

Automatisation (de la source vers la production)

Sommaire

Description

Pré-requis

- Paramétrage du synchronizer
- Paramétrage de la source
- Droits utilisateurs
- Authentification

Forcer l'import d'une source

- URL
- Paramètres
- Format de retour

Attendre la fin du calcul des différences

- URL
- Paramètres
- Format de retour

Accepter les nouveaux éléments et les différences

- URL
- Paramètres
- Format de retour

Mettre en production

- Fonctionnement de la mise en production
- URL
- Paramètres
- Format de retour

Exemples

- Forcer l'import d'une source
- Attendre la fin du calcul des différences
- Importer les nouveaux éléments et accepter les différences
- Mettre en production
- Script complet

Description

L'acceptation des hôtes via l'UI de Configuration peut s'avérer une perte de temps lorsqu'aucune action de vérification n'est nécessaire pour l'administrateur Shinken lors de l'import d'un hôte.

Si vous avez une source qui vous mets à disposition des hôtes en lequel vous avez 100% confiance, il peut devenir intéressant d'automatiser les actions suivantes :

- Forcer l'import d'un collecteur (n'aura aucun effet sur les listeners et analysers);
- Attendre la fin du calcul des différences;
- Importer les nouveaux éléments provenant de cette source en staging;
- Accepter les différences provenant de cette source;
- Mettre en production les éléments précédemment importés.

L'automatisation de ces actions est possible grâce à plusieurs URLs mises à votre disposition. Chaque URL permet d'effectuer une ou plusieurs actions pour une seule source. Si vous souhaitez automatiser plusieurs sources, il faudra répéter ces actions.

Il vous est donc possible d'écrire un script afin de lancer ces différentes actions selon vos besoins.



Seule la source "syncui" ne peut pas être utilisée dans l'automatisation

Pré-requis

Dans cette documentation, les différentes URL sont de type "GET".

Avant de vous lancer dans l'écriture d'un script d'automatisation, certains prérequis sont à respecter :

Paramétrage du synchronizer

Par défaut, les URL sont désactivées sur le Synchronizer. Pour les activer, il faut ajouter un paramètre dans le fichier de configuration du Synchronizer. Il est conseillé de modifier le fichier suivant :

```
/etc/shinken-user/configuration/daemons/synchronizers/synchronizer_cfg_overload.cfg
```

```
api_trusted_source_enabled=1
```

Mettre ce paramètre à 0 désactivera les URL.

Après modification de ce paramètre, il convient de redémarrer le Synchronizer pour que le changement soit pris en compte.

Paramétrage de la source

L'automatisation peut être mise en place pour tous les types de sources. Les prérequis concernant les sources sont :

- La source doit être présente dans la liste des sources;
- La source ne doit pas être en statut critique (pas d'erreur sur les fichiers ou erreur de connexion);
- La source doit être activée.



L'automatisation doit porter sur une source considérée comme étant fiable, car :

- Tous les hôtes de cette source seront mis en production sans exception;
- Toutes les différences provenant cette source seront acceptées et mises en production;
- Aucune confirmation ne sera demandée.



Actuellement ce mécanisme n'est utilisable que pour importer des hôtes. Nous vous conseillons d'utiliser une source spécifique uniquement pour l'automatisation et une seconde source pour importer les modèles ou tout autre élément pouvant être utilisés par cette première source.

Droits utilisateurs

Un compte utilisateur défini dans le Synchronizer sera nécessaire pour simuler les différentes actions :

- Le compte doit être activé
- Le compte doit disposer des droits Administrateur Shinken
- Vous devez connaître le mot de passe du compte



Nous vous conseillons de créer un compte spécifique pour cette utilisation afin d'identifier rapidement les éléments importés. Il vous sera possible de le désactiver rapidement en cas de besoin.

Authentification

- Les URL sont soumises à l'authentification;
- Chacune étant indépendante, l'authentification devra être effectuée pour chaque URL;
- Les mots de passe devront être encodés en base 64;
- Les modules d'authentification (ex. : Active-directory) configurés sur le Synchronizer seront également disponibles pour authentifier les utilisateurs.



