La rétention des données des Schedulers

Sommaire

Concept de la rétention

Dans Shinken Entreprise, lorsque des éléments sont en supervision, des vérifications régulières sont effectuées sur les hôtes, clusters et checks.

Suite à ces vérifications, un statut (OK, Attention, Critique, Inconnu) ainsi que un ou plusieurs contextes (Flapping, Période de maintenance, Prise en compte) sont attribués à chaque élément.

Sans rétention, lorsque Shinken doit être redémarré (maintenance du serveur de supervision, ou bien mise à jour de Shinken), ces statuts et contextes sont perdus, et les éventuelles notifications déclenchées sur un état non voulu seront envoyées !

Activer la rétention permet de conserver les états des hôtes, clusters et checks entre les redémarrages de Shinken et ainsi bénéficier d'une vision claire de l'état des éléments supervisés à tout moment.

Cette rétention s'effectue au niveau du démon Scheduler qui est chargé d'ordonnancer la vérification des éléments et de récupérer et analyser les résultats de ces vérifications.

Les différents types de rétention

Plusieurs modules de rétention sont livrés dans une installation de Shinken Entreprise afin d'être utilisés dans le Scheduler.

- La Rétention par fichier plat
  - Il s'agit du type de rétention le plus simple. C'est la rétention par fichier plat (module PickleRetentionFile) qui est d'ailleurs utilisée par défaut dans les Schedulers.
  - Avec ce type de module, les statuts et contextes des éléments sont sauvegardés dans un fichier plat, sur le même serveur que le Scheduler sur lequel il est activé.

- La rétention par une base de données
  - Ce type de rétention est utilisé par le module MongodbRetention, qui stocke les statuts et contextes des éléments supervisés dans une base de données Mongo.
  - Ce module a l'avantage de pouvoir être utilisé avec des architectures de Shinken distribuées sur plusieurs serveurs, avec plusieurs schedulers.

Cadre d'utilisation des différents types de modules de rétention

Utilisation de la rétention par fichiers plats

Par défaut dans Shinken Entreprise, les Schedulers utilisent le module PickleRetentionFile, qui sauvegarde les données de rétention dans un fichier plat.

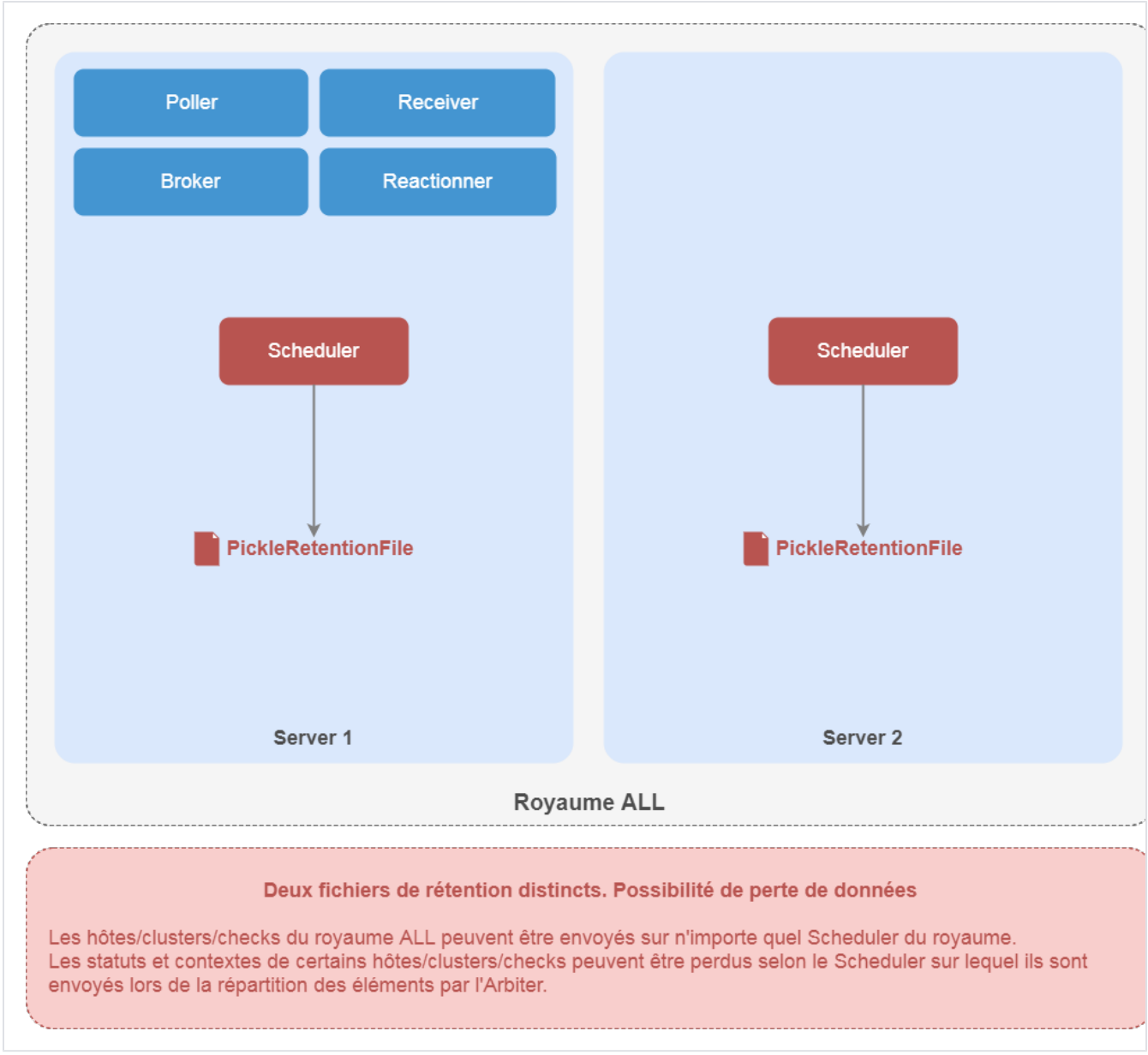
Ce module, simple et rapide, possède cependant ses limites dans un environnement distribué.

