七载耕耘,全面盘点:Zabbix 实战文章精华大全分享

——整理 by 乐维社区

基础知识

一、Zabbix_sender 介绍及简单使用

1. zabbix_sender 是什么?有什么作用?

zabbix 获取 key 值有超时时间,如果自定义的 key 脚本一般需要执行很长时间,这根本没法去做监控,那怎么办呢?这时候就需要使用 zabbix 监控类型 zabbix trapper,配合zabbix_sender 给它传递数据。所以说 zabbix_sender 是更新 items 值最快的方式

2. zabbix_sender 命令解析

方法一:

zabbix_sender -z server -s host -k key -o value

方法二:

zabbix_sender -c config-file -k key -o value

方法三:

zabbix_sender -z server -i file

-c -- config zabbix_agent 配置文件绝对路径

-z --zabbix-server zabbix server 的 IP 地址

整理 by 乐维社区(https://forum.lwops.cn)

- -p --port zabbix server 端口.默认 10051
- -s --host 主机名,与 zabbix_server web 上主机的 hostname
- -I --source-address 源 IP
- -k --key 监控项的 key
- -o --value key 值
- -i --input-file 从文件里面读取 hostname、key、value 一行为一条数据,使用

空格作为分隔符,如果主机名带空格,那么请使用双引号包起来

- -r --real-time 将数据实时提交给服务器
- -v --verbose 详细模式, -vv 更详细

3. 案例

创建监控项

test	
Zabbix采集器 ~ 《 类型需要选采集器	
key.key	选择
_{数字(无正负)} ~ 自定义的key	
Do not keep history Storage period 90d	
Do not keep trends Storage period 365d	
不变	∨ 展示值映射
- 7. -	
	test Zabbix采集器 key key 数字 (无正负)

测试发送数据

zabbix_sender -z 192.168.1.12 -s test -k key.key -o 1234

查看是否可以正常获取数据

13.01主 E 成	坦 在北嶋入授業 机 <mark>isst X</mark> 在北嶋入授業 集	选择 选择 送择	会旅 重着无波料项目 ✓ 重着恐节 □			
▼ □ 主机 :	3称▲		最近检查记录	最新数据	更改	
▼ test	other - (1 监控项)					
	est		2022-09-29 18:48:01	1234		图形
					显示已自	动发现的 1中的1

测试可以正常获取到数据

二、浅谈 Zabbix 的四大监控方式

- Agent

顾名思义,也就是需要在被监控的操作系统安装 agent,通过 agent 和 server 端通讯传输 数据。

优点:

1、占用系统资源少,每个系统以200个监控项计算,平均占用资源不到0.5%,网络带宽 不到10k/s;

- 2、支持被动、主动的方式传输数据;
- 3、支持接收远程执行命令,可配置关联触发器执行某些进程、服务重启操作;
- 4、支持自定义监控键值,根据监控需求自定义键值、脚本获取某些监控数据;
- 5、保留监控日志,如监控报错可通日志排查。

缺点:

需要逐台安装配置,解决办法,可已下载免编译版本或者 rpm 安装包,编写安装脚本,一 般安装方法是使用命令安装后,修改对应的参数,通常只需要修改 Server、ServerActive、 Hostname,如需要自动注册则另外配置 Hostmetadata=system.name;后续可在 zabbix server 安装 nginx,使用 nginx 做文件服务器,直接在 linux 上使用命令安装即可, nginx 参考配置文章 利用 Nginx 实现免上传安装 zabbix agent_乐维_lwops 的博客-CSDN 博客。

二、 SNMP

SNMP 也就是简单网络管理协议 (Simple Network Management Protocol, 是专门设计 用于在 IP 网络管理网络节点 (服务器、工作站、路由器、交换机及 HUBS 等)的一种标 准协议,它是一种应用层协议);通常需要结合设备的 MIB(Management Information Base)文件使用。在网络设备,如交换机、路由器、防火墙、行为管理器、AP、AC、加密 设备、防毒墙等,以及物理服务器、存储等支持 snmp 功能的都可以通过开启 snmp 相关 的服务和配置 (不同设备的 snmp 开启方法,建议参考帮助文档或者官方文档操作),用 zabbix 进行简单的数据监控。

当然,操作系统也可以使用 snmp 方式监控,但只能是监控到系统层面的运行情况,如 cpu、 内存、系统分区、网络流量,支持自定义 oid,但是配置麻烦。

SNMP 优点:

1、适用性广, 主要应用在物理设备, snmp 使用的设备性能可以忽略不计;

2、配置简单,在管理页面操作几下或者执行6条左右命令即可配置;

3、数据简洁, snmp 功能的数据可读性比较简洁, 通常一个 oid 对应一个数据。

缺点:

1、设备太多,需要根据厂家的帮助手册来配置;

2、每个厂家的 mib 文件大多私有, 部分厂家不向外开放;

3、需要根据设备定制化 snmp 模板;

4、可监控性比较基础,基本上出厂内置,不易拓展性。

三、IPMI

IPMI (智能平台管理接口), Intelligent Platform Management Interface 的缩写。原 本是一种 Intel 架构的企业系统的周边设备所采用的一种工业标准。通常是在物理服务器、 存储设备最常见,用户可以通过 IPMI 进行设备的常规配置及管理,例如修改管理口 IP、修 改管理员账号信息、重启设备等操作,在 zabbix 还可以使用 IPMI 获取设备组件的运行状 态,如主板、电源、风扇、传感器的运行状态、运行温度等,一般是在只支持 IPMI 的情况 下使用,而且指标比较少,同一厂家不同型号之间的命令也不通用,用的比较少。

四、JMX

JMX(Java Management Extensions)是一个为应用程序植入管理功能的框架 ,从 Java5.0 开始引入到标准 Java 技术平台中。也就是只适用于 java 语言开发的中间件,像 Tomact、 Jboss、Resin、Weblogic、IBM WAS、ActiveMQ 等,国产的像中创、金蝶、东方通等

都支持使用 jmx 获取运行数据, 非国产的中间件多数可以在网络上找得到 jmx 的开启方式, 国产的一般需要厂家的支持。JMX 做为 java 类应用的一种监控方式, 能通过开启 jmx 功能 支持获取中间件的 jvm 运行状态、jvm 的内存池、线程池、老年代青年代的垃圾回收、节 点运行状态、会话数等信息。另外, JMX 配置都有固定的格式, 同时也支持加密方式, 可 以参考网上的案例或者官网文档自行配置。

三、zabbix 历史数据与趋势数据

历史数据(history)和趋势数据(trends)是 Zabbix 中存储收集到的数据的两种方式。 历史数据:

1: 每分钟收集到的监控数据

在正常项目中,一般不建议保留过长的历史数据,历史数据包含了太多信息,对数据库
 会造成较大的负载压力。

3: 可以选择保留较长的趋势数据, 来替代历史数据。

趋势数据:

1: 按小时统计计算后的平均数据

2: 趋势数据是一种 zabbix 内建的历史数据压缩机制,可以用来存储数字类型监控项的每

小时的最小值、最大值、平均值和记录数量。

3: 趋势数据设置的的留存时间应当比历史数据留存时间设置的长。

历史数据表

从 history_uint 表里面可以查询到设备监控项目的最大,最小和平均值,即存储监控数据的

原始数据。

history history_log history_str history_text history_uint

趋势数据表:

趋势数据是经过 Zabbix 计算的数据,数据是从 history_uint 里面汇总的,从 trends_uint 可以查看到监控数据每小时最大,最小和平均值,即存储监控数据的汇总数据。



历史数据、趋势数据修改:

可通过前端界面更改配置,具体如下:



四、更多.....

安全

十九、如何实现 kafka 与 zookeeper 的 SSL 认证

作者 乐维社区 (forum.lwops.cn/) 乐乐

在构建现代的分布式系统时,确保数据传输的安全性是至关重要的。Apache Kafka 和 Zookeeper 作为流行的分布式消息队列和协调服务,提供了 SSL (Secure Sockets Layer) 认证机制,以增强数据传输过程中的安全性。本文将详细探讨"如何实现 Kafka 与 Zookeeper 的 SSL 认证",从生成 SSL 证书到配置服务端和客户端的全过程,确保你的数 据在传输过程中得到充分的保护。无论你是系统管理员还是开发人员,了解并实施 SSL 认 证都是保障系统安全的关键步骤。接下来,我们将一步步引导你完成 Kafka 与 Zookeeper 的 SSL 认证配置,确保你的集群通信安全无忧。

一、配置 Kafka 账号密码:

1 、 首 先 , 需 要 修 改 kafka 配 置 文 件 : vim /asop/kafka/kafka 2.11-2.1.0/config/server.properties

broker.id=0

listeners=SASL_PLAINTEXT://:9092

advertised.listeners=SASL_PLAINTEXT://10.176.31.137:9092

num.network.threads=3

num.io.threads=8

socket.send.buffer.bytes=102400 socket.receive.buffer.bytes=102400 socket.request.max.bytes=104857600 log.dirs=/asop/kafka/logs num.partitions=1 num.recovery.threads.per.data.dir=1 offsets.topic.replication.factor=1 transaction.state.log.replication.factor=1 transaction.state.log.min.isr=1 log.retention.hours=168 log.segment.bytes=1073741824 log.retention.check.interval.ms=300000 zookeeper.connect=localhost:2181 zookeeper.connection.timeout.ms=6000 group.initial.rebalance.delay.ms=0 #使用的认证协议 security.inter.broker.protocol=SASL PLAINTEXT

#SASL 机制

sasl.enabled.mechanisms=PLAIN

sasl.mechanism.inter.broker.protocol=PLAIN

#完成身份验证的类

authorizer.class.name=kafka.security.auth.SimpleAclAuthorizer #如果没有找到 ACL(访问控制列表)配置,则允许任何操作。 allow.everyone.if.no.acl.found=false #需要开启设置超级管理员,设置 visitor 用户为超级管理员

super.users=User:visitor

2、其次,为 server 创建登录验证文件,可以根据自己爱好命名文件,如 vim /asop/kafka/kafka_2.11-2.1.0/config/kafka_server_jaas.conf , 文件内 容如下

KafkaServer {

org.apache.kafka.common.security.plain.PlainLoginModule required username="visitor" password="qaz@123" user_visitor="qaz@123";

};

3、然后修改 kafka 安装目录 vim

/asop/kafka/kafka_2.11-2.1.0/bin/kafka-server-start.sh, 在文件最上面添加 变量 export KAFKA_OPTS="

-Djava.security.auth.login.config=/asop/kafka/kafka_2.11-2.1.0/config/k

afka server jaas.conf"



4、接下来为 consumer 和 producer 创建登录验证文件,可以根据爱好命名

文件,如 kafka_client_jaas.conf,文件内容如下(如果是程序访问,如

springboot 访问,可以不配置)

vim /asop/kafka/kafka_2.11-2.1.0/config/kafka_client_jaas.conf

KafkaClient {

org.apache.kafka.common.security.plain.PlainLoginModule required

```
username="visitor"
```

```
password="qaz@123";
```

```
};
```

5、在 consumer.properties 和 producer.properties 里分别加上如下配置:

vim /asop/kafka/kafka_2.11-2.1.0/config/consumer.properties

vim /asop/kafka/kafka_2.11-2.1.0/config/producer.properties

security.protocol=SASL PLAINTEXT

sasl.mechanism=PLAIN

6、修改 kafka 安装目录 bin/kafka-console-producer.sh 和

bin/kafka-console-consumer.sh,在文件最上面添加变量

vim /asop/kafka/kafka_2.11-2.1.0/bin/kafka-console-producer.sh

vim /asop/kafka/kafka_2.11-2.1.0/bin/kafka-console-consumer.sh

export KAFKA_OPTS="

-Djava.security.auth.login.config=/asop/kafka/kafka_2.11-2.1.0/config/k afka client jaas.conf"



7、分别启动 zookeeper 和 kafka,至此服务端 kafka 用户登录验证配置完成(先

关闭 kafka 后关闭 zookeeper)

关闭服务 kafka

/asop/kafka/kafka_2.11-2.1.0/bin/kafka-server-stop.sh -daemon

/asop/kafka/kafka_2.11-2.1.0/config/server.properties

启动服务 kafka

#/asop/kafka/kafka_2.11-2.1.0/bin/kafka-server-start.sh -daemon

/asop/kafka/kafka 2.11-2.1.0/config/server.properties

关闭服务 zookeeper-3.4.13	
/asop/zk/zookeeper-3.4.13/bin/zkServer.sh	stop
/asop/zk/zookeeper-3.4.13/conf/zoo.cfg	
启动服务 zookeeper-3.4.13	
/asop/zk/zookeeper-3.4.13/bin/zkServer.sh	start
/asop/zk/zookeeper-3.4.13/conf/zoo.cfg	

8、创建及查看主题

/asop/kafka/kafka_2.11-2.1.0/bin/kafka-console-producer.sh --broker-list 10.176.31.137:9092 --topic cmdb --producer-property security.protocol=SASL_PLAINTEXT --producer-property sasl.mechanism=PLAIN [root@22CXSSC91176X31137 -]# /asop/kafka/kafka_2.11-2.1.0/bin/kafka-console-producer.sh --broker-list 10.176.31.137:9092 --topic cmdb
--test

接收消息

/asop/kafka/kafka_2.11-2.1.0/bin/kafka-console-consumer.sh --bootstrap-server

10.176.31.137:9092 --topic cmdb --from-beginning --consumer-property

security.protocol=SASL_PLAINTEXT --consumer-property sasl.mechanism=PLAIN

[root@zzZXXSSS176x31137 ~]# /asop/kafka/kafka_2.11-2.1.0/bin/kafka-console-consumer.sh --bootstrap-server 10.176.31.137:9092 --topic cmdb test _abcd

二、zk和 kafka 配置 ssal 账号密码:

1. Zookeeper 配置 SASL

1.1 新建 zoo_jaas.conf 文件

zoo_jaas.conf 文件名、文件所在路径没有特殊要求,一般放置在 \${ZOOKEEPER_HOME}/conf 目 录 下 vim

/asop/zk/zookeeper-3.4.13/conf/zoo_jaas.conf

Server {

org.apache.kafka.common.security.plain.PlainLoginModule required

username="admin"

password="admin@12"

user_kafka="kafka@123";

};

Server.username、Server.password为 Zookeeper 内部通信的用户名和密码,

因此保证每个 zk 节点该属性一致即可

Server.user_xxx 中 xxx 为自定义用户名,用于 zkClient 连接所使用的用户名和 密码,即为 kafka 创建的用户名

1.2 配置 /asop/zk/zookeeper-3.4.13/conf/zoo.cfg 文件

authProvider.1=org.apache.zookeeper.server.auth.SASLAuthenticationProvider

requireClientAuthScheme=sasl

jaasLoginRenew=3600000

zookeeper.sasl.client=true

zookeeper.sasl.client 设置为 true,开启客户端身份验证,否则 zoo_jaas.conf 中配置的 用户名将不起作用,客户端仍然可以无 jaas 文件连接,只是带有 WARNNING 而已

1.3 导入依赖包

因 为 使 用 的 权 限 验 证 类 为 : org.apache.kafka.common.security.plain.PlainLoginModule,所以需要 kafka 相关 jar 包,新建文件夹 zk_sasl_lib,如下:从kafka/lib目录下复制以下几个jar包到zookeeper 的 lib 和新建的 zk sasl lib 目录下:

kafka-clients-2.4.1.jar

lz4-java-1.6.0.jar

slf4j-api-1.7.28.jar

slf4j-log4j12-1.7.28.jar

snappy-java-1.1.7.3.jar

mkdir /asop/zk/zookeeper-3.4.13/zk_sasl_lib

cp /asop/kafka/kafka_2.11-2.1.0/libs/kafka-clients-2.1.0.jar

/asop/zk/zookeeper-3.4.13/lib/

cp /asop/kafka/kafka_2.11-2.1.0/libs/lz4-java-1.5.0.jar

/asop/zk/zookeeper-3.4.13/lib/

cp /asop/kafka/kafka_2.11-2.1.0/libs/slf4j-api-1.7.25.jar

/asop/zk/zookeeper-3.4.13/lib/

cp /asop/kafka/kafka_2.11-2.1.0/libs/slf4j-log4j12-1.7.25.jar

/asop/zk/zookeeper-3.4.13/lib/

cp /asop/kafka/kafka_2.11-2.1.0/libs/snappy-java-1.1.7.2.jar

/asop/zk/zookeeper-3.4.13/lib/

cp /asop/kafka/kafka_2.11-2.1.0/libs/kafka-clients-2.1.0.jar

/asop/zk/zookeeper-3.4.13/zk_sasl_lib

cp /asop/kafka/kafka_2.11-2.1.0/libs/lz4-java-1.5.0.jar

/asop/zk/zookeeper-3.4.13/zk_sasl_lib

cp /asop/kafka/kafka_2.11-2.1.0/libs/slf4j-api-1.7.25.jar

/asop/zk/zookeeper-3.4.13/zk_sasl_lib

cp /asop/kafka/kafka_2.11-2.1.0/libs/slf4j-log4j12-1.7.25.jar

/asop/zk/zookeeper-3.4.13/zk_sasl_lib

cp /asop/kafka/kafka_2.11-2.1.0/libs/snappy-java-1.1.7.2.jar

/asop/zk/zookeeper-3.4.13/zk_sasl_lib

chmod 755 -R /asop/zk/zookeeper-3.4.13/zk_sasl_lib/

chmod 755 -R /asop/zk/zookeeper-3.4.13/zk_sasl_lib/

1.4 修改 zkEnv.sh 文件/asop/zk/zookeeper-3.4.13/bin/zkEnv.sh

修改前:如果没有就直接加

export SERVER_JVMFLAGS="-Xmx\${ZK_SERVER_HEAP}m \$SERVER_JVMFLAGS"

修改后:

for jar in /asop/zk/zookeeper-3.4.13/zk_sasl_lib/*.jar;

do

```
CLASSPATH="$jar:$CLASSPATH"
```

done

export SERVER_JVMFLAGS="

-Djava.security.auth.login.config=/asop/zk/zookeeper-3.4.13/conf/zoo_jaas.conf "

重启 Zookeeper 服务即可

关闭服务 zookeeper-3.4.13	
/asop/zk/zookeeper-3.4.13/bin/zkServer.sh	stop
/asop/zk/zookeeper-3.4.13/conf/zoo.cfg	
启动服务 zookeeper-3.4.13	
/asop/zk/zookeeper-3.4.13/bin/zkServer.sh	start
/asop/zk/zookeeper-3.4.13/conf/zoo.cfg	

2. Kakfa 配置 SASL

2.1 新建 kafka_server_jaas.co	nf 文件		
kafka_server_jaas.conf	文件名和存放路径没	有要求,	一般放置在
\${KAFKA_HOME}/config	目	录	不
/asop/kafka/kafka_2.11-2.1.0/co	onfig/kafka_server_jaas.co	nf	

```
KafkaServer {
```

org.apache.kafka.common.security.plain.PlainLoginModule required

```
username="visitor"
```

```
password="qaz@123"
```

```
user_visitor="qaz@123";
```

};

Client{

org.apache.kafka.common.security.plain.PlainLoginModule required username="kafka" password="kafka@123"; };

KafkaServer.username、KafkaServer.password 为 broker 内部通信的用户名 密码,同上

KafkaServer.user_xxx 其中 xxx 必须和 KafkaServer.username 配置的用户名一致,密码也一致

KafkaServer.user_producer、KafkaServer.user_consumer 为了之后的 ACL 做准备, 达到消费者生产者使用不同账号且消费者账号只能消费数据, 生产者账号只能生产数据

Client.username、Client.password 填写 Zookeeper 中注册的账号密码,用于 broker 与 zk 的通信(若 zk 没有配置 SASL 可以忽略、若 zookeeper.sasl.client 为 false 也可以忽略只是带有,日志如下)

[2021-06-29 17:14:30,204] WARN SASL configuration failed:

javax.security.auth.login.LoginException: No JAAS configuration section named 'Client' was found in specified JAAS configuration file:

'/Users/wjun/env/kafka/config/kafka_server_jaas.conf'. Will continue connection to Zookeeper server without SASL authentication, if Zookeeper server allows it. (org.apache.zookeeper.ClientCnxn)

2.2 修改 server.properties 文件

broker.id=0

```
listeners=SASL PLAINTEXT://:9092
```

advertised.listeners=SASL_PLAINTEXT://192.168.157.198:9092

num.network.threads=3

num.io.threads=8

socket.send.buffer.bytes=102400

socket.receive.buffer.bytes=102400

socket.request.max.bytes=104857600

log.dirs=/asop/kafka/logs

num.partitions=1

num.recovery.threads.per.data.dir=1

offsets.topic.replication.factor=1

transaction.state.log.replication.factor=1

transaction.state.log.min.isr=1

log.retention.hours=168

log.segment.bytes=1073741824

log.retention.check.interval.ms=300000

zookeeper.connect=127.0.0.1:2181

zookeeper.connection.timeout.ms=6000

group.initial.rebalance.delay.ms=0

#使用的认证协议

security.inter.broker.protocol=SASL_PLAINTEXT

#SASL 机制

sasl.enabled.mechanisms=PLAIN

sasl.mechanism.inter.broker.protocol=PLAIN

#完成身份验证的类

authorizer.class.name=kafka.security.auth.SimpleAclAuthorizer

#如果没有找到 ACL (访问控制列表) 配置,则允许任何操作。

allow.everyone.if.no.acl.found=false

#需要开启设置超级管理员,设置 visitor 用户为超级管理员

super.users=User:visitor

其中 localhost 需要修改成 IP 地址

super.users 配置超级用户, 该用户不受之后的 ACL 配置影响

2.3 修改启动脚本

修改 kafka-server-start.sh 文件, 使之加载到 kafka_server_jaas.conf 文件 /asop/kafka/kafka_2.11-2.1.0/bin/kafka-server-start.sh

修改前:

```
if [ "x$KAFKA_HEAP_OPTS" = "x" ]; then
```

```
export KAFKA_HEAP_OPTS="-Xmx1G -Xms1G"
```

fi

修改后:

(先在首行加这一行,如果有了就不用加了) export KAFKA_OPTS="

-Djava.security.auth.login.config=/asop/kafka/kafka_2.11-2.1.0/config/kafka_server

_jaas.conf"

if ["x\$KAFKA_HEAP_OPTS" = "x"]; then

export KAFKA_HEAP_OPTS="-Xmx1G -Xms1G

-Djava.security.auth.login.config=/asop/kafka/kafka_2.11-2.1.0/config/kafka_server

_jaas.conf"

fi

设置 zookeeper 的 ACL 规则

/asop/zk/zookeeper-3.4.13/bin/zkCli.sh #进入 zk 的命令行模式

addauth digest admin:admin@12 #切换登陆用户(超级管理员是在 zk 的配置文 件/asop/zk/zookeeper-3.4.13/conf/zoo_jaas.conf 里面)

setAcl / ip:127.0.0.1:cdrwa,auth:kafka:kafka@123:cdrwa #(设置可以登陆的 IP 和用 户账号密码,admin 是上面的 zk 的配置文件里面定义的管理员,Kafka 用户是 /asop/kafka/kafka_2.11-2.1.0/config/kafka_server_jaas.conf 文件里面的定义的 kafka 连接 zk 的用户 (Client 下面的))

addauth digest kafka:kafka@123 #再切换为 kafka 用户再设置一次 acl setAcl / ip:127.0.0.1:cdrwa,auth:kafka:kafka@123:cdrwa



setAcl

ip:127.0.0.1:cdrwa,auth:kafka:kafka@123:cdrwa,auth:admin:admin@12:cdrwa,ip:1.1.

/

1.1

需要恢复权限,不设置 acl 的话就运行

setAcl / world:anyone:cdrwa

重启 kafka 服务即可

关闭服务 kafka

/asop/kafka/kafka_2.11-2.1.0/bin/kafka-server-stop.sh -daemon

/asop/kafka/kafka_2.11-2.1.0/config/server.properties

启动服务 kafka

#/asop/kafka/kafka_2.11-2.1.0/bin/kafka-server-start.sh -daemon

/asop/kafka/kafka_2.11-2.1.0/config/server.properties

至此,完成 kafka 与 zookeeper 配置 ssl 认证。大家好,我是乐乐专注 IT 运维技术研究与

分享,更多运维技巧欢迎关注乐维社区,更多运维问题也欢迎留言提问。

二十、如何在 Weblogic 环境中启动认证方式对接 Zabbix 监控

在 WebLogic Server 中,启动认证可用于确保只有经过授权的用户和系统能够访问 WebLogic Server 及其应用程序,通过合理配置认证提供者和安全领域,管理员可以有效 管理和控制用户访问。

本文将详细介绍如何在 Weblogic 环境中配置启动认证并对接 Zabbix 监控,通过设置 Weblogic 的相关配置文件、创建认证文件并配置监控部分,确保只有授权用户才能启动 Weblogic 服务器,并实现对其运行状态的实时监控。

这一实践将会增强 Weblogic 服务器的安全性并提高系统的运维效率。

1.配置及创建 weblogic 相关配置文件

1.1 环境

1.1.1 配置 weblogic 配置文件相关位置为 domain 下的 setDomainEnv.sh

[weblogic@zabbix-s	e <u>rver bin]\$ ls</u>	
nodemanager	setDomainEnv.sh	<pre>stopManagedWebLogic.sh</pre>
server_migration	<pre>startManagedWebLogic.sh</pre>	stopWebLogic.sh
service_migration	startWebLogic.sh	
[weblogic@zabbix_s	erver binl\$ pwd	
/home/weblogic/Ora	cle/Middleware/user_proje	cts/domains/base_domain/
bin		
[weblogic@zabbix s	erver bin]\$ <mark>-</mark>	
. , .		

相关修改如下:

JAVA_OPTIONS="\${JAVA_OPTIONS} -Djava.rmi.server.hostname=172.17.10.100"

JAVA_OPTIONS="\${JAVA_OPTIONS}

-Djavax.management.builder.initial=weblogic.management.jmx.mbeanserver.WLS

MBeanServerBuilder"

JAVA_OPTIONS="\${JAVA_OPTIONS} -Dcom.sun.management.jmxremote=true"

JAVA_OPTIONS="\${JAVA_OPTIONS}

-Dcom.sun.management.jmxremote.port=12345"

JAVA OPTIONS="\${JAVA OPTIONS}

-Dcom.sun.management.jmxremote.rmi.port=12345"

JAVA_OPTIONS="\${JAVA_OPTIONS} -Dcom.sun.management.jmxremote.ssl=false"

JAVA_OPTIONS="\${JAVA_OPTIONS}

-Dcom.sun.management.jmxremote.authenticate=true"

JAVA_OPTIONS="\${JAVA_OPTIONS}

-Dcom.sun.management.jmxremote.access.file=/home/weblogic/Oracle/Middlewa

re/user_projects/domains/base_domain/jmxremote.access"

JAVA_OPTIONS="\${JAVA_OPTIONS}

-Dcom.sun.management.jmxremote.password.file=/home/weblogic/Oracle/Middl

eware/user_projects/domains/base_domain/jmxremote.password"

export JAVA_OPTIONS

注: /home/weblogic/Oracle/Middleware/user_projects/domains/base_domain/位置

为自己定义文件的位置或文件所在位置, jmxremote.access、jmxremote.password 文

件有则使用,没有则创建。

1.1.2 创建认证文件(weblogic 账号、密码),修改相应权限。

[weblogic@zabbix-server base_domain]\$ cat jmxremote.access
weblogic readonly
[weblogic@zabbix-server base_domain]\$ cat jmxremote.password
weblogic weblogic@123

Chmod 600 jmxremote.*

 rw----- 1
 weblogic weblogic
 18
 4月
 24
 10:11
 jmxremote.access

 rw----- 1
 weblogic weblogic
 22
 4月
 24
 10:11
 jmxremote.password

注: access 文件前面的 weblogic 为用户,设为只读, password 文件前面为用户后面为

密码,权限修改必做,否则可能会发生错误无法启动!!!

1.1.3 启动 weblogic (我之前是停止的,按实际需求操作)

```
[weblogic@zabbix-server base domain]$ ls
autodeploy fileRealm.properties lib
                                               startWebLogic.sh
bin
            init-info
                                  security
            jmxremote.access
config
                                  servers
console-ext jmxremote.password
                                  shutdown.pv
[weblogic@zabbix-server base domain]$ ./startWebLogic.sh
Starting WLS with line:
/usr/local/java/bin/java -server -Xms256m -Xmx512m -XX:MaxPermSi
ze=256m -Dweblogic.Name=AdminServer -Djava.security.policy=/home/we
blogic/Oracle/Middleware/wlserver_10.3/server/lib/weblogic.policy
-Dweblogic.ProductionModeEnabled=true -da -Dplatform.home=/home/w
eblogic/Oracle/Middleware/wlserver_10.3 -Dwls.home=/home/weblogic/O
racle/Middleware/wlserver 10.3/server -Dweblogic.home=/home/weblogi
c/Oracle/Middleware/wlserver_10.3/server
                                       -Dweblogic.management.di
scover=true -Dwlw.iterativeDev=false -Dwlw.testConsole=false -Dwlw
.logErrorsToConsole=false -Dweblogic.ext.dirs=/home/weblogic/Oracle
/Middleware/patch wls1036/profiles/default/sysext manifest classpat
h:/home/weblogic/Oracle/Middleware/patch ocp371/profiles/default/sy
sext manifest classpath -Djava.rmi.server.hostname=172.17.10.100 -D
javax.management.builder.initial=weblogic.management.jmx.mbeanserve
r.WLSMBeanServerBuilder -Dcom.sun.management.jmxremote=true -Dcom.s
un.management.jmxremote.port=12345 -Dcom.sun.management.jmxremote.r
mi.port=12345 -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.ma
.access.file=/home/weblogic/Oracle/Middleware/user_projects/domains
/base_domain/jmxremote.access -Dcom.sun.management.jmxremote.passwo
rd.file=/home/weblogic/Oracle/Middleware/user_projects/domains/base
domain/jmxremote.password weblogic.Server
Java HotSpot(TM) 64-Bit Server VM warming: ignoring option MaxPermS
ize=256m; support was removed in 8.0 这里提示认证开启成功,并开始加载配置文件
```

Entor	NOOL THE	to boo		dic con	orwoh	logic		
Enter	nassword		Weblo	dic serv	/er.web	LUGIC	输入用户	名密码
2024		10 10					1007 107	
360>	<serve< td=""><td>r sta</td><td>rted</td><td>in RUN</td><td>INING</td><td>mode</td><td>> 启动</td><td>成功</td></serve<>	r sta	rted	in RUN	INING	mode	> 启动	成功

2. 监控部分配置

2.1 环境

2.1.1 如果已经监控过并触发更新数据,先执行清除操作,如果没有见 2.1.2

模 Int	板 nerited items 29 Not inher	rited iter	ns 22						
有	触发器								
	名称 *	股汉弱	發值	间隔	历史记录	编码	#2	秋志	标记
⊻	获取数据库执行的语句: 数据库buildInfo语句每小时执行次数		mongodb[buildInfo.total]		7d	365d	Zabbix采集器	已启用	Application: 数据库语
··· •	設取數据率执行的语句: 数据率buildInfo语句每小时执行的共政次数		mongodb[buildInfo,failed]		7d	365d	Zabbto采集器	已度用	Application: 数据库语
⊻	获取数据库执行的语句: 数据库delete语句每小时执行次数		mongodb[delete,total]		7d	365d	Zabbix采集器	已度用	Application: 数据库语
⊻	获取数据库执行的语句: 数据库delete语句每小时执行的失败次数		mongodb[delete,failed]		7d	365d	Zabbix采集器	已度用	Application: 数据库谱
··· 🖌	設取数据库执行的语句: 数据库endSessions语句每小时执行次数		mongodb[endSessions,total]		7d	365d	Zabbix采集器	已度用	Application: 数据库语
✓ …	較取數攝單执行的情句: 数攝庫endSessions语句每小时执行的失败次数		mongodb[endSessions,failed]		7d	365d	Zabbix采集器	已度用	Application: 数据库语
••••	获取数据库执行的语句:数据库find语句每小时执行次数		mongodb[find.total]		7d	365d	Zabbix采集器	已度用	Application: 数据库语
•	該取數攝率执行的语句: 数据率find语句每小时执行的失败次数		mongodb[find,failed]		7d	365d	Zabbix采集器	已度用	Application 数据库语
• •••	获取数据库执行的谱句:数据库isMaster语句每小时执行次数		mongodb[isMaster,total]		7d	365d	Zabbix采集器	已度用	Application: 数据库谱
••••	該取數攝率执行的语句: 数据率isMaster语句每小时执行的失败次数		mongodb[isMaster,failed]		7d	365d	Zabbix采集器	已度用	Application: 数据库语
•••	設設數攝率执行的语句: 数据率listIndexes语句每小时执行次数		mongodb[listIndexes,total]		7d	365d	Zabbix采集器	已度用	Application: 数据库语
•••	获取数据库执行的谱句: 数据库listIndexes语句每小时执行的失败次数		mongodb(listIndexes,failed)		7d	365d	Zabbix采集器	日月用	Application: 数据库语
¥	获取数据率执行的语句: 数据库replSetGetStatus语句每小时执行次数		mongodb[replSetGetStatus,total]		7d	365d	Zabbix采集器	已度用	Application: 数据车语
··· 🖌	設款數據庫执行的语句: 数据庫replSetGetStatus语句每小时执行的失败次数		mongodb[replSetGetStatus,failed]		7d	365d	Zabbix采集器	已度用	Application: 数据库语
···· 🗹	获取数据库执行的语句: 数据库sasiContinue语句每小时执行次数		mongodb[saslContinue,total]		7d	365d	Zabbix采集器	已度用	Application: 数据库语
•	获取数据库执行的语句:数据库sasiContinue语句每小时执行的失败次数		mongodb[saslContinue,failed]		7d	365d	Zabbix采集器	已度用	Application: 数据库语
•••	設款數据库执行的语句: 数据库satiStart语句每小时执行次数		mongodb[sasIStart,total]		7d	365d	Zabbix采集器	已度用	Application: 数据库语
✓ ····	获取数据库执行的谱句:数据库sasiStarti景句每小时执行的失败次数		mongodb[saslStart,failed]		7d	365d	Zabbix采集器	日月月	Application: 数据库谱
••••	获取数据库执行的语句: 数据库serverStatus语句每小时执行次数		mongodb[serverStatus,total]		7d	365d	Zabbix采集器	已病用	Application:数据库语
···· [获取数据库执行的语句:数据库serverStatus语句每小时执行的失败次数		mongodb[serverStatus.failed]		7d	365d	Zabbtx采集器	已病用	Application: 数据库语
···	获取数据库执行的语句: 数据库whatsmyuri语句每小时执行次数		mongodb[whatsmyuri,total]		7d	365d	Zabbix采集器	日月用	Application: 数据库谐
···· 🖌	获取数据率执行的语句:数据率whatsmyuri语句每小时执行的失政次数		mongodb[whatsmyuri,failed]		7d	365d	Zabbix采集器	已有用	Application 数据库语
									显示 已自动发现的
2 18.45	出用 第月 Execute now 清読历史 算制 松園東新 開設								

