

Présentation détaillée

Organisation de la fonctionnalité

Dans Shinken Entreprise, les addons sont des fonctionnalités supplémentaires faisant en général appel à des outils externes à Shinken pouvant être activés et désactivés. La visualisation de l'architecture de l'installation Shinken est rendue possible grâce à l'addon "**nagvis-shinken-architecture**".

Lorsque la configuration de l'architecture est modifiée (configuration des démons ou des royaumes), Shinken effectue deux opérations:

- Il génère des représentations visuelles de l'architecture qui seront ensuite rendues disponibles pour les utilisateurs Shinken, et accessibles depuis l'interface de configuration.
- Il génère des hôtes dans l'interface de Configuration. Ces hôtes seront ensuite utilisés par les vues pour afficher le statut des démons Shinken.



Cette fonctionnalité permet d'afficher l'organisation des royaumes, démons et les hôtes (avec leur adresse) qui interviennent dans l'installation Shinken. Pour des raisons de confidentialité, **son accès est uniquement réservé aux utilisateurs définis comme Administrateurs Shinken** dans la configuration.

Génération des représentations visuelles de l'architecture

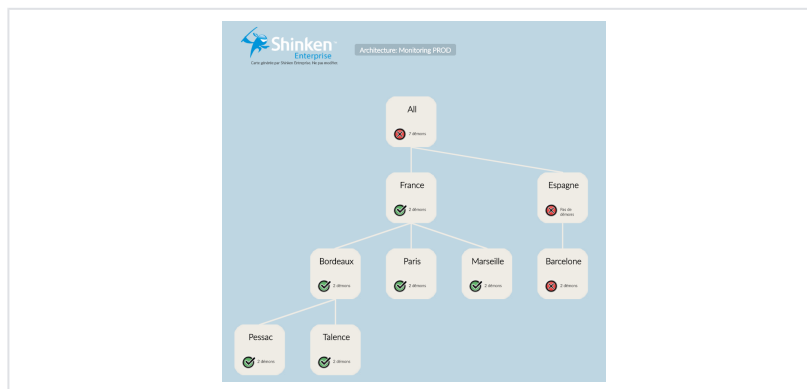
Lorsqu'une modification de l'architecture est détectée, Shinken crée des visualisations de son architecture de 3 types différents.

Arbre des royaumes

Cette vue permet de visualiser rapidement l'agencement des différents royaumes de l'installation Shinken. Les royaumes sont affichés sous forme d'ensemble d'arbres.

Chaque royaume est relié à ses royaumes fils et royaumes parents. Les royaumes de plus haut niveau sont affichés en haut, puis les sous-royaumes successifs sont affichés en dessous. Les royaumes de plus haut niveau seront donc ceux affichés en haut de la page.

Dans l'exemple, le royaume "**All**" est le royaume principal, et il a comme sous-royaumes les royaumes France et Espagne. Ces royaumes ont ensuite leurs propres sous-royaumes, affichés de la même manière.



Individuellement, chaque royaume est affiché possède les informations suivantes:

- **Son nom**
- **Le nombre de démons qu'il possède**
- **Une icône de statut**, qui affiche le statut consolidé de tous les démons contenus dans ce royaume. Un clic sur cette icône amène vers la vue détaillée du royaume, dans laquelle on peut voir l'ensemble des sous-royaumes, machines et démons contenus dans le royaume.