Limitations de la rétention par fichiers plats

Le cas de figure montrant les limites de la rétention par fichiers plats est le suivant:

- On dispose dans notre architecture Shinken d'un royaume possédant plusieurs Schedulers.

- A chaque démarrage de l'Arbiter, celui ci répartit les hôtes du royaume de manière équitable entre les Schedulers de ce royaume. Chaque Scheduler sauvegarde donc les données de rétention dans le fichier qu'il a à disposition.
- Au prochain démarrage de l'Arbiter, celui ci va redistribuer à nouveaux les hôtes du royaume dans les Scheduler. Mais, rien ne garantit que la répartition soit identique à la précédente.
- Un hôte qui était auparavant sur le Scheduler 1, avec ses données de rétention sauvegardées dans le fichier du serveur 1, est maintenant sur le Scheduler 2. Le Scheduler 2 ne possède pas les données de retention de cet hôte, on a alors une perte de données

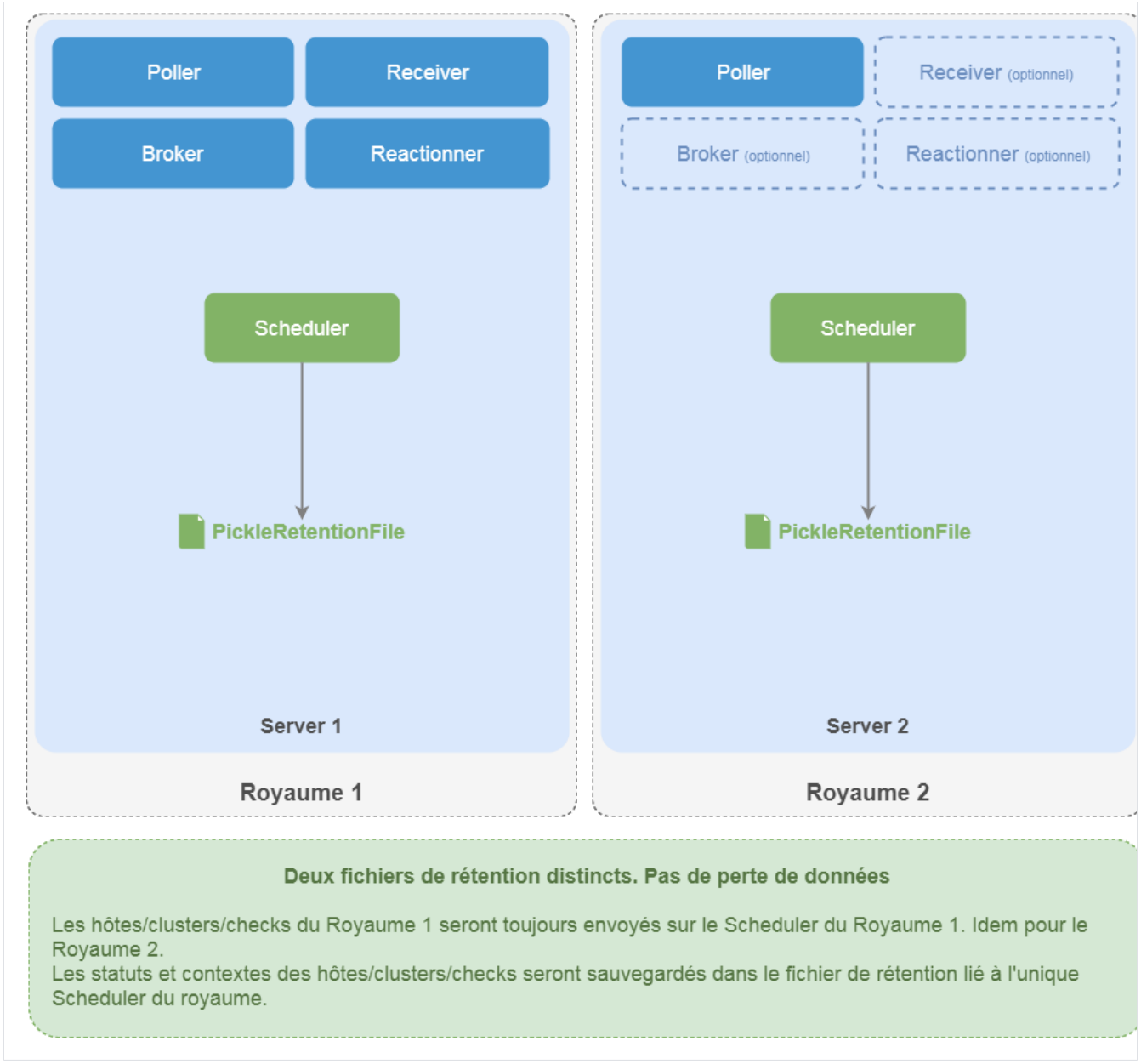


Un cas d'utilisation valide avec plusieurs royaumes

Lorsque chaque royaume possède un unique Scheduler, le cas de figure précédent ne se produit pas.

Chaque hôte, affecté à un royaume, sera distribué sur l'unique Scheduler du royaume, qui aura toutes les données de rétention nécessaires.

On ne retrouve pas le cas précédent provoquant la perte de données.



