Modèle shinken-receiver

Sommaire

Introduction

Le module architecture-export permet

- d'envoyer une description de l'architecture d'installation Shinken
- de recevoir cette configuration (ou celle d'une autre installation) et d'en générer :
  - des cartes NagVis
  - des hôtes Shinken qui seront envoyés à leur tour au Synchronizer pour les superviser

Une description détaillée de ce module, son utilité et son utilisation sont présentes dans les pages de documentations associées : Visualiser l'architecture de son installation Shinken

Ci-dessous sont présentées de manière synthétique les différentes options de configuration de ce module, leur rôle ainsi que leurs valeurs par défaut.

Activation du module

Le module **architecture-export** est un module qui peut être activé seulement sur le démon sur Arbiter.

- L'activation du module s'effectue en ajoutant le nom de ce module dans le fichier de configuration de l'Arbiter.
- Pour ce faire, il faut ouvrir le fichier de configuration du Arbiter à l'emplacement **/etc/shinken/arbiters/** , et ajouter le nom de votre module de type "*architecture-export*".

Exemple : par défaut, nous livrons un module dont le nom est " **architecture-export** " :

```
define arbiter {
    [...]
    modules                 synchronizer-import, architecture-export
    [...]
}
```

Pour prendre en compte les changements de configuration, il faut redémarrer l'Arbiter :

```
service shinken-arbiter restart
```

Pour que l'architecture-export génère des cartes, il faut que l'addon nagvis-shinken-architecture soit activé (*voir Configuration de la Visualisation de l'architecture*)

Configuration

La configuration du module se trouve par défaut dans le fichier **/etc/shinken/modules/architecture-export.cfg**

- Vous trouverez aussi systématiquement un exemple dans **/etc/shinken-user-example/configuration/daemons/arbiters/modules /architecture-export/architecture-export.cfg**