En shell le base64 du mot de passe peut être obtenu à l'aide de la commande suivante : `echo -ne "password" | base64`

Pour toutes les URL suivantes, si une erreur d'authentification est rencontrée les codes d'erreurs suivants seront renvoyés :

Code d'erreur	Message	Raison
403	The user <contact_name> is not enabled	Le contact est désactivé
403	The user <contact_name> is not a shinken admin	Le contact n'est pas administrateur Shinken
400	Parameter password is not in base64	Le mot de passe donné n'est pas en base 64

400	Parameter login and password are mandatory	Le login ou le mot de passe (ou les deux) n'ont pas été saisis
403	Forbidden user	Le mot de passe ou le login (ou les deux) ne sont pas corrects



Il n'existe pas de méthode fiable pour savoir si une chaîne de caractères est bien en base64. Il se peut donc qu'un mot de passe envoyé en clair soit décodé alors qu'il n'est pas en base64. L'utilisateur obtiendra donc une erreur "403 Forbidden user" au lieu de l'erreur "400 Parameter password is not in base64".

Forcer l'import d'une source

La première étape est d'importer la source afin qu'elle propose de nouveaux éléments ou des différences avec les éléments existants.

- Il est donc possible de forcer cet import avec un appel afin de travailler sur des objets à jour.
- Le comportement de cet appel est le même que le bouton "Forcer l'import" sur la page des sources : Il demande à la source d'importer tous ses éléments.
- Comme sur la page des sources, l'utilisation de cette fonctionnalité déclenche le calcul des différences une fois terminé.



Cette étape n'est utile que pour les collecteurs:

En effet:

- Les écouteurs n'ont pas de mécanisme d'import. Dès qu'ils reçoivent une requête, son résultat est automatiquement disponible sans intervention de l'utilisateur.
- Les analyseurs nécessitent le choix d'une plage d'adresses ou une sélection d'hôte pour fonctionner. Cet appel bien que fonctionnel n'aura donc aucun effet sur un analyseur.

URL

L'URL est la suivante :

```
http://ip-du-synchroniser:7765/trusted-source/v1/force-source-import
```

Paramètres

Cette URL a besoin des paramètres suivants:

Option	Exemple	Description
login	login=mon-user	Le nom du contact utilisé pour se connecter
password	password=VGvZdF8wMQ==	Le mot de passe du contact, en base 64
source_name	source_name=source_aws	Le nom de la source qui sera importé

Format de retour

Si aucune erreur n'est rencontrée, l'URL retourne le code 200 et le message "OK".

En plus des codes d'erreur spécifiques à l'authentification, voici les codes d'erreur possibles :

Code d'erreur	Message	Raisons
400	Missing parameters : <parameters>	Des paramètres obligatoires n'ont pas été saisis
500	Source [<source_name>] is a internal source, we cannot import elements from this source	La source demandée est une source interne et ne peut être utilisée ici
503	Source [<source_name>] is disabled	La source demandée est désactivée
504	Syncui is not ready for the moment, need to retry in a few moment	Le Synchronizer est en train de redémarrer et n'est pas prêt. Vous pouvez réessayer dans quelques instants.
400	Unknow source	La source demandée n'est pas présente sur le Synchronizer

Attendre la fin du calcul des différences

Le calcul des différences est une action qui est automatiquement lancée après chaque import de sources. Son rôle est de comparer les éléments déjà présents en staging avec ceux importés par les sources. C'est cette étape qui permet d'obtenir les nouveaux éléments et les différences sur les éléments existants.

L'URL permet d'obtenir l'état du calcul des différences après un import.



Cette étape est la seule qui ne nécessite pas d'authentification puisque l'utilisateur n'interagit pas directement avec Shinken, mais demande seulement l'état du calcul des différences.

URL

L'URL est la suivante :

```
http://ip-du-synchroniser:7765/trusted-source/v1/get-api-source-controller-importing
```

Paramètres

Cette URL ne nécessite aucun paramètres.

