

- **Key Usage**
 - digitalSignature, keyEncipherment
- **Extended Key Usage**
 - TLS Web Server Authentication

Les certificats serveurs générés doivent prendre en compte des alias « web » (subjectAltName).

Le *subjectAltName* des certificats serveurs (deployment/environments/certs/server/hosts/*) doit contenir le nom DNS du service sur consul associé.

Exemple avec un cas standard : <composant_vitam>.service.<consul_domain>. Ce qui donne pour le certificat serveur de access-external par exemple :

```
X509v3 Subject Alternative Name:
DNS:access-external.service.consul, DNS:localhost
```

Il faudra alors mettre le même nom de domaine pour la configuration de Consul (fichier deployment/environments/group_vars/all/advanced/vitam_vars.yml, variable *consul_domain*)

Cas particulier pour ihm-demo et ihm-recette : il faut ajouter le nom *DNS* qui sera utilisé pour requêter ces deux applications, si celles-ci sont appelées directement en frontal https.

4.2.4.2.2.2 Certificat clients

- **Key Usage**
 - digitalSignature
- **Extended Key Usage**
 - TLS Web Client Authentication

4.2.4.2.2.3 Certificats d'horodatage

Ces certificats sont à générer pour les composants logbook et storage.

- **Key Usage**

- digitalSignature, nonRepudiation

- **Extended Key Usage**

- Time Stamping

4.2.4.2.3 Intégration de certificats existants

Une fois les certificats et *CA* mis à disposition par votre *PKI*, il convient de les positionner sous environments/certs/.... en respectant la structure indiquée ci-dessous.

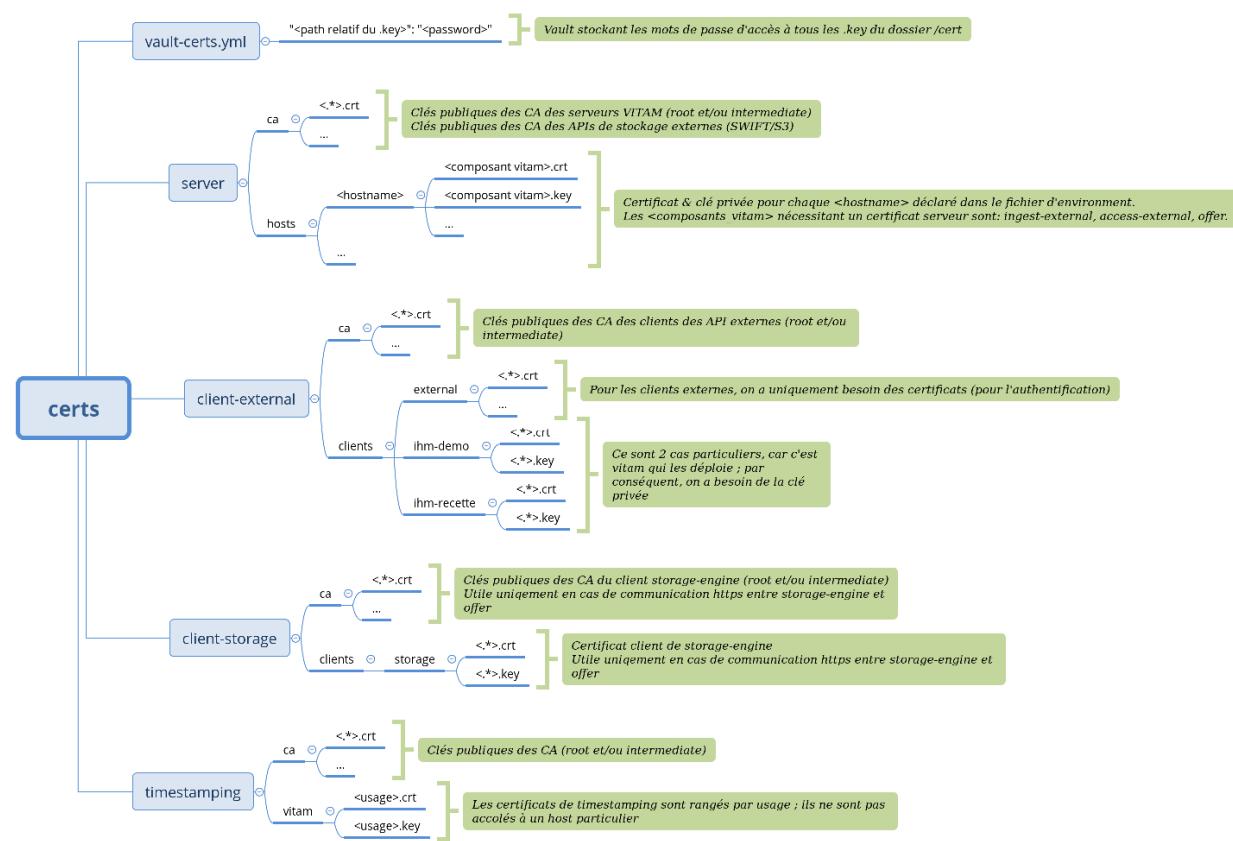


FIG. 3 – Vue détaillée de l'arborescence des certificats

Astuce : Dans le doute, n'hésitez pas à utiliser la *PKI* de test (étapes de génération de *CA* et de certificats) pour générer les fichiers requis au bon endroit et ainsi observer la structure exacte attendue ; il vous suffira ensuite de remplacer ces certificats « placeholders » par les certificats définitifs avant de lancer le déploiement.

Ne pas oublier de renseigner le vault contenant les *passphrases* des clés des certificats : environments/certs/vault-certs.yml

Pour modifier/créer un vault ansible, se référer à la documentation Ansible sur [cette url](#)¹⁴.

Prudence : Durant l'installation de VITAM, il est nécessaire de créer un certificat « vitam-admin-int » (à placer sous deployment/environments/certs/client-external/clients/vitam-admin-int).

Prudence : Durant l'installation des extra de VITAM, il est nécessaire de créer un certificat « gatling » (à placer sous deployment/environments/certs/client-external/clients/gatling).

4.2.4.2.4 Intégration de certificats clients de VITAM

4.2.4.2.4.1 Intégration d'une application externe (cliente)

Dans le cas d'ajout de certificats *SIA* externes au déploiement de la solution logicielle *VITAM* :

- Déposer le certificat (.crt) de l'application client dans environments/certs/client-external/clients/external/
- Déposer les *CA* du certificat de l'application (.crt) dans environments/certs/client-external/ca/
- Editer le fichier environments/group_vars/all/advanced/vitam_security.yml et ajouter le(s) entrée(s) supplémentaire(s) (sous forme répertoire/fichier.crt, exemple : external/mon_sia.crt) dans la directive admin_context_certs pour que celles-ci soient associés aux contextes de sécurité durant le déploiement de la solution logicielle *VITAM*.

Note : Les certificats *SIA* externes ajoutés par le mécanisme de déploiement sont, par défaut, rattachés au contexte applicatif d'administration admin_context_name lui-même associé au profil de sécurité admin_security_profile et à la liste de tenants vitam_tenant_ids (voir le fichier environments/group_vars/all/advanced/vitam_security.yml). Pour l'ajout de certificats applicatifs associés à des contextes applicatifs autres, se référer à la procédure du document d'exploitation (*DEX*) décrivant l'intégration d'une application externe dans Vitam.

4.2.4.2.4.2 Intégration d'un certificat personnel (personae)

Dans le cas d'ajout de certificats personnels au déploiement de la solution logicielle *VITAM* :

- Déposer le certificat personnel (.crt) dans environments/certs/client-external/clients/external/
- Editer le fichier environments/group_vars/all/advanced/vitam_security.yml et ajouter le(s) entrée(s) supplémentaire(s) (sous forme répertoire/fichier.crt, exemple : external/mon_personae.crt) dans la directive admin_personal_certs pour que ceux-ci soient ajoutés à la base de données du composant security-internal durant le déploiement de la solution logicielle *VITAM*.

4.2.4.2.5 Cas des offres objet

Placer le .crt de la *CA* dans deployment/environments/certs/server/ca.

14. http://docs.ansible.com/ansible/playbooks_vault.html

4.2.4.2.6 Absence d'usage d'un reverse

Dans ce cas, il convient de :

- supprimer le répertoire deployment/environments/certs/client-external/clients/reverse
- supprimer les entrées **reverse** dans le fichier vault_keystore.yml

4.2.4.3 Intégration de CA pour une offre *Swift* ou *s3*

En cas d'utilisation d'une offre *Swift* ou *s3* en https, il est nécessaire d'ajouter les **CA** du certificat de l'*API Swift* ou *s3*.

Il faut les déposer dans environments/certs/server/ca/ avant de jouer le script ./generate_keystores.sh

4.2.4.4 Génération des magasins de certificats

En prérequis, les certificats et les autorités de certification (**CA**) doivent être présents dans les répertoires attendus.

Prudence : Avant de lancer le script de génération des *stores*, il est nécessaire de modifier le vault contenant les mots de passe des *stores* : environments/group_vars/all/main/vault_keystores.yml, décrit dans la section *Déclaration des secrets* (page 53).

Lancer le script : ./generate_stores.sh

Ce script génère sous environments/keystores les *stores* (aux formats jks / p12) associés pour un bon fonctionnement dans la solution logicielle **VITAM**.

Il est aussi possible de déposer directement les *keystores* au bon format en remplaçant ceux fournis par défaut et en indiquant les mots de passe d'accès dans le vault : environments/group_vars/all/main/vault_keystores.yml

Note : Le mot de passe du fichier vault_keystores.yml est identique à celui des autres *vaults* ansible.

4.2.5 Paramétrages supplémentaires

4.2.5.1 Tuning JVM

Prudence : En cas de colocalisation, bien prendre en compte la taille **JVM** de chaque composant (VITAM : -Xmx512m par défaut) pour éviter de *swapper*.

Un *tuning* fin des paramètres **JVM** de chaque composant **VITAM** est possible. Pour cela, il faut modifier le contenu du fichier deployment/environments/group_vars/all/main/jvm_opts.yml

Pour chaque composant, il est possible de modifier ces 3 variables :

- memory : paramètres Xms et Xmx
- gc : paramètres gc
- java : autres paramètres java

4.2.5.2 Installation en mode conteneur

Note : Fonctionnalité disponible partir de la V7.1 .

Prudence : Ce mode de déploiement est en mode bêta , merci de ne pas l'appliquer dans un environnement de production

Il est possible de déployer vitam en mode conteneur. Pour cela, il faut éditer le contenu du fichier environments/group_vars/all/main/repositories.yml. Pour cela il faut rajouter les paramètres présentés dans l'exemple :

Exemple :

```
install_mode: container

container_repository:
  registry_url: <url de la registry docker>
  username: <login>
  password: <password>
```

Avertissement : Dans le cas d'utilisation d'une registry interne il vous faudra effectuer une synchronisation à partir de la registry docker du programme Vitam : docker.programmevitam.fr

4.2.5.3 Installation des *griffins* (greffons de préservation)

Note : Fonctionnalité disponible partir de la R9 (2.1.1) .

Prudence : Cette version de *VITAM* ne mettant pas encore en oeuvre de mesure d'isolation particulière des *griffins*, il est recommandé de veiller à ce que l'usage de chaque *griffin* soit en conformité avec la politique de sécurité de l'entité. Il est en particulier déconseillé d'utiliser un griffon qui utiliserait un outil externe qui n'est plus maintenu.

Il est possible de choisir les *griffins* installables sur la plate-forme. Pour cela, il faut éditer le contenu du fichier deployment/environments/group_vars/all/main.yml au niveau de la directive vitam_griffins. Cette action est à rapprocher de l'incorporation des binaires d'installation : les binaires d'installation des greffons doivent être accessibles par les machines hébergeant le composant **worker**.

Exemple :

```
vitam_griffins: ["vitam-imagemagick-griffin", "vitam-jhove-griffin"]
```

Voici la liste des greffons disponibles au moment de la présente publication :

```
vitam-imagemagick-griffin
vitam-jhove-griffin
vitam-libreoffice-griffin
vitam-odfvalidator-griffin
```

(suite sur la page suivante)

(suite de la page précédente)

```
vitam-siegfried-griffin  
vitam-tesseract-griffin  
vitam-verapdf-griffin  
vitam-ffmpeg-griffin
```

Avertissement : Ne pas oublier d'avoir déclaré au préalable sur les machines cibles le dépôt de binaires associé aux *griffins*.

4.2.5.4 Rétention liée aux logback

La solution logicielle **VITAM** utilise logback pour la rotation des log, ainsi que leur rétention.

Il est possible d'appliquer un paramétrage spécifique pour chaque composant VITAM.

Éditer le fichier `deployment/environments/group_vars/all/advanced/vitam_vars.yml` (et `extra_vars.yml`, dans le cas des extra) et appliquer le paramétrage dans le bloc `logback_total_size_cap` de chaque composant sur lequel appliquer la modification de paramétrage. Pour chaque **APPENDER**, la valeur associée doit être exprimée en taille et unité (exemple : 14GB ; représente 14 gigabytes).

Note : des *appenders* supplémentaires existent pour le composant storage-engine (appender `offersync`) et offer (offer_tape et offer_tape_backup).

4.2.5.4.1 Cas des accesslog

Il est également possible d'appliquer un paramétrage différent par composant VITAM sur le logback *access*.

Éditer le fichier `deployment/environments/group_vars/all/advanced/vitam_vars.yml` (et `extra_vars.yml`, dans le cas des extra) et appliquer le paramétrage dans les directives `access_retention_days` et `access_total_size_Gb` de chaque composant sur lequel appliquer la modification de paramétrage.

4.2.5.5 Paramétrage de l'antivirus (ingest-external)

L'antivirus utilisé par ingest-external est modifiable (par défaut, ClamAV) ; pour cela :

- Éditer la variable `vitam.ingestexternal.antivirus` dans le fichier `deployment/environments/group_vars/all/advanced/vitam_vars.yml` pour indiquer le nom de l'antivirus à utiliser.
- Créer un script shell (dont l'extension doit être .sh) sous `environments/antivirus/` (norme : `scan-<vitam.ingestexternal.antivirus>.sh`) ; prendre comme modèle le fichier `scan-clamav.sh`. Ce script shell doit respecter le contrat suivant :
 - Argument : chemin absolu du fichier à analyser
 - **Sémantique des codes de retour**
 - 0 : Analyse OK - pas de virus
 - 1 : Analyse OK - virus trouvé et corrigé
 - 2 : Analyse OK - virus trouvé mais non corrigé
 - 3 : Analyse NOK

- **Contenu à écrire dans stdout / stderr**

- stdout : Nom des virus trouvés, un par ligne ; Si échec (code 3) : raison de l'échec
- stderr : Log « brut » de l'antivirus

Prudence : En cas de remplacement de clamAV par un autre antivirus, l'installation de celui-ci devient dès lors un prérequis de l'installation et le script doit être testé.

Avertissement : Il subsiste une limitation avec l'antivirus ClamAV qui n'est actuellement pas capable de scanner des fichiers > 4Go. Ainsi, il n'est pas recommandé de conserver cet antivirus en environnement de production.

Avertissement : Sur plate-forme Debian, ClamAV est installé sans base de données. Pour que l'antivirus soit fonctionnel, il est nécessaire, durant l'installation, de le télécharger ; il est donc nécessaire de renseigner dans l'inventaire la directive `http_proxy_environnement` ou de renseigner un [miroir local privé](#)¹⁵).

4.2.5.5.1 Extra : Avast Business Antivirus for Linux

Note : Avast étant un logiciel soumis à licence, Vitam ne fournit pas de support ni de licence nécessaire à l'utilisation de Avast Antivirus for Linux.

Vous trouverez plus d'informations sur le site officiel : [Avast Business Antivirus for Linux](#)¹⁶

À la place de clamAV, il est possible de déployer l'antivirus **Avast Business Antivirus for Linux**.

Pour se faire, il suffit d'éditer la variable `vitam.ingestexternal.antivirus: avast` dans le fichier `deployment/environments/group_vars/all/advanced/vitam_vars.yml`.

Il sera nécessaire de fournir le fichier de licence sous `deployment/environments/antivirus/license.avastlic` pour pouvoir déployer et utiliser l'antivirus Avast.

De plus, il est possible de paramétriser l'accès aux repositories (Packages & Virus definitions database) dans le fichier `deployment/environments/group_vars/all/advanced/cots_vars.yml`.

Si les paramètres ne sont pas définis, les valeurs suivantes sont appliquées par défaut.

```
## Avast Business Antivirus for Linux
## if undefined, the following default values are applied.

avast:
  # logs configuration
  logrotate: enabled # or disabled
  history_days: 30 # How many days to store logs if logrotate is set to 'enabled'
  manage_repository: true
  repository:
    state: present
    # For RedHat family
    baseurl: https://repo.avcdn.net/linux-av/rpm/e19/release
    gpgcheck: no
    proxy: _none_
```

(suite sur la page suivante)

15. <https://www.clamav.net/documents/private-local-mirrors>

16. <https://www.avast.com/fr-fr/business/products/linux-antivirus>

(suite de la page précédente)

```
# For Debian family
baseurl: 'deb https://repo.avcdn.net/linux-av/deb debian-bookworm release'
vps_repository: http://linux-av.u.avcdn.net/linux-av/avast/x86_64
## List of sha256 hash of excluded files from antivirus. Useful for test environments.
whitelist:
- <EMPTY>
```

Avertissement : Vitam gère en entrée les SIPs aux formats : ZIP ou TAR (tar, tar.gz ou tar.bz2); cependant et d'après les tests effectués, il est fortement recommandé d'utiliser le format .zip pour bénéficier des meilleures performances d'analyses avec le scan-avast.sh.

De plus, il faudra prendre en compte un dimensionnement supplémentaire sur les ingest-external afin de pouvoir traiter le scan des fichiers >500Mo.

Dans le cas d'un SIP au format .zip ou .tar, les fichiers >500Mo contenus dans le SIP seront décompressés et scannés unitairement. Ainsi la taille utilisée ne dépassera pas la taille d'un fichier.

Dans le cas d'un SIP au format .tar.gz ou .tar.bz2, les SIPs >500Mo seront intégralement décompressés et scannés. Ainsi, la taille utilisée correspondra à la taille du SIP décompressé.

4.2.5.6 Paramétrage des certificats externes (*-externe)

Se reporter au chapitre dédié à la gestion des certificats : *Gestion des certificats* (page 65)

4.2.5.7 Placer « hors Vitam » le composant ihm-demo

Sous deployment/environments/host_vars, créer ou éditer un fichier nommé par le nom de machine qui héberge le composant ihm-demo et ajouter le contenu ci-dessous :

```
consul_disabled : true
```

Il faut également modifier le fichier deployment/environments/group_vars/all/advanced/vitam_vars.yml en remplaçant :

- dans le bloc accessexternal, la directive host: "access-external.service.{{ consul_domain }}" par host: "<adresse IP de access-external>" (l'adresse IP peut être une *FIP*)
- dans le bloc ingestexternal, la directive host: "ingest-external.service.{{ consul_domain }}" par host: "<adresse IP de ingest-external>" (l'adresse IP peut être une *FIP*)

A l'issue, le déploiement n'installera pas l'agent Consul. Le composant ihm-demo appellera, alors, par l'adresse *IP* de service les composants « access-external » et « ingest-external ».

