Service de Traitement de l'Information de la Gendarmerie

IPMS

Résumé de la politique d'architecture

1 Suivi des versions

Version	Date	Éditeur	Modification
1.0	19 déc. 2019	E. Déchaux	Publication de la première version
1.1	21/10/2021	E. Déchaux	Prise en compte modifications politique IPMS
1.2	18/11/21	E. Déchaux	Retrait MariaDB

Table des matières

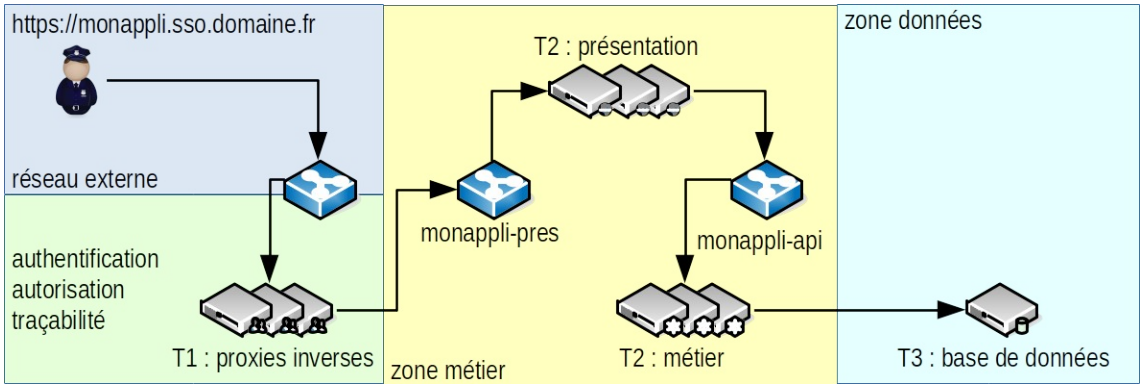
2 Introduction

2.1 But du document

Ce document est un résumé de la politique d'architecture du STIG. Pour obtenir le document complet, il suffit de vous adresser à vos contacts au sein du Ministère de l'Intérieur, du ST(SI)² ou du STIG.

2.2 Principes d'hébergement et contraintes

L'infrastructure de production mutualisée et secourue (IPMS) héberge des applications généralement construites selon un modèle n-tiers.



Le premier niveau étant la brique de serveurs mandataires inverses et d'authentification basée sur LemonLDAP-NG. Le dernier, une instance de base de données presque exclusivement relationnelle. Entre les deux, les niveaux applicatifs qui sont souvent organisés en niveaux de présentation, immédiatement derrière le SSO puis un niveau métier.

D'autres applications plus complexes peuvent utiliser un service de messagerie applicative, de stockage clé-valeur ou de multiples moteurs applicatifs sans pour autant remettre en cause ce principe d'architecture : le passage par le module de proxies inverses est obligatoire.

Toutes les applications sont hébergées sur une infrastructure virtualisée : par la solution de VMware pour les tiers 1 et 2, par la solution d'Oracle sous Solaris (LDOM, zones) pour le tiers 3.

3 La résilience

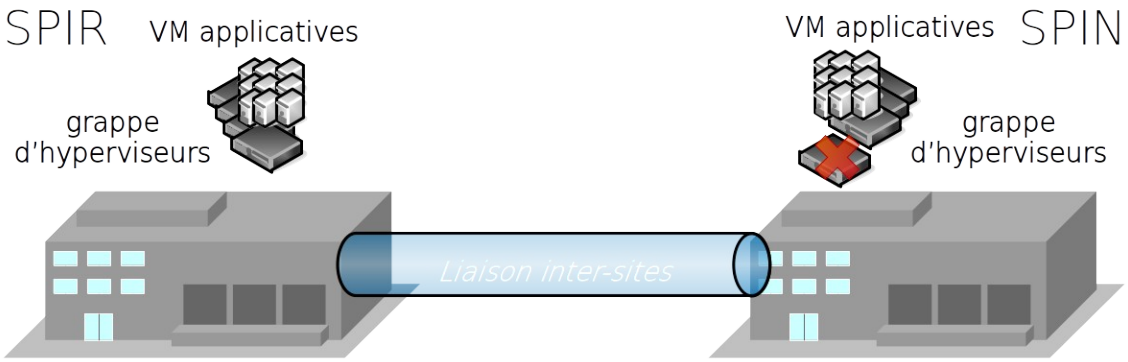
La résilience est assurée par l'infrastructure et la localisation sur deux centres de données : le SPIR et le SPIN. Il est généralement inutile voire proscrit d'utiliser des mécanismes de résilience supplémentaires.

