

Définir de nouveaux royaumes

Multi clients et/ou datacenters: ROYAUMES

L'architecture de Shinken Enterprise comme vue précédemment permet d'avoir un emplacement unique pour la configuration et pour les données. Les hôtes sont répartis parmi les Schedulers, qui définissent les commandes à exécuter. Les Pollers viennent alors récupérer ces tâches pour les exécuter.

Ou presque ! En fait, si vous avez une architecture distribuée sur plusieurs continents, vous pouvez avoir des problèmes. Si l'architecture est commune à plusieurs réseaux, un Scheduler d'un client A peut avoir un Poller d'un client B lui demandant des tâches, ce qui n'est pas une bonne idée pour des questions d'efficacité du réseau (même avec un réseau distribué).

C'est là que devient intéressante la gestion des clients par site. Dans Shinken Enterprise, on le gère à travers les "royaumes".

Un royaume est un groupe de ressources qui va gérer les hôtes et les groupes d'hôtes. Ce lien doit être unique, un hôte ne peut pas appartenir à plusieurs royaumes. Si vous mettez un groupe d'hôtes dans un royaume, tous les hôtes rattachés feront partie de ce royaume (sauf si vous avez déjà défini un royaume particulier sur un hôte) .

Un royaume, c'est :

- au moins un Scheduler
- au moins un Poller
- au moins un Reactionner (facultatif si un royaume parent possède un Reactionner avec l'option `manage_sub_realms` à 1)
- peut avoir un Broker
- peut avoir un Receiver

Dans un royaume, tous les pollers vont prendre les tâches de tous les schedulers de ce royaume.



Important

Il n'y a qu'**UN SEUL Arbitre** (et son spare) pour **TOUS** les royaumes. L'Arbitre gère TOUS les royaumes et ce qui s'y rattache.

Sous-royaumes

Un royaume peut avoir des sous-royaumes. Cela ne change rien pour les schedulers, mais peut être utile pour les autres satellites et spares. Les reactionners et brokers sont liés au même royaume, mais ils peuvent traiter les tâches des sous-royaumes également. De cette façon, vous pouvez avoir moins de reactionners et de brokers.

Cela se fait grâce au paramètre `manage_sub_realms`. Pour les pollers, la valeur par défaut est 0, mais c'est 1 pour les reactionners/brokers.

Exemples d'utilisation des royaumes

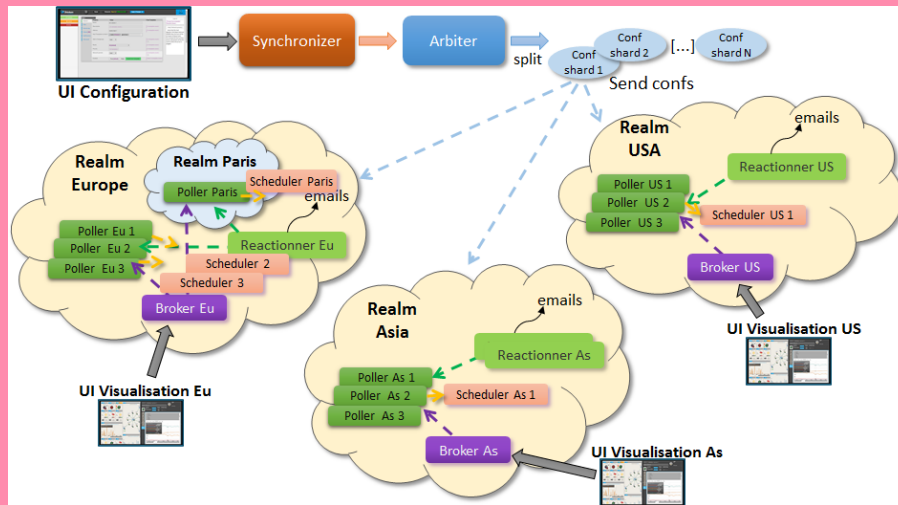
Pour faire simple : vous mettez vos hôtes et groupes d'hôtes dans un royaume. Celui-ci est considéré comme un pool de ressources. Vous n'avez pas besoin de modifier la définition de vos hôtes et groupes d'hôtes si vous avez besoin de plus/moins de performance dans le royaume ou si vous souhaitez rajouter des satellites.

Les royaumes sont une manière de gérer les ressources. Ce sont plusieurs petits "nuages" dans votre infrastructure cloud globale.