Il est également fortement recommandé de positionner la valeur de la directive vitam.ihm_demo.metrics_enabled à false dans le fichier deployment/environments/group_vars/all/advanced/vitam_vars.yml, afin que ce composant ne tente pas d'envoyer des données sur « elasticsearch-log ».

4.2.5.8 Paramétrier le secure_cookie pour ihm-demo

Le composant ihm-demo (ainsi qu'ihm-recette) dispose d'une option supplémentaire, par rapport aux autres composants VITAM, dans le fichier deployment/environments/group_vars/all/advanced/vitam_vars.

yml : le `secure_cookie` qui permet de renforcer ces deux *IHM* contre certaines attaques assez répandues comme les CSRF (Cross-Site Request Forgery).

Il faut savoir que si cette variable est à *true* (valeur par défaut), le client doit obligatoirement se connecter en https sur l'*IHM*, et ce même si un reverse proxy se trouve entre le serveur web et le client.

Cela peut donc obliger le reverse proxy frontal de la chaîne d'accès à écouter en https.

4.2.5.9 Paramétrage de la centralisation des logs VITAM

2 cas sont possibles :

- Utiliser le sous-système de gestion des logs fourni par la solution logicielle *VITAM* ;
- Utiliser un *SIEM* tiers.

4.2.5.9.1 Gestion par VITAM

Pour une gestion des logs par *VITAM*, il est nécessaire de déclarer les serveurs ad-hoc dans le fichier d'inventaire pour les 3 groupes :

- hosts_logstash
- hosts_kibana_log
- hosts_elasticsearch_log

4.2.5.9.2 Redirection des logs sur un SIEM tiers

En configuration par défaut, les logs VITAM sont tout d'abord routés vers un serveur rsyslog installé sur chaque machine. Il est possible d'en modifier le routage, qui par défaut redirige vers le serveur logstash, via le protocole syslog en TCP.

Pour cela, il est nécessaire de placer un fichier de configuration dédié dans le dossier `/etc/rsyslog.d/`; ce fichier sera automatiquement pris en compte par rsyslog. Pour la syntaxe de ce fichier de configuration rsyslog, se référer à la documentation rsyslog¹⁷.

Astuce : Pour cela, il peut être utile de s'inspirer du fichier de référence *VITAM* `deployment/ansible-vitam/roles/rsyslog/templates/vitam_transport.conf.j2` (attention, il s'agit d'un fichier template ansible, non directement convertible en fichier de configuration sans en ôter les directives jinja2).

4.2.5.10 Passage des identifiants des référentiels en mode esclave

La génération des identifiants des référentiels est gérée par *VITAM* lorsqu'il fonctionne en mode maître.

Par exemple :

- Préfixé par PR- pour les profils
- Préfixé par IC- pour les contrats d'entrée
- Préfixé par AC- pour les contrats d'accès

Depuis la version 1.0.4, la configuration par défaut de *VITAM* autorise des identifiants externes (ceux qui sont dans le fichier json importé).

- pour le tenant 0 pour les référentiels : contrat d'entrée et contrat d'accès.

17. <http://www.rsyslog.com/doc/v7-stable/>

- pour le tenant 1 pour les référentiels : contrat d'entrée, contrat d'accès, profil, profil de sécurité et contexte.

La liste des choix possibles, pour chaque tenant, est :

TABLEAU 1: Description des identifiants de référentiels

Nom du référentiel	Description
INGEST_CONTRACT	contrats d'entrée
ACCESS_CONTRACT	contrats d'accès
PROFILE	profils <i>SEDA</i>
SECURITY_PROFILE	profils de sécurité (utile seulement sur le tenant d'administration)
CONTEXT	contextes applicatifs (utile seulement sur le tenant d'administration)
ARCHIVEUNITPROFILE	profils d'unités archivistiques

Si vous souhaitez gérer vous-même les identifiants sur un service référentiel, il faut qu'il soit en mode esclave.

Par défaut tous les services référentiels de Vitam fonctionnent en mode maître. Pour désactiver le mode maître de *VITAM*, il faut modifier le fichier ansible deployment/environments/group_vars/all/main/main.yml dans les sections vitam_tenants_usage_external (pour gérer, par tenant, les collections en mode esclave).

4.2.5.11 Paramétrage du batch de calcul pour l'indexation des règles héritées

La paramétrage du batch de calcul pour l'indexation des règles héritées peut être réalisé dans le fichier deployment/environments/group_vars/all/advanced/vitam_vars.yml.

La section suivante du fichier vitam_vars.yml permet de paramétrer la fréquence de passage du batch :

```
vitam_timers:
  metadata:
    - name: vitam-metadata-computed-inherited-rules
      frequency: "*-*-* 02:30:00"
```

La section suivante du fichier vitam_vars.yml permet de paramétrer la liste des tenants sur lesquels s'exécute le batch :

```
vitam:
  worker:
    # api_output_index_tenants : permet d'indexer les règles de gestion, les_
    ↪chemins des règles et les services producteurs
    api_output_index_tenants: [0,1,2,3,4,5,6,7,8,9]
    # rules_index_tenants : permet d'indexer les règles de gestion
    rules_index_tenants: [0,1,2,3,4,5,6,7,8,9]
```

4.2.5.12 Durées minimales permettant de contrôler les valeurs saisies

Afin de se prémunir contre une alimentation du référentiel des règles de gestion avec des durées trop courtes susceptibles de déclencher des actions indésirables sur la plate-forme (ex. éliminations) – que cette tentative soit intentionnelle ou non –, la solution logicielle *VITAM* vérifie que l'association de la durée et de l'unité de mesure saisies pour chaque champ est supérieure ou égale à une durée minimale définie lors du paramétrage de la plate-forme, dans un fichier de configuration.

Pour mettre en place le comportement attendu par le métier, il faut modifier le contenu de la directive vitam_tenant_rule_duration dans le fichier ansible deployment/environments/group_vars/all/advanced/vitam_vars.yml.

Exemple :

```

vitam_tenant_rule_duration:
  - name: 2 # applied tenant
    rules:
      - AppraisalRule : "1 year" # rule name : rule value
  - name: 3
    rules:
      AppraisaleRule : "5 year"
      StorageRule : "5 year"
      ReuseRule : "2 year"

```

Par *tenant*, les directives possibles sont :

TABLEAU 2: Description des règles

Règle	Valeur par défaut
AppraisalRule	
DisseminationRule	
StorageRule	
ReuseRule	
AccessRule	0 year
ClassificationRule	

Les valeurs associées sont une durée au format <nombre> <unité en anglais, au singulier>

Exemples :

6 month 1 year 5 year

Voir aussi :

Pour plus de détails, se rapporter à la documentation métier « Règles de gestion ».

4.2.5.13 Augmenter la précision sur le nombre de résultats retournés dépassant 10000

Suite à une évolution d'ElasticSearch (à partir de la version 7.6), le nombre maximum de résultats retournés est limité à 10000. Ceci afin de limiter la consommation de ressources sur le cluster elasticsearch.

Pour permettre de retourner le nombre exact de résultats, il est possible d'éditer le paramètre vitam.accessexternal.authorizeTrackTotalHits dans le fichier de configuration environments/group_vars/all/vitam_vars.yml

Il sera nécessaire de réappliquer la configuration sur le groupe hosts_access_external :

```
ansible-playbook ansible-vitam/vitam.yml --limit hosts_access_external --tags update_
  ↵vitam_configuration -i environments/hosts.<environnement> --ask-vault-pass
```

Ensuite, si l'API de recherche utilise le type d'entrée de DSL « SELECT MULTIPLE », il faut ajouter \$track_total_hits : true au niveau de la partie « filter » de la requête d'entrée.

Ci-dessous, un exemple de requête d'entrée :

```
{
  "$roots": [],
  "$query": [
    {
      "$match": {
        "Title": "héritage"
      }
    }
  ]
}
```

(suite sur la page suivante)

(suite de la page précédente)

```

        }
    ],
    "$filter": {
        "$offset": 0,
        "$limit": 100,
        "$track_total_hits": true
    },
    "$projection": {}
}

```

4.2.5.14 Fichiers complémentaires

A titre informatif, le positionnement des variables ainsi que des dérivations des déclarations de variables sont effectuées dans les fichiers suivants :

- deployment/environments/group_vars/all/main.yml, comme suit :

```

1  ---
2
3  # TENANTS
4  # List of active tenants
5  vitam_tenant_ids: [0,1,2,3,4,5,6,7,8,9]
6  # For functional-administration, manage master/slave tenant configuration
7  # http://www.programmavita.m.fr/ressources/DocCourante/html/installation/
8  # installation/21-addons.html#passage-des-identifiants-des-referentiels-en-mode-
9  # esclave
10 vitam_tenants_usage_external:
11   - name: 0
12     identifiers:
13       - INGEST_CONTRACT
14       - ACCESS_CONTRACT
15       - MANAGEMENT_CONTRACT
16       - ARCHIVE_UNIT_PROFILE
17   - name: 1
18     identifiers:
19       - INGEST_CONTRACT
20       - ACCESS_CONTRACT
21       - MANAGEMENT_CONTRACT
22       - PROFILE
23       - SECURITY_PROFILE
24       - CONTEXT
25
26  # GRIFFINS
27  # Vitam griffins required to launch preservation scenario
28  # Example:
29  # vitam_griffins: ["vitam-imagemagick-griffin", "vitam-libreoffice-griffin",
30  #   "vitam-jhove-griffin", "vitam-odfvalidator-griffin", "vitam-siegfried-griffin",
31  #   "vitam-tesseract-griffin", "vitam-verapdf-griffin", "vitam-ffmpeg-griffin"]
32  vitam_griffins: []
33
34  # CONSUL
35  consul:
36    network: "ip_admin" # Which network to use for consul communications ? ip_admin_
37    # or ip_service ?
38  consul_remote_sites:
39    # wan contains the wan addresses of the consul server instances of the external_
40    # vitam sites

```

(suite sur la page suivante)

(suite de la page précédente)

```

35 # Exemple, if our local dc is dc1, we will need to set dc2 & dc3 wan conf:
36 #   - dc2:
37 #     wan: ["10.10.10.10", "1.1.1.1"]
38 #   - dc3:
39 #     wan: ["10.10.10.11", "1.1.1.1"]
40
41 # LOGGING
42 # vitam_defaults:
43 #   access_retention_days: 365 # Number of days for accesslog retention
44 #   access_total_size_cap: "5GB" # total acceptable size
45 #   logback_max_file_size: "10MB"
46 #   logback_total_size_cap:
47 #     file:
48 #       history_days: 365
49 #       totalsize: "5GB"
50 #     security:
51 #       history_days: 365
52 #       totalsize: "5GB"
53
54 # ELASTICSEARCH
55 # 'number_of_shards': number of shards per index, every ES shard is stored as a lucene index
56 # 'number_of_replicas': number of additional copies of primary shards
57 # Total number of shards: number_of_shards * (1 primary + M number_of_replicas)
58 # CAUTION: The total number of shards should be lower than or equal to the number of elasticsearch-data instances in the cluster
59 # More details in groups_vars/all/advanced/tenants_vars.yml file
60 vitam_elasticsearch_tenant_indexation:
61   default_config:
62     # Default settings for masterdata collections (1 index per collection)
63     masterdata:
64       number_of_shards: 1
65       number_of_replicas: 2
66     # Default settings for unit indexes (1 index per tenant)
67     unit:
68       number_of_shards: 1
69       number_of_replicas: 2
70     # Default settings for object group indexes (1 index per tenant)
71     objectgroup:
72       number_of_shards: 1
73       number_of_replicas: 2
74     # Default settings for logbook operation indexes (1 index per tenant)
75     logbookoperation:
76       number_of_shards: 1
77       number_of_replicas: 2
78     # Default settings for collect_unit indexes
79     collect_unit:
80       number_of_shards: 1
81       number_of_replicas: 2
82     # Default settings for collect_objectgroup indexes
83     collect_objectgroup:
84       number_of_shards: 1
85       number_of_replicas: 2
86
87     collect_grouped_tenants:
88     - name: 'all'
89       # Group all tenants for collect's indexes (collect_unit & collect_objectgroup)

```

(suite sur la page suivante)

(suite de la page précédente)

```

90   tenants: "{{ vitam_tenant_ids | join(',') }}"
91
92 elasticsearch:
93   log:
94     index_templates:
95       default:
96         shards: 1
97         replica: 1
98   data:
99     index_templates:
100       default:
101         shards: 1
102         replica: 2
103
104 curator:
105   indices:
106     vitam:
107       close: 30
108       delete: 365
109     access:
110       close: 30
111       delete: 180
112     system:
113       close: 7
114       delete: 30

```

Note : Installation multi-sites. Déclarer dans `consul_remote_sites` les datacenters Consul des autres site; se référer à l'exemple fourni pour renseigner les informations.

- `deployment/environments/group_vars/all/advanced/vitam_vars.yml`, comme suit :

```

1  -----
2  ### global ###
3
4  # Vitam deployment mode. Allowed values are :
5  # - "prod" (default): Enforces additional security checks (disallow development/
6  #   ↵debug tools, reverse proxy does NOT forward traffic to vitam service ports...)
7  # - "dev" (NOT for sensitive / production environments): Allow development/debug
8  #   ↵tools, reverse proxy forwards traffic to vitam service ports.
9  deployment_mode: prod
10
11 # TODO MAYBE : permettre la surcharge avec une syntax du genre vitamopts.folder_
12 #   ↵root / default(vitam_default.folder_root) dans les templates ?
13 droid_filename: "DROID_SignatureFile_V109.xml"
14 droid_container_filename: "container-signature-20221102.xml"
15
16 # The global defaults parameters for vitam & vitam components
17 vitam_defaults:
18   folder:
19     root_path: /vitam
20     folder_permission: "0750"
21     conf_permission: "0440"
22     folder_upload_permission: "0770"
23     script_permission: "0750"
24
25 users:

```

(suite sur la page suivante)

(suite de la page précédente)

```

22   vitam: "vitam"
23   vitamdb: "vitamdb"
24   group: "vitam"
25   services:
26     # Default log level for vitam components: logback values (TRACE, DEBUG, INFO, U
27     ↵WARN, ERROR, OFF)
28     log_level: WARN
29     start_timeout: 300
30     stop_timeout: 3600
31     port_service_timeout: 86400
32     api_call_timeout: 120
33     api_long_call_timeout: 300
34     status_retries_number: 60
35     status_retries_delay: 5
36     at_boot: false
37     ### Trust X-SSL-CLIENT-CERT header for external api auth ? true / falseU
38     ↵(default)
39     # Should only be enabled when accessing to vitam externals through a ReverseU
40     ↵Proxy that does "SSL offloading"
41     # NGINX configuration      : proxy_set_header X-SSL-CLIENT-CERT $ssl_client_
42     ↵escaped_cert;
43     # Apache httpd configuration : RequestHeader set X-SSL-CLIENT-CERT "%{SSL_
44     ↵CLIENT_CERT}s"
45     # Important : When enabled, special care must be taken to ensure firewall rulesU
46     ↵are properly set to ensure only
47     #           reverse proxy can access vitam external applications throughU
48     ↵their respective port_service to avoid
49     #           malicious header injection.
50     trust_client_certificate_header: false
51     ### Force chunk mode : set true if chunk header should be checked
52     vitam_force_chunk_mode: false
53     # syslog_facility
54     syslog_facility: local0
55
56 ##### Default Components parameters
57     ### Uncomment them if you want to update the default value applied on allU
58     ↵components
59
60     ### Ontology cache settings (max entries in cache & retention timeout inU
61     ↵seconds)
62     # ontologyCacheMaxEntries: 100
63     # ontologyCacheTimeoutInSeconds: 300
64     ### Elasticsearch scroll timeout in milliseconds settings
65     # elasticSearchScrollTimeoutInMilliseconds: 300000
66
67     ### The following values can be overwritten for each components in vitam:U
68     ↵parameters.
69     jvm_log: false
70     performance_logger: false
71
72     # consul_business_check: 10 # value in seconds
73     # consul_admin_check: 10 # value in seconds
74
75
76     ### Logs configuration for reconstruction services (INFO or DEBUG for activeU
77     ↵logs).

