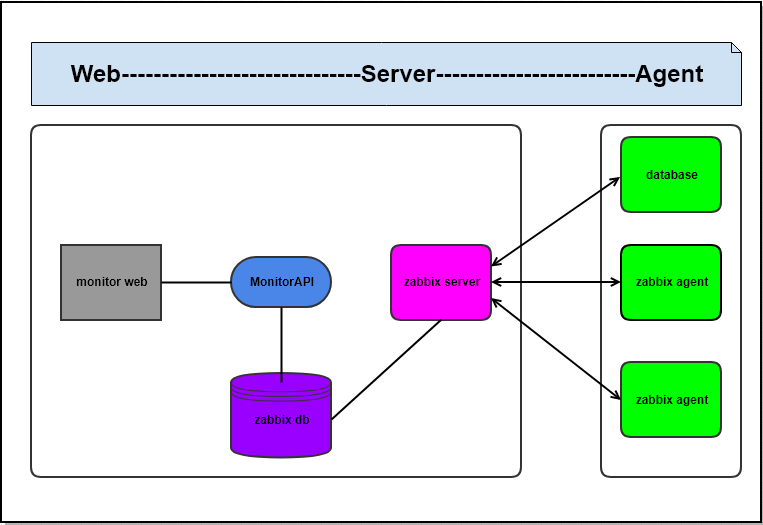
数据库监控

# 架构

采用客户端服务器架构。

监控服务端：为监控台，包括web，监控API和Zabbix Server和MySQL数据库部署节点。

监控客户端：为被监控对象，包括不限于数据库主机，其为Zabbix Agent部署节点。



# 安装

## Zabbix server

**基于CentOS Linux release 7.5.1804 (Core)**

cat /etc/resolv.conf

nameserver 119.29.29.29

nameserver 8.8.8.8

### 安装zabbix源

rpm -i <http://repo.zabbix.com/zabbix/3.2/rhel/7/x86_64/zabbix-release-3.2-1.el7.noarch.rpm>

**or**

yum install wget

wget <http://repo.zabbix.com/zabbix/3.2/rhel/7/x86_64/zabbix-release-3.2-1.el7.noarch.rpm>

rpm -ivh zabbix-release-3.2-1.el7.noarch.rpm

### 安装启动 mariadb数据库

yum install -y mariadb-server

systemctl enable mariadb.service

systemctl start mariadb.service

### 安装zabbix server mysql

yum -y install epel-release

yum -y install zabbix-server-mysql

yum install -y zabbix-web-mysql

systemctl enable zabbix-server.service

systemctl enable httpd.service

yum install zabbix-get

### 创建数据库并分配权限

mysql -e 'create database zabbix character set utf8 collate utf8\_bin;'

mysql -e 'grant all privileges on zabbix.\* to zabbix@localhost identified by "zabbix";'

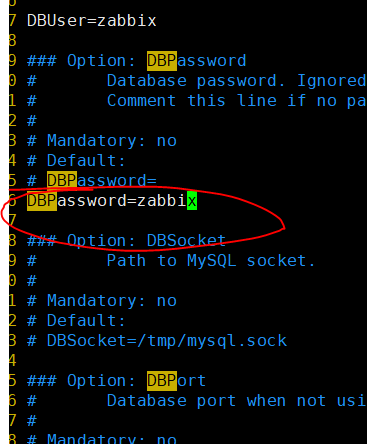
mysql -uzabbix -pzabbix

### 建立zabbix相关库表结构并导入数据

zcat /usr/share/doc/zabbix-server-mysql-3.2.11/create.sql.gz | mysql -uzabbix -pzabbix zabbix

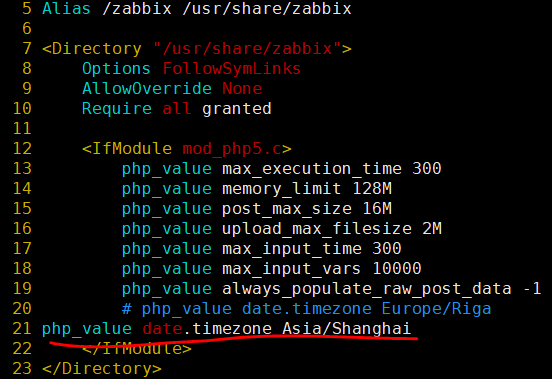
### 配置zabbix server 连接mysql数据库

sed -i.ori '126a DBPassword=zabbix' /etc/zabbix/zabbix\_server.conf



### 修改zabbix的时区

sed -i.ori '20a php\_value date.timezone Asia/Shanghai' /etc/httpd/conf.d/zabbix.conf