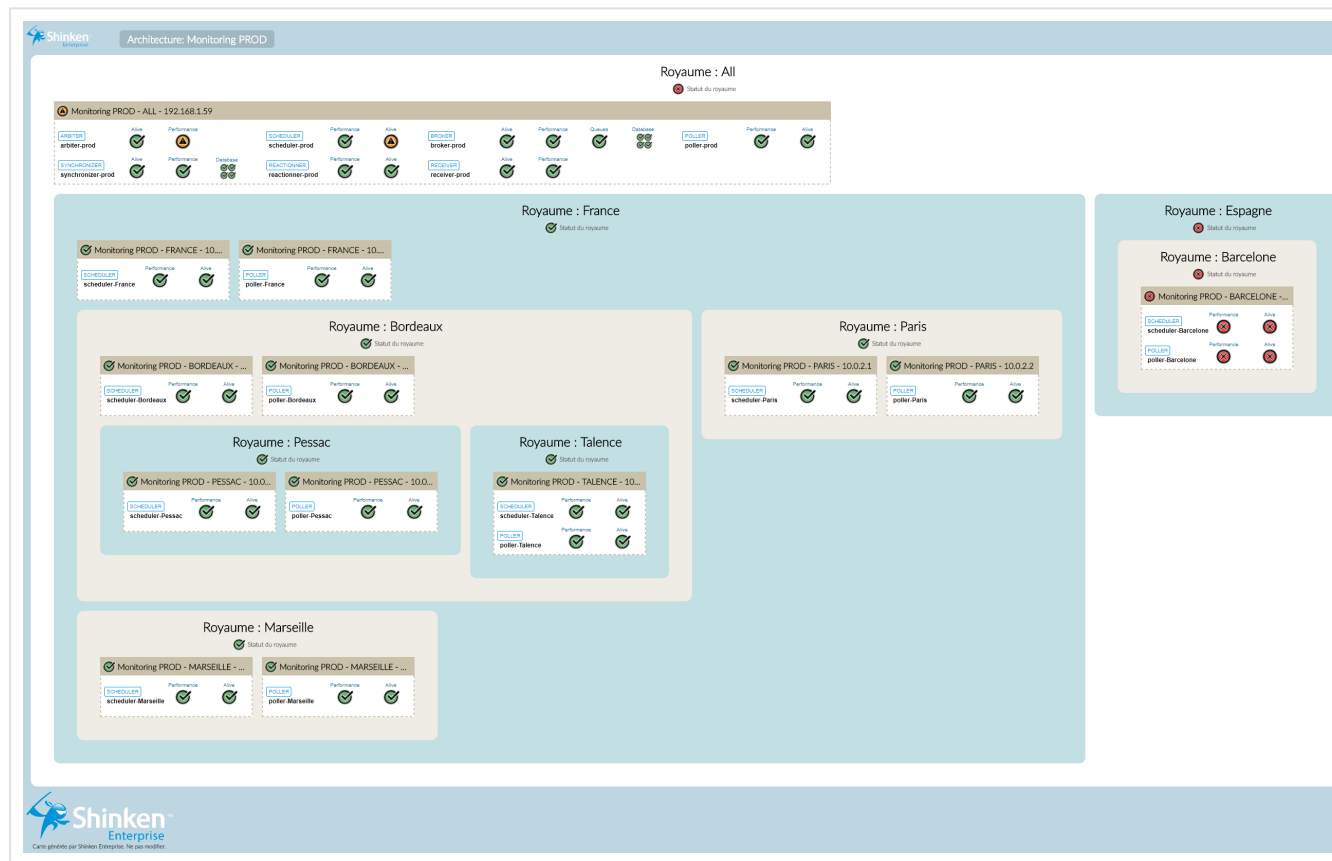
Vue détaillée de l'architecture globale

Cette vue est utilisée pour afficher de manière détaillée le contenu de l'ensemble des royaumes de l'installation Shinken.

Chaque royaume est délimité graphiquement, permettant de voir quels sous-royaumes il possède ainsi que ses démons.

L'affichage d'un royaume contient plusieurs informations:

- Les démons qui appartiennent à ce royaume
- Les sous-royaumes



Dans un royaume, l'affichage des démons est réparti selon les machines sur lesquelles ils sont installés. Les démons seront donc affichés dans un bloc contenant le nom de la machine sur laquelle ils sont installés. Le nom de la machine est généré automatiquement et est construit comme suivant:

 **<nom_architecture>** - **<nom_royaume>** - **<adresse_mach**
ne>

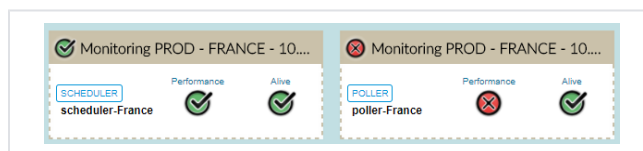
Ces valeurs sont prises de la manière suivante:

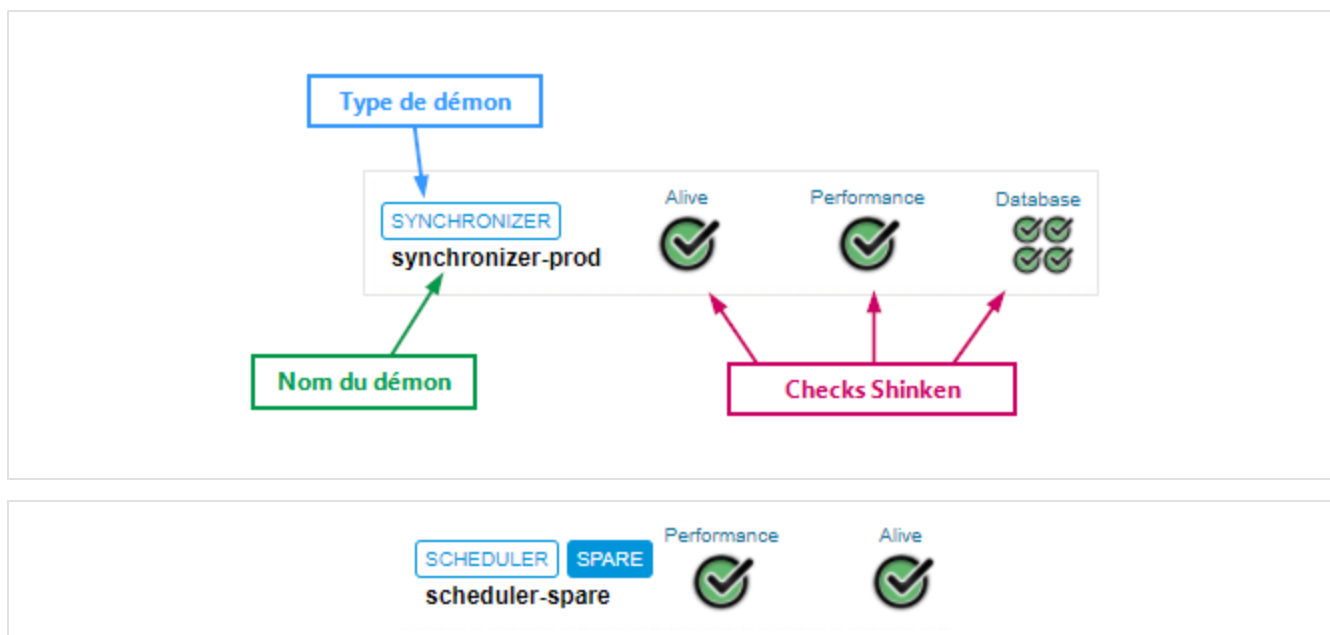
- **Nom de l'architecture:** Dans le fichier de configuration du module architecture-export ([Configuration de la Visualisation de l'architecture](#))
- **Nom du royaume:** Royaume du démon
- **Adresse de la machine:** Adresse telle que définie dans le fichier de configuration du démon (peut être une adresse IP ou un nom DNS)

