



VITAM - Documentation de montées de version

Version 2.6.1

VITAM

juin 04, 2019

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	2
2.3	Glossaire	2
3	Généralités sur les versions	5
4	Montées de version	6
4.1	Etats attendus	6
4.1.1	Pré-migration	6
4.1.2	Post-migration	6
4.2	Principes généraux	6
4.2.1	Montées de version <i>bugfix</i>	6
4.2.2	Montées de version mineure	7
4.2.3	Montées de version majeure	7
4.3	Montées de version bugfix	7
4.3.1	Notes et procédures spécifiques R6	7
4.3.2	Notes et procédures spécifiques R7	7
4.3.2.1	Support de Debian 8	7
4.3.3	Notes et procédures spécifiques R8	8
4.3.4	Notes et procédures spécifiques R9	8
4.3.5	Notes et procédures spécifiques R10	8
4.4	Montées de version mineure	8
4.4.1	Notes et procédures spécifiques R7	8
4.4.1.1	Prérequis à la montée de version	8
4.4.1.1.1	Montée de version Consul	8
4.4.1.1.2	Reprise des données des contextes applicatifs	8
4.4.1.2	Montée de version	9
4.4.1.3	Etapas de migration	9
4.4.1.3.1	Avant de procéder à la migration	9
4.4.1.3.2	Procédure de migration des données	9
4.4.1.3.3	Après la migration	10

4.4.1.3.4	Vérification de la bonne migration des données	10
4.4.2	Notes et procédures spécifiques R8	10
4.4.3	Notes et procédures spécifiques R9	10
4.4.3.1	Prérequis à la montée de version	11
4.4.3.1.1	Gestion de la rétro-compatibilité des données des offres	11
4.4.3.1.2	Arrêt des <i>timers</i> systemd	11
4.4.3.1.3	Reprise des données de certificats	11
4.4.3.1.4	Montée de version MongoDB 3.4 vers 4.0	11
4.4.3.2	Montée de version	12
4.4.3.3	Etapas de migration	12
4.4.3.3.1	Procédure de migration des données	12
4.4.3.3.2	Procédure de réindexation des ObjectGroup	13
4.4.3.3.3	Après la migration	13
4.4.3.3.4	Une fois le site secondaire <i>up</i>	14
4.4.3.3.5	Vérification de la bonne migration des données	14
4.4.4	Notes et procédures spécifiques R10	14
4.4.4.1	Prérequis à la montée de version	14
4.4.4.2	Montée de version	14
4.4.4.3	Etapas de migration	14
4.4.4.3.1	Procédure de réindexation des ObjectGroup	14
4.4.4.3.2	Migration des données de certificats	15
4.4.4.3.3	Migration des contrats d'entrée	15

1.1 Objectif de ce document

Ce document décrit les procédures et informations utiles à une équipe d'exploitants de *VITAM* afin de réaliser les montées de version 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 et/ou exploiter la solution logicielle *VITAM*.

Note : Ce document ne décrit que les chemins de montées de versions vers les versions *VITAM* maintenues. Se référer au chapitre *Généralités sur les versions* (page 5) pour plus d'informations.

2.1 Information concernant les licences

La solution logicielle *VITAM* est publiée sous la licence [CeCILL 2.1](http://www.cecill.info/licences/Licence_CeCILL_V2.1-fr.html)¹ ; la documentation associée (comprenant le présent document) est publiée sous [Licence Ouverte V2.0](https://www.etalab.gouv.fr/wp-content/uploads/2017/04/ETALAB-Licence-Ouverte-v2.0.pdf)².

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

2.2.2 Référentiels externes

2.3 Glossaire

API Application Programming Interface

AU *Archive Unit*, unité archivistique

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

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

BDD Base De Données

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.

DAT Dossier d'Architecture Technique

DC Data Center

DEX Dossier d'EXploitation

DIN Dossier d'INstallation

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 DNSSEC](#)³

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

DUA Durée d'Utilité Administrative

EBIOS 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

ELK *Elasticsearch Logstash Kibana*

IHM Interface Homme Machine

IP *Internet Protocol*

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

JVM Java Virtual Machine ; Cf. *JRE*

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. [Explication](#)⁴

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

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

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

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.