Exemple de fichier de configuration

```
#===============================================================================
# Architecture-Export
#===============================================================================
# Daemons that can load this module:
#  - arbiter
# This module exports Shinken architecture to others architecture-export modules
# It also creates NagVis Maps for received architecture
#===============================================================================

define module {

    #  #
    #      MODULE IDENTITY      #
    #  #

    #  Module name [ Must be unique ]                                          [ MANDATORY ]
    #
    module_name                                     architecture-export

    #  Module type [ Do not edit ]                                             [ MANDATORY ]
    #
    module_type                                     architecture_export

    #  #
    #      NAGVIS FILE PATH      #
    #  #

    #  Path of NagVis installation
    #  Used to store configuration maps files and to update NagVis settings
    #
    #           Default : /etc/shinken/external/nagvis
    #
    path                                            /etc/shinken/external/nagvis

    #  #
    #      MODULE OPTIONS      #
    #  #

    #  Base of URL used to display links in the Visualization UI
    #
    #           Default : Use Arbiter URL
    #
    # map_base_url                                    http://example.com/

    #  Architecture description recipients
    #  When the architecture of this Shinken installation changes ( realms and daemons configuration ),
    #  and the arbiter is restarted, the architecture description will be sent to the following hosts.
    #
    #           Default : http://127.0.0.1:7780 ( locally )
    #
    send_my_architecture_to_recipients              http://127.0.0.1:7780

    #  #
    #      MODULE COMMUNICATION      #
    #  #

    #   Listening socket configuration   #

    #  This module opens a listening socket on which other Shinken installations
    #  will send their architecture description.
    #  Network interface used for the listening socket ( 0.0.0.0 = all interfaces )
    #
    #           Default : 0.0.0.0
    #
    host                                            0.0.0.0
```

```
#  Port to use for the listening socket
#
#           Default : 7780
#
port                                            7780

#  Protocol to use for the listening socket
#
#           ...     : Enable  => 1 ( Use HTTPS )
#           Default : Disable => 0 ( Use HTTP )
#
use_ssl                                         0

#  SSL Certificate to use for the listening socket ( if HTTPS )
#
#           Default : /etc/shinken/certs/server.cert
#
ssl_cert                                        /etc/shinken/certs/server.cert

#  SSL Key to use for the listening socket ( if HTTPS )
#
#           Default : /etc/shinken/certs/server.key
#
ssl_key                                         /etc/shinken/certs/server.key

#   Connection with the listener-shinken   #

#  Connection parameters for the module to communicate with the listener-shinken
#  ( used to create hosts for maps )
#  Protocol used by listener-shinken
#
#           ...     : Enable  => 1 ( Use HTTPS )
#           Default : Disable => 0 ( Use HTTP )
#
# listener_use_ssl                               0

#  Listener-shinken configured login
#
#           Default : Shinken
#
# listener_login                                 login

#  Listener-shinken configured password
#
#           Default : Default password generated for listener-shinken
#
# listener_password                              pass

#   Connection with the graphite host via ssh   #

#  Connection parameters for the module to communicate with the graphite host via ssh
#  ( used to get the graphite configuration )
#  These parameters will be the same for every architecture received by the module
#  so every graphite host should allow ssh connection with these parameters
#  SSH Port
#
#           Default : 22
#
# ssh_port                                       22

#  SSH user
#
#           Default : shinken
#
# ssh_user                                       shinken

#  SSH key file
#
#           Default : /var/lib/shinken/.ssh/id_rsa
#
```

```
    # ssh_key_file                                          /var/lib/shinken/.ssh/id_rsa

    #  SSH timeout
    #
    #           Default : 5 seconds
    #
    # ssh_timeout                                          5

    #  #
    #      MAPS CREATION PARAMETERS      #
    #  #

    #  When an architecture description is received by the module,
    #  it creates corresponding hosts and NagVis maps.
    #  Name with which this Shinken installation will be identified in the NagVis maps
    #  The following characters are forbidden in the architecture name: ~!$%^&*"'|<>?,()=/+
    #
    #           Default : Shinken
    #
    architecture_name                                    Shinken

    #
    #           ...     : Sort order for realms in the NagVis maps
    #           ...     : sort_by_name => Sort realms by alphabetic order
    #           Default : sort_by_size => Sort realms by size (on the map)
    #
    # map_realm_layout                                     sort_by_size

    #  #
    #      BROKER CONNECTION PARAMETERS      #
    #  #

    #  These parameters are used to allow NagVis to communicate with the Broker and modules you want
    #  Name of the Broker holding the modules you want NagVis to communicate with
    #
    #           Default : broker-master
    #
    # architecture_export__broker_connection__broker_name broker-master

    #  Name of the Livestatus module you want NagVis to communicate with to retrieve objects information
    #
    #           Default : Livestatus
    #
    # architecture_export__broker_connection__broker_livestatus Livestatus

    #  Type of the target WebUI you want to communicate with
    #  This allow redirection when clicking on object on the maps
    #
    #           Default : module => Use a WebUI module configuration to get it's address
    #           ...     : url    => Use a URL ( the WebUI address is behind a reverse proxy or use an DNS
    #                               address )
    #
    # architecture_export__broker_connection__broker_webui_communication_type module

    #  Targeted WebUI to communicate with
    #  If previous parameter is set to "module", this must be a WebUI name
    #  If previous parameter is set to "url", this must be a URL
    #
    #           Default : WebUI
    #
    # architecture_export__broker_connection__broker_webui_target WebUI

}
```

Détails des sections composant le fichier de configuration

Identification du module

Il est possible de définir l'identité du module de type **architecture-export** .

Nom	Type	Unité	Défaut	Commentaire
`module_name`	Texte	---	**architecture-export**	**Nous vous conseillons de choisir un nom en fonction de l'utilisation du module pour que votre configuration soit simple à maintenir.** Doit être unique.
`module_type`	Texte	---	**architecture-export**	Ne peut être modifié.

Chemin d'accès à NagVis

```
    ...

#  #
#      NAGVIS FILE PATH      #
#  #

#  Path of NagVis installation
#  Used to store configuration maps files and to update NagVis settings
#
#           Default : /etc/shinken/external/nagvis
#
path                                                /etc/shinken/external/nagvis

    ...
```

Afin de modifier les paramètres de NagVis et de sauvegarder les fichiers de cartes générées, le module doit connaître le chemin d'accès vers le NagVis utilisé

Nom	Type	Unité	Défaut	Commentaire
`path`	Texte	---	**/etc/shinken/external/nagvis**	Chemin d'accès vers l'installation NagVis utilisée par le module.