2.1.2 配置账号密码以修改模板宏方式配置 (账号密码为真实账号密码)

 * 模版名称 [Template Java JMX Genera] 可见的名称 中间件Java JMX环境模板[不迁移克隆通用] 模板 在此输入搜索 透择 * 群組 中间件模板 × 在此输入搜索 透择 描述 	板标记宏值映射		
可见的名称 中间件Java JMX环境模板[不迁移克隆通用] 模板 在此输入搜索 * 群組 中间件模板 × 在此输入搜索 选择 描述 「	* 模版:	名称 Template Java JMX General	
模板 在此输入搜索 选择 * 群組 中间件模板 × 在此输入搜索 选择 描述	可见的行	名称 中间件Java JMX环境模板[不迁移克隆通用]	
 * 群组 中间件模板 × 在比输入搜索 描述 	ं	莫板 在此输入搜索	选择
描述	*	詳組 中间件模板 ★ 在此输入搜索	选择
	1	苗述	

链接模板

瓦标记宏2值映射 			
* 模版名称	Template Java JMX General-clone		
可见的名称	中间件Java JMX环境模板克隆[不迁移克隆通	用]	
模板	名称	动作	
	中间件Weblogic模板[通用JMX]	取消链接 取消链接并清理	
	在此输入搜索		选择
* 群组	中间件模板 🗙		选择
	在此输入搜索		
描述			

相关的自动发现规则中 username、password 改为相应宏值

本文档为样章,完整版文档请添加乐乐 (lerwee) 获取

* 名称	JMX Garbage	e Collector(垃圾收集器)			
类型	JMX agent代	理程序 ≻			
键值	jmx.discovery	/[beans,":type=Garbag	eCollector,name=*"]		
点惹 XML*	service:jmx:n	mi:///jndi/rmi://{HOST.CC	NN}:{HOST.PORT}/jmxrmi		
用户名称	{\$JMX_USEF	RNAME}			
密码	{\$JMX_PASS	word}			
* 更新间隔	3600s				
自定义时间间隔	类型	间隔	期间	动作	
	灵活调	度 50s	1-7,00:00-24:00	移除	
	添加				
*资源周期不足	30d				

已启田 🔽

监控项原型也要改

监控项原型

所有模板 / 中间件Java JMX环境模板	克隆[不 自动发现清单 / JMX Garbage Collector(垃圾收集器) 监控项原型 3 触发器类型 图形原型 主机模板
监控项原型 标记 1 进程	
*名称	GC (#JMXNAME)垃圾收集器可用状态
类型	JMX agent代理程序 V
* 键值	jmx[(#JMXOBJ),Valid] 选择
信息类型	字符 ~
点識 XML *	service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi
用户名称	{\$JMX_USERNAME}
密码	{\$JMX_PASSWORD}
* 更新间隔	180
自定义时间间隔	类型 间隔 期间 动作 灵活 调度 50s 1-7,00:00-24:00 移除 添加
* 历史数据保留时长	Do not keep history Storage period 90d
值映射	在此输入搜索 选择

相关监控项批量修改

~	•••	已提交的非堆内存	触发器 1	jmx["java.lang:type=Memory",NonHeapMemoryUsage.committed]	180	90d	365d	JMX agent代理程序	已启用	Application: 堆内存
		当前JIT编译器的名称	触发器 1	jmx["java.lang:type=Compilation",Name]	180	90d		JMX agent代理程序	已启用	Application: 编译器
		总加载类计数		jmx["java.lang:type=ClassLoading",TotalLoadedClassCount]	180	90d	365d	JMX agent代理程序	已启用	Application: 类计数
	••••	总物理内存大小		jmx["java.lang:type=OperatingSystem",TotalPhysicalMemorySize]	180	90d	365d	JMX agent代理程序	已启用	Application: 系统
	••••	打开文件描述符计数	<u> 触发器</u> 1	jmx["java.lang:type=OperatingSystem",OpenFileDescriptorCount]	180	90d	365d	JMX agent代理程序	已启用	Application: 系统
		提交虚拟内存大小		jmx["java.lang:type=OperatingSystem",CommittedVirtualMemorySize]	180	90d	365d	JMX agent代理程序	已启用	Application: 系统
		最大文件描述符计数	触发器 1	jmx["java.lang:type=OperatingSystem",MaxFileDescriptorCount]	3600	90d	365d	JMX agent代理程序	已启用	Application: 系统
		活动守护线程的当前数目		jmx["java.lang:type=Threading",DaemonThreadCount]	180	90d	365d	JMX agent代理程序	已启用	Application: 线程
		活动线程的当前数目,包括守护线程和非守护线程		jmx["java.lang:type=Threading",ThreadCount]	180	90d	365d	JMX agent代理程序	已启用	Application: 线程
	••••	空闲物理内存大小		jmx["java.lang:type=OperatingSystem",FreePhysicalMemorySize]	180	90d	365d	JMX agent代理程序	已启用	Application: 系统
		系统CPU负载		jmx["java.lang:type=OperatingSystem",SystemCpuLoad]	180	90d	365d	JMX agent代理程序	已启用	Application: 系统
		系统名称		jmx["java.lang:type=OperatingSystem",Name]	3600	90d		JMX agent代理程序	已启用	Application: 系统
		系统平均负载		jmx["java.lang:type=OperatingSystem",SystemLoadAverage]	180	90d	365d	JMX agent代理程序	已启用	Application: 系统
		编译花费的总时间		jmx["java.lang:type=Compilation",TotalCompilationTime]	180	90d	365d	JMX agent代理程序	已启用	Application: 编译器
		自由交换空间大小		jmx["java.lang:type=OperatingSystem",FreeSwapSpaceSize]	180	90d	365d	JMX agent代理程序	已启用	Application: 系统
		进程CPU负载		jmx["java.lang:type=OperatingSystem",ProcessCpuLoad]	180	90d	365d	JMX agent代理程序	已启用	Application: 系统
•		非堆内存最大值	触发器 2	jmx["java.lang:type=Memory",NonHeapMemoryUsage.max]	180	90d	365d	JMX agent代理程序	已启用	Application: 堆内存
									周示 F	自动发现的 31中的31

31 选择 启用 禁用 复制 批量更新 删除

批量更新



模板 宏 继承模板的宏				
		· The user	of weblogic	移陸
			or weblogic	1916
{SJMX_PASSWORD}	18	1 v The pass	sword of weblogic	19PRA
添加				
添加取消				

所有主机 weblogic 已启用 JMX 自动发现清单 / JMX Garbage Collector(垃圾收集器) 监控项原型 3 触发器类型 图形原型 主机模板

×

主机						×
主机 IPMI 标记	宏2 资	≃记录● 加密 值映射				
* 主机名称	weblogic10	00				
可见的名称	weblogic			Í.		
模板	名称		动作用			
	中间件Webl	logic模板[通用JMX]	取消链接 取消链接并清理			
	在此输入搜	「家		选择		
* 群组	中间件模糊	ž ×		选择		
	在此输入想	家		-		
Interfaces	类型	IP地址	DNS名称	连接到	端口	默认
	JMX	172.17.10.100		IP DNS	5 12345	● 移除
	添加					
描述						
					更新	克隆全克隆删除取消

主机 IPMI 标记	宏 2 资产	记录 • 加密 值映射					×
* 主机名称	weblogic100)					-
可见的名称	weblogic						- 1
模板	中间件Java 在此输入搜	JMX环境模板克隆[不迁移克 索] [2] [2] [2] [2] [2] [2] [2] [2] [2] [2]	选择			- 1
* 群组	中间件模板在此输入搜索	× 家		选择			- 1
Interfaces	类型	IP地址	DNS名称	连接到	端口	默认	- 1
	JMX	172.17.10.100		IP DNS	12345	● 移除	- 1
	添加						
描述							
					更新	克隆全克隆	取消

修改监控资源



修改相应值为真实值

	录● 加密 值映射		
主加加 组织和CX主机	可用值值	模板值 T ~ Eii ← 中间件Java JMX环境模板克隆(不迁移克	全局值 (配置) 隆通用]: ""
The password of weblogic			
{\$JMX_USERNAME} The user of weblogic	值	T ~ 更改 ← 中间件Java JMX环境模板克隆[不迁移克	釜通用]. ™
{\$SNMP_COMMUNITY}	public	T、 更改	← "public"
description			
添加			
		更新 克隆	全克隆

等待数据采集,并查看监控结果:

ZABBIX « 🗉	最新数据												36 *
٩	< 7												< > n
		主机群组 在此输入搜索			选择		标记 与成	(默认) 或					
Contraction of the second seco		主机 在此输入搜索			洗择		标记		包含	× (iii	移除		
0.55		冬節 weblogic					源加						
±#1						屋	示标签 无	1 2 3 1	原签名 填满 /细	夏 无			
最新数据						标签显示	优先级 以逗号	分隔的列表					
拓扑圈						4	表明 竹 □						
自动发现													
Ô 1855 -		1980-152				Save as	应用重调	2					
	主机	192365											
	weblogic 8												
□ 报表 →	TAG VALUES												
ペ 配置	Application: JDBC 8												
🖸 管理 🗸	□ 主机	名称▲			像近	会查记录	新教園	更改	后海				信息
	weblogic	weblogic AdminServer.jm	s JMS 服务器总额	收	2m 2	1s 0			Application: JDBC	3		图形	
9 支持	weblogic	weblogic AdminServer.jm	s 当前 JMS 服务	器计数	2m 2	1s 0			Application: JDBC	3		图形	
E Integrations	weblogic	weblogic AdminServer.jm	s 当前连接计数		2m 2	1s 0			Application: JDBC	3		图形	
? 帮助	weblogic	weblogic AdminServer.jm	s 最大 JMS 服务	器数	2m 2	1s 0			Application: JDBC	1		图形	
💄 User settings 🗸 🗸	weblogic	weblogic AdminServer.jm	s 最大连接计数		2m 2	1s 0			Application: JDBC	3		图形	
di anni	weblogic	weblogic AdminServer Jm	5 连援总数		2m 2	15 0	A A A A A A A A A A A A A A A A A A A		Application: JDBC	3		IEINS	
	webiogic	webiogic Adminiserver #2	列油打扰型		2111 2	IS K	UNINING		Application: JDBC	3		00£	-
ZAPPIX // M	易新粉探												57
ZABBIX « 🗈	最新数据												
ZABBIX « ज्ञ ०	最新数据 < ▼												
ZABBIX 《 5 へ	最新数据 < ▼	主机器相 在此输入搜索			选择		标记 与成	(武法) 或					
ZABBIX ≪ № へ ◎ 胎測 へ 反衆版	最新数据 < ▼	主机群组 在此输入搜索 主机 在此输入搜索			选择		标记 与/或 标记	(373) 或	御 合	v (@	影致		• > (
ZABBIX ≪ 5 Q © 监測 へ (2未版 同题	最新数据 < V	 主机詳細 在此知入授家 主机 在此知入授家 名称 weblogc 			选择		标記 与/或 研記 孫記	(武化) 或	9a	v @	務談		• • • •
ZABBIX ≪ 5 Q © 监測 へ (な参板 问题 主机	最新数据 < ∇	主机時間 在成納入限家 主机 在成納入限家 主机 在成約入股家 客称 weblogic			选择		标記 与成 研记 添加 示标法 无	(数以) 或 1 2 3 4	包含	▼ 値	551k		* >: > >
ZABBIX ペ いいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいい	最新数据 < Y	主机時間 在点加入探索 主机 在点加入探索 名称 weblogic			选择	星	 标记 与/或 标记 通加 示标签 无 优先级 以道号; 	(第代从) 戦 1 2 3 1 回隔的別友	<u>協合</u> 示法者 (現) (19)	× ∈ 至 无	影陵		* > >
このまたは、 このまたは、 このまたは、 のまたは、	最新数据 < Y	主利期相 主利期相 主利 工芸加入理系 主利 工芸加入理系 名称 weblogic			选择	显标变量示	 标记 标记 添加 示标签 无 优先级 以進号/ 看细节 	(武化) 式 1 2 3 1 37編の列表	- 1933 新芸者 - 1133年 - 1899	> 値	新設		•
ZABBIX ≪ 5 ○ 1538 ペ 5 (公表版) 戸路 玉和、 老が数第 5日が数 の である。 での での の の の の の の の の の の の の の	最新数据	主机時間 在氏知入理家 主机 在氏知入理家 名称 weblogc			选择	日 一 一 一 一 一 一 一 一 一 一 一 一 一	 标记 与100 研え 源加 示标法 元 成先级 Q1里母/ 有細节 <th>(武化) 死 1 2 3 4 日前の列表</th><th>933 京芸者 1936 (1937</th><th>>) 値売</th><th>5200</th><th></th><th>••••••••••••••••••••••••••••••••••••••</th>	(武化) 死 1 2 3 4 日前の列表	933 京芸者 1936 (1937	>) 値売	5 200		••••••••••••••••••••••••••••••••••••••
このは、のでは、ので、ので、ので、ので、ので、ので、ので、ので、ので、ので、ので、ので、ので、	最新数据 < ▼ 过滤器 只影响过速后的	主机時相 在出知入理業 主机 在出知入理業 名称 weblogc			选择	显 就在显示 查 Save as	 新記 新記 新記 須辺 赤塚笠 无 氏先級 (以信号) 着磁街 マ 病用 第四 	(数以) 取 1 2 3 4 所知の列後 2	- 533 京委者 - 1516 (1837		65 00		••••••••••••••••••••••••••••••••••••••
こののでは、またのでは、またので、またので、またので、またので、またので、またので、またので、またので	最新数据 < V	主机時間 在止加入現象 主机 在止加入現象 名称 weblogc 変現			选择	星 報意显示 章 Save as	特記 与9歳 後記 預加 示特弦 え. え 代代現最 (2進号) 着磁世 ♥ 前用 単位	(数以) 取 1 2 3 1 例の列の	533 R258 1936 199		1688.		€€
このまたのでは、またので、またので、またので、またので、またので、またので、またので、またので	最新数据 < V 过滤器只影响过滤后的 ±机 weblogic 8	主10時頃 在止加入現象 主机 在止加入現象 名称 weblogic 酸胡			选择	星 報意望示 室 Sone as	 特記 毎記 様記 様記 様記 様記 意記 第二 (北京政 2018年) (北京政 2018年) (本) (************************************	(飲ん) 取 1 2 3 1 利用の利用の 2	1933年 第25名 1933年 1989	▼ (倍 ■ 无	553 8		• • • •
このまたのでは、またので、またので、またので、またので、またので、またので、またので、またので	最新数据	主切時相 石止加入原来 主机 石止加入原来 名称 weblogic 激振			选择	星 報志显示 童 Save as	新記	(\$KU) 52 1 2 3 1 3 March 13 2	95 REX 104 19	<) 億 	9528		• • • •
このまたのでは、またので、またので、またので、またので、またので、またので、またので、またので	最新数据	主切時相 石止加入原来 主机 石止加入原来 名称 weblogic 遊加			234	里 報志显示 室 Sare as	新記 - 546 - 552 - 555 - 5	(次心) 50 1 2 3 1 (昭石) 외국 2		▼ 億 夏 え	9522		€€ ~ > ()
このまたのでは、またいでは、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、	最新数据	主切時相 (元点知人現来) 主机 (元点知人現来) 名称 weblogic 遊却人現来 名称 weblogic	网络 历史	1377 3		型 税在型示 <u>室</u> Save as	 特記 第回 <li< th=""><th>(数化) 数 1 2 3 4 例 (部27) 対象</th><th>(9)合 時空名 (10)時 (19) 時日 (10) (10) (10) (10) (10) (10) (10) (10)</th><th>× 億 種 无</th><th>9502</th><th>972</th><th>★ > 0</th></li<>	(数化) 数 1 2 3 4 例 (部27) 対象	(9)合 時空名 (10)時 (19) 時日 (10) (10) (10) (10) (10) (10) (10) (10)	× 億 種 无	9502	972	★ > 0
マート ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	最新数据	主切時相 行业加入股票 主机 行业加入股票 名称 weblogic 380日 	10月 万丈 180 万丈	16375 S		日 将在至示 室 Save as	 特望 第9回 第回 <l< th=""><th>(BCU) \$2 1 2 3 4 98079796 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3</th><th> S名 S名 総合 1006 400 400</th><th></th><th>1950R</th><th>ESTRI-</th><th>€ €</th></l<>	(BCU) \$2 1 2 3 4 98079796 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3	 S名 S名 総合 1006 400 400		1950R	ESTRI-	€ €
このののののののののののののののののののののののののののののののののののの	最新数据	土切時田 石山か入田宗 土乳 石山か入田宗 主乳 石山か入田宗 名称 weblogic 名称 weblogic 超辺宗 と話。 を称 ann Server Jm. J. mel Com base Server Fluarism	1984 55史 180 7d	<u>新研</u> 3 3055d J 3055d J		日 秋志豊示 査 Save as 148年12月 1 1	 新記 新記 新記 第加 第加 第加 第二 <l< th=""><th>(ましし) れた 1 2 3 4 1 注 3 4 2 3 第8457月1日 2 2 2 2</th><th>SE SE SE Application JOB</th><th> ● ●</th><th>9520</th><th>國形。</th><th>(n.0.</th></l<>	(ましし) れた 1 2 3 4 1 注 3 4 2 3 第8457月1日 2 2 2 2	SE SE SE Application JOB	 ● ●	9520	國形。	(n.0.
このでは、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、	最新数据	主机研想 「正然新入現来」 主代 「正然新入現来」 主代 「正然新入現来」 を ない たの たの たの たの たの たの たの たの たの た	<i>різв.</i> 180 7. 180 7. 180 7.	超功 5 385d J 385d J	送鮮 ・読择 ・読择 ・読録 ・加 ・加 ・加 ・ ・	型 核差示 室 Save as	新記 新記 新記 第記 第記 第記 第記 第記 第記 第記 第記 第記 第	(また人) れた 1 2 3 4 3 日本 第二次 第二次	512 Fig. 2 Application JDB	 一 一 元 一 二 二		205 205 205	
このののでは、またでので、またでので、、、、、、、、、、、、、、、、、、、、、、、、、、、、	最新数据 く マ 対応器只影响过速にの 生形 weeklogic の TAG VALUES Application _2000 の 二まれ 4 weeklogic の 、 weeklogic の 、 weeklogic の 、 weeklogic の 、 weeklogic の 、 weeklogic の 、 weeklogic の 、 、 、 、 、 、 、 、 、 、 、 、 、	the second	100編 55年 1800 7d 1800 7d 1800 7d 1800 7d	10577 5 3055d J 3055d J 3055d J	12月 12月	臣 報查意示 室 Save as	 新記 新記 第記 第記 第記 第二 <li< th=""><th>(800) 82 1 2 3 1 1 2 3 1 3 3 1 3</th><th>52 52 52 52 622 Application: JDB Application: JDB Application: JDB</th><th></th><th>JARR</th><th></th><th></th></li<>	(800) 82 1 2 3 1 1 2 3 1 3	52 52 52 52 622 Application: JDB Application: JDB Application: JDB		JARR		

整理 by 乐维社区(https://forum.lwops.cn)

-			口的数据									*
2	ABRIX « 🛛	主机 www.hlunia.0										
	۹	webiogic o										
0	监测 ^	Application: JDBC 8										l
	仪表板		(7.2+	00.00	E.e.	+0.25	No. 171	AN CHARGE TH	and cashing	10.24		
	问题		weblasic AdminSequering 1	190	74	2654	IMY and	426	0	30.04	Application: IDBC	IEAAA
	±€L	weblogic	jmx["com.bea.ServerRuntim	100	74	3034	JIMA aye	425	0		Application: 5086	Latito
	最新数据	weblogic	weblogic AdminServer.jms	180	7d	365d	JMX age	42s	0		Application: JDBC	四开
	拓扑图		jmx["com.bea:ServerRuntim									
	自动发现	weblogic	jmx["com.bea:ServerRuntim	180	7d	365d	JMX age	42s	0		Application: JDBC	181Hs
Ō	服务 、	weblogic	weblogic AdminServer.jms jmx["com.bea:ServerRuntim	180	7d	365d	JMX age	42s	0		Application: JDBC	图形
=	资产记录 •	weblogic	weblogic AdminServer.jms jmx["com.bea:ServerRuntim	180	7d	365d	JMX age	42s	0		Application JDBC	图形
	报表 、	weblogic	weblogic AdminServer.jms	180	7d	365d	JMX age	42s	0		Application: JDBC	間用
	管理	weblogic	weblogic AdminServer 服务 jmx("com.bea:Name=Admin	180	7d		JMX age	42s	RUNNING		Application: JDBC	历史
9		weblogic	weblogic AdminServer 版本 jmx["com.bea:Name=Admin	180	7d		JMX age	42s	WebLogic Server		Application: JDBC	历史
12	Integrations											显示 已自动发现的 8中的8
-	mogradiona	0选择 显示堆磷数据	國家教授國									
?	帮助											
:	User settings v											
Ċ	退出							Zabbix 6.0.6.	© 2001–2022, Zabbix 5	IA		<1

到此,全部配置完成。大家好,我是乐乐,专注 IT 运维技术研究与分享,更多运维技巧及 运维问题欢迎关注乐维社区并留言~

二十一、如何通过 Zabbix Docker 配置 HTTPS 访问系统?

概述

前面文章曾介绍过<u>如果使用 docker-compose 快速部署一个 Zabbix 系统</u>,但是部署的 Zabbix 系统是使用 http 协议进行访问的。有时候为了保证安全。我们需要配置使用 https 协议进行访问。

下面就讲述如何使用自签名的 ssl 证书配置 https 访问。(注:若是有签发的证书,也可使 用配置,无须自己生成自签名证书。)

前提条件

本文主要讲述 Zabbix 官方镜像如何配置 https 访问,需为 Zabbix 官方提供的 zabbix docker 镜像来部署的 Zabbix 监控系统。

生成自签名证书

首先,先生成自签名证书。

这里提供一个快速生成证书的脚本,执行脚本需要输入一个 IP 的参数,然后会在脚本所在 目录下面生成名为 ssl.crt 的证书和 ssl.key 的密钥。
```
[rost@localhost ssl]# ls
ssl.sh
irrost@localhost ssl]# cat ssl.sh
irrost@localhost ssl]# ssl.sh
irrost@localho
```

```
#!/bin/bash
baseshell=$(cd $(dirname $0);pwd)
# 变量定义
IP=$1
# 生成证书配置文件
cat > ${baseshell}/ssl.cnf << EOF</pre>
distinguished_name = req_distinguished_name
x509_extensions = v3_req
prompt = no
[req_distinguished_name]
C = CN
ST = GD
L = GZ
0 = EDGE
OU = BASE
CN = \$IP
[v3_req]
keyUsage = critical, digitalSignature, keyAgreement
extendedKeyUsage = serverAuth
subjectAltName = @alt_names
[ alt_names ]
DNS.1 = \$IP
DNS.2 = 8.8.8.8
IP.1 = \$IP
EOF
. . .
```

生成 dhparam 证书

然后使用 openssl 生成 dhparam 证书。

openssl dhparam -out dhparam.pem 2048

	[root@localhost ssl]# openssl dhparam -out dhparam.pem 2048
	Generating DH parameters, 2048 bit long safe prime, generator 2
	This is going to take a long time
	+
	整理 /var/spool/mail/root 中滑影件
1	Trotietocathost Sst1= IS
	dinparam.pem ssn.sn sst.crt sst.key

配置证书

接下来就需要将 ssl 证书和 dhparam 证书一共挂载进 ZABBIX WEB 镜像里面, 随后重启即可生效。

● 若是 zabbix 系统使用 docker-compose 部署(可参考文章:<u>使用 docker-compose</u>

<u>部署 zabbix 监控系统</u>) ,则修改对应的 docker-compose.yaml 文件,如:

o docker-compose.yant inghtx.com	
[root@localhost zabbix]# cat docker-compose.yaml	
version: '3.7'	
services:	
postgresgl-6:	
container name: zabbix postgresgl 6 test	
image: postgres:14.4	
restart: alwavs	
environment:	
POSTGRES DB: zabbix	
POSTGRES PASSWORD: zabbix@2022	
POSTGRES USER: zabbix	
ports:	
- "45432:5432"	
volumes:	
- /6/data:/var/lib/nostgresgl/data	
- /6/ont:/ont	
zahhix-server-6:	
container name: zabbix server 6 test	
image: zabhiy/zabhiy.server.nggd:6 A 6-centos	
restart, alwaye	
environment:	
POSTCRES DR. Zabbiy	
POSTORES_DB. Zabbix02022	
POSTORES_IRED, zabbiy	
DD SEDVED LIGT, 102 160 2 141	
DD_SERVER_R031, 152,108,3,141	
DB_SERVER_FORT: 45452	
"40051,10051"	
volumoc.	
/6/zahbix/alortecripte//uer/lib/zabbix/alortecripte	
/6/zabbix/ateriscripts./usr/lib/zabbix/ateriscripts	
zabbix wob 6.	
container name, zabbix web 6 test	您证书 共 我 讲 wob 培 像 的 日
image, zabbix/zabbix web_gipx_pacal.6 0 6 ubuptu	
image: Zabbix/Zabbix-web-nginx-pgsqt:0.0.0-ubuntu	-
environment:	录
POSTGRES DB: Zabbix	
POSTGRES PASSWORD, zabbiyo2022	
POSTGRES LISER: Zabbix	
DB SERVER HOST: 102 168 3 141	
DB SERVER PORT: 45432	
78Y SERVER HOST: 102 168 2 141	
78X SERVER PORT: 40051	
parts:	
- "49090+9090"	
- "48443:8443"	
" /sc]./etc/sc]/nginy"	

修改完后,直接使用命令:

docker-compose up -d

重新创建容器即可生效

● 若是 zabbix 系统直接使用 docker 命令启动(可参考文章:如何使用 docker 快速部

<u>署 zabbix 监控系统</u>) ,则需重新创建 ZABBIX WEB 容器,如:

先删除旧容器,删除命令:

docker rm -f 容器名

然后重新创建容器命令:

docker run --name zabbix-web -e ZBX_SERVER_HOST=192.168.75.31 -e ZBX_SERVE

R_PORT=10051 -e DB_SERVER_HOST=192.168.75.31 -e DB_SERVER_PORT=3306

-e MYSQL_DATABASE=zabbix -e MYSQL_USER=root -e MYSQL_PASSWORD=zabbix
 -p 48080:8080 -p 48443:8443 -v /data/zabbix/ssl:/etc/ssl/nginx -d zabbix/zabbix-web-nginx-mysql:latest

挂载容器参数解释: -v ./ssl:/etc/ssl/nginx

-v —— 指启用卷挂载,格式:宿主机目录(文件)/容器目录(文件)

/data/zabbix/ssl:/etc/ssl/nginx —— 指, 将宿主机/data/zabbix/ssl 目录挂载到容器的

/etc/ssl/nginx 目录

访问测试

配置好,重新创建容器后,查看容器日志:

docker logs -f 容器名 (容器 ID)

可以看到,已经启用了 SSL



最后到浏览器进行访问测试:

https://IP:PORT

C

本文档为样章,完整版文档请添加乐乐 (lerwee)获取

如图,配置已完成。

大家好,我是乐乐,专注运维技术研究与分享,关注我,了解更多 Zabbix 知识。如有 Zabbix 问题也可以到<u>乐维社区</u>留言提问,共同探讨解决方案。

二十二、更多……

安装

二十七、龙蜥操作系统(Anolis OS) 安装 Zabbix 7.0 LTS 教程

龙蜥操作系统(Anolis OS)作为龙蜥社区发行的开源 Linux 发行版,以其稳定、高性能、 安全、可靠和 100%兼容 CentOS 8 软件生态的特点,成为众多企业和开发者的首选操作系 统。它不仅支持多计算架构,如 X86、ARM、RISC-V 等,还针对云端场景进行了优化,为 云上典型场景带来显著的性能提升和故障率降低。

本文将详细介绍如何在龙蜥操作系统(Anolis OS)上安装 Zabbix 7.0 LTS,帮助用户搭建 起一套高效、稳定的监控系统,实现对系统、网络、应用等各个方面的全面监控,为企业业 务的顺利运行提供有力保障。

概述

基于 yum 安装 Zabbix 7.0 LTS 版本,本次部署环境:

操作系统: Anolis OS 8.8 (x86 架构)

数据库: PostgreSQL 16.4

中间件: Nginx 1.14.1、PHP 8.0.30

Zabbix: zabbix server 7.0.3 、 zabbix agent 7.0.3

部署时, 需要联网访问 yum 源。

附 : Anolis 操 作 系 统 下 载 地 址 :

https://mirrors.aliyun.com/anolis/8.8/isos/GA/x86_64/AnolisOS-8.8-x86_64-dvd.iso

安装 Zabbix 7.0 LTS

安装 zabbix 的 yum 源: (如果是 arm 架构, 需要修改 zabbix 的源)

附: 自行查找对应操作系统版本 yum 源的方式为:

- 1、通过浏览器访问 https://repo.zabbix.com/zabbix/7.0/
- 2、逐层下钻查看对应操作系统版本,名字带 zabbix-release 的文件为 yum 源文件

curl	-0	zabbix-release-latest.el8.noarch.rpm -k
https://repo	.zabbix.com	/zabbix/7.0/rhel/8/x86_64/zabbix-release-latest.el8.noarch.
rpm		
rpm -Uvh za	bbix-release	e-latest.el8.noarch.rpm
dnf clean all		
dnf repolis	t	
[root@localhost -]# cui % Total % Received 100 32504 100 32504 [root@localhost -]# ls anaconda-ks.cfg zabbi] [root@localhost -]# rop 警告: zabbix-release-ld Verifying: #卷中 准备中 正在升级/安装 1:zabbix-release-7.([root@localhost -]# dni 51 个文年已刪解 [root@localhost -]# dni 6库 id AppStream Base05	rl -o zabbix-release-la d % Xferd Average Spet Dload Upto: 0 0 18426 (x-release-latest.el8.n n -Ubw zabbix-release- atest.el8.noarch.rpm: ########## 5-5.el8 ####### f clean all f repolist	test.el8.noarch.rpm -k https://repo.zabbix.com/zabbix/7.0/rhel/8/x86_64/zabbix-release-latest.el8.noarch.rpm d Time Time Current d Total Spent Lett Speed 0:00:01 0:00:01 -: 18436 arch.rpm atest.el8.noarch.rpm kv4 RSA/SNA512 Signature, 密明 ID b5333005: NOKEY ####################################
BaseUS DDE Extras HighAvailability Plus PowerTools kernel-5.10 zabbix zabbix-non-supported zabbix-tools		AnolisUS-8 - HaseUS AnolisOS-8 - DDE AnolisOS-8 - Extras AnolisOS-8 - Plus AnolisOS-8 - Plus AnolisOS-8 - PowerTools AnolisOS-8 - Kernel 5.10 Zabbix Official Repository - x86_64 Zabbix Official Repository (non-supported) - x86_64 Zabbix Official Repository (tools) - x86_64

整理 by 乐维社区(https://forum.lwops.cn)

切换 PHP 版本为 8.0:

dnf module switch-to php:8.0

如出现以下提示:



执行:

vi /etc/yum.repos.d/zabbix.repo

vi /etc/yum.repos.d/zabbix-tools.repo

给每个仓库源加上 sslverify=0

可选把 gpgcheck=1 改完 gpgcheck=0,安装时如提示验证不通过,则可修改

然后继续执行即可

[root@localhost ~]# dnf module Zabbix Official Repository - xf Zabbix Official Repository (nor Zabbix Official Repository (tor 依赖关系解决。	switch-to php:8.8 16_64 ⊡-supported) - x86_64 0ls) - x86_64			29 kB/s 74 kB 00: 790 B/s 1.4 kB 00: 905 B/s 1.6 kB 00:	92 91 91
软件包	架构	版本	仓库		大小
启用模块流: httpd nginx php		2.4 1.14 8.0			
事务概要					
确定吗? [y/N]; y 完毕! [root@localhost ~]#					