Cela permet d'identifier rapidement les machines entrant en jeu dans le royaume concerné. Lorsqu'une machine constituant le royaume est en erreur, on peut le voir rapidement dans la vue.

Les démons sont donc affichés dans un bloc d'hôte. Chaque démon est affiché avec les informations suivantes:

- **Le type:** Arbitrer, Broker, Poller, Scheduler, ...
- **Le nom du démon,** tel que défini dans le fichier de configuration du démon
- **Les checks associés:** ils seront affichés sous forme d'icônes. Le statut affiché est celui du check du pack Shinken correspondant. Un clic sur l'icône de statut d'un check renvoie vers l'interface de Visualisation, ou il sera possible d'avoir plus de détails sur le check (historique, métriques, Prise en compte, etc...)
- **La caractéristique "Spare"** le cas échéant. Cette propriété est affichée lorsque le paramètre "spare" est positionné à 1 dans le fichier de configuration du démon en question.





Vue détaillée par royaume

La vue globale détaillée contient l'ensemble de royaumes de l'architecture Shinken. Dans le cas d'une architecture complexe, ou simplement pour se concentrer sur un royaume particulier, il peut être nécessaire de pouvoir visualiser la visualisation d'un royaume particulier.

Des vues spécifiques sont donc générées pour chaque royaume. Elles ont la même logique de présentation que la vue détaillée globale, à la différence qu'elles sont restreintes à un royaume particulier.

Pour y accéder, il y a 2 choix:

- Dans une vue détaillée, un clic sur le statut d'un royaume envoie vers la vue détaillée de ce royaume
- Dans la barre latérale, la hiérarchie des royaumes permet d'accéder rapidement au royaume voulu. On remarque au passage que l'arbre des royaumes et la vue détaillée globale sont aussi présents tout en haut de cette hiérarchie.

Génération des hôtes dans l'interface de Configuration

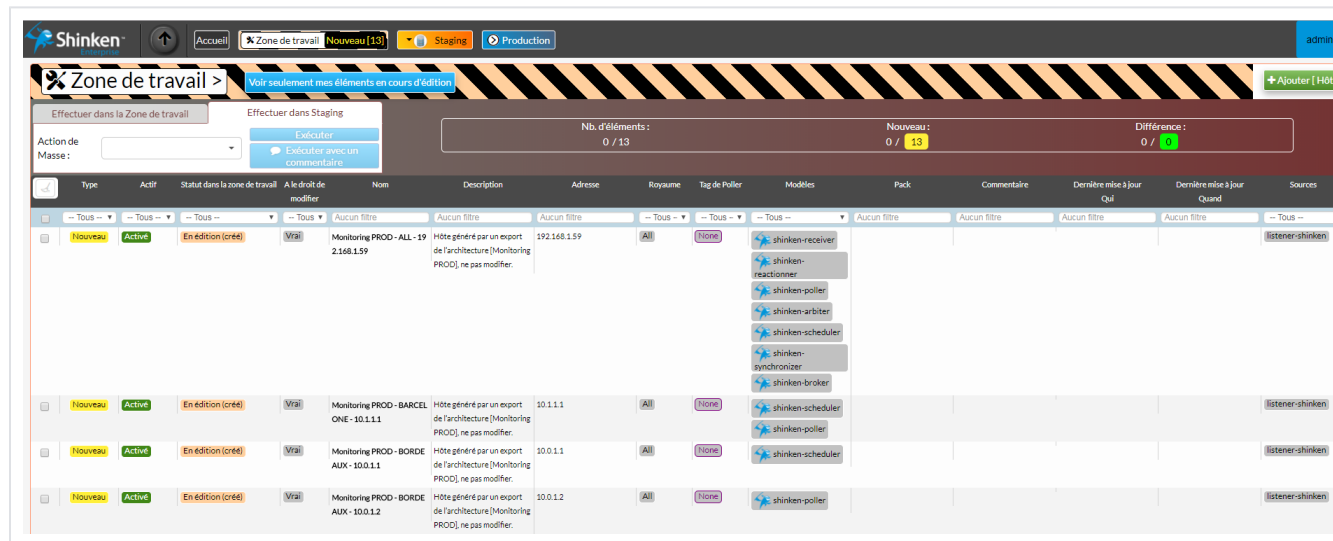
Démons Shinken

Nous avons vu que lorsque l'architecture de l'installation Shinken est changée, des vues représentatives de cette architecture sont générées. Dans ces vues, les statuts affichés proviennent de Shinken. Pour pouvoir déterminer les statuts des checks Shinken des démons, des hôtes doivent être créés dans Shinken.

