EPAM Systems, RD Dep., RD Dep.

MTN.NIX.05.Zabbix.API

REVISION HISTORY					
Ver.	**Description of Change**	**Author**	**Date**	**Approved**	
				Name	**Effective Date**
1.0	Initial Version	Siarhei Beliakou	23/May/2017		

Legal Notice This document contains privileged and/or confidential information and may not be disclosed, distributed or reproduced without the prior written permission of EPAM Systems.

Task. Zabbix API

Task:

You should develop a script (on Python 2.x) which registers given host in Zabbix.

Testing Infrastructure:

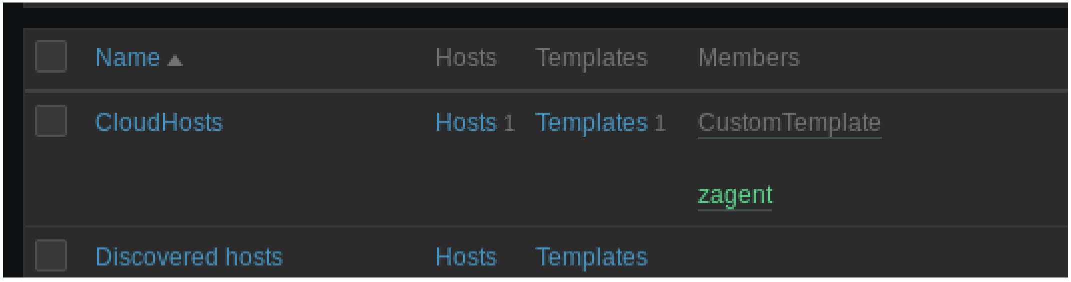
Vagrantfile to spin up 2 VMs (virtualbox):

- zabbix server, provisioned by Vagrant provisioner
- Linux VM with zabbix agent, script for registration on zabbix server, all provisioned by Vagrant provisioner

Registering Script requirements:

- Written on Python 2.x
- Starts at VM startup or on provision phase
- Host registered in Zabbix server should have Name = Hostname (not IP)
- Host registered in Zabbix server should belong to ”CloudHosts” group
- Host registered in Zabbix server should be linked with Custom template
- This script should create group “CloudHosts” id it doesn’t exist

For both VMs use vagrant box “sbeliakou/centos-7.3-x86_64-minimal”



