Table of Contents

Bridging Dirac/Sam Jobs to SAM/nagios

Configuring SAM/Nagios

SAM/Nagios can be configured to accept external ("passive") service checks following the documentation under :

https://tomtools.cern.ch/confluence/display/SAM/Support+for+external+publishers

The following steps have been taken to configure a new service-check on `sam-lhcb-dev`:

1. Adding the new metrics to the Profile using POEM `sam-lhcb-dev/poem` (see https://tomtools.cern.ch/confluence/display/SAMDOC/POEM+User%27s+Guide)

2. Adding the metrics to the ncg-config files under =/etc/metrics-config.d/myNewMetric.conf

```
{
   "org.lhcb.DiracTest" : {
      "parent" : "org.lhcb.CE-AllLHCb",
      "docurl" : "No Documentation Yet, Test for Summer Student Project",
      "flags" : {
         "OBSESS" : 1,
         "VO" : 1,
         "PASSIVE" : 1
      },
      "metricset" : "org.lhcb.CE"
   }
```

Publishing Test Messages from a `stomppy` client

At the moment, I use the `stomppy` library to publish test messages containing the results of the newly created service checks. The messages will arrive on `sam-lhcb-dev` when sent to the queue = /queue/grid.probe.metricOutput.EGEE.sam-lhcb-dev_cern_ch= on the broker `sam-validation.msg.cern.ch:6163`

The complete code for the publisher reads:

```
#!/usr/bin/python

############################################################
# Preliminary script to publish test messages to SAM/Nagios #
# via activemq/stomp. The Message broker and the queue are  #
# specified in the config file CONF_FILE.                   #
# Part of a Summer Student Project.                         #
# 12.07.2013 Contact: valentin volkl cern ch                #
#                                                           #
############################################################



import time
import sys
import stomp

import argparse
import ConfigParser

CONF_FILE = 'DiracPublisher.cfg'
```

```
# check for correct argument usage
# print help message
parser = argparse.ArgumentParser(description="""Test Publishing
    Application for Passive Service Checks to SAM/Nagios.""")
parser.add_argument("-v","--verbose", help="increase output verbosity",
                    action="store_true")

args = parser.parse_args()

# read config file
config = ConfigParser.ConfigParser()

try:
    with open(CONF_FILE,'r') as f: pass
except IOError:
    print """The configuration file could not be found.
Check if it is located as: """ + CONF_FILE
    raise IOError

config.read(CONF_FILE)

BROKER = config.get("Connection", "BROKER")
PORT = config.getint("Connection", "PORT")
QUEUE = config.get("Connection", "QUEUE")



#TODO: write proper parser for details of service check
with open('testmsg_minimal.msg') as f:
    nagmsg = f.read()

if args.verbose:
    print '######## Message to be sent: ########'
    print nagmsg
    print '######## End of Message      ########'
    print 'Broker: ', BROKER
    print 'Port: ', PORT
    print 'Queue: ', QUEUE

class MyListener(object):
    def on_error(self, headers, message):
        print 'received an error %s' % message

    def on_message(self, headers, message):
        print 'received a message %s' % message

conn = stomp.Connection( [ (BROKER, PORT) ] )
conn.set_listener('',MyListener())
conn.start()
conn.connect()

#TODO: check what effects subscribing has to sent messages (queue behaviour!)
# subscribing would be quite useful as nagios returns error messages when not
# correctly formatted messages arrive
#conn.subscribe(destination=QUEUE, ack='auto')

conn.send(nagmsg, destination=QUEUE)
if args.verbose:
    print '######## Message successfully sent! ########'


conn.disconnect()
```

A correctly formatted example message (content of `testmsg.msg`) is:

```
hostName: ce.hpc.iit.bme.hu
metricStatus: OK
timestamp: 2013-11-09T17:59:19Z
nagiosName: org.lhcb.DiracTest-lhcb
summaryData: External publishing fully successful
serviceURI: atlas-cream01.na.infn.it
serviceFlavour: CE
siteName: myTestSite
metricName: org.lhcb.DiracTest
gatheredAt: sam-developers-machine
role: site
voName: lhcb
serviceType: org.lhcb.CE
detailsData: Test Service for the Publication of Dirac Probes to Nagios. Contact: Valentin Volkl
EOT
```

The way this message is parsed is documented in MsgNagiosBridgeParser.

TODO:

1. write some sort of metricparser, that will assemble the message above from some input

2. check alternatives to stomppy

Integration to DIRAC

For the development of Dirac, a LocalDiracDeveloperEnvironment on a local machine is helpful. There i will proceed to rewrite the above script with the proper Dirac Log and Config functions and write an Utility `NagiosConnector`. This text will be updated with details.

This topic: Sandbox > ValentinVolklSandbox
Topic revision: r3 - 2013-07-26 - ValentinVolkl