3.1 Résilience Tiers 1 et 2

Il existe deux types de résiliences pour la partie application :

3.1.1 Active/active

L'hébergement applicatif est réparti sur les deux sites de production informatique. Sur chacun, plusieurs grappes de serveurs de virtualisation sont formées. Chacune est construite selon le principe de construction N+1 (N serveurs plus un serveur de secours) permettant la perte d'un serveur de la grappe sans impact sur les performances des machines hébergées.

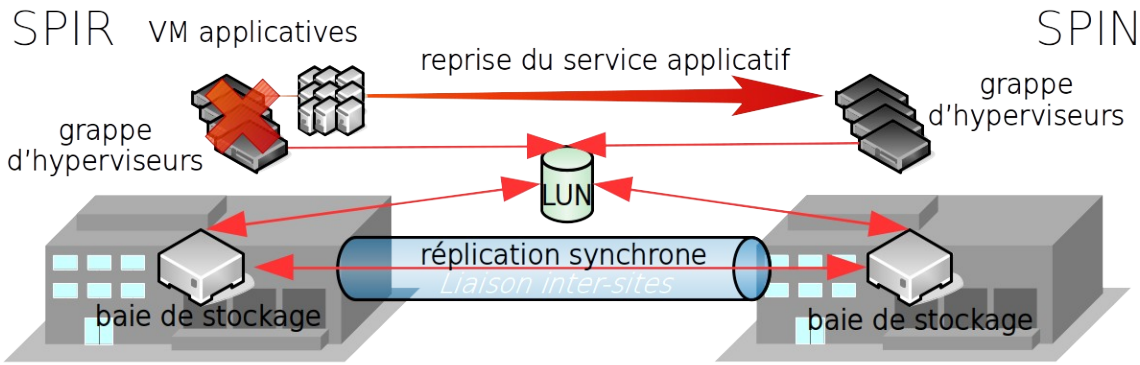


On parle de service actif/actif : les instances applicatives sont réparties sur les grappes d'hyperviseurs des deux sites. La perte d'une grappe ou d'un site implique la disparition des instances applicatives associées. L'utilisation d'un équilibreur de charges permet de rendre transparent un tel dysfonctionnement. En cas d'occurrence, seules les transactions en cours sur le serveur concerné sont impactées. La remise en service est quasi-immédiate grâce à l'équilibreur de charge.

3.1.2 Active / passive

Alors que le mécanisme précédant permet une résilience locale, un mécanisme à plus grande échelle est mis en place afin de permettre la reprise de service d'un site à l'autre. Celui-ci se base sur la technologie de stockage

hautement-disponible. Ainsi une grappe du SPIN et une grappe du SPIR sont en mesure d'utiliser simultanément le même stockage. L'objectif étant, dans un premier temps, d'être en mesure de redémarrer toutes les machines virtuelles d'une grappe sur le second site, lors de la défaillance du premier. Ce service d'hébergement est appelé actif/passif : les machines virtuelles ne sont hébergées qu'à un seul endroit.