Utilisation de la retention Mongo

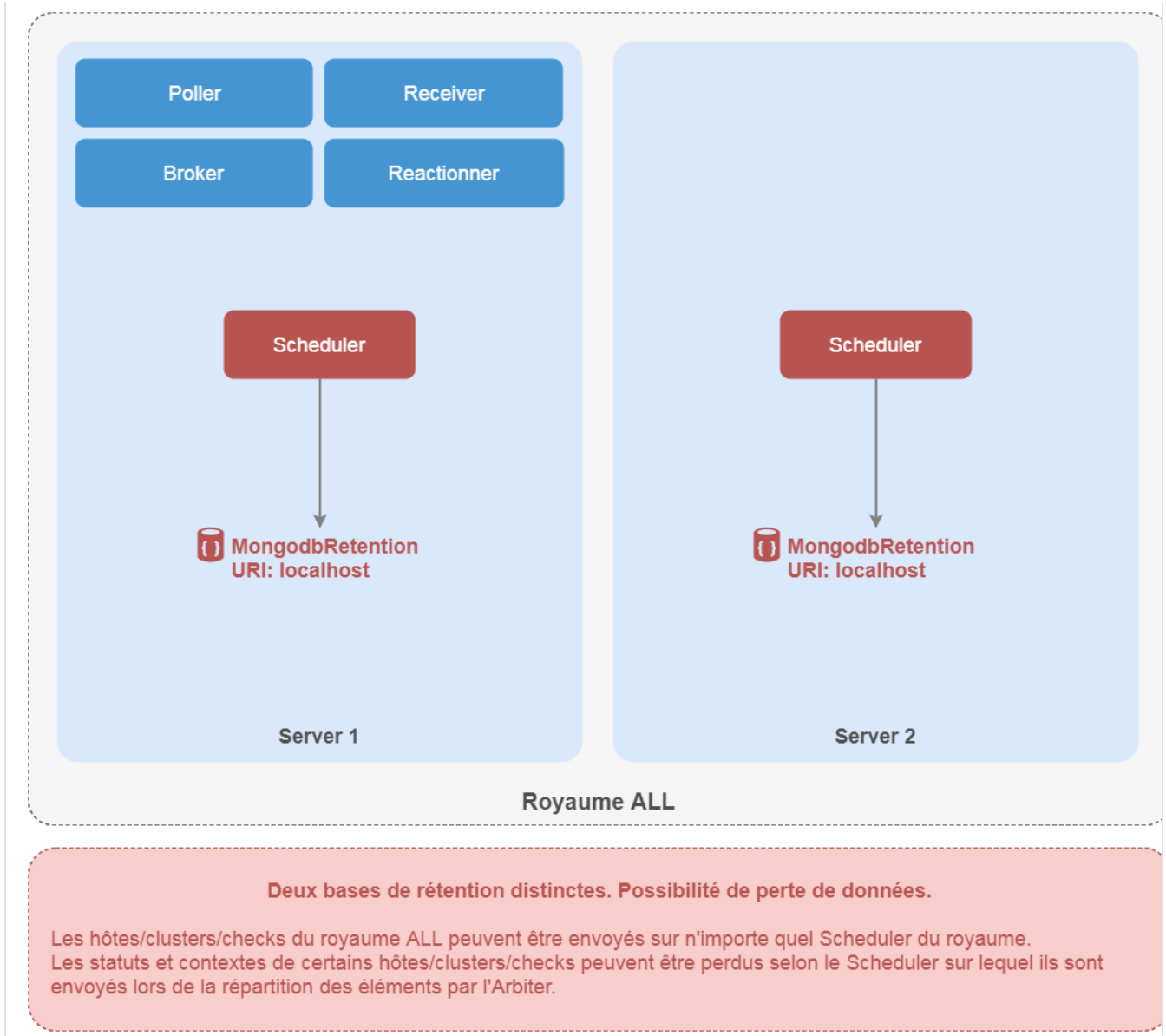
Pour pallier au cas d'erreur lié à l'utilisation d'une rétention par fichier plat dans un royaume avec plusieurs Schedulers, la rétention avec Mongo doit être utilisée.

Cependant, la configuration doit être vérifiée pour éviter de tomber dans des cas d'erreurs classiques.

Une erreur de configuration classique du module de rétention Mongo

Par exemple, on est dans un environnement possédant un royaume avec plusieurs Schedulers, sur des serveurs différents.