```

(suite sur la page suivante)

(suite de la page précédente)

```

68  ### Logs will be present only on secondary site.
69  ### Available for the following components: logbook, metadata & functional-
70  ↵administration.
71  reconstruction:
72    log_level: INFO
73
74  # Used in ingest, unitary update, mass-update
75  classificationList: [ "Non protégé", "Secret Défense", "Confidentiel Défense" ]
76  # Used in ingest, unitary update, mass-update
77  classificationLevelOptional: true
78  # Packages install retries
79  packages_install_retries_number: 1
80  packages_install_retries_delay: 10
81
82  # Request time check settings. Do NOT update except if required by Vitam support
83  # Max acceptable time desynchronization between machines (in seconds).
84  acceptableRequestTime: 10
85  # Critical time desynchronization between machines (in seconds).
86  criticalRequestTime: 60
87  # Request time alert throttling Delay (in seconds)
88  requestTimeAlertThrottlingDelay: 60
89
90  # Reconstruction config
91  restoreBulkSize: 10000
92
93  vitam_timers:
94    # //\ IMPORTANT :
95    # Please ensure timer execution is spread so that not all timers run_
96    ↵concurrently (eg. *:05:00, *:35:00, *:50:00...),
97    # Special care for heavy-load timers that run on same machines or use same_
98    ↵resources (eg. vitam-traceability-*).
99
100   #
101   # Quartz cron nomenclature
102   # minutely → 0 * * * * ?
103   # hourly → 0 0 * * * ?
104   # daily → 0 0 0 * * ?
105   # monthly → 0 0 0 1 * ?
106   # weekly → 0 0 0 ? * MON *
107   # yearly → 0 0 0 1 1 ?
108   # quarterly → 0 0 0 1 1/3 ?
109   # semiannually → 0 0 0 1 1/6 ?
110
111  logbook: # all have to run on only one machine
112    # Sécurisation des journaux des opérations
113    frequency_traceability_operations: "* 05 0/1 * * ?" # every hour
114    # Sécurisation des journaux du cycle de vie des groupes d'objets
115    frequency_traceability_lfc_objectgroup: "* 15 0/1 * * ?" # every hour
116    # Sécurisation des journaux du cycle de vie des unités archivistiques
117    frequency_traceability_lfc_unit: "* 35 0/1 * * ?" # every hour
118    # Audit de traçabilité
119    frequency_traceability_audit: "0 55 00 * * ?"
120    # Reconstruction (uniquement sur site secondaire)
121    frequency_logbook_reconstruction: "0 0/5 * * * ?"
122
123  storage:
124    # Sécurisation du journal des écritures
125    frequency_traceability_log: "0 40 0/4 * * ?" # every 4 hours
126    # Sauvegarde des journaux d'accès
127    vitam_storage_accesslog_backup: "0 10 0/4 * * ?" # every 4 hours

```

(suite sur la page suivante)

(suite de la page précédente)

```

122 # Sauvegarde des journaux des écritures
123 vitam_storage_log_backup: "0 15 0/4 * * ?" # every 4 hours
124
125 functional_administration:
126   frequency_create_acquisition_register_symbolic: "0 50 0 * * ?"
127   frequency_acquisition_register_reconstruction: "0 0/5 * * * ?"
128   frequency_rule_management_audit: "0 40 * * * ?"
129   frequency_reconstruction: "0 0/5 * * * ?"
130 metadata:
131   frequency_store_graph: "0 10/30 * * * ?"
132   frequency_reconstruction: "0 0/5 * * * ?"
133   frequency_computed_inherited_rules: "0 30 2 * * ?"
134   frequency_purge_dip: "0 0 * * * ?"
135   frequency_purge_transfers_sip: "0 25 2 * * ?"
136   frequency_audit_mongodb_es: "0 0 0 1 JAN ? 2020"
137   frequency_persistent_identifier_reconstruction: "0 0 0 1 1 ? 2020"
138
139 offer:
140   # Compaction offer logs
141   frequency_offerlog_compaction: "0 40 * * * ?"
142
143 scheduler:
144   job_parameters:
145     integrity_audit:
146       - key: SYSTEM
147       selected_tenants: [1]
148       operations_delay_in_minutes: 1440
149       frequency: "0 0 2 ? * * *" # Every day at 2am
150     existence_audit:
151       - key: SYSTEM
152       selected_tenants: [1]
153       operations_delay_in_minutes: 1440
154       frequency: "0 0 2 ? * * *" # Every day at 2am
155
156 ##### consul #####
157 # WARNING: consul_domain should be a supported domain name for your organization
158 #           You will have to generate server certificates with the same domain
159 #           ↳ name and the service subdomain name
160 #           Example: consul_domain=vitam means you will have to generate some
161 #           ↳ certificates with .service.vitam domain
162 #           access-external.service.vitam, ingest-external.service.vitam,
163 #           ↳ ...
164
165 consul_domain: consul
166 consul_folder_conf: "{{ vitam_defaults.folder.root_path }}/conf/consul"
167
168 # Workspace should be useless but storage have a dependency to it...
169 # elastic-kibana-interceptor is present as kibana is present, if kibana-data &
170 #   ↳ interceptor are not needed in the secondary site, just do not add them in the
171 #   ↳ hosts file
172 vitam_secondary_site_components: [ "scheduler", "logbook", "metadata",
173   ↳ "functional-administration", "storage", "storageofferdefault", "offer",
174   ↳ "elasticsearch-log", "elasticsearch-data", "logstash", "kibana", "mongoc",
175   ↳ "mongod", "mongos", "elastic-kibana-interceptor", "consul" ]
176
177 # containers list
178 containers_list: [ 'units', 'objects', 'objectgroups', 'logbooks', 'reports',
179   ↳ 'manifests', 'profiles', 'storagelog', 'storageaccesslog', 'storagegettraceability
180   ↳ ', 'rules', 'dip', 'agencies', 'backup', 'backupoperations', 'un{(suite sur la page suivante)
181   ↳ 'objectgroupgraph', 'distributionreports', 'acquisitionregistersdetail',
182   ↳ 'acquisitionregisterssymbolic', 'tmp', 'archivaltransferreply' ]

```

(suite de la page précédente)

```

170
171     ### Composants Vitam ###
172
vitam:
173     ### All available parameters for each components are described in the vitam_
174     ↵defaults variable
175
176     ### Example
177     # component:
178     #   at_boot: false
179     #   logback_rolling_policy: true
180     ## Force the log level for this component. Available logback values are (TRACE,_
181     ↵DEBUG, INFO, WARN, ERROR, OFF)
182     ## If this var is not set, the default one will be used (vitam_defaults.
183     ↵services.log_level)
184     #   log_level: "DEBUG"
185
186
accessexternal:
187     # Component name: do not modify
188     vitam_component: access-external
189     # DNS record for the service:
190     # Modify if ihm-demo is not using consul (typical production deployment)
191     host: "access-external.service.{{ consul_domain }}"
192     port_admin: 28102
193     port_service: 8444
194     baseuri: "access-external"
195     https_enabled: true
196     # Use platform secret for this component ? : do not modify
197     secret_platform: "false"
198     authorizeTrackTotalHits: false # if false, limit results to 10K. if true,_
199     ↵authorize results overs 10K (can overload elasticsearch-data)
200
accessinternal:
201     vitam_component: access-internal
202     host: "access-internal.service.{{ consul_domain }}"
203     port_service: 8101
204     port_admin: 28101
205     baseuri: "access-internal"
206     https_enabled: false
207     secret_platform: "true"
208
functional_administration:
209     vitam_component: functional-administration
210     host: "functional-administration.service.{{ consul_domain }}"
211     port_service: 8004
212     port_admin: 18004
213     baseuri: "adminmanagement"
214     https_enabled: false
215     secret_platform: "true"
216     cluster_name: "{{ elasticsearch.data.cluster_name }}"
217     # Number of AccessionRegisterSymbolic creation threads that can be run in_
218     ↵parallel.
219     accessionRegisterSymbolicThreadPoolSize: 16
220     # Number of rule audit threads that can be run in parallel.
221     ruleAuditThreadPoolSize: 16
222     # Reconstruction metrics cache in minutes (secondary site)
223     reconstructionMetricsCacheDurationInMinutes: 15
224
scheduler:
225     vitam_component: scheduler
226     host: "scheduler.service.{{ consul_domain }}"

```

(suite sur la page suivante)

(suite de la page précédente)

```

222 port_service: 8799
223 port_admin: 28799
224 baseuri: "scheduler"
225 https_enabled: false
226 secret_platform: "true"
227 schedulerThreadSize: 8
228 elastickibanainterceptor:
229   vitam_component: elastic-kibana-interceptor
230   host: "elastic-kibana-interceptor.service.{{ consul_domain }}"
231   port_service: 8014
232   port_admin: 18014
233   baseuri: ""
234   https_enabled: false
235   secret_platform: "false"
236   cluster_name: "{{ elasticsearch.data.cluster_name }}"
237 batchreport:
238   vitam_component: batch-report
239   host: "batch-report.service.{{ consul_domain }}"
240   port_service: 8015
241   port_admin: 18015
242   baseuri: "batchreport"
243   https_enabled: false
244   secret_platform: "false"
245 ingestexternal:
246   vitam_component: ingest-external
247   # DNS record for the service:
248   # Modify if ihm-demo is not using consul (typical production deployment)
249   host: "ingest-external.service.{{ consul_domain }}"
250   port_admin: 28001
251   port_service: 8443
252   baseuri: "ingest-external"
253   https_enabled: true
254   secret_platform: "false"
255   # this variable has containerization purposes only : must not be used in
256   # production environment
257   ignore_antivirus_check: false
258   # Directory where files should be placed for local ingest
259   upload_dir: "/vitam/data/ingest-external/upload"
260   # Directory where successful ingested files will be moved to
261   success_dir: "/vitam/data/ingest-external/upload/success"
262   # Directory where failed ingested files will be moved to
263   fail_dir: "/vitam/data/ingest-external/upload/failure"
264   # Action done to file after local ingest (see below for further information)
265   upload_final_action: "MOVE"
266   # upload_final_action can be set to three different values (lower or upper
267   # case does not matter)
268   # MOVE : After upload, the local file will be moved to either success_dir
269   # or fail_dir depending on the status of the ingest towards ingest-internal
270   # DELETE : After upload, the local file will be deleted if the upload
271   # succeeded
272   # NONE : After upload, nothing will be done to the local file (default
273   # option set if the value entered for upload_final_action does not exist)
274 ingestinternal:
275   vitam_component: ingest-internal
276   host: "ingest-internal.service.{{ consul_domain }}"
277   port_service: 8100
278   port_admin: 28100

```

(suite sur la page suivante)

(suite de la page précédente)

```

274  baseuri: "ingest"
275  https_enabled: false
276  secret_platform: "true"
277  ihm_demo:
278    vitam_component: ihm-demo
279    host: "ihm-demo.service.{{ consul_domain }}"
280    port_service: 8446
281    port_admin: 28002
282    baseurl: "/ihm-demo"
283    static_content: "{{ vitam_defaults.folder.root_path }}/app/ihm-demo/v2"
284    baseuri: "ihm-demo"
285    https_enabled: true
286    secret_platform: "false"
287    # User session timeout in milliseconds (for shiro)
288    session_timeout: 1800000
289    secure_cookie: true
290    # Specify here the realms you want to use for authentication in ihm-demo
291    # You can set multiple realms, one per line
292    # With multiple realms, the user will be able to choose between the allowed_
293    ↵realms
294      # Example: authentication_realms:
295      #           - x509Realm
296      #           - ldapRealm
297      # Authorized values:
298      # x509Realm: certificate
299      # iniRealm: ini file
300      # ldapRealm: ldap
301      authentication_realms:
302        # - x509Realm
303        - iniRealm
304        # - ldapRealm
305      allowedMediaTypes:
306        - type: "application"
307          subtype: "pdf"
308        - type: "text"
309          subtype: "plain"
310        - type: "image"
311          subtype: "jpeg"
312        - type: "image"
313          subtype: "tiff"
314  logbook:
315    vitam_component: logbook
316    host: "logbook.service.{{ consul_domain }}"
317    port_service: 9002
318    port_admin: 29002
319    baseuri: "logbook"
320    https_enabled: false
321    secret_platform: "true"
322    cluster_name: "{{ elasticsearch.data.cluster_name }}"
323    # Temporization delay (in seconds) for recent logbook operation events.
324    # Set it to a reasonable delay to cover max clock difference across servers +
325    ↵VM/GC pauses
326      operationTraceabilityTemporizationDelay: 300
327      # Max delay between 2 logbook operation traceability operations.
328      # A new logbook operation traceability is generated after this delay, even if_
329      ↵tenant has no
330      # new logbook operations to secure

```

(suite sur la page suivante)

(suite de la page précédente)

```

328  # Unit can be in DAYS, HOURS, MINUTES, SECONDS
329  # Hint: Set it to 690 MINUTES (11 hours and 30 minutes) to force new_
4  ↳traceability after +/- 12 hours (supposing
330  # logbook operation traceability timer run every hour +/- some clock delays)
331  operationTraceabilityMaxRenewalDelay: 690
332  operationTraceabilityMaxRenewalDelayUnit: MINUTES
333  # Number of logbook operations that can be run in parallel.
334  operationTraceabilityThreadPoolSize: 16
335  # Temporization delay (in seconds) for recent logbook lifecycle events.
336  # Set it to a reasonable delay to cover max clock difference across servers +
4  ↳VM/GC pauses
337  lifecycleTraceabilityTemporizationDelay: 300
338  # Max delay between 2 lifecycle traceability operations.
339  # A new unit/objectgroup lifecycle traceability is generated after this delay,
4  ↳ even if tenant has no
340  # new unit/objectgroups to secure
341  # Unit can be in DAYS, HOURS, MINUTES, SECONDS
342  # Hint: Set it to 690 MINUTES (11 hours and 30 minutes) to force new_
4  ↳traceability after +/- 12 hours (supposing
343  # LFC traceability timers run every hour +/- some clock delays)
344  lifecycleTraceabilityMaxRenewalDelay: 690
345  lifecycleTraceabilityMaxRenewalDelayUnit: MINUTES
346  # Max entries selected per (Unit or Object Group) LFC traceability operation
347  lifecycleTraceabilityMaxEntries: 100000
348  # Reconstruction metrics cache in minutes (secondary site)
349  reconstructionMetricsCacheDurationInMinutes: 15
350
metadata:
351  vitam_component: metadata
352  host: "metadata.service.{{ consul_domain }}"
353  port_service: 8200
354  port_admin: 28200
355  baseuri: "metadata"
356  https_enabled: false
357  secret_platform: "true"
358  cluster_name: "{{ elasticsearch.data.cluster_name }}"
359  # Archive Unit Profile cache settings (max entries in cache & retention
4  ↳timeout in seconds)
360  archiveUnitProfileCacheMaxEntries: 100
361  archiveUnitProfileCacheTimeoutInSeconds: 300
362  # Schema validator cache settings (max entries in cache & retention timeout
4  ↳in seconds)
363  schemaValidatorCacheMaxEntries: 100
364  schemaValidatorCacheTimeoutInSeconds: 300
365  # DIP cleanup delay (in minutes)
366  dipTimeToLiveInMinutes: 10080 # 7 days
367  criticalDipTimeToLiveInMinutes: 1440 # 1 day
368  transfersSIPTimeToLiveInMinutes: 10080 # 7 days
369  unitsStreamThreshold: 1000000 # 1 million
370  unitsStreamExecutionLimit: 3 # 3 times
371  persistentIdentifierReconstructionDelayInMinutes: 1440 # 1 day
372  persistentIdentifierReconstructionThreadPoolSize: 10 # parallel tenants_
4  ↳reconstruction
373  persistentIdentifierReconstructionBulkSize: 1000 # bulk size
374  objectsStreamThreshold: 1000000 # 1 million
375  objectsStreamExecutionLimit: 3 # 3 times
376  workspaceFreespaceThreshold: 25 # when below use critical time to live when_
4  ↳above use normal time to live

```

(suite sur la page suivante)

(suite de la page précédente)

```

377      elasticsearch_mapping_dir: "{{ vitam_defaults.folder.root_path }}/conf/
378      ↵metadata/mapping" # Directory of elasticsearch metadata mapping
379      ##### Audit data consistency MongoDB-ES ##### (Experimental / Not for_
380      ↵Production)
381      isDataConsistencyAuditRunnable: false
382      enableDataConsistencyRectificationMode: false
383      dataConsistencyAuditOplogMaxSize: 100
384      # Reconstruction metrics cache in minutes (secondary site)
385      reconstructionMetricsCacheDurationInMinutes: 15
386      context_path: "/metadata"
387
processing:
388      vitam_component: processing
389      host: "processing.service.{{ consul_domain }}"
390      port_service: 8203
391      port_admin: 28203
392      baseuri: "processing"
393      https_enabled: false
394      secret_platform: "true"
395      maxDistributionInMemoryBufferSize: 100000
396      maxDistributionOnDiskBufferSize: 100000000
397
security_internal:
398      vitam_component: security-internal
399      host: "security-internal.service.{{ consul_domain }}"
400      port_service: 8005
401      port_admin: 28005
402      baseuri: "security-internal"
403      https_enabled: false
404      secret_platform: "true"
405
storageengine:
406      vitam_component: storage
407      host: "storage.service.{{ consul_domain }}"
408      port_service: 9102
409      port_admin: 29102
410      baseuri: "storage"
411      https_enabled: false
412      secret_platform: "true"
413      storageTraceabilityOverlapDelay: 300
414      restoreBulkSize: 1000
415      # Storage write/access log backup max thread pool size
416      storageLogBackupThreadPoolSize: 16
417      # Storage write log traceability thread pool size
418      storageLogTraceabilityThreadPoolSize: 16
419      # Offer synchronization batch size & thread pool size
420      offerSynchronizationBulkSize: 1000
421      offerSyncThreadPoolSize: 32
422      # Retries attempts on failures
423      offerSyncNumberOfRetries: 3
424      # Retry wait delay on failures (in seconds)
425      offerSyncFirstAttemptWaitingTime: 15
426      offerSyncWaitingTime: 30
427      # Offer synchronization wait delay (in seconds) for async offers_
428      ↵(synchronization from a tape-storage offer)
429      offerSyncAccessRequestCheckWaitingTime: 10
430      logback_total_size_cap:
431          offersync:
432              history_days: 30
433              totalsize: "5GB"

```

(suite sur la page suivante)

(suite de la page précédente)

```

431 offerdiff:
432     history_days: 30
433     totalsize: "5GB"
434     # unit time per kB (in ms) used while calculating the timeout of an http_
435     ↵request between storage and offer.
436     timeoutMsPerKB: 100
437     # minimum timeout (in ms) for writing objects to offers
438     minWriteTimeoutMs: 60000
439     # minimum timeout per object (in ms) for bulk writing objects to offers
440     minBulkWriteTimeoutMsPerObject: 10000
441 storageofferdefault:
442     vitam_component: "offer"
443     port_service: 9900
444     port_admin: 29900
445     baseuri: "offer"
446     https_enabled: false
447     secret_platform: "true"
448     logback_total_size_cap:
449         offer_tape:
450             history_days: 30
451             totalsize: "5GB"
452         offer_tape_backup:
453             history_days: 30
454             totalsize: "5GB"
455 worker:
456     vitam_component: worker
457     host: "worker.service.{{ consul_domain }}"
458     port_service: 9104
459     port_admin: 29104
460     baseuri: "worker"
461     https_enabled: false
462     secret_platform: "true"
463     api_output_index_tenants: [ 0,1,2,3,4,5,6,7,8,9 ]
464     # Archive Unit Profile cache settings (max entries in cache & retention_
465     ↵timeout in seconds)
466     archiveUnitProfileCacheMaxEntries: 100
467     archiveUnitProfileCacheTimeoutInSeconds: 300
468     # Schema validator cache settings (max entries in cache & retention timeout_
469     ↵in seconds)
470     schemaValidatorCacheMaxEntries: 100
471     schemaValidatorCacheTimeoutInSeconds: 300
472     # Batch size for bulk atomic update
473     queriesThreshold: 100000
474     # Bulk atomic update batch size
475     bulkAtomicUpdateBatchSize: 100
476     # Max threads that can be run in concurrently is thread pool for bulk atomic_
477     ↵update
478     bulkAtomicUpdateThreadPoolSize: 8
479     # Number of jobs that can be queued in memory before blocking for bulk atomic_
480     ↵update
481     bulkAtomicUpdateThreadPoolQueueSize: 16
482     # Dip/transfer threshold file size
483     binarySizePlatformThreshold: 1
484     binarySizePlatformThresholdSizeUnit: "GIGABYTE"
485     # this variable has containerization purposes only : must not be used in_
486     ↵production environment

```

(suite sur la page suivante)

(suite de la page précédente)

```

482 ignore_antivirus_check: false
483 workspace:
484   vitam_component: workspace
485   host: "workspace.service.{{ consul_domain }}"
486   port_service: 8201
487   port_admin: 28201
488   baseuri: "workspace"
489   https_enabled: false
490   secret_platform: "true"
491   context_path: "/workspace"
492 collect_internal:
493   vitam_component: collect-internal
494   host: "collect-internal.service.{{ consul_domain }}"
495   port_service: 8038
496   port_admin: 28038
497   baseuri: "collect-internal"
498   https_enabled: false
499   secret_platform: "true"
500   transactionStatusThreadPoolSize: 4
501   statusTransactionThreadFrequency: 5
502   bulkAtomicUpdateThreadPoolSize: 8
503   bulkAtomicUpdateThreadPoolQueueSize: 16
504   # Max waiting time (in seconds) for transaction validation before timeout on
505   ↵SIP ingest.
506   maxWaitDelayForTransactionValidationInSeconds: 3600
507   bulkAtomicUpdateBatchSize: 100
508 collect_external:
509   vitam_component: collect-external
510   host: "collect-external.service.{{ consul_domain }}"
511   port_service: 8030
512   port_admin: 28030
513   baseuri: "collect-external"
514   https_enabled: true
515   secret_platform: "false"
516   authorizeTrackTotalHits: false # if false, limit results to 10K. if true, ↵
517   ↵authorize results overs 10K (can overload elasticsearch-data)
518   ingestionThreadPoolSize: 4
519   ingestionThreadFrequencySeconds: 5
520 metadata_collect:
521   vitam_component: metadata-collect
522   host: "metadata-collect.service.{{ consul_domain }}"
523   port_service: 8290
524   port_admin: 28290
525   baseuri: "metadata-collect"
526   https_enabled: false
527   secret_platform: "true"
528   cluster_name: "{{ elasticsearch.data.cluster_name }}"
529   # Archive Unit Profile cache settings (max entries in cache & retention
530   ↵timeout in seconds)
531   archiveUnitProfileCacheMaxEntries: 100
532   archiveUnitProfileCacheTimeoutInSeconds: 300
533   # Schema validator cache settings (max entries in cache & retention timeout
534   ↵in seconds)
535   schemaValidatorCacheMaxEntries: 100
536   schemaValidatorCacheTimeoutInSeconds: 300
537   # DIP cleanup delay (in minutes)
538   dipTimeToLiveInMinutes: 10080 # 7 days