Options de fonctionnement du module

```
    ...

#  #
#      MODULE OPTIONS      #
#  #

#  Base of URL used to display links in the Visualization UI
#
#           Default : Use Arbiter URL
#
# map_base_url                                        http://example.com/

#  Architecture description recipients
#  When the architecture of this Shinken installation changes ( realms and daemons configuration ),
#  and the arbiter is restarted, the architecture description will be sent to the following hosts.
#
#           Default : http://127.0.0.1:7780 ( locally )
#
send_my_architecture_to_recipients                  http://127.0.0.1:7780

    ...
```

Le module génère des cartes et envois son architecture à d'autres modules du même type. Il est possible de spécifier l'URL d'accès aux cartes, mais aussi la liste des modules à qui envoyer sa propre architecture

Nom	Type	Unité	Défaut	Commentaire
`map_base_url`	Texte	---	**Url de l'Arbiter**	URL d'accès aux cartes générées par le module. L'URL sera utilisée pour les liens dans l'interface de Visualisation.
`send_my_architecture_to_recipients`	Texte	---	**http://127.0.0.1:7780**	Liste des adresses des modules architecture-export vers lesquels envoyer son architecture. Les adresses doivent être séparées par des virgules.

Communication du module

Le module doit écouter vers l'extérieur pour recevoir des architectures, mais il doit aussi communiquer avec le listener-shinken du Synchronizer pour lui envoyer les hôtes générés pour ses cartes. Ces paramètres de communications sont modifiables pour correspondre à chaque architecture.

Configuration des connexions entrantes

```
    ...

#   Listening socket configuration

#  This module opens a listening socket on which other Shinken installations
#  will send their architecture description.
#  Network interface used for the listening socket ( 0.0.0.0 = all interfaces )
#
#           Default : 0.0.0.0
#
host                                            0.0.0.0

#  Port to use for the listening socket
#
#           Default : 7780
#
port                                            7780

#  Protocol to use for the listening socket
#
#           ...     : Enable  => 1 ( Use HTTPS )
#           Default : Disable => 0 ( Use HTTP )
#
use_ssl                                         0

#  SSL Certificate to use for the listening socket ( if HTTPS )
#
#           Default : /etc/shinken/certs/server.cert
#
ssl_cert                                        /etc/shinken/certs/server.cert

#  SSL Key to use for the listening socket ( if HTTPS )
#
#           Default : /etc/shinken/certs/server.key
#
ssl_key                                         /etc/shinken/certs/server.key

    ...
```

Nom	Type	Unité	Défaut	Commentaire
`host`	Texte	---	**0.0.0.0**	Interface réseau sur lequel le module écoutera. **Remarque** : 0.0.0.0 correspond à toutes les interfaces.
	Texte	---	**7780**	Port d'écoute du module.

port				
use_ssl	Booléen	---	**0**	Paramètre activant le mode SSL (*HTTPS*). Valeurs possibles : • **0** • **1**
ssl_cert	Texte	---	**/etc/shinken/certs/server.cert**	Chemin d'accès vers le certificat SSL à utiliser (*si use_ssl est à 1*).
ssl_key	Texte	---	**/etc/shinken/certs/server.key**	Chemin d'accès vers la clé SSL à utiliser (*si use_ssl est à 1*).

Communication avec le listener-shinken

```
    ...

#   Connection with the listener-shinken

#  Connection parameters for the module to communicate with the listener-shinken
#  ( used to create hosts for maps )
#  Protocol used by listener-shinken
#
#           ...     : Enable  => 1 ( Use HTTPS )
#           Default : Disable => 0 ( Use HTTP )
#
# listener_use_ssl                                  0

#  Listener-shinken configured login
#
#           Default : Shinken
#
# listener_login                                    login

#  Listener-shinken configured password
#
#           Default : Default password generated for listener-shinken
#
# listener_password                                 pass

    ...
```

Nom	Type	Unité	Défaut	Commentaire
listener_use_ssl	Texte	---	**0**	Paramètre activant le mode SSL pour la communication avec le listener-shinken (*HTTPS*). Valeurs possibles : • **0** • **1**
listener_login	Texte	---	**Shinken**	Nom d'utilisateur à utiliser pour communiquer avec le listener-shinken.
listener_password	Texte	---	**mot de passe généré à l'installation**	Mot de passe à utiliser pour communiquer avec le listener-shinken.

Communication avec les machines graphite

```
      ...

      #   Connection with the graphite host via ssh   #

  #  Connection parameters for the module to communicate with the graphite host via ssh
  #  ( used to get the graphite configuration )
  #  These parameters will be the same for every architecture received by the module
  #  so every graphite host should allow ssh connection with these parameters
  #  SSH Port
  #
  #           Default : 22
  #
  # ssh_port                                          22

  #  SSH user
  #
  #           Default : shinken
  #
  # ssh_user                                          shinken

  #  SSH key file
  #
  #           Default : /var/lib/shinken/.ssh/id_rsa
  #
  # ssh_key_file                                      /var/lib/shinken/.ssh/id_rsa

  #  SSH timeout
  #
  #           Default : 5 seconds
  #
  # ssh_timeout                                       5

 ...
```