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 PKI](#)⁶

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 REST](#)⁷

RGI Référentiel Général d’Interopérabilité

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

SAE Système d’Archivage Électronique

SEDA Standard d’Échange de Données pour l’Archivage

SGBD Système de Gestion de Base de Données

SIA Système d’Informations Archivistique

SIEM Security Information and Event Management

SIP Submission Information Package

SSH *Secure SHell*

Swift OpenStack Object Store project

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

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

<https://fr.wikipedia.org/wiki/NoSQL>
https://fr.wikipedia.org/wiki/Infrastructure_%C3%A0_cl%C3%A9s_publicues
https://fr.wikipedia.org/wiki/Representational_state_transfer

Généralités sur les versions

La numérotation des versions logicielles *VITAM* respecte le schéma suivant : X.Y.Z(-P).

- **X** : version majeure (V1, V2, V3)
- **Y** : version mineure (de type release, intitulées « R.Y . », contenant les nouvelles fonctionnalités)
- **Z** : version bugfix
- **P** : patch suite à bug critique (ne porte que sur les composants impactés)

Tableau 1: Tableau récapitulatif des versions de la solution logicielle *VITAM*

Code release	Version générale	Version associée	Version LTS	Version dépréciée
R6	1	1.0.x		X
R7	1	1.4.x	X	
R8	1	1.10.x		X
R9	2	2.1.x		
R10	2	2.6.x		

La release R7 est la version *LTS* de la version majeure V1 *VITAM*. Il est donc recommandé de ne plus utiliser les releases V1 R6 et R8, indiquées comme *deprecated* dans la suite du document.

Les releases R9 et R10 sont des *releases* de la version majeure V2 de la solution logicielle *VITAM*.

Pour plus d'informations, se reporter au chapitre « Maintenance » du document de [Présentation de la solution logicielle *VITAM*](#)⁸.

http://www.programmevitam.fr/ressources/DocCourante/autres/fonctionnel/VITAM_Presentation_solution_logicielle.pdf

4.1 Etats attendus

4.1.1 Pré-migration

Avant toute migration, il est attendu de la part des exploitants de vérifier :

- Que la solution logicielle *VITAM* fonctionne normalement
- Que l'ensemble des *timers* systemd sont stoppés
- Qu'aucun *workflow* n'est en statut **FATAL**

Se référer au chapitre « Suivi de l'état du système » du *DEX* pour plus d'informations.

4.1.2 Post-migration

A l'issue de toute migration, il est attendu de la part des exploitants de vérifier :

- Que la solution logicielle *VITAM* fonctionne normalement
- Que l'ensemble des *timers* systemd sont bien redémarrés (les redémarrer, le cas échéant)
- Qu'aucun *workflow* n'est en statut **FATAL**

Se référer au chapitre « Suivi de l'état du système » du *DEX* pour plus d'informations.

4.2 Principes généraux

4.2.1 Montées de version *bugfix*

Au sein d'une même *release*, la montée de version depuis une version *bugfix* vers une version *bugfix* supérieure est réalisée par réinstallation de la solution logicielle *VITAM* grâce aux playbooks ansible fournis, et selon la procédure d'installation classique décrite dans le *DIN*.

Les montées de version *bugfix* ne contiennent à priori pas d'opérations de migration ou de reprises de données particulières. Toutefois, des spécificités propres aux différentes versions *bugfixes* peuvent s'appliquer ; elles sont explicitées dans le chapitre *Montées de version bugfix* (page 7).

Prudence : Parmi les versions *bugfixes* publiées au sein d'une même *release*, seuls les chemins de montées de version d'une version *bugfix* à la version *bugfix* suivante sont qualifiés par *VITAM*.

4.2.2 Montées de version mineure

La montée de version depuis une version mineure (de type *release*) vers une version mineure supérieure est réalisée par réinstallation de la solution logicielle *VITAM* grâce aux playbooks ansible fournis, et selon la procédure d'installation classique décrite dans le *DIN*.