Pour chaque machine affichée dans la vue détaillée, une création hôte est proposée dans l'interface de Configuration. Ces hôtes sont préconfigurés avec les modèles d'hôtes Shinken correspondants aux démons qu'ils contiennent. Il ne reste plus qu'à les importer depuis l'interface de Configuration.

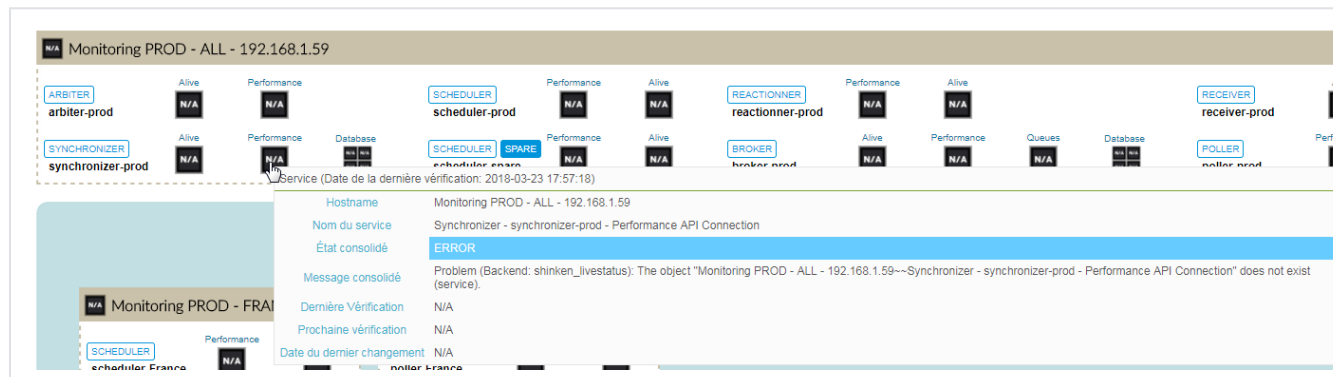
Leur nom est choisi comme décrit précédemment:

 **<nom_architecture>** - **<nom_royaume>** - **<ip_machine>**



Type	Actif	Statut dans la zone de travail	A le droit de modifier	Nom	Description	Adresse	Royaume	Tag de Poller	Modèles	Pack	Commentaire	Dernière mise à jour	Dernière mise à jour	Sources
Nouveau	Actif	En édition (créé)	Vital	Monitoring PROD - ALL - 192.168.1.59	Hôte généré par un export de l'architecture (Monitoring PROD), ne pas modifier.	192.168.1.59	All	None	shinken-receiver shinken-reactionner shinken-poller shinken-arbiter shinken-scheduler shinken-synchronizer shinken-broker					listener-shinken
Nouveau	Actif	En édition (créé)	Vital	Monitoring PROD - BARCELONE - 10.1.1.1	Hôte généré par un export de l'architecture (Monitoring PROD), ne pas modifier.	10.1.1.1	All	None	shinken-scheduler shinken-poller					listener-shinken
Nouveau	Actif	En édition (créé)	Vital	Monitoring PROD - BORDEAUX - 10.0.1.1	Hôte généré par un export de l'architecture (Monitoring PROD), ne pas modifier.	10.0.1.1	All	None	shinken-scheduler					listener-shinken

Si les hôtes générés ne sont pas importés dans l'interface de Configuration, les vues afficheront une icône indiquant que les hôtes et les checks ne sont pas trouvés dans Shinken.



Hostname	Nom du service	État consolidé	Message consolidé	Dernière Vérification	Prochaine vérification	Date du dernier changement
Monitoring PROD - ALL - 192.168.1.59	Synchronizer - synchronizer-prod - Performance API Connection	ERROR	Problem (Backend: shinken_livestatus): The object "Monitoring PROD - ALL - 192.168.1.59--Synchronizer - synchronizer-prod - Performance API Connection" does not exist (service).	N/A	N/A	N/A

Lorsque les hôtes sont déjà existants dans Shinken, une modification de l'architecture est accompagnée d'une modification des hôtes dans l'interface de Configuration. Lorsque des modèles d'hôtes doivent être enlevés/ajoutés, des différences sont proposées dans l'interface de Configuration.

☐	Différence	Activé	Validé	Vrai	Monitoring PROD - ALL - 19 2.168.1.59	Hôte généré par un export de l'architecture [Monitoring]	192.168.1.59	All	None	shinken-receiver
Clé d'import	Propriété	Valeur validée	Valeur proposée	De						shinken-reactionner
use	Modèles d'hôte hérités	shinken-receiver, shinken-reactionner, shinken-poller,shinken- scheduler,shinken- synchronizer	shinken-receiver, shinken-reactionner, shinken-poller,shinken- arbitrer,shinken- scheduler,shinken- synchronizer,shinken- broker	• [FORCE] listener- shinken (shinken- receiver,shinken- reactionner,shinken- poller,shinken- arbitrer,shinken- scheduler,shinken- synchronizer,shinken- broker)						shinken-poller
._BROKER_LIST	._BROKER_LIST [data]	Valeur héritée ou valeur par défaut	broker-prod\$(7772)\$	• listener-shinken						shinken-schedule
._ARBITER_LIST	._ARBITER_LIST [data]	Valeur héritée ou valeur par défaut	arbitrer-prod\$(7770)\$	• listener-shinken						shinken-synchronizer



Hôtes générés et renommage

Puisque dans les vues détaillées et l'arbre des royaumes, les hôtes et checks sont reconnus grâce à leur nom, un renommage aura pour effet de les rendre introuvables par les vues de l'architecture. Les hôtes générés ne devront donc pas être renommés.