### 解决中文乱码

yum -y install wqy-microhei-fonts

cp /usr/share/fonts/wqy-microhei/wqy-microhei.ttc /usr/share/fonts/dejavu/DejaVuSans.ttf

cp: overwrite ‘/usr/share/fonts/dejavu/DejaVuSans.ttf’? y

### 设置开机自启动

systemctl enable mariadb.service

Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service to /usr/lib/systemd/system/mariadb.service.

systemctl enable httpd.service

Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.

systemctl enable zabbix- server.service

Created symlink from /etc/systemd/system/multi-user.target.wants/zabbix-server.service to /usr/lib/systemd/system/zabbix-server.service.

### 安装requests

pip install requests

### 2.2.8 开启Zabbix server

systemctl restart zabbix-server

~~http://192.168.1.131/zabbix/~~

~~username/password: Admin/Zabbix~~

## Zabbix agent

基于CentOS release 6.5

### ~~2.2.0 安装oracle server(测试oracle版本)~~

样例安装数据库服务端：Oracle11g XE.

Download oracle-xe-11.2.0-1.0.x86\_64(use my QQ account via oracle web)

yum install libaio bc flex

groupadd oinstall

groupadd dba

useradd -m -g oinstall -G dba oracle

id oracle

passwd oracle

->oracle

mkdir -p /u01/app

chown -R oracle:oinstall /u01/app

chmod -R 775 /u01/app

yum install libaio libaio-devel bc man net-tools -y

(--1G swap is not enough, but 2G.

Oracle Database 11g

Express Edition requires 1982 MB of swap space. This system has 1023 MB

of swap space

)

swapoff /swapfile

rm -rf /swapfile

dd if=/dev/zero of=/swapfile bs=1024 count=2197152

mkswap /swapfile

swapon /swapfile

cp /etc/fstab /etc/fstab.backup\_$(date +%N)

echo '/swapfile swap swap defaults 0 0' >> /etc/fstab

chown root:root /swapfile

chmod 0600 /swapfile

swapon -a

swapon -s

rpm -ivh oracle-xe-11.2.0-1.0.x86\_64.rpm

password set as input: sys

/etc/init.d/oracle-xe configure

su - oracle

修改.bash\_profile.在其中添加如下内容：

# Oracle Settings

TMP=/tmp; export TMP

TMPDIR=$TMP; export TMPDIR

ORACLE\_BASE=/u01/app/oracle; export ORACLE\_BASE

ORACLE\_HOME=$ORACLE\_BASE/product/11.2.0/xe; export ORACLE\_HOME

ORACLE\_SID=XE; export ORACLE\_SID

ORACLE\_TERM=xterm; export ORACLE\_TERM

PATH=/usr/sbin:$PATH; export PATH

PATH=$ORACLE\_HOME/bin:$PATH; export PATH

TNS\_ADMIN=$ORACLE\_HOME/network/admin

LD\_LIBRARY\_PATH=$ORACLE\_HOME/lib:/lib:/usr/lib; export LD\_LIBRARY\_PATH

CLASSPATH=$ORACLE\_HOME/jlib:$ORACLE\_HOME/rdbms/jlib; export CLASSPATH

sqlplus /nolog

conn / as sysdba

SQL> conn / as sysdba

Connected.

SQL> startup

SQL> show user

USER is "SYS"

SQL> show parameter instance\_name

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

instance\_name string XE

SQL>

SQL> select instance\_name from v$instance;

lsnrctl status

LSNRCTL for Linux: Version 11.2.0.2.0 - Production on 16-JUN-2019 00:18:38

Copyright (c) 1991, 2011, Oracle. All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC\_FOR\_XE)))

STATUS of the LISTENER

------------------------

Alias LISTENER

Version TNSLSNR for Linux: Version 11.2.0.2.0 - Production

Start Date 16-JUN-2019 00:05:40

Uptime 0 days 0 hr. 12 min. 58 sec

Trace Level off

Security ON: Local OS Authentication

SNMP OFF

Default Service XE

Listener Parameter File /u01/app/oracle/product/11.2.0/xe/network/admin/listener.ora

Listener Log File /u01/app/oracle/diag/tnslsnr/cfBareos/listener/alert/log.xml

Listening Endpoints Summary...