□	Name ▲	Hosts	Templates	Members
□	CloudHosts	Hosts 1	Templates 1	CustomTemplate zagent
□	Discovered hosts	Hosts	Templates	

```
    #!/bin/bash

#***********************************************************************
#Beginning of custom variables
#Set these to appropriate values before executing script...

baseDir=$(pwd)

zabbixServer='192.168.56.141'

#########################
#only for zabbix API
zabbixUsername='Admin'
zabbixPassword='zabbix'
###########################
#zabbixHostGroup='Linux servers'
#maintenanceWindowName="Maintenance Window for $zabbixHostGroup"

hgroup="CloudHosts"
```

```
ipaddr=$(hostname -I | awk '{print $2}')
client=$(hostname)

customtemp="CustomTemplate"

if [ "$(hostname)" = zabbix-server ]
then

sudo rpm -Uvh https://repo.zabbix.com/zabbix/4.4/rhel/7/x86_64/zabbix-release-4.4-1.el7.noarch.rpm

sudo yum clean all

sudo yum install -y zabbix-server-pgsql zabbix-web-pgsql zabbix-get zabbix-sender zabbix-agent postgresql-server vim htop  postgresql-contrib zabbix-sender zabbix-get zabbix-java-gateway
sudo postgresql-setup initdb
sudo systemctl start postgresql
sudo systemctl enable postgresql

sudo -u postgres psql -c "CREATE USER zabbix WITH PASSWORD 'zabbix'";
sudo -u postgres createdb -O zabbix zabbix
zcat /usr/share/doc/zabbix-server-pgsql-4.4.0/create.sql.gz | sudo -u zabbix psql zabbix

cat > /var/lib/pgsql/data/pg_hba.conf  <<HBA
# TYPE  DATABASE        USER            ADDRESS                 METHOD
host    zabbix          zabbix          127.0.0.1/32            password
local   all             all                                     peer
host    all             all             ::1/128                 md5
HBA

sudo sed -i 's/#\ php_value date.timezone\ Europe\/Riga/php_value date.timezone Europe\/Minsk/g' /etc/httpd/conf.d/zabbix.conf 

sudo sed -i 's/#\ DBPassword=/DBPassword=zabbix/' /etc/zabbix/zabbix_server.conf
sudo sed -i 's/#\ DBHost=localhost/DBHost=127.0.0.1/' /etc/zabbix/zabbix_server.conf
sudo sed -i 's/#\ JavaGateway=/JavaGateway=127.0.0.1/' /etc/zabbix/zabbix_server.conf
sudo sed -i 's/#\ JavaGatewayPort=10052/JavaGatewayPort=10052/' /etc/zabbix/zabbix_server.conf
sudo sed -i 's/#\ StartJavaPollers=0/StartJavaPollers=5/' /etc/zabbix/zabbix_server.conf

cat <<EOF > /etc/zabbix/web/zabbix.conf.php
<?php
// Zabbix GUI configuration file.
global \$DB;

\$DB['TYPE']      = 'POSTGRESQL';
\$DB['SERVER']    = '127.0.0.1';
```

```
\$DB['PORT']     = '0';
\$DB['DATABASE'] = 'zabbix';
\$DB['USER']     = 'zabbix';
\$DB['PASSWORD'] = 'zabbix';

// Schema name. Used for IBM DB2 and PostgreSQL.
\$DB['SCHEMA'] = '';

\$ZBX_SERVER      = 'localhost';
\$ZBX_SERVER_PORT = '10051';
\$ZBX_SERVER_NAME = '';

\$IMAGE_FORMAT_DEFAULT = IMAGE_FORMAT_PNG;
EOF
sudo chown apache:apache /etc/zabbix/web/zabbix.conf.php

cat <<EOF > /etc/httpd/conf.d/zabbix.conf
<VirtualHost *:80>
DocumentRoot /usr/share/zabbix
ServerName 192.168.56.141

 <Directory "/usr/share/zabbix">
  Options FollowSymLinks
  AllowOverride None
  Require all granted

  <IfModule mod_php5.c>
    php_value max_execution_time 300
    php_value memory_limit 128M
    php_value post_max_size 16M
    php_value upload_max_filesize 2M
    php_value max_input_time 300
    php_value always_populate_raw_post_data -1
    php_value date.timezone America/Toronto
  </IfModule>
 </Directory>

 <Directory "/usr/share/zabbix/conf">
  Require all denied
 </Directory>

 <Directory "/usr/share/zabbix/app">
  Require all denied
 </Directory>

 <Directory "/usr/share/zabbix/include">
  Require all denied
 </Directory>
```

```
  <Directory "/usr/share/zabbix/local">
    Require all denied
  </Directory>
</VirtualHost>
EOF

sudo systemctl restart zabbix-server zabbix-agent httpd postgresql zabbix-java-gateway
sudo systemctl enable zabbix-server zabbix-agent httpd postgresql zabbix-java-gateway



################################################################################
#zabbix clients
################################################################################
else

sudo rpm -Uvh https://repo.zabbix.com/zabbix/4.4/rhel/7/x86_64/zabbix-release-4.4-1.el7.noarch.rpm

sudo yum clean all

sudo yum install -y zabbix-agent vim zabbix-sender zabbix-get jq epel-release vim java java-devel

sudo sed -i 's/127.0.0.1/192.168.56.141/g' /etc/zabbix/zabbix_agentd.conf
sudo sed -i 's/Hostname=Zabbix\ server/Hostname=zagent/g' /etc/zabbix/zabbix_agentd.conf

sudo systemctl enable zabbix-agent
sudo systemctl start zabbix-agent

#cho 'install epel'
#sudo yum install -y epel-release vim> /dev/null
#echo 'install JDK'