Architecture de métrologie (Graphite)

Une architecture Shinken utilise différents services pour la gestion de ses données, et en particulier Graphite pour la gestion des métriques.

Dans une architecture Shinken Entreprise distribuée, le service Graphite utilisé peut lui aussi faire l'objet d'une installation distribuée pour permettre d'augmenter les performances, la disponibilité ainsi que la résistance aux pannes.

Pour aider à la supervision de votre architecture Shinken Entreprise, les différents hôtes et modèles nécessaires pour la supervision des démons Graphite sont également créés et envoyés à l'interface de Configuration lors de la génération des hôtes.

Chaque démon carbon-cache et carbon-relay utilisé par Shinken subit le traitement suivant:

- Si la machine sur laquelle est hébergé le démon est une machine Shinken avec un Broker, les modèles "shinken-broker-module-metrology-writer" (si un module Graphite-Perdata est présent) et "shinken-broker-module-visualisation-ui" (si un module WebUI est présent) sont ajoutés
- Sinon, un hôte avec le modèle "shinken-graphite" est créé avec la convention de nommage suivante:



<nom_architecture> - <ip_machine>

Plus d'informations sur la haute disponibilité de la base de métrologie ainsi que sa supervision sont disponibles sur les pages de documentation dédiées:

- Haute disponibilité de la métrologie: [Haute disponibilité de la base de métrologie \(Graphite\)](#)
- Supervision des démons Graphite:
 - [Modèle shinken-broker-module-metrology-writer](#) (contient aussi la documentation sur le modèle "shinken-graphite")
 - [Modèle shinken-broker-module-metrology-writer](#)
 - [Modèle shinken-broker-module-visualisation-ui](#)
 - [Modèle shinken-broker-module-visualisation-ui](#)

Fonctionnement et rôle du module architecture-export

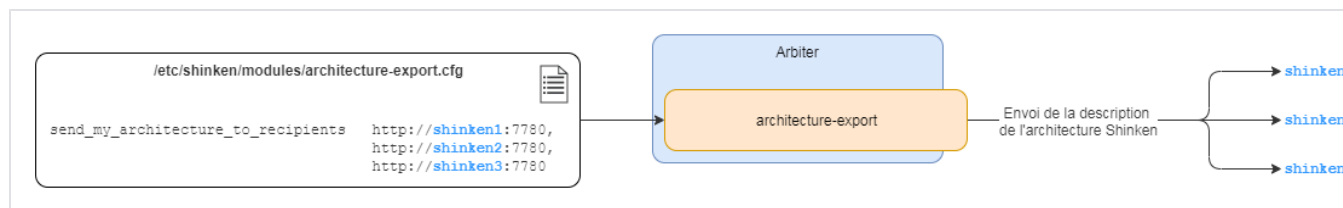
Le travail de génération des cartes NagVis qui représentent une architecture et de génération des hôtes correspondants dans l'interface Shinken est effectué par le module architecture-export présent sur l'Arbitrer.

Ce module a plusieurs rôles qui sont décrits dans la suite de ce document.

Envoi d'une description d'architecture

Le premier rôle du module architecture-export est d'envoyer la description de l'architecture Shinken à un autre module. Cette description d'architecture contient l'organisation des démons et royaumes, ainsi que les modules utilisés par chaque démon.

Dans la configuration du module (`/etc/shinken/modules/architecture-export.cfg`), on spécifie la liste des destinataires auxquels il faut envoyer la description de l'architecture. Au démarrage du démon Arbiter, le module `architecture-export` collecte les informations sur l'architecture et envoie ces informations à chaque destinataire.



L'envoi de la description de l'architecture aux destinataires choisis est déclenché au démarrage du module `architecture-export`, c'est-à-dire au démarrage du démon Arbiter.

Il est également possible de déclencher cet envoi manuellement, sans avoir à redémarrer le démon Arbiter, en envoyant une requête HTTP POST à l'URL suivante:

```
adresse_arbiter:7780/v1/architecture/send
```

Par exemple avec cURL:

```
curl -v -X POST adresse_arbiter:7780/v1/architecture/send
```