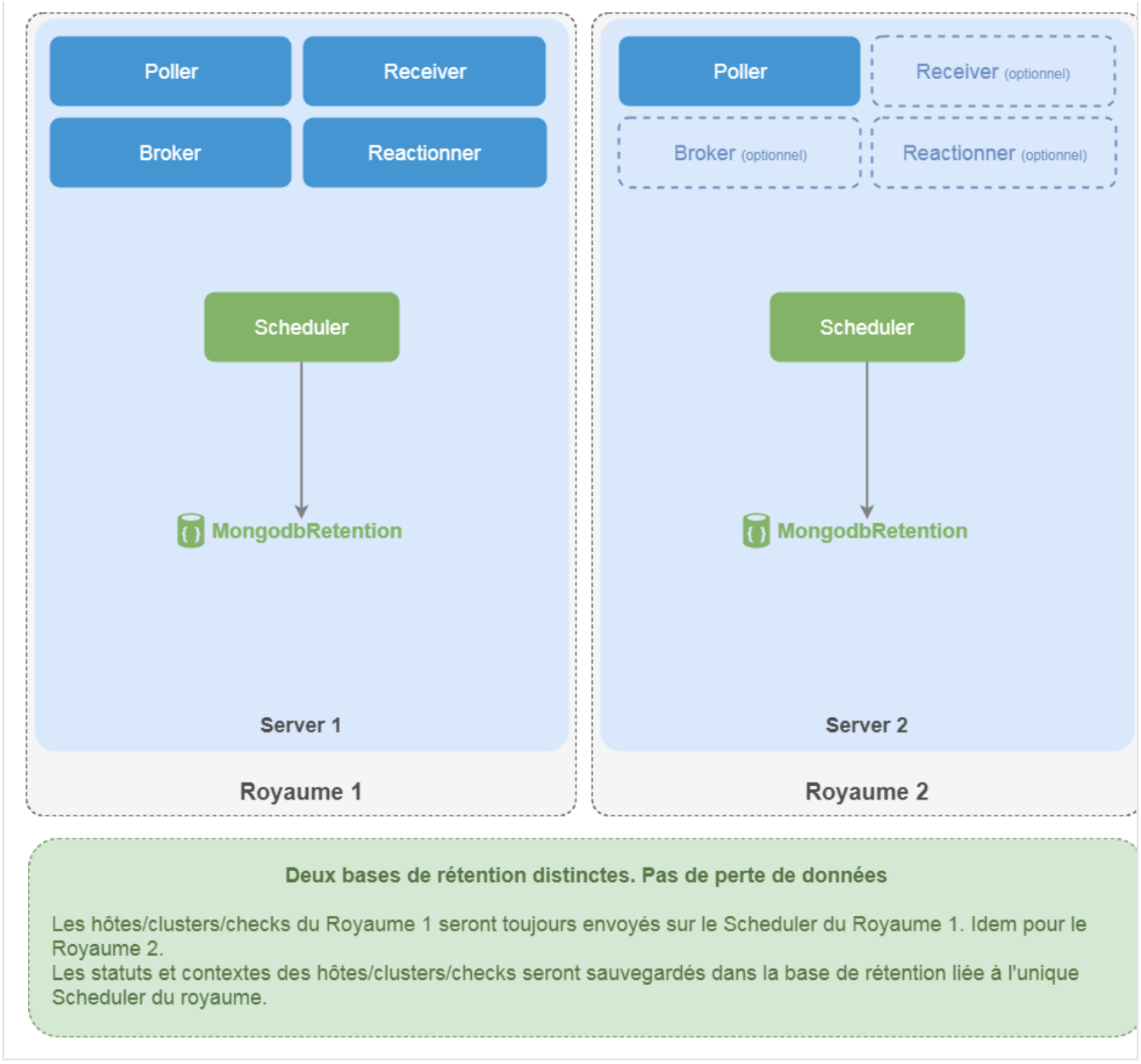
- Sachant que la rétention par fichier plat ne pourra pas fonctionner, on active sur nos Schedulers la rétention Mongo. Mais par defaut, la configuration du module de retention Mongo précise localhost comme adresse de la base à utiliser.
- On se retrouve alors avec la rétention enregistrée sur 2 bases distinctes, et on est dans le même cas d'erreur qu'avec la rétention par fichiers.