安装 Zabbix 服务端, Web 前端, 客户端

dnf	install	zabbix-server-pgsql	zabbix-web-pgsql	zabbix-nginx-conf
zabbi	x-sql-scrip	ots zabbix-selinux-policy	/ zabbix-agent	

驗证 : zabbix-web-dops-7.0.3-releasel.el8.noarch 验证 : zabbix-web-gogql-7.0.3-releasel.el8.noarch 验证 : fping-5.1-1.el8.x86_64		48/50 49/50 58/50
<pre>=== : : : : : ::::::::::::::::::::::::</pre>	de javu - fonts-common -2.35-7. an8. noarch fontconfig -2.13.1-4. an8. s86_64 pping 5.1 -1.48. s86_64 libbar - 1.6.9. smb. s86_64 libbar - 1.6.9. smb. s86_64 libbar - 1.6.9. smb. s86_64 libbar - 1.8.5. s8. s8. s66_64 libbar - 1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1	20/20
[root@cocatilost =]#		

安装 postgresql 数据库

参考官网安装步骤: https://www.postgresql.org/download/linux/redhat/

先安装 postgresql 源: (如果是 arm 架构, 需要修改 zabbix 的源)

dnf

install

-у

https://download.postgresql.org/pub/repos/yum/reporpms/EL-8-x86_64/pgdg-re

dhat-repo-latest.noarch.rpm

如果提示 SSL certificate problem: certificate is not yet valid 。同步本地的时间即可。

禁用系统内置 yum 源的 PostgreSQL 安装模块

dnf -qy module disable postgresql



安装 postgresql

初始化数据库并启动

/usr/pgsql-16/bin/postgresql-16-setup initdb systemctl enable postgresql-16 systemctl start postgresql-16



创建数据库: (创建数据库用户时, 自行输入密码并保存记录)

sudo -u postgres createuser --pwprompt zabbix sudo -u postgres createdb -O zabbix zabbix [root@localhost ~]# sudo -u postgres createuser --pwprompt zabbix 为新角色输入的口令: 再输入一遍: [root@localhost ~]# sudo -u postgres createdb -O zabbix zabbix [root@localhost ~]#

导入初始架构和数据

```
zcat /usr/share/zabbix-sql-scripts/postgresql/server.sql.gz | sudo -u zabbix psql
```

zabbix

INSERT	0	1
INSERT	0	1
DELETE	84	1865
COMMIT		
[root@]	loc	calhost ~]# 📒

为 Zabbix server 配置数据库

vi /etc/zabbix/zabbix_server.conf

DBPassword=创建数据库用户时,自行输入的密码

修改后,结果如下:

[root@localhost ~]# grep -Ev "^#|^\$" /etc/zabbix/zabbix_server.conf LogFile=/var/log/zabbix/zabbix_server.log LogFileSize=0 PidFile=/run/zabbix/zabbix_server.pid SocketDir=/run/zabbix DBName=zabbix DBName=zabbix DBUser=zabbix DBPassword=zabbix SNMPTrapperFile=/var/log/snmptrap/snmptrap.log Timeout=4 LogSlowQueries=3000 StatsAllowedIP=127.0.0.1 EnableGlobalScripts=0 [root@localhost ~]#

为 Zabbix 前端配置 PHP

vi /etc/nginx/conf.d/zabbix.conf

#取消注释第2行和第3行

server { listen 808 server_name exa

8080; example.com;

防火墙 和 SELinux

防火墙放通端口: (其他端口可以按需放通)

firewall-cmd --add-port=8080/tcp --permanent

firewall-cmd -reload

关闭 SELinux:

setenforce 0

vi /etc/sysconfig/selinux #开机不启动的设置

修改为 SELINUX=disabled

启动网页和 zabbix 服务

启动 nginx 和 php 服务:

systemctl start php-fpm nginx

systemctl status php-fpm nginx

ss -tnl #能看到 8080 端口

php-fpm.	service - The PHP	FastCGI Process Ma	nager		
Loaded:	loaded (/usr/lib,	/systemd/system/php	-fpm.service; disabled; ven	dor preset: disable	ed)
Active:	active (running)	since Tue 2024-09-	03 16:41:49 CST; 11s ago		
Main PID:	20373 (php-fpm)				
Status:	"Processes active	e: 0, idle: 10, Rec	uests: 0, slow: 0, Traffic:	0req/sec"	
Tasks:	11 (limit: 5710)				
Memory:	35.5M				
CGroup:	/system.slice/ph	p-fpm.service			
	-20373 php-fpm:	master process (/e	etc/php-fpm.conf)		
	—20379 php-fpm:	pool www			
	-20380 php-fpm:	pool www			
	-20381 php-fpm:	pool www			
	-20382 php-fpm:	pool www			
	-20383 php-fpm:	pool www			
	-20384 php-fpm:	pool zabbix			
	-20385 php-fpm:	pool zabbix			
	-20387 php-fpm:	pool zabbix			
	-20388 php-fpm:	pool zabbix			
	-20389 php-fpm:	pool zabbix			
9月 03 16: 9月 03 16: • nginx.se Loaded: Drop-In: Active: Process: Process: Process: Main PID: Tasks:	<pre>#1:49 localhost s #1:49 localhost s rvice - The nginx loaded (/usr/lib, /usr/lib/systemd, php-fpm.conf active (running) 20377 ExecStart=, 20376 ExecStartP 20374 ExecStartP 20386 (nginx) 3 (limit: 5710)</pre>	ystemd[1]: Starting ystemd[1]: Started HTTP and reverse p /systemd/system/ngi /system/nginx.servi since Tue 2024-09- /usr/sbin/nginx (cc re=/usr/sbin/nginx re=/usr/sbin/rm -f /	The PHP FastCGI Process Ma The PHP FastCGI Process Man proxy server inx.service; disabled; vendo ice.d 03 16:41:49 CST; 11s ago ode=exited, status=0/SUCCESS -t (code=exited, status=0/S 'run/nginx.pid (code=exited,	nager ager. r preset: disabled;) UCCESS) status=0/SUCCESS)	
[root@localhost ~]# s	s -tnl	<u>م</u>		Deen Address Deet	D
LISTEN	6 128	V.	0.0.0.0:22	0.0.0.0:*	Frucess

State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port	Process
LISTEN		128	0.0.0:22	0.0.0:*	
LISTEN		200	127.0.0.1:5432	0.0.0:*	
LISTEN			0.0.0:80	0.0.0.0:*	
LISTEN			0.0.0:8080	0.0.0.0:*	
LISTEN		128	[::]:22		
LISTEN		200	[::1]:5432		
LISTEN			[::]:80	[::]:*	
[root@localhost	~]#				

启动 zabbix 服务:

systemctl start zabbix-server zabbix-agent systemctl status zabbix-server zabbix-agent



可选: 查看 zabbix-server 启动后, 日志是否有报错

tail -n 100 /var/log/zabbix/zabbix_server.log

<pre>[root@localhost ~1# tail -r</pre>	100 /var/log/zabbix/zabbix server.log
20502:20240903:164637.214	Starting Zabbix Server, Zabbix 7.0.3 (revision d93ce022627).
20502:20240903:164637.214	****** Enabled features *****
20502:20240903:164637.214	SNMP monitoring: YES
20502:20240903:164637.214	IPMI monitoring: YES
20502:20240903:164637.214	Web monitoring: YES
20502:20240903:164637.214	VMware monitoring: YES
20502:20240903:164637.214	SMTP authentication: YES
20502:20240903:164637.214	ODBC: YES
20502:20240903:164637.214	SSH support: YES
20502:20240903:164637.214	IPv6 support: YES
20502:20240903:164637.214	TLS support: YES
20502:20240903:164637.214	*******************
20502:20240903:164637.214	using configuration file: /etc/zabbix/zabbix_server.conf
20502:20240903:164637.331	current database version (mandatory/optional): 07000000/07000003
20502:20240903:164637.331	required mandatory version: 07000000
20507:20240903:164637.336	starting HA manager
20507:20240903:164637.364	HA manager started in active mode
20502:20240903:164637.366	server #0 started [main process]
20509:20240903:164637.366	server #1 started [service manager #1]
20510:20240903:164637.367	server #2 started [configuration syncer #1]
20514:20240903:164637.563	server #3 started [alert manager #1]
20515:20240903:164637.567	server #4 started [alerter #1]
20518:20240903:164637.567	server #7 started [preprocessing manager #1]
20520:20240903:164637.568	server #9 started [lld worker #1]
20522:20240903:164637.569	server #11 started [housekeeper #1]
20524:20240903:164637.575	server #13 started [http poller #1]
20526:20240903:164637.580	server #15 started [discovery manager #1]
20528:20240903:164637.580	server #17 started [history syncer #2]

初始化 zabbix 页面

通过浏览器访问 http://IP:8080

根据页面提示的步骤对 zabbix 进行初始化。

	欢迎来到 Zabbis ^{默认语言 简体中文 (zh_CN)}	x 7.0)	题 下一步
	许可下 AGPLv3			
	Zabbix 7.0.3. © 2001–2024, <u>Z</u>	abbix SIA		
ZABBIX	检查必要条件	当前值	要求的	
欢迎	PHP版本	8.0.30	8.0.0	ок
检查必要条件	PHP选项"memory_limit"	128M	128M	ОК
设置	PHP选项"post_max_size"	16M	16M	ОК
安装前汇总	PHP选项"upload_max_filesize"	2M	2M	ок
安装	PHP选项"max_execution_time"	300	300	ОК
	PHP选项"max_input_time"	300	300	ок
	PHP支持的资料库	PostgreSQL		ОК
	PHP bcmath	于		OK
	PHP mbstring	于		ОК
	PHP选项"mbstring.func_overload"	关闭	关闭	ок
			返回	下—步
	许可下 AGPLv3			
	Zabbie 7.0.0 @ 0004_0004_7-55	siy SIA		
思捉实际情况修改季要	zabox 7.0.3.©2001-2024,Zabb	^{东文内} 左主机如里写成	1D +地+1トュ	T能会导致连续

数据库失败,需要对连接数据库的用户进行远程访问授权)

	数据库类型 Postar		`
欢迎			
检查必要条件	数据库主机 localho	ost	
配置数据库连接	数据库端口 5432	0 - 使用默认端口	
设直 安靖前汇兑	数据库名称 zabbix	(
安装	数据库概要		·
	将凭据存储在 文本	HashiCorp Vault CyberArk Vault	
	用户 zabbix		
	密码		
	数据库 TLS 加密 🗸		
	验证数据库证书		
			返回下一步
	许	F可下 AGPLv3	

Zabbix 主机名称 可为空,默认时区选择 UTC+8

ZABBIX	设置	
	Zabbix丰机名称	
欢迎		
检查必要条件	默认时区 (UTC+08:00) Asia/Shanghai	~
配置数据库连接	默认主题 蓝 🖌	
设置		
安装前汇总		
安装		
		返回 下一步
	许可下 AGPLv3	
	Zabbix 7.0.3. © 2001–2024, Zabbix SIA	

本文档为样章,完整版文档请添加乐乐 (lerwee)获取

ZABBIX	安装前汇总
	スイストリーノン
欢迎检查必要条件	数据库类型 PostgreSQL 数据库服务器 localhost
配置数据库连接 设置	数据库端口 5432 数据库名称 zabbix
安装前汇总 安装	数据库用户 zabbix 数据库密码 *****
3	数据库概要 数据库 TLS 加密 true
	返回
	许可下 AGPLv3
	Zabbix 7.0.3. © 2001–2024, Zabbix SIA

提示的前端配置文件较为重要,如后续需要调整前端连接的数据库信息等,可在改配置文件

ZABBIX	安装
欢迎 检查必要条件 配置数据库连接 设置 安装前汇总 安装	恩!Zabbix前端已经配置好了!千山万水第一步!加油吧, 少年! 配置文/# "etc/zabbix/web/zabbix.conf.php"已创建,
	许可下 AGPLv3
	Zabbix 7.0.3. © 2001–2024, Zabbix SIA

中进行调整。

整理 by 乐维社区(https://forum.lwops.cn)

默认用户名: Admin

默认密码:zabbix

;	▲ 不安全	192.	168.80.144:8080/index.php					\$
					ZABBIX			
				用户名称	尔	_		
				密码				
				✔ 30天	内记住我			
					登录			
					帮助・支持			
					© 2001–2024, Zabbix SIA			
Z	ABBIX «	2		vulues per second	zabbix前讀版本	7.0.3		1/
		Q			主机設量 (已启用/已禁用)	1 17	0	Shanghai
88	仪表盘				機板数量 监控项数量 (已启用/已禁用/不支持)	315	7/0/11	- · · · · · · · · · · · · · · · · · · ·
60	として	*	•		触发器数量 (已启用/已禁用 [问题/正常])	77 77	/ 0 [0 / 77] •	
ŧ	服务		主机可用性按严重性	的问题		◎ … 地理	地图	
¢	资产记录		1 0 0 0 1 0	0	0 0 0	0 +		∇
6	报表	~	可用 不可用 菜谷 未知的 台计 灭难	#± -	段/*重 古智 信息	未分类 -		
•	数据采集	~	Current problems					
4	告警	Ť	时间▼ 信息 主机 问題・严重性	期间	更新动作标识	2		
ă A	, HIP Marina	Ž						
201	- 昌旺							
				Q				
G			No d	ata found			9	
۲	集成							
0	帮助							Leaflet © OpenStreetMap contributors
౾	用户设置	•						
Ċ	退出			Zabbix 7.	0.3. @ 2001-2024, Zabbix SIA			5+···

二十八、轻松上手 | RockyLinux 9.4 安装及 Zabbix 7.0 部署教程详解

2024 年 6 月 30 日, CentOS 停止更新和维护, 不少企业用户开始寻求新的替代方案。 RockyLinux, 一个开源、社区拥有和管理、免费的企业 Linux 发行版, 提供强大的生产级 平台, 可作为 CentOS 停止维护(改为滚动更新的 Stream 版)后, RHEL 的下游 Linux 操 作系统替代方案, 并继承了原 CentOS 的开源免费特点。

本文介绍了 RockyLinux 9.4 安装以及在该环境中部署 Zabbix 7.0 的详细教程, 可帮助 原有的 CentOS 与 Zabbix 用户轻松上手并快速完成系统切换。以下是详细介绍:

RockyLinux 下载地址: <u>https://rockylinux.org/zh-CN/download</u>

版本	版本释义
DVD	完整版
Boot	自定义
Minimal	最小化

一、安装 RockyLinux9.4,选择典型或自定义都可以。



(1) 选择虚拟机操作系统。

新建虚拟机向导		×
选择客户机操作系统 此虚拟机中将安装哪种操	e作系统?	
客户机操作系统 ○ Microsoft Windows(<u>W</u>) ● Linux(L) ○ VMware ESX(X) ○ 其他(<u>O</u>)		
版本(⊻)		
Rocky Linux 64 位		~
帮助	< 上一步(<u>B</u>) 下一步(<u>N</u>) >	取消

(2) 根据电脑自身配置自行选择。

件 选项 设备 ② 内存 ④ 硬盘 (NVMe) ④ 硬盘 2 (NVMe) ④ 硬盘 2 (NVMe) ④ CD/DVD (IDE) ● 网络适配器 ● USB 控制器 ④ 声卡 ■ 显示器	摘要 8 GB 8 20 GB 正在使用文件 D:\ISO\Rocky-9. NAT 存在 自动检测 自动检测	内存 指定分配给此虚拟机的内存量。内存大小必须为 4 MB 的倍数。 此虚拟机的内存(M): 8192 → MB 128 GB - 64 GB - 32 GB - 16 GB - 8 GB - 4 GB - 4 GB - 2 GB - 128 GB - 2 GB - 128 GB - 2 GB - 128 MB - 64 MB - 32 MB - 16 MB - 8 MB - 4 MB -
	添加 (A) 移除(R)	

(3) 选择第一个安装 rocky Linux。



(4) 建议选择英语,一方面可以锻炼英语能力,另一方面会提高软件的兼容性,之后点击

continue.

Rocky Linux				ROCKY LINU	JX 9.4 帮助
3	欢迎使用 ROCKY LINUX 9.4。	建	议选择English		
the second second	您在安装过程中想使用哪种语言?				
1000 1	中文	Mandarin Chinese 🥇	简体中文 (中国)		
	العربية	Arabic	繁體中文 (台灣)		
5 m 5 m	English	English	繁體中文 (中華人民共和國香港特別行政區)		
Carl and	Français	French			
	Deutsch	German			
	日本語	Japanese			
A LE	Русский	Russian			
and the second	Español	Spanish			
	Afrikaans	Afrikaans			
State Luca	አ <i>ማር</i> ኛ	Amharic			
11.	অসমীয়া	Assamese			
	Asturianu	Asturian			
	Беларуская	Belarusian			
	Български	Bulaarian			
	(1	A			

(5) 设置完之后点击 begin installation。

Rocky Linux	INSTALLATION SUMMARY			
		选择安	装磁盘	
	LOCALIZATION	SOFTWARE	SYSTEM	
	Keyboard English (US)	Installation Source Local media	Installation Destination No disks selected	
	Language Support English (United States)	Software Selection Server with GUI	KDUMP Kdump is enabled	
	S Time & Date Asia/Shanghai timezone		Network & Host Name Connected: ens160	
and the	USER SETTINGS		Security Profile No profile selected	
	Root Password Root account is disabled			
	User Creation No user will be created	root密码设置		
Star.				

(6) 至此系统安装完成。

Rocky Linux	INSTALLATION PROGRESS	ROCKY LINUX 9.4 ■ us	INSTALLATION
	C Downloading packages		
to de la president		Quit	Reboot System

Rocky Linux	INSTALLATION PROGRESS	ROCKY LINUX 9.4 INSTALLATION
A		
A A		
	Completel	
1 2 2		
100		
		Rocky Linux is now successfully installed and ready for you to use!
		Go ahead and reboot your system to start using it! Reboot System
a la	Use of this product is subject to the license agreement found at /usr/share/rocky-release/EULA	

二、配置 IP 地址,子网,网关信息:

注 : 以 往 的 Redhat/Centos 配 置 lp 等 信 息 都 是 在 /etc/sysconfig/network-scripts/ifcfg-eth33 或者是 ifcfg-eth0, Rocky Linux 不同在 /etc/sysconfig/network-scripts 只有个 readme-ifcfg-rh.txt 文件里面会指引配置 lp 地址 的路径信息。



(1) 根据指引来到 /etc/NetworkManager/system-connections 下面有一个

ens160.nmconnection 文件,这里就是配置 IP 地址等信息的位置。

[root@localhost system-connections]# /etc/NetworkManager/system-connection	pwd Is	
[root@localhost sustem-connections]#		
[root@localhost system-connections]#		
<pre>[root@localhost system-connections]#</pre>		
<pre>[root@localhost system-connections]#</pre>	ls	
ens160.nmconnection		
<pre>[root@localhost system-connections]#</pre>		
<pre>[root@localhost system-connections]#</pre>		
<pre>[root@localhost system-connections]#</pre>	_	

(2) 根据情况自行修改。

初始配置	修改后配置	备注
[ipv4]	[ipv4]	\$#method=auto 是
method=auto	method=manual	DHCP 自动获取 IP 地
	address=192.168.10.94/24,192.168.10.2	址方式。
	dns=223.5.5.5	

may-fail=false	\$#method=manual
	是手动配置 IP 地址方
	式。
	\$#Address=Ip/ 子
	网,【英文逗号】网关
	地址
	\$#dns=阿里的 dns
	地址【114 或 8 都可
	以]
	\$# may-fail=false
	如果遇到任何错误或
	失败,系统不会忽略
	这些错误,而是会停
	止配置过程并报告错
	误。

<pre>uiii 19/4/Elka 8/3/-300-301/-613046/285 type=ethena==nst00 time=trans==nst00 time=trans==nst0 time=trans==nst00 time=trans==nst00 time=trans==nst00 tim</pre>
uiui d-194/eB/B-d-280-dB/D-eB/J 4e/J 3Bab (285 uiuje esthernet autocommect-priority=990 interfact-amerensia0 timestamp=1724090515 Tethernet1 (jpvd) method-manual address=122, 15, 5, 5 mu_failed fipvd) method-atoto [jpvd]
uii 14 14/e184 47.43 - 366 - 4817 - e1.13 48.61 c285 uipe = there net uii connect - priori Uj=-993 inter fact - maxe=casi L60 timest connect - priori Uj=-993 [inpot] method=manual address=122, 15, 5, 5 may=fail=14 18 ac [inpot] [inpot] oddress=122, 163, 18, 94/24, 192, 168, 19, 2 dim=set mod=eut i64 method=auto [inpot] [inpos] address=122, 163, 18, 94/24, 192, 168, 19, 2 dim=set mod=eut i64 method=auto [inpos]
unit=194/e184-er/13-dbid-val1/-e113dbid/c285 type=themet autoconnect-prior typ=-993 interface-maxeems108 timestamp=1724090151 Tethernet1 (ipvd) method=manol address=122,165.18, 94-24, 192, 160.18, 2 dmg=ren_mode=auti64 method=auto [ipvd] -
unid=194/e184-er/13-3086-4917-e113086fc285 type=ethermet autocommet-priority==993 interface=nmme:ens108 timestamp=172740505151 tethernet1 (ipv6) address=223,5,5,5,5 muy-fail=false (ipv6) addr-gen=mode=eni64 method=uto [proxy]
unital 1947e/184-0708-000-0017-ebi38a6c205 type=ethernet autocommet-prior ity=-999 interface-manse:emi60 timestam=1724090151 [ethernet] [ipv4] method=mankal address=192.100.100.94/24.192.160.10.2 dim=222.5.5.5 muj=7a11=fa1se [ipv6] addr-gen=mode=eui64 method=auto [uroxy]
unital 1947e/184-8708-3806-3917-eb138afc285 type=therenet autoconnect-priority=-393 inter1ace-maxeems160 timestamp=1724990151 Iethernet1 Iipod1 methodsannal adversg2:165:16:0.99724,192.168.18.2 dms=23:55:5.5 msgrfail=false Iipod1 addr-gen-mode=eui64 methodsauto [proxg]
unital 1947/plia-8/38-3806-297-20138abfc285 type=ethernet autocommect-priority=-999 interface-name=ensi60 timestamp=1724696151 [ethernet] [ijpv1] method=manual address=192.168.18.94/24,192.168.18.2 dins=223.5.5.5 mourfail=false [ijpv6] addr-ger=mode=eu164 method=auto [proxg]]
unital 1947/e10a-8/38-3806-297-e5138abfc285 tipe=ethernet autocommect-priority=-999 interface-name=mai508 timestamp=1724696151 [ethernet] [ijv01] method=manual addregen=mode=eui64 method=auto [proxg]
unital 1947/e10a-04730-2010-2017-20130abf c205 tupe=ethernet autocommect-priority=-999 interface=name=ensi60 timestamp=1724696151 [ethernet] [ipv4] method=annul address=192.160.10.94/24,192.160.10.2 dms=223.5.5.5.5 may=fail=false [ipv6] addr-gen=mode=eusi64 method=auto [iproxy]
uuta1947k18a-47.48-3666-4917-461.38a6fc285 type=ctbernet utcoconnect-priority=-993 interface-name=ensi60 timestamp=1724696151 [tipv4] method=mannal address=192.160.18.94/24,192.168.18.2 dns=223.55.55 muyf-aii-faise [ipv6] addr-gen-mode=eui64 method=auto [rroxg]] -
uuta ¹ 947k ¹⁰⁶ d ² /d ² -d ⁰ /d ² -d ¹ /d ² -d ¹ /d ²
urid=1947e18a-dr2a-306a-d817-e8138a6fc285 tipge=thermet autocommect-priority==993 interface-name=ensi60 timestam=1724698151 [ethermet] [ipv4] method=manual address=192.168.10.94/24,192.168.10.2 din=223.5,5.5 mug=fail=faise [ipv6] didr=gen=mode=eui64 method=auto [proxg]
ur 14 194 76 18 - 47.4 - 36 06 - 431 7 - 61.38 06 f c 285 type = c there met autoconnect - pri or i ty = -999 inter 1 ac- name = ms.16 0 t imestamp = 172 46 96 151 [rether name an 1 1 address = 132.168.18.24 dms=223.55.5 muy-fail=false [yrosy] [yrosy]
unital 1947/e10a-8/30-3040-297/-e113046/c205 type=ethernet autocommect-priority=-999 interface-name=ensi60 timestamp=1724696151 [ethernet] [ipv4] method=annul address=192.168.10.94/24,192.160.10.2 dms=223.5.5.5.5 may=fail=false [ipv6] addr-gen=mode=eui64 method=auto [proxy]
uurid1997/2084-07.08-30506-2017-e0138a6fc285 tige=cthermet autocommect-prior ity==999 interface=name=ens160 timestamp=1724696151 [ethermet] [ipv01 method=manual address=192.168.10.94/24,192.168.18.2 dms=223.5.5.5 may=fail=false [ipv6] address=ui64 method=auto [proxg]]
uu 14 1947618-47.48-3066-4917-61.3846f c285 type=etherment autocommect-prior ity==999 inter1ac=-name=ens160 timestamp=1724696151 [ethermet] [ijpv4] method=amual address=192.168.18.94/24,192.168.18.2 dms=223.55.5 may=Fail=false [ijpv6] addr-gen=mode=eui64 method=auto [ijrvog]
uu 14 1947618-47.48-3066-4917-61.3846f c285 type=etherment autocommect-prior ity=-993 interface-name=ensi60 timestamp=1724696151 IethernetJ Iipv41 method=manual address=192.168.10.94/24,192.168.10.2 din=223.5.5.5 may=7 all=false Iipv61 addr-gen-mode=eui64 method=auto Iproxy]
uuta1947e18a47c8-3066-2917-e6138a6fc285 type=ethermet autocommect-priority==999 interface-name=ens160 timestamp=1724696151 [tip04] method=mmunl address=192.168.18.94/24,192.168.18.2 dms=223.5.5.5 muy=fail=faise [ip61] addr=gen=mode=eui64 method=auto [proxy]
uu ta' 1947e 18a - 47.43 - 3646 - e317 - e61.38.66 (285 type = et herment autocommect - pri or i ty = -999 inter1ac - name:ensi60 timestamp=1724696151 [tipv4] method=annual address=192.166.10.94.74,192.166.10.2 dns=223.5.5.5 may-fail=false [tipv6] addr-gen-mode=zu164 method=auto [proxg] - -
uurid=1947e/B8-d7.48-3066-d917-e61.38d6fc285 type=stherenet autocommect-prior ity==999 interface=name=ensi60 timestamp=1724696151 [ethermet] [ipv4] method=manual address=192.168.18.94/24.192.168.18.2 dns=es23.5.5.5 method=uto [ipv6] addr-gue=mode=eui64 method=uto
uurid=1947e/18a-d7X-d8.3066-d317-e6138d6fc285 type=etherment autocommect-priority==999 interface=name=ensi60 timestamp=1724696151 [etherment] [ipv01] method=manual addresss=192.168.10.94v24.192.168.18.2 dms=223.55.5 muy=fail=false [ipv61] addr-gen=mode=eusi64 method=auto [prosug]
unid=1947e/B8-d7X-d813666-d817-e8138a6fc285 type=ethernet autocommect-priority=-999 interface-name=ensi60 timestamp=1724696151 [ethernet] tipv41 method=manual address=192.168.10.94v24,192.168.10.2 dafs=223.5.5.5 muj - All=false [ipv6] inethod=auto [iproxg] ~
uurid=1947e/186-d73-3066-d917-e6138d6fc285 type=ethernet autoconnect-prior ity==999 interface=name=ens16d0 timestamp=1724696151 [ethernet] [ipv4] method=manual address=192.168.10.94/24,192.168.10.2 may=fail=false [ipv6] addr =gen=mode=eui64 method=auto [proxy]
uu id 1947e/Ba-d728-3506-4917-e6138a6fc285 type=ethernet autocommect-priority=-999 interface-name=ens160 t imestamp=1724696151 [tipv4] method=manual address=192.168.10.94/24,192.168.18.2 dns=223.55.5 mu_fail=false [tipv6] addr-gen-mode=eu164 method=auto
uu id 1947e/184-87.28-3906-2917-eb138d6fc285 type=ethernet autocommect-priority=-999 interface=name:ensi60 timestamp=1724696151 [ethernet] [ipv4] method=manual address=192.168.18.94/24,192.168.18.2 dms=223.5.5.5 muj=fail=false [ipv6] address=uto
uurid=1997e/18a-d7.2a-3956-a917-e6138a6fc285 type=ethernet autocommect-prior ity==999 interface=name=ens160 timestamp=1724696151 [cthernet] [ipv4] method=manual address=192.168.18.94/24.192.168.18.2 dms=223.5.5.5 [ipv6] [ipv6]
uu id 1947e18d=8728-380b=4917=e0138abfc285 type=ettement autocommect=priority==999 interface=name=ens160 timestamp=1724696151 Itipv41 method=manual address=192.166.10.94×24.192.168.10.2 dns=223.5.5.5
uu id 1997/2186 d7.38-3506-3917-eb138d6f c285 type=sthernet autocommect-prior ity=-999 interface-name=smst60 timestamp=1724696151 [ethernet] [ipv4] method=manua] address=192.168.18.94/24,192.168.18.2 dms=223.5.5.5
uu id 1947e/184-87.28-39.06-89.17-eb.138.66 c285 type=cthernet autocommect-priority=-999 interface-name=ens160 timestamp=1724696151 [ethernet] (ipv1) method=manual address=192.168.10.94/24,192.168.10.2 dms=223.55.5
uu id 1947ella 47/38-380b-4917eb138a6fc285 type=thermet autocommect-priority=-999 interface-name=ensi60 timestamp=1724096151 tethermet] fipo41 method-manual method-manual
uu id =1947e1Ba-#278-38Bb-a917-eb13BaBfc285 type=stherment autocommect-priority=-999 interface-name=ems160 timestamp=1724696151 [ethernet]
um 1d 1947e18649728-3806-A917-eb138a6fc285 type=sthermet autocommect-priority=-999 interface-mame=sms160 timestamp=1724696151 [Ethermet]
uu id 1947e/184-0728-0906-0917-eb138d6fc285 type=ethernet autocommect-priority=-999 interface-name:ens160 t imestamp=1724696151
uu 1d = 1997/e18a = 4738 = 3936 = a917 - eb 138a6f c285 type=zethermet autoconnect-priority=-993 interface=name=ensi568 timestamp=1724696151
uu 14 ⁻ 1947e18a-4738-3506-a917-e6138a6fc285 type=sthermet autocommect-priori typ=-999 interface-assessess1618
uu 1d=194/e18a-8/38-380b-a917-e0138a6/c285 type=cthernet # terement main: / tere 000
uu 1d=1947e18a-8738-38bb-a917-eb138a6fc285
id=ens160

(3) 配置完成后重启命令也与 Redhat 有区别:

\$# nmcli c reload

\$# nmcli c up ens160.nmconnection 或者 ens160

<pre>[root@localhost system-connections]# nmcli c up ens160 Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/5) Iroot@localhost system-connections]# _</pre>
LTOOLUIOCALINOST SUSTEM-COMMECTIONSI# 1P a
1: 10: <luuybhuk, luwlk_uf="" uf,=""> MTU 5555 QAISC NOQUEUE STATE UNKNUWN Group Aeiauit gien 1800</luuybhuk,>
ing/loupack weige weige weige braile and a weige weige weige and a second weige weige and a second here the second here there there there ther
List 127.0.0.170 Scope Host To
inef
ualid lft forever mederred lft forever
2: ens160: (BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000 link/ether 00:0c:29:2e:05:24 bd ff:ff:ff:ff:ff:
altname enp3sØ
inet 192.168.10.94/24 brd 192.168.10.255 scope global noprefixroute ens160
valid_lft forever preferred_lft forever
inet6 fe80::20c:29ff:fe2e:524/64 scope link noprefixroute
valid_lft forever preferred_lft forever
[root@localhost_system-connections]#

<mark>注</mark>:这里重启网卡会遇到两种情况,在 Vmware 虚机中 cd 到…/network-scripts 目录下重

启会是 ens160.nmconnection, 如果是 Xshell 会显示 ens160, 无需在意使用 Tab 会解决

这个顾虑

(4) 重启完成后,还有一项较重要的 SSH 配置,否则无法使用 Xshell 连接 SSH,即使 22

端口存在也无法连接:

(1)在/etc/ssh 目录下修改 sshd_config 文件找到#PermitRootLogin prohibit-password

这项注释的参数在源参数复制一份出来再修改,修改为: PermitRootLogin yes



<mark>用意</mark>:意思是表示可以通过 root 进行 ssh 访问。

\$# systemctl restart sshd #重启 ssh 服务

成功连接Xshell

[root@localhost ~]#

(5) 接下来配置 yum 源参考阿里云开源镜像站配置方式:

https://developer.aliyun.com/mirror/?serviceType=&tag=&keyword=Rockylinux



\$#sed -e 's|^mirrorlist=|#mirrorlist=|g' \ -е 's|^#baseurl=http://dl.rockylinux.org/\$contentdir|baseurl=https://mirrors.aliyun.co m/rockylinux|g' \

-i.bak \

/etc/yum.repos.d/rocky*.repo

\$#dnf makecache

至此系统基础配置都以完成,开始部署 zabbix7.0~

三、部署 Zabbix7.0:

【Zabbix 基本概念: <u>https://www.zabbix.com/documentation/current/zh/】</u>

(1) 使用 Rpm 安装对应版本的 zabbix 仓库:

\$#

rpm

-Uvh

https://repo.zabbix.com/zabbix/7.0/rocky/9/x86_64/zabbix-release-7.0-2.el9.noarc

<u>h.rpm</u>



<mark>注</mark>:默认是 Zabbix 官方仓库如网络问题无法下载,可更换阿里仓库。

<pre>[zabbix] name=Zabbix Official Repository = \$basearch #baseur_l=https://irrors.aliyun.com/zabbix/7.0/rocky/9/\$basearch/ baseur_l=https://irrors.aliyun.com/zabbix/Zabbix/7.0/rocky/9/\$basearch/ gpgcheck=1 gpgcheck=1 gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-ZABBIX-B5333005 [zabbix-non-supported] name=Zabbix Official Repository non-supported = \$basearch #baseur_l=https://mirrors.aliyun.com/zabbix/non-supported/rhel/@{sbasearch/ baseur_l=https://mirrors.aliyun.com/zabbix/non-supported/rhel/@{sbasearch/ enabled=1 gpgkey=file://etc/pki/rpm-gpg/RPM-GPG-KEY-ZABBIX-08EFA7DD gpgkey=file://etc/pki/rpm-gpg/RPM-GPG-KEY-ZABBIX-08EFA7DD gpgkey=file://etc/pki/rpm-gpg/RPM-GPG-KEY-ZABBIX-08EFA7DD</pre>		
[zabbix-sources] name_Zabbix (Prieo. zabbix, com/zabbix/7.0/rocky/9/SRPMS enabled=0 gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-ZABBIX-B5333005 gpgcheck=1		
~ "/etc/yum.repos.d/zabbix.repo" 22L, 780B	12, 62	全部
[zabbix]		

baseurl=https://mirrors.aliyun.com/zabbix/zabbix/7.0/rocky/9/\$basearch/

[zabbix-non-supported]

baseurl=https://mirrors.aliyun.com/zabbix/non-supported/rhel/9/\$basearch/

(2) 安装 Zabbix_Server, Web+agent:

\$# yum install zabbix-server-mysql zabbix-web-mysql zabbix-nginx-conf zabbix-sql-scripts zabbix-selinux-policy zabbix-agent -y

 腔正 : php-bomath-8.0.30-1.el9.2.x86,64 能证 : nginx-core-1:1.20.1-14.el9.2.1.x86.64 能证 : zabbix-agent-7.0.3-release1.el9.x86.64 Watu: zabbix-selinux-policy-7.0.3-release1.el9.x86 zabbix-web-deps-7.0.3-release1.el9.x86 zabbix-web-deps-7.0.3-release1.el9.x86 zabbix-web-deps-7.0.3-release1.el9.x86 	rch x86_64 664 rrch h		36/47 37/47 38/47 39/47 40/47 41/47 42/47 43/47 43/47 45/47 46/47 47/47
已安装: Open IPM - I ibs-2. 0. 32-3. e19. x86_64 fonts-filesystem-1:2. 0. 5-7. e19. 1. noarch gd-2. 3. 2-3. e19. x86_64 httpd-filesystem-2. 4. 57-11. e19_4. 1. noarch libX11-1. 7. 0-9. e19. x86_64 libtiff-4. 4. 0-12. e19. x86_64 mariadb-connector-o-3. 2. 6-1. e19. 0. x86_64 npin-bcmatt-80. 30-1. e19. 2. x86_64 php-gd-80. 30-1. e19. 2. x86_64 zabbix-agent-7t. 0. 3-release1. e19. x86_64 zabbix-agent-7. 0. 3-release1. e19. x86_64 zabbix-web-deps-7. 0. 3-release1. e19. noarch	de javu-sans-fonts-2. 37-18. e19. noarch fping-5. 1-1. e19. x86_64 graphite2-1. 3. 14-9. e19. x86_64 jbigkit-libs-2. 1-23. e19. x86_64 libx11-common-1. 7. 0-9. e19. noarch libjpeg-turbe2. 0. 90-7. e19. x86_64 libwebp-1. 2. 0-8. e19. x86_64 php-tomonector-c-coorfig-3. 2. 6-1. e19_0. noarch nginx-core-1:1. 20. 1-14. e19. 2. 1. x86_64 php-tdap-8. 0. 30-1. e19_2. x86_64 php-tdap-8. 0. 30-1. e19_2. x86_64 php-tdap-8. 0. 30-1. e19_2. x86_64 unix008c-2. 3. 9-4. e19. x86_64 zabbix-nginx-corf-7. 0. 3-release1. e19. noarch zabbix-nginx-corf-7. 0. 3-release1. e19. noarch	fontconfig=2.14.0-2.e19_1.x86_64 freetype=2.10.4-9.e19.x86_64 langpackscore=fontten=3.0-16.e19.noarch libbau=1.0.9-8.e19.x86_64 libpog=2.16.37-12.e19.x86_64 libtocb=1.13.1-9.e19.x86_64 net_snmpc-libs=1:5.9.1-13.e19.x86_64 nginx=filesystem=1:1.20.1-14.e19_2.1.noarch php=mbstring=8.0.30-1.e19_2.x86_64 php=mbstring=8.0.30-1.e19_2.x86_64 zabbix=selinux=policy=7.0.3-release1.e19.x86_64 zabbix=web=7.0.3-release1.e19.noarch	
完毕! [root@localhost ~]# <mark> </mark>			

(3) 安装数据库、初始化:

安装数据库:	\$# yum install -y mariadb	mariadb-server	"图-1"
初始化: \$#।	mysql_secure_installation		"图-2"

图-1

验证 : perl-parent-1:0.238-460.el9.noarch			80/80
已安装:			
checkpolicy=3, 6=1, el9, x86, 64	mariadb=3:10.5.22=1.el9 2.x86 64	mariadb-backup-3:10.5.22-1.el9 2.x86 64	
mariadb-common-3:10 5 22-1 el9 2 x86 64	mariadb-errmsg-3:10 5 22-1 el9 2 x86 64	mariadb-gssani-server-3:10 5 22-1 el9 2 x86 64	
mariadb-server-3:10.5.22-1.el9.2.x86.64	mariadb-server-utils-3:10.5.22-1.el9 2.x86 64	mysgl-selinux-1.0.10-1.el9.noarch	
perl-Autoloader-5 74-481 el9 poarch	perI-B-1 80-481 el9 x86 64	perl-Carp-1 50-460 el9 poarch	
perI-Class-Struct-0.66-481.el9.noarch	perI-DBD-MariaDB-1,21-16,el9,0,x86,64	per I-DBI-1, 643-9, e19, x86, 64	
perl-Data-Dumper-2 174-462 el9 x86 64	perl-Digest-1 19-4 el9 noarch	per1-Digest-MD5-2 58-4 e19 x86 64	
per I-Dynal oader-1, 47-481, e19, x86, 64	per I-Encode-4:3 08-462 e19 x86 64	perI-Frrno-1 30-481 el9 x86 64	
perl-Exporter-5,74-461,el9,noarch	perI-EcntI-1, 13-481, el9, x86, 64	perI-File-Basename-2, 85-481, el9, noarch	
perl-File-Copy-2 34-481 el9 poarch	perI-File-Path-2 18-4 el9 noarch	perI-File-Temp-1:0 231 100-4 el9 noarch	
perI-File-stat-1.09-481.el9.noarch	perI-FileHandle-2,03-481,el9, noarch	perl-Getopt-Long-1:2,52-4,el9, noarch	
perl-Getopt-Std-1, 12-481, el9, noarch	perI-HTTP-Tinv-0.076-462.el9.noarch	per 1-10-1, 43-481, e19, x86, 64	
perI-IQ-Socket-IP-0, 41-5, eI9, noarch	perI-I0-Socket-SSL-2, 073-1, el9, noarch	per - PC-0pen3-1, 21-481, e 9, noarch	
perI-MIME-Base64-3, 16-4, e19, x86, 64	perl-Math-BigInt-1:1,9998,18-460,el9, noarch	perl-Math-Complex-1.59-481.el9.noarch	
perl-Mozilla-CA-20200520-6.el9.noarch	perI-NDBM File-1, 15-481, el9, x86, 64	per I-Net-SSLeav-1, 92-2, e19, x86, 64	
per -POSIX-1,94-481,el9,x86_64	per -PathTools-3, 78-461, el9, x86 64	perI-Pod-Escapes-1:1.07-460.el9.noarch	
per I-Pod-Per Idoc-3, 28, 01-461, e19, noarch	perI-Pod-Simple-1:3, 42-4, el9, noarch	perI-Pod-Usage-4:2,01-4,el9, noarch	
perl-Scalar-List-Utils-4:1,56-461,e19,x86 64	perI-SelectSaver-1, 02-481, el9, noarch	per I-Socket-4:2,031-4,e19,x86 64	
perl-Storable-1:3,21-460,el9,x86_64	per -Symbol-1,08-481,el9,noarch	perI-Svs-Hostname-1, 23-481, eI9, x86 64	
perI-Term-ANSIColor-5,01-461,el9,noarch	perI-Term-Cap-1, 17-460, e19, noarch	perI-Text-ParseWords-3, 30-460, el9, noarch	
perI-Text-Tabs+Wrap-2013.0523-460.el9.noarch	perI-Time-LocaI-2:1, 300-7, el9, noarch	perI-URI-5, 09-3, eI9, noarch	
perl-base-2, 27-481, el9, noarch	perl-constant-1, 33-461, el9, noarch	perl-if-0,60,800-481,el9,noarch	
per I-interpreter-4:5. 32. 1-481. el9. x86 64	perI-libnet-3, 13-4, el9, noarch	perI-libs-4:5. 32. 1-481. el9. x86 64	
perI-mro-1, 23-481, el9, x86 64	perl-overload-1,31-481,el9,noarch	perl-overloading-0.02-481.el9.noarch	
per I-parent-1:0. 238-460. eI9. noarch	perl-podlators-1:4.14-460.el9.noarch	perI-subs-1.03-481.el9.noarch	
perl-vars-1.05-481.el9.noarch	policycoreutils-python-utils-3.6-2.1.el9.noarch	python3-audit-3.1.2-2.el9.x86 64	
python3-distro-1.5.0-7.el9.noarch	python3-libsemanage-3.6-1.el9.x86 64	python3-policycoreutils-3, 6-2, 1, el9, noarch	
python3-setools-4.4.4-1.el9.x86_64	python3-setuptools-53.0.0-12.el9_4.1.noarch		
元毕!			
Liooteiocamost j#			

图-2 注:初始化遇到找不到 sock 文件问题,手动启动数据库服务即可解决。

[root@localhost ~]# mysql_secure_installation	
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MA SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFUL	ariaDB _Y!
In order to log into MariaDB to secure it, we'll need the currer password for the root user. If you've just installed MariaDB, an haven't set the root password yet, you should just press enter !	nt nd nere.
Enter current password for root (enter for none): ERROR 2002 (HYOOO): Can't connect to local MySQL server through Enter current password for root (enter for none): ERROR 2002 (HYOOO): Can't connect to local MySQL server through Enter current password for root (enter for none):	socket '/var/lib/mysql/mysql.sock' (2) socket '/var/lib/mysql/mysql.sock' (2)
ERROR 2002 (HY000): Can't connect to local MySQL server through Enter current password for root (enter for none): ERROR 2002 (HY000): Can't connect to local MySQL server through Enter current password for root (enter for none):	socket '/var/lib/mysql/mysql.sock' (2) socket '/var/lib/mysql/mysql.sock' (2)
初始化选项:	选择 Y/n
Enter current password for root (enter for	直接回车

. . . .

none):	
#:输入 root 的当前密码(输入表示无)	
Switch to unix_socket authentication [Y/n]	у
#:切换到 unix_socket 身份验证[是/否]	
Change the root password? [Y/n]	у
#:更改 root 密码?[是/否]	
Remove anonymous users? [Y/n]	у
#:删除匿名用户?[是/否]	
Disallow root login remotely? [Y/n]	n
#:不允许远程 root 登录?[是/否]	
Remove test database and access to it?	У
[Y/n]	
#:删除测试数据库并访问它?[是/否]	
Reload privilege tables now? [Y/n]	у
#:现在重新加载权限表吗?[是/否]	

Remove anonymous users? [Y/n] y ... Success!

Normally, root should only be allowed to connect from 'localhost'. This ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] n ... skipping.

By default, MariaDB comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

Remove test database and access to it? [Y/n] y - Dropping test database... ... Success! - Removing privileges on test database... ... Success!

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

Reload privilege tables now? [Y/n] y ... Success!

Cleaning up...

All done! If you've completed all of the above steps, your $\ensuremath{\mathsf{MariaDB}}$ installation should now be secure.

Thanks for using MariaDB! [root@localhost ~]# <mark>|</mark>

н

(4) 导入初始架构和数据:

导入前准备工作:先创建用户与数据库。

create database zabbix character set	创建名为 zabbix 的数据库,并设置字符集为	
utf8mb4 collate utf8mb4_bin;	utf8mb4。	
create user zabbix@localhost identified	创建名为 zabbix 的用户,并设置其密码为	
by 'pwd123';	pwd123。	
grant all privileges on zabbix.* to	授予zabbix用户在zabbix数据库上的所有	
zabbix@localhost;	权限。	
MariaDB [(none)]> create database zabbix character set utf8mb4 collate utf8mb4_bin; Query OK, 1 row affected (0.000 sec)		
MariaDB [(none)]> create user zabbix≋localhost identified by 'pwd123'; Query DK, 0 rows affected (0.001 sec)		

MariaDB [(none)]> grant all privileges on zabbix.* to zabbix@localhost; Query OK, O rows affected (0.001 sec)

● 导入数据: \$# zcat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql

--default-character-set=utf8mb4 -uzabbix -p zabbix 。

• 查看库大小: SELECT table_schema AS "zabbix",

ROUND(SUM(data_length + index_length) / 1024 / 1024, 2) AS "Size (MB)"

FROM information_schema.TABLES

GROUP BY table_schema;
MariaDB [zabbix]> SELECT table_schema AS "zabbix", -> ROUND(SUM(data_length + index_length) / 1024 / 1024, 2) AS "Size (MB)" -> FROM information_schema.TABLES -> CGPUIP BY table_schema:

zabbix	Size (MB)
information schema	0.20
mysal	3, 19
performance schema	0.00
zabbix	60.69

MariaDB [zabbix]> 📘

(5) 接下来到最后配置 Server 和 Nginx:

● Nginx 配置:

\$#vim /etc/nginx/conf.d/zabbix.conf



- Server 配置:
- \$#: vim /etc/zabbix/zabbix_server.conf

### Option: DBUser # Database user. # Mandatory: no # Dofault: # Dofault: # DBUser=			
DBUser=zabbix			
### Option: DBPaseword # Database pawword. # Comment this live if no password is used. # Mandatory: no # Default- DBPassword=pwd12			
### Option: DBSocket # Path to MySQL socket. # Mandatory: no # Default: # DBSocket=			
<pre>### Option: DBPort</pre>			
# Default: "/etc/zabbix/zabbix_server.conf" 1126L, 29842B	131, 17	10%	I

● 启动服务: Server、nginx、Agents(前面已启动过数据库):



● 访问页面: <u>http://IP:8080</u>, 默认账密: Admin/zabbix。

Back Net dep
Lisensed under AGPLv2

	2	Global view					? 新田仪表録 🗮 👯
	۹	 (次表盘已更新) 					×
88 仪表盘		仪表盘 / Global view					
" "		Top hosts by CPU utilization		系统信息			
a. 1899		Host name Utilization 1m avg 5m avg 15m avg Processes	1 751	华权	值	详情	
ල කිංසයන්		Zabbix server 0.12 % 0.00 0.00 0.00 345	Zabbix server	Zabbix服务器端运行中	疉	localhost:10051	40.20
局报表			Values per second	zabbix能明编版本	7.0.3	Up to date	10:39
() #####				主机数量(已启用)已禁用)	1	1/0	Chanabai
0 ##				模板数量	315		Shanghai
				监控项数量 (已启用/已禁用/不支持)	128	117/0/11	
S, HP				*** *********************************	77	77 / 0 [0 / 77] -	
(5) 管理		主机可用性按严重性	生的问题		_	地理地图	
		1 0 0 0 1 0 时用 不可用 用合 未1000 合计 天和	0 ≓⊞ -		ŝ	+	7
0.77		Current problems				•	
6.4 1 210		时间▼ 信息 主机 问题・严重性	問问	更新 动作 标签			
③ 生成							0
⑦ 1880							
<u>冬</u> 用户设置							
			Q				

以上就是本期的全部内容。大家好,我是乐乐专注 IT 运维技术研究与分享,更多运维技巧 欢迎关注乐维社区,更多运维问题也欢迎到乐维社区留言提问。

二十九、如何在离线环境中编译安装 Zabbix

说明

有时候为了安全,公司的内网环境是不连接外网的,然后又需要针对性的重新编译一个特殊 功能的 Zabbix 监控系统,但是相关的依赖的安装是个比较麻烦的问题,要么制作一个本地 yum 源进行依赖安装,要么做一个网络映射,让内网机器可以访问外网。如果既不想搭建 yum 源,又不想做网络映射,那该怎么办?

为解决这个问题。本文将介绍使用 centos 系统来下载编译所需的相关依赖包,然后将依赖 包和 Zabbix 源码包一起上传到内网机器,从而离线环境中编译安装 Zabbix。

前提条件

确定好内网机器的操作系统版本

系统要能执行 yum 命令

下载 rpm 依赖包和 zabbix 源码包

确保系统一致性

首先确定好内网机器的操作系统版本。这里我的环境机器系统版本是 centos7.9

[root@docker ~]# cat /etc/redhat-release CentOS Linux release 7.9.2009 (Core) [root@docker ~]#

然后在外网找一台操作系统一致的机器 (一定要确保操作系统一致)

```
[root@general-test ~]# cat /etc/redhat-release
CentOS Linux release 7.9.2009 (Core)
[root@general-test ~]#
```

下载全量依赖包

安装下载命令

yum -y install yum-utils

下载全量依赖包

文章编译 zabbix 所需的依赖有 gcc gcc-c++ net-snmp-devel libevent-devel OpenIPMI

OpenIPMI-devel openssl-libs mariadb-devel device-mapper rpm libaio*

执行下载命令:

repotrack gcc gcc-c++ net-snmp-devel libevent-devel OpenIPMI OpenIPMI-devel

openssl-libs mariadb-devel device-mapper rpm libaio*

等待下载完成

```
[root@general-test rpm]#
[root@general-test rpm]# ls | wc -l
317
[root@general-test rpm]# cd ..
[root@general-test ~]# zip -r rpm.zip rpm/
```

然后打包

下载 zabbix 源码包

https://www.zabbix.com/download_sources

到官网下载需要的版本

上传依赖包和源码包到离线服务器

然后将打包的 rpm.zip 包和源码包上传到离线服务器



安装依赖包

解压依赖包



yum -y localinstall ./rpm/*.rpm

若是出现安安装失败,或者缺少依赖,可忽略依赖。命令: yum -y localinstall ./rpm/*.rpm

--skip-broken

亦或是在下载依赖时补全缺少的依赖。

[root@docker	zabbix]# ls
rpm rpm.zip	zabbix-6.2.6.tar.gz
[root@docker	zabbix]# yum -y localinstall rpm/*.rpm

安装完成

Verifying : xz-5.2.2-1.el7.x86_64 Verifying : python-2.7.5-89.el7.x86_64				100/101 101/101
Installed: audit-1108.1686 0:2.8.5-4.el7 gdm.1688 0:1.108.el7 glubc-haders.086 64 0:2.17-326.el7_9 lubdm.1686 0:1-3.212-25.el7 lubgmp:1600 0:4.6.5-4.4.el7 norrses-1106.15.904.4.el7 norrses-1106.15.904.4.el7 per:0.686 0:3.217.el7 pelte:1.686 0:3.7.17.6.el7_7.1	cpp.x80_64 0:4.8.5.44.017 gbm.dewel.i686 0:1.10.8.017 libb/-dewel.i686 0:5.371 libb/-dewel.i686 0:5.37125.017 libb/-dewel.i686 0:5.37125.017 libb/-sept.s20.017	cracklib.1666 0:2.9.0.11.el7 gdm. devel.065 64 0:1.10-8.el7 kernel-hacker.x65 64 0:1.10-8.el7 libgicality.1565 0:1.3.2.2.5.el7 libgicality.1565 0:1.3.2.5.el7 liblicality.1565 0:1.3.2.5.el7 liblicality.1565 0:1.3.2.5.el7 liblicality.1565 0:1.3.2.5.el7 perf-textliblicality.15468/def.ans.0.6.16.6.3.el7 perf-textliblicality.15468/def.ans.0.6.16.6.3.el7 perf-textliblicality.15468/def.ans.0.6.16.0.19.el7 perfectivel.15468/def.ans.0.6.4.0.13.el7	9cc.308_64 0;4.8.5.44.817 9lbb:-devel.1686 0;2.7.7.230.817.9 keyttls-lbb.1686 0;3.3.83.817 lbf1;1.686 0;3.3.013.013.01 lbs4lum:.086 0;2.5.15.41 11 lbs4lum:.080 0;2.5.15.41 11 141:trust.1680 0;0.23.5.3.817 pyrt.5xUUL+homireft.nast.01.01.6244.817 pyparsing.noarch 0;1.5.6-9.817	gcc.c++.x88_64 0:4.8.5.44.el7 glib:dewel.x88_64 0:2.17-326.el7_9 lib:ga:pn:d66 0:0.7.5.4.el7 lib:gci.x888 0:2.5.10.el7 jib:gci.x888 0:2.5.10.el7 pm:d60 0:1.1.8.23 el7 pm:l:stutis.ParesK.march 1:3.18-3.el7 readine.x886 0:6.2.11.el7
Updated: bash.x86_64_0:4.2.46-35.el7_9 glibc.1686_0:2.17-326.el7_9 nspr.x86_64_0:4.34.0-3.1.el7_9 nss-softkn=frebl.x86_64_0:3.79.0-4.el7_9 python-libs.x86_64_0:2.7.5-92.el7_9 zlib.x86_64_01:2.7.2-0.el7_9	binutils.x86_64 0:2.27-44,base.el7_9.1 ca-certi glibc.x86_64 0:2.17-326.el7_9 glibc-co nss.i680 0:3.79.0-4.el7_9 nss.x80 nss-sysini.x86_64 0:3.79.0-4.el7_9 nss.tool tzdata.noarch 0:2022g-1.el7 xz.x86_6	ficates.noarch 0:2022.2.54-74.el7_9 coreutils.x86.64 0:1 mmon.x86.64 0:2.17-326.el7_9 grip.x86.64 0:1 64 0:3.79.0-4.el7_9 nss-softak.n58 x86.64 0:3.79.0-4.el7_9 nss-util.1686 0 4 0:5.2.2-2.el7_9 xz-libs.1686 0:	40:8.22-24.el7_9.2 expat.1666 0:2.1.0-15.el7_9 .5:11.el7_9 kpartx.x86_64 0:0.4.9-136.el7_1 16 0:3.79.0-4.el7_9 nss-softokn.x86_64 0:3.79.0-4. 13.79.0-1.el7_9 nss-ut1.x86_64 0:5.70.0-1.el7. 5.2.2-2.el7_9 xz-libs.x86_64 0:5.2.2-2.el7_9	expat.x86_64 0:2.1.0-15.el7_9 nspr.i686 0:4.34.0-3.1.el7_9 817_9 nss-softokn.freebl.i686 0:3.79.0-4.el7_9 9 python.x86_64 0:2.7.5.92.el7_9 2lib.i686 0:1.2.7-20.el7_9

开始编译 zabbix

编译检查

解压 zabbix 源码包,并开始编译检查

./configure --prefix=/data/zabbix --enable-server --enable-agent --with-mysql

--with-net-snmp --with-libcurl --with-libxml2 --with-openipmi --enable-ipv6



结果无问题,开始编译安装

make -j2 && make install

Moubus:	110		
Linker flags:	-rdynamic		
Libraries:	-lz -lpthread	-lcurl -lm -ldl -lresolv	-lpcre
Configuration file	/data/zabhix/etc	/zabbix agentd conf	
Moduloc	/data/zabbix/lib	/madulac	
Moud ces :	/udid/2d001X/liu	/modules	
Enable agent 2:	no		
Enable web service:	no		
Enable Java gateway:	no		
LDAP support:	no		
IPv6 support:	yes		
*****	*****	*****	
* Now run 'ma	ke install'	*	
*		*	
* Thank you f	for using Zabbix!	*	
* <http: td="" w<=""><td>ww.zabbix.com></td><td>*</td><td></td></http:>	ww.zabbix.com>	*	
******	*****	******	
[root@lwops zabbix-6.2.6]# make -j2 && make	install.	

编译完成

	manoral in control of the state
	make[2]: Entering directory `/tmp/zabbix-6.2.6/man'
	make[2]: Nothing to be done for `install-exec-am'.
	/usr/bin/mkdir -p '/data/zabbix/share/man/manl'
	/usr/bin/install -c -m 644 'zabbix_get.man' '/data/zabbix/share/man/manl/zabbix_get.l'
	/usr/bin/install -c -m 644 'zabbix_sender.man' '/data/zabbix/share/man/manl/zabbix_sender.1'
	/usr/bin/mkdir -p '/data/zabbix/share/man/man8'
	/usr/bin/install -c -m 644 'zabbix_agentd.man' '/data/zabbix/share/man/man8/zabbix_agentd.8'
	/usr/bin/install -c -m 644 'zabbix_server.man' '/data/zabbix/share/man/man8/zabbix_server.8'
	make[2]: Leaving directory `/tmp/zabbix-6.2.6/man'
	make[1]: Leaving directory `/tmp/zabbix-6.2.6/man'
	Making install in misc
	make[1]: Entering directory `/tmp/zabbix-6.2.6/misc'
	make[2]: Entering directory `/tmp/zabbix-6.2.6/misc'
	make[2]: Nothing to be done for `install-exec-am'.
_	make[2]: Nothing to be done for `install-data-am'.
	make[2]: Leaving directory `/tmp/zabbix-6.2.6/misc'
	<pre>make[1]: Leaving directory `/tmp/zabbix-6.2.6/misc'</pre>
-	<pre>make[1]: Entering directory `/tmp/zabbix-6.2.6'</pre>
	<pre>make[2]: Entering directory `/tmp/zabbix-6.2.6'</pre>
	make[2]: Nothing to be done for `install-exec-am'.
	make[2]: Nothing to be done for `install-data-am'.
	make[2]: Leaving directory `/tmp/zabbix-6.2.6'
	<pre>make[1]: Leaving directory `/tmp/zabbix-6.2.6'</pre>
	[root@lwops zabbix-6.2.6]#



复制启动脚本:

cp -ra ./misc/init.d/fedora/core/* /etc/init.d/

修改脚本路径:

vim /etc/init.d/zabbix_server

vim /etc/init.d/zabbix_agentd

将里面的:

BASEDIR=/usr/local

修改为:

BASEDIR=/data/zabbix

完成后便可使用系统命令来启动或停止 zabbix

service zabbix-server start/stop/status/restart

service zabbix-agentd start/stop/status/restrat

或

systemctl start/stop/status/restart zabbix_server

systemctl start/stop/status/restart zabbix_agentd

三十、更多……

告警

三十七、Zabbix 事件告警监控:如何实现对相同部件触发 器告警及恢复的强关联

有一定 Zabbix 使用经验的小伙伴可能会发现,接收告警事件时,其中可能包含着大量不同 的部件名,同一部件的事件在逻辑上具有很强关联性,理论上应保持一致的告警/恢复状态, 但 Zabbix 默认并未对它们进行关联,直接后果是运维人员只能进行大量重复操作,进而对 部件的状态进行校正。

实际上,虽然 Zabbix 默认未对相同部件进行关联,但却可以通过手动配置实现关联操作。 本文将深入探讨 Zabbix 在处理相同部件触发器的告警和恢复过程中的强关联机制。通过这 种机制,我们可以确保一旦触发器状态发生变化,相关的告警能够被准确触发,并且在问题 解决后,告警状态能够及时恢复,从而避免无效告警的干扰和资源的浪费。

例如,当前监控中有对硬件设备事件采集监控项(数据如下),要对其配置触发器告警,但 是每个事件中包含告警对应的部件名,希望对相同部件的告警实现告警-恢复事件的自动关 联。

test_ipmi_192.168	3.3.195: 事件日志			以複	图 值 ~	统文字 🔀
				く 缩小	> 还剩1小时 (3	过滤器 🏹
		EJA now	还则 2 天 还则 7 天 还则 30 天 还则 3 个月 还则 6 个月 还则 1 年 还则 2 年	昨天 前天 上同的这一天 上一周 上一个月 去年	今天 今天到目前为止 本間 本周到目前为止 本月 这个月到目前为止 本年 今年到目前为止	还则 5 分钟 还则 15 分钟 还则 30 分钟 还则 1 小时 还则 3 小时 还则 6 小时 还则 12 小时 还则 1 天
时间戳 当地时间	目 爊					
2024-05-10 10:11:30	["0060", "05/10/2024 10:04:18", " Presence Detected", "Assertion", " DINM131 (0x4e)", " Memory"]					
2024-05-10 10:11:29	["005f", "05/10/2024 10:04:17", " Presence Detected", "Assertion", " DINM130 (0x4d)", " Memory"]					
2024-05-10 10:11:29	["005e", "05/10/2024 10:04:17", " Presence Detected", "Assertion", " DIMM121 (0x4b)", " Memory"]					
2024-05-10 10:11:29	["005d", "05/10/2024 10:04:17", " Presence Detected", "Assertion", " DIMM120 (0x4a)", " Memory"]					
2024-05-10 10:11:29	["005c", "05/10/2024 10:04:17", " Presence Detected", "Assertion", " DINM111 (0x48)", " Memory"]					
2024-05-10 10:11:29	["005b", "05/10/2024 10:04:17", " Presence Detected", "Assertion", " DIMM110 (0x47)", " Memory"]					

图 1

整理 by 乐维社区(https://forum.lwops.cn)

数据字段解析:

监控取值中,第4位为告警状态,"Assertion"表示告警产生,"Deassertion"表示告警恢复。

配置思路:

方法一(建议):

参考方法一中基础配置,额外补充事件匹配-标签

ATTING A DAMAGE AND THE PARTY								
1671日1 16680大张			触发器 4	标记 1 依赖关系				
* 名称	{硬件}{HOST.NAME} 告替 {ITEM.VALUE}				Г	Trianas tons Inherited and trianas tons	1	
Event name	(硬件){HOST.NAME} 告答 {ITEM.VALUE}					and and any and any	175	2h
Operational data						PartName	{{TEM.VALUE}.regsub(tion\''\\s*\'\'s*\(S)+\s*\(.\1)}	80
严重性	未知 信息 習音 次要 严重 紧急					酒加	10.000	
* 问题考现形式	find//IR Template for IPMT all	ið til						
10000000000	server/event_logs_json,,"like","Assertion")=1	704064				更新克隆創除取消		
	表达式构造器							
事件成功迭代	表达式 恢复表达式 无							
* 恢复表达式	find(/LW Template for IPMI all server/event logs ison, "like", "Deassertion")=1	添加						
	201-210-288							
问题事件生成模式	HA ST							
The state of the s								
专行40640天140	HTHEIUBE ATHEIUBERGROUNDERSTREET							
* Prestoric	PartName							
允许手动关闭								
URL								
描述								
	A							
已启用	~							

标记中"值"参数支持写法如下

{{ITEM.VALUE}.regsub(pattern, output)}

{{ITEM.VALUE}.iregsub(pattern, output)}

{{#LLDMACRO}.regsub(pattern, output)}

{{#LLDMACRO}.iregsub(pattern, output)}

图中示例写法:

{{ITEM.VALUE}.regsub("tion\"\,\s*\"\s*(\S+)\s*\(",\1)}

测试数据如下:

test_ipmi_192.168.3.	.195: 事件日志				以根据	e (iii v	统文字 💱
		自从 到	now-1h ::: now :::	还则 2 天 还则 7 天 还则 30 天	く 縮小 昨天 前天 上周的这一天	> 还剩1小时 (今天 今天到目前为止 本周	这述编器 ▼ 还则 5 分钟 还则 15 分钟 还则 30 分钟
			应用	还购3个月 还购6个月 还购1年 还购2年	上一周 上一个月 去年	本周到目前为止 本月 这个月到目前为止 本年 今年到目前为止	还剩1小时 还剩3小时 还剩6小时 还剩12小时 还剩1天
时间数 当地时间 鱼	ā.						
2024-05-10 11:41:58 ["0061", "05/10/2024 11:29:20", " Presence Detected", "Deassertion", " DIMM110 (0x47)", " Memory"]						
2024-05-10 11:41:52	"0060", "05/10/2024 11:28:18", " Fresence Detected", "Assertion", " DIMM131 (0x4e)", " Memory"]						
2024-05-10 11:41:51 ["005f", "05/10/2024 11:28:17", " Presence Detected", "Assertion", " DIMM130 (0x4d)", " Memory"]						
2024-05-10 11:41:51	"005e", "05/10/2024 11:28:17", " Presence Detected", "Assertion", " DIMM121 (0x4b)", " Memory"]						
2024-05-10 11:41:51 ("005d", "05/10/2024 11:28:17", " Presence Detected", "Assertion", " DIMM120 (0x4a)", " Memory"]						
2024-05-10 11:41:51	"005c", "05/10/2024 11:28:17", " Presence Detected", "Assertion", " DIMM111 (0x48)", " Memory"]						
2024-05-10 11:41:51	"005b", "05/10/2024 11:28:17", " Fresence Detected", "Assertion", " DINM110 (0x47)", " Memory"]						