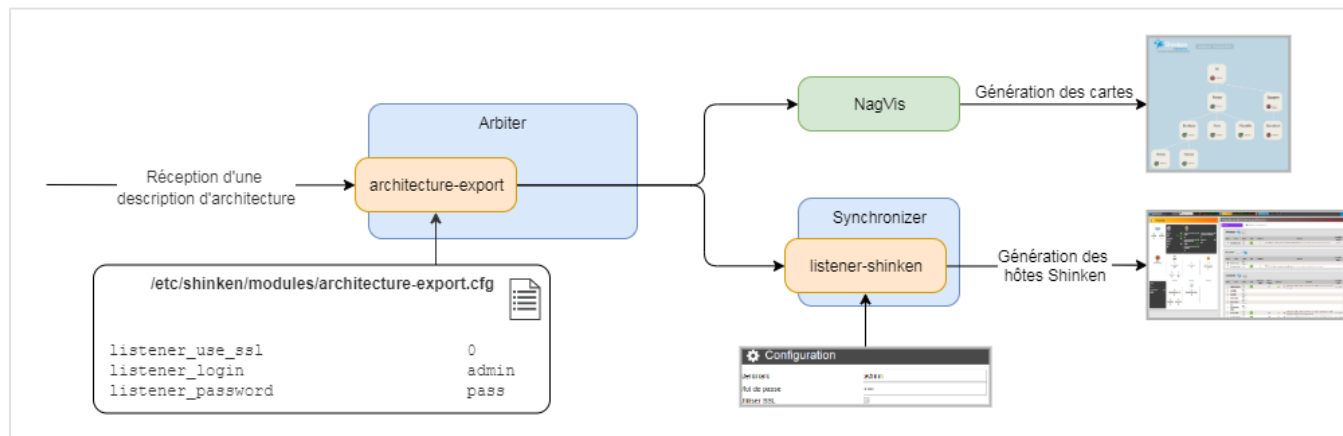
Réception d'une architecture

Le deuxième rôle du module `architecture-export` est de recevoir les descriptions d'architecture Shinken et d'y utiliser les informations trouvées pour générer:

- Des cartes NagVis correspondant à l'architecture reçue
- Des hôtes dans l'interface de Configuration de Shinken permettant de superviser l'architecture reçue.

Quand un module `architecture-export` reçoit une description d'architecture, il l'analyse et construit en fonction des démons, de leurs modules et des royaumes de l'architecture des cartes NagVis qui représentent cette répartition de démons et royaumes.

Pour permettre de superviser les démons de l'architecture reçue par le module, des hôtes Shinken sont générés dans l'interface de Configuration via l'écouter `listener-shinken`. Pour l'authentification auprès du `listener-shinken`, le module `architecture-export` utilise les paramètres correspondants dans son fichier de configuration (`/etc/shinken/modules/architecture-export.cfg`). Il faut alors que les identifiants présents dans le fichier de configuration du module `architecture-export` correspondent à ceux définis dans l'écouteur `listener-shinken` dans l'interface de Configuration.



Cas d'utilisations

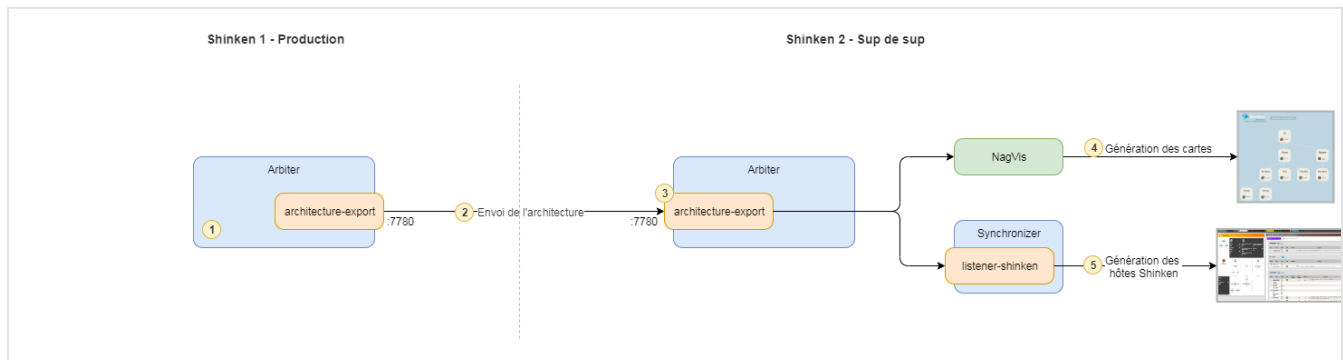
Envoi d'une architecture sur une Sup de Sup

Le principe de ce module est d'associer ces 2 rôles pour permettre sur une machine Shinken de générer des hôtes et cartes NagVis qui représentent l'architecture d'une autre installation Shinken.

Dans notre exemple, on possède 2 infrastructures Shinken indépendantes:

- La première qui représente notre production Shinken. C'est la plateforme Shinken qui effectue la supervision de notre réseau et équipement.

- La deuxième est une plateforme Shinken de "Sup de Sup", c'est à dire qui permet de superviser notre Shinken de Production et nous alerter en cas de dysfonctionnement.



