Exemple d'automatisation

Sommaire

Forcer l'import d'une source

```
root@shinken: ~/ 15:58 : $ curl "http://ip-du-synchroniser:7765/trusted-source/v1/force-source-import?
password=VGVzdF8wMQ==&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://ip-du-synchroniser:7765/trusted-source/v1/get-api-source-controller-
importing"
true
root@shinken: ~/ 15:59 : $ curl "http://ip-du-synchroniser: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://ip-du-synchroniser:7765/v1/trusted-source/force-trusted-source-
behaviour?password=VGVzdF8wMQ==&filter=sources:source_aws&login=Test_01"
""
root@shinken: ~/ 15:59 : $
```

Mettre en production

```
root@shinken: ~/ 15:59 : $ curl "http://ip-du-synchroniser:7765/v1/trusted-source/put-in-production?
password=VGVzdF8wMQ==&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 !


$
```