#sudo yum localinstall -y /vagrant/jdk-8u221-linux-x64.rpm

echo 'create user'
sudo mkdir /opt/tomcat
sudo groupadd tomcat
sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat

#sudo echo 'export JAVA_HOME=/usr/java/jdk1.8.0_221-amd64/jre' >> ~/.bash_profile
#sudo echo 'export PATH=$JAVA_HOME/bin:$PATH' >> ~/.bash_profile

cd /opt/
sudo curl http://ftp.byfly.by/pub/apache.org/tomcat/tomcat-8/v8.5.47/bin/apache-tomcat-
```

```
8.5.47.tar.gz --output ./apache-tomcat-8.5.47.tar.gz
sudo tar -xzvf apache-tomcat-8.5.47.tar.gz

sudo mv apache-tomcat-8.5.47/* tomcat/

sudo curl http://repo2.maven.org/maven2/org/apache/tomcat/tomcat-catalina-jmx-remote/8.5.6/tomcat-catalina-jmx-remote-8.5.6.jar --output /opt/tomcat/lib/tomcat-catalina-jmx-remote-8.5.6.jar

cat <<EOF > /tmp/setenv.sh
#!/usr/bin/env bash
export JAVA_OPTS="-Dcom.sun.management.jmxremote=true
-Xms256m
-Xmx512m
-verbose:gc
-XX:+PrintGCDetails
-XX:+PrintGCTimeStamps
-XX:+PrintGCDateStamps
-XX:+PrintGCCause
-Xloggc:/opt/tomcat/logs/gc.log
-XX:+HeapDumpOnOutOfMemoryError
-XX:HeapDumpPath=/opt/tomcat/logs/HeapOutMemDump.hprof
-Djava.rmi.server.hostname=$ipaddr
-Dcom.sun.management.jmxremote.port=12345
-Dcom.sun.management.jmxremote.rmi.port=12346
-Dcom.sun.management.jmxremote.local.only=false
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false"

EOF
sudo mv /tmp/setenv.sh /opt/tomcat/bin/setenv.sh
sudo chmod +x /opt/tomcat/bin/setenv.sh

cat <<EOF > /opt/tomcat/conf/server.xml
<?xml version="1.0" encoding="UTF-8"?>
<Server port="8005" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.startup.VersionLoggerListener" />
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
 <Listener
 className="org.apache.catalina.mbeans.JmxRemoteLifecycleListener"
 rmiRegistryPortPlatform="8097"
 rmiServerPortPlatform="8098"
 />
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />
  <GlobalNamingResources>
    <Resource name="UserDatabase" auth="Container"
              type="org.apache.catalina.UserDatabase"
```

```
       description="User database that can be updated and saved"
       factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
       pathname="conf/tomcat-users.xml" />
 </GlobalNamingResources>
 <Service name="Catalina">
  <Connector port="8080" protocol="HTTP/1.1"
        connectionTimeout="20000"
        redirectPort="8443" />
  <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
  <Engine name="Catalina" defaultHost="localhost">
   <Realm className="org.apache.catalina.realm.LockOutRealm">
    <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
        resourceName="UserDatabase"/>
   </Realm>
   <Host name="localhost"  appBase="webapps"
      unpackWARs="true" autoDeploy="true">
    <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
        prefix="localhost_access_log" suffix=".txt"
        pattern="%h %l %u %t &quot;%r&quot; %s %b" />
   </Host>
  </Engine>
 </Service>
</Server>

EOF

sudo chown -hR tomcat:tomcat /opt/tomcat

echo 'create systemd unit'
cd /opt

cat <<EOF > /etc/systemd/system/tomcat.service
[Unit]
 Description=Apache Tomcat 9 Servlet Container
 After=syslog.target network.target
[Service]
 User=tomcat
 Group=tomcat
 Type=forking
 Environment=CATALINA_PID=/opt/tomcat/tomcat.pid
 Environment=CATALINA_HOME=/opt/tomcat
 Environment=CATALINA_BASE=/opt/tomcat
 ExecStart=/opt/tomcat/bin/startup.sh
 ExecStop=/opt/tomcat/bin/shutdown.sh
 Restart=on-failure
[Install]
 WantedBy=multi-user.target
EOF
```

```
sudo curl https://tomcat.apache.org/tomcat-7.0-doc/appdev/sample/sample.war --output /opt/tomcat/webapps/sample.war

sudo systemctl enable tomcat

sudo chown -hR tomcat:tomcat /opt/tomcat
sleep 3
sudo systemctl start tomcat

#############################################################################
#API settings
#############################################################################

cd /tmp

cat <<EOF > user.login.json
{
 "jsonrpc": "2.0",
 "method": "user.login",
 "params": {
  "user": "Admin",
  "password": "zabbix"
 },
 "id": 1
}
EOF

cat <<EOF > hgroup.json
{
 "jsonrpc": "2.0",
 "method": "hostgroup.get",
 "params": {
   "output": "extend",
   "filter": {
     "name": [
       "HGROUP"
     ]
   }
 },
 "auth": AUTHID,
 "id": 1
}
EOF

cat <<EOF > createhostgroup.json
{
 "jsonrpc": "2.0",
 "method": "hostgroup.create",
 "params": {
   "name": "HGROUP"
 },
 "auth": AUTHID,
 "id": 1
}
EOF

cat <<EOF > template.json
{
  "jsonrpc": "2.0",
  "method": "template.get",
  "params": {
    "output": "shorten",
    "filter": {
      "host": [
        "CUSTOMTEMP"
      ]
    }
  },
  "auth": AUTHID,
  "id": 1
}
EOF

