

Paramétrage par défaut de NagVis

Sommaire

[Paramètres généraux](#)
[Choix du backend](#)

Paramètres généraux

Pour l'addon "**nagvis-shinken-architecture**", une installation de NagVis est gérée par les scripts d'installation et de mise à jour de Shinken Entreprise.

Celle-ci est disponible à l'adresse suivante:



Adresse de NagVis pour la visualisation de l'architecture

`http://adresse_arbiter/shinken-core-map`



Dossier d'installation

Par défaut l'installation de NagVis se fait dans le dossier `/var/lib/shinken-nagvis`


Si ce dossier se trouve être un point de montage, NagVis ne pourra procéder à son installation, le dossier d'installation est alors modifié en `/var/lib/shinken-nagvis/nagvis`

Au final, le lien symbolique `/etc/shinken/external/nagvis/` permet d'arriver dans le dossier d'installation de NagVis, quel qu'il soit.

Cette version vient préconfigurée pour pouvoir fonctionner avec Shinken de manière transparente, sans configuration nécessaire.

Les différentes options de configuration utilisées sont les suivantes:

Option	Type	Valeur par défaut	Commentaire
authmodule	Texte	CoreAuthModShinken	Module d'authentification des utilisateurs en liaison avec Shinken. (Voir la page Gestion de l'authentification).
autorisationmodule	Texte	CoreAutorisationModShinken	Module de gestion des autorisations des utilisateurs. (Voir la page Gestion de l'authentification).
logonmodule	Texte	LogonShinkenMixed	Module de connexion. (Voir la page Gestion de l'authentification).
shinken_features	Booléen	1	Donne accès à certaines fonctionnalités de Shinken dans NagVis, comme l'impact métier des objets.
shinken_auth_restrict_to_shinken_admin	Booléen	1	Restreint la connexion à NagVis aux administrateurs shinken. (Voir la page Gestion de l'authentification).
shinken_auth_protocol	Texte	http	Cette valeur est automatiquement renseignée par le module architecture-export de l'Arbiter. Précise le protocole à utiliser pour contacter la WebUI. Les valeurs possibles sont les suivantes : <ul style="list-style-type: none">▪ http : pour une connexion non chiffrée▪ https : pour une connexion chiffrée
shinken_auth_port	Entier	7767	Cette valeur est automatiquement renseignée par le module architecture-export de l'Arbiter. Précise le port réseau à utiliser pour contacter la WebUI (Voir la page Module architecture-export).

shinken_auth_address	Texte	localhost	Cette valeur est automatiquement renseignée par le module architecture-export de l'Arbiter. Précise le nom d'hôte à utiliser pour se connecter à la WebUI (Voir la page Module architecture-export).
shinken_auth_remote_user_variable	Texte	(vide)	Cette valeur est automatiquement renseignée par le module architecture-export de l'Arbiter. Précise le nom de la variable à rechercher dans les entêtes HTTP pour activer l'identification automatique lorsqu'on arrive de la WebUI (Voir la page Module architecture-export).
shinken_authentication_ssl_verify_certificate	Booléen	0	Activer la vérification du certificat reçu de la WebUI quand elle est configurée en https <ul style="list-style-type: none"> ▪ 0 : non ▪ 1 : oui
shinken_authentication_ssl_verify_certificate_name	Booléen	1	Quand la vérification du certificat de la WebUI est activé, vérifier si le nom d'hôte de la WebUI correspond au nom enregistré dans le certificat <ul style="list-style-type: none"> ▪ 0 : non • 1 : oui <div style="border: 1px solid red; padding: 5px; margin-top: 10px;"> <p> Sous CentOS 7 (<i>ayant une version de PHP < 7</i>), la vérification du nom du certificat ne fonctionne pas quand ce certificat est dans la chaîne de confiance (<i>paramètre shinken_authentication_ssl_certificate_authority_file</i>).</p> <p>Vous pouvez mettre votre version de PHP à jour en version 7.2 si vous avez besoin de cette fonctionnalité</p> </div>
shinken_authentication_ssl_allow_self_signed_certificate	Booléen	1	Quand la vérification du certificat de la WebUI est activé, autoriser les certificats auto-signés <ul style="list-style-type: none"> ▪ 0 : non ▪ 1 : oui
shinken_authentication_ssl_certificate_authority_file	Texte	(vide)	Définit le certificat d'autorité à utiliser <ul style="list-style-type: none"> • la valeur système par défaut est "/etc/ssl/certs/ca-bundle.trust.crt" • pour autoriser un certificat auto signé généré pour Shinken, on peut utiliser "/etc/shinken/certs/ca.pem"
backend	Texte	shinken_livestatus	Backend utilisé pour la connexion à Shinken
eventsound	Booléen	0	Pas d'alerte sonore lors d'un changement de statut
urltarget	Texte	_blank	Les liens vers les autres cartes NagVis et le détail des éléments dans Shinken sont ouverts dans un nouvel onglet
hosturl	Texte	http://ip_broker:7767/detail-by-name/[host_name]	Adresse utilisée pour le détail des hôtes. La valeur de ce paramètre est remplacé à chaque démarrage du module " architecture-export " par l'URL présente dans le paramètre " architecture_export_broker_connection_broker_webui_target " du module "Module architecture-export" (Voir la page Module architecture-export).
servicegroupurl	Texte	http://ip_broker:7767/detail-by-name/[host_name]/checks/[service_description]	Adresse utilisée pour le détail des checks. La valeur de ce paramètre est remplacé à chaque démarrage du module " architecture-export " par l'URL présente dans le paramètre " architecture_export_broker_connection_broker_webui_target " du module "Module architecture-export" (Voir la page Module architecture-export).

Ces options peuvent être par la suite modifiées de 2 manières différentes:

- Depuis l'interface de NagVis
- Par fichier de configuration (/etc/shinken/external/nagvis/etc/nagvis.ini.php)

Choix du backend

L'intérêt de NagVis est qu'il est capable de se connecter à une plateforme de supervision pour récupérer le statut des hôtes et checks.

Cette connexion est configurée via ce qui est appelé "backend" dans NagVis.

Lors de l'installation de l'addon "nagvis-shinken-architecture" effectué pendant une installation/mise à jour de Shinken. Un backend est créé automatiquement. Il utilise Livestatus pour se connecter au Broker présent sur la même machine que l'Arbiter (127.0.0.1), sur le port de Livestatus par défaut (50000).

Si un autre Broker doit être utilisé, le backend par défaut peut être modifié depuis l'interface de configuration.

The screenshot shows the NagVis configuration interface. The main window displays the 'Architecture: Monitoring PROD' with a status bar indicating 'Monitoring PROD - ALL - 192.168.1.59'. Below this, several components are listed with their status: 'ARBITER arbiter-prod' (Alive, Performance), 'SYNCHRONIZER synchronizer-prod' (Alive, Performance), 'SCHEDULER scheduler-prod' (Performance), and 'SCHEDULER SPARE scheduler-spare' (Performance). A 'Gérer les Backends' modal window is open on the right, showing options to manage backends. The 'Default Backend' is set to 'shinken_livestatus'. The modal also includes sections for 'Add Backend', 'Edit Backend', and 'Delete Backend', each with a 'Backend ID' field and a 'Save' button.

Il est également possible de modifier le backend utilisé depuis le fichier de configuration de NagVis si besoin:

/etc/shinken/external/nagvis/etc/nagvis.ini.php

```
[backend_shinken_livestatus]
backendtype="mklivestatus"
socket="tcp:127.0.0.1:50000"
```