Ce document décrit les chemins de montées de version depuis une version mineure, vers la version mineure maintenue supérieure.

Les montées de version mineure doivent être réalisées en s'appuyant sur les dernières versions *bugfixes* publiées.

Les opérations de migration ou de reprises de données propres aux différentes versions *releases* sont explicitées dans le chapitre *Montées de version mineure* (page 8).

Prudence : Parmi les versions mineures publiées au sein d'une même version majeure, seuls les chemins de montées de version depuis une version mineure maintenue, vers la version mineure maintenue suivante sont qualifiés par *VITAM*.

4.2.3 Montées de version majeure

La montée de version depuis une version majeure vers une version majeure supérieure s'appuie sur les chemins de montées de version mineure décrits dans le chapitre *Montées de version mineure* (page 8).

4.3 Montées de version bugfix

4.3.1 Notes et procédures spécifiques R6

Note : Cette version n'est à ce jour plus officiellement supportée.

4.3.2 Notes et procédures spécifiques R7

4.3.2.1 Support de Debian 8

Prudence : Les *packages* Debian 8 *jessie-backports* ne sont plus accessibles depuis les sites miroirs *stable* de Debian. Ils sont désormais disponibles sur le [site des archives de Debian](#)⁹. Les dépôts *jessie-updates* (*security-updates* et *stable-updates*) devront également être désactivés. Pour plus d'informations, vous pouvez consulter [cet article](#)¹⁰.

4.3.3 Notes et procédures spécifiques R8

Note : Cette version n'est à ce jour plus officiellement supportée.

4.3.4 Notes et procédures spécifiques R9

Sans objet.

4.3.5 Notes et procédures spécifiques R10

Sans objet.

4.4 Montées de version mineure

4.4.1 Notes et procédures spécifiques R7

Prudence : Rappel : la montée de version vers la release R7 s'effectue depuis la release R6 (V1, *deprecated*) et doit être réalisée en s'appuyant sur les dernières versions bugfixes publiées.

4.4.1.1 Prérequis à la montée de version

4.4.1.1.1 Montée de version Consul

Le composant `vitam-consul` a été monté de version ; la procédure ci-dessous permet la mise en conformité des fichiers de configuration du service afin qu'ils soient compatibles avec la nouvelle version Consul.

La montée de version Consul s'effectue à l'aide du *playbook* d'installation (*VITAM* et/ou *extra*) au lancement duquel il est nécessaire d'ajouter la directive `--tags consul_conf`.

Exemples :

```
ansible-playbook ansible-vitam/vitam.yml -i environments/<fichier d'inventaire>
--vault-password-file vault_pass.txt --tags consul_conf
```

```
ansible-playbook ansible-vitam-extra/extra.yml -i environments/<fichier
d'inventaire> --ask-vault-pass --tags consul_conf
```

A l'issue de l'exécution du *playbook*, s'assurer que la montée de version Consul a été réalisée avec succès en vérifiant que l'ensemble des services Consul est à l'état **OK**.

4.4.1.1.2 Reprise des données des contextes applicatifs

En prérequis, il est également nécessaire d'effectuer une reprise des données des contextes applicatifs (base de données MongoDB `masterdata`, collection `Context`).

<http://archive.debian.org>
<https://www.lucas-nussbaum.net/blog/?p=947>

Deux champs liés aux contextes applicatifs ont été mis à jour en version R7 et doivent être migrés avant le déploiement de la nouvelle version de la solution logicielle *VITAM*.

Sous `deployment`, exécuter la commande suivante :

```
ansible-playbook ansible-vitam-exploitation/preinstall_r7.yml
--vault-password-file vault_pass.txt
```

ou, si `vault_pass.txt` n'a pas été renseigné :

```
ansible-playbook ansible-vitam-exploitation/preinstall_r7.yml --ask-vault-pass
```

A l'issue de l'exécution du *playbook*, si le *playbook* ne remonte pas d'erreur, la reprise des contextes applicatifs a été réalisée avec succès.

