

Autoadd new devices in microservices

Костыль для добавления хостов в МО по ping.

По мотивам [Автоматизация добавления новых хостов в МО \(bash, nmap\)](#) (develop)

cron.sh – сам скрипт

excludescan.tmp – временный файл для выгрузки исключений

excludescan.txt – постоянный файл исключений

getexclude.sql – запрос для выгрузки уже существующих адресов из NOC

networkscan.txt – сети, среди которых искать отвечающие на пинг устройства.

Настройка:

1. Дебаг
2. Проверяем, если всё плохо, goto 1.
3. Ставим в крон как cron.sh scan

cron.sh

```
#!/bin/bash
export LANG="ru_RU.UTF-8"
export LOCALE="ru_RU"
export BASEDIR=/opt/scripts/add-mo
export PGPASSWORD='noc';
MAXTHREAD=20
NMAPBIN="/usr/bin/nmap -sn -n -iL $BASEDIR/networkscan.txt --excludefile $BASEDIR/excludescan.tmp -oG $BASEDIR/out.scan.txt"
# IP , .
/usr/bin/psql -h 127.0.0.1 -U noc --dbname noc -X -A -t -q --no-align -f $BASEDIR/getexclude.sql -o $BASEDIR/excludescan.tmp
#
cat $BASEDIR/excludescan.txt >> $BASEDIR/excludescan.tmp
# by-default csv
is_managed="True"
administrative_domain="default"
profile_name="Generic.Host"
object_profile="default"
description=$(/bin/date +%d.%m.%Y)
segment="ALL"
auth_profile="Generic.Host"
scheme="1"
address=""
port=""
user=""
password=""
super_password=""
remote_path=""
trap_source_ip=""
trap_community=""
snmp_ro=""
snmp_rw=""
vc_domain=""
vrf=""
termination_group=""
service_terminator=""
shape="Cisco/router"
config_filter_rule=""
config_diff_filter_rule=""
```

```

config_validation_rule=""
max_scripts="5"
tags="autoadd"
pool="default"
container=""
pool="default"
trap_source_type="m"
syslog_source_type="m"
object_profile="default"
time_pattern=""
x=""
y=""
default_zoom=""
headcvs="name,is_managed,container,administrative_domain,segment,pool,profile_name,object_profile,description,
auth_profile,scheme,address,port,user,password,super_password,remote_path,trap_source_type,trap_source_ip,
syslog_source_type,syslog_source_ip,trap_community,snmp_ro,snmp_rw,vc_domain,vrf,termination_group,
service_terminator,shape,time_pattern,config_filter_rule,config_diff_filter_rule,config_validation_rule,
max_scripts,x,y,default_zoom,tags"
#
if [ "$1" = "scan" ]
then
#
cd $BASEDIR && rm result* out*
#
$NMAPBIN
grep "Host" $BASEDIR/out.scan.txt | grep "Status: Up" | awk '{print $2}' > $BASEDIR/out.hostsscan.txt
fi
#
while read -r line
do
name=$line
echo "$name"
read -r address <<< "$line"
#
if [ -n "$address" ]
then
# Load Average
while [ $($(/bin/ps aux | grep -c sql.sh) - 1) -ge "$MAXTHREAD" ]
do
# echo $($(/bin/ps aux | grep "cli-commands" | wc -l) - 1).
sleep 1
# echo "sleep"
done
SQLQUERY="SELECT address FROM sa_managedobject where address = '$address' and name not like 'wiping-%'
limit 1 ;"
SQLRESULT=$(echo "$SQLQUERY" | psql -h 127.0.0.1 -U noc --dbname noc -X -A -t -q --no-align)
#
RETVAL=$?
echo $RETVAL
if [ $RETVAL -eq 1 ]
then
echo "Postgress is unavailable"
exit
fi
if [ -n "$SQLRESULT" -a $RETVAL -eq 0 ]
then
# SNMP
echo -e "$SQLRESULT" " "
#
echo -e "$address" >> $BASEDIR/out.hostsadd.txt
else
echo -e "$SQLRESULT" " "
# default
# out.hostsadd.txt

```

```

# ping syslog
  trap_source_ip="$address"
  syslog_source_ip="$address"
# , PTR
  fqdn=$(host "$address" | awk '{print $5}' | sed 's/\.domain\.ru\.$/')

ADDRROW="$fqdn,$is_managed,$container,$administrative_domain,$segment,$pool,$profile_name,$object_profile,$description,$auth_profile,$scheme,$address,$port,$user,$password,$super_password,$remote_path,$trap_source_type,$trap_source_ip,$syslog_source_type,$syslog_source_ip,$trap_community,$snmp_ro,$snmp_rw,$vc_domain,$vrf,$termination_group,$service_terminator,$shape,$time_pattern,$config_filter_rule,$config_diff_filter_rule,$config_validation_rule,$max_scripts,$x,$y,$default_zoom,$tags"
  echo -e "$ADDRROW" >> $BASEDIR/out.hostsadd.txt
fi

fi
echo -e "-----\n"
done < $BASEDIR/out.hostsscan.txt

# .
# CVS " " "

sleep 5

if [ -a $BASEDIR/out.hostsadd.txt ]
then
echo $headcvcs > $BASEDIR/result.hostsadd.csv
cat out.hostsadd.txt >> $BASEDIR/result.hostsadd.csv

# .
cd /opt/noc && ./noc csv-import sa.managedobject --resolve=skip $BASEDIR/result.hostsadd.csv
RETVAL=$?

if [ $RETVAL -ne 0 ]
then
echo "error import"
/usr/bin/mail -s " " -A $BASEDIR/result.hostsadd.csv noc@domain.ru < $BASEDIR/result.hostsadd.csv
fi
#
/usr/bin/mail -s " " -A $BASEDIR/result.hostsadd.csv noc@domain.ru < $BASEDIR/result.hostsadd.csv
fi

# .
if [ ! -f $BASEDIR/data/"${/bin/date +%Y}" ]
then
mkdir $BASEDIR/data/"${/bin/date +%Y}"
fi
cp $BASEDIR/result.hostsadd.csv $BASEDIR/data/"${/bin/date +%Y%m%d} ".csv

```