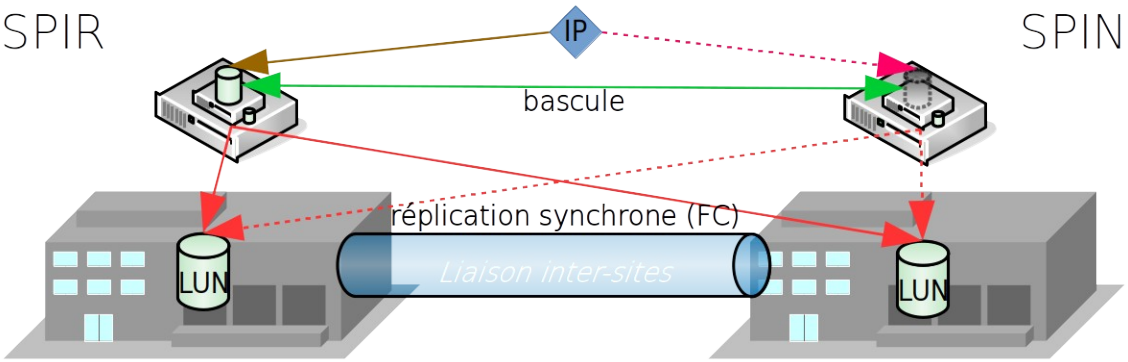


En raison du coût de mise en œuvre de cette solution – le coût du stockage est doublé car consommé sur les deux sites – elle est réservée aux applications à état, stockant les données en local. Par exemple une messagerie, une base de données, un serveur de fichiers, etc.

En cas de dysfonctionnement, le service est complètement arrêté afin d'être redémarré sur le second site. La durée d'interruption est dépendante du nombre de machines à redémarrer et peut couvrir plusieurs heures.

3.2 Résilience Tiers 3

Toutes les instances de bases de données sont configurées pour fonctionner en haute-disponibilité via un mécanisme apporté par le socle. Ainsi les serveurs sont répartis sur les deux sites et appariés afin de constituer une grappe de haute-disponibilité. De cette façon les instances de bases de données peuvent basculer d'un nœud – pouvant être une zone ou un LDOM Solaris – à l'autre de façon entièrement automatisée. A chaque instance est associé un ensemble de configurations dont une adresse IP et un stockage qui la suivent en cas de bascule. Le stockage est répliqué entre les deux nœuds via la mise en œuvre d'un miroir au niveau du gestionnaire de volumes.



Un quorum externe, sur un troisième site,est en place sur un afin de permettre à l'architecture de gérer correctement les partitions des réseaux SAN et LAN.

4 Technologies de stockage

Les trois technologies standards du marché de stockage sont offertes sur l'IPMS :

- mode bloc, via deux baies « full flash » pour le stockage principal. Pour la sauvegarde, deux baies hybrides « ssd + sata » sont utilisées ;
- mode fichiers, via une grappe de deux passerelles NAS connectées aux baies de stockage principales. Les protocoles NFS et CIFS sont disponibles. En parallèle, les deux baies hybrides fournissent également un stockage fichier local ;
- mode objet, via une infrastructure distribuée servant le protocole S3.

5 Équilibrage de charge

Le seul point d'entrée, connu du client, est un URL pointant vers une adresse IP virtuelle publique de l'équilibreur de charge. Toutes les autres adresses IP virtuelles de l'équilibreur de charge sont internes et non accessibles directement par l'utilisateur.

Chaque module applicatif dispose de sa propre adresse IP virtuelle interne afin qu'il puisse être résilient.

Dans le cas où un module applicatif s'auto-consomme – c'est-à-dire qu'il s'appelle lui-même – il est préférable de le faire via l'adresse IP virtuelle interne du module plutôt que de s'interroger sur sa propre adresse IP afin de répartir la charge sur toutes les machines du module.

5.1 Test de vie

Lors de l'utilisation d'un équilibreur de charge, la qualité du service rendu est entièrement liée à la qualité de détection de bon fonctionnement des instances applicatives. Cette détection est réalisée à travers un test de vie exécuté par l'équilibreur de charge à intervalles réguliers sur chaque instance. Ce test de vie, propre à chaque module applicatif, doit être intégré aux livraisons applicatives.

Il est généralement intégré sous forme d'une page appelée par l'équilibreur de charge via le protocole HTTP. En fonction d'un message affiché ou d'un code de retour HTTP, celui-ci décide des évolutions d'état de chaque instance applicative.