Format de retour

L'appel à cette URL devra donc être répété (toutes les secondes par exemple) tant que le calcul n'est pas terminé. Cette étape terminée, il est possible de faire l'action suivante () .

Il n'y a pas de code d'erreur pour cet appel.

Code de retour	Message	Raisons
200	true	Le calcul des différences est en cours
200	false	Le calcul des différences est terminé

Accepter les nouveaux éléments et les différences

Une fois le calcul des différences terminé, il est possible d'accepter :

- les différences provenant de cette source sur les éléments existants
- les nouveaux éléments provenant de cette source.



Les hôtes seront directement importés en staging et pas en zone de travail.

URL

L'URL est la suivante :

```
http://ip-du-synchroniser:7765/trusted-source/v1/force-trusted-source-behaviour
```

Paramètres

Cette URL a besoin des paramètres suivants:

Option	Exemple	Description
login	login=mon-user	Le nom du contact utilisé pour se connecter
password	password=VGvzdF8wMQ==	Le mot de passe du contact, en base 64
filter	filter=sources:source_aws	Le filtre utilisé. Voir la note ci-après

Note sur le filtre : il correspond au filtre de la liste tous les éléments. Sa valeur est composée :

- du nom du filtre
- du caractère ":"
- de la valeur du filtre



Actuellement, le filtre ne supporte qu'un seul paramètre : le nom de la source. Son format est donc `filter=sources:<source_name>`

Format de retour

Si l'appel s'est bien déroulé, il renvoie un code 200 et aucun message.

En plus des codes d'erreur spécifiques à l'authentification, voici les codes d'erreur possibles :

Code d'erreur	Message	Raisons
400	Missing parameters : <parameters>	Des paramètres obligatoires n'ont pas été saisis
500	Source [<source_name>] is a internal source, we cannot import elements from this source	La source demandée est une source interne et ne peut être utilisée ici.
500	Source [<source_name>] has errors, we cannot import elements from this source.	L'état de la source est en Critique
500	Source [<source_name>] is not configured.	La source est dans l'état Non configurée. Il est donc impossible d'importer des éléments
503	Source [<source_name>] is disabled	La source demandée est désactivée
504	Syncui is not ready for the moment, need to retry in a few moment	Le Synchronizer est en train de redémarrer et n'est pas prêt. Vous pouvez réessayer dans quelques instants.
500	Invalid filter	Le format du filtre est invalide
500	Source [<source_name>] in filter was unknow	Le filtre est valide mais la source indiquée est inconnue
506	applied difference on [hosts in staging named : <host_name>] failed with error : [<error>]	L'acceptation des différences a échoué avec l'erreur indiquée. Cette erreur peut être générée lorsque la différence cause une erreur de validation de la propriété ou donnée.

Mettre en production

Une fois que toutes les modifications apportées par la source ont été acceptées et importées, il reste à mettre ces modifications en production.

Dans le cas de l'automatisation, seules les modifications apportées par la source sur les hôtes doivent être envoyées en production.

Fonctionnement de la mise en production

Pour réaliser cette mise en production, le Synchronizer réalise les étapes suivantes :

- Réalisation d'une copie de sauvegarde de la base de production
- Ajout des hôtes présent en Staging dont la source est celle indiquée dans l'appel dans cette copie de sauvegarde
- Vérification de la configuration depuis la copie de sauvegarde :
 - Si elle est correcte, on l'envoie à l'Arbiter.
 - Sinon on renvoie une erreur (code 510) et aucune action n'est réalisée.



Si la configuration est envoyée à l'Arbiter, celui-ci redémarrera pour appliquer la nouvelle configuration. Si l'Arbiter était éteint il sera démarré.

Les éléments suivants ne seront pas mis en production :

- les hôtes n'appartenant pas à la source indiquée dans l'appel
- les éléments qui ne sont pas des hôtes, même s'ils appartiennent à la source indiquée

Si la source définit des hôtes utilisant des éléments qui ne sont pas présents en production, l'automatisation échouera à cause d'une erreur de configuration.