(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC\_FOR\_XE)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=cfBareos)(PORT=1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=cfBareos)(PORT=8080))(Presentation=HTTP)(Session=RAW))

Services Summary...

Service "PLSExtProc" has 1 instance(s).

Instance "PLSExtProc", status UNKNOWN, has 1 handler(s) for this service...

Service "XE" has 1 instance(s).

Instance "XE", status READY, has 1 handler(s) for this service...

Service "XEXDB" has 1 instance(s).

Instance "XE", status READY, has 1 handler(s) for this service...

The command completed successfully

[oracle@cfBareos ~]$ more /u01/app/oracle/product/11.2.0/xe/network/admin/listener.ora

# listener.ora Network Configuration File:

SID\_LIST\_LISTENER =

(SID\_LIST =

(SID\_DESC =

(SID\_NAME = PLSExtProc)

(ORACLE\_HOME = /u01/app/oracle/product/11.2.0/xe)

(PROGRAM = extproc)

)

)

LISTENER =

(DESCRIPTION\_LIST =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC\_FOR\_XE))

(ADDRESS = (PROTOCOL = TCP)(HOST = cfBareos)(PORT = 1521))

)

)

DEFAULT\_SERVICE\_LISTENER = (XE)

[oracle@cfBareos ~]$ more /u01/app/oracle/product/11.2.0/xe/network/admin/tnsnames.ora

# tnsnames.ora Network Configuration File:

XE =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = cfBareos)(PORT = 1521))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = XE)

)

)

EXTPROC\_CONNECTION\_DATA =

(DESCRIPTION =

(ADDRESS\_LIST =

(ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC\_FOR\_XE))

)

(CONNECT\_DATA =

(SID = PLSExtProc)

(PRESENTATION = RO)

)

)

lsnrctl start ： 启动监听器

lsnrctl stop ： 关闭

### 2.2.1 安装oracle client

--python access usage

<https://github.com/zhujinhe/collection-of-zabbix-templates/blob/master/zabbix_oracle/Pyora-master/src/oracle-instantclient11.2-basic-11.2.0.3.0-1.x86_64.rpm>

download oracle-instantclient11.2-basic-11.2.0.3.0-1.x86\_64.rpm

rpm -ivh oracle-instantclient11.2-basic-11.2.0.3.0-1.x86\_64.rpm

ln -s /usr/lib/oracle/11.2/client64/lib/libclntsh.so.11.1 /usr/lib/oracle/11.2/client64/lib/libclntsh.so

vi /etc/ld.so.conf.d/oracle-instantclient.conf

/usr/lib/oracle/11.2/client64/lib

ldconfig

### 2.2.2 安装Zabbix agent

--python access usage

http://repo.zabbix.com/zabbix/3.2/rhel/6/x86\_64/zabbix-agent-3.2.1-1.el6.x86\_64.rpm

rpm -ivh zabbix-agent-3.2.1-1.el6.x86\_64.rpm

<http://repo.zabbix.com/zabbix/3.2/rhel/6/x86_64/>zabbix-get-3.2.1-1.el6.x86\_64.rpm

rpm -ivh zabbix-get-3.2.1-1.el6.x86\_64.rpm

service zabbix-agent start

zabbix\_get -s 127.0.0.1 -k "mysql.version"

### 2.2.3 agent设置开机启动

如果是systemd的话:systemctl enable zabbix-agent.service

Created symlink from /etc/systemd/system/multi-user.target.wants/zabbix-agent.service to /usr/lib/systemd/system/zabbix-agent.service.

### 2.2.4 安装argparse

yum install -y python-argparse

### 2.2.5 安装cx\_Oracle

yum -y install epel-release

edit via vi /etc/yum.repos.d/epel.repo

[epel]

name=Extra Packages for Enterprise Linux 6 - $basearch

baseurl=http://download.fedoraproject.org/pub/epel/6/$basearch

#mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-6&arch=$basearch

failovermethod=priority

enabled=1

gpgcheck=1

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6

yum install python-pip

pip install cx\_Oracle

### 2.2.6 创建zabbix oracle用户

[root@rac2 script]# su - oracle

[oracle@rac2 ~]$ sqlplus /nolog

SQL\*Plus: Release 11.2.0.3.0 Production on Sat Aug 24 13:10:30 2019

Copyright (c) 1982, 2011, Oracle. All rights reserved.