```

(suite sur la page suivante)

(suite de la page précédente)

```

535     criticalDipTimeToLiveInMinutes: 1440 # 1 day
536     transfersSIPTimeToLiveInMinutes: 10080 # 7 days
537     workspaceFreespaceThreshold: 25 # when below use critical time to live when ↵
      ↵ above use normal time to live
538     elasticsearch_mapping_dir: "{{ vitam_defaults.folder.root_path }}/conf/
      ↵metadata-collect/mapping" # Directory of elasticsearch metadata-collect mapping
539     refreshElasticIndexPostIndexingMode: true
540     context_path: "/metadata-collect"
541     workspace_collect:
542       vitam_component: workspace-collect
543       host: "workspace-collect.service.{{ consul_domain }}"
544       port_service: 8291
545       port_admin: 28291
546       baseuri: "workspace-collect"
547       https_enabled: false
548       secret_platform: "true"
549       context_path: "/workspace-collect"
550     antivirus:
551       vitam_component: antivirus
552       host: "antivirus.service.{{ consul_domain }}"
553       port_service: 8082
554       port_admin: 28082
555       baseuri: "antivirus"
556       https_enabled: false
557       secret_platform: "true"
558       engine: "clamav" # or avast
559
560     # http://www.programmevitam.fr/ressources/DocCourante/html/installation/
      ↵installation/21-addons.html#durees-minimales-permettant-de-controler-les-
      ↵valeurs-saisies
561     vitam_tenant_rule_duration:
562       # - name: 2 # applied tenant
563       #   rules:
564       #     - AppraisalRule : "1 year" # rule name : rule value
565
566     # If you want to deploy vitam in a single VM, add the vm name in this array
567     single_vm_hostnames: [ 'localhost' ]

```

Note : Cas du composant ingest-external. Les directives upload_dir, success_dir, fail_dir et upload_final_action permettent de prendre en charge (ingest) des fichiers déposés dans upload_dir et appliquer une règle upload_final_action à l'issue du traitement (NONE, DELETE ou MOVE dans success_dir ou fail_dir selon le cas). Se référer au [DEX](#) pour de plus amples détails. Se référer au manuel de développement pour plus de détails sur l'appel à ce cas.

Avertissement : Selon les informations apportées par le métier, redéfinir les valeurs associées dans les directives classificationList et classificationLevelOptional. Cela permet de définir quels niveaux de protection du secret de la défense nationale, supporte l'instance. Attention : une instance de niveau supérieur doit toujours supporter les niveaux inférieurs.

- deployment/environments/group_vars/all/advanced/cots_vars.yml, comme suit :

```

1  ---
2
3 consul:
4   retry_interval: 10 # in seconds
5   check_interval: 10 # in seconds
6   check_timeout: 5 # in seconds
7   log_level: WARN # Available log_level are: TRACE, DEBUG, INFO, WARN or ERR
8   at_boot: true
9
10 # Please uncomment and fill values if you want to connect VITAM to external SIEM
11 # external_siem:
12 #   host:
13 #   port:
14
15 elasticsearch:
16   log:
17     host: "elasticsearch-log.service.{{ consul_domain }}"
18     port_http: "9201"
19     at_boot: true
20     groupe: "log"
21     baseuri: "elasticsearch-log"
22     cluster_name: "elasticsearch-log"
23     consul_check_http: 10 # in seconds
24     consul_check_tcp: 10 # in seconds
25     action_log_level: error
26     https_enabled: false
27     indices_fielddata_cache_size: '30%' # related to https://www.elastic.co/
28     ↵guide/en/elasticsearch/reference/8.13/modules-fielddata.html
29     indices_breaker_fielddata_limit: '40%' # related to https://www.elastic.
30     ↵co/guide/en/elasticsearch/reference/8.13/circuit-breaker.html#fielddata-circuit-
31     ↵breaker
32     dynamic_timeout: 30s
33     # log configuration
34     log_appenders:
35       root:
36         log_level: "info"
37       rolling:
38         max_log_file_size: "10MB"
39         max_total_log_size: "2GB"
40       deprecation_rolling:
41         max_log_file_size: "10MB"
42         max_files: "20"
43         log_level: "warn"
44       index_search_slowlog_rolling:
45         log_level: "warn"
46       index_indexing_slowlog_rolling:
47         log_level: "warn"
48
49     # By default, is commented. Should be uncommented if ansible computes_
50     ↵badly vCPUs number ; values are associated vCPUs numbers ; please adapt to_
51     ↵your configuration
52     # thread_pool:
53     #   index:
54     #     size: 2
55     #   get:
56     #     size: 2
57     #   search:
58     #     size: 2

```

(suite sur la page suivante)

(suite de la page précédente)

```

53      #      write:
54      #          size: 2
55      #      warmer:
56      #          max: 2
57 data:
58     host: "elasticsearch-data.service.{{ consul_domain }}"
59     # default is 0.1 (10%) and should be quite enough in most cases
60     #index_buffer_size_ratio: "0.15"
61     port_http: "9200"
62     groupe: "data"
63     baseuri: "elasticsearch-data"
64     cluster_name: "elasticsearch-data"
65     consul_check_http: 10 # in seconds
66     consul_check_tcp: 10 # in seconds
67     action_log_level: debug
68     https_enabled: false
69     indices_fielddata_cache_size: '30%' # related to https://www.elastic.co/
70     ↵guide/en/elasticsearch/reference/8.13/modules-fielddata.html
71     indices_breaker_fielddata_limit: '40%' # related to https://www.elastic.
72     ↵co/guide/en/elasticsearch/reference/8.13/circuit-breaker.html#fielddata-circuit-
73     ↵breaker
74     dynamic_timeout: 30s
75     # log configuration
76     log_appenders:
77         root:
78             log_level: "info"
79         rolling:
80             max_log_file_size: "10MB"
81             max_total_log_size: "5GB"
82         deprecation_rolling:
83             max_log_file_size: "10MB"
84             max_files: "20"
85             log_level: "warn"
86         index_search_slowlog_rolling:
87             log_level: "warn"
88         index_indexing_slowlog_rolling:
89             log_level: "warn"
90     # By default, is commented. Should be uncommented if ansible computes_
91     ↵badly vCPUs number ; values are associated vCPUs numbers ; please adapt to_
92     ↵your configuration
93         # thread_pool:
94         #      index:
95             #          size: 2
96         #      get:
97             #          size: 2
98         #      search:
99             #          size: 2
100            #      write:
101                #          size: 2
102                #      warmer:
103                    #          max: 2
104
105 mongodb:
106     mongos_port: 27017
107     mongoc_port: 27018
108     mongod_port: 27019
109     mongo_authentication: "true"

```

(suite sur la page suivante)

(suite de la page précédente)

```

105    check_consul: 10 # in seconds
106    drop_info_log: false # Drop mongo (I)nformational log, for Verbosity Level of _
107    ↵ 0
108    # logs configuration
109    logrotate: enabled # or disabled
110    ↵ '
111
logstash:
112    host: "logstash.service.{{ consul_domain }}"
113    port: 10514
114    rest_port: 20514
115    at_boot: true
116    check_consul: 10 # in seconds
117    ## logstash xms & xmx in Megabytes
118    # jvm_xms: 256 # default to memory_size/8
119    # jvm_xmx: 1024 # default to memory_size/4
120    # workers_number: 4 # default to cores*threads
121    log_appenders:
122        rolling:
123            max_log_file_size: "10MB"
124            max_total_log_size: "2GB"
125        json_rolling:
126            max_log_file_size: "10MB"
127            max_total_log_size: "2GB"
128
129
filebeat:
130    at_boot: true
131
132    # Prometheus params
133
prometheus:
134    metrics_path: /admin/v1/metrics
135    check_consul: 10 # in seconds
136    prometheus_config_file_target_directory: # Set path where "prometheus.yml" _
137    ↵ file will be generated. Example: /tmp/
138    server:
139        port: 9090
140        at_boot: true
141        tsdb_retention_time: "15d"
142        tsdb_retention_size: "5GB"
143
node_exporter:
144    enabled: true
145    port: 9101
146    at_boot: true
147    metrics_path: /metrics
148    log_level: "warn"
149    logrotate: enabled # or disabled
150    history_days: 30 # How many days to store logs if logrotate is set to
151    ↵ 'enabled'
152
consul_exporter:
153    enabled: true
154    port: 9107
155    at_boot: true
156    metrics_path: /metrics
157
elasticsearch_exporter:
158    enabled: true
159    port: 9114

```

(suite sur la page suivante)

(suite de la page précédente)

```

158     at_boot: true
159     metrics_path: /metrics
160     log_level: "warn"
161     logrotate: enabled # or disabled
162     history_days: 30 # How many days to store logs if logrotate is set to
163     ↵ 'enabled'
164     alertmanager:
165         api_port: 9093
166         cluster_port: 9094
167         at_boot: true
168         #receivers: # https://grafana.com/blog/2020/02/25/step-by-step-guide-to-
169         ↵ setting-up-prometheus-alertmanager-with-slack-pagerduty-and-gmail/
170         #- name: "slack_alert"
171         # slack_configs:
172         # - api_url: "https://hooks.slack.com/services/xxxxxxxxx/
173         ↵ XXXXXXXXXXXXXXXXXXXXXXXXX"
174         #     channel: '#your_alert_channel'
175         #     send_resolved: true
176     blackbox_exporter:
177         enabled: true
178         port: 9115
179         at_boot: true
180         logrotate: enabled # or disabled
181         history_days: 30 # How many days to store logs if logrotate is set to
182         ↵ 'enabled'
183         targets:
184             ## List all the targeted URLs that must be controled with blackbox
185             - "{{ vitam_reverse_external_protocol | default('https') }}://{{_
186             ↵ vitam_reverse_external_dns }}:{{ reverse_proxy_port | default(443) }},reverse"
187     mongodb_exporter:
188         enabled: true
189         port_mongoc: 9216
190         port_mongod: 9217
191         at_boot: true
192
193     grafana:
194         check_consul: 10 # in seconds
195         http_port: 3000
196         at_boot: true
197
198     # Curator units: days
199     curator:
200         at_boot: true
201         indices:
202             metricbeat:
203                 close: 5
204                 delete: 10
205             packetbeat:
206                 close: 5
207                 delete: 10
208
209     kibana:
210         header_value: "reporting"
211         import_delay: 10
212         import_retries: 10
213         # logs configuration
214         logrotate: enabled # or disabled

```

(suite sur la page suivante)

(suite de la page précédente)

```

210      history_days: 30 # How many days to store logs if logrotate is set to 'enabled'
211      ↵ '
212      log:
213          baseuri: "kibana_log"
214          api_call_timeout: 120
215          groupe: "log"
216          port: 5601
217          at_boot: true
218          default_index_pattern: "logstash-vitam*"
219          check_consul: 10 # in seconds
220          # default shards & replica
221          shards: 1
222          replica: 1
223          # pour index logstash-*
224          metrics:
225              shards: 1
226              replica: 1
227              # pour index metricbeat-*
228              metricbeat:
229                  shards: 3 # must be a factor of 30
230                  replica: 1
231          data:
232              baseuri: "kibana_data"
233              # OMA : bugdette : api_call_timeout is used for retries ; should ceate a ↵
234              #separate variable rather than this one
235              api_call_timeout: 120
236              groupe: "data"
237              port: 5601
238              default_index_pattern: "logbookoperation_*"
239              check_consul: 10 # in seconds
240              # index template for .kibana
241              shards: 1
242              replica: 1
243
244      syslog:
245          # value can be syslog-ng, rsyslog or filebeat (default)
246          name: "filebeat"
247
248          # filebeat:
249          # Default values are under ansible-vitam/roles/filebeat/defaults/main.yml
250
251      cerebro:
252          baseuri: "cerebro"
253          port: 9000
254          check_consul: 10 # in seconds
255          # logs configuration
256          logrotate: enabled # or disabled
257          history_days: 30 # How many days to store logs if logrotate is set to 'enabled'
258          ↵ '
259
260      siegfried:
261          port: 19000
262          consul_check: 10 # in seconds
263
264      clamav:
265          port: 3310
266          # logs configuration

```

(suite sur la page suivante)

(suite de la page précédente)

```

264    logrotate: enabled # or disabled
265    history_days: 30 # How many days to store logs if logrotate is set to 'enabled'
266    ↵
267    freshclam:
268      # frequency freshclam for database update per day (from 0 to 24 - 24_
269      ↵meaning hourly check)
270      db_update_periodicity: 1
271      private_mirror_address:
272      use_proxy: "no"
273
274
275    ## Avast Business Antivirus for Linux
276    ## if undefined, the following default values are applied.
277    # avast:
278      #   # logs configuration
279      #   logrotate: enabled # or disabled
280      #   history_days: 30 # How many days to store logs if logrotate is set to
281      ↵'enabled'
282      #   manage_repository: true
283      #   repository:
284      #     state: present
285      #     # For RedHat family
286      #     baseurl: https://repo.avcdn.net/linux-av/rpm/e19/release
287      #     gpgcheck: no
288      #     proxy: _none_
289      #     # For Debian family
290      #     baseurl: 'deb https://repo.avcdn.net/linux-av/deb debian-bookworm_
291      ↵release'
292      #     vps_repository: http://linux-av.u.avcdn.net/linux-av/avast/x86_64
293      #     ## List of sha256 hash of excluded files from antivirus. Useful for test_
294      ↵environments.
295      #     whitelist:
296      #       - XXXXXX
297      #       - YYYYYY
298
299    mongo_express:
300      baseuri: "mongo-express"
301
302    ldap_authentication:
303      ldap_protocol: "ldap"
304      ldap_server: "{% if groups['ldap']|length > 0 %}{{ groups['ldap']|first }}%_
305      ↵endif %}"
306      ldap_port: "389"
307      ldap_base: "dc=programmевитам,dc=fr"
308      ldap_login: "cn=Manager,dc=programmевитам,dc=fr"
309      uid_field: "uid"
310      ldap_userDn_Template: "uid={0},ou=people,dc=programmевитам,dc=fr"
311      ldap_group_request: "(&(objectClass=groupOfNames)(member={0}))"
312      ldap_admin_group: "cn=admin,ou=groups,dc=programmевитам, dc=fr"
313      ldap_user_group: "cn=user,ou=groups,dc=programmевитам, dc=fr"
314      ldap_guest_group: "cn=guest,ou=groups,dc=programmевитам, dc=fr"
315
316    # Backup tool on storage-offer
317    restic:
318      snapshot_retention: 30 # number of snapshots to keep
319      # default run backup at 23:00 everydays
320      cron:
321        minute: '00'

```

(suite sur la page suivante)

(suite de la page précédente)

```

315     hour: '23'
316     day: '*'
317     month: '*'
318     weekday: '*'
319     # [hosts_storage_offer_default] must be able to connect to the listed databases below to properly backup.
320     backup:
321         # mongo-offer
322         - name: "{{ offer_conf }}"
323             type: mongodb
324             host: "{{ offer_conf }}-mongos.service.{{ consul_domain }}:{{ mongodb.mongos_port }}"
325             user: "{{ mongodb[offer_conf].admin.user }}"
326             password: "{{ mongodb[offer_conf].admin.password }}"
327             # # mongo-data (only if mono-sharded cluster)
328             # - name: mongo-data
329             #   type: mongodb
330             #   host: "mongo-data-mongos.service.{{ consul_domain }}:{{ mongodb.mongos_port }}"
331             #   user: "{{ mongodb['mongo-data'].admin.user }}"
332             #   password: "{{ mongodb['mongo-data'].admin.password }}"
333             # # mongo-vitamui (only if vitamui is deployed)
334             # - name: mongo-vitamui
335             #   type: mongodb
336             #   host: mongo-vitamui-mongod.service.{{ consul_domain }}:{{ mongodb.mongod_port }}
337             #   # Add the following params on environments/group_vars/all/main/vault-vitam.yml
338             #   # They can be found under vitamui's deployment sources on environments/group_vars/all/vault-mongodb.yml
339             #   user: "{{ mongodb['mongo-vitamui'].admin.user }}"
340             #   password: "{{ mongodb['mongo-vitamui'].admin.password }}"

```

Note : Concernant Curator, en environnement de production, il est recommandé de procéder à la fermeture des index au bout d'une semaine pour les index de type « logstash » (3 jours pour les index « metrics »), qui sont le reflet des traces applicatives des composants de la solution logicielle **VITAM**. Il est alors recommandé de lancer le *delete* de ces index au bout de la durée minimale de rétention : 1 an (il n'y a pas de durée de rétention minimale légale sur les index « metrics », qui ont plus une vocation technique et, éventuellement, d'investigations).