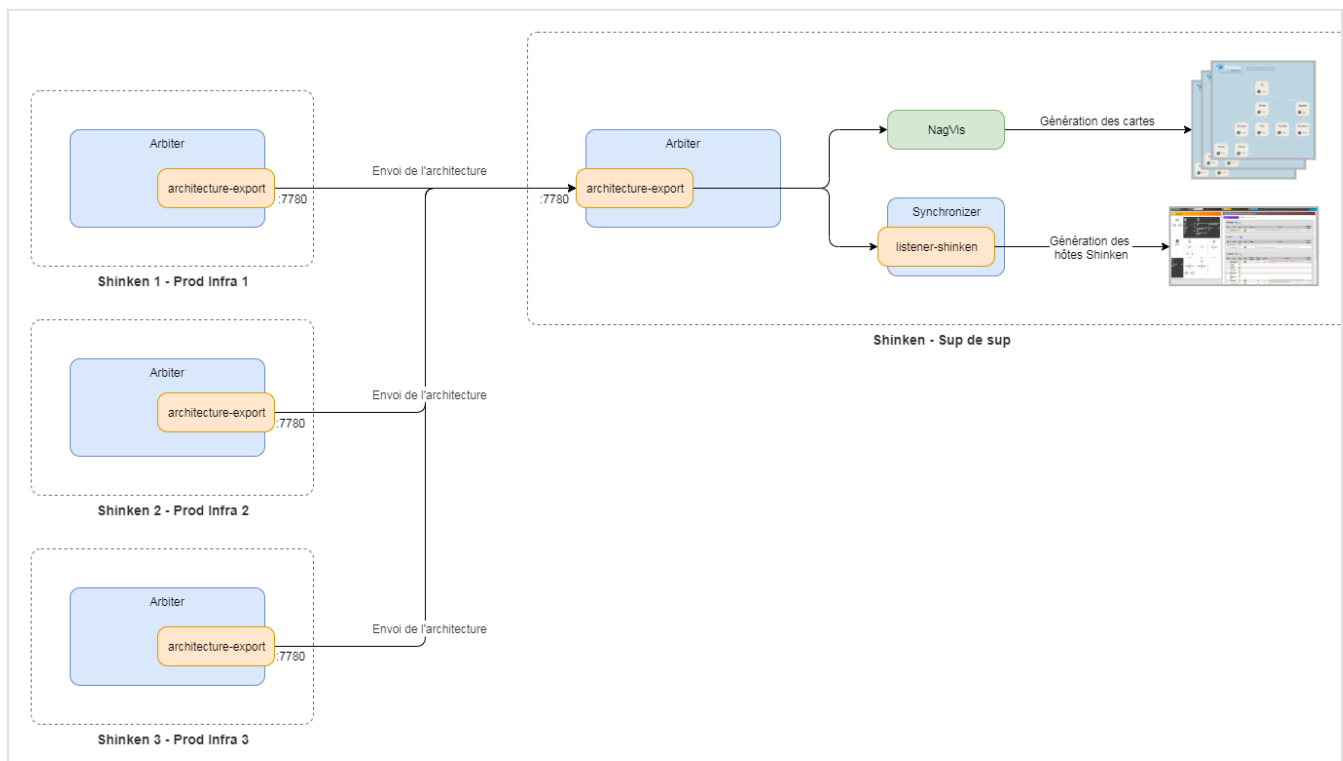
L'utilisation de l'addon nagvis-shinken-architecture se fait en plusieurs étapes présentées sur le schéma ci-dessus:

1. **Sur Shinken Production:** On configure le module *architecture-export* pour envoyer la description de l'architecture de Shinken Production sur Shinken Sup de sup.
2. **Sur Shinken Production:** Au démarrage de l'Arbiter, le module *architecture-export* récupère depuis l'Arbiter la configuration des démons, leurs modules et leurs répartitions dans les différents royaumes. Il envoie cette description d'architecture au module *architecture-export* de Shinken Sup de sup.
3. **Sur Shinken Sup de Sup:** Le module *architecture-export* écoute sur son port 7780 en attente d'une réception d'une description d'architecture. À la réception d'une description d'architecture de la part de Shinken Production, le module *architecture-export* commence l'analyse de la description d'architecture reçue.
4. **Sur Shinken Sup de Sup:** Avec les informations de l'architecture de Shinken Production en main, des cartes NagVis sont générées permettant de visualiser l'organisation des royaumes et démon.
5. **Sur Shinken Sup de Sup:** Toujours en fonction des informations trouvées dans la description d'architecture de Shinken Production, un hôte est créé automatiquement dans l'interface de Configuration pour chaque machine qui héberge un démon Shinken. Les modèles d'hôtes correspondants aux démons Shinken correspondants sont accrochés sur ces hôtes.

Envoi de plusieurs architectures sur une Sup de Sup

De la même manière que vue précédemment, il est possible sur notre Shinken Sup de Sup de générer les cartes et hôtes pour plusieurs architectures différentes.