测试结果:

< 7										◇ 〉 今年到目前为止 く 熔小 〉
时间▼	严重性	恢复时间 状态 信息	主机	问题			持续时间	确认	动作	标记
11:41:52	警告	问题	test_ipmi_192.168.3.195	[硬件]test_ipmi_192.168.3.195 告答 ["0060", "05/10/2024 11:28:18", " Presence Detected", "Assertion",	DIMM131	(0x4e)", " Memory"]	1m 4s	不	* ² →	Application: 常规 PartName DIMM131
11:41:51	警告	问题	test_ipmi_192.168.3.195	[硬件]test_ipmi_192.168.3.195 告答 ["005f", "05/10/2024 11:28:17", " Presence Detected", "Assertion", "	DIMM130	0x4d)", " Memory"]	1m 5s	不	.2	Application: 常规 PartName DIMM130
11:41:51	警告	问题	test_ipmi_192.168.3.195	[硬件]test_ipmi_192.168.3.195 告答 ["005e", "05/10/2024 11:28:17", " Presence Detected", "Assertion",	DIMM121	(0x4b)", " Memory"]	1m 5s	不	* ² →	Application: 常规 PartName DIMM121
11:41:51	警告	问题	test_ipmi_192.168.3.195	[硬件]test_ipmi_192.168.3.195 告答 ["005d", "05/10/2024 11:28:17", " Presence Detected", "Assertion",	DIMM120	(0x4a)", " Memory"]	1m 5s	不	*⇒	Application: 常规 PartName DIMM120
11:41:51	警告	问题	test_ipmi_192.168.3.195	[硬件]test_ipmi_192.168.3.195 告替 ["005c", "05/10/2024 11:28:17", " Presence Detected", "Assertion",	DIMM111	0x48)", " Memory']	1m 5s	不	*→	Application: 常规 PartName DIMM111
11:41:51	警告	11:41:58 已解决	test_ipmi_192.168.3.195	[硬件]test_ipmi_192.168.3.195 告答 ["005b", "05/10/2024 11:28:17", " Presence Detected", "Assertion",	DIMM110	(0x47)", " Memory"]	7s	不	.3 →	Application: 常规 PartName DIMM110

从结果中可以看到,部件名称被正则截取到标记中。

同时,只有 DIMM110 是既存在"Assertion"记录,又存在"Deassertion"记录的,所 以只有 DIMM110 部件的告警是恢复了的。

优点:

- 1. 配置简单, 仅配置一条即可
- 2. 告警事件不遗漏, 多个部件告警信息, 则产生多个告警事件
- 3. 可以实现单个部件的告警、恢复记录的关联,不会因为其他此部件的恢复记录,触发其

他部件告警的恢复操作

缺点:

1. 配置逻辑较复杂,涉及标记、正则、内置宏等多方面

方法二:

配置单个触发器,如下图2

- ① 将告警最新值带进告警标题中,区分具体告警部件等信息。
- ② 检测到关键字"Assertion"则告警
- ③ 检测到关键字"Deassertion"则恢复
- ④ 问题事件生成模式:多重:触发器未恢复的情况下可以多次产生告警;单个:多次规则
 匹配,如果已经产生的告警为恢复的情况下,不会重新产生新告警。效果如下图 3

以及諸	
有模板 / 硬件IPMI[通用] 监控项 9 触发	器 3 图形 1 仪表盘 自动发现规则 7 Web 场景
KARIF TUNIL WORKCAR	
* 名称	[硬件]{HOST.NAME} 告督 {ITEM.VALUE}
Event name	[硬件]{HOST.NAME} 告警 {ITEM.VALUE}
Operational data	
严重性	未知 信息 <mark>警告</mark> 次要 严重 紧急
* 问题表现形式	find(/LW Template for IFMI all ② server/event_logs_json,,"like","Assertion")=1 流加
事件成功迭代	<u>表达式称语器</u> 表达式 恢复表达式 无
* 恢复表达式	find(/LW Template for IPMI all server/event_logs_json,,"like" "Deassertion")=1 添加
	表达式构造器
问题事件生成模式	单个 多重 ④
事件成功关闭	所有问题 所有问题如果标签值匹配
允许手动关闭	
URL	
描述	
Jan A.L.	
已启用	
	更新 克隆 删除 取消

图 2

时间マ	严重性	恢复时间 状态 信	信息	主机	问题	持续时间	确认	动作	标记
10:20:26	警告	问题		test_ipmi_192.168.3.195	(硬件)test_ipmi_192.168.3.195 告替 ["0060", "05/10/2024 10.04.18", "Presence Detected", "Assertion", "DIMM131 (0x4e)", "Memory"]	2m 45s	不	•→	Application: 常规
10:20:25	양 뜸	问题		test_ipmi_192.168.3.195	[硬件]test_ipmi_192.168.3.195 告罄 ['005f', '05/10/2024 10:04:17', " Presence Detected", "Assertion", " DIMM130 (0x/40)', " Memory']	2m 46s	不	e	Application: 常規
10:20:25	警告	问题		test_ipmi_192.168.3.195	[硬件]test_ipmi_192.168.3.195 告答 ["005e", "05/10/2024 10.04:17", " Presence Detected", "Assertion", " DIMM121 (0x4b)", " Memory"]	2m 46s	不	a⇒	Application: 常規
10:20:25	警告	问题		test_ipmi_192.168.3.195	[硬件]test_ipmi_192.168.3.195 告答 ["005d", "05/10/2024 10.04:17", " Presence Detected", "Assertion", " DIMM120 (0x4a)", " Memory"]	2m 46s	不	• →	Application: 常規
10:20:25	警告	问题		test_ipmi_192.168.3.195	[硬件]test_ipmi_192.168.3.195 告答 ["005c", "05/10/2024 10:04:17", " Presence Detected", "Assertion", " DIMM111 (0x48)", " Memory"]	2m 46s	不	_2 ●→	Application: 常規
10:11:29	警告	问题		test_ipmi_192.168.3.195	[硬件]test_ipmi_192.168.3.195 告答 ["005b", "05/10/2024 10:04:17", " Presence Detected", "Assertion", " DIMM110 (0x47)", " Memory"]	11m 42s	不	2	Application: 常規

图 3

优点:

- 1. 配置简单, 仅配置一条即可
- 2. 告警事件不遗漏, 多个部件告警信息, 则产生多个告警事件

缺点:

1. 一个部件告警恢复,则其他部件告警一并恢复,如图 4 —— 仅 DIMM131 部件恢复,

其他部件的告警也被恢复了,其实并没有

问题			导出到CSV
	~ >	今年到目前为止	ト く 縮小 ->
■从 nowy	訳 は は 一周 と 一 月 と 一 一 月 と 一 一 周 と 一 一 月 し 一 一 八 月 し 二 一 八 月 し 二 一 八 月 し 二 一 八 月 し 二 一 八 月 し 二 一 八 月 し 二 一 八 月 し 二 一 八 月 し 二 一 八 月 し 二 一 八 月 し 二 一 八 月 し 二 一 八 月 し 二 一 八 月 し 二 一 八 月 し 二 一 八 月 し 二 一 八 月 し 二 一 八 月 二 一 一 一 一 一 一 一 一 一 一 一 一 一	今天 今天到目前为止 本間 本周到目前为止 本月 這个月到目前为止 本年 今年到目前为止	 还剩 5 分钟 还剩 15 分钟 还剩 30 分钟 还剩 30 分钟 还剩 3 小时 还剩 3 小时 还剩 6 小时 还剩 6 小时 还剩 12 小时 还剩 1 天
10月▼ □ 戸屋住 株規10月 秋本 信息 主10. 月間	持续时间	确认动作者	5/8
192026 · 音音 10.4331 已解決 lest.jpml_192.168.3.195 保健性test.jpml_192.168.3.195 告誓 [10660", "05/10.0224.10:04.18", "Presence Detected", "Assention", "DIMM131 (0x4e)", "Memory"]	23m 5s	⊼ .8 Ι	Application: 常规
10.2025 · 常告 10.43.31 日解決 test_jpmi_192.168.3.195 博使性test_jpmi_192.168.3.195 告告 [1005ff; "05/10.2024 10.04.17; "Presence Detected", "Assertion", "DIMM130 (0x4d;," "Memory"]	23m 6s	⊼ .≗ Ι	Application: 常规
102025 · 音音 10.43.31 日解決 test_jpm192.158.3.195 语便作test_jpm192.158.3.195 告音["0059"; "05/10/2024 10.04.17", "Presence Detected", "Assertion", "DIMM121 (0x40),", "Memory"]	23m 6s	杰 .≗ 【	Application: 常規
102225 · 回答 10.4331 已解決 lest.jpmi_192.168.3.195 (硬約]est.jpmi_192.168.3.195 告留[10564"; '05/10.2024 10.04.17"; 'Presence Detected'; 'Assertion'; 'DIMM120 (0x4a); 'Memory']	23m 6s	▲ 🕴 🛛	Application: 常規
102025 - 圖音 10.43.31 日解決 <u>test_jonn_192.168.3.195</u> [硬件[test_jonn_192.168.3.195 告音 ["0055"; "05/102024 10.04.17, " Presence Detected", "Assertion", "DIMM111 (0x49)," Memory"]	23m 6s	不 👌 🛛	Application: 常则
10.1129 日 音音 10.4331 日解決 lest.jpmi_192.168.3.195 信便件Best.jpmi_192.168.3.195 音盲[10559]" (55/10.2024 10.04.17]", "Presence Detected", "Assention", "DIMM110.(0x47)", "Memory"]	32m 2s	⊼ .≗ !	Application: 常则
10.06/20 日音 10.08/23 日解决 test_ipmi_192.168.3.195 音音 [0959', "05/10/2024 10.04-17", "Presence Detected", "Assertion", "DIMM110 (0x47)," Memory"]	2m 3s	≭ .≟	Application: 常規
0.234 2.395		显示	已自动发现的 7中的7
2024-05-10 10:43.31 ["0060", "05/10/2024 10:43:10", " Presence Detected", "Deassertion", "DIM0131 (0x40)", " Memory"]			
2024-05-10 10.20.26 ["0060", "05/10/2024 10:04:18", " Presence Detected", "Assertion", " DIMM131 (0x4e)", " Memory"]			
2024-05-10 10.20.25 ["005f", "05/10/2024 10:04:17", " Presence Detected", "Assertion", " DIMM130 (0x4d)", " Memory"]			
2024-05-10 10 20 25 ["005e", "05/10/2024 10:04:17", " Presence Detected", "Assertion", " DIM0121 (Oxtb)", " Memory"]			
2024-05-10 10:20:25 ["005d", "05/10/2024 10:04:17", " Presence Detected", "Assertion", " DIML20 (0x4a)", " Henory"]			
2024/05/10/10/2025 ["008c", "05/10/2024 10:04:17", " Presence Detected", "Assertion", " DIMMIII (0x48)", " Hemory"]			

图4

方法三:

对每个部件添加一个触发器(如下图2):

DIMM110 部件告警触发器:

触发表达式:监控值包含 DIMM110 且 包含关键字 "Assertion";

恢复表达式:监控值包含 DIMM110 且 包含关键字 "Deassertion";

反复操作,添加 DIMM111、DIMM120、DIMM121 等多个触发器。

* 名称	[硬件]{HOST.NAME} DIMM110 部件告答
Event name	[硬件]{HOST.NAME} DIMM110 部件告答
Operational data	
严重性	未知 信息 警告 次要 严重 紧急
* 问题表现形式	find(/LW Template for IPMI all server/event_logg_json,,"like","DIMM110")=1 and find(/LW Template for IPMI all server/event_logs_json,,"like","Assertion")=1
	表达式构造器
事件成功迭代	表达式 恢复表达式 无
* 恢复表达式	find(/LW Template for IPMI all server/event_logg_json,, "like", "DIMM110")=1 and find(/LW Template for IPMI all server/event_logg_json,, "like", "Deassertion")=1
	表达式构造器
问题事件生成模式	单个多重
事件成功关闭	所有问题 所有问题如果标签值匹配
允许手动关闭	
URL	
描述	
已启用	\checkmark

图 2

优点:

 可以实现单个部件的告警、恢复记录的关联,不会因为其他此部件的恢复记录,触发其 他部件告警的恢复操作

缺点:

- 1. 配置工作量巨大,每个部件都需要定义一个对应触发器
- 2. 可能会丢失、遗漏告警,因为可能部件关键字未加入触发器中

结论:

上述三种方法可以看出,逻辑上方法二、方法三更加简单明了,但是皆有不满足场景需求的 情况;方法一则更贴合场景需求,且善用触发器的标记功能,也更利于监控平台的维护管理。 参考该第一方法,可延伸较多场景,如日志事件告警恢复 ID 关联、snmptrap 端口 up\down 数据告警关联、硬件事件相同部件名告警恢复关联、远程登入登出记录关联等。

以上就是这一期 Zabbix 技术分享。大家好,我是乐乐,专注运维技术研究与分享,关注我 学习更多 Zabbix 使用技巧,更多问题也欢迎到乐维社区进行留言。

三十八、zabbix 监控进阶:如何分时段设置不同告警阈值 (多阈值告警)

在生产环境中,企业的业务系统状态并不是一成不变的。在业务高峰时段,如节假日、 促销活动或特定时间段,系统负载和用户访问量会大幅增加,此时可能需要设置更高的告警 阈值来适应更高的负载,反之,低谷期则要将告警阈值调低。

实践中,针对不同的业务状态调高或调低告警阈值,可以对 zabbix 配置多个触发器, 设定在不同的时间段生效来实现。本文将以配置两个时段为例,讲解如何分时段设置不同告 警阈值。



1. 创建触发器

2. 触发器配置

2.1. 配置第一个触发器

假定 0-8 点为企业业务低谷,期间业务系统 CPU 使用率一般不超过 75%,可以将触发器配置为:当 CPU 使用率超过 75%时即触发告警。

触发器		
所有主机 / 监控系统-Server 已启	用 ZBX 监控项 140 触发器 61 图形 19 自动发现规则 4 Web 场景	
触发器 标记 依赖关系		
* 名称	{HOST.NAME} CPU使用率在0-8点期间大于75%	
Event name	(HOST.NAME) CPU使用率在0-8点期间大于75%	
Operational data		
严重性	未知 信息 警告 次要 严重 紧急	
*表达式	<pre>now()>=000000 and now()<=080000 and last(/itops- server/system.cpu.util[Usage])>75</pre>	添加
	表达式构造器	
事件成功迭代	表达式 恢复表达式 无	
问题事件生成模式	单个 多重	
事件成功关闭	所有问题 所有问题如果标签值匹配	
允许手动关闭		
URL		
描述		
已启用		
	添加 取消	

表达式:

now()>=000000	and	now()<=080000	and
last(/itops-server/system.	cpu.util[Usage])>75	
表达式含义:			
now()>=000000 #当前时间	同大于 0 点时触发	〔(时间格式为 HHMMSS)	
now()<=080000 #当前时间	回小于 8 点时触发	克(时间格式为 HHMMSS)	

last(/itops-server/system.cpu.util[Usage])>75 #主机 CPU 使用率指标最新取值大于 75

时触发

注: 表达式用 and 将各个函数串联, 所有条件满足时触发器才会触发告警

2.2. 配置第二个触发器

假定 8-24 点为企业业务高峰,期间业务系统 CPU 使用率可能接近或短暂超过 90%,可以将触发器配置为:当 CPU 使用率超过 90%时触发告警。该触发器与第一个类似,只需修改时段、触发阈值。

41	4200	
ШH	17 25	
10-		

所有主机 / 监控系统-Server 已启	用 ZBX 监控项 140 触发器 63 图形 19 自动发现规则 4 Web 场景
触发器 标记 依赖关系	
* 名称	(HOST.NAME) CPU使用率在8-24点期间大于90%
Event name	{HOST.NAME} CPU使用率在8-24点期间大于90%
Operational data	
严重性	未知 信息 警告 次要 严重 紧急
" 表达式	time()>080000 and time()<=235959 and last(/itops- server/system.cpu.util[Usage])>90
	表达式构造器
事件成功迭代	表达式 恢复表达式 无
问题事件生成模式	单个 多重
事件成功关闭	所有问题 所有问题如果标签值匹配
允许手动关闭	
URL	
描述	
已启用	✓ 更新 克隆 删除 取消

表达式:

time()>080000	and	time()<=235959	and

last(/itops-server/system.cpu.util[Usage])>90

当完成以上配置后, 主机 CPU 使用率在 0-8 点期间大于 75%时告警, 在 8-24 点期间大于

90%时告警,至此完成分时段不同告警阈值的设置。

以上就是本期的全部内容。大家好,我是乐乐专注 IT 运维技术研究与分享,更多 zabbix 等开源监控工具使用技巧欢迎关注乐维社区,更多运维问题也欢迎留言提问。

三十九、如何生成好看的 zabbix 告警报表并发送邮件

一、场景模拟

小东是一名资深的 IT 运维人员,其直属领导想要了解公司业务系统的健康状态以及小 东日常的工作情况等,要求小东每周统计系统告警情况并发邮件给到他。小东所在公司搭建 了一套 zabbix 开源监控,于是小东利用自己的专业知识,实现了 zabbix 告警统计,并生 成漂亮的告警报表发送给自己的领导,得到了领导的称赞。

Zabbix 开源监控是 IT 监控领域的佼佼者,拥有强大的告警统计与报表生成能力,但对于刚上手的小白来说可能还有些难度,本文将详细介绍 zabbix 告警报表的生成近程及发送邮件的操作步骤。

二、实现原理

环境说明:本人部署的 zabbix 版本为 6.0 加 postgresql14.4 数据库,如果用 mysql 的话, 查询语句可能不一致

1、数据来源

基于 python 脚本实现,安装 psqcopg2 库,查询 pg 数据库的数据,主要是统计出最近一个月内告警出现最多的触发器,返回 10-20 条数据即可。统计最近一个月内哪些对象出现的告警次数比较多,我个人的话,返回的数据是 1000 条

2、数据清理

把主要是统计出最近一个月内告警出现最多的触发器的数据整理成一个列表数据,数据格式 如下

[{ "告警标签" },{ "触发器名称" },{ "告警次数" },{ "告警颜色" }]

统计最近一个周内哪些对象出现的告警次数比较多(本人加了限制,就是一个对象同一个触发器起码一周内出现超过 5 次才统计),我个人的话,返回的数据是 1000 条 [{"对象名称"},{"对象 ip"},{"触发器名称"},{"告警出现次数"}]

3、脚本实现过程

通过 python 脚本把最近一个月告警出现最多的触发器动态封装成 html 发到邮件正文。把 一周内哪些对象出现的告警次数多的填充到 excel 表,并作为附件发送到邮件

三、实现步骤

需要安装一下 python 库



from email.mime.multipart import MIMEMultipart

from email.mime.text import MIMEText

from email.mime.base import MIMEBase

from email import encoders

import os

#!/usr/bin/env python3

```
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
import psycopg2
import openpyxl
from openpyxl.styles import Alignment, Font
from openpyxl.utils import get_column_letter
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEText
from email.mime.base import MIMEBase
from email import encoders
import os
```

```
部分代码
```

Postgresql 数据库连接代码

```
config = {
    'host': '数据库 ip',
    'port': 5432, # PostgreSQL 默认端口是 5432
    'user': '用户名',
    'password': '密码',
    'database': '数据库名'
```

try:

connection = psycopg2.connect(**config)

with connection.cursor() as cursor:

sql_query="""SELECT t.description AS trigger_name, t.priority AS trigger_level, COUNT(e.eventid) AS occurrence_count FROM events e JOIN triggers t ON e.objectid = t.triggerid WHERE e.source = 0 AND e.object = 0 AND e.clock > = EXTRACT(EPOCH FROM (NOW() - INTERVAL '1 WEEK')) GROUP BY t.description, t.priority ORDER BY occurrence_count DESC LIMIT 100;"""

cursor.execute(sql_query)

results = cursor.fetchall()

for row in results:

excel_data.append([row[0],row[1],row[2],row[3]])

except Exception as e:

Print(e)



数据清洗封装动态 html 部分代码(有些 css 样式在邮件会不生效,所以可能需要用内嵌

style)

table_rows = ""			
for trigger in trigger_stats:			
table_rows += " <td< td=""></td<>			
style='width:8%;border: 1px solid #999;text-align:center;padding: 5px 0;'> <span< td=""></span<>			
style='background-color: {}; padding: 2px 6px; border-radius: 3px;			
color:white;'>{} <td style="width:10%;border: 1px solid</td></tr><tr><td>#999;text-align:center;padding: 5px 0;color: #f40;">{}</td>	{}		
style='width:10%;border: 1px solid #999;text-align:center;padding: 5px			
0;'>{}".format(trigger['color'],trigger['level'],trigger["name"],			
trigger["count"])			

Excel 数据填充部分代码(data 为清理好的数据)

def generate_excel(data):
创建一个新的工作簿 # 创建一个新的工作簿
wb = openpyxl.Workbook()
sheet = wb.active
sheet.title = '告警报表'
headers = ['业务名称', 'IP', '告警信息', '告警次数']
写入表头 # 写入表头
for col_idx, header in enumerate(headers, start=1):
cell = sheet.cell(row=1, column=col_idx)
cell.value = header
cell.font = Font(bold=True)
cell.alignment = Alignment(horizontal='center')
填充数据
for row_idx, row_data in enumerate(data, start=2):

for col_idx, value in enumerate(row_data, start=1):

cell = sheet.cell(row=row_idx, column=col_idx)

cell.value = value

print(cell.value)

保存工作簿

excel_file = './alert_data.xlsx'

wb.save(os.path.basename(excel_file))

```
}
def generate_excel(data):
    # 创建一个新的工作簿
   wb = openpyxl.Workbook()
    sheet = wb.active
    sheet.title = '告警报表'
   # 定义表头
   headers = ['业务名称', 'IP', '告警信息', '告警次数']
   # 写入表头
    for col_idx, header in enumerate(headers, start=1):
        cell = sheet.cell(row=1, column=col_idx)
        cell.value = header
        cell.font = Font(bold=True)
        cell.alignment = Alignment(horizontal='center')
   # 填充数据
    for row_idx, row_data in enumerate(data, start=2):
        for col_idx, value in enumerate(row_data, start=1):
           cell = sheet.cell(row=row idx, column=col idx)
           cell.value = value
           print(cell.value)
   #保存工作簿
    excel_file = './alert_data.xlsx'
   wb.save(os.path.basename(excel_file))
```

邮件发送部分代码

```
# 创建 MIMEMultipart 对象
    msg = MIMEMultipart('alternative')
    msg['From'] = sender email
    msg['To'] = receiver_email
    msg['Subject'] = subject
    # 附加 HTML 内容
    msg.attach(MIMEText(html_content, 'html'))
    generate_excel(excel_data)
   with open('./alert data.xlsx', 'rb') as attachment:
        part = MIMEBase('application', 'octet-stream')
        part.set_payload(attachment.read())
    encoders.encode_base64(part)
    part.add_header('Content-Disposition',
                                                                    'attachment;
filename="{}"'.format('./alert_data.xlsx'))
    msg.attach(part)
   try:
        # 连接到 QQ 邮箱的 SMTP 服务器
        server = smtplib.SMTP_SSL('smtp.qq.com', 465)
        server.login(sender_email, sender_password)
```



四、成果演示

1、邮件一个月的告警统计报表

8.2	1.2.1	COL 1 1 1 1 2 2 1 (1 2 4	
收任	#人:		0@qq.com>
附	件:	1 个 (III alert_data	.xlsx)

告警统计

以下是最近一个月告警的统计信息:

告警等级	告警名称	统计次数
严重	[操作系统] { 的主动监控模式无法及时采集 到数据	2039
严重	[网络设备] { [{] [{] }] CPU 使用率平均3分钟高于90%	1085
严重	[网络链路] {HOST.NAME} 设备端口 { }无法采集到 带宽数据	451
警告	[网络设备]版卡[、CPU 使用率平均3分钟高于80%	308
严重	[网络设备] SNMP采集中断	129
严重	[网络设备]、 网络 医马马卡联,持续3分钟未响应	62
严重	[网络链路][设备端口{\$PORT}带 宽接收利用率连续3次取值大于0.01%	8
严重		8
严重	[硬件设备] { }失联,持续3分钟未响应,设备可 能宕机	2

2、告警统计 excel 邮件附件(部分宏值需要额外替换成具体值)

	日 ちっ ご ・ = alert_data (1).xlsx [受保护的视题] - Excel(产品激活失敗)						(活失败)			
3	2件 开始	插入页面布局	公司	式 数据	审阅	视图 P	DF工具集	百度网盘	♀告诉我您想要做什么 【	<u>З ф •, (</u>
Ĥ	受保护的视	图 请注意 - 来自 Inte	rnet 的	文件可能包含	病毒。	能忽需要编	長 、否则保	侍在受保护初	图中比较安全。 	
A	A1 ~ :: × ✓ f 业务名称									
		٨			D				C	D
1	J	A (冬夕な			TP				告察信自	生態水粉
2	192 168	3	192 -	38			「操作系	统]192	38的主动监控模式无法及时采集到数据	665
3	192 168	12 1	192	202			「网络设	各]192	202 1板卡[#SNMPVALUE]]CPU使用率平均3分	380
4	192, 168, 3	0-GigabitEthe	192	. 200			网络链	路]192	200-GigabitEthernet0-0-11设备端口(\$PORT	139
5	192, 168, 2	2 1	192	, 202			网络设	备]192	202 1板卡[{#SNMPVALUE}]CPU使用率平均3分	90
6	192.168.)-AC	19:	3.200			[网络设	备]192	200-AC失联, 持续3分钟未响应	48
7	192.168.	-AC	192	3. 200			[网络设	备]19:	200-ACSNMP采集中断	32
8	192.168.3	-H3C	192.	3. 203			[网络设	备]19	. 203-H3C失联, 持续3分钟未响应	16
9	192. 168. 3		192.1	1. 206			[网络设	备]19	. 206失联, 持续3分钟未响应	16
10	A6005		192. :	. 208			[网络设	备]A6	,持续3分钟未响应	8
11	192.16	1	192. :	201			[硬件设	备]19	3. 201失联, 持续3分钟未响应, 设备可能宕机	8
12	192.16	2-Ethernet0-0	192. :	202			[网络链	路]19	1. 202-Ethernet0-0-3设备端口 {\$PORT} 无法采身	8
13	192.16		192. :	38			[操作系	统]19	2.38失联,持续3分钟未响应,系统可能宕机	8
14	192. 1	2_1	192.	202			[网络设	备]19	1.202_1失联,持续3分钟未响应	8
15	192.1	:01-AC	192.	201			[网络设	备]192.	201-AC失联,持续3分钟未响应	8
16	192.16	:01-AC7	192.	. 201			[网络设	备]192.	201-AC7失联,持续3分钟未响应	8
17	192.16	202-Ethernet0-0	192.	. 202			[网络链	路]192.	202-Ethernet0-0-20设备端口{\$PORT}无法采	8
18										



监控配置

四十三、Zabbix 监控 Spark 中间件配置教程

本文将介绍以 JMX 方式监控 Spark 中间件。JMX 具有跨平台、灵活性强、监控能力强、 易于集成与扩展、图形化界面支持以及安全性与可配置性等多方面的优势,是监控 Spark 等复杂 Java 应用程序的重要工具之一。

Apache Spark 是一个开源的大数据处理框架,它提供了快速、通用和可扩展的数据处理能力,适用于执行大规模的数据处理和分析任务,特别是在批处理、实时流处理、机器学习和图计算等领域。

JMX(Java Management Extensions)作为 Java 平台标准的一部分,提供了一种标 准化的机制,用于监控和管理应用程序、系统对象、设备和服务。JMX 技术可以跨越不同 的操作系统、体系结构和网络传输协议,灵活地开发无缝集成的系统、网络和服务管理应用。

JMX 可以被用于多种管理任务,包括:

- 系统监控:监控系统的 CPU 使用率、内存消耗、线程数等指标。
- 性能调优:获取应用程序的性能数据,如方法执行时间、请求响应时间等,帮助开 发者找出性能瓶颈并进行优化。
- 故障排查:当系统出现故障时,通过 JMX 快速定位问题所在,如查看日志、监控线 程状态等。
- 安全管理:实现系统的安全控制,如用户认证、访问控制等。
- **服务管理**:对于分布式系统,监控服务状态、管理服务实例,提高系统的可用性和
 可维护性。

二、Spark 配置开启 JMX 服务

1、编写 Spark 配置文件:

在安装的 spark 服务里找到名称为 "spark-env.sh" 的配置文件, 进行编辑修改。

2、在 Spark 配置文件里启用 JMX:

输入命令: vi spark-env.sh 打开配置文件,并在文件中添加以下参数来开启 JMX 并设置 JMX 的监听端口。

export SPARK_DAEMON_JAVA_OPTS="\$SPARK_DAEMON_JAVA_OPTS -Dcom.sun. management.jmxremote -Dcom.sun.management.jmxremote.port=7099 -Dcom.su n.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote. ssl=false"

解析:开启 JMX 服务并设置端口为 7099。

3、重启 Spark 服务:

根据自身所搭建的 Spark 模式来重启 Spark,以便 Spark 服务应用添加这些参数的更改

三、下载测试工具 cmdline-jmxclient-0.10.3.jar 包进行连通性测试

1、下载测试工具 cmdline-jmxclient-0.10.3.jar 包

cmdline-jmxclient-0.10.3.jar 为一个测试工具,可用来测试 JMX 是否配置正确,下载 cmdline-jmxclient-0.10.3.jar(下载到任意目录)。

wget http://crawler.archive.org/cmdline-jmxclient/cmdline-jmxclient-0.10.3.jar

2、测试 JAR 包, 注意 IP 地址与 JMX 端口需与配置文件一致

java -jar cmdline-jmxclient-0.10.3.jar - 192.168.3.34:7099

3、测试成功效果图



测试成功则证明可以成功连接并且返回数据。

四、如何进行监控项添加监控

1、需要使用 windows 的 jconsole 控制台制作监控项,安装方式可以自行去搜索对应教程

进行安装。

- 2、运用方式:
 - (1) 【win+r】---【输入 jconsole】

🧧 运行				×
	Windows 将根 件夹、文档或 Ir	据你所输入 nternet 资源	的名称,为你打到 原。	开相应的程序、文
打开(<u>O</u>)	jconsole			Ý
		确定	取消	浏览(<u>B</u>)

(2) 选择远程进程, 输入 IP:JMX 端口来进行登录

▲ JConsole: 新建连接				
新建连接				
○本地进程(<u>L</u>):				
名称	PID			
sun.tools.jconsole.JConsole	37156			
 ● 远程进程(R): 192.168.3.116:12346 用法: <hostname>:<port>或 service:jmx:<pr< li=""> 用户名(U): □令(P): 连接(C) </pr<></port></hostname> 	otocol>: <sap> 取消</sap>			

(3) 添加监控项

下图是关于 Spark JMX 形式的监控项例子

ZABBIX « 5	监控项					
٩	所有機板 / 中(町件Spark(通用) 监控项 31	触发器 8 图形 仪表盘 自动发现规则 2 Web 场展				
o 1210 -	监控项 标记 1 进程	监控项 标记 1 进程				
⑦ 服务 →	1 * 名称	_{交换空间绕大小} 名称可自行定义,方便管理即可				
≔ 资产记录 ~	2 类型	JMX agent代理程序 java.lang.type=OperatingSystem,表示JMX				
11 报表 ~	3 * 谜值	JmxTjava lang type=OperatingSystem* TotalSwapSpaceSize」 TotalSwapSpaceSize 表示IMX MBean層性类型				
く 配置 -	4 信息类型	☆字(元正気) ✓ 选择对应类型的返回数据类型 下的唯一标识符				
主机群组	5 * JMX 端点	service:mx:rm:///jnd/rmi:/i/HOSTCONNI/(HOST PORT)/mxrmi 保持默认,尤需修改				
模板	用户各标					
主机	単位	B				
推护	*更新间隔	180				
关联项事件	自定义时间间隔	类型 间隔 期间 动作				
自动发现		現在 505 1-7,00:00-24:00 修務				
• 管理 •		満加				
	*历史数据保留时长	Do not keep history Storage period 90d				
	* 趋勢存储时间	Do not keep trends Storage period 365d				
	值映射	在此施入撤票 选择				
	填入主机资产纪录栏位	- 赤・ 、 、 、 、 、 、 、 、 、 、 、 、 、 、 、 、 、 、				
	描述	Committed/VirtualMemorySize				
9 支持						
Integrations	已启用					
? 帮助		添加 影就 取消				
Ilear cattings	-					

四十四、如何配置 SNMPTrap 监控

SNMP 是一种网络管理协议,SNMPTrap 则是基于此协议的一种数据传输方式。在 Zabbix 监控实践中,SNMP 的应用非常广泛,下面我们来看一下如何进行 SNMPTrap 配 置监控。

当我们获取到设备发送过来的 trap 的时候,需要是从 trap 信息中获取到 Trap OID,

如下图,这里我们会用到一个工具叫 MIB Browser。

以华为 RH5885V3 为例:

1、定位 TrapOID

首先看到 snmpTrapOID (横线部分),对应的 value,就是我们所需要的 TrapOID 了

(方框部分)。

14:40:29 2019/11/20 ZBXTRAP 192.	168.22.19	
PDU INFO:		
notificationtype	TRAP	
version	1	
receivedfrom	UDP: [192.168.22.19]:52	791->[10.142.88.81]
errorstatus	0	
messageid	Θ	
community	TrapHuawei12#\$	
transactionid	2478220	
errorindex	Θ	
requestid	2019432088	
VARBINDS:		
DISMAN-EVENT-MIB::sysUpTimeIns	tance type=67 value=Time	ticks: (225886099) 26 days, 3:27:40.99
SNMPv2-MIB::snmpTrapOID.0	type=6 value=0ID: SNMP	v2-SMI::enterprises.2011.2.235.1.1.500.10.7.81
SNMPv2-SMI::enterprises.2011.2	.235.1.1.500.1.1 type=2	value=INTEGER: 9628
SNMPv2-SMI::enterprises.2011.2	.235.1.1.500.1.2 type=4	value=STRING: "CPU2 Status"
SNMPv2-SMI::enterprises.2011.2	.235.1.1.500.1.3 type=4	value=STRING: "Configuration error"
SNMPv2-SMI::enterprises.2011.2	.235.1.1.500.1.4 type=2	value=INTEGER: 4
SNMPv2-SMI::enterprises.2011.2	.235.1.1.500.1.5 type=4	value=STRING: "0x0705ffff"
SNMPv2-SMI::enterprises.2011.2	.235.1.1.500.1.6 type=2	value=INTEGER: 255
SNMPv2-SMI::enterprises.2011.2	.235.1.1.500.1.7 type=2	value=INTEGER: 255
SNMPv2-SMI::enterprises.2011.2	.235.1.1.500.1.8 type=4	value=STRING: "022CUU10F1000361"
_ SNMPv2-SMI::enterprises.2011.2	.235.1.1.500.1.9 type=4	value=""

2、TrapOID 解析

打开 MIB Browser 浏览器,加载华为 RH5885V3 的 MIB 文件,并复制 TrapOID 数字 部分,即: 2011.2.235.1.1.500.10.7.81,然后到 MIB Browser 浏览器上 ctrl + F 进行搜索,
就可以知道是什么部件发生了告警,如下图可得知 CPU 出现故障。

Address:	✓ Advanced 0ID: .1.3.6.1.4.1.2011.	2. 235. 1. 1. 500. 10. 7. 81	
SIMP MIBs		Result Table	
r1≉ (1 -	<pre>ap</pre>	IIame/OID Find objects in MIB tree	×
Name ha	*CPIICfgError	rind what. 2011.2.235.1.1.500.10.7.81	Hext
OID .1	1. 3. 6. 1. 4. 1. 2011. 2. 235. 1. 1. 500. 10. 7. 81	March whole word only Can	cel
MIB HU	JAWEI-SERVER-IMANA-MIB		
Access			
Status cu	urrent		
DefVal			
Objects ba Descr Cl	Itepieg_ heltepiererlisme, hwlrspEvent, hwlrspSeverity PU configure error 描述		

如果我们想知道更详细的信息,可以用同样的方式,复制后面的数字 OID 到 MIB Browser 浏览器里面进行搜索。例如我不知道'CPU2 Status'代表的是什么信息,可以复 制下 OID,然后搜索一下,可得知'CPU2 Status'是传感器名,如下图所示。

ddress:	 Advanced OID: .1.3.6.1 	1.4.1.2011.2.	235. 1. 1. 500. 1. 2. 0	
SIMP MIBs			Result Table	
	hwTrap hwTrapVar hwTrapSeq hwTrapSensorName hwTrapSeverity hwTrapEventCode hwTrapEventData2 hwTrapEventData3 hwTrapEvertIata3 hwTrapLocation hwFrapLocation hwServerTRAPObject		I Name/OID	
¢	Hr⊂PU Fr⊂∮hrCPUCATError	`	Find objects in MIB tree Pind what: 2011.2.235.1.1.500.1.2 Find	Next
ame ID IB	hwTrapSensorName .1.3.6.1.4.1.2011.2.235.1.1.500.1.2 HUATEL-GEBUED-IMANA-MTE		Match whole word only Can	cel
vntax	DISPLAYSTRING			
ccess	read-only			
tatus efVal nderes	current			
escr	Trap sensor name 描述			

3、Zabbix 创建 Trap 监控项

到 zabbix 上的模板创建监控项,如下图:

进入模板,点击右上角创建监控项

				Q,	Z Share	?	•	С
					2	za	abbix-s	erver
						С. Ан	は手にたち	क्षात
)×±m1	E9X
≝所有 ▼	状态	所有	¥					
a.	状态	所有						
3 ⁶	触发器	所有	۲					
	模板	所有	•					
		: 1.1						
時呈								
名称 CPU故障	→ 自	定义名	称					
类型 SNMP trap	•		米刑	미히	送 年 SNIM	Dtr	an	
键值 snmptrap[2011.2.23	35.1.1.500.10.7.81]		大王		选择	I U	ap	
信息类型 字符 ▼		1	各式:	snr	nptrap	告聲		D]
历史致谐保留时长 90d 音奏值 石杰					▼ 展示值吨时			
新的应用集 CPU (息类型面	「字符」	文木					
应用集 -无-					740			
填入主机资产纪录栏位 -无-	¥	_						

填好对应信息后,最下方点击'添加'即可。

名称▲	触发器	键值	间隔	历史记录	趋势	类型
CPU故障		snmptrap[2011.2.235.1.1.500.10.7.81]		90d		SNMP trap

4、Zabbix 创建 Trap 触发器

创建完监控项之后,我们开始创建触发器。

点击模板'触发器'——'创建触发器'

₹ 报表 配置 管理		ଦ, 🖬	ihare ? 💄 🖱
关联项事件 自动发现 服务			zabbix-server
鼎將且 所有 ▼	主机 <11173>test-allen		▼ 创建加速发展
1 监控项1 触发器 图形 聚合图形 自动发现规则1 Web 场易	96 A		2
严重性 所有 未分类 信息 状态 所有 正常 未知的 状态 所有 已启用 停用的	警告 一般严重 严重 灾难		
应用	重设		
名称▲	表达式	状态	
未发现	迷时居		

点击'添加',触发器名称、严重性、问题事件生成模式可自定义。

名称	CPU故障	
严重性	未分类 信息 警告 一般严重 严重 灾难	
表达式	1	添加
		1
	表达式构造器	
件成功迭代	表达式恢复表达式无	
件生成模式	单个 多重	
件成功关闭	所有问题 所有问题如果标签值匹配	
标记	标记	移除
		10.10

选择要做触发器的监控项与触发器功能:

DE VELLEMAND CHARLE

THE DESIGNATION OF

:控项	<11173>test-allen: CPU故庫 1	选择
功能	最新的T值=N	•
ጠኅ	期间 T 的值总数 > N 期间 T 的值总数不是 N	•
排 班	期间T的平均值 < N 期间T的平均值 = N	
Ν	期间T的平均值 > N 期间T的平均值不是 N	
	期间T的最大与最小值的差 < N 期间T的最大与最小值的差 = N	
	期间T的最大与最小值的差 > N 期间T的最大与最小值的差不是 N	_
	期间T的最新值匹配正则表达式V,则N=1,否则N=0 期间T的最新值匹配正则表达式V,则N=1,否则N=0(不区分大小写)	
	期间T的最新值匹配正则表达式V,则N非1,否则N=0 期间T的最新值匹配正则表达式V,则N非1,否则N=0(不区分大小写)	2 这四个功能都可以选,这里使 个做示范
	期间 T 的 百分比 P 数据的值 < N 期间 T 的百分比 P 数据的值 = N	
	期间 T 的百分比 P 数据的值 > N 期间 T 的百分比 P 数据的值不等于 N	
	查找字符串V的最近值.如果找到则N=1,否则N=0	

填写好对应信息后,点击'插入',恢复表达式同理。



勾选'允许手动关闭',填写描述,最后勾选'启用',点击'添加'即可。

名称	CPU故障		
严重性	未分类 信息 警告 一般	严重 严重 灾难	
表达式	{test- allen:snmptrap[2011.2.235.1.1.50) =1	D.10.7.81].regexp(2011.2.235.1.1.500.10.7.81)}	添加
	表达式构造器		
事件成功迭代	表达式恢复表达式无		
]题事件生成模式	单个 多重		
事件成功关闭	所有问题 所有问题如果标签值	匹費已	
标记	标记	值	移除
	添加		
允许手动关闭	×		
URL			
描述	CPU故障		
- N-1			1
完成后如下:			
一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一	な▲ 表达式		
	の の の の で の の の の の の の の の の の の の の の	11.2.235.1.1.500.10.7.811.regexp(2011.2.235.1.1.500.10.7.)	31))≔1

5、snmptrap.fallback

此监控项可以接收所有设备发送过来的 snmptrap 信息。也就是一个监控项对应多个

触发器。

名称	设备发生故障	
类型	SNMP trap	
键值	snmptrap.fallback	选择
信息类型	字符 ▼	
历史数据保留时长	90d	
查看值	不变	▼展示值映射
新的应用集		

以上就是这一期的Zabbix技术分享。

大家好,我是乐乐,关注我,学习更多 Zabbix 使用小技巧,如在 Zabbix 使用过程中 碰到问题,还可以到<u>乐维社区</u>进行留言提问。

SNMP trap 官方参考文档:

https://www.zabbix.com/documentation/3.4/zh/manual/config/items/itemtypes/s

<u>nmptrap</u>

四十五、使用 Zabbix SNMP 添加自定义 OID

1、编写脚本(脚本内容如下)

#!/bin/sh
php_conn=`/bin/ps aux /bin/grep nginx egrep -v 'grep' wc -l`
echo \$php_conn
<pre>[root@itim_6 ~]# cat /root/php_online.sh #!/bin/sh php_conn=`/bin/ps aux /bin/grep nginx egrep -v 'grep' wc -l` echo \$php_conn [root@itim 6 ~]#</pre>

2、确认 oid 是否被系统占用,如.1.3.6.1.4.1.2023.6900

[root@itim_6 ~]# snmpwalk -v 2c -c public 192.168.46.188 .1.3.6.1.4.1.2023.690

[root@itim_6 ~]# snmpwalk -v 2c -c public 192.168.46.188 .1.3.6.1.4.1.2023.6900 SNMPv2-SMI::enterprises.2023.6900 = No Such Object available on this agent at this OID [root@itim_6 ~]# []

通过上图可知 oid 不存在

3、配置 snmpd

[root@itim_6 ~]# vim /etc/snmp/snmpd.conf

在 snmpd.conf 配置文件末尾插入如下内容:

extend .1.3.6.1.4.1.2023.6900 php /bin/sh /root/php_online.sh



保存配置文件并退出, 重新启动 snmp 服务

[root@itim_6 ~]# systemctl restart snmpd.service

4、测试自定义 OID

[root@itim_6 ~]# snmpwalk -v 2c -c public 192.168.46.188 .1.3.6.1.4.1.2023.690
0
[root@itim_6 ~]# snmpwalk -v 2c -c public 192.168.46.188 .1.3.6.1.4.1.2023.6900
SNMPv2-SMI::enterprises.2023.6900.1.0 = INTEGER: 1
SNMPv2-SMI::enterprises.2023.6900.2.1.2.3.112.104.112 = STRING: "/bin/sh"
SNMPv2-SMI::enterprises.2023.6900.2.1.3.3.112.104.112 = STRING: "/root/php_online.sh"
SNMPv2-SMI::enterprises.2023.6900.2.1.4.3.112.104.112 = ""
SNMPv2-SMI::enterprises.2023.6900.2.1.5.3.112.104.112 = INTEGER: 5
SNMPv2-SMI::enterprises.2023.6900.2.1.6.3.112.104.112 = INTEGER: 1
SNMPv2-SMI::enterprises.2023.6900.2.1.7.3.112.104.112 = INTEGER: 1
SNMPv2-SMI::enterprises.2023.6900.2.1.20.3.112.104.112 = INTEGER: 4
SNMPv2-SMI::enterprises.2023.6900.2.1.21.3.112.104.112 = INTEGER: 1
SNMPv2-SMI::enterprises.2023.6900.3.1.1.3.112.104.112 = STRING: "11"
SNMPv2-SMI::enterprises.2023.6900.3.1.2.3.112.104.112 = STRING: "11"
SNMPv2-SMI::enterprises.2023.6900.3.1.3.3.112.104.112 = INTEGER: 1
SNMPv2-SMI::enterprises.2023.6900.3.1.4.3.112.104.112 = INTEGER: 0
SNMPv2-SMI::enterprises.2023.6900.4.1.2.3.112.104.112.1 = STRING: "11"
[rootditim 6 ~]#

以上为获取自定义 oid 的所有数据,最后一行是我们需要获取的数据,在 zabbix web 界面 中填写的 oid 就是:..1.3.6.1.4.1.2023.6900.4.1.2.3.112.104.112.1 5、创建 snmp agent 类型监控项

		0746-3-40 × 2246-8746-98 2746-80	
ZABB	IX « 🗉	87月王61 / 辺塘服労器 已信用	SINIANZ BATY 90 RECEIVES SU DECITO 12 HEISTIZISUSUUU S WED TAIDE
	م	<u> 田代明</u> (1710) 近程	
		* 名称	监控PHP进程
• maa		検型	SNI/IP agent
③服务		* 線価	php_online 选择
≡ 资产	记录	信息类型	数字(元正氏) ~
11 报表		* 主机接口	3,188,181
く 配置		* SNMP OID	1.3 8 1 4 1 2023 6900 4 1 2 3 112 104 112 1
主机	#18	単位	
模板		* 更新间隔	1m
主机		自定义时间间隔	类型 间隔 期间 动作
₩₽P			<u>現時</u> 網底 50s 1-7,00:00-24:00 18 秒
动作			1800
关联项	页事件	*历史数据保留时长	Do not keep history Storage period 90d
自动发	比现	* 趋势存储时间	Do not keep trends Storage period 365d
2 管理		值映射	在此输入现象 选择
		填入主机资产纪录栏位	元 ~
Zi Integ	rations	描述	
? 帮助			
LUSET	settings ·		A
d) iBHI		已启用	

6、验证是否正常获取数据

Z	ABBIX	« 🖸	最新数据					50
		۹	< 7					~ >
0	监测			主机群组 在此输入搜索		选择 标记	与/成(默认) 载	
	仪表板			主机 运输服务器 ×		选择	「 标记 包含 ~ 」 値	移除
	问题			在此输入搜索			澤加	
	±€L			名称 PHP		显示标签	无 1 2 3 标签名 填满 缩短 无	
	最新数据					标签显示优先级	以逗号分隔的列索	
	拓扑圈					查看细节		
	自动发现							
Ō	服务					Save as	重设	
:=	资产记录		过滤器只影响过滤	后的数据				
_			主机	名称▲	间隔 历 趋势 类型	最近检查记录 最新数据	夏政 标记	信息
	报表		运维服务器	监控PHP进程	1m 7d 365d SNMP	°a 11s 11		图形
٩	RE			hub_ounte				
	管理							显示 已自动发现的 1中的1
			0选择 显示堆叠数1	居图 显示数据图				

至此,Zabbix SNMP 添加自定义 OID 已完成

四十六、更多……

数据库

五十、如何使用 Zabbix 监控 TiDB 数据库?

概述

TiDB 数据库是一个常见的开源分布式关系型数据库,通过使用分布式事务、分布式 SQL 引擎和分布式存储引擎来实现高可用性和横向扩展性。而 Docker 则是一个开源的容 器化平台,它可以帮助开发者在不同的环境中轻松地部署和运行应用程序。

本文将介绍如何使用 Docker 快速安装和配置 TiDB, 并使用 Zabbix 监控 TiDB。

安装步骤

1、安装 docker 并拉取镜像。

yum install docker

systemctl start docker

docker search pingcap/tidb # 搜索镜像,如果搜索不到需要设置 docker 仓库源

[root@cenos]	7 ~]# docker search pingcap/tidb				
INDEX	NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
docker.io	docker.io/pingcap/tidb	TiDB docker for PingCAP.	105		[0K]
docker.io	docker.io/pingcap/tidb-operator		5		
docker.io	docker.io/pingcap/tidb-binlog		4		
docker.io	docker.io/pingcap/tidb-arm64		1		
docker.io	docker.io/pingcap/tidb-backup-manager		1		
docker.io	docker.io/pingcap/tidb-backup-manager-arm64		Θ		
docker.io	docker.io/pingcap/tidb-binlog-arm64		Θ		
docker.io	docker.io/pingcap/tidb-binlog-enterprise		Θ		
docker.io	docker.io/pingcap/tidb-cloud-backup		Θ		
docker.io	docker.io/pingcap/tidb-control		Θ		
docker.io	docker.io/pingcap/tidb-dashboard-installer		Θ		
docker.io	docker.io/pingcap/tidb-debug		Θ		
docker.io	docker.io/pingcap/tidb-enterprise		Θ		
docker.io	docker.io/pingcap/tidb-enterprise-tools		Θ		
docker.io	docker.io/pingcap/tidb-lightning		Θ		
docker.io	docker.io/pingcap/tidb-lightning-arm64		Θ		
docker.io	docker.io/pingcap/tidb-lightning-enterprise		Θ		
docker.io	docker.io/pingcap/tidb-monitor-initializer		Θ		
docker.io	docker.io/pingcap/tidb-monitor-initializer-arm64		Θ		
docker.io	docker.io/pingcap/tidb-monitor-initializer-enterprise		Θ		
docker.io	docker.io/pingcap/tidb-monitor-reloader		Θ		
docker.io	docker.io/pingcap/tidb-operator-arm64		Θ		
docker.io	docker.io/pingcap/tidb-patched-jira		Θ		
docker.io	docker.io/pingcap/tidb-tools		Θ		
docker.io	docker.io/pingcap/tidb-vision	TiDB cluster visualization tool	Θ		
[root@cenos	7~]#				

整理 by 乐维社区(https://forum.lwops.cn)

设置仓库源如:

vi /etc/docker/daemon.json

```
"registry-mirrors": [
```

```
" https://docker.anyhub.us.kg"
```

] }

{

systemctl restart docker

- docker pull pingcap/tidb # 拉取镜像
- docker pull pingcap/tikv # 拉取镜像,单机运行可不拉取
- docker pull pingcap/pd # 拉取镜像,单机运行可不拉取

如果需要从国内仓库源上拉取则执行:

docker pull docker.anyhub.us.kg/pingcap/tidb

docker pull docker.anyhub.us.kg/pingcap/tikv #单机运行可不拉取

docker pull docker.anyhub.us.kg/pingcap/pd #单机运行可不拉取

docker images # 查看拉取的镜像

[root@cenos7 ~]# docker images				
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
docker.anyhub.us.kg/pingcap/pd	latest	a6ed80ff7339	4 months ago	404 MB
<pre>docker.anyhub.us.kg/pingcap/tikv</pre>	latest	173ec5e96e96	4 months ago	773 MB
docker.anyhub.us.kg/pingcap/tidb	latest	bd253ee807c6	4 months ago	393 MB

2、初始化容器。

mkdir /tidb

docker run --privileged=true -d --name tidb-server -p 4000:4000 -p 10080:10080 docker.anyhub.us.kg/pingcap/tidb #创建并运行容器,将会在后台启动 一个名为 tidb-server 的容器,并将容器的 4000 端口映射到宿主机的 4000 端口、容器的 10080 端口映射到宿主机的 10080 端口。

docker ps -a #查看容器运行状态

[root@cenos7 ~]# docker ps -a COMMAND CREATED STATUS PORTS OCC AND CREATED STATUS PORTS NAMES CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES Balad#ZF44d docker.anyhub.us.kg/pingcap/tidb */tidb-server* About a minute ago Up About a minute 0.0.0.0.0.4600->4000/tcp, 0.0.0.100000->100000/tcp tidb-server [root@cenos7 ~]#
mysql -h 127.0.0.1 -P 4000 -u root -D test #使用 mysql 客户端命令尝试登录 tidb
[root@cenos7 ~]# mysql -h 127.0.0.1 -P 4000 -u root -D test Welcome to the MariaDB monitor. Commands end with ; or \g. Your MySQL connection id is 2097166 Server version: 8.0.11-TiDB-v7.5.1 TiDB Server (Apache License 2.0) Community Edition, MySQL 8.0 compatible
Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
<pre>MySQL [test]> show databases; </pre>
MySQL [test]>

监控 TiDB

1、导入监控模板

使用 Zabbix 官方提供的监控模板:

https://www.zabbix.com/cn/integrations/tidb

Z	ABBIX «	5	模板												创建模板 导入
		۹		导入								×			3 过滤器 🏹
0		4	主机职		导入文件	选择文	(件 terr	nplate_d	b_tidb_tidb_	http.yaml	4				
õ		*	链接的模		规则	Dec0		更	新现有的 信	國建新的書	關始失敗		~ 值	核除	
:=	资产记录	*	8			模板			~	~					
		•				映射值				~					
٩	配置	~	□ 名称 ▲			Templat 模板间的	e dashb 5关联	oards	~	× ×					标记
			AIX代理模板[主动模式]			监控项				~					
	1915 Z		DB2数据库题本模板6番用1			自动发现	动规则		× ×	~					
	主机					图形				~					
		5	Dell iDRAC8 SNMP通用模板[10892]			Web 场			~	~					
			Docker Prometheus监控模板[通用]									导入 取消			
	自动发现		Docker通用模板[agent2]	主机	212	脱炭	121	仪表	自动发	Web <u>ili</u>	RI				
		*			项 44	H 3	形 5	盘1	现 2						
9			ECSINER	±00	项 22	RELOCAR	BED/D	1X.RE	E8.40122.90	Webini	13				
2			ECS模板-采集器	主机	监控 项 83	触发 器 10	國形	仪表盘	自动发现	Web朏	RI				
?			EMC VNX 监控模板[通用]	主机	盛控 155 17	秋秋 1844	图 形 1	仪表盘	自动发 现 5	Web <u>能</u>	则硬件ICMP模拟通用]				
÷		4	F5负载均衡通用模模[3375]	主机	监控 項 21	<u>較</u> 設 器 10	HEI FRE	仪表盘	自动发 现 7	Web	则 网络编口通用模板64位_V2、网络设备通用信息模板V2	10 11			

2、添加 TiDB 监控

点击配置->主机->创建主机,填写主机名称,选择刚刚导入的 TiDB 监控模板,设置

一个群组。

ZAB	BIX « 🗉	主机								C	创建主机	し、自入
		New host						×			3	过滤器 🏹
o 163	d -	主机 IPMI 标记 宏	资产记录 加密 值映射									
Ó 183		* 主机名称 tidb-	-192 168 80 141					1				
≔ 资/		可见的名称 tidb-	-192.168.80.141									
16.4		模板 TiDE	B by HTTP 🗙		选择				188			
₹ 688	<u>د ا</u>	* 群组 数图	四年 (新) ×		选择							
主机		在此	(输入搜索									
校告		Interfaces	类型 IP地址 春户識 192.168.80.141	DNS名称	连续到 IP DNS	3배口 10050	武以 彩陰	4				
主 di 维护		2 名称 潜加							状态	可用性	agent 加密	信息 标记
in the second		192 摄述						·	日度用	SNMP	无	
关联		Infac							已启		无	
自动	h发现	□ <u>监控</u>		h					已启		无	
<u>ы</u> на		Bagent代理程序监测(无ag	agent代理程序) ~						用			
9 支援							5		用			
🖬 Inte		监控						15 to 1225	<u>已</u> 扇 用	ZBX	无	
? 相助		<u></u>	TN 1					50F9	28		无	
🚨 Use	er settings 🔍 🗸									50	R已自动发	1029 644896

点击宏,点击"继承以及主机宏",填写相关信息:

ZABBIX «	s ±	三机						创建主机 导入
1	۹.	N	ew host			×		过滤器 文
o 🕅	-		主机 IPMI 标记 宏 1 资产记录 加	密 值映射				
③ 服务	•		(\$TIDB OPEN FDS MAX WARN)	90	T ~ 更改 \leftarrow TIOB by HTTP "90"			
≔ 资产记录	•		Maximum percentage of used file descriptors					
🛄 报表	~		(STIDB.PORT)	10080	T * 更改 \leftarrow TiDB by HTTP: "10080"		移除	
A. 8000			The port of TiDB server metrics web endpoint					
			(\$TIDB.REGION_ERROR MAX WARN)	50	T ~ 更改 ← TiDB by HTTP: "50"			
王作品开始			Maximum number of region related errors					
0.62 ±10		_	(\$TIDB SCHEMA_LEASE_ERRORS MAX	0	T * TOR by HTTP "0"			
±01		名称	WARN)				状态 可	用性 agent 加密 信息 标记
SE 9"		192	Maximum number of schema lease errors				28 5	NMP 无
対理で表出	÷	To Bar	(\$TIDB.SCHEMA_LOAD_ERRORS.MAX.W ARN)	1	T ~ 更改 ← TiD8 by HTTP. "1"		田	673
自动发现			Maximum number of load schema errors				用	245
1 管理		1212	(\$TIDB.TIME_JUMP_BACK MAX WARN)	1	T * 更改 ← TiDB by HTTP: "1"		已启 (2)用	元
		論接	Maximum number of times that the operating	system rewinds every second			ee II	天
0 支持		Г	(\$TIDB.URL)	192.168.80.141	T > 修建 \leftarrow TiDB by HTTP: "localhost"		用	
Integrations		論控	2			-	已启 Z	X 无
? 帮助		监控	and the second second second second			添加取消	es z	33 无
				29 1			用	
 User settings 	1							显示已自动发现的 6中的6

宏说明:

{\$TIDB.PORT}: 填写 TiDB 的接口端口,不是连接用的端口,默认为 10080。

{\$TIDB.URL}:填写 TiDB 的 IP 地址。

其他宏可使用默认值。

信息填写完成后,点击添加按钮即可完成监控主机的添加。

ZABBIX « 🔊	主机						创建主机 导入
٩	New host				ж		过滤器 🏹
◎ 监测 -	主机 IPMI 标记 宏1 资产记录 加	密 值映射					
⑦ 服务 →	(\$TIDB.OPEN.FDS.MAX.WARN)	90	T ~ 更改 ← TIDB by HTTP. "90"				
≔ 资产记录 -	Maximum percentage of used file descriptors						
11 报表 🗸	(\$TIDB.PORT)	10080	T ~ 更改 ← TiDB by HTTP: "10080"			移除	
义 配置	The port of TiDB server metrics web endpoint						
+ #1.8¥40	(\$TIDB.REGION_ERROR.MAX.WARN)	50	T ← 更改 ← TIDB by HTTP: "50"				
模板	Maximum number of region related errors						
主机	(\$TIDB.SCHEMA_LEASE_ERRORS.MAX. WARN)	0	T [™] 更改 ← TiDB by HTTP: "0"				
维护	192 Maximum number of schema lease errors					秋心 可)	HE agent ZORE 1648 1942
र ग ाः >	(\$TIDB SCHEMA_LOAD_ERRORS MAX W	1	T [™] 更改 ← TIDB by HTTP: "1"			用	2741
关联项事件	Influ ARN]					見眉用	无
		4	The Way is TOO IN UTTO 117			民宿 [2]	X Z
「「「「「」」」	(a fibe, fine_doint_excertised	system rewinds every serond				用	
9 支持	(\$TIDB.URL)	192.168.80.141	T v 総計 e TIDB by HTTP: "localbost"	1	2	用	AL SC.
2 Integrations					<u> </u>	已启 20	× 无
? 帮助	10HP				添加 取消	已启 [2]	
Iser settings		测1				用	-
- User settings V							显示 已自动发现的 6中的6

添加完成后,可在主机管理界面,对 http 代理类型的监控项触发立即执行,使其快速获取 数据和创建自动发现的监控项。

Z	ABBIX « 🗈	监控项	创建监控项
	٩	所有主机 / bdb-192-198-80-141 已即用 2023 【血控系 35 (触发器 12 图形 3 自动发现规则 6 Web 场景	ilise V
0	<u></u> 塩渕 、		
ō	服务	主机 6db-192168 80.141 × 选择 信息类型 所有 ∨ 标记 包含 ∨ 低 好法	
:=	资产记录 ,	在此地入股炭 原加 原加	
	版史 -	名称 脑膀 脑膀	
4	APR 4	理査 更新削減 研約 使数 (水本 条件 已成用 停用的 (水本 条件 日成用 停用的 (水本 条件 日成用)	
Ľ	主机料组		
	模板	1111日 22 22 711日 22 22 71	
	±11 2		
	1819-	1000年 重成 过速時にの数据	
	动作 >	柳记	
	自动发现	component: appacation 0 component: connections 0 component: connections 0 component: appacation 0 component: appacation 0 component: appacation 0 component: appacations 0	
	管理 🗸	処型 4 HTTF化現 2 H生現 4 H生 4 H生	
~	4.4	信息关系	
		その10 数子(TuIISU) 文字2 みの800 有数2000	
4	Integrations	无触发器 2 有敏災器 0	
?	帮助	历史记录 92 7d0	
•	Hear eattinge		
Z	ABBIX « 🔊	2時留 更新網路 秋本 所有 已出用 得用的	
	۹	· 全体研 在市场入校会 选择 触发器 新行 是 不	
0	監測 ・	Inheritad 第月 是 不	
Ō	服务 、		
:=	资产记录	「「「「」」「「「」」」「「」」」「「」」」「「」」」「「」」」「「」」」「「」」」「」」」「」」」「」」」「」」」「」」」「」」」」	
	报表 、	162	
4	配置 ^	component appearance of component contractors of component appearance component as a component server of component means of component appearance of the component means of component mea	
	主机群组	类型 HTTP:代表 2 - 相关项目 +33	
	模板	信息炭型 学会の教学(王氏山)の文字の「伝命殿」	
	主机	TTO WT WAT VALUE V ALL PROME O	
	动作 >	天教技器2 有物力器0	
	关联项事件	02 740	
	自动发现	▶ 1 名称▲ 航波器 酸脂 间隔 历史记录 趋势 类型 状态 标记	信息
	管理	TIDB by HTTP: TIDB Get instance metrics tob get_metrics 1m 0 HTTP 代題 已印用 component raw	
9		TIDB by HTTP: TIDB: Get instance status tidb.get_status 1m 0 HTTP 代題 已居用 component health component raw	
Z	Integrations		RBS 244852
?	帮助	2 (2)# [<u>新</u> 州] <u>Execute now</u>] · 清 時初史 [夏朝] 批准更新] <u></u> 期時	
	User settings	2	
	18H	Zabbox 6.0.8 @ 2001-2022, Zabbox SM	

到此,监控配置完成。查看监控数据:

Z	ABBIX	<< 51	最新数据								55	Î
		۹	< 🗸								~ >	
0	监测			主机群组 在此输入搜索		选择	标记	与/或 (默认) 或				P
	仪表板			主机 tidb-192 168.80 14	×	选择]	标记	包含 ~ 位	8519:		
	问题			在此输入搜索			1	添加				
	主机			名称			显示标签	无 1 2 3 数	家签名 填满 缩短 无			
	最新数据						标签显示优先级	以這号分隔的列表				
	拓扑图						查看细节	×				
	日动发现											
Ø	服务		过滤器 只影响过油服	新教展			Save as	10.02				
=	资产记录		标记									
11.	报表		component 96 type 61									
a	80.00		TAG VALUES			and the second se		-	10	and C. Ministerior de C.		
_	HP1		type: batch ge	t 1 batch resolve 1 commit 1 e	pired 1 FieldList 3 get	InitDB 3 lock ke	vs 1 not expired 1 Ping	3 Query 3 query check	secondary locks 1 query resolve locks 1	guery resolve lock lite 1		
	管理		query_tx StmtExe	n_status 1 guery_bxn_status_com cute 3 StmtFetch 3 StmtPrepare	mitted 1 guery_txn_status 3 StmtReset 3 StmtSen	rolled_back 1 Qu dLongData 3 Use 1	it 3 resolve 1 resolve_a 1 wait_expired 1 write_c	sync_commit 1 resolve_fo conflict 1	r_write 1 rollback 1 Select 1 SetOption 3	Show 1 Sleep 3 StmtClose 3		
9			DATA									
17	Integration		With data 70 Without	data 26								
	integration		□ 主机	名称▲	间隔 历史 趋势	类型 最近核	金直记录 最新数据	更改	标记		信息	
?	帮助		tidb-192.168.80	TIDB: CPU	7d 365d	相关项目 27s	5.3481 9	6 -0.3577 %	component: cpu	图形		
Ē	User settin	gs v	tidb-192.168.80	TIDB: DDL waiting jobs	7d 365d	相关项目 27s	0		component: operations	國用》		
0	1901		tidb-192.168.80	TiDB: Failed Query, rate	7d 365d	相关项目 27s	0		component sgl	國用		