- deployment/environments/group_vars/all/advanced/jvm_opts.yml, comme suit :

```

1  ---
2
3  ## JVM configuration for Vitam components
4
5  ### Global default configuration, act as default values if they are set
6  # vitam_defaults:
7  #   jvm_opts:
8  #     memory: "-Xms128m -Xmx512m"
9  #     gc: "-Xlog:gc*,gc+age=trace,safepoint:file={{ vitam_folder_log }}/gc.log:utctime,pid,tags:filecount=8,filesize=32m"
10 #     java: ""
11
12 ### Specific configuration for each components

```

(suite sur la page suivante)

(suite de la page précédente)

```

13  vitam:
14    accessinternal:
15      jvm_opts:
16        # memory: "-Xms128m -Xmx512m"
17        # gc: ""
18        # java: ""
19    accessexternal:
20      jvm_opts:
21        # memory: "-Xms128m -Xmx512m"
22        # gc: ""
23        # java: ""
24    elastickibanainterceptor:
25      jvm_opts:
26        # memory: "-Xms128m -Xmx512m"
27        # gc: ""
28        # java: ""
29    batchreport:
30      jvm_opts:
31        # memory: "-Xms128m -Xmx512m"
32        # gc: ""
33        # java: ""
34    ingestinternal:
35      jvm_opts:
36        # memory: "-Xms128m -Xmx512m"
37        # gc: ""
38        # java: ""
39    ingestexternal:
40      jvm_opts:
41        # memory: "-Xms128m -Xmx512m"
42        # gc: ""
43        # java: ""
44    metadata:
45      jvm_opts:
46        # memory: "-Xms128m -Xmx512m"
47        # gc: ""
48        # java: ""
49    ihm_demo:
50      jvm_opts:
51        # memory: "-Xms128m -Xmx512m"
52        # gc: ""
53        # java: ""
54    ihm_recette:
55      jvm_opts:
56        # memory: "-Xms128m -Xmx512m"
57        # gc: ""
58        # java: ""
59    logbook:
60      jvm_opts:
61        # memory: "-Xms128m -Xmx512m"
62        # gc: ""
63        # java: ""
64    workspace:
65      jvm_opts:
66        # memory: "-Xms128m -Xmx512m"
67        # gc: ""
68        # java: ""
69    processing:

```

(suite sur la page suivante)

(suite de la page précédente)

```

70      jvm_opts:
71          # memory: "-Xms128m -Xmx512m"
72          # gc: ""
73          # java: ""
74
75      worker:
76          jvm_opts:
77              # memory: "-Xms128m -Xmx512m"
78              # gc: ""
79              # java: ""
80
81      storageengine:
82          jvm_opts:
83              # memory: "-Xms128m -Xmx512m"
84              # gc: ""
85              # java: ""
86
87      storageofferdefault:
88          jvm_opts:
89              # memory: "-Xms128m -Xmx512m"
90              # gc: ""
91              # java: ""
92
93      functional_administration:
94          jvm_opts:
95              # memory: "-Xms128m -Xmx512m"
96              # gc: ""
97              # java: ""
98
99      scheduler:
100         jvm_opts:
101             # memory: "-Xms128m -Xmx512m"
102             # gc: ""
103             # java: ""
104
105     security_internal:
106         jvm_opts:
107             # memory: "-Xms128m -Xmx512m"
108             # gc: ""
109             # java: ""
110
111     library:
112         jvm_opts:
113             memory: "-Xms32m -Xmx128m"
114             # gc: ""
115             # java: ""
116
117     collect_internal:
118         jvm_opts:
119             # memory: "-Xms128m -Xmx512m"
120             # gc: ""
121             # java: ""
122
123     collect_external:
124         jvm_opts:
125             # memory: "-Xms128m -Xmx512m"
126             # gc: ""
127             # java: ""
128
129     metadata_collect:
130         jvm_opts:
131             # memory: "-Xms128m -Xmx512m"
132             # gc: ""
133             # java: ""
134
135     workspace_collect:
136         jvm_opts:
137             # memory: "-Xms128m -Xmx512m"

```

(suite sur la page suivante)

(suite de la page précédente)

```

127      # gc: ""
128      # java: ""
129
130      antivirus:
131          jvm_opts:
132              # memory: "-Xms128m -Xmx512m"
133              # gc: ""
134              # java: ""

```

Note : Cette configuration est appliquée à la solution logicielle [VITAM](#); il est possible de créer un tuning par « groupe » défini dans ansible.

4.2.5.15 Paramétrage de l'Offre Froide (librairies de cartouches)

Voir aussi :

Les principes de fonctionnement de l'offre froide sont décrits dans la documentation externe dédiée (« Archivage sur Offre Froide »).

La librairie et les lecteurs doivent déjà être configurés sur la machine devant supporter une instance de ce composant (avec login automatique en cas de reboot).

La commande `lsscsi -g` peut permettre de vérifier si des périphériques sont détectés.

Note : Une offre froide est mono-instantiable uniquement. Elle ne peut être déployée en haut-disponibilité.

Le paramétrage de l'offre froide se fait via la configuration du fichier `deployment/environments/group_vars/all/offer_opts.yml`. L'ensemble des clés disponibles est listé dans le fichier `deployment/environments/group_vars/all/offer_opts.yml.example`

L'offre froide doit être configurée avec le flag `AsyncRead` défini à `True` dans la stratégie par défaut de Vitam via `vitam_strategy` ou dans une stratégie additionnelle `other_strategies`.

Exemple :

```

vitam_strategy:
  - name: offer-tape-1
    referent: false
    asyncRead: true
  - name: offer-fs-2
    referent: true
    asyncRead: false

```

Une offre froide doit être définie dans la rubrique `vitam_offers` avec un provider de type *tape-library*

Exemple :

```

vitam_offers:
  offer-tape-1:
    provider: tape-library
    tapeLibraryConfiguration:
      ...

```

La section `tapeLibraryConfiguration` décrit le paramétrage général de l'offre froide.

- **maxTarEntrySize** Taille maximale (en octets) au-delà de la laquelle les fichiers entrants seront découpés en segments. Typiquement 1 Go, maximum 8 Go.
- **maxTarFileSize** Taille maximale (en octets) des *tars* à constituer. Typiquement 10 Go.
- **forceOverrideNonEmptyCartridges** Permet de passer outre le contrôle vérifiant que les bandes nouvellement introduites sont vides. Par défaut à *false*. Ne doit être défini à *true* que sur un environnement de recette où l'écrasement d'une bande de test est sans risque.
- **cachedTarMaxStorageSpaceInMB** Permet de définir la taille maximale du cache disque (en Mo) (Ex. 10 To pour un env de production)
- **cachedTarEvictionStorageSpaceThresholdInMB** Permet de définir la taille critique du cache disque (en Mo). Une fois ce seuil atteint, les archives non utilisées sont purgées (selon la date de dernier accès). Doit être plus petit que la taille maximale **cachedTarMaxStorageSpaceInMB**. (Ex. 8 To pour un env de production)
- **cachedTarSafeStorageSpaceThresholdInMB** Seuil « confortable » d'utilisation du cache (en Mo). Le processus d'éviction des archives du cache s'arrête lorsque ce seuil est atteint. Doit être plus petit que la taille critique **cachedTarEvictionStorageSpaceThresholdInMB**. (Ex. 6 To pour un env de production)
- **maxAccessRequestSize** Définit un seuil technique du nombre d'objets que peut cibler une demande d'accès. Par défaut de 10000. À ne pas modifier.
- **readyAccessRequestExpirationDelay** Valeur du délais d'expiration des demandes d'accès. Une fois une demande d'accès à des objets est prête, l'accès immédiat aux objets est garantie durant cette période.
- **readyAccessRequestExpirationUnit** Unité du délais d'expiration des demandes d'accès (une valeur parmi « SECONDS » / « MINUTES » / « HOURS » / « DAYS » / « MONTHS »).
- **readyAccessRequestPurgeDelay** Valeur du délais de purge complète des demandes d'accès.
- **readyAccessRequestPurgeUnit** Unité du délais de purge complète des demandes d'accès (une valeur parmi « SECONDS » / « MINUTES » / « HOURS » / « DAYS » / « MONTHS »).
- **accessRequestCleanupTaskIntervalDelay** Valeur de la fréquence de nettoyage des demandes d'accès.
- **accessRequestCleanupTaskIntervalUnit** Unité de la fréquence de nettoyage des demandes d'accès (une valeur parmi « SECONDS » / « MINUTES » / « HOURS » / « DAYS » / « MONTHS »).

Note : maxTarEntrySize doit être strictement inférieur à maxTarFileSize

Note : cachedTarEvictionStorageSpaceThresholdInMB doit être strictement inférieur à cachedTarMaxStorageSpaceInMB

Note : cachedTarSafeStorageSpaceThresholdInMB doit être strictement inférieur à cachedTarEvictionStorageSpaceThresholdInMB

Note : Se référer à la documentation [DAT](#) pour les éléments de dimensionnement du cache.

Note : La durée de purge des demandes d'accès doit être strictement supérieure à leur durée d'expiration.

Note : Le monitoring de l'offre froide est fortement recommandé afin de s'assurer du bon fonctionnement de l'offre, et du dimensionnement du disque local. Un dashboard dédié à l'offre froide de Vitam est déployé avec les composants « extra » prometheus et grafana.

Exemple :

```



```

Le paragraphe `topology` décrit la topologie de l'offre doit être renseigné. L'objectif de cet élément est de pouvoir définir une segmentation de l'usage des bandes pour répondre à un besoin fonctionnel. Il convient ainsi de définir des `buckets`, qu'on peut voir comme un ensemble logique de bandes, et de les associer à un ou plusieurs tenants.

- `tenants` tableau de 1 à n identifiants de tenants au format [1,...,n]
- `tarBufferingTimeoutInMinutes` Valeur en minutes durant laquelle une archive TAR peut rester ouverte (durée maximale d'accumulation des objets dans un TAR) avant que le TAR soit finalisé / planifié pour écriture sur bande.

Exemple :

```

topology:
  buckets:
    test:
      tenants: [0]
      tarBufferingTimeoutInMinutes: 60
    admin:
      tenants: [1]
      tarBufferingTimeoutInMinutes: 60
    prod:
      tenants: [2,3,4,5,6,7,8,9]
      tarBufferingTimeoutInMinutes: 60

```

Note : Tous les tenants doivent être affectés à un et un seul bucket.

Prudence : L'affectation d'un tenant à un bucket est définitive. i.e. Il est impossible de modifier le bucket auquel un tenant a été déjà affecté car les données ont déjà été écrites sur bandes. Il est possible cependant, lors de l'ajout d'un tout nouveau tenant à Vitam, d'affecter ce nouveau tenant à un bucket existant.

La section `tapeLibraries` permet de définir le paramétrage des bibliothèques de bandes pilotées par l'offre froide.

Note : Une offre de stockage Vitam ne peut manipuler qu'une seule bibliothèque de bandes. Afin de piloter plusieurs bibliothèques de bandes, il convient d'utiliser des offres Vitam différentes.

Une bibliothèque de bandes est composée d'un robot (bras articulé), et d'un ensemble de lecteurs.

Note : Seul un robot doit être configuré pour piloter une librairie de bandes. La configuration de plusieurs robots pour une même librairie de bandes n'est actuellement PAS supportée.

La commande `ls -l /dev/tape/by-id/` permet de lister les chemins des périphériques (lecteurs et bras articulés) à configurer.

Exemple :

```
$ ls -l /dev/tape/by-id/
total 0
lrwxrwxrwx 1 root root  9 Mar  7 11:07 scsi-1HP_E-Series_B4B0AC0000 -> ../../sg1
lrwxrwxrwx 1 root root  9 Mar  7 11:07 scsi-SHP_DLT_VS80_B4B0A00001 -> ../../st0
lrwxrwxrwx 1 root root 10 Mar  7 11:07 scsi-SHP_DLT_VS80_B4B0A00001-nst -> ../../nst0
lrwxrwxrwx 1 root root  9 Mar  7 11:07 scsi-SHP_DLT_VS80_B4B0A00002 -> ../../st1
lrwxrwxrwx 1 root root 10 Mar  7 11:07 scsi-SHP_DLT_VS80_B4B0A00002-nst -> ../../nst1
lrwxrwxrwx 1 root root  9 Mar  7 11:07 scsi-SHP_DLT_VS80_B4B0A00003 -> ../../st2
lrwxrwxrwx 1 root root 10 Mar  7 11:07 scsi-SHP_DLT_VS80_B4B0A00003-nst -> ../../nst2
lrwxrwxrwx 1 root root  9 Mar  7 11:07 scsi-SHP_DLT_VS80_B4B0A00004 -> ../../st3
lrwxrwxrwx 1 root root 10 Mar  7 11:07 scsi-SHP_DLT_VS80_B4B0A00004-nst -> ../../nst3
```

Prudence : Ne pas utiliser les chemins `/dev/*` dont l'index peut changer en cas de redémarrage. Utiliser les chemins `/dev/tape/by-id/*` (qui utilisent le numéro de série du device cible).

Prudence : Seuls les devices de lecteurs de type `/dev/nstX` (par exemple : `/dev/tape/by-id/scsi-SHP_DLT_VS80_B4B0A00001-nst -> /dev/nst0`) peuvent être utilisés dans Vitam. Les devices de lecteurs de type `/dev/stX` (par exemple : `/dev/tape/by-id/scsi-SHP_DLT_VS80_B4B0A00001 -> /dev/st0`) ne doivent PAS être utilisés (car ils causent à rebobinage automatique de la bande après chaque opération).

- **robots** : Définition du bras robotique de la librairie.
 - **device** : Chemin du fichier de périphérique scsi générique associé au bras. (ex. `/dev/tape/by-id/scsi-1HP_E-Series_B4B0AC0000`)
 - **mtxPath** : Chemin vers la commande Linux de manipulation du bras.
 - **timeoutInMilliseconds** : timeout en millisecondes à appliquer aux ordres du bras.
- **drives** : Définition du/ou des lecteurs de cartouches de la librairie.
 - **index** : Numéro de lecteur, valeur débutant à 0.
 - **device** : Chemin du fichier de périphérique scsi SANS REMBOBINAGE associé au lecteur. (ex. `/dev/tape/by-id/scsi-SHP_DLT_VS80_B4B0A00001-nst`)
 - **mtPath** : Chemin vers la commande Linux de manipulation des lecteurs.
 - **ddPath** : Chemin vers la commande Linux de copie de bloc de données.
 - **timeoutInMilliseconds** : timeout en millisecondes à appliquer aux ordres du lecteur.

- **fullCartridgeDetectionThresholdInMB** Seuil de détection de bande pleine (en Mo) Permet pour détecter en cas d'erreur d'écriture sur bande, la cause probable de l'erreur :

- Si le volume des données écrites sur bande > seuil : La bande est considérée comme pleine
- Si le volume des données écrites sur bande < seuil : La bande est considérée comme corrompue

Typiquement, 90% de la capacité théorique de stockage des cartouches (hors compression).

Exemple :

```
tapeLibraries:
  TAPE_LIB_1:
    robots:
      -
        device: /dev/tape/by-id/scsi-1HP_EML_E-Series_B4B0AC0000
        mtxPath: "/usr/sbin mtx"
        timeoutInMilliseconds: 3600000
    drives:
      -
        index: 0
        device: /dev/tape/by-id/scsi-SHP_DLT_VS80_B4B0A00001-nst
        mtPath: "/bin/mt"
        ddPath: "/bin/dd"
        timeoutInMilliseconds: 3600000
      -
        index: 1
        device: /dev/tape/by-id/scsi-SHP_DLT_VS80_B4B0A00002-nst
        mtPath: "/bin/mt"
        ddPath: "/bin/dd"
        timeoutInMilliseconds: 3600000
      -
        index: 2
        device: /dev/tape/by-id/scsi-SHP_DLT_VS80_B4B0A00003-nst
        mtPath: "/bin/mt"
        ddPath: "/bin/dd"
        timeoutInMilliseconds: 3600000
      -
        index: 3
        device: /dev/tape/by-id/scsi-SHP_DLT_VS80_B4B0A00004-nst
        mtPath: "/bin/mt"
        ddPath: "/bin/dd"
        timeoutInMilliseconds: 3600000
  fullCartridgeDetectionThresholdInMB : 2_000_000
```

4.2.5.16 Sécurisation SELinux

Depuis la release R13, la solution logicielle **VITAM** prend désormais en charge l'activation de SELinux sur le périmètre du composant worker et des processus associés aux *griffins* (greffons de préservation).

SELinux (Security-Enhanced Linux) permet de définir des politiques de contrôle d'accès à différents éléments du système d'exploitation en répondant essentiellement à la question « May <subject> do <action> to <object> », par exemple « May a web server access files in user's home directories ».

Chaque processus est ainsi confiné à un (voire plusieurs) domaine(s), et les fichiers sont étiquetés en conséquence. Un processus ne peut ainsi accéder qu'aux fichiers étiquetés pour le domaine auquel il est confiné.

Note : La solution logicielle **VITAM** ne gère actuellement que le mode *targeted* (« only *targeted* processes are protected »)

Les enjeux de la sécurisation SELinux dans le cadre de la solution logicielle **VITAM** sont de garantir que les processus associés aux *griffins* (greffons de préservation) n'auront accès qu'au ressources système strictement requises pour leur fonctionnement et leurs échanges avec les composants *worker*.

Note : La solution logicielle **VITAM** ne gère actuellement SELinux que pour le système d'exploitation AlmaLinux

Avertissement : SELinux n'a pas vocation à remplacer quelque système de sécurité existant, mais vise plutôt à les compléter. Aussi, la mise en place de politiques de sécurité reste de mise et à la charge de l'exploitant. Par ailleurs, l'implémentation SELinux proposée avec la solution logicielle **VITAM** est minimale et limitée au greffon de préservation Siegfried. Cette implémentation pourra si nécessaire être complétée ou améliorée par le projet d'implémentation.

SELinux propose trois modes différents :

- *Enforcing* : dans ce mode, les accès sont restreints en fonction des règles SELinux en vigueur sur la machine ;
- *Permissive* : ce mode est généralement à considérer comme un mode de débogage. En mode permissif, les règles SELinux seront interrogées, les erreurs d'accès logguées, mais l'accès ne sera pas bloqué.
- *Disabled* : SELinux est désactivé. Rien ne sera restreint, rien ne sera loggué.

La mise en oeuvre de SELinux est prise en charge par le processus de déploiement et s'effectue de la sorte :

- Isoler dans l'inventaire de déploiement les composants worker sur des hosts dédiés (ne contenant aucun autre composant **VITAM**)
- Positionner pour ces hosts un fichier *hostvars* sous *environments/host_vars/* contenant la déclaration suivante

```
selinux_state: "enforcing"
```

- Procéder à l'installation de la solution logicielle **VITAM** grâce aux playbooks ansible fournis, et selon la procédure d'installation classique décrite dans le DIN

4.2.5.17 Installation de la stack Prometheus

Note : Si vous disposez d'un serveur Prometheus et alertmanager, vous pouvez installer uniquement les exporters souhaités.

Prometheus server et alertmanager sont des addons dans la solution **VITAM**.

Voici à quoi correspond une configuration qui permettra d'installer toute la stack prometheus.

```
prometheus:  
  metrics_path: /admin/v1/metrics  
  check_consul: 10 # in seconds  
  prometheus_config_file_target_directory: # Set path where "prometheus.yml" file  
  ↪will be generated. Example: /tmp/  
  server:  
    port: 9090
```

(suite sur la page suivante)

(suite de la page précédente)

```

tsdb_retention_time: "7d"
tsdb_retention_size: "5GB"
node_exporter:
  enabled: true
  port: 9101
  metrics_path: /metrics
consul_exporter:
  enabled: true
  port: 9107
  metrics_path: /metrics
elasticsearch_exporter:
  enabled: true
  port: 9114
  metrics_path: /metrics
alertmanager:
  api_port: 9093
  cluster_port: 9094

```

- L'adresse d'écoute de ces composants est celle de la patte d'administration.
- Vous pouvez surcharger la valeur de certaines de ces variables (Par exemple le port d'écoute, le path de l'API).
- Pour générer uniquement le fichier de configuration prometheus.yml à partir du fichier d'inventaire de l'environnement en question, il suffit de spécifier le répertoire destination dans la variable `prometheus_config_file_target_directory`

4.2.5.17.1 Playbooks ansible

Veuillez vous référer à la documentation d'exploitation pour plus d'information.