SQL> conn /as sysdba;

Connected.

>

create user zabbix identified by "zabbix" default tablespace system temporary tablespace temp profile default account unlock;

GRANT CONNECT TO ZABBIX;

GRANT RESOURCE TO ZABBIX;

ALTER USER ZABBIX DEFAULT ROLE ALL;

GRANT SELECT ANY TABLE TO ZABBIX;

GRANT CREATE SESSION TO ZABBIX;

GRANT SELECT ANY DICTIONARY TO ZABBIX;

GRANT UNLIMITED TABLESPACE TO ZABBIX;

GRANT SELECT ANY DICTIONARY TO ZABBIX;

GRANT SELECT ON V\_$SESSION TO ZABBIX;

GRANT SELECT ON V\_$SYSTEM\_EVENT TO ZABBIX;

GRANT SELECT ON V\_$EVENT\_NAME TO ZABBIX;

GRANT SELECT ON V\_$RECOVERY\_FILE\_DEST TO ZABBIX;

### 2.2.7 agent配置

~~sed -i.ori 's#Server=127.0.0.1#Server=172.16.111.55#' /etc/zabbix/zabbix\_agentd.conf~~

vim /etc/zabbix/zabbix\_agentd.conf

Server=192.168.1.131

ServerActive=192.168.1.131

vi /etc/zabbix/zabbix\_agentd.conf

UserParameter=oracle.query[\*],python /home/cf/script/dbagent.py --username $1 --password $2 --address $3 --database $4 $5 $6 $7 $8

### 2.2.8 开启Zabbix agent

~~systemctl start zabbix-agent.service~~

service zabbix-agent restart

## 源代码

1. MonitorZ里的Oracle 数据获取文件: dbagent.py .

安装于指定目录，如/home/cf/script/dbagent.py被agent的zabbix\_agentd.conf指定.

1. MonitorZ里的其他代码作为前端代码的zabbix server库. 接口文件:monitorapi.py

## 启动zabbix server+agent

### 2.4.1 验证安装

**代理端:**

/etc/zabbix/zabbix\_agentd.conf

1. zabbix\_agentd.conf中的Server/ServerActive指向Zabbix Server的IP地址。

vim /etc/zabbix/zabbix\_agentd.conf

Server=192.168.1.131

ServerActive=192.168.1.131

1. zabbix\_agentd.conf中的UserParameter指向的dbagent.py为实际dbagent存放路径

UserParameter=oracle.query[\*],python /home/cf/script/dbagent.py --username $1 --password $2 --address $3 --database $4 $5 $6 $7 $8

service zabbix-agent restart

**服务端:**

systemctl restart zabbix-server

-- oracle.query[zabbix,Zabbix,Zabbix agent IP, 数据库名称, dbagent.py里的函数名, dbagent.py里的函数参数]

--用zabbix\_get工具 测试zabbix安装是否允许正常:

zabbix\_get -s 192.168.1.8 -k oracle.query[zabbix,zabbix,192.168.1.8,orcl,tablespace,SYSTEM]

example output: 3

zabbix\_get -s 192.168.1.8 -p 10050 -k oracle.query[zabbix,zabbix,192.168.1.8,orcl,currentscn]

example output: 2777996

# 3监控API

## 目录结构

.

├── agent

│   └── dbagent.py

├── centre

│   ├── configs.py

│   ├── configs.pyc

│   ├── import\_template.py

│   ├── \_\_init\_\_.py

│   ├── \_\_init\_\_.pyc

│   ├── jsonparams.py

│   ├── **monitorapi.py**

│   ├── monitorapi.pyc

│   ├── xml

│   │   └── monitorz\_template1.xml

│   ├── zabbixlib.py

│   └── zabbixlib.pyc

├── **main.py**

├── monitorZ.log

├── README.md

└── tests

├── dbagent.log.md

├── pyhome.sh

├── template11.2.xml.backup

└── test\_config.py

4 directories, 19 files

## 使用

web接口使用. /centre/monitorapi.py

使用方法(./main.py样例):

1. Import monitorapi.py
2. monitorAPI = MonitorAPI()，获取对象然后使用函数接口

## API

### 创建主机组hostgroup

参数:主机组名称：函数返回：主机组id

def hostgroup\_create(self, name):

def hostgroup\_get(self, name):

def hostgroup\_delete(self, name):

### 创建主机模板

参数:模板名称，关联主机组hostgroup名称：函数返回：模板id

def template\_create(self, name, hostgroup\_name):

def template\_get(self, name):

def template\_delete(self, name):

### 创建监控项

参数:监控项名称, key键如(oracle.query[zabbix,zabbix,192.168.1.8,orcl,version]),监控项数据类型,关联模板名称

函数返回：监控项id

# value\_type: 0 numeric float,1-character,2-log,3-numeric unsigned,4-text

def item\_create(self, name, key, value\_type, template\_name):

def item\_get(self, key):

def item\_delete(self, key):

### 创建基于模板的主机

参数:主机名，ip地址，端口，主机组，模板名称.