Exemple :

- La source définit des hôtes utilisant un parent nommé "firewall".
- Un administrateur définit cet hôte en staging depuis l'interface de configuration
- Il vérifie la configuration sans le mettre en production : la configuration est correcte.
- Un appel est réalisé pour mettre la source en production.
 - L'hôte firewall n'étant pas en production, les hôtes utilisent un parent inexistant et ne sont pas corrects. l'appel échoue avec une erreur 510
- L'administrateur revérifie la configuration : elle est correcte, car elle prend tous les éléments de staging

Dans cet exemple, si l'administrateur effectue une mise en production depuis la base de staging, les prochains appels aboutiront, car le parent existe en production.

URL

L'URL est la suivante :

```
http://ip-du-synchroniser:7765/trusted-source/v1/put-in-production
```

Paramètres

Cette URL a besoin des paramètres suivants:

Option	Exemple	Description
login	login=mon-user	le nom du contact utilisé pour se connecter
password	password=VGvzdF8wMQ==	Le mot de passe du contact, en base 64
filter	filter=sources: source_aws	Le filtre utilisé. Il est du même format et avec les mêmes contraintes que pour l'appel d'import et d'acceptation des différences.
item_type	item_type=hosts	le type d'élément à mettre en production



Actuellement seul le type d'élément hôte est supporté. son format est donc "item_type=hosts"

Format de retour

Si l'appel s'est bien déroulé, il renvoie un code 200 avec le message "arbiter reload OK"

En plus des codes d'erreur spécifiques à l'authentification, voici les codes d'erreur possibles :

Code d'erreur	Message	Raisons
400	Missing parameters : <parameters>	Des paramètres obligatoires n'ont pas été saisis
500	Source [<source_name>] is a internal source, we cannot import elements from this source	La source demandée est une source interne et ne peut être utilisée ici.
500	Source [<source_name>] has errors, we cannot import elements from this source.	L'état de la source est en critique
504	Syncui is not ready for the moment, need to retry in a few moment	Le Synchronizer est en train de redémarrer et n'est pas prêt. Vous pouvez réessayer dans quelques instants.
500	Invalid filter	le format du filtre est invalide
500	Source [<source_name>] in filter was unknow	Le filtre est valide mais la source indiquée est inconnue
500	This item_type is not yet implemented	Actuellement seul les éléments de type hôte sont supportés
505	Arbiter is reloading a new configuration	L'Arbiter est en train de recharger une configuration (redémarrage du démon ou application de la configuration sur l'interface de configuration)
510	<configuration_error>	La configuration est incorrecte. L'erreur complète est indiquée dans le message.

Exemples

Forcer l'import d'une source

```
root@shinken: ~/ 15:58 : $ curl "http://localhost:7765/trusted-source/v1/force-source-import?
password=VG Vz d F 8 w M Q == & login = Test_01 & source_name = source_aws"
"OK"
root@shinken: ~/ 15:58 : $
```

Attendre la fin du calcul des différences

```
root@shinken: ~/ 15:58 : $ curl "http://localhost:7765/trusted-source/v1/get-api-source-controller-importing"
true
root@shinken: ~/ 15:59 : $ curl "http://localhost:7765/trusted-source/v1/get-api-source-controller-importing"
false
root@shinken: ~/ 15:59 : $
```

Importer les nouveaux éléments et accepter les différences

```
root@shinken: ~/ 15:59 : $ curl "http://localhost:7765/trusted-source/v1/force-trusted-source-behaviour?
password=VG Vz d F 8 w M Q == & filter = sources : source_aws & login = Test_01"
""
root@shinken: ~/ 15:59 : $
```

Mettre en production

```
root@shinken: ~/ 15:59 : $ curl "http://localhost:7765/trusted-source/v1/put-in-production?
password=VG Vz d F 8 w M Q == & filter = sources : source_aws & item_type = hosts & login = Test_01"
"arbiter reload OK"
root@shinken: ~/ 16:00 : $
```