4.4.1.2 Montée de version

La montée de version vers la release R7 est réalisée par réinstallation de la solution logicielle *VITAM* grâce aux *playbooks* ansible fournis, et selon la procédure d'installation classique décrite dans le *DIN*.

4.4.1.3 Etapes de migration

Dans le cadre d'une montée de version R6 vers R7, il est nécessaire d'appliquer un *playbook* de migration de données à l'issue de la réinstallation de la solution logicielle *VITAM*. Ceci est dû à des changements de modèles de données suite à la mise en place de l'ontologie.

4.4.1.3.1 Avant de procéder à la migration

Les commandes suivantes sont à lancer depuis le répertoire `deployment` sur les différents sites hébergeant la solution logicielle *VITAM* :

```
ansible-playbook -i environments/<inventaire> ansible-vitam-exploitation/
stop_vitam_timers.yml --vault-password-file vault_pass.txt
```

ou, si `vault_pass.txt` n'a pas été renseigné :

```
ansible-playbook -i environments/<inventaire> ansible-vitam-exploitation/
stop_vitam_timers.yml --ask-vault-pass
```

A l'issue de l'exécution du *playbook*, les *timers* systemd ont été arrêtés, afin de ne pas perturber la migration.

Il est également recommandé de ne lancer la procédure de migration qu'après s'être assuré que plus aucun *workflow* n'est en cours de traitement.

4.4.1.3.2 Procédure de migration des données

La migration des données est réalisée en exécutant la commande suivante (sur le site primaire uniquement, dans le cas d'une installation multi-sites) :

```
ansible-playbook -i environments/<inventaire> ansible-vitam-exploitation/
migration_r6_r7.yml --vault-password-file vault_pass.txt
```

ou, si `vault_pass.txt` n'a pas été renseigné :

```
ansible-playbook -i environments/<inventaire> ansible-vitam-exploitation/
migration_r6_r7.yml --ask-vault-pass
```

Avertissement : Selon la volumétrie des données précédemment chargées, le *playbook* peut durer jusqu'à plusieurs heures.

Note : Durant la migration, il est fortement recommandé de ne pas procéder à des versements de données. En effet, le *playbook* se charge d'arrêter les composants `ingest-external` et `access-external`, avant de réaliser les opérations de migration de données, puis de redémarrer les composants `ingest-external` et `access-external`.

Les opérations de migration réalisées portent, entre autres, sur les éléments suivants :

- Graph / SEDA
- Mise à jour d'un champ des contextes applicatifs
- Réindexations Elasticsearch

4.4.1.3.3 Après la migration

Exécuter la commande suivante afin de réactiver les timers `systemd` sur les différents sites hébergeant la solution logicielle *VITAM* :

```
ansible-playbook -i environnements/<inventaire> ansible-vitam-exploitation/
start_vitam_timers.yml --vault-password-file vault_pass.txt
```

ou, si `vault_pass.txt` n'a pas été renseigné :

```
ansible-playbook -i environnements/<inventaire> ansible-vitam-exploitation/
start_vitam_timers.yml --ask-vault-pass
```

A l'issue de l'exécution du *playbook*, les *timers* `systemd` ont été redémarrés.

4.4.1.3.4 Vérification de la bonne migration des données

A l'issue de la migration, il est fortement conseillé de lancer un « Audit de cohérence » sur les différents *tenants*.

4.4.2 Notes et procédures spécifiques R8

Note : Cette version n'est à ce jour plus officiellement supportée.

4.4.3 Notes et procédures spécifiques R9

Prudence : Rappel : la montée de version vers la release R9 s'effectue depuis la release R7 (*LTS V1*) ou la release R8 (*V1, deprecated*) et doit être réalisée en s'appuyant sur les dernières versions bugfixes publiées.