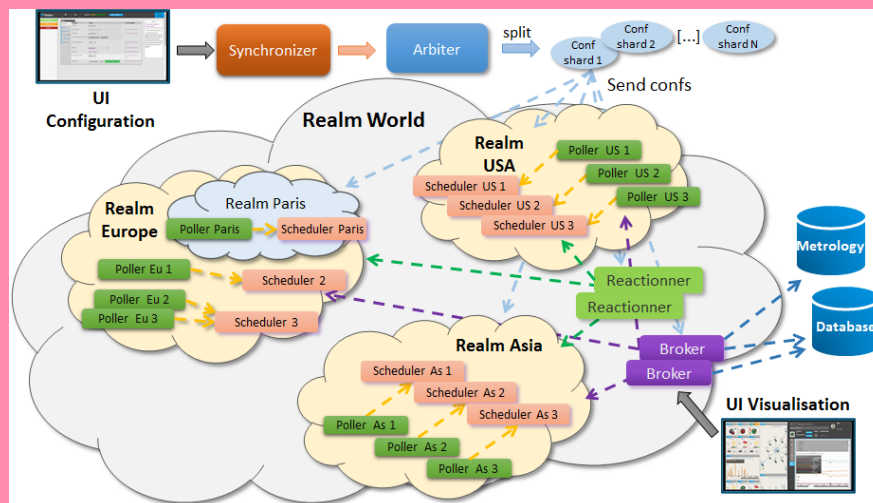
PS : cette fonctionnalité est optionnelle, un royaume par défaut est créé.

Prenons 2 exemples d'architecture distribuée à travers le monde. Dans le premier cas, l'administrateur ne souhaite pas partager les ressources entre royaumes. Dans le second, les Reactionners et Brokers sont partagés entre royaumes (donc toutes les notifications sont envoyées d'un seul endroit, idem pour le stockage des données).

Version isolée (cas 1)



Version partagée (cas 2)



Comme vous le voyez, tous les éléments sont dans un royaume unique, on utilise le sous-royaumes pour les Reactionners/Brokers.

Royaume Monde et ses sous-Royaumes (Configuration)

Voici la configuration pour l'architecture distribuée :

Realm

```
define realm {
    realm_name      World
    # Now you define SUB REALMS of World
    realm_members   Europe,US,Asia
    # Element without explicit realm setting will be set in the World realm
    default         1
}

# We define our SUB REALMS
# EUROPE
define realm {
    realm_name      Europe
    # This one have it's own SUB REALM
    realm_members   Paris
}
# Paris: sub realm for Europe
define realm {
    realm_name      Paris
}

# USA
define realm {
    realm_name      USA
}

# Asia
define realm {
    realm_name      Asia
}

# For example the daemons for the Paris realm
define scheduler {
    scheduler_name   scheduler_Paris
    realm           Paris
}

# Example of a TOP level realm (WORLD) daemon that can reach daemons of the SUB realms
# so will reach Europe, Paris, USA and Asia
define reactionner {
    reactionner_name reactionner-master
    realm           World
}
```



Vous devez placer ces configurations dans `/etc/shinken/realms`. Pour une gestion plus facile des royaumes, nous conseillons de placer chaque définition de royaumes dans un fichier dédié (royaume All dans `all.cfg`, royaume Paris dans `paris.cfg`, etc...).

De la même manière, nous conseillons aussi fortement de placer les définition des démons dans le dossier approprié puisque les outils livrés avec Shinken supposent que ces conventions sont respectées. Ainsi, on placera par exemple les définitions des Reactionners dans `/etc/shinken/reactionners`, des Brokers dans `/etc/shinken/brokers` etc...

Le lien de l'hôte vers le royaume est fait dans sa page de configuration :

Zone de travail

Hôte dans la Zone de travail > En édition (créé) Switch

Général *	Propriété	Valeur
Données [0]	Nom *	Switch ?
Droits de l'utilisateur	Description	[Par défaut, le nom] ?
Supervision	Adresse	[Par défaut, le nom] ?
Checks [0]	Modèles d'hôte hérités	[Par défaut {Aucun}] ?
Notifications	Ajouter dans le groupe d'hôtes	☒ [Par défaut {Aucun}] ?
Expert	Royaume	? Par défaut [All]
	Impact métier	? <input type="range"/> Par défaut ★★★

Brokers multi-niveaux

Dans les exemples précédents, si vous mettez plusieurs brokers dans le royaume, chaque Scheduler aura accès à un seul broker en même temps. Il est impossible d'avoir un Broker commun pour tous.

Vous pouvez activer le multi-brokers avec le paramètre **broker_complete_links** (0 par défaut).

Vous devez également activer cette option dans **TOUS** les royaumes. Par exemple :

Realm

```
define realm {
    realm_name Europe
    broker_complete_links 1
}
```

Cela permettra à chaque Scheduler d'être lié à chaque broker. Cela permet également d'avoir un broker dédié dans un même royaume (un pour l'interface web et un autre pour Graphite par exemple).