Des cas d'utilisation valides

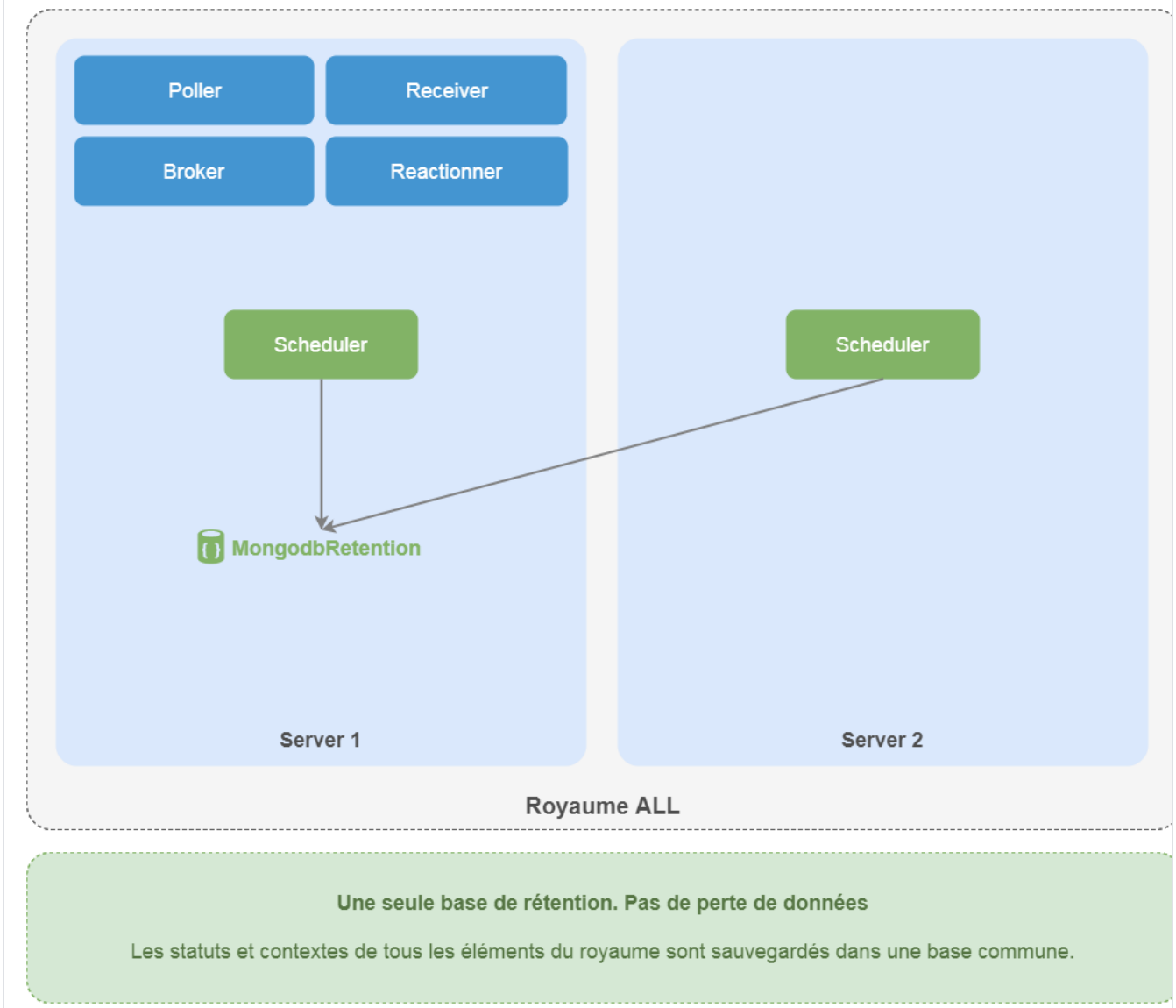
Aussi, de la même manière que dans le cas de la rétention par fichier plats, ce problème disparait lorsqu'on possède des royaumes avec un seul Scheduler.

Chaque Scheduler a accès à tous les éléments du royaume, et a donc toutes les données de rétention nécessaires.



La solution pour utiliser la rétention Mongo dans un royaume avec plusieurs Schedulers est de spécifier une adresse d'un serveur Mongo (autre que localhost) dans la configuration de la rétention Mongo.

Dans ce cas la, tous les Schedulers du royaume sauvegarderont les données de rétention dans une base de données centrale. En cas de besoin, toutes les données de retention seront disponible au même endroit et la restauration des statuts et contextes pourra être effectuée sans problèmes.



Comment configurer la rétention

Changer le module de rétention d'un Scheduler

Le choix du module de rétention s'effectue directement dans le fichier de configuration du Scheduler en question. Dans Shinken Entreprise, les Schedulers sont définis dans /etc/shinken/schedulers.

/etc/shinken/schedulers/mon_scheduler.cfg

```
define scheduler {
...
...
...
    #======== Modules to enable for this daemon =========
    # Available:
    # - PickleRetentionFile : (if you have only one scheduler into a realm) save retention data (element
state and scheduling) into a file
    # - MongodbRetention    : (if you have more than one scheduler into a realm) save retention data
(element state and scheduling) into a mongodb database
    modules                 MongodbRetention
...
...
...
}
```

Configurer la rétention Mongodb

Une fois la rétention Mongo activée sur les Schedulers concernés, il faut modifier l'URI de la base Mongo pour pointer vers l'adresse de la base de données qui hébergera les données de rétention. Des explications détaillées sur la configuration du module de rétention Mongo se trouve dans la page Rétention Mongodb.

L'installation de Shinken comporte une installation de Mongo. Il est donc possible d'utiliser un serveur Shinken comme serveur utilisé pour la rétention.

Bien évidemment, un serveur externe peut être utilisé pour sauvegarder la rétention.