函数返回：主机id

def host\_create(self, host\_name, host\_ip, host\_port, hostgroup\_name, template\_name):

def host\_get(self, name):

def host\_delete(self, name):

def item\_get\_by\_host(self, host\_name):

### 获取数据项历史数据

参数:主机名称, key, key监控项数据类型

# value\_type: 0 numeric float,1-character,2-log,3-numeric unsigned,4-text

函数返回：监控项数据

def history\_get\_host(self, host\_name, key, value\_type):

def history\_get(self, key, value\_type):

def host\_get\_abnormal(self):

### 串联创建数据项关联主机组，模板

def transaction\_create\_item\_on\_template(self, hostgroup\_name,template\_name, item\_name, key, value\_type):

def hostmacro\_create(self, host\_name, varmacro, value):

def hostmacro\_get(self, host\_name, varmacro):

def hostmacro\_delete(self, host\_name, varmacro):

### 创建触发器

参数:触发器名称desc, 触发器主机: 如host712, 键key, 触发函数: last()，比较符: >0.

({host721:oracle.query[{$USERNAME},{$PASSWORD},{$ADDRESS},{$DATABASE},check\_active].last()}>0监控项数据类型,关联模板名称

函数返回：触发器id

def trigger\_create(self, desc, host, key, func, compareto):

def trigger\_get(self, host\_name, desc):

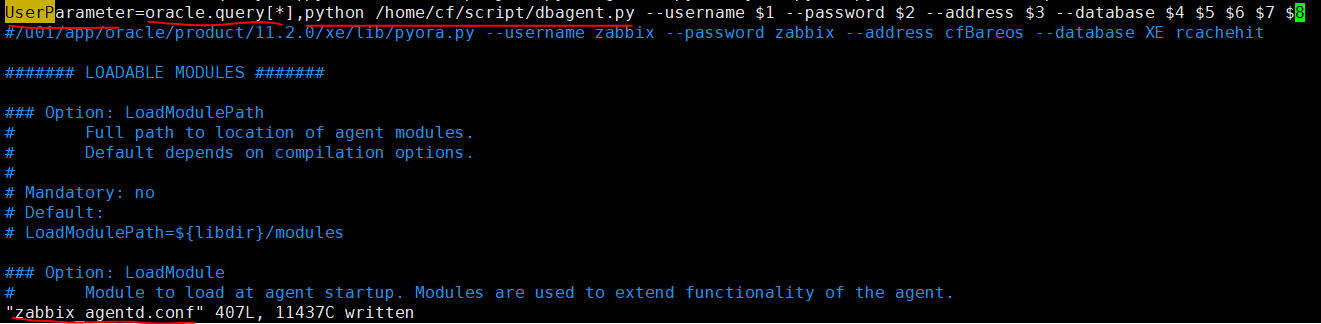
def trigger\_get\_problem\_by\_desc(self, host\_name, desc):

def trigger\_get\_problem\_by\_host(self, host\_name):

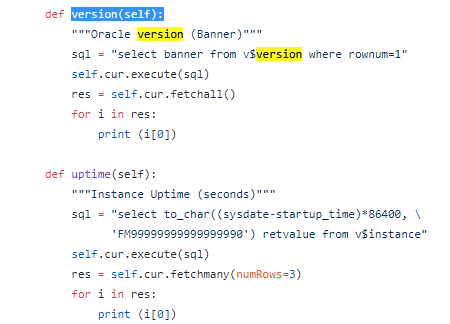
def trigger\_delete(self, host\_name, desc):

## 如何添加全新的oracle 数据 API

* Zabbix Agent基于/etc/zabbix/zabbix\_agentd.conf加载oracle数据接口脚本



* **Zabbix Agent的API如version添加（在dbagent.py）**



1. **Zabbix Server的API调用，添加version的数据项**



**创建过程参考main.py**