

NagVis (Addon)

Sommaire

[Introduction](#)
[Activation automatique](#)
[Manipulation](#)
[Configuration par défaut](#)
 [Inventaire des options](#)
 [Récupération des statuts dans Shinken](#)
[Choix du backend](#)

Introduction

NagVis est un logiciel Open Source qui permet de visualiser sous forme de cartes les données tirées de différents outils de supervision: Shinken, Nagios, ... (voir la page <http://nagvis.org/about>)

NagVis est régulièrement installé en parallèle de Shinken afin de visualiser le statut des hôtes sur des représentations graphiques avancées et personnalisées.

- C'est pourquoi à partir de la version V02.05.00, une instance de NagVis est installée automatiquement lors d'une installation ou d'une mise à jour de Shinken Entreprise.
- Cette installation permet notamment d'éviter à l'utilisateur de devoir configurer NagVis pour le lier avec Shinken, puisque l'instance de NagVis installée est aussi pré configurée lors de son installation.

À partir de la version V02.05.00, l'architecture d'une installation Shinken peut être visualisée de manière dynamique dans un outil externe : NagVis

- ([v1.9.5](#)) entre les versions V02.05.00 et V02.07.06 de Shinken Entreprise.
- ([v1.9.33](#)) à partir de la V02.08.01.

Les sections suivantes décrivent comment manipuler cette installation NagVis, ainsi que la configuration par défaut disponible.

Une fois configuré, vous aurez accès aux cartes via l'url **http://ip_machine/shinken-map**

Activation automatique

Cette installation de NagVis est présentée dans Shinken comme un addon.

- Cet addon est automatiquement activé lors d'une installation avec le démon Broker activé.
- Il sera donc activé lors de l'installation dans les cas suivants :

```
./install.sh  
./install.sh --brokernode
```

Mais désactivé par exemple sur une installation d'un Poller uniquement:

```
./install.sh --pollernode
```

Lors d'une mise à jour depuis une version V02.04.XX, l'addon n'est pas activé pour rester dans le même périmètre de fonctionnalité.

Lors d'une mise à jour depuis une version V02.05.XX, ou ultérieur, l'addon conserve son état d'activation précédent.

Manipulation



Accès à NagVis

Lorsque l'addon est activé, NagVis est disponible sur la machine correspondante à l'adresse suivante:

http://ip_machine/shinken-map

L'instance de NagVis installée par l'installateur Shinken Entreprise est présentée sous la forme d'un addon "**nagvis**".

Cet addon peut être activé et désactivé grâce aux commandes d'Activation/désactivation (voir la page [Activation - désactivation des outils supplémentaires \(addons \)](#)) des addons suivantes :

```
shinken-addons-enable nagvis
```

```
shinken-addons-disable nagvis
```

L'activation de l'addon "**nagvis**" effectue les opérations suivantes:

- Activation du module Livestatus dans le Broker, pour permettre à NagVis de communiquer avec Shinken (*seulement lorsqu'on exécute la commande d'activation sur la machine de l'Arbiter*).
- Activation de NagVis au niveau d'Apache, pour permettre à l'interface Web de NagVis d'être accessible.

La désactivation de l'addon effectue les opérations suivantes:

- Désactivation de NagVis au niveau d'Apache, pour que l'interface Web ne soit plus accessible.
- Le module Livestatus n'est pas désactivé sur le Broker, car il est possible que d'autres outils aient besoin d'utiliser Livestatus.



- Puisque les commandes d'activation/désactivation effectuent des modifications de la configuration des démons, il faut les exécuter sur le serveur central (*celui contenant l'Arbiter*) pour que les modifications de la configuration soient prises en compte.
- Aussi, puisque ces commandes effectuent des modifications de configuration locales à la machine (*changement de paramètres Apache*), il faudra aussi les exécuter localement sur la machine ou l'on veut activer l'addon.

Configuration par défaut

Un des intérêts principaux de NagVis est de pouvoir être relié à un logiciel de supervision (*en l'occurrence Shinken*) pour récupérer le statut des hôtes et des checks.

L'addon "nagvis" vient donc avec une version de NagVis préconfigurée pour éviter d'avoir à effectuer ces réglages manuellement.



Les réglages suivants ne sont que des paramètres définis à l'installation. L'administrateur a bien sûr par la suite la liberté de les modifier et de s'approprier cette installation NagVis comme il le souhaite.