4.4.3.1 Prérequis à la montée de version

4.4.3.1.1 Gestion de la rétro-compatibilité des données des offres

En préalable à l'installation, et uniquement dans le cas d'une montée de version (ne concerne pas le cas d'une installation R9 *from scratch*), il est nécessaire d'éditer le fichier d'inventaire ansible sur le modèle du fichier `deployment/environments/hosts.example` afin de décommenter la ligne ci-dessous :

```
# On offer, value is the prefix for all containers' names. If upgrading from R8, you
↳MUST UNCOMMENT this parameter AS IS !!!
vitam_prefix_offer=""
```

Cela est dû à la mise en place, à partir de la version R9, d'un préfixe au niveau des noms de conteneurs de *tenants* logiques *VITAM* sur les offres de stockage. Dans le cas d'une montée de version, cette étape préalable à l'installation permettra de garantir la rétro-compatibilité des données entre les versions précédentes et la version R9 (et supérieures).

4.4.3.1.2 Arrêt des *timers systemd*

Les commandes suivantes sont à lancer depuis le répertoire `deployment` sur les différents sites hébergeant la solution logicielle *VITAM* :

```
ansible-playbook -i environments/<inventaire> ansible-vitam-exploitation/
stop_vitam_timers.yml --vault-password-file vault_pass.txt
```

ou, si `vault_pass.txt` n'a pas été renseigné :

```
ansible-playbook -i environments/<inventaire> ansible-vitam-exploitation/
stop_vitam_timers.yml --ask-vault-pass
```

À l'issue de l'exécution du *playbook*, les *timers systemd* ont été arrêtés, afin de ne pas perturber la migration.

Il est également recommandé de ne lancer la procédure de migration qu'après s'être assuré que plus aucun *workflow* n'est en cours de traitement.

4.4.3.1.3 Reprise des données de certificats

La version R9 apporte une nouvelle fonctionnalité permettant la révocation des certificats *SIA* et *Personae*, afin d'empêcher des accès non autorisés aux *API VITAM* (vérification dans la couche https des *CRL*). Cette fonctionnalité impose d'effectuer une reprise des données des certificats (base MongoDB *identity*, collections *Certificate* et *PersonalCertificate*).

Les commandes sont à lancer depuis le répertoire `deployment` sur les différents sites hébergeant la solution logicielle *VITAM* :

```
ansible-playbook ansible-vitam-exploitation/migration_r7_certificates.yml
--vault-password-file vault_pass.txt
```

ou, si `vault_pass.txt` n'a pas été renseigné :

```
ansible-playbook ansible-vitam-exploitation/migration_r7_certificates.yml
--ask-vault-pass
```

4.4.3.1.4 Montée de version MongoDB 3.4 vers 4.0

La montée de version R7 vers R9 comprend une montée de version de la bases de données MongoDB de la version 3.4 à la version 4.0.

Les commandes suivantes sont à lancer depuis le répertoire `deployment` sur les différents sites hébergeant la solution logicielle *VITAM* :

- Arrêt de *VITAM* (*playbook* `ansible-vitam-exploitation/stop_vitam.yml`)

Avertissement : A partir de là, la solution logicielle *VITAM* est arrêtée ; elle ne sera redémarrée qu'au déploiement de la nouvelle version.

- Démarrage des différents cluster mongodb (*playbook* `ansible-vitam-exploitation/start_mongodb.yml`)
- Upgrade de mongodb en version 3.6 (*playbook* `ansible-vitam-exploitation/migration_mongodb_36.yml`)
- Upgrade de mongodb en version 4.0 (*playbook* `ansible-vitam-exploitation/migration_mongodb_40.yml`)

4.4.3.2 Montée de version

La montée de version vers la release R9 est réalisée par réinstallation de la solution logicielle *VITAM* grâce aux *playbooks* ansible fournis, et selon la procédure d'installation classique décrite dans le *DIN*.

4.4.3.3 Etapes de migration

Dans le cadre d'une montée de version R7 vers R9, il est nécessaire d'appliquer un *playbook* de migration de données à l'issue de réinstallation de la solution logicielle *VITAM*.

Prudence : Dans le cas particulier d'une installation multi-sites, il sera nécessaire de d'abord lancer la migration des données sur le site secondaire afin de purger les registres des fonds, puis de lancer la migration sur le site primaire, et enfin de lancer la reconstruction des registres des fonds sur le site secondaire.