- Installer prometheus et alertmanager

```
ansible-playbook ansible-vitam-extra/prometheus.yml -i environments/hosts.
˓→<environnement> --ask-vault-pass
```

- Générer le fichier de conf `prometheus.yml` dans le dossier `prometheus_config_file_target_directory`

```
ansible-playbook ansible-vitam-extra/prometheus.yml -i environments/hosts.
˓→<environnement> --ask-vault-pass
```

`-tags gen_prometheus_config ..`

4.2.5.18 Installation de Grafana

Note : Si vous disposez déjà d'un Grafana, vous pouvez l'utiliser pour l'interconnecter au serveur Prometheus.

Grafana est un addon dans la solution **VITAM**.

Grafana sera déployé sur l'ensemble des machines renseignées dans le groupe `[hosts_grafana]` de votre fichier d'inventaire.

Pour se faire, il suffit d'exécuter le playbook associée :

```
ansible-playbook ansible-vitam-extra/grafana.yml -i environments/hosts.<environnement>
  ↳ --ask-vault-pass
```

4.2.5.18.1 Configuration

Les paramètres de configuration de ce composant se trouvent dans le fichier environments/group_vars/all/advanced/cots_vars.yml. Vous pouvez adapter la configuration en fonction de vos besoins.

4.2.5.18.2 Configuration spécifique derrière un proxy

Si Grafana est déployé derrière un proxy, vous devez apporter des modifications au fichier de configuration ansible-vitam-extra/roles/grafana/templates/grafana.ini.j2

Voici les variables modifiées par la solution *VITAM* pour permettre le fonctionnement de Grafana derrière un proxy apache.

```
[server]
root_url = http://{{ ip_admin }}:{{ grafana.http_port | default(3000) }}/grafana
serve_from_sub_path = true

[auth.basic]
enabled = false
```

Avertissement : Lors de la première connexion, vous devrez changer le mot de passe par défaut (login : admin ; password : admin1234), configurer le datasource et créer/importer les dashboards manuellement.

4.2.5.19 Installation de restic

restic est un addon (beta) de la solution *VITAM*.

restic sera déployé sur l'ensemble des machines du groupe [hosts_storage_offer_default] qui possèdent le paramètre restic_enabled=true. Attention à ne renseigner qu'une seule fois ce paramètre par offer_conf.

Pour se faire, il suffit d'exécuter le playbook associé :

```
ansible-playbook --vault-password-file vault_pass.txt ansible-vitam-extra/restic.yml -
  ↳ i environments/hosts.<environnement>
```

4.2.5.19.1 Configuration

Les paramètres de configuration de ce composant se trouvent dans les fichiers environments/group_vars/all/advanced/cots_vars.yml et environments/group_vars/all/main/vault-cots.yml. Vous pouvez adapter la configuration en fonction de vos besoins.

4.2.5.19.2 Limitations actuelles

restic est fourni en tant que fonctionnalité beta. À ce titre, il ne peut se substituer à des vérifications régulières de l'état des sauvegardes de vos bases.

restic ne fonctionne pas avec les providers *openstack-swift*, *openstack-swift-v2* et *tape-library*.

restic ne fonctionne pas avec un cluster mongo multi-shardé. Ainsi, mongo-data ne peut être sauvegardé via restic que dans de petites instances de Vitam.

4.2.6 Procédure de première installation

4.2.6.1 Déploiement

4.2.6.1.1 Cas particulier : utilisation de ClamAv en environnement Debian

Dans le cas de l'installation en environnement Debian, la base de données n'est pas intégrée avec l'installation de ClamAv. C'est la commande `freshclam` qui en assure la charge. Si vous n'êtes pas connecté à internet, la base de données doit être installée manuellement. Les liens suivants indiquent la procédure à suivre : [Installation ClamAv¹⁸](#) et Section Virus Database¹⁹

4.2.6.1.2 Fichier de mot de passe des vaults ansible

Par défaut, le mot de passe des *vault* sera demandé à chaque exécution d'ansible avec l'utilisation de l'option `--ask-vault-pass` de la commande `ansible-playbook`.

Pour simplifier l'exécution des commandes `ansible-playbook`, vous pouvez utiliser un fichier `!repertoire_deployment!“vault_pass.txt“` contenant le mot de passe des fichiers *vault*. Ainsi, vous pouvez utiliser l'option `--vault-password-file=vault_pass.txt` à la place de l'option `--ask-vault-pass` dans les différentes commandes de cette documentation.

Avertissement : Il est déconseillé de conserver le fichier `vault_pass.txt` sur la machine de déploiement ansible car ce fichier permet d'avoir accès à l'ensemble des secrets de [VITAM](#).

4.2.6.1.3 Mise en place des repositories VITAM (optionnel)

[VITAM](#) fournit un playbook permettant de définir sur les partitions cible la configuration d'appel aux repositories spécifiques à [VITAM](#) :

Editer le fichier `!repertoire_inventory!“group_vars/all/main/repositories.yml“` à partir du modèle suivant (décommenter également les lignes) :

```

1  ---
2
3  # Vitam installation mode.
4  # Allowed values are: legacy, container
5  # Caution: container installation is in beta mode. Do not use it in production environments.
6  install_mode: legacy
7
8  ## Must be set when install_mode == 'legacy'
9  # vitam_repositories:
10 #   - key: repol # Mandatory: Only on RedHat family (AlmaLinux or RockyLinux)
11 #     value: http://path_to_repol # Mandatory: Path to the repository

```

(suite sur la page suivante)

18. <https://www.clamav.net/documents/installing-clamav>

19. <https://www.clamav.net/downloads>

(suite de la page précédente)

```
12 #     gpgcheck: 1 # Optionnal: Default to 0 (equivalent as [trusted=yes] on Debian)
13 #     gpgkey: path_to_custom_key # Optionnal: Only if gpgcheck is enabled; Default to _  
→official Vitam GPG Key
14 #     subtree: "./" # Optionnal: Only on Debian; Default to ./
15 #     proxy: http://proxy_url # Optionnal: Only on RedHat family (AlmaLinux or_  
→RockyLinux); Default to _none_
16
17 ## Must be set when install_mode == 'container'
18 # container_repository:
19 #   registry_url:
20 #   username:
21 #   password:
22 # Add vitam_container_version for specific container version deployment (default:_  
→latest)
23 vitam_container_version: '9.0.0'
```

Ce fichier permet de définir une liste de repositories. Décommenter et adapter à votre cas.

Pour mettre en place ces repositories sur les machines cibles, lancer la commande :

```
ansible-playbook ansible-vitam-extra/bootstrap.yml -i environments/hosts.  
→<environnement> --ask-vault-pass
```

Note : En environnement CentOS, il est recommandé de créer des noms de *repository* commençant par *vitam-*.

4.2.6.1.4 Génération des *hostvars*

Une fois l'étape de *PKI* effectuée avec succès, il convient de procéder à la génération des *hostvars*, qui permettent de définir quelles interfaces réseau utiliser. Actuellement la solution logicielle *VITAM* est capable de gérer 2 interfaces réseau :

- Une d'administration
- Une de service

4.2.6.1.4.1 Cas 1 : Machines avec une seule interface réseau

Si les machines sur lesquelles *VITAM* sera déployé ne disposent que d'une interface réseau, ou si vous ne souhaitez en utiliser qu'une seule, il convient d'utiliser le playbook `repertoire_playbook ansible‘‘generate_hostvars_for_1_network_interface.yml‘‘`

Cette définition des host_vars se base sur la directive `ansible_default_ipv4.address`, qui se base sur l'adresse *IP* associée à la route réseau définie par défaut.

Avertissement : Les communications d'administration et de service transiteront donc toutes les deux via l'unique interface réseau disponible.

4.2.6.1.4.2 Cas 2 : Machines avec plusieurs interfaces réseau

Si les machines sur lesquelles *VITAM* sera déployé disposent de plusieurs interfaces et si celles-ci respectent cette règle :

- Interface nommée eth0 = ip_service
- Interface nommée eth1 = ip_admin

Alors il est possible d'utiliser le playbook ansible-vitam-exploitation/generate_hostvars_for_2_network_interfaces.yml

Note : Pour les autres cas de figure, il sera nécessaire de générer ces hostvars à la main ou de créer un script pour automatiser cela.

4.2.6.1.4.3 Vérification de la génération des hostvars

A l'issue, vérifier le contenu des fichiers générés sous `repertoire_inventory/host_vars/` et les adapter au besoin.

Prudence : Cas d'une installation multi-sites. Sur site secondaire, s'assurer que, pour les machines hébergeant les offres, la directive `ip_wan` a bien été déclarée (l'ajouter manuellement, le cas échéant), pour que site le site *primaire* sache les contacter via une IP particulière. Par défaut, c'est l'IP de service qui sera utilisée.

4.2.6.1.5 Tests d'infrastructure

Il est possible de lancer une série de tests d'infrastructure en amont du déploiement, ceci afin de se prémunir d'éventuelles erreurs durant l'installation.

Les tests sont basés sur des prérequis de la solution *VITAM* et sont génériques. De ce fait, des « faux-positifs » peuvent être remontés dû à une configuration spécifique de votre environnement. Il est à votre charge d'analyser le rapport à l'issue des tests et de juger de la pertinence des résultats.

Les tests sont les suivants :

- Version d'Ansible
- Accès aux recursors (serveurs DNS)
- Présence de Java
- Accès aux repositories
- Accès aux offres objet

Comme pour le déploiement, les tests s'effectuent depuis la machine *ansible*. La commande pour les effectuer est la suivante :

```
ansible-playbook ansible-vitam/checks_infra.yml -i environments/hosts.<environnement>_<---ask-vault-pass
```

4.2.6.1.6 Déploiement

Une fois les étapes précédentes correctement effectuées (en particulier, la section *Génération des magasins de certificats* (page 70)), le déploiement s'effectue depuis la machine *ansible* et va distribuer la solution *VITAM* selon l'inventaire correctement renseigné.

Une fois l'étape de la génération des hosts effectuée avec succès, le déploiement est à réaliser avec la commande suivante :

```
ansible-playbook ansible-vitam/vitam.yml -i environments/hosts.<environnement> --ask-vault-pass
```

Note : Une confirmation est demandée pour lancer ce script. Il est possible de rajouter le paramètre `-e confirmation=yes` pour bypasser cette demande de confirmation (cas d'un déploiement automatisé).

Note : Il est possible d'effectuer les tests d'infrastructure décrits dans la partie précédente en ajoutant le paramètre `-e checks_infra=yes`. Un rapport s'affichera à l'issue des tests et il sera donné la possibilité de poursuivre ou non le déploiement.

Note : Il est également possible de forcer la suppression de profils de sécurité et de leurs données associées (contextes applicatifs et certificats) en ajoutant le paramètre `-e delete_security_profiles=yes`. Cela peut éventuellement être utile dans le cas d'un nouveau lancement de l'installation suite à un échec.

Prudence : Dans le cas où l'installateur souhaite utiliser un *repository* de binaires qu'il gère par lui-même, il est fortement recommandé de rajouter `--skip-tags "enable_vitam_repo"` à la commande `ansible-playbook`; dans ce cas, le comportement de `yum` n'est pas impacté par la solution de déploiement.

4.2.7 Éléments *extras* de l'installation

Prudence : Les éléments décrits dans cette section sont des éléments « *extras* »; il ne sont pas officiellement supportés, et ne sont par conséquence pas inclus dans l'installation de base. Cependant, ils peuvent s'avérer utile, notamment pour les installations sur des environnements hors production.

Prudence : Dans le cas où l'installateur souhaite utiliser un *repository* de binaires qu'il gère par lui-même, il est fortement recommandé de rajouter `--skip-tags "enable_vitam_repo"` à la commande `ansible-playbook`; dans ce cas, le comportement de `yum` n'est pas impacté par la solution de déploiement.

4.2.7.1 Configuration des *extras*

Le fichier `répertoire_inventory/group_vars/all/advanced/extra_vars.yml` contient la configuration des *extras* :

```

1  ---
2
3 vitam:
4   ihm_recette:
5     vitam_component: ihm-recette
6     host: "ihm-recette.service.{{ consul_domain }}"
7     port_service: 8445
8     port_admin: 28204
9     baseurl: /ihm-recette
10    static_content: "{{ vitam_defaults.folder.root_path }}/app/ihm-recette"
11    baseuri: "ihm-recette"
```

(suite sur la page suivante)

(suite de la page précédente)

```

12      secure_mode:
13          - authc
14
15      https_enabled: true
16      secret_platform: "false"
17      cluster_name: "{{ elasticsearch.data.cluster_name }}"
18      session_timeout: 1800000
19      secure_cookie: true
20      use_proxy_to_clone_tests: "yes"
21      elasticsearch_mapping_dir: "{{ vitam_defaults.folder.root_path }}/conf/ihm-
22      ↪recette/mapping"
23
24      library:
25          vitam_component: library
26          host: "library.service.{{ consul_domain }}"
27          port_service: 8090
28          port_admin: 28090
29          baseuri: "doc"
30          https_enabled: false
31          secret_platform: "false"
32          consul_business_check: 30 # value in seconds
33          consul_admin_check: 30 # value in seconds
34
35      tenant_to_clean_before_tnr: ["0", "1"]
36
37      # Period units in seconds
38
39      metricbeat:
40          enabled: false
41          system:
42              period: 10
43          mongodb:
44              period: 10
45          elasticsearch:
46              period: 10
47
48      packetbeat:
49          enabled: false
50
51      browser:
52          enabled: false
53
54      docker_opts:
55          registry_httponly: yes
56          vitam_docker_tag: latest
57          ## Custom CIDR address for docker bridge networks
58          # docker_bip: 192.168.191.1/24
59          ## Custom CIDR address settings for docker internal networks
60          # docker_address_pools_cidr: 192.168.192.1/18
61          # docker_address_pools_size: 24
62
63      gatling_install: false
64      docker_install: false # whether or not install docker & docker images

```

Avertissement : À modifier selon le besoin avant de lancer le playbook ! Les composant ihm-recette et ihm-demo ont la variable `secure_cookie` paramétrée à `true` par défaut, ce qui impose de pouvoir se connecter dessus uniquement en `https` (même derrière un reverse proxy). Le paramétrage de cette variable se fait dans le fichier

```
environments/group_vars/all/advanced/vitam_vars.yml
```

Note : La section metricbeat permet de configurer la périodicité d'envoi des informations collectées. Selon l'espace disponible sur le *cluster* Elasticsearch de log et la taille de l'environnement **VITAM** (en particulier, le nombre de machines), il peut être nécessaire d'allonger cette périodicité (en secondes).

Le fichier `repertoire_inventory/group_vars/all/main/vault-extra.yml` contient les secrets supplémentaires des *extras*; ce fichier est encrypté par `ansible-vault` et doit être paramétré avant le lancement de l'orchestration du déploiement, si le composant ihm-recette est déployé avec récupération des **TNR**.

```
1 # Example for git lfs ; uncomment & use if needed
2 #vitam_gitlab_itest_login: "account"
3 #vitam_gitlab_itest_password: "change_it_4DU42JVf2x2xmPBS"
```

Note : Pour ce fichier, l'encrypter avec le même mot de passe que `vault-vitam.yml`.

4.2.7.2 Déploiement des *extras*

Plusieurs playbooks d'*extras* sont fournis pour usage « tel quel ».

4.2.7.2.1 ihm-recette

Ce *playbook* permet d'installer également le composant **VITAM** ihm-recette.

```
ansible-playbook ansible-vitam-extra/ihm-recette.yml -i environments/hosts.
↳<environnement> --ask-vault-pass
```

Prudence : Avant de jouer le playbook, ne pas oublier, selon le contexte d'usage, de positionner correctement la variable `secure_cookie` décrite plus haut.

4.2.7.2.2 Extras complet

Ce *playbook* permet d'installer :

- des éléments de monitoring système
- un serveur Apache pour naviguer sur le `/vitam` des différentes machines hébergeant **VITAM**
- mongo-express (en docker; une connexion internet est alors nécessaire)
- le composant **VITAM** library, hébergeant la documentation du projet
- le composant **VITAM** ihm-recette (utilise si configuré des dépôts de jeux de tests)
- un reverse proxy, afin de fournir une page d'accueil pour les environnements de test
- l'outillage de tests de performance

Avertissement : Pour se connecter aux *IHM*, il faut désormais configurer `reverse_proxy_port=443` dans l'inventaire.

```
ansible-playbook ansible-vitam-extra/extrा.ym� -i environments/hosts.<environnement> -  
˓→-ask-vault-pass
```

CHAPITRE 5

Procédures de mise à jour de la configuration

Cette section décrit globalement les processus de reconfiguration d'une solution logicielle *VITAM* déjà en place et ne peut se substituer aux recommandations effectuées dans la « release-notes » associée à la fourniture des composants mis à niveau.

Se référer également aux *DEX* pour plus de procédures.

5.1 Cas d'une modification du nombre de tenants

Modifier dans le fichier d'inventaire la directive `vitam_tenant_ids`, et dans toutes les directives concernées (ex. `api_output_index_tenants`, `rules_index_tenants`, `vitam_removed_tenants`, `dedicated_tenants`, `grouped_tenants`...)

Exemple :

```
vitam_tenant_ids=[0,1,2]
```

A l'issue, il faut lancer le playbook de déploiement de *VITAM* (et, si déployé, les extras) avec l'option supplémentaire `--tags update_vitam_configuration`.

Exemple :

```
ansible-playbook ansible-vitam/vitam.yml -i environments/hosts.<environnement> --ask-vault-pass --tags update_vitam_configuration  
ansible-playbook ansible-vitam-extra/extras.yml -i environments/hosts.<environnement> --ask-vault-pass --tags update_vitam_configuration
```

Note : Si une offre froide est configurée, la liste des buckets configurés doit être mise à jour en conséquence.

5.2 Cas d'une modification des paramètres JVM

Se référer à *Tuning JVM* (page 70)

Pour les partitions sur lesquelles une modification des paramètres *JVM* est nécessaire, il faut modifier les « hostvars » associées.

A l'issue, il faut lancer le playbook de déploiement de *VITAM* (et, si déployé, les *extras*) avec l'option supplémentaire `--tags update_jvmoptions_vitam`.

Exemple :

```
ansible-playbook ansible-vitam/vitam.yml -i environments/hosts.<environnement> --ask-vault-pass --tags update_jvmoptions_vitam
ansible-playbook ansible-vitam-extra/extras.yml -i environments/hosts.<environnement> --ask-vault-pass --tags update_jvmoptions_vitam
```

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

5.3 Cas de la mise à jour des *griffins*

Modifier la directive `vitam_griffins` contenue dans le fichier `environments/group_vars/all/main/main.yml`.

Note : Dans le cas d'une montée de version des composant *griffins*, ne pas oublier de mettre à jour l'URL du dépôt de binaire associé.

Relancer le script de déploiement en ajoutant en fin de ligne `--tags griffins` pour ne procéder qu'à l'installation/mise à jour des *griffins*.

CHAPITRE 6

Post installation

6.1 Validation du déploiement

La procédure de validation est commune aux différentes méthodes d'installation.