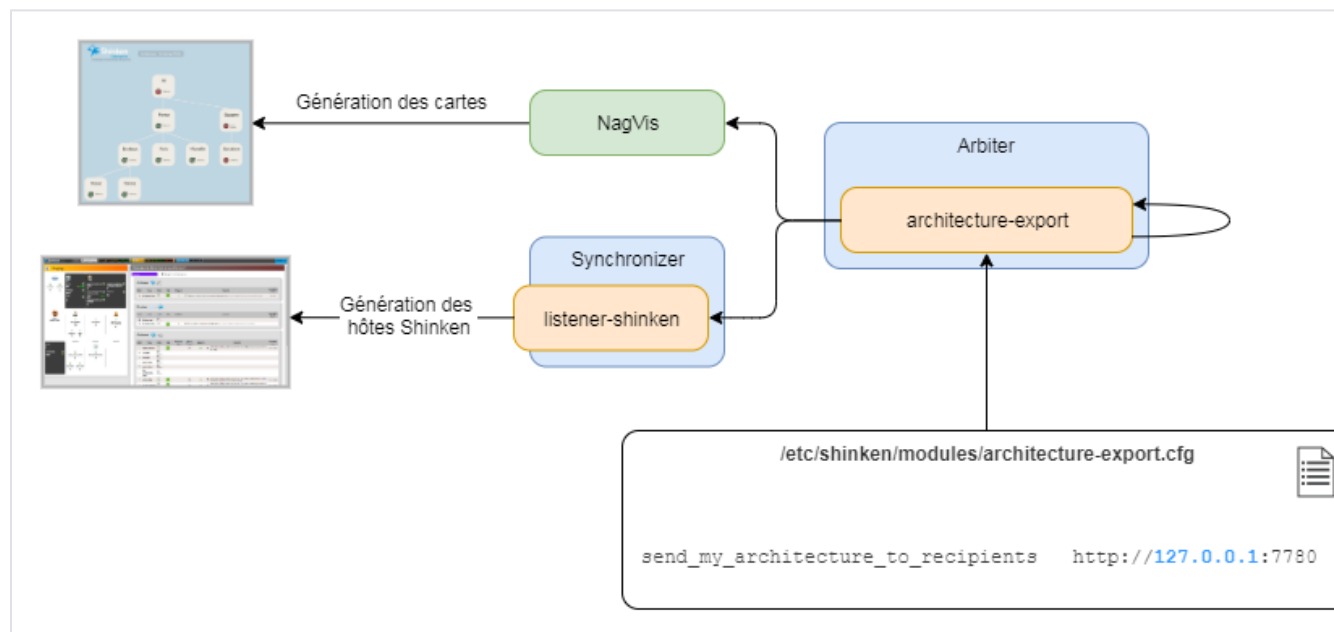
En suivant le même de configuration, le principe de fonctionnement est décrit sur le schéma suivant:



Envoi d'une architecture sur soi-même (comportement par défaut)

Dans une installation Shinken par défaut, le module architecture-export rassemble les rôles d'envoi et de réception d'une description d'architecture. Le module architecture-export s'envoie lui-même la description de l'architecture Shinken courante, ce qui permet de visualiser dans des cartes NagVis l'architecture de notre installation.

Schématiquement, la configuration du module se présente comme ceci sur une installation par défaut:



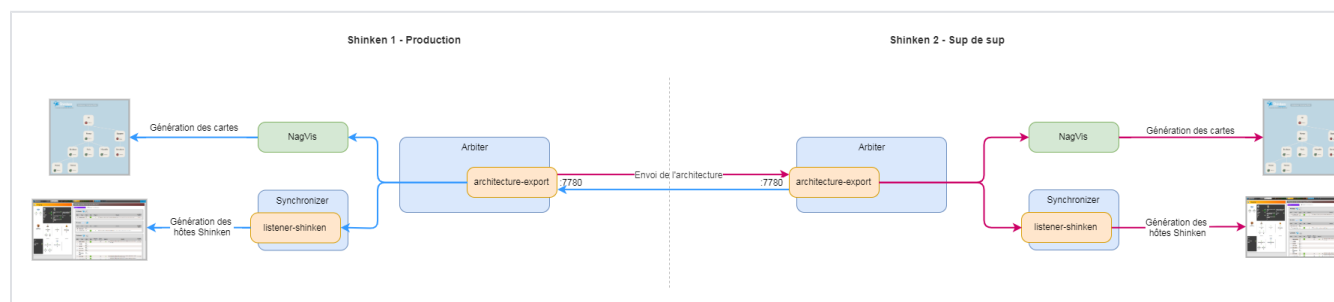
Envois mutuels

On a vu dans les exemples précédents qu'il est possible d'envoyer la description d'une architecture *Shinken Production* sur une *Shinken Sup de Sup*, ce qui permet à *Shinken Sup de sup* de superviser la plateforme Shinken de production et d'alerter en cas d'erreur.

Par contre, la plateforme *Shinken Sup de Sup* n'est pas supervisée, ce qui peut causer des problèmes si cette plateforme entre en erreur. On peut tirer avantage des modules architecture-export de ces 2 plateformes pour envoyer l'architecture de *Shinken Sup de Sup* sur *Shinken Production*, et donc permettre à *Shinken Production* de superviser notre plateforme de Sup de sup.

La procédure pour mettre en place cette configuration est décrite de manière détaillée dans la page de documentation suivante: [Exemple pratique - Mise en place automatisée d'une plateforme de supervision secondaire](#)

Le fonctionnement des modules architecture-export avec ce type de configuration se visualise de la façon suivante:



Envoi de la description de l'architecture

A chaque redémarrage

L'envoi de la description de l'architecture aux destinataires choisis (paramètre : `send_my_architecture_to_recipients`) est déclenché au démarrage du module architecture-export, c'est-à-dire au démarrage du démon Arbiter.

Manuellement

Il est également possible de déclencher cet envoi manuellement, sans avoir à redémarrer le démon Arbiter, en envoyant une requête HTTP POST à l'URL suivante:

```
adresse_arbiter:7780/v1/architecture/send
```


Par exemple, avec curl:

```
curl -v -X POST adresse_arbiter:7780/v1/architecture/send
```