自有业务部署zabbix 4.2详细方案

一 服务器系统及配置,监控模式

Zabbix选择zabbix4.2

1 服务器系统及配置

zabbix-server

xxx.xx.xxx.xx(Ubuntu 18.04.3) 双核8G

zabbix-agent

xxx.xx.xxx.xx(Ubuntu 18.04.3) 4核16G

xxx.xx.xxx.xx(Ubuntu 16.04.2) 4核16G

xxx.xx.xxx.xx(Ubuntu 16.04.2) 4核16G

2 监控模式

由于zabbix-server空闲资源足够且被监控主机仅3台则选择默认监控模式（被动监控），也无需部署zabbix-proxy，也无需使用自动发现规则(被监控服务器规模小)

二 zabbix-server、zabbix-agent部署搭建

1 zabbix-server的部署

安装zabbix所需的PHP模块：

sudo apt install php7.2-bcmath php7.2-xml php7.2-mbstring

下载zabbix软件包

wget <https://mirrors.aliyun.com/zabbix/zabbix/4.2/ubuntu/pool/main/z/zabbix-release/zabbix-release_4.2-1%2bbionic_all.deb>

安装带有 MySQL 支持和 PHP 前端的 Zabbix 服务器

sudo apt-get install zabbix-server-mysql zabbix-frontend-php

配置mysql并导入初始表和数据

mysql> CREATE DATABASE zabbix CHARACTER SET utf8 COLLATE utf8\_bin;

mysql> GRANT ALL PRIVILEGES ON zabbix.\* TO zabbix@localhost IDENTIFIED BY 'usr\_strong\_pwd';

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

修改 /etc/zabbix/zabbis\_server.conf配置文件，分别写明数据库的用户密码及host即可

修改PHP配置/etc/zabbix/apache.conf

<IfModule mod\_php7.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 Asia/Shanghai

</IfModule>

2 zabbix-agent的部署

修改/etc/zabbix/zabbix\_agentd.conf配置文件

Server=172.18.105.61

ServerActive=172.18.105.61

启动apache并启动zabbix-server和agent后在前端页面做基础配置即可

/etc/init.d/apache2 apache2 restart

/etc/init.d/ zabbix-server start

/etc/init.d/ zabbix-agent start

三 zabbix监控策略（自带模版监控+自定义监控）

1 服务器的常规监控（如cpu、内存、文件系统等）

使用zabbix自带对应操作系统模版监控即可

2 自定义监控(redis和mysql是阿里云上买的服务无需监控)

（1）对流入、流出流量、TCP连接状况的监控

（2）对pv、service、error详情的监控

（3）服务器上微服务的工作状态的监控

（4）采集日志中接口的调用情况对业务的监控

四 zabbix调用外部接口告警

1 考虑平时工作软件采用钉钉机器人告警方式

1. 告警脚本（使用python脚本）

#!/usr/bin/python2

# -\*- coding: utf-8 -\*-

# Author: lauheng

import requests

import json

import sys

import os

headers = {'Content-Type': 'application/json;charset=utf-8'}

api\_url = "钉钉机器人接口url"

def msg(text):

json\_text= {

"msgtype": "text",

"at": {

"atMobiles": [

"想@的人的电话号码"

],

"isAtAll": False

},

"text": {

"content": "业务报警" + text

}

}

print requests.post(api\_url,json.dumps(json\_text),headers=headers).content

if \_\_name\_\_ == '\_\_main\_\_':

text = sys.argv[1]

msg(text)

2 对重要指标配置告警

对一些重要的监控指标配置触发器并关联告警动作实现监控告警

五 Zabbix数据库选择和后期优化

1 zabbix最大的瓶颈不在zabbix服务，而是mysql数据库的压力上，为了便于后期管理和优化不在本地搭建mysql，选择用RDS服务

2 zabbix数据库的优化

（1）定时查看表的占用情况（select table\_name, (data\_length+index\_length)/1024/1024 as total\_mb, table\_rows from information\_schema.tables where table\_schema='zabbix'; Zabbix一般的大表会有如下几个：history，history\_log，history\_str，history\_text，history\_uint，trends，trends\_uint

（2）zabbix的监控数据量随时间推移数据量会变大，且考虑数据安全性等方面的问题，所以选择innodb存储引擎（zabbix在innodb的性能比在myisam快）

（3）在允许的情况下，对具有较好离散度的列单独创建索引，这样可以提高该索引的使用弹性

（4）减少history保存时间，减少item获取间隔时间

（5）减少不必要的监控项

六 后期实现zabbix监控与业务相结合方案

后期完善python调用zabbix api接口抓取业务监控数据生产报表等，使监控与业务相结合