Script complet

Voici un exemple de script écrit en python :

[Télécharger le script.](#)

api_trusted_source.py

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
#
# Copyright (C) 2013-2019:
# This file is part of Shinken Enterprise, all rights reserved.

import time
import sys
from optparse import OptionParser
import pycurl
from StringIO import StringIO
from urllib import urlencode
import base64

def call_url(url, debug):
    buffer = StringIO()
    my_curl = pycurl.Curl()
```

```

my_curl.exception = None
my_curl.setopt(my_curl.URL, url)
my_curl.setopt(my_curl.WRITEFUNCTION, buffer.write)
try:
    my_curl.perform()
except pycurl.error as pycurl_error:
    if pycurl_error.args[0] == pycurl.E_COULDNT_CONNECT and my_curl.exception:
        print 'The curl failed on url "%s" with the error : %s' % (url, my_curl.exception)
        sys.exit(2)
    else:
        print 'The curl failed on url "%s" with the error : %s' % (url, pycurl_error)
        sys.exit(2)
http_code = my_curl.getinfo(my_curl.RESPONSE_CODE)
my_curl.close()
data = buffer.getvalue().strip()
if http_code != 200:
    print 'The call to "%s" return an http error %s' % (url, http_code)
    if debug:
        print 'The full result is :\n%s' % data
    sys.exit(2)
return data

def call_api(opts, route, data={}):
    host = opts.host
    port = opts.port
    password = opts.password
    if password:
        url = "http://%s:%s/%s?password=%s&%s" % (host, port, route, base64.b64encode(password), urlencode
(data))
    else:
        url = "http://%s:%s/%s?%s" % (host, port, route, urlencode(data))

    if opts.debug:
        print '[DEBUG] Will call url [%s]' % url
    return call_url(url, opts.debug)

def get_parsed_options():
    parser = OptionParser(description='trusted source')
    parser.add_option('-u', '--user', dest="user", default='admin', help='Set the user')
    parser.add_option('-p', '--password', dest="password", default='admin', help='Set the password')
    parser.add_option('-H', '--host', dest="host", default='localhost', help='Set the host')
    parser.add_option('-P', '--port', dest="port", default='7765', help='Set the port')
    parser.add_option('-s', '--source', dest="source", help='Set the source')
    parser.add_option('-t', '--type', dest="item_type", default='hosts', help='Set the item_type')
    parser.add_option('-d', '--debug', action='store_true', dest='debug', help='Set debug mode')

    opts, args = parser.parse_args()
    if not opts.source:
        parser.error('Please set the source.')
    return opts

def main():
    opts = get_parsed_options()

    data = {
        'login' : opts.user,
        'source_name': opts.source,
    }

    print "\nCall the import for source [%s]" % opts.source
    ret = call_api(opts, 'trusted-source/v1/force-source-import', data)
    if not ret == "OK":
        sys.exit(-2)
    print "OK !\n"
    if opts.debug:
        print "[DEBUG] Result is : %s" % ret

    print "\nMerge in progress "

```

```

while call_api(opts, 'trusted-source/v1/get-api-source-controller-importing') == 'true':
    time.sleep(1)
    print '.'

print "OK !\n"

print"\nApply diff and import to staging"
del data['source_name']
data['filter'] = "sources:%s" % opts.source
ret = call_api(opts, 'trusted-source/v1/force-trusted-source-behaviour', data)
print "OK !\n"
if opts.debug:
    print "[DEBUG] Result is : %s" % ret

print"\nPut in production"
data['item_type'] = opts.item_type
ret = call_api(opts, 'trusted-source/v1/put-in-production', data)
print "OK !\n"
if opts.debug:
    print "[DEBUG] Result is : %s" % ret

if __name__ == '__main__':
    main()

```

Voici son exécution :

```

$ ./api_trusted_sources.py -s source_aws -u Test_01 -p
Test_01
Call the import for source [source_aws]
OK !

Merge in progress
.
.
.
.
.
.
OK !

Apply diff and import to staging
OK !

Put in production
OK !

$

```