/etc/shinken/modules/retention-mongodb.cfg

```
define module {


    #======== Module identity =========
    # Module name. Must be unique
    module_name     MongodbRetention

    # Module type (to load module code). Do not edit.
    module_type     mongodb_retention


    #======== Mongodb connection =========
    # uri: to connect the mongodb server
    uri             mongodb://ip_du_serveur/?safe=false
    use_ssh_tunnel  0
    ssh_user        shinken
    ssh_keyfile     ~/.ssh/id_rsa

    # database: which mongodb database to use
    database        shinken

    # Advanced option if you are running a cluster mongodb environnement
    # replica_set

}
```

Il se peut également que plusieurs royaumes veulent définir une rétention Mongo sur un serveur différent pour chaque royaume. Dans ce cas, il faut faire plusieurs définition de module de rétention.

- Le module_type sera identique, tandis que le reste de la configuration du module pourra changer.
- Il faudra ensuite, dans la configuration du Scheduler, spécifier le nom du module approprié.

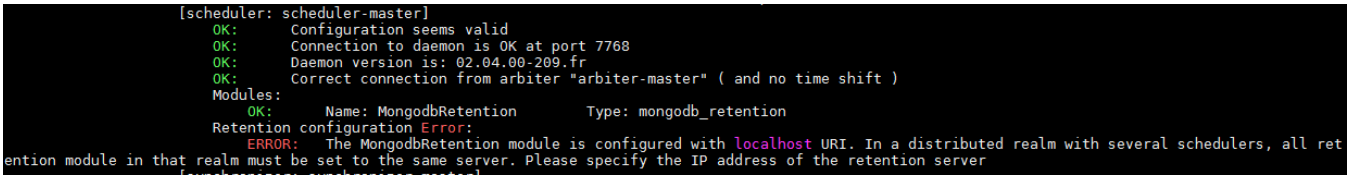
Important

Même si ce n'est pas obligatoire, nous vous conseillons de faire un fichier séparé par définition de module nommé du nom du module de rétention (dans un but de clarté de votre configuration)

Vérifier la configuration de la rétention

Afin de vérifier cette configuration et de la valider de votre choix sur la rétention d'un scheduler, vous pouvez utiliser la commande shinken-healthcheck.

- En cas d'erreur de configuration, elle sera signalée dans le retour de la commande shinken-healthcheck ainsi que dans les logs du Scheduler au démarrage de ce démon.



Erreur de configuration et Arbiter

Si la configuration de la rétention au niveau d'un ou plusieurs Schedulers est incorrecte, l'Arbiter refuse de démarrer.

La command *shinken-healthcheck* permet de détecter ces erreurs avant qu'elles ne deviennent problématiques lors du démarrage de l'Arbiter.

Comment changer de module de rétention sans perte de données

L'intérêt des modules de rétention du Scheduler est multiple:

- Assurer la continuité des données de SLA et des statuts dans l'interface de Visualisation
- Empêcher l'envoi potentiel massif de notifications lors du redémarrage du Scheduler

Lorsqu'on change de type de rétention sur un Scheduler, il faut suivre certaines étapes particulières. Si on change directement de module et qu'on redémarre, le Scheduler démarre avec le nouveau module de rétention qui va charger une rétention vide.

La description de la procédure pour migrer la rétention sans perte de données se trouve dans une page dédiée: Changement de type de rétention sans perte de données

Résumé

A retenir

- La rétention par fichier plat ne peut pas être utilisée dans un royaume comportant plusieurs Schedulers
- Il faut dans ce cas la utiliser une rétention Mongodb, dont l'adresse de la base de rétention pointe vers un serveur particulier
- Pour la rétention Mongodb, nous vous conseillons fortement d'établir un tunnel SSH pour l'établissement des connexions (plus d'information ici)

Remarques

- Les Schedulers en spare doivent également être pris en compte dans la configuration de la rétention. Un Scheduler spare avec une rétention mal configurée peut provoquer une perte des données de rétention lorsqu'il devient actif.
- Une mauvaise configuration des modules de rétention sur les schedulers en spare (ou non) entraînera des erreurs visibles au lancement de la commande shinken-healthcheck. Dans le cas d'erreur de configuration de Schedulers non spare, L'Arbiter pourra même refuser de se lancer et affichera les erreurs correspondantes dans ses Logs.
- Dans un royaume, tous les Schedulers doivent avoir le même type de rétention. Par contre, un royaume peut avoir une rétention Mongo pendant qu'un autre a une rétention par fichiers.