Nom	Type	Unité	Défaut	Commentaire
ssh_port	Nombre	---	**22**	Port SSH vers lequel se connecter aux machines graphite
ssh_user	Texte	---	**shinken**	Nom d'utilisateur à utiliser pour se connecter aux machines graphite
ssh_key_file	Texte	---	**/var/lib/shinken/.ssh/id_rsa**	Clé SSH à utiliser pour se connecter aux machines graphite
ssh_timeout	Nombre	secondes	**5**	Nombre de secondes maximum pour se connecter aux machines graphite

Paramètres de création de cartes

```
      ...

  #  #
  #      MAPS CREATION PARAMETERS      #
  #  #

  #  When an architecture description is received by the module,
  #  it creates corresponding hosts and NagVis maps.
  #  Name with which this Shinken installation will be identified in the NagVis maps
  #  The following characters are forbidden in the architecture name: ~!$%^&*"'|<>?,()=/+
```

```
#
#           Default : Shinken
#
architecture_name                                   Shinken

#
#           ...     : Sort order for realms in the NagVis maps
#           ...     : sort_by_name => Sort realms by alphabetic order
#           Default : sort_by_size => Sort realms by size (on the map)
#
# map_realm_layout                                    sort_by_size

    ...
```

Il est possible de paramétrer certains aspects de la création des cartes NagVis

Nom	Type	Unité	Défaut	Commentaire
`architecture_name`	Texte	---	**Shinken**	Nom de l'architecture de l'installation Shinken où se situe le module. Ce nom sera affiché sur les cartes NagVis et les hôtes générés
`map_realm_layout`	Texte	---	**sort_by_size**	Ordre dans lequel les royaumes sont affichés sur les cartes NagVis. Valeurs possibles : • **sort_by_name** (*Ordre alphabétique*) • **sort_by_size** (*Ordonné par taille sur la carte*)

Paramètre de communication avec le Broker et ses modules

```
    ...

#  #
#      BROKER CONNECTION PARAMETERS      #
#  #

#  These parameters are used to allow NagVis to communicate with the Broker and modules you want
#  Name of the Broker holding the modules you want NagVis to communicate with
#
#           Default : broker-master
#
# architecture_export__broker_connection__broker_name broker-master

#  Name of the Livestatus module you want NagVis to communicate with to retrieve objects information
#
#           Default : Livestatus
#
# architecture_export__broker_connection__broker_livestatus Livestatus

#  Type of the target WebUI you want to communicate with
#  This allow redirection when clicking on object on the maps
#
#           Default : module => Use a WebUI module configuration to get it's address
#           ...     : url    => Use a URL ( the WebUI address is behind a reverse proxy or use an DNS
#                               address )
#
# architecture_export__broker_connection__broker_webui_communication_type module

#  Targeted WebUI to communicate with
#  If previous parameter is set to "module", this must be a WebUI name
#  If previous parameter is set to "url", this must be a URL
#
#           Default : WebUI
#
# architecture_export__broker_connection__broker_webui_target WebUI
```

```
...
```

Afin d'avoir les statuts des hôtes sur les cartes NagVis, mais aussi d'être correctement redirigé vers une WebUI lors du clic sur l'un d'entres eux, il existe des paramètres pour rendre la communication entre NagVis et Shinken possible

Nom	Type	Unité	Défaut	Commentaire
`architecture_export__broker_connection__broker_name`	Texte	---	**broker-master**	Nom du Broker sur lequel sont les modules avec lesquels nous allons communiquer
`architecture_export__broker_connection__broker_livestatus`	Texte	---	**Livestatus**	Nom du module Livestatus avec lequel NagVis va communiquer pour afficher les statuts des hôtes sur ses cartes
`architecture_export__broker_connection__broker_webui_communication_type`	Texte	---	**module**	Type de communication avec la WebUI souhaitée. Ce paramètre est utilisé pour la redirection lorsqu'on clique sur les liens de la carte. Valeurs possibles : • **module** (*si on souhaite spécifier le nom d'un module WebUI pour obtenir son adresse*) • **URL** (*si on souhaite renseigner une URL : l'adresse de la WebUI est derrière un reverse proxy ou utilise une adresse DNS*)
`architecture_export__broker_connection__broker_webui_target`	Texte	---	**WebUI**	WebUI avec laquelle communiquer. Si le paramètre précédent est à **module** celui-ci doit être le nom d'un module WebUI Sinon, ce doit être une URL redirigeant vers un module WebUI

Envoi de la description de l'architecture

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.

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
```