4.4.3.3.1 Procédure de migration des données

Lancer les commandes ci-après dans l'ordre suivant :

1. D'abord sur le site secondaire pour purger les registres des fonds
2. Ensuite sur le site primaire pour la migration des registres des fonds.

```
ansible-playbook -i environments/<inventaire> ansible-vitam-exploitation/migration_r7_r8.yml --vault-password-file vault_pass.txt
```

ou, si `vault_pass.txt` n'a pas été renseigné :

```
ansible-playbook -i environments/<inventaire> ansible-vitam-exploitation/migration_r7_r8.yml --ask-vault-pass
```

Avertissement : Selon la volumétrie des données précédemment chargées, le *playbook* peut durer jusqu'à plusieurs heures.

Note : Durant la migration, il est fortement recommandé de ne pas procéder à des versements de données. En effet, le *playbook* se charge d'arrêter les composants « ingest-external » et « access-external » avant de réaliser les opérations de migration de données, puis de redémarrer les composants « ingest-external » et « access-external ».

Les opérations de migration réalisées portent, entre autres, sur les éléments suivants :

- **Les registres des fonds (Accession Registers)**
 - **Diff AccessionRegisterDetail :**
 - Suppression du champs `Identifier`, remplacé par `Opc` (Opération courante)
 - Suppression du champs `OperationGroup`, remplacé par `Opi` (Opération d'ingest)
 - Suppression du champs `Symbolic`
 - Suppression des champs `attached`, `detached`, `symbolicRemained` des sous objets (`TotalUnits`, `TotalObjectGroups`, `TotalObjects`, `ObjectSize`)
 - Ajout d'un sous objet `Events`
 - **Diff AccessionRegisterSummary :**
 - Suppression des champs `attached`, `detached`, `symbolicRemained` des sous objets (`TotalUnits`, `TotalObjectGroups`, `TotalObjects`, `ObjectSize`)
 - **Le journal des opérations**
 - Seules seront disponibles les données du registre des fonds selon le nouveau modèle dans le `evDetData` du journal de l'opération d'*ingest*.
-

Note : Se reporter à la documentation du nouveau modèle de données de la release R9.

4.4.3.3.2 Procédure de réindexation des ObjectGroup

Sous deployment, exécuter la commande suivante :

```
ansible-playbook -i environments/<inventaire> ansible-vitam-exploitation/migration_r7_r9.yml --vault-password-file vault_pass.txt
```

ou, si `vault_pass.txt` n'a pas été renseigné :

```
ansible-playbook -i environments/<inventaire> ansible-vitam-exploitation/migration_r7_r9.yml --ask-vault-pass
```

Les changement apportés touchent le mapping Elasticsearch sur l'attribut `qualifier.version` de la collection `ObjectGroup` (passé en `nested`)

Note : Ce *playbook* ne supprime pas les anciens indexes pour laisser à l'exploitant le soin de vérifier que la procédure de migration s'est correctement déroulée. A l'issue, la suppression des index devenus inutiles devra être réalisée manuellement.

4.4.3.3.3 Après la migration

Exécuter la commande suivante afin de réactiver les timers systemd sur les différents sites hébergeant la solution logicielle *VITAM* :

```
ansible-playbook -i environments/<inventaire> ansible-vitam-exploitation/start_vitam_timers.yml --vault-password-file vault_pass.txt
```


ou, si `vault_pass.txt` n'a pas été renseigné :

```
ansible-playbook -i environments/<inventaire> ansible-vitam-exploitation/
start_vitam_timers.yml --ask-vault-pass
```

A l'issue de l'exécution du *playbook*, les timers `systemd` ont été redémarrés.

4.4.3.3.4 Une fois le site secondaire *up*

Sur le site secondaire, vérifier sur les machines hébergeant le composant `functional-administration` que le processus de reconstruction des registres des fonds a bien démarré.

La commande à exécuter (en tant que `root`) est la suivante :

```
systemctl status vitam-functional-administration-accession-register-reconstruction.
service
```

4.4.3.3.5 Vérification de la bonne migration des données

A l'issue de la migration, il est fortement conseillé de lancer un « Audit de cohérence » sur les différents tenants.