Z	ABBIX < 🛯	udb-192.106.60	tidb.server_query.get_metrics	74		相大坝日				component raw	<i>0</i> .0 <u>9</u>	Î
	۹	tidb-192.168.80	TiDB: Goroutine count 1	7d	365d	相关项目	27s	140	+26	component goroutines	間形	
0	监测 ^	tidb-192.168.80	TIDB: Heap memory usa	7d	365d	相关项目	27s	376.03 MB	+18.25 MB	component: memory	图形	
	(父表版 回题	tidb-192.168.80	TiDB: Keep alive, rate	7d	365d	相关项目				component: health	图形	
	主机	tidb-192.168.80	TIDB: KV backoff, rate	7d	365d	相关项目				component: tikv-com	图形	
	最新数据	tidb-192.168.80	TiDB: KV commands, rate	7d	365d	相关项目	27s	0.7754 Ops	+0.03918 Ops	component: tikv-com	國形	
	自动发现	tidb-192.168.80	TiDB: KV Commands: b	7d	365d	相关项目	27s	0 Ops		component: tiky-com [type: batch_get	图形	
Ō	服务 •	tidb-192.168.80	tidb.tikvclient_txn.rate[batch TiDB: KV Commands: c	7d	365d	相关项目	27s	0 Ops		component: tikv-com [type: commit]	图形	
	资产记录	tidb-192.168.80	tidb.tikvclient_txn.rate[commit] TiDB: KV Commands: g	7d	365d	相关项目	27s	0 Ops		component: tikv-com [type: get	國形	
<u>ا</u>	版表 ×	tidb-192.168.80	tidb.tikvclient_txn.rate[get] TiDB: KV Commands: Io	7d	365d	相关项目	27s	0 Ops		component tiky-com	國形	
	管理 ~	tidb-192 168 80	tidb.tikvclient_txn.rate[lock_k	7d	365d	相关项目	275	0 Ops		companent tiky.com	图形	
6	支持	tidb-192 168 80	tidb.tikvclient_txn.rate[rollback]	7d	365d	相关项目				component domain	間形	
E	Integrations	tidb-192 168 80	tidb.domain_load_schema.fa	74	3654	相关而日	27¢	0.04561	+0.001439	component: domain	100.000	
?	帮助		tidb.domain_load_schema.rate	74	0054	10,000	275	0.04301	10.001438	component: domain	DELT()	
n÷	User settings ~	tidb-192.168.80	tidb.tikvclient_lock_resolver	/d	3650	相天坝日	2/5	0 Ops		component: liky-com	RELTS -	
U U	退出	tidb-192.168.80	TiDB: Lock resolves: bat 2 tidb.tikvclient_lock_resolver	7d	365d	相关项目	27s	0 Ops		component: locks type: batch_resolve	图形	
7	ABBIX « 🔊		tidb.tikvclient_lock_resolver									•
	Q	tidb-192.168.80	TIDB: Lock resolves: ex	7d	365d	相关项目	27s	0 Ops		component locks type expired	國形	
0	监测 ^	tidb-192.168.80	TiDB: Lock resolves: not	7d	365d	相关项目	27s	0 Ops		component: locks type. not_expired	图形	
F	(父表板) (7)	tidb-192.168.80	TiDB: Lock resolves: qu	7d	365d	相关项目	27s	0 Ops		component: locks type: query_check_s	图形	
H	主机	tidb-192.168.80	TiDB: Lock resolves: gu	7d	365d	相关项目	27s	0 Ops		component: locks type: query_resolve_t	國用	
	最新数据	tidb-192.168.80	TiDB: Lock resolves: qu	7d	365d	相关项目	27s	0 Ops		component: locks type: query_resolve_I	图形	
	白动发现	tidb-192.168.80	TiDB: Lock resolves: qu	7d	365d	相关项目	27s	0 Ops		component: locks type: query_txn_status	图形	
Ō	服务 、	tidb-192.168.80	TIDB: Lock resolves: qu	7d	365d	相关项目	27s	0 Ops		component locks type: query_txn_statu	間形	
==	资产记录 ,	tidb-192.168.80	TIDB: Lock resolves: qu	7d	365d	相关项目	27s	0 Ops		component: locks type: query_txn_statu	图形	
. III.	报表 ~	tidb-192.168.80	TiDB: Lock resolves: res	7d	365d	相关项目	27s	0 Ops		component: locks type: resolve	图形	
	11日 、	tidb-192.168.80	TiDB: Lock resolves: res	7d	365d	相关项目	27s	0 Ops		component locks type: resolve_async	間形	
		tidb-192.168.80	tidb.tikvclient_lock_resolver TiDB: Lock resolves: res	7d	365d	相关项目	27s	0 Ops		component: locks type: resolve_for_write	图形	
9	支持	tidb-192.168.80	tidb.tikvclient_lock_resolver TiDB: Lock resolves: wai	7d	365d	相关项目	27s	0 Ops		component: locks type: wait_expired	图形	
?	帮助	tidb-192.168.80	tidb.tikvclient_lock_resolver TiDB: Lock resolves: writ	7d	365d	相关项目	27s	0 Ops		component locks type write conflict	图形	
n -	User settings ~	tidb-192,168.80	tidb.tikvclient_lock_resolver TiDB: Open file descriptors	7d	365d	相关项目	27s	17		component fds	國形	
ს	退出	tidb-192 168 80	tidb.process_open_fds	7d	365d	相关项目	275	1048576		companent fits	图形	
7		000-192.168.80	TIDB: Server guery OK protide.server_guery.ok.rate[St	/0	3000	相大坝日	2/5	u ups		component: quenes type: SimtReset	IEUTS	
	۹ ۹	tidb-192.168.80	TiDB: Server query "OK"	7d	365d	相关项目	27s	0 Qps		component: queries type: StmtSendLong	图形	
0	监测 ^	tidb-192.168.80	TiDB: SQL statements, r	7d	365d	相关项目	27s	0		component: sqt	間形	
П	仪表板	tidb-192.168.80	TIDB: SQL statements:	7d	365d	相关项目	27s	0		component sql type Select	图形	
ŀ	主机	tidb-192.168.80	TiDB: SQL statements:	7d	365d	相关项目	27s	0		component: sql type: Show	图形	
	最新数据	tidb-192.168.80	TiDB: SQL statements:	7d	365d	相关项目	27s	0		component: sql type: Use	圖形	
	白动发现	tidb-192.168.80	TIDB: Status	7d		相关项目	27s	Up (1)		component health	历史	
Ø	服务	tidb-192.168.80	TiDB: TiClient region err	7d	365d	相关项目	27s	0 Ops		component regions	图形	
==	资产记录 ,	tidb-192.168.80	tidb.tikvclient_region_err.rate TiDB: Time jump back, r	7d	365d	相关项目	27s	0 Ops		component: application	图形	
	报表 、	tidb-192.168.80	tidb.monitor_time_jump_bac TIDB: Total "error" server	7d	365d	相关项目	27s	0 Qps		component: queries	图形	
	16日 ·	tidb-192 168 80	tidb.server_query.error.rate	74	3654	相关项目	27s	0 Qps		component: queries	图形	
		tidb_100_100_00	tidb.server_query.ok.rate	74	3854	相至清日	276	0.0ps			(Sector)	
0	支持	Nido 400.00	tidb.server_query.rate	**	205-	19人※日	276	00:20:01	+00:04:00	component: quertes	parts.	
?	帮助	udb-192.168.80	tidb.uptime	/d	Jobd	相夫项目	2/5	JU.JU.U1	+00:01:28	component appreation	DELTES	
n •	User settings -	tidb-192.168.80	tidb.version	7d		相关项目	2/5	8.0.11-TiDB-v7.5.1		component: application	历史	
ს	退出	0.3412									显示 已自动发现的 96中的96	2
		● 124年 至774年重要改任										¥

以上就是本期的全部内容。大家好,我是乐乐专注 IT 运维技术研究与分享,更多运维技巧

欢迎关注乐维社区,更多运维问题也欢迎到乐维社区留言提问。

五十一、如何使用 Zabbix 监控神通数据库?

神通数据库,即神舟通用数据库(ShenTong Database),是我国自主研发的一款关系型数据库管理系统。它在国内市场有一定的应用,尤其是在一些对数据安全、独立性有较高要求的领域,如政府、金融、电信、能源、医疗等行业。

Zabbix 是一款被广泛应用的开源监控工具,常常被用于企业机构等 IT 基础架构的监控。

本文将介绍如何使用 Docker 快速安装和配置神舟通用数据库,并使用 Zabbix 监控 神舟通用数据库。

安装步骤参考:

https://blog.csdn.net/weixin_46603727/article/details/131300046 docker 镜像下载:https://pan.baidu.com/s/1-W_tuGk4waewNhr6C8Z00g 提取码:9572

1. 安装步骤

1.1. 安装 docker,下载神舟通用数据库镜像 shentong_342_163_x86_64bit.zip 并上传至服务器。

yum install docker

systemctl start docker

unzip shentong_342_163_x86_64bit.zip

docker load -i shentong_342_163_x86_64bit

导入完成后,可以使用 docker images 查看导入的镜像。结果显示如下:

<pre>[root@cenos7 shentong]# ls</pre>											
shentong_342_163_x86_64bit.zip											
[root@cenos7 shentong]# unzip shentong_342_163_x86_64bit.zip											
Archive: shentong 342 163 x	86 64bit.zip										
inflating: shentong_342_163_x86_64bit											
[root@cenos7 shentong]# docker load -i shentong 342_163_x86_64bit											
174f56854903: Loading layer	[======================================		>] [211.7	MB/211.7 MB						
6d1a884f63aa: Loading layer	[======================================		======>]	1.451	GB/1.451 GB						
41567f6995e1: Loading layer	[======================================		>] (6.656	kB/6.656 kB						
04299a68c8bb: Loading layer	[======================================		>] 4	4.608	kB/4.608 kB						
c937bcbb63a5: Loading layer	[======================================		>]	2.56	kB/2.56 kB						
ce384df97985: Loading layer	[======================================		>]	11.78	kB/11.78 kB						
17f04db55620: Loading layer	[======================================		=====>]	10.24	kB/10.24 kB						
b2309a771903: Loading layer	[======================================		>]	1.451	GB/1.451 GB						
Loaded image: shentong 342 10	63 x86 64bit:latest										
[root@cenos7 shentong]# dock	er images										
REPOSITORY	TAG	IMAGE ID	CREATED		SIZE						
docker.io/pingcap/tidb	latest	bd253ee807c6	6 weeks ago		393 MB						
shentong 342 163 x86 64bit	latest	38f7dd5ab67b	11 months ago		3.1 GB						
[root@cenos7 shentong]# ls											
shentong 342 163 x86 64bit	shentong 342 163 x86	64bit.zip									
[root@cenos7 shentong]#											

1.2. 初始化容器。

date -s "2023/06/20" #该镜像有试用授权期限,因此需要调整日期

docker run -d -p 2003:2003 --name Oscar shentong_342_163_x86_64bit

#创建并运行容器,将会在后台启动一个名为 oscar 的容器,并将容器的 2003 端口映

射到宿主机的 2003 端口,将数据目录映射到/opt/ShenTong/。

docker ps -a #查看容器运行状态

[root@cenos7 shentong]# docker run -d -p 2003:2003name oscar shentong 342 163 x86 64bit										
8bcc5e27d74f8	8bcc5e27d74f85789c7cf6816df9ce710c1cd0b0b1e2c013aab55910359a0b48									
[root@cenos7 :	shentong]# docker ps -a									
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES				
8bcc5e27d74f	shentong 342 163 x86 64bit	"bash /opt/s.sh"	5 seconds ago	Up 3 seconds	0.0.0.0:2003->2003/tcp, 5555/tc	oscar				
[root@cenos7	shentong]# 🗧									

正常启动后能看到容器里启动了/opt/ShenTong/bin/oscar 和

/opt/ShenTong/bin/oscaragent 进程。如果没有/opt/ShenTong/bin/osca 进程,可能

是授权到期了导致启动失败,需要调整本机时间。

Troordcent	057~]# (unckei	ps	- d								
CONTAINER	ID	IMA	GE				COMMAND	CREATED	STATUS	PORTS		NAMES
1c001e022	f37	she	nto	ong_34	2_163_	x86_64bit	"bash /opt/s.sh"	Less than a second ago	Up 2 minutes	0.0.0.0:2003->2003/tcp, 5	5555/tcp	Oscar
[root@cent	os7 ~]# (docker	e)	kec -i	t Osca	ar bash						
[root@lc00	01e022f3	7 /]#	ps	-ef								
UID	PID	PPID		STIME	TTY	TIME	CMD					
root		0	0	16:01		00:00:00	bash /opt/s.sh					
root				16:01		00:00:01	/opt/ShenTong/bin,	/oscar -o nor				
root	46		0	16:01		00:00:00	/opt/ShenTong/bin,	/oscaragent -				
root	48		0	16:01		00:00:00	tail -f /dev/null					
root	60	0	0	16:02		00:00:00	bash					
root		0		16:04		00:00:00	bash					
root	90		0	16:04		00:00:00	ps -ef					
[root@lc00	01e022f3	7 /]#										

默认用户名: SYSDBA, 密码: szoscar55, 库: OSRDB, schema: SYSDBA

容器内可使用/opt/ShenTong/bin/isql -U SYSDBA/szoscar55 -d osrdb 命令连接数

据库。

[root@cenos7 opt]# docker exec -it Oscar bash	
<pre>[root@1c001e022f37 /]# /opt/ShenTong/bin/isql -U SYSDBA/szoscar55 -d osrdb</pre>	
conversion between SQL_ASCII and UTF8 is not supported	
Welcome to isql 7.1.20220902 the ShenTongDB interactive terminal.	
Type: COPYRIGHT for distribution terms	
HELP for help with SQL commands	
? for help on internal commands	
! to run system commands	
EXIT to quit	
SQL> exit	
[root@lc001e022f37 /]# exit	
exit	
[root@cenos7 opt]#	

测试连接并执行命令:(需要指定连接客户端的字符集为 UTF-8)

export LANG=en_US.UTF-8

export LC ALL=en US.UTF-8

/opt/ShenTong/bin/isql -U SYSDBA/szoscar55 -d osrdb sysdba

select version;



2. 准备监控神舟通用数据库

2.1. 将神舟通用数据库提供的 python 模块及需要的 lib 文件复制到容器外的/opt 目录下。

docker cp Oscar:/opt/ShenTong/STPython/ /opt

docker cp Oscar:/opt/ShenTong/bin/libaci.so /lib64/

Idconfig -v | grep libaci #加载 lib



2.2. 安装 python3.5 用于监控神舟通用数据库。

安装步骤略

2.3. 安装连接神舟通用数据库用的 python 模块。

python3 -m pip install --no-index --find-links=/opt/STPython/Python35

STPython



2.4. 创建监控用的脚本。

在 zabbix 的外部检查目录下创建 2 个监控用的脚本, 我使用的是

/itops/zabbix/share/zabbix/externalscripts/目录,需要根据实际情况进行修改。

vi /itops/zabbix/share/zabbix/externalscripts/pyshentongdb

#!/usr/bin/bash
/usr/bin/python3 /itops/zabbix/share/zabbix/externalscripts/pyshentongdb.py
--username \$1 --password \$2 --address \$3 --port \$4 --database \$5 \$6 \$7 \$8

vi /itops/zabbix/share/zabbix/externalscripts/pyshentongdb.py

#!/usr/bin/env python3

coding: utf-8

author: cxh
import argparse
import STPython
import inspect
import json
import re
version = 0.2
class Checks(object):
def version(self):
"""查数据库版本"""
sql = "select version"
self.cur.execute(sql)
res = self.cur.fetchall()
for i in res:
print(i[0])
class Main(Checks):

```
def init (self):
    parser = argparse.ArgumentParser()
    parser.add argument('--username')
    parser.add argument('--password')
    parser.add argument('--address')
    parser.add_argument('--port')
    parser.add_argument('--database')
    subparsers = parser.add subparsers()
    for name in dir(self):
        if not name.startswith("_"):
            p = subparsers.add_parser(name)
            method = getattr(self, name)
            argnames = inspect.getargspec(method).args[1:]
            for argname in argnames:
                p.add argument(argname)
            p.set_defaults(func=method, argnames=argnames)
    self.args = parser.parse args()
def db_connect(self):
    a = self.args
    username = a.username
```

```
password = a.password
        address = a.address
        port = a.port
        database = a.database
        self.db
                                                                                   =
STPython.Connection(user=username,password=password,dsn=str(address)+':'+st
r(port)+'/'+str(database))
        self.cur = self.db.cursor()
    def db_close(self):
        self.db.close()
    def __call__(self):
        try:
             a = self.args
             callargs = [getattr(a, name) for name in a.argnames]
             self.db_connect()
            try:
                 return self.args.func(*callargs)
            finally:
                 self.db_close()
        except Exception as err:
```

print("0")	
print(str(err))	
ifname == "main":	
main = Main()	
main()	

因时间关系,本文章仅展示了神舟通用数据库的一个监控指标添加例子,后续需要神舟

通用数据库的 DBA 提供常用的监控相关的 SQL 语句进行监控指标的扩充。

ZABBIX « 🗈	最新数据							50
٩	< 7							~ >
 	主机群组	午此纳入搜索	选择	1512	当成(数认) 取			
仪表板	主机	申舟通用数据库 192 168.80.141 ¥	选择	1	际已	包含 ~	(a)	多除
问题	e e e e e e e e e e e e e e e e e e e	在此輸入搜索		18	E du			
主机	名称		3	显示标签	无 1 2 3	标签名 墳漏 縮短 :	无	
最新数据			标签显示	示优先级 1	以這号分隔的列表			
拓扑圈			3	直着細节 🔽	2			
			Save as	獻田	WIG			
	过滤器 只影响过滤后的数据			12010				
≔ 资产记录 -	TAG VALUES							
111 报表 🗸	Application: 常规 1							
🔧 配置 🔷 🗸	□ 主机 名称▲	间隔 历史	最近检查记录	最新数据	更改	标记		信息
管理 ~	■ 神舟通用数据库- 数据库版本 ■ pyshentongdb[「 7200 7d 外部地 SYSDBA,szos…	企査 1h 9m 49s	伸通数据周	奉7.0兆	Application: 常规		历史
					神递数据库7.0非正式	比授权版7.0.8.191204 for Lin	ux(x86 64bit) (65535 connection	ns) (license invalid after 133 days)
9 支持	0 选择 型示体叠数偏回 型示数据	10						
2 Integrations								
? 税助								
🚨 User settings 🔍								
di mur								
U IRH			Zabbix 6.0.6 © 2	2001-2022,	Zabbix SIA			
	监控项		Zabbix 6.0.6. © 2	2001-2022,	Zabbix SIA			
	监控项 所有主机 / 神舟通用数据库-192 168.80	141 已四用 ZBX 盛建项 87 触发器 23	Zabbix 6 0 6 @ 2 图形 14 自动发现规则 2	20012022, Web 场景	Zabbix SIA			
	监控项 所有主机 / 神舟週用数源時-192168.80 重短项 际记1 進程	141 已启用 28X 重改项 87 触发音 23	Zabbox 6 0 6 0 2 國用5 14 自动波和规则 2	2001-2022, Web 场景	Zabbix SIA			
C mathin ZABBIX 《 1 の 胎調 ~ の 略な ~	<u>監控项</u> 所有主机 / 特引適用数据準-192168.80 重控項 新記 1 道程 上度監修項 1	141 已四月 223、 [11:53] 67 1023篇 23	Zabbox 6 0 6 0 0 2 國形 14 自动发展URE 2	2001-2022, Web 场景	Zabbox SIA			
ZABBIX ≪ 1 Q 協調 ~ 協議 ~	<u>協控项</u> 所有主机 / 特引適用数型準-192 168.80 <u>単短荷</u> 新記 1 道程 上度监控液 1 ・名称	141 已四月 2013 至1508 97 1422番23 9行通用数据年期年標的通用] 数据年期本	Zabbox 6.0.6.0; 國形 14. 自动波到规则 2.	2001-2022, Web 场量	Zabbox SIA			
 ZABBIX ≪ 10 Q ● 监測 → ○ 服務 → □ 銀門記录 → 	<u>協控項</u> 所有主机 / 神丹通用教題年-192 168.80 <u>1015頃</u> 6月21 道程 上環道投稿 / ・名称: 実型	141 已四月 22X 国地派 87 触发着 23 中行通用规模单群中模拟通用] 数据年版本	Zabbox 6.0.6.0 2	2001-2022, Web 场量	Zabbox SIA			
C 並回 C 並同 C 単調 C 単調 C 膨弱 C 膨弱 C 膨弱 C 膨弱 C 膨弱 C し の C の C の C の C の C の C の C の C の C の C の C の の の の の の の の の の の の の	<u>協控項</u> 所有主机 / 時行週用数型体-192 168.80 <u>単記2版</u> 時日 1 連短 上型証拠 ・ 名称 ・ 電磁	141 已即用 201X 国际項 97 触波音 23 9円通用規模集製牛模切(通用) 数編集版本 外部始置 ~ pypehentogeti(SUSERNAME),(SPASSWORD),(HO	Zabbox 6.0.6.6.2 IBH: 14 ##3028306(R) 2 IST.CONNI, (SPORT), (SDATA)	2001-2022, Web 场展	Zabba SIA			
C 単示 C 単 C 単 C 単 C 単 C 単 C 単 C 単 C 単	<u>協控功</u> 所有主机 / 時行通用数製体-192 168.80 単記2項 56日 1 道程 上原出投現 ・ 今取 ・保政 信用型	141 已昭用 2000 国際26 87 触波音 23 中月通用取扱キ製牛模物(通用) 数編年版本 外部始重 ~ phentomogle(JUSERNAME) (#ASSWORD) (HO 学符 ~	Zabba 6 0 6 0 2	2001-2022, Web 场景 BA	Zabba SIA			
C 単語 C 単語 C 単語 C 服務 - で 服務 - で 服務 - で - で - 一 で - で - で - で - で - で - で - で - で - で - で - で - で - で - で - で - で - で - で - - - で - - - - - - - - - - - - -	<u>協控功</u> 所有主机 / 時有通用数量準-192 168.80 <u>進短週</u> 時日 : 道程 上型進時間 : 考報 ・登載 ・登載 ・登載 ・登載 ・登載 ・登載 ・登載 ・登載	141 已起用 2000 (11:556 87 社208 23 中行通用的(日本部本体物(通用) 数編件板体 外部始重 ~ Pyshentongsb((SUSERNAME) (\$PASSWORD) (HO 学祥 ~ 2012 (188.00.1411.10050 ~	Zabba 6 0 6 0 2	2001-2022, 3 Web 妨留 BA	Zabba SIA			
C 単語 C 単語 C 単語 C 単語 C 単語 C 単語 で 単語 た で 単語 で 一 で 一 で 一 の の の の の の の の の の の の の	広控 功 新音主 戦 / 時奇道 周敏震 年-192 168.80 進 短	141 已起用 2011 (11:55:0-87) M2X音 23 中行通用数据年期子標(10:6月) 数据年級本 外部地畫 ~ Pyshentangsdc(10:ERNAME) (IPASSWORD) (HO 字符 ~ 192:168.00.141.10050 ~ 7200	Zabbe 6.0.6.02	2001-2022, Web 场景 BA	Zabba SIA			
C 単位 ZABBIX 《 1 Q Q D D D D D D D D D D D D D	出 注 立 が 新音主 戦 / 特 奇 通 地 短 新音 主 戦 / 特 奇 語 の ま に ま 定 思 を か ま の ま の ま の ま の ま の ま の ま の ま の ま の ま	141 已初期 20X (法認知 87 14X7番 23 中行通用数据年期年代物(通用) 数期序版本 外部地量 ~ 字符 ~ 192 168 80 141 10050 ~ 7200 P/56 8 8	Zabba 6.0.6.02	2001-2022, Web 场量 BA	Zabba SIA			
C 単版 C 単数 C 単数 C 単数 C 単数 たま 	出 注 立 が 新音 主 パ ・ 特 行 語 ま の ・ 合 新 の ・ 全 、 の ・ 全 、 の 、 の ま の に の の の の の の の の の の の の の の の の	141 ご回川 送回川 ビー ジョ	Zabba 6 0 6 0 2 1876 14 material (SPORT) (SDATA) ST CONN) (SPORT) (SDATA) 1976 00 24 0	2001-2022, Web 场間 BA 动作: 逐發	Zabba SIA			
C 並回	出生が可し、 が育まれ、/ 特別通用数型等-192 168.80 当社交 6月2 1 単程 上原証拠点 1 ・名称 学校型 ・健理 信息での ・登録 の 主たり時间構 自生文分明明構	141 こ日田 2000 出版地 87 地加速 23 中月湯用前原車第十環境(通用) 数調率版本 労用地量 > 学校 > 学校 > 192.168.00.141.10050 > デジー - デジー - デジー - デジー - デジー - アン - デジー - ジジー -	Zabbox 6 0 6 0 7 2007 2000 2000 2000 2000 2000 2000 2000	2001-2022, Web 场置 BA 动作: 逐转	Zabba SIA			
C 単語 C 単語	出生空玩 所有主机 / 特利通用数類年192 188.80 重空型 転记 : 世程 上层监控派 ・ ・ 金和 ・ 登型 ・ 登載 ・ 世形順四 自生文分明明編 ・ ・ のた実数期得編时代	141 こ日田 正式 正式 正式 2.3 中井道用田田田市山田田 町田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田	Zabbox 6 0 6 0 2 EERF 14 643722 EUKER 2 ST CONNI, (SPORT), (SDATAL 800 800 117,00 00-24 00	2001-2022, Web 场景 BA	Zabba SIA			
C 単語 F 田 F 田	広控 功 新音 述 パ 特音 述 パ 特音 述 パ 特音 述 パ 特合 活 の な の の の の の の の の の の の の の の の の の	141 已旧用 (回文) (出社)(名 67) 社会)(名 23) 中月通用数(現体第2 + 14(数()通用)) 数据(形成系 年) 外型均衡 少 外型均衡 少 7200 完成 回家(第 50%) 2530 Do not keep history Strange period 90d	Zabbox 6 0 6 0 2	2001-2022, Web 场量 BA 动作: 杨雅	Zabbix SIA			
ZABBIX 《 1 Q 监測 L版 版法 化型 化 1 化 1	出生現式 / 特介通用数型本-192 168 80 当日3回 6月2 1 道理 所有主机 / 特介通用数型本-192 168 80 上型正常 7 上型正常 7 257 6月2 1 道理 上型正常 7 小型市場 小型市場 <t< th=""><th>141 已四時 (三文) (当地26.97) 地22(音 23) 中行通用我[現年第24歳年間(週月)] 教師後面 (一) 学時 (198,80) 141 10050 (199) 学校 (200,80) (200) 学校 (200,80) (200) 学校 (200,80) (200) (200) (200) (200) (200) (200) (200) (200) (200) (200) (200) (200) (200) (200) (200) (201) (202) (203) (204) (205) (205) (205) (205) (205) (205) (205) (205) (206) (207) (207)</th><th>Zabba 6 0 6 0 2 ERF 14 63722 EUKRY 2 ST CONNI, (SPORT), (SDATAI 899 1.7,00.00.24 00 </th><th>2001-2022, Web 场景 BA 회가: 原陳 記述</th><th>Zabbix SIA</th><th></th><th></th><th></th></t<>	141 已四時 (三文) (当地26.97) 地22(音 23) 中行通用我[現年第24歳年間(週月)] 教師後面 (一) 学時 (198,80) 141 10050 (199) 学校 (200,80) (200) 学校 (200,80) (200) 学校 (200,80) (200) (200) (200) (200) (200) (200) (200) (200) (200) (200) (200) (200) (200) (200) (200) (201) (202) (203) (204) (205) (205) (205) (205) (205) (205) (205) (205) (206) (207) (207)	Zabba 6 0 6 0 2 ERF 14 63722 EUKRY 2 ST CONNI, (SPORT), (SDATAI 899 1.7,00.00.24 00 	2001-2022, Web 场景 BA 회가: 原陳 記述	Zabbix SIA			
C 単語 C 単語	出生が近の 新有主机「神丹通用数選集-192.168.80 当日初 「中月通用数選集-192.168.80 上度出日気 「 ・名称」 ・名称」 ・名称」 ・名称」 ・名称」 ・名称」 ・名称」 ・名称」	141 已日用 (注文) (注文) 57 1427(音 23) 467) 通用資源年期14歳(1月) 数回年版本 外部始重 ア目かけの可念(SUSERNAME) (SPASSWORD) (HO 学校 学校 7200 思知 思知 思知 原稿 四 のれたeep halony Starage period 第2 天 Version on (SDATABASE)	Zabba 6 0 6 0 2 RR#: 14 (astro: SURR) 2 IST.CONNI, (SPORT), (SDATAI SNI 1-7,00 00-24 00 1-7,00 00-24 00	001-2022, Web 45월 BA BA 양한 2015	Zabbix SIA			
C and C and	出生初に、「神行通用影響体-1921日のある。 第行主託「神行通用影響体-1921日のある。 正常正常に、「空話記録」・ ・名称、「・名称、「・名称、「・名称、「・名称、「・名称、「・名称、「・名称、「	141 已回用 回訳 国社会社 97 社会社 823 401/通用数据集算 + は彼の(週用) 数据年版本	Zabba 6 0 6 0 2 RRF; 14 (hajge 30,000 2 ST CONN), (\$PORT), (\$DATAI 809 1-7,00 00-24 00	884 894 894 895 895 895 895 895 895 895 895 895 895	Zabba SIA			
C 単語 C 単語	出生初に、「サイオ通用数型体-192 168.80 当社25 5日日 単連 「日本111」「サイオ通用数型体-192 168.80 」 上型社25日 ・ ・名称 ・名称 ・名称 ・名称 ・名称 ・名称 ・名称 ・名称 ・ の電 ・ の電 ・ の電 ・ ので ・ ので の の の の の の の の の の の の の の の	141 已日用 2000 第1526 97 社21首 23 中月通用数据年期 中間時(週用) 数編年版本 外型時面での目気(USEENNAME) (SPASSWORD) (HO 学校 ~ 192.168.80.141.10050 ~ 7200 二〇〇 電波 505 二〇〇 電波 505 二〇 四 四 四 四 四 四 四 四 四 四 四 四 四 四 四 四 四 四 四	Zabba 6 0 6 0 2 REF: 14 (ha)2030/REF 2 ST CONN, (SPORT), (SDATA) 80/0 1-7,00 00-24 00 1-7,00 00-24 00	884 884 897년 897년	Zabba SIA			
C 2ABBIX 《 1 ○ 単語 () ○ 1 ○ 1 ○ 1 ○ 1 ○ 1 ○ 1 ○ 1 ○ 1	出生式 / サイオ通用数量体-192 168.80 当223 56日 1 連短 上层並投気 +	141 已日用 201 第123 87 社28 23 471通用度13年期 44歳年(2月)1 数編年版本 外部始進 ・ 中的時間の16(USERNAME) (SPASSWORD) (HO 学校 ・ 192 168.60 141 10050 ・ 7200 - 第二 (195 150 141 10050 ・ 7200 - 第二 (195 150 141 10050 ・ 720 - 720 - 72	Zabba 6 0 6 0 2	2001-2022, Web 场音 BA 화가 別王	Zabba SIA			

监控结果如下:

附:

数据库版本监控指标配置示例:

名称:数据库版本

类型:外部检查

键值:

pyshentongdb[{\$USERNAME},{\$PASSWORD},{HOST.CONN},{\$PORT},{\$DATABA

SE},version]

更新间隔:7200 (按需进行设置)

需要在模板上增加以下宏:

{\$USERNAME}:填写数据库的连接用户名

{\$PASSWORD}: 填写数据库的连接密码

{\$PORT}: 填写数据库的监听端口

{\$DATABASE}:填写数据库的实例名,本文是 osrdb

以上就是本期文章的全部内容。大家好,我是乐乐,专注运维技术研究与分享,想要了 解更多 zabbix 开源监控使用技巧,欢迎关注<u>乐维社区</u>,同时 zabbix 使用问题也欢迎到乐 维社区留言提问~

五十二、如何使用 Zabbix 监控 InfluxDB 数据库?