5.2 Affinité de session

Nous utilisons trois mécanismes d'affinité de session applicative au niveau de l'équilibrage de charge :

- par adresse source, pour tous les protocoles basés sur TCP et UDP ;
- par insertion de cookie pour les protocoles HTTP/HTTPS ;
- aucun. Il s'agit de la configuration recommandée pour les services sans état.

Lors d'une demande d'adresse IP virtuelle, il faut préciser le mécanisme souhaité. Par défaut, une affinité de session par insertion de cookie est mise en œuvre pour les protocoles basés sur HTTP. La désactivation de l'affinité de session doit être explicitement demandée par le chef de projet afin d'être réalisée en connaissance de cause avec acceptations des impacts.

5.3 Incompatibilités

L'utilisation d'un équilibreur de charge est incompatible avec le protocole Apache Jserv Protocol ou AJP. Celui-ci ne doit donc pas être utilisé et remplacé par le protocole HTTP.

En effet l'aspect binaire du protocole le rend très opaque au diagnostic et ses fonctionnalités intrinsèques sont très largement étendues par l'utilisation d'un équilibreur de charge matériel :

- présence de tests de vie applicatif ;
- meilleure gestion des équilibrages et persistances ;
- capacités d'accélérations matérielles ;
- capacités de multiplexage et réutilisation des sessions TCP via HTTP 1.1 et supérieur ;
- compatible avec n'importe quelle technologie, pas uniquement le couple Apache httpd / Java.

6 Systèmes d’exploitation

En fonction du besoin, le STIG mettra à disposition des demandeurs différents systèmes d’exploitation.

Tous les systèmes sont installés à partir d’une image de base sur laquelle est appliquée une configuration de base commune à toutes les installations. En conséquence, il n’est pas admis, la livraison d’une image préconfigurée : la capacité du STIG à prendre en main une application dépend de la compréhension de tous les modules applicatifs, de leur méthode d’installation et de leur paramétrage.

6.1 Debian GNU/Linux

C’est le système d’exploitation par défaut, utilisé lorsqu’il n’y a pas de contre indication justifiée. Les constructions sont réalisées à partir d’une configuration durcie afin de tendre vers le niveau renforcé de l’ANSSI.

6.2 Red Hat Enterprise Linux

C’est la distribution Linux commerciale à utiliser lorsqu’un produit commercial sur étagère doit être installé. L’utilisation de cette distribution n’est autorisée qu’à partir du moment où le projet a mis en place un contrat de refacturation permettant d’abonder le budget IPMS du montant du maintien en conditions opérationnelles de ce système d’exploitation.

6.3 Windows Server

L’utilisation de Windows Server est suspendue.

7 Système de gestion de bases de données

Le STIG est en mesure de réaliser une administration des moteurs de bases de données Oracle, PostgreSQL et MySQL.

Il n'y a pas d'offre de service pour tous les autres systèmes de gestion de base de données, qu'ils soient relationnels, objets ou orientés documents. Ils sont donc exploitées de façon générique selon le principe d'obligation de moyen. La mise en place d'une stratégie autour d'un produit spécifique, d'un plan de formation adapté et d'un délai acceptable de mise en œuvre d'un socle pourra permettre d'étendre le catalogue.

7.1 SGBD Oracle

La mise en œuvre de bases de données Oracle n'est pas souhaitée et son utilisation est soumise à validation du comité des applications, à défaut le sous-directeur des applications de commandement, à défaut le chef du BCOF.

La version recommandée est la dernière version disposant d'un support à long terme.

Seule la version « entreprise » est installée et toutes les fonctionnalités optionnelles sous licences additionnelles sont désactivées et interdites d'utilisation. Il s'agit notamment du partitionnement, de l'extension spatiale et du tuning pack.

7.2 SGBD Mysql

Seule la version communautaire de MySQL est utilisée. Il n'y a pas de contrainte d'utilisation particulière. Aucune extension n'est fournie. La version recommandée est la dernière disponible.

7.3 SGBD Postgresql

Ce moteur représente la technologie recommandée pour l'hébergement de bases de données relationnelles. La version recommandée est la dernière disponible.