6.1.1 Sécurisation du fichier `vault_pass.txt`

Le fichier `vault_pass.txt` est très sensible; il contient le mot de passe du fichier `!reperatoire_inventory!“group_vars/all/vault.yml“` qui contient les divers mots de passe de la plate-forme. A l'issue de l'installation, il est primordial de le sécuriser (suppression du fichier ou application d'un `chmod 400`).

6.1.2 Validation manuelle

Chaque service **VITAM** (en dehors de bases de données) expose des URL de statut à l'adresse suivante : `<protocole web http ou https>://<host>:<port>/admin/v1/status` Cette URL doit retourner une réponse HTTP 204 sur une requête HTTP GET, si OK.

Un playbook d'appel de l'intégralité des autotests est également inclus (`deployment/ansible-vitam-exploitation/status_vitam.yml`). Il est à lancer de la même manière que pour l'installation de **VITAM** (en renommant le playbook à exécuter).

Il est également possible de vérifier la version installée de chaque composant par l'URL :

`<protocole web http ou https>://<host>:<port>/admin/v1/version`

6.1.3 Validation via Consul

Consul possède une **IHM** pour afficher l'état des services **VITAM** et supervise le « `/admin/v1/status` » de chaque composant **VITAM**, ainsi que des check TCP sur les bases de données.

Pour se connecter à Consul : `http://<Nom du 1er host dans le groupe ansible hosts_consul_server>:8500/ui`

Pour chaque service, la couleur à gauche du composant doit être verte (correspondant à un statut OK).

Si une autre couleur apparaît, cliquer sur le service « KO » et vérifier le test qui ne fonctionne pas.

6.1.4 Post-installation : administration fonctionnelle

A l'issue de l'installation, puis la validation, un **administrateur fonctionnel** doit s'assurer que :

- le référentiel PRONOM ([lien vers pronom²⁰](#)) est correctement importé depuis « Import du référentiel des formats » et correspond à celui employé dans Siegfried
- le fichier « rules » a été correctement importé via le menu « Import du référentiel des règles de gestion »
- à terme, le registre des fonds a été correctement importé

Les chargements sont effectués depuis l'[IHM](#) demo.

6.2 Sauvegarde des éléments d'installation

Après installation, il est fortement recommandé de sauvegarder les éléments de configuration de l'installation (i.e. le contenu du répertoire déploiement/environnements); ces éléments seront à réutiliser pour les mises à jour futures.

Astuce : Une bonne pratique consiste à gérer ces fichiers dans un gestionnaire de version (ex : git)

Prudence : Si vous avez modifié des fichiers internes aux rôles, ils devront également être sauvegardés.

6.3 Troubleshooting

Cette section a pour but de recenser les problèmes déjà rencontrés et y apporter une solution associée.

6.3.1 Erreur au chargement des *index template* kibana

Cette erreur ne se produit qu'en cas de *filesystem* plein sur les partitions hébergeant un cluster elasticsearch. Par sécurité, kibana passe alors ses *index* en READ ONLY.

Pour fixer cela, il est d'abord nécessaire de déterminer la cause du *filesystem* plein, puis libérer ou agrandir l'espace disque.

Ensuite, comme indiqué sur [ce fil de discussion²¹](#), vous devez désactiver le mode READ ONLY dans les *settings* de l'*index .kibana* du cluster elasticsearch.

Exemple :

```
PUT .kibana/_settings
{
  "index": {
    "blocks": {
```

(suite sur la page suivante)

20. <http://www.nationalarchives.gov.uk/aboutapps/pronom/droid-signature-files.htm>

21. <https://discuss.elastic.co/t/forbidden-12-index-read-only-allow-delete-api/110282/2>

(suite de la page précédente)

```

        "read_only_allow_delete": "false"
    }
}
}

```

Indication : Il est également possible de lancer cet appel via l'[IHM](#) du kibana associé, dans l'onglet Dev Tools.

A l'issue, vous pouvez relancer l'installation de la solution logicielle [VITAM](#).

6.3.2 Erreur au chargement des tableaux de bord Kibana

Dans le cas de machines petitement taillées, il peut arriver que, durant le déploiement, la tâche `Wait for the kibana port to be opened` prenne plus de temps que le `timeout` défini (`vitam_defaults.services.start_timeout`). Pour fixer cela, il suffit de relancer le déploiement.

6.4 Retour d'expérience / cas rencontrés

6.4.1 Crash rsyslog, code killed, signal : BUS

Il a été remarqué chez un partenaire du projet Vitam, que rsyslog se faisait *killer* peu après son démarrage par le signal SIGBUS. Il s'agit très probablement d'un bug rsyslog <= 8.24 <https://github.com/rsyslog/rsyslog/issues/1404>

Pour fixer ce problème, il est possible d'upgrader rsyslog sur une version plus à jour en suivant cette documentation :

- RedHat ²²
- Debian ²³

6.4.2 Mongo-express ne se connecte pas à la base de données associée

Si mongoDB a été redémarré, il faut également redémarrer mongo-express.

6.4.3 Elasticsearch possède des shard non alloués (état « UNASSIGNED »)

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

22. <https://www.rsyslog.com/rhelcentos-rpms/>
 23. <https://www.rsyslog.com/debian-repository/>

```
{
  "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 officielle²⁴.

6.4.4 Elasticsearch possède des shards non initialisés (état « INITIALIZING »)

Tout d'abord, il peut être difficile d'identifier les shards en questions dans cerebro ; une requête HTTP GET sur l'API _cat/shards permet d'avoir une liste plus compréhensible. Un shard non initialisé correspond à un shard en cours de démarrage (Cf. une ancienne page de documentation²⁵. 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.

6.4.5 Elasticsearch est dans l'état « *read-only* »

Lorsque Elasticsearch répond par une erreur 403 et que le message suivant est observé dans les logs ClusterBlockException[blocked by: [FORBIDDEN/xx/index read-only / allow delete (api)]], cela est probablement consécutif à un remplissage à 100% de l'espace de stockage associé aux index Elasticsearch. Elasticsearch passe alors en lecture seule s'il ne peut plus indexer de documents et garantit ainsi la disponibilité des requêtes en lecture seule uniquement.

Afin de rétablir Elasticsearch dans un état de fonctionnement nominal, il vous faudra alors exécuter la requête suivante :

```
curl -XPUT -H "Content-Type: application/json" http://<es-host>:<es-port>/_all/_settings -d '{"index.blocks.read_only_allow_delete": null}'
```

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

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

6.4.6 MongoDB semble lent

Pour analyser la performance d'un cluster MongoDB, ce dernier fournit quelques outils permettant de faire une première analyse du comportement : [mongostat](#)²⁶ et [mongotop](#)²⁷.

Dans le cas de VITAM, le cluster MongoDB comporte plusieurs shards. Dans ce cas, l'usage de ces deux commandes peut se faire :

- soit sur le cluster au global (en pointant sur les noeuds mongos) : cela permet d'analyser le comportement global du cluster au niveau de ses points d'entrées ;

```
mongostat --host <ip_service> --port 27017 --username vitamdb-admin --
←password <password ; défaut : azerty> --authenticationDatabase admin
mongotop --host <ip_service> --port 27017 --username vitamdb-admin --
←password <password ; défaut : azerty> --authenticationDatabase admin
```

- soit directement sur les noeuds de stockage (mongod) : cela donne des résultats plus fins, et permet notamment de séparer l'analyse sur les noeuds primaires & secondaires d'un même replicaset.

```
mongotop --host <ip_service> --port 27019 --username vitamdb-localadmin --
←password <password ; défaut : qwerty> --authenticationDatabase admin
mongostat --host <ip_service> --port 27019 --username vitamdb-localadmin --
←password <password ; défaut : qwerty> --authenticationDatabase admin
```

D'autres outils sont disponibles directement dans le client mongo, notamment pour troubleshooter les problèmes dûs à la réplication²⁸ :

```
mongo --host <ip_service> --port 27019 --username vitamdb-localadmin --password
←<password ; défaut : qwerty> --authenticationDatabase admin
> rs.printSlaveReplicationInfo()
> rs.printReplicationInfo()
> db.runCommand( { serverStatus: 1 } )
```

D'autres commandes plus complètes existent et permettent d'avoir plus d'informations, mais leur analyse est plus complexe :

```
# returns a variety of storage statistics for a given collection
> use metadata
> db.stats()
> db.runCommand( { collStats: "Unit" } )
```

Enfin, un outil est disponible en standard afin de mesurer des performances des lecture/écritures avec des patterns proches de ceux utilisés par la base de données ([mongoperf](#)²⁹) :

```
echo "{nThreads:16,fileSizeMB:10000,r:true,w:true}" | mongoperf
```

6.4.7 Les shards de MongoDB semblent mal équilibrés

Normalement, un processus interne à MongoDB (le balancer) s'occupe de déplacer les données entre les shards (par chunk) pour équilibrer la taille de ces derniers. Les commandes suivantes (à exécuter dans un shell mongo sur une instance mongos - attention, ces commandes ne fonctionnent pas directement sur les instances mongod) permettent de s'assurer du bon fonctionnement de ce processus :

26. <https://docs.mongodb.com/manual/reference/program/mongostat/>

27. <https://docs.mongodb.com/manual/reference/program/mongotop/>

28. <https://docs.mongodb.com/manual/tutorial/troubleshoot-replica-sets/>

29. <https://docs.mongodb.com/manual/reference/program/mongoperf/>

- `sh.status()` : donne le status du sharding pour le cluster complet; c'est un bon premier point d'entrée pour connaître l'état du balancer.
- `use <dbname>, puis db.<collection>.getShardDistribution()`, en indiquant le bon nom de base de données (ex : `metadata`) et de collection (ex : `Unit`) : donne les informations de répartition des chunks dans les différents shards pour cette collection.

6.4.8 L'importation initiale (profil de sécurité, certificats) retourne une erreur

Les playbooks d'initialisation importent des éléments d'administration du système (profils de sécurité, certificats) à travers des APIs de la solution VITAM. Cette importation peut être en échec, par exemple à l'étape `TASK [init_contexts_and_security_profiles : Import admin security profile to fonctionnal-admin]`, avec une erreur de type 400. Ce type d'erreur peut avoir plusieurs causes, et survient notamment lors de redéploiements après une première tentative non réussie de déploiement; même si la cause de l'échec initial est résolue, le système peut se trouver dans un état instable. Dans ce cas, un déploiement complet sur environnement vide est nécessaire pour revenir à un état propre.

Une autre cause possible ici est une incohérence entre l'inventaire, qui décrit notamment les offres de stockage liées aux composants offer, et le paramétrage `vitam_strategy` porté par le fichier `offers_opts.yml`. Si une offre indiquée dans la stratégie n'existe nulle part dans l'inventaire, le déploiement sera en erreur. Dans ce cas, il faut remettre en cohérence ces paramètres et refaire un déploiement complet sur environnement vide.

6.4.9 Problème d'ingest et/ou d'accès

Si vous repérez un message de ce type dans les log **VITAM** :