Inventaire des options

Les différentes options de configuration modifiées 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).
authorisationmodule	Texte	CoreAuthorisationModShinken	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 : • 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 • 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 • 0 : non • 1 : oui  Sous CentOS 7 (ayant une version de PHP < 7), la vérification du nom du certificat ne fonctionne pas quand ce certificat est dans la chaîne de confiance (paramètre <code>shinken_authentication_ssl_certificate_authority_file</code>). Vous pouvez mettre votre version de PHP à jour en version 7.2 si vous avez besoin de cette fonctionnalité
shinken_authentication_ssl_allow_self_signed_certificate	Booléen	1	Quand la vérification du certificat de la WebUI est activé, autoriser les certificats auto-signés • 0 : non • 1 : oui
shinken_authentication_ssl_certificate_authority_file	Texte	vide	Définit le certificat d'autorité à utiliser • la valeur système par défaut est <code>"/etc/ssl/certs/ca-bundle.trust.crt"</code> • pour autoriser un certificat auto signé généré pour Shinken, on peut utiliser <code>"/etc/shinken/certs/ca.pem"</code>
backend	Texte	shinken_livestatus	Backend utilisé pour la connexion à Shinken
eventsound	Booléen	0	Pas d'alerte sonore lors d'un changement de status
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-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).

Récupération des statuts dans Shinken

NagVis doit être relié à un "backend" pour pouvoir récupérer les statuts des hôtes et checks. Dans le cadre de Shinken, cette liaison avec NagVis s'effectue via le module Livestatus du Broker.

Ce module est automatiquement activé lorsque l'addon est activé avec la commande d'activation des addons (voir section précédente).

Par défaut, NagVis va récupérer les statuts des éléments de Shinken sur le Broker situé sur la machine sur laquelle il est installé (127.0.0.1), en utilisant le port de Livestatus par défaut (50000).

Si un Broker doit être utilisé, les paramètres du backend peuvent être configurés directement dans NagVis, par l'interface Graphique ou bien via le fichier de configuration de NagVis.

/opt/nagvis/etc/nagvis.ini.php

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

The screenshot shows the NagVis web interface with a sidebar menu on the left containing 'Configuration Générale', 'Gérer les Backends', 'Gérer les Fonds de Carte', 'Gérer les Cartes', and 'Gérer les Figures'. The 'Gérer les Backends' modal window is open, displaying four sections: 'Default Backend', 'Add Backend', 'Edit Backend', and 'Delete Backend'. The 'Default Backend' section shows 'shinken_livestatus' selected in a dropdown with a 'Save' button. The 'Add Backend' section has empty input fields for 'Backend ID' and 'Backend Type' (dropdown) with a 'Save' button. The 'Edit Backend' section shows 'shinken_livestatus' for 'Backend ID', 'tcp:127.0.0.1:50000' for 'socket', '10' for 'timeout', and empty fields for 'statushost', 'htmlcgi', 'custom_1', 'custom_2', and 'custom_3', with a 'Save' button. The 'Delete Backend' section has a dropdown for 'Backend ID' and a 'Supprimer' button.

Gérer les Backends

Default Backend

Default Backend: shinken_livestatus [Save]

Add Backend

Backend ID: []
Backend Type: Please choose [Save]

Edit Backend

Backend ID: shinken_livestatus
socket: tcp:127.0.0.1:50000
timeout: 10
statushost: []
htmlcgi: []
custom_1: []
custom_2: []
custom_3: [] [Save]

Delete Backend

Backend ID: Please choose [Supprimer]

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.

Le backend 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 web interface. The main menu includes 'NagVis', 'Ouvrir', 'Actions', and 'Options'. A dropdown menu under 'Options' shows 'Configuration Générale' and 'Gérer les Backends'. The main content area displays the 'Architecture: Shinken-model-dev-centos7.2' with a status overview for 'Shinken-model-dev-centos7.2 - ALL - 192.168.1.45'. The overview shows components like ARBITER (arbiter-master), SYNCHRONIZER (synchronizer-master), SCHEDULER (scheduler-master), and REACTIONNER (reactionner-master) with their respective status (Alive, Performance, Database). A modal window titled 'Gérer les Backends' is open, showing options to manage the default backend, add a new backend, edit an existing one, or delete one. The 'Default Backend' is currently set to 'shinken_livestatus'.

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

/opt/nagvis/etc/nagvis.ini.php

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