8 Moteurs applicatifs

8.1 Versions utilisées

Par défaut, les moteurs applicatifs fournis avec les distributions sont utilisés. Cela apporte beaucoup de souplesse dans la gestion du parc, son uniformité et son cycle de vie et la sécurité globale du système d'information. Cela implique l'utilisation de versions plus anciennes mais plus éprouvées.

Il n'est plus admis l'utilisation de moteurs applicatifs fournis par les chefs de projets en raison de l'absence de suivi des mises à jour de sécurité.

Il reste cependant possible d'utiliser un dépôt tierce lorsque le moteur n'est pas disponible dans la version actuelle du système d'exploitation. Par exemple il est possible d'utiliser le dépôt de PostgreSQL ou de Microsoft pour fournir un client PostgreSQL à jour ou .NET Core absent des distributions.

8.2 Conteneurs applicatifs

L'usage de conteneurs sur IPMS est détaillé dans le document de politique[1] adhoc.

8.3 Technologies nouvelles

Le STIG propose un hébergement sur deux sites. L'utilisation d'une technologie nécessitant l'utilisation de 2N+1 instances pour fonctionner nécessite la validation de l'architecture par le groupe architecture et expertise du STIG.

1 POL GAE CONTENEURS IPMS.

9 Exploitation

9.1 Installations automatisées

Si l'application est prévue d'être installée par un mécanisme automatisé tel qu'un script ou un playbook Ansible, celui-ci doit être livré et documenté dès la première installation.

9.2 Gestion des journaux

Chaque application doit gérer ses journaux, c'est à dire leur taille, durée de conservation et rotation. Elle doit également séparer les journaux techniques des journaux fonctionnels.

9.3 Sécurité

Les flux entrant ou sortant du centre de données doivent être chiffrés, idéalement avec le protocole HTTPS et utiliser une double authentification via certificat X509.

Les systèmes d'exploitation sont mis à jour par le STIG des patches de sécurité lorsqu'ils sont disponibles. Cela est imposé aux chefs de projets qui disposent d'un préavis de 7 jours pour réaliser les tests de retro-compatibilité s'ils sont nécessaires.

9.4 Métrologie

Si une application nécessite la collecte d'indicateurs de métrologie spécifiques, le système de collecte et de présentation doit être fourni par l'équipe projet qui peux mutualiser ce système à plusieurs de ses applications. Le système de métrologie doit cependant être totalement indépendant des application (chaînage faible)

10 Maintien en condition de sécurité

Toutes les briques d’une application, du système d’exploitation, aux moteurs de bases de données en passant par les moteurs applicatifs et les bibliothèques applicatives intégrées doivent faire l’objet d’une maintenance régulière afin d’appliquer les correctifs de sécurité et réaliser les évolutions de versions nécessaires en cas d’obsolescence.

11 Règles de l’art

Le respect des règles de l’art est impératif pour toute application. Ainsi celles-ci doivent présenter, entre autres, les caractères de qualité basique suivants :

- les secrets nécessaires à la connexion aux systèmes tiers sont définis et simplement modifiables par l’exploitation. Il ne sont en aucun cas connus des chefs de projets ;
- de la même façon, les paramètres de connexion à ces systèmes tiers, tels que les noms d’hôtes, doivent également être modifiables par l’exploitant sans nécessiter de nouvelle livraison applicative ;
- le niveau de verbosité des journaux doit être configuré au juste niveau. Une application en production ne doit pas avoir des journaux en mode diagnostic en permanence ;
- il n’existe pas d’erreur normale. Si un message d’erreur est tracé dans les journaux, il doit exister une action pour remédier à cette erreur. Dans le cas contraire ce message ne doit pas apparaître ;
- les applications doivent s’exécuter selon le principe du moindre privilège. Aucun ne doit s’exécuter avec des privilèges super-utilisateur ;
- des mécanismes de purge de données doivent être proposés afin de ne conserver en ligne que les données strictement nécessaires. Ne pas respecter.