```
fr.gouv.vitam.common.security.filter.RequestAuthorizationValidator.  
↳checkTimestamp(AuthorizationWrapper.java:102) : [vitam-env-int8-app-04.vitam-  
↳env:storage:239079175] Timestamp check failed. 16s  
fr.gouv.vitam.common.security.filter.RequestAuthorizationValidator.  
↳checkTimestamp(AuthorizationWrapper.java:107) : [vitam-env-int8-app-04.vitam-  
↳env:storage:239079175] Critical timestamp check failure. 61s
```

Il faut vérifier / corriger l'heure des machines hébergeant la solution logicielle **VITAM** .. caution :: Si un *delta* de temps important (10s par défaut) a été détecté entre les machines, des erreurs sont tracées dans les logs et une alerte est remontée dans le dashboard Kibana des Alertes de sécurité. Au delà d'un seuil critique (60s par défaut) d'écart de temps entre les machines, les requêtes sont systématiquement rejetées, ce qui peut causer des dysfonctionnements majeurs de la solution.

CHAPITRE 7

Montée de version

Pour toute montée de version applicative de la solution logicielle *VITAM*, se référer au *DMV*.

Annexes

8.1 Vue d'ensemble de la gestion des certificats

8.1.1 Liste des suites cryptographiques & protocoles supportés par VITAM

Il est possible de consulter les *ciphers* supportés par la solution logicielle [VITAM](#) dans deux fichiers disponibles sur ce chemin : *ansible-vitam/roles/vitam/templates/*

- **Le fichier `jetty-config.xml.j2`**

- La balise contenant l'attribut name= »IncludeCipherSuites » référence les ciphers supportés
- La balise contenant l'attribut name= »ExcludeCipherSuites » référence les ciphers non supportés

- **Le fichier `java.security.j2`**

- La ligne jdk.tls.disabledAlgorithms renseigne les *ciphers* désactivés au niveau java

Avertissement : Les 2 balises concernant les *ciphers* sur le fichier jetty-config.xml.j2 sont complémentaires car elles comportent des *wildcards* (*); en cas de conflit, l'exclusion est prioritaire.

Voir aussi :

Ces fichiers correspondent à la configuration recommandée ; celle-ci est décrite plus en détail dans le [DAT](#) (chapitre sécurité).

8.1.2 Vue d'ensemble de la gestion des certificats

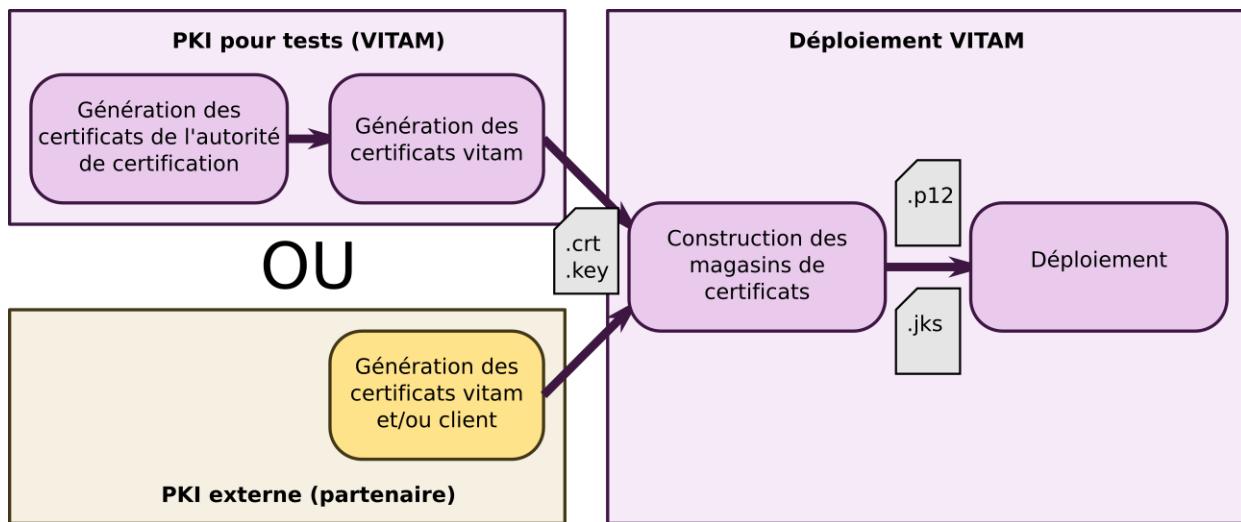
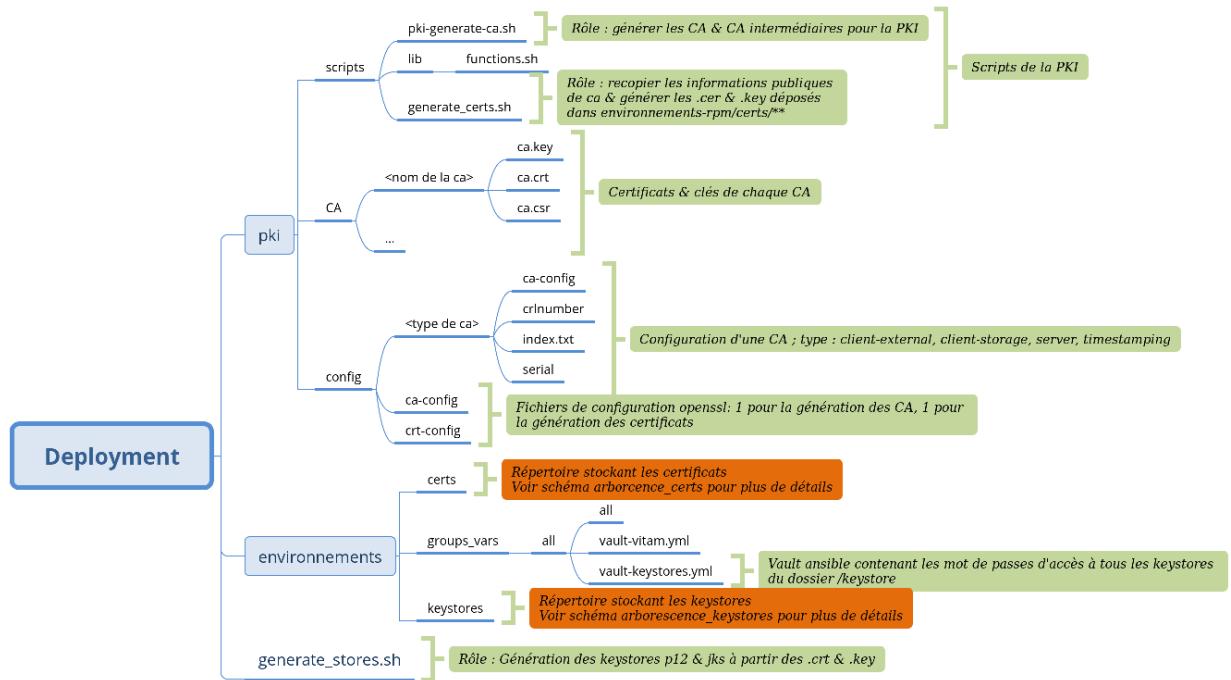


FIG. 1 – Vue d'ensemble de la gestion des certificats au déploiement

8.1.3 Description de l'arborescence de la PKI

Tous les fichiers de gestion de la **PKI** se trouvent dans le répertoire `deployment` de l'arborescence **VITAM** :

- Le sous répertoire `pki` contient les scripts de génération des **CA** & des certificats, les **CA** générées par les scripts, et les fichiers de configuration d'openssl
- Le sous répertoire `environments` contient tous les certificats nécessaires au bon déploiement de **VITAM** :
 - certificats publics des **CA**
 - certificats clients, serveurs, de timestamping, et coffre fort contenant les mots de passe des clés privées des certificats (sous-répertoire `certs`)
 - magasins de certificats (keystores / truststores / grantedstores), et coffre fort contenant les mots de passe des magasins de certificats (sous-répertoire `keystores`)
- Le script `generate_stores.sh` génère les magasins de certificats (keystores), cf la section *Fonctionnement des scripts de la PKI* (page 129)

FIG. 2 – Vue l’arborescence de la **PKI** Vitam

8.1.4 Description de l'arborescence du répertoire deployment/environments/certs

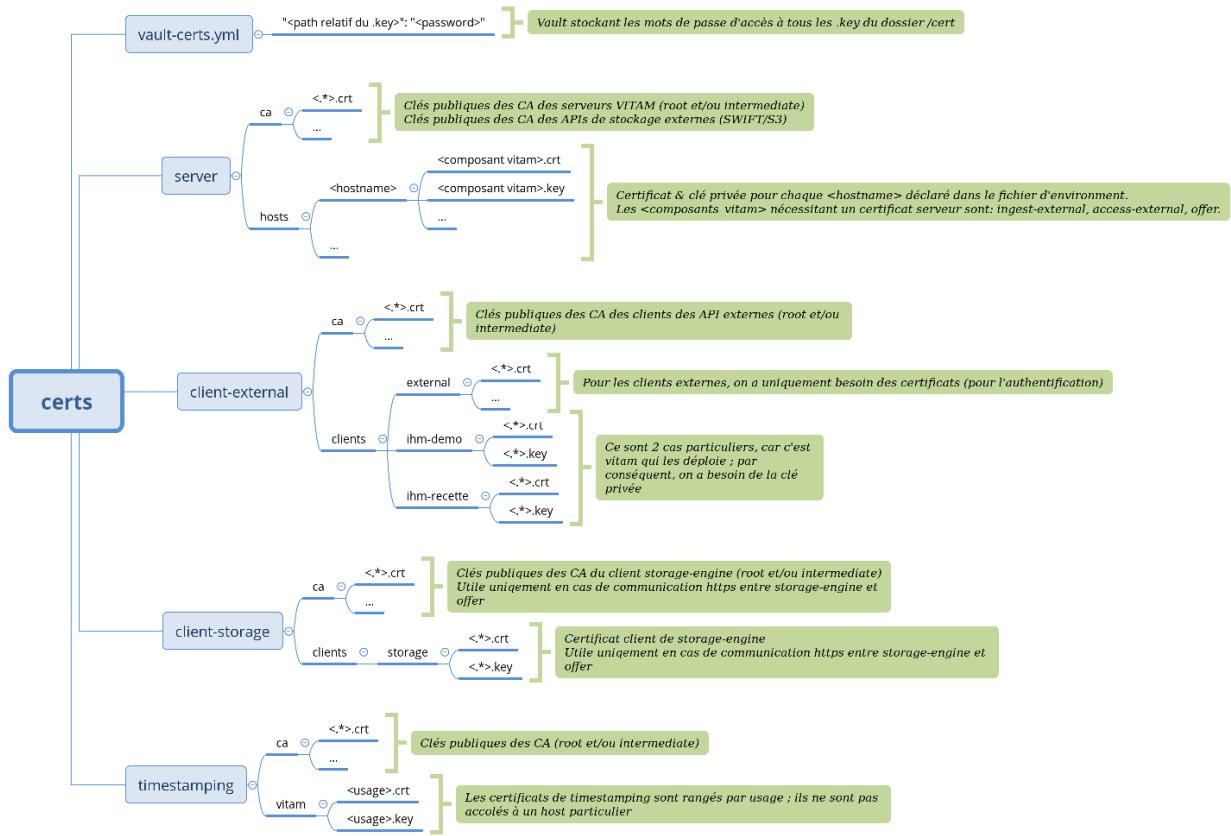


FIG. 3 – Vue détaillée de l'arborescence des certificats

8.1.5 Description de l'arborescence du répertoire deployment/environments keystores

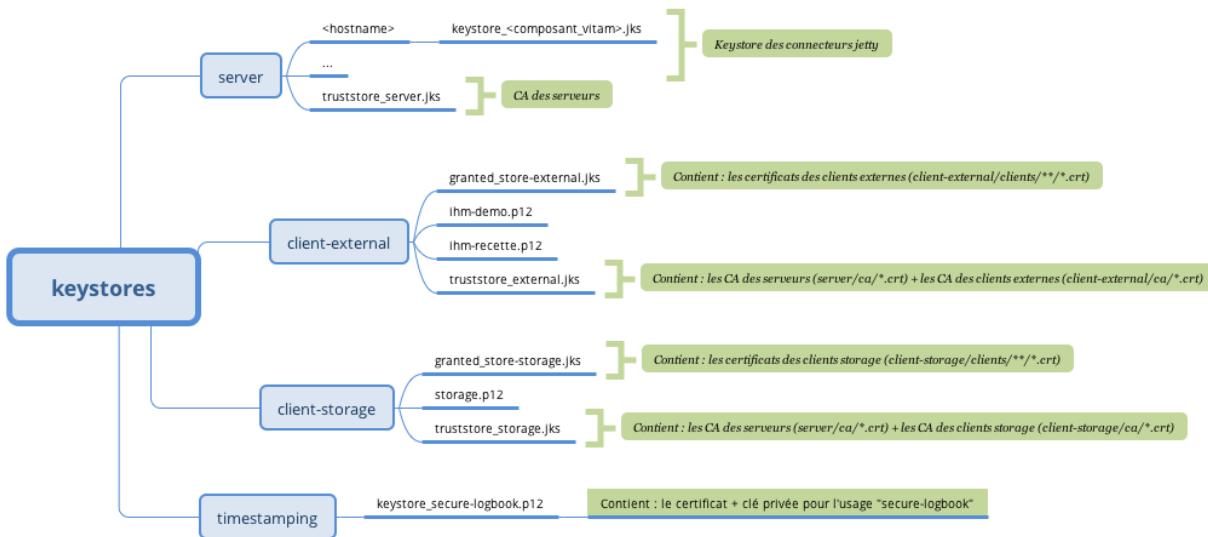


FIG. 4 – Vue détaillée de l'arborescence des keystores

8.1.6 Fonctionnement des scripts de la PKI

La gestion de la *PKI* se fait avec 3 scripts situés dans le répertoire `deployment` de l'arborescence *VITAM* :

- `pki/scripts/generate_ca.sh` : génère des autorités de certifications (si besoin)
- `pki/scripts/generate_certs.sh` : génère des certificats à partir des autorités de certifications présentes (si besoin)
 - Récupère le mot de passe des clés privées à générer dans le vault `environments/certs/vault-certs.yml`
 - Génère les certificats & les clés privées
- `generate_stores.sh` : génère les magasins de certificats nécessaires au bon fonctionnement de *VITAM*
 - Récupère le mot de passe du magasin indiqué dans `environments/group_vars/all/vault-keystore.yml`
 - Insère les bon certificats dans les magasins qui en ont besoin

Si les certificats sont créés par la *PKI* externe, il faut les positionner dans l'arborescence attendue avec le nom attendu pour certains (cf *l'image ci-dessus* (page 128)).

8.2 Spécificités des certificats

Trois différents types de certificats sont nécessaires et utilisés dans *VITAM* :

- Certificats serveur
- Certificats client
- Certificats d'horodatage

Pour générer des certificats, il est possible de s'inspirer du fichier `pki/config/crt-config`. Il s'agit du fichier de configuration openssl utilisé par la *PKI* de test de *VITAM*. Ce fichier dispose des 3 modes de configurations nécessaires pour générer les certificats de *VITAM* :

- `extension_server` : pour générer les certificats serveur
- `extension_client` : pour générer les certificats client
- `extension_timestamping` : pour générer les certificats d'horodatage

8.2.1 Cas des certificats serveur

8.2.1.1 Généralités

Les services *VITAM* qui peuvent utiliser des certificats serveur sont : `ingest-external`, `access-external`, `offer` (les seuls pouvant écouter en https). Par défaut, `offer` n'écoute pas en https par soucis de performances.

Pour les certificats serveur, il est nécessaire de bien réfléchir au *CN* et *subjectAltName* qui vont être spécifiés. Si par exemple le composant `offer` est paramétré pour fonctionner en https uniquement, il faudra que le *CN* ou un des *subjectAltName* de son certificat corresponde à son nom de service sur `consul`.

8.2.1.2 Noms DNS des serveurs https VITAM

Les noms *DNS* résolus par *Consul* seront ceux ci :

- `<nom_service>.service.<domaine_consul>` sur le datacenter local
- `<nom_service>.service.<dc_consul>.<domaine_consul>` sur n'importe quel datacenter

Rajouter le nom « `Consul` » avec le nom du datacenter dedans peut par exemple servir si une installation multi-site de *VITAM* est faite (appels storage -> `offer` inter *DC*)

Les variables pouvant impacter les noms d'hosts *DNS* sur *Consul* sont :

- `consul_domain` dans le fichier `environments/group_vars/all/advanced/vitam_vars.yml` -> `<domain_consul>`
- `vitam_site_name` dans le fichier d'inventaire `environments/hosts` (variable globale) -> `<dc_consul>`
- Service `offer` seulement : `offer_conf` dans le fichier d'inventaire `environments/hosts` (différente pour chaque instance du composant `offer`) -> `<nom_service>`

Exemples :

Avec `consul_domain: consul`, `vitam_site_name: dc2`, l'offre `offer-fs-1` sera résolue par

- `offer-fs-1.service.consul` depuis le *dc2*
- `offer-fs-1.service.dc2.consul` depuis n'importe quel *DC*

Avec `consul_domain: preprod.vitam`, `vitam_site_name: dc1`, les composants `ingest-external` et `access-external` seront résolu par

- `ingest-external.service.preprod.vitam` et `access-external.service.preprod.vitam` depuis le *DC* local
- `ingest-external.service.dc1.preprod.vitam` et `access-external.service.dc1.preprod.vitam` depuis n'importe quel *DC*

Avertissement : Si les composants `ingest-external` et `access-external` sont appelés via leur *IP* ou des records *DNS* autres que ceux de *Consul*, il faut également ne pas oublier de les rajouter dans les *subjectAltName*.

8.2.2 Cas des certificats client

Les services qui peuvent utiliser des certificats client sont :

- N'importe quelle application utilisant les !term :API *VITAM* exposées sur ingest-external et access-external
- Le service storage si le service offer est configuré en https
- **Un certificat client nommé vitam-admin-int est obligatoire**
 - Pour déployer *VITAM* (nécessaire pour initialisation du fichier pronom)
 - Pour lancer certains actes d'exploitation

8.2.3 Cas des certificats d'horodatage

Les services logbook et storage utilisent des certificats d'horodatage.

8.2.4 Cas des certificats des services de stockage objets

En cas d'utilisation d'offres de stockage objet avec *VITAM*, si une connexion https est utilisée, il est nécessaire de déposer les *CA* (root et/ou intermédiaire) des serveurs de ces offres de stockage dans le répertoire deployment/environments/certs/server/ca. Cela permettra d'ajouter ces *CA* dans le **truststore** du serveur offer lorsque les **keystores** seront générés.

8.3 Cycle de vie des certificats

Le tableau ci-dessous indique le mode de fonctionnement actuel pour les différents certificats et *CA*. Précisions :

- Les « procédures par défaut » liées au cycle de vie des certificats dans la présente version de la solution *VITAM* peuvent être résumées ainsi :
 - Création : génération par *PKI* partenaire + copie dans répertoires de déploiement + script generate_stores.sh + déploiement ansible
 - Suppression : suppression dans répertoires de déploiement + script generate_stores.sh + déploiement ansible
 - Renouvellement : régénération par *PKI* partenaire + suppression / remplacement dans répertoires de déploiement + script generate_stores.sh + redéploiement ansible
- Il n'y a pas de contrainte au niveau des *CA* utilisées (une *CA* unique pour tous les usages *VITAM* ou plusieurs *CA* séparées – cf. *DAT*). On appelle ici :
 - « *PKI* partenaire » : *PKI* / *CA* utilisées pour le déploiement et l'exploitation de la solution *VITAM* par le partenaire.
 - « *PKI* distante » : *PKI* / *CA* utilisées pour l'usage des frontaux en communication avec le back office *VITAM*.

Classe	Type	Usages	Origine	Création	Suppression	Renouvellement
Interne	CA	ingest & access	PKI partenaire	proc. par défaut	proc. par défaut	proc. par défaut
Interne	CA	offer	PKI partenaire	proc. par défaut	proc. par défaut	proc. par défaut
Interne	Certif	Horodatage	PKI partenaire	proc. par défaut	proc. par défaut	proc. par défaut
Interne	Certif	Storage (<i>Swift</i>)	Offre de stockage	proc. par défaut	proc. par défaut	proc. par défaut
Interne	Certif	Storage (s3)	Offre de stockage	proc. par défaut	proc. par défaut	proc. par défaut
Interne	Certif	ingest	PKI partenaire	proc. par défaut	proc. par défaut	proc. par défaut
Interne	Certif	access	PKI partenaire	proc. par défaut	proc. par défaut	proc. par défaut
Interne	Certif	offer	PKI partenaire	proc. par défaut	proc. par défaut	proc. par défaut
Interne	Certif	Timestamp	PKI partenaire	proc. par défaut	proc. par défaut	proc. par défaut
IHM demo	CA	ihm-demo	PKI partenaire	proc. par défaut	proc. par défaut	proc. par défaut
IHM demo	Certif	ihm-demo	PKI partenaire	proc. par défaut	proc. par défaut	proc. par défaut
SIA	CA	Appel API	PKI distante	proc. par défaut (PKI distante)	proc. par défaut	proc. par défaut (PKI distante)+recharger Certifs
SIA	Certif	Appel API	PKI distante	Génération + copie répertoire + deploy(par la suite appel API d'insertion)	Suppression Mongo	Suppression Mongo + API d'insertion
Personae	Certif	Appel API	PKI distante	API ajout	API suppression	API suppression + API ajout

Remarques :

- Lors d'un renouvellement de CA SIA, il faut s'assurer que les certificats qui y correspondaient soient retirés de MongoDB et que les nouveaux certificats soient ajoutés par le biais de l' API dédiée.
- Lors de toute suppression ou remplacement de certificats SIA, s'assurer que la suppression ou remplacement des contextes associés soit également réalisé.
- L'expiration des certificats n'est pas automatiquement prise en charge par la solution VITAM (pas de notification en fin de vie, pas de renouvellement automatique). Pour la plupart des usages, un certificat expiré est proprement rejeté et la connexion ne se fera pas ; les seules exceptions sont les certificats Personae, pour lesquels la validation de l'arborescence CA et des dates est à charge du front office en interface avec VITAM.

8.4 Ansible & SSH

En fonction de la méthode d'authentification sur les serveurs et d'élevation de privilège, il faut rajouter des options aux lignes de commande ansible. Ces options seront à rajouter pour toutes les commandes ansible du document .

Pour chacune des 3 sections suivantes, vous devez être dans l'un des cas décrits

8.4.1 Authentification du compte utilisateur utilisé pour la connexion SSH

Pour le login du compte utilisateur, voir la section *Informations plate-forme* (page 22).

8.4.1.1 Par clé SSH avec passphrase

Dans le cas d'une authentification par clé avec passphrase, il est nécessaire d'utiliser ssh-agent pour mémoriser la clé privée. Pour ce faire, il faut :

- exécuter la commande ssh-agent <shell utilisé> (exemple ssh-agent /bin/bash) pour lancer un shell avec un agent de mémorisation de la clé privée associé à ce shell
- exécuter la commande ssh-add et renseigner la passphrase de la clé privée

Vous pouvez maintenant lancer les commandes ansible comme décrites dans ce document.

A noter : ssh-agent est un démon qui va stocker les clés privées (déchiffrées) en mémoire et que le client **SSH** va interroger pour récupérer les informations privées pour initier la connexion. La liaison se fait par un socket UNIX présent dans /tmp (avec les droits 600 pour l'utilisateur qui a lancé le ssh-agent). Cet agent disparaît avec le shell qui l'a lancé.

8.4.1.2 Par login/mot de passe

Dans le cas d'une authentification par login/mot de passe, il est nécessaire de spécifier l'option –ask-pass (ou -k en raccourci) aux commandes ansible ou ansible-playbook de ce document .

Au lancement de la commande ansible (ou ansible-playbook), il sera demandé le mot de passe

8.4.1.3 Par clé SSH sans passphrase

Dans ce cas, il n'y a pas de paramétrage particulier à effectuer.

8.4.2 Authentification des hôtes

Pour éviter les attaques de type **MitM**, le client **SSH** cherche à authentifier le serveur sur lequel il se connecte. Ceci se base généralement sur le stockage des clés publiques des serveurs auxquels il faut faire confiance (~/.ssh/known_hosts).

Il existe différentes méthodes pour remplir ce fichier (vérification humaine à la première connexion, gestion centralisée, **DNSSEC**). La gestion de fichier est hors périmètre **VITAM** mais c'est un pré-requis pour le lancement d'ansible.

8.4.3 Elévation de privilèges

Une fois que l'on est connecté sur le serveur cible, il faut définir la méthode pour accéder aux droits **root**

8.4.3.1 Par sudo avec mot de passe

Dans ce cas, il faut rajouter les options `--ask-sudo-pass`

Au lancement de la commande `ansible` (ou `ansible-playbook`), il sera demandé le mot de passe demandé par `sudo`

8.4.3.2 Par su

Dans ce cas, il faut rajouter les options `--become-method=su --ask-su-pass`

Au lancement de la commande `ansible` (ou `ansible-playbook`), il sera demandé le mot de passe `root`

8.4.3.3 Par sudo sans mot de passe

Il n'y a pas d'option à rajouter (l'élévation par `sudo` est la configuration par défaut)

8.4.3.4 Déjà Root

Dans ce cas, il n'y a pas de paramétrages supplémentaires à effectuer.

Table des figures

1	Cinématique de déploiement	15
2	Vue détaillée des certificats entre le storage et l'offre en multi-site	21
3	Vue détaillée de l'arborescence des certificats	68
1	Vue d'ensemble de la gestion des certificats au déploiement	126
2	Vue l'arborescence de la <i>PKI</i> Vitam	127
3	Vue détaillée de l'arborescence des certificats	128
4	Vue détaillée de l'arborescence des keystores	129

Liste des tableaux

1	Documents de référence VITAM	2
1	Matrice de compétences	7
1	Description des identifiants de référentiels	76
2	Description des règles	77

Index

A

API, 3
AU, 3

B

BDD, 3
BDO, 3

C

CA, 3
CAS, 3
CCFN, 3
CN, 3
COTS, 3
CRL, 3
CRUD, 3

D

DAT, 3
DC, 3
DEX, 3
DIN, 3
DIP, 3
DMV, 3
DNS, 3
DNSSEC, 3
DSL, 3
DUA, 3

E

EAD, 3
EBIOS, 3
ELK, 3

F

FIP, 3

G

GOT, 3

I

IHM, 3
IP, 3
IsaDG, 3

J

JRE, 3
JVM, 4

L

LAN, 4
LFC, 4
LTS, 4

M

M2M, 4
MitM, 4
MoReq, 4

N

NoSQL, 4
NTP, 4

O

OAIS, 4
OOM, 4
OS, 4
OWASP, 4

P

PCA, 4
PDMA, 4
PKI, 4
PRA, 4

R

REST, 4
RGAA, 4
RGI, 4

RPM, **4**

S

SAE, **4**

SEDA, **4**

SGBD, **5**

SGBDR, **5**

SIA, **5**

SIEM, **5**

SIP, **5**

SSH, **5**

Swift, **5**

T

TLS, **5**

TNA, **5**

TNR, **5**

TTL, **5**

U

UDP, **5**

UID, **5**

V

VITAM, **5**

VM, **5**

W

WAF, **5**

WAN, **5**