cat <<EOF > createtemplate.json
{
  "jsonrpc": "2.0",
  "method": "template.create",
  "params": {
    "host": "CUSTOMTEMP",
    "groups": {
      "groupid": CREATETEMPLATE
    }
  },
  "auth": AUTHID,
  "id": 1
}
EOF

cat <<EOF > add.host.json
{
  "jsonrpc": "2.0",
  "method": "host.create",
  "params": {
    "host": "CLIENT",
    "interfaces": [
      {
        "type": 1,
        "main": 1,
        "useip": 1,
        "ip": "IPADDR",
        "dns": "",
        "port": "10050"
      }
    ],
    "groups": [
      {
        "groupid": SELGROUPHOST
      }
    ],
    "templates": [
      {
        "templateid": "CHECKTEMP"
      }
    ]
  },
  "auth": AUTHID,
  "id": 1
}
EOF

cat <<EOF > user.logout.json
{
 "jsonrpc": "2.0",
 "method": "user.logout",
 "params": [],
 "auth": AUTHID,
 "id": 1
}
EOF

header='Content-Type:application/json'
zabbixApiUrl="http://$zabbixServer/api_jsonrpc.php"

function exit_with_error() {
 echo '********************************'
 echo "$errorMessage"
 echo '--------------------------------'
 echo 'INPUT'
 echo '--------------------------------'
 echo "$json"
 echo '--------------------------------'
 echo 'OUTPUT'
 echo '--------------------------------'
 echo "$result"
 echo '********************************'
 exit 1
}

#-----------------------------------------------------
# Auth to zabbix
#-----------------------------------------------------
errorMessage='*ERROR* - Unable to get Zabbix authorization token'
json=`cat user.login.json`
json=${json/USERNAME/$zabbixUsername}
json=${json/PASSWORD/$zabbixPassword}
result=`curl --silent --show-error --insecure --header $header --data "$json" $zabbixApiUrl`
auth=`echo $result | jq '.result'`
if [ $auth == null ]; then exit_with_error; fi
echo "Login successful - Auth ID: $auth"

#-------------------------------------
#select custom group ot create it
#-------------------------------------

errorMessage="Can't select Host Group"
json=`cat hgroup.json`
json=${json/HGROUP/$hgroup}
json=${json/AUTHID/$auth}
#echo "$json"
result=`curl --silent --show-error --insecure --header $header --data "$json" $zabbixApiUrl`
selgrouphost=`echo $result | jq -r '.result | .[0] | .groupid'`
echo "1 step $selgrouphost"
if [ $selgrouphost == null ]
then
errorMessage="Can't greate Host Group"
json=`cat createhostgroup.json`
json=${json/HGROUP/$hgroup}
json=${json/AUTHID/$auth}
#echo "$json"
result=`curl --silent --show-error --insecure --header $header --data "$json" $zabbixApiUrl`
selgrouphost=$(echo "$result" | awk -F'"' '{print $10}')
#echo "$result"
echo "this is group number $selgrouphost"
if [ $selgrouphost == null ]; then exit_with_error; fi
echo "HostGroup was created - GROUPID: $selgrouphost"
fi

#-------------------------------------
#get custom templateip or create it
#
echo "Before create template $hgroup"
errorMessage="Can't Find Custom templateID"
json=`cat template.json`
json=${json/CUSTOMTEMP/$customtemp}
```

```
json=${json/AUTHID/$auth}
#echo "$json"
result=`curl --silent --show-error --insecure --header $header --data "$json" $zabbixApiUrl`
echo "$result"
checktemp=$(echo "$result" | awk -F'"' '{print $10}')
echo "check template iD $checktemp"
if [ -z "$checktemp" ]
then
errorMessage="Can't create Custom template"
json=`cat createtemplate.json`
json=${json/CUSTOMTEMP/$customtemp}
json=${json/CREATETEMPLATE/$selgrouphost}
json=${json/AUTHID/$auth}
echo "$json"
result=`curl --silent --show-error --insecure --header $header --data "$json" $zabbixApiUrl`
echo "template ID pri sozdanii: $result"
checktemp=$(echo "$result" | awk -F'"' '{print $10}')
fi

#------------------------
#Add host to group with template
#------------------------------

errorMessage="Unable to create host"
json=`cat add.host.json`
json=${json/CLIENT/$client}
json=${json/IPADDR/$ipaddr}
json=${json/SELGROUPHOST/$selgrouphost}
json=${json/CHECKTEMP/$checktemp}
json=${json/AUTHID/$auth}
#echo "$json"
result=`curl --silent --show-error --insecure --header $header --data "$json" $zabbixApiUrl`
hostresult=`echo $result | jq '.result'`
if [ $hostresult == null ]; then exit_with_error; fi
echo "add successful -groupids: $hostresult"

#-------------------------------------------------
# Logout of zabbix
#-------------------------------------------------
errorMessage='*ERROR* - Failed to logout'
json=`cat user.logout.json`
json=${json/AUTHID/$auth}
result=`curl --silent --show-error --insecure --header $header --data "$json" $zabbixApiUrl`
logout=`echo $result | jq '.result'`
if [ $logout == null ]; then exit_with_error; fi
echo 'Successfully logged out of Zabbix'
fi
```