4.4.4 Notes et procédures spécifiques R10

Prudence : Rappel : la montée de version vers la release R10 s'effectue depuis la *release* R9 et doit être réalisée en s'appuyant sur les dernières versions *bugfixes* publiées.

4.4.4.1 Prérequis à la montée de version

Sans objet.

4.4.4.2 Montée de version

La montée de version vers la release R10 est réalisée par réinstallation de la solution logicielle *VITAM* grâce aux *playbooks* ansible fournis, et selon la procédure d'installation classique décrite dans le *DIN*.

4.4.4.3 Etapes de migration

4.4.4.3.1 Procédure de réindexation des ObjectGroup

Sous `deployment`, exécuter la commande suivante :

```
ansible-playbook -i environments/<inventaire> ansible-vitam-exploitation/
reindex_es_data.yml --vault-password-file vault_pass.txt --tags objectgroup
```

ou, si `vault_pass.txt` n'a pas été renseigné :

```
ansible-playbook -i environments/<inventaire> ansible-vitam-exploitation/
reindex_es_data.yml --ask-vault-pass --tags objectgroup
```

Les changements apportés touchent le mapping Elasticsearch de la collection `ObjectGroup`.

Note : Ce *playbook* ne supprime pas les anciens indexes pour laisser à l'exploitant le soin de vérifier que la procédure de migration s'est correctement déroulée. A l'issue, la suppression des index devenus inutiles devra être réalisée manuellement.

4.4.4.3.2 Migration des données de certificats

La version R10 apporte une modification quant à la déclaration des certificats. En effet, un bug empêchait l'intégration dans la solution *VITAM* de certificats possédant un serial number long.

La commande suivante est à depuis le répertoire `deployment` sur les différents sites hébergeant la solution logicielle *VITAM* :

```
ansible-playbook -i environments/<inventaire> ansible-vitam-exploitation/
R10_upgrade_serial_number.yml --vault-password-file vault_pass.txt
```

ou, si `vault_pass.txt` n'a pas été renseigné :

```
ansible-playbook -i environments/<inventaire> ansible-vitam-exploitation/
R10_upgrade_serial_number.yml --ask-vault-pass
```

4.4.4.3.3 Migration des contrats d'entrée

La montée de version vers la release R10 requiert une migration de données (contrats d'entrée) suite à une modification sur les droits relatifs aux rattachements. Cette migration s'effectue à l'aide de la commande suivante, exécutée (une seule fois) sur l'une des machines `[hosts-mongod-data]` de l'inventaire :

```
mongo --host=<hosts-mongod-data> admin --username='<mongodb.mongo-data.admin.user>' --
↳password='<mongodb.mongo-data.admin.password>' migration_ingest_contract.js
```

Où `migration_ingest_contract.js` contient :

```
// Switch to masterdata database
db = db.getSiblingDB('masterdata');

// Update IngestContract
db.IngestContract.find({}).forEach(function(item)
{
  if (item.CheckParentLink == "ACTIVE") {
    item.checkParentId = new Array(item.LinkParentId);
  }

  item.CheckParentLink = "AUTHORIZED";

  //   printjson(item);
  db.IngestContract.update({_id: item._id}, item);
});
```

Table des figures

Liste des tableaux

1	Documents de référence VITAM	2
1	Tableau récapitulatif des versions de la solution logicielle VITAM	5

A

API, 2
AU, 2

B

BDD, 3

C

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

D

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

E

EBIOS, 3
ELK, 3

I

IHM, 3
IP, 3

J

JRE, 3
JVM, 3

L

LAN, 3

LFC, 3
LTS, 3

M

M2M, 3
MitM, 3

N

NoSQL, 4

O

OAIS, 4
OS, 4
OWASP, 4

P

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

R

REST, 4
RGI, 4
RPM, 4

S

SAE, 4
SEDA, 4
SGBD, 4
SIA, 4
SIEM, 4
SIP, 4
SSH, 4
Swift, 4

T

TNR, 4
TTL, 4

U

UDP, 4

UID, 4

V

VITAM, 4

W

WAF, 4

WAN, 4