一、简介

InfluxDB 是一个时序数据库,旨在处理时间戳数据的高写入和查询负载。它是用 Go 编程语言编写的开源数据库,专门用于存储和查询时间序列数据,如指标、事件和日志。 InfluxDB 通常用于监控和可观测性、物联网应用和实时分析。它支持类似 SQL 的查询语言, 并与各种数据可视化和监控工具集成。

作为一款流行的开源时序数据库, InfluxDB 常年在 DB-Engines TSDB rank 中位居首位, 可见, InfluxDB 还是非常受欢迎的。

🗌 in	clude s	second	ary database models	42 systems in ran	42 systems in ranking, January 2024					
Rank		(S	core				
Jan 2024	Dec 2023	Jan 2023	DBMS	Database Model	Jan 2024	Dec 2023	Jan 2023			
1.	1.	1.	InfluxDB 🛨	Time Series, Multi-model 👔	27.56	-0.56	-1.80			
2.	2.	↑ 3.	Prometheus	Time Series	8.95	+0.61	+2.19			
3.	3.	4 2.	Kdb 🚹	Multi-model 👔	7.97	-0.31	+1.08			
4.	4.	4.	Graphite	Time Series	5.27	-0.19	-1.44			
5.	5.	5.	TimescaleDB	Time Series, Multi-model 👔	5.24	-0.05	+0.70			
6.	6.	6.	DolphinDB	Time Series, Multi-model 🛐	4.46	+0.55	+1.66			
7.	7.	个 9.	Apache Druid	Multi-model 👔	3.58	+0.24	+1.26			
8.	8.	↑ 10.	TDengine 🗄	Time Series, Multi-model 👔	3.43	+0.22	+1.33			
9.	9.	4 7.	RRDtool	Time Series	2.80	-0.01	+0.09			
10.	10.	个 12.	QuestDB 🔠	Time Series, Multi-model 👔	2.72	+0.42	+0.83			
11.	♠ 12.	4 8.	OpenTSDB	Time Series	2.21	+0.16	-0.36			
12.	4 11.	4 11.	GridDB 🗄	Time Series, Multi-model 👔	2.17	-0.06	+0.13			
13.	13.	13.	Fauna	Multi-model 🛐	1.96	+0.20	+0.19			
14.	14.	↑ 15.	VictoriaMetrics	Time Series	1.57	+0.11	+0.50			

本教程将介绍基于 CentOS 7.5, docker 进行 InfluxDB 数据库的安装, 并使用 Zabbix

对 InfluxDB 数据库进行监控。其中, Zabbix 版本为 6.0.6。



1、安装 docker 并拉取镜像。

yum install docker

systemctl start docker

docker search influxdb

搜索镜像,如果搜索不到需要设置 docker 仓库源

[TOOL@Centos	~]# docker search intrakub				
INDEX	NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
docker.io	docker.io/influxdb	InfluxDB is an open source time series dat	1839	[OK]	
docker.io	docker.io/telegraf	Telegraf is an agent for collecting metric	649	[OK]	
docker.io	docker.io/chronograf	Chronograf is a visualization tool for tim	360	[OK]	
docker.io	docker.io/tutum/influxdb	InfluxDB image - DEPRECATED. See https://d	223		[OK]
docker.io	docker.io/arm32v7/influxdb	InfluxDB is an open source time series dat	28		
docker.io	docker.io/bitnami/influxdb		13		
docker.io	docker.io/arm64v8/influxdb	InfluxDB is an open source time series dat	10		
docker.io	docker.io/prom/influxdb-exporter	A server that accepts InfluxDB metrics via	8		
docker.io	docker.io/rapidfort/influxdb	RapidFort optimized, hardened image for In	8		
docker.io	docker.io/influxdb/influxdb				
docker.io	docker.io/sillydong/influxdb-ui	web ui for influxdb query			
docker.io	docker.io/matisq/influxdb	TIG Stack - InfluxDB			[0K]
docker.io	docker.io/amd64/influxdb	InfluxDB is an open source time series dat	Θ		
docker.io	docker.io/bitnami/influxdb-relay-archived	A copy of the container images of the depr	Θ		
docker.io	docker.io/bitnamicharts/influxdb		Θ		
docker.io	docker.io/forestscribe/influxdb-docker-collect	collect docker stats from mesos and transm	Θ		[0K]
docker.io	docker.io/hephy/influxdb		Θ		
docker.io	docker.io/illusiveman/influxdb-to-s3	Backups influxdb to s3	Θ		
docker.io	docker.io/map3/influxdb		Θ		
docker.io	docker.io/monasca/influxdb-init		Θ		
docker.io	docker.io/objectscale/influxdb		Θ		
docker.io	docker.io/objectscale/influxdb-operator		Θ		
docker.io	<pre>docker.io/simonsobs/ocs-influxdb-publisher-agent</pre>	InfluxDB Publisher Agent	Θ		
docker.io	docker.io/ustclug/influxdb	influxdb used by USTC LUG	Θ		[0K]
docker.io	docker.io/vulhub/influxdb		0		
[root@cenos]	7 ~]#				

docker pull influxdb # 拉取镜像

docker images # 查看拉取的镜像

[root@cenos7 ~]# docker pull influxdb			
Using default tag: latest			
Trying to pull repository docker.io/libr	ary/influxdb		
latest: Pulling from docker.io/library/i	nfluxdb		
2f44b7a888fa: Pull complete			
60b879f7f0b9: Pull complete			
391f2a951491: Pull complete			
bc0f8d17c9bc: Pull complete			
1eea01495215: Pull complete			
c74718bd32e2: Pull complete			
ab287bfba4b0: Pull complete			
30f5135083ba: Pull complete			
8acfea7e99e3: Pull complete			
0c3fd5840e75: Pull complete			
Digest: sha256:f3765ab8ae6c9fddebe36b95c	e1a85097c92146573d1a	809dfe7bed60920a2d2	
Status: Downloaded newer image for docke	r.io/influxdb:latest		
[root@cenos7 ~]# docker images			
REPOSITORY TAG	IMAGE ID	CREATED	SIZE
docker.io/influxdb latest	7a1b63100937	2 weeks ago	378 MB
[root@cenos7 ~]# 📕			

2、初始化容器。

mkdir -p /data/influxdb #创建宿主机路径

docker run -dit --name influxdb --restart always -p 8086:8086 -v /data/influxdb:/var/lib/influxdb influxdb #创建并运行容器, 映射 8086 端口, 映射宿 主机的/data/influxdb 到容器里/var/lib/influxdb influxdb 目录。

docker ps -a #查看容器运行状态

<pre>[root@cenos7 ~]# doc f32f6b8edb480741dc8a</pre>	ker run -ditname 003890001339e1e0383	influxdbrestart always 067dad162b88c3c0588dc6d28	s -p 8086:8086 -v /d	ata/influxdb:/var/li	b/influxdb influxdb	
[root@cenos7 ~]# doc CONTAINER ID f32f6b8edb48 [root@cenos7 ~]#	ker ps -a IMAGE influxdb	COMMAND "/entrypoint.sh in"	CREATED 8 seconds ago	STATUS Up 5 seconds	PORTS 0.0.0.0:8086->8086/tcp	NAMES influxdb

3、为 influxdb 创建一个只读权限的 token,参考

https://docs.influxdata.com/influxdb/v2/admin/tokens/create-token/

本次采用的操作方法是:使用浏览器访问 influxdb 的 8086 端口。按以下步骤进行操

作:

÷	\rightarrow	C	▲ 不安全	192.168.253.154:8086/onboarding/0	\$ ☆	Ð	• :	8
				Welcome to InfluxDB				
				Get started in just a tew easy steps				
				GET STARTED				
								Π
				Powered by InfluxOsta				l

根据提示填写初始化信息:

← →	C	▲ 不安全	192.168.2	253.154:8086/onboarding/1				© • 🔤 ☆	Ð	•
					Walcome Ini	tal Her	consists			
					welcome Int	dal Use	Complete			
					Setup I	Initi	al User			
				Username						
				admin						
				Password		¢	Confirm Password			
				••••••						
				Initial Organization Name						
				myorg						
				Initial Bucket Name 🕕						
				mybucket						
								co	NTINUE	

初始化后会提供管理员权限的 token ,可直接使用或再创建一个只读 token:

	Welcome ——— Initial User Setup ——— Comp	lete				
	You are ready to go!					
	Make sure to copy your operator API token now. This token enables superuser privileges like creating users, orgs, etc. You won't b	be able to see it again!				
	frm-#UWZ501Wp-Pp4Gly1zUPk1J91Yq22nEwoEpf5BeJHh43joF1ZznbLsJBwlqymqkojWTy9rfGmUfmubuFLQ==					
COPY TO CLIPBOARD						
	Let's start collecting data!					
QUICK ST	ART	CONFIGURE LATER				
Timing is ever	ything! Whoa looks like you're an expert!	I've got this				
	Powered by InfluxData					

继续创建只读 token:

÷	→ (C	▲ 不安全	192.168.253.154:8086/onboarding/2		©⊽ \$§ ☆	Ð	•		
					Welcome Initial User Setun Complete					
					recome sinusi del delap compre					
	You are ready to go!									
			4	This 1	Make sure to copy your operator API token now. oken enables superuser privileges like creating users, orgs, etc. You won't be able to see it again!					
				frm-W	WZ501Wp-Pp4Gly1zUPk1J91Yq22nEwoEpf5BeJHh43joF1ZznbLsjBwlqymqkojWTy9rfGmUfmubuFLQ==					
				COPY TO CLIPBOARD						
	Let's start collecting data!									
				QUICK START	ADVANCED CONFIGURE LAT	ER				
				Timing is everything!	Whoa looks like you're an expert! I've got this					

٩	Load Data		
m			
	SOURCES BUCKETS TELEGRAF SCRAPERS	API TOKENS	
<u> </u>	Sources Buckets		
-	Telegraf	Sort by Description (A + Z) +	3 + GENERATE API TOKEN -
-U 	API Token		8 D
iii tii	Created at: 2024-01-26 17:22:31 Owner: admin Last Modified: 4		
0			
Ч ~			
ę			
0			
D			
۲	Load Data		
a la		API TOKENS	
Image: Second seco	Load Data sources buckets telegraf scrapers	API TOKENS	
R 🖯 📑 😵	Load Data SOURCES BUCKETS TELEGRAF SCRAPERS	API TOKENS Sort by Description (A + Z) *	+ GENERATE API TOKEN A
😁 R 🕨 📑	Load Data sources buckets telegraf scrapers	API TOKENS	GENERATE API TOKEN All Access API Token
	Load Data SOURCES BUCKETS TELEGRAF SCRAPERS G Filter Tokens admin's Token	API TOKENS	GENERATE API TOKEN All Access API Token Custom API Token 2
 응 E 4 2 3 6 4 4	Load Data SOURCES BUCKETS TELEGRAF SCRAPERS Q Filter Tokens ddmin's Token Created at: 2024-01-26 17:22:31 Owner: admin Last Modifiet: 4	API TOXENS Sort by Description (A + Z) * minutes ago	All Access API Token Custom API Token 2
 응 E H L H E H H	Load Data SOURCES BUCKETS TELEGRAF SCRAPERS Q Filter Tokens admin'S Token Created at: 2024-01-26 17:22:31 Owner: admin Last Modified: 4	API TOKENS Sort by Description (A = Z) •	+ GENERATE API TOKEN All Access API Token Custom API Token 2
 중 전 때 때 그 또 ト 	Load Data SOURCES BUCKETS TELEGRAF SCRAPERS	API TOKENS Sort by Description (A - Z) *	GENERATE API TOKEN All Access API Token 2
② む 🕮 🗂 R 🕨 🚽 📀	Load Data SOURCES BUCKETS TELEGRAF SCRAPERS Q Filter Tokens admin's Token Created at: 2024-01-26 17:22:31 Owner: admin Last Modified: 4	API TOKENS Sort by Description (A + Z) * minutes ago	1 CENERATE API TOKEN A Al Access API Token Custom API Token 2
 한 전 1 한 전 2 <li< th=""><th>Load Data SOURCES BUCKETS TELEGRAF SCRAPERS</th><th>API TOKENS Sort by Description (A + 2) * Initialities ago</th><th>GENERATE API TOKEN Ali Access APi Token Custom APi Token 2</th></li<>	Load Data SOURCES BUCKETS TELEGRAF SCRAPERS	API TOKENS Sort by Description (A + 2) * Initialities ago	GENERATE API TOKEN Ali Access APi Token Custom APi Token 2
) () () () () () () () () () () () () ()	Load Data SOURCES BUCKETS TELEGRAF SCRAPERS	API TOKENS Sort by Description (A - Z)	1 CENERATE API TOKEN A Ali Access API Token Custom API Token
• · · · · · · · · · · · · · · · · · · ·	Load Data SOURCES BUCKETS TELEGRAF SCRAPERS	API TOKENS Sort by Description (A + Z) * minutes ago	1 CENERATE API TOKEN A Al Access API Token Custom API Token 2

填写名字并选择只读权限:

Load Data	Description			
	Zabbix			
	Q Filter Access Permissions			
	Resources	Read	Write	
	- Buckets			
	All Buckets			
	Individual Bucket Names			
	mybucket			
	_monitoring			
	_tasks			
	- Telegrafs			
	All Telegrafs			
	Other Resources			
	All Other Resources			
	All Annotations			
	All Authorizations			

点击确定:

		Individual Bucket Names		1	
	Load Data	mybucket			
m		_monitoring			
1		_tasks			
		- Telegrafs			
		All Telegrafs			
1		 Other Resources 			
88					
菌		All Other Resources	-		
~		All Annotations			
С.		All Authorizations	•		
٢		All Checks			
		All Dashboards			
		All Dbrp			
		All Documents	2		
0			CANCEL		
Ð					

复制生成的 token:

Load Data	
You've successfully created an API Token X	
admin's Token	
Constant in Constant Constan	
1A7Xy1sfdmrifU5kVFjGAznqhs9qBDHRZkH2r9j00ps3Tib#BcRkfyn010tsxMxbipb7pAwB1CdnTWZ Zabbbx TAznAxQ==	
Created at: 2014-01-06 17-27-47	

三、监控 InfluxDB

1、导入监控模板

InfluxDB by HTTP[中文].yaml

也可使用 Zabbix 官方提供的监控模板:

https://www.zabbix.com/cn/integrations/influxdb

ZABBIX « 🔊	模板		创建模板 导入
			3. 过滤器 文
◎ 监测 -	主机群	* 导入文件 通復文件 InfluxDB by HTTP(中文) yaml 4	
(2)服务 →	傳播的機	规则 更新现有的 创建新的 删除失败 《 值 移绘	
	e	RF4E V V	
111 报表 🗸		Template dashboards	
▲ 配置 ♪	1 🗌 名称 🛦	模板间的关联	Linked to templates #532
主机制度	□ 华为ECS模板	曲投頭 ダ マ	
横板 2	□ 华为RH系列SNMP通用模板[2011]	自动数规规则 🔽 🔽 🔤	
主机	□ 华为VPC模板		
维护	华为专线NQA链路探测	Web 场景 V 5	
न्नभः •	□ 华为云ECS自动扫描模板	日本 御湯	
关联项事件	华为云RDS模版		
自动发现	◎ 拳为云RDS自动归摇模板.	主机 监控项 10 触发器 图形 仪表量 自动规规 1 Web监测	
2 管理 -	华为云VPC自动扫描模板	主机。监控项 触发器 图形 仪表盘 自动发现: Web监测	
·9 支持	□ 半方交换机模板(2011)		
	学为父亲机模板[2011][S系列最狭翼元初		
29 integrations			
? 帮助		TAR TELEVES RECENSE REPORT HINTERS A MARKED A	The Display The Annaly
🚨 User settings 🔍	the free to be free and free		SEAC Characterization (24-43) (2
の 現出	0.25H HO ~ RECUI		
ZABBIX « 🔊	模板		創建模板 导入
模板			×
= Added	templates		
- 1015	+ - + templates	'Tofluyng by HTTP'	
InfluxDB by HTTP	+ name: 'Ir	fluxDB by HTTP' 2. That TelluyTB watrice by UTTD event from Dromethous watrice androist \v\sExt evenintian discovery textlate and to use	Authorization wis ADT to
* (CASTR	+ taga:	n. Get initiatus meticas sy nite syste itom filosensus meticas subjetici (reitor organization discovery campiate need to use	MENOLIZACIÓN VIA API CO
教得采集	+ tagi		
数据库健康检查状态	+ -	, database	
Boltdb读取連率	+ value	arget : influxdb	
Boltdb写入連率	+ macros: + -		
Buckets总数	+ maoro + descr	: '(#INFLUXDE.AFI.TOKEN)' iption: 'InfluxDE AFI Authorization Token'	
Dashboards总数	+ - + macro	: '(SINFLUXDB.ORG_NAME.WATCHES)'	
Organizations总数	+ value + descr	; ',*' iption; 'Filter of discoverable organizations'	
Scrapers总数	+ - + macro	: '(\$INFLUXDB.ORG_NAME.NOT_MAICHEB)'	
任务运行失败百分比	6 + value + descr	: CHANGE_IF_NEEDED iption: 'Filter to exclude discovered organizations'	
任务运行成功百分比	+ - + maoro	: '{\$influxdb.reo.fail.nax.warm}'	
正在运行任务的Wo	rkers忠数 + value + descr	: '2' iption: 'Maximum number of query requests failures for trigger expression.'	
Workers繁忙度	+ -	1.101001000 0100 0101 0101 010011	
Telegrafs总数			
			导入取消
🚨 User settings 🐭			显示已自动发现的 12中的12

2、添加 InfluxDB 监控

点击配置->主机->创建主机,填写主机名称,选择刚刚导入的 InfluxDB 监控模板,设

置一个群组。

Z	ABBIX « 🔊	主机	创建主机 导入		
		New host	2 过滤器 7		
0		主机 IPM 标记 宏 资产记录 加密 值時的			
õ		* 主約名称 InfluxDB-192.188 253.154 3			
:=	资产记录 -	可见的云称: Influx0B-192 168 253 154			
		模板 InfluctOB by HTTP X Trifted A state 4 初経	移除		
٩	配置 ^	* 計超 InduxDB (新) × 初指			
		在此地入提来 5			
-	模板	Interfaces No interfaces are defined.			
L.	±81	_ 名称	秋态 可用性 agent 加密 信息 标记		
		□ 182. 描述	已启 SNMP 无用		
			28 ZBX X		
		A.	H.		
		位在 曲agenf代理程序监测 (元agenf代理程序) ~	用		
9			已自 ZBX 无 用		
2		Sata Sata Sata	已启 [ZBX] 元 用		
?			显示已自动发现的 5中的5		
*					

点击宏,点击"继承以及主机宏",填写相关信息:

ZABBIX « S	主机	创建主机 导入
٩	New host	过滤器 🏹
 ● 监测 	主机 IPMI 标记 索 2 资产记录 加密 偏純的	
⑦ 服务 →	主机效 现员以及主机 宏	
≔ 资产记录 -	家 可用価 特徴価 全局価(配置)	
🖬 报表 🗸 🗸	(SINFLUXOB.API.TOKEN) 147X/sidminFUSK/FjGArandsdegDHR2AH T- 23(90C)os31WoldKolfx0far010sad.bbbp07pAw BiCdnTWZTArnArQ==	<u>1899</u>
く 配置 -	InfluxDB API Authorization Token	
主机即组	(\$INFLUXDB ORG_NAME MATCHES) ・ T 工 取文 winfluxDB by HTTP. ***	
模板	Filter of discoverable organizations	
主机	S/6 (SINFLUXDB ORG_NAME NOT_MATCHE CHANGE_IF_NEEDED T * IEX & InfluxDB by HTTP. "CHANGE_IF_NEEDED"	态 可用性 agent 加密 信息 标记
	192 Filter to exclude discovered organizations	部 SNMP 无
关联项事件	(\$INFLUXDB REQ FAIL MAX WARN) 2 T Y T St = InfluxDB by HTTP. "2"	E 78X Z
自动发现	Maximum number of query requests failures for trigger expression.	
管理	(SINFLUXDB TASK RUN FAIL MAX WARN) 2 T 学 既致 = InfluxDB by HTTP: "2" 用	古 [233] [九]
0 ====	Maximum number of tasks runs failures for trigger expression.	图 ZBX 无
-9 x.0	(\$INFLUXDB.URL) Mttp://192.168.253.154.8086 T → 接触 ← InfluxDB by HTTP: "http://ocalhost.8086"	启 [ZBX] 家
Integrations		
? 帮助		显示已自动发现的 5中的5
💄 User settings 🖂	Dar in the second	
(¹) (13.14)		

宏说明:

{\$INFLUXDB.API.TOKEN}: 填写 InfluxDB 的 token。

{\$INFLUXDB.ORG_NAME.MATCHES}: 表示需要监控的 ORG 名称, 默认.*表示监控全部 ORG。

{\$INFLUXDB.ORG_NAME.NOT_MATCHES}: 表示不需要监控的 ORG 名称, 默认 CHANGE_IF_NEEDED。

{\$INFLUXDB.REQ.FAIL.MAX.WARN}:设置请求失败数量告警阈值,默认是2。

{\$INFLUXDB.TASK.RUN.FAIL.MAX.WARN}:设置任务失败数量告警阈值,默认是 2。

{\$INFLUXDB.URL}: 设置 InfluxDB 的 URL, 默认: http://IP 地址:8086

ZABBIX « 🔊	主机		创建主机 导入		
٩	New host	New host			
● 監測 ・	主机 IPMI 标记 宏 2 资产记录 1	主机 IPMI 标记 宏 2 资产记录加密 值种的			
ا الله الله الله الله الله الله الله ال	主机宏 继承以及主机 宏				
≔ 资产记录 -	宏	可用值	模板值	全局值 (配置)	
🖬 报表 🗸 🗸	(\$INFLUXDB.API.TOKEN)	1A7XyIsfdmniFU5kVFjGAznqhs9qBDHRZkH 2r9jOOps3TibWBcRkfyn01OtsxMxbipb7pAw BICdnTWZTAznAxQ==	H T Y 終陸 ⇔ InfluxDB by HTTP: ***		修辞
🔧 配置 🔷	InfluxDB API Authonization Token				
主机用料图	(\$INFLUXDB.ORG_NAME.MATCHES)	2	T ~ 更改 ← InfluxDB by HTTP: ***		
樽板	Filter of discoverable organizations				
主机	(\$INFLUXDB.ORG_NAME.NOT_MATCHE S)	CHANGE_IF_NEEDED	T × 更改 ← InfluxDB by HTTP: "CHANGE_IF_NEEDED"		秋态 可用性 agent 加密 信息 标记
2017 ×	192 Filter to exclude discovered organizations				已启 SNMP 无
关联项事件	\$INFLUXDB.REQ.FAIL.MAX.WARN}	2	T \sim <u>E</u> α ← InfluxDB by HTTP: "2"		28 283 2
自动发现	Maximum number of query requests failures	i for trigger expression.			
管理 ·	(\$INFLUXDB TASK RUN FAIL MAX WARN	2	T ← 更改 ← InfluxDB by HTTP: "2"		已启 [283] 无 用
0	Maximum number of tasks runs failures for	trigger expression.			已自 ZBX 无
9 支持	(SINFLUXDB URL)	http://192.168.253.154.8086	T → 務除 ← InfluxDB by HTTP: "http://localhost.8086"		
Integrations	<u></u>				<u>已宕 [ZBX]</u> 用
? 帮助				150m	显示已自动发现的 5中的5
LUser settings 🗸					

信息填写完成后,点击添加按钮即可完成监控主机的添加。

添加完成后,可在主机管理界面,对自动发现规则触发立即执行,使其快速创建自动发现的

监控项。

Z	ABBIX « 🗈	自动发现规则		3							创建	使现境则
• •	 監調 ・ 服务 ・ 資产记录 ・ 报表 ・ 	所有主机 Inhux0B-162 168 253 154 2 3	ビジョボ 2015月 2015月1日 1023日本 1025月1日 1023日本 ビジョン 2015日 20150 1001111111111111111111111111111	[現則] Web 场置 选择 选择	类型 更新间隔 资源周期不足	所有	>	状态 所有 正常 状态 所有 已启用	不支持的 停用的		10	8勝
4	配置 ^ 主机排组			S.	UH EQ							
C	模版 主机 1	 ✓ 主机 ✓ InfluxDB-192 168 253 154 4 	名称 ▲ InfluxD8 by HTTP: Organizations discovery	监控项 监控项原型 6	触发器 触发器类型 1	图形 图形原型 2	主机	總值 influxdb.orgs.discovery	间确 10m	类型 HTTP 代理	状态 已応用	信息
	理デ 动作 > 关联项争件 自动发現	1 选择 居用 盖用 Execut	te now 開始 5							1 7521		3 14983 1
•	管理 🗸											
2	Integrations											
ب	User settings v			Zabbix 6.0.6	© 20012022, <u>Zat</u>	obix SIA						

查看监控数据,监控完成。

ZABBIX « 🗈	最新数据	20	ľ
٩	< 7	~ >	
• 監測 •	±1711年日 だがら入IPで 38.08 15(2 15(2 15(2 15)))		
仪表版	고 V M 1 2 U U U U U U U U U U U U U U U U U U		
问题	2 译如		
主机 1	名称 显示标签 无 1 2 3 标签名 規調 编版 无		
最新数据	杨紫雪元优先级 以返号分编的列表		
拓扑圈	資料行行		
自动发现			
④服务 →	Save as 应用 重化		
😑 资产记录 🗸 🗸	TAG VALUES		l
11 报表 🗸	Application: 健康状态 1 常规 16 数据采集 1 重询 6		
<u>م التق</u>	DATA		
四 \$\$\$\$\$\$			
	王邦 谷麻 和 助び检查记录 単新振躍 更成 初记 Infly (102.150.252.1 Bolth E) 建本 間 1m 14e 0 Infly (102.150.252.1 Bolth E) 建本 間 1m 14e 0	信息	
9 支持	Immunute-rez. 100.2003.1. Delitable rez. Delitable rez.	INTS INTS	
Z Integrations	InfluxDB-192_168_253.1 Buckets@#K	間形	
2 zilah	InfluxDB-192 (168 253 1. Dashboards 580) 3m 14s 0 Acceleration 5800	图形	
-	InfluxDB-192.168.253.1 Organizations语致 2 3m 14s 1 Application 完弱	國用	
User settings	□ InfluxDB-192168.253.1 Scrapers会政 2 3m 14s 0 Application: 市田	图形	
し 退出	InfluxDB-192.168.253.1 Telegraf plugins 💱 👔	國形	
	InfluxDB-192.168.253.1 Dashboards.588 2 27m 12s 0 Application 1938	國形	
۹.	InfluxDB-192-168-253.1Organizations@ep. 04 27m 12s 1 Application #300	1921HS	
● 监测 ^	InfluxUo-192.100.233.1. SUdeptisess pin 2/111.25 0 Application Hange InfluxUo-192.100.233.1 Talenza hunins 510 2 Application Hange	BEUTS BEITE:	
仪表板	InfluxDB-192 (58 253 1 Telegrafs/SBX Z Z7m 12s 0 Available	國形	
问题	InfluxDB-192 169 253 1 Tokens⊜#X 🚰 27m 12s 2 Appendix 1993	國用彩	
主机	InfluxDB-192.168.253.1 Users急权 2 27m 12s 1 Application: 常规	國用	
最新数据	InfluxDB-1921882531 Worker\$繁代度 @ 1m 12s 0 % Application: 1時間	图形	
拓扑图	□ InfluxDB-192.168.253.1 [myorg] 每秒面词响应失败的字节大小 🗾 Application 面前	图形	
自动发现	□ InfluxDB-192.168.253.1 [myorg] 每份重编编位成DBS字书大小 21	图形	5
O ₩3 ·	□ InfluxOB-192.168.253.1 (myorg) 每秒重调失现的调求数 2	國形	
≔ 资产记录 ~	□ InfluxDB-192.168.253.1 (myorg) 每秒查询成功的请求数 2	國形	
🛄 报表 🗸 🗸	InfluxDB-192.168.253.1 (myorg) 每秒重击调求失败的字节大小 2 Application: 查询	图形	
× 800 -	InfluxDB-192.168.253.1 [myorg] 每份重金编束成功的字节大小 Application: 首位	图形	
1 管理 。	□ Inftw08-192.198.203.1. 任美語学現文母会化 ■ Application 常規	BEHS PRITY	
	Immuzub-192.100.253.1 1179/26/1704/0192712 Application 発現 Application 発現 Application 発現 Application 発現	DELTES DELTES	
0 支持	ImmonOp/102.1002400.1 RODE+REPROVERSING	nan»	
Z Integrations	InflamDB-1921時2531 教授業書	历史	
? 帮助	□ InfluxDB-192.168.253.1正在运行任务的Workers受教	間形	
	□ InfluxDB-162.168 253.1 运行时长 2 1h 9m 36s +2m 217.06 Application: %初期	國用利	
User settings ~		己自动发现的 24中的24	1
じ退出	1432 D-T+050020 D-T+050		


云监控

七十七、通过 zabbix 监控华为云 RDS

通过 zabbix 监控华为云 RDS 思路通过华为云 RDS 实例列表接口获取 RDS 实例,用于资源的自动发现通过华为云 CES 获取监控数据准准备工作用于访问华为云的 AK/SKpython3 环境用于调用华为云 api 的相关模块依...



通过华为云 RDS 实例列表接口获取 RDS 实例,用于资源的自动发现

通过华为云 CES 获取监控数据准

二、准备工作

и 华为云 🔢 🎧 控制台 Ξ 我的凭证 基本信息 访问密钥 ⑦ 安全设置 API%E ① 如果访问密钥泄露,会带来数据 访问密钥 Q ⊙ 新增访 问密钥 您还可以添加1个访问密钥。 0 M 描述 ↓= 状态 見日 BID J≣ 创建时间 ↓Ξ 0 10□ 通用API 0 倉田 2022/10/24 15: 切换角色 标签管理 0 4 0 (2)
 (2)
 (3)
 (4)
 (5)
 (5)
 (6)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 (7)
 æ 0

用于访问华为云的 AK/SK

python3 环境

用于调用华为云 api 的相关模块依赖

https://github.com/huaweicloud/huaweicloud-sdk-python-v3

三、编写实例自动发现脚本

#!/bin/python3

coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials

from huaweicloudsdkrds.v3.region.rds_region import RdsRegion

from huaweicloudsdkcore.exceptions import exceptions

from huaweicloudsdkrds.v3 import *

import json

if __name__ == "__main__":

#这里填访问密钥

ak = ""

```
sk = ""
```

credentials = BasicCredentials(ak, sk) \

client = RdsClient.new_builder() \

```
.with_credentials(credentials) \
```

.with_region(RdsRegion.value_of("cn-north-4")) \

.build()

try:

```
request = ListInstancesRequest()
```

response = client.list_instances(request)

response = json.loads(str(response))

#拼接成用于自动发现的 json 数据, id 为 RDS 的实例 ID, name 为 RDS 的名称

r = []

for i in response['instances']:

buf = {'{#INSTANCE}':i['id'],'{#INSTANCE_NANME}':i['name']}

r.append(buf)

```
print(json.dumps({"data":r}))
```

except exceptions.ClientRequestException as e:

print(e.status_code)

print(e.request_id)

print(e.error_code)

print(e.error_msg)

四、编写监控数据获取脚本

```
#!/bin/python3
```

```
# coding: utf-8
```

from huaweicloudsdkcore.auth.credentials import BasicCredentials

from huaweicloudsdkces.v1.region.ces_region import CesRegion

from huaweicloudsdkcore.exceptions import exceptions

from huaweicloudsdkces.v1 import *

import json

import time

import sys

#传入的监控项

key = sys.argv[1]

#请求时间戳

from_time = int(round(time.time() * 1000)-300000)

to_time = int(round(time.time() * 1000))

if __name__ == "__main__":

#这里填访问密钥

ak = ""

```
sk = ""
```

credentials = BasicCredentials(ak, sk) \

client = CesClient.new_builder() \

.with_credentials(credentials) \

.with_region(CesRegion.value_of("cn-north-4")) \

.build()

```
try:
   request = BatchListMetricDataRequest()
   listDimensionsMetrics = [
      MetricsDimension(
         name="rds_cluster_id",
         #这里是实例 ID,可改为变量,模板宏即{HOST.HOST}
         value=""
      )
   ]
   listMetricsbody = [
      MetricInfo(
         namespace="SYS.RDS",
         metric_name=key,
         dimensions=listDimensionsMetrics
      )
   ]
   request.body = BatchListMetricDataRequestBody(
      to=to_time,
      _from=from_time,
      filter="max",
      period="1",
      metrics=listMetricsbody
```

)

response = client.batch_list_metric_data(request)

res = json.loads(str(response))

instance = json.dumps(res)

instances = json.loads(instance)['metrics']

for i in instances:

#取最新的一个值

print(i['datapoints'][0]['max'])

#print(i)

except exceptions.ClientRequestException as e:

print(0)

#print(e.request_id)

#print(e.error_code)

#print(e.error_msg)

五、新增模板

新增自动发现规则,用于实例自动发现

-	- 1	(I) TO LO DI	
	-11		
	111		
-	~		

所有模板 / 华为云RDS自动扫描模板	自动发现清单 / 实例名称自动发现 监控项原	型 触发器类型 图形原型	主机模板 1
自动发现规则 进程 LLD macros	; 过滤器 覆盖		
* 名称	实例名称自动发现		
类型	外部检查 🗸		
* 键值	hwrdsinfo.py		
* 更新间隔	1d		
自定义时间间隔	类型 间隔	期间	动作
	灵活 调度 50s	1-7,00:00-24:00	移除
	添加		
*资源周期不足	30d		
描述			
			1.
已启用	\checkmark		
	更新 克隆 测试 删除 取消		

自动发现规则,链接监控数据获取模板

++1	1+#+5
土①	山发仪

标记宏资产记录发	10密	
* 主机名称	(#INSTANCE)]
可见的名称	HWRDS_{#INSTANCE_NANME}]
模板	名称 动作	
	华为云RDS模版 取消链接	ž
	在此输入搜索	选择
* 群组	华为云 ★	选择
	在此输入搜索	
组模板	{#MACRO}	移除
	添加	
Interfaces	Inherit 习惯	
启用新的		
Discover		
	更新 克隆 删除 取消	

监控项														創建监控项
所有模板 华为云RDS模版 监持	空项 81 触	发器 2 图形 仪表盘	自动发现规则	则 Web 场景										过滤器 🏹
	主机群组	在此输入搜索	选择	类型	所有	~	标记	与成(默认) 或	Ē					
	模板	华为云RDS模版 × 在此轴入搜索	选择	信息类型	所有 >			标记 添加	ŧ	5a	۷ (ά.	移除	
	名称			历史记录			状态	所有 已启用	停用的					
	罉值			更新间隔			触发器	所有 是 不						
	值映射	在此输入搜索	选择			Ir	nherited	所有 是 不						
过滤器 只影响过滤后的数据						应用 1	file -							
标记 Buffe: Buffe 4 InnoDB: InnoDB 12	使用室: 使用	日 率 4 常規:常規 21 磁想	1: 磁盘 8 缓)	助机 緩冲流 3 【	网络: 网络 2 语句: 语句	7 溶产值局: 治	产信息 1	0						
类型 Zabbix采集器 10 <u>外部检查</u> 71														
信息类型 数字(无正负)3 文本7 浮点数71														
有触发器 无触发器 80 有触发器 1														
			触发器	键值					间隔	历史记录	趋势	类型	状态	标记
默认用户名				db_user_n	ame					90d		Zabbix采集器	已启用	资产信息 资产信息
长事务指标				hwrds.py[r	ds_long_transaction]				5m	90d	365d	外部检查	已启用	
连接数使用率				hwrds.py[r	ds072_conn_usage]				5m	90d	365d	外部检查	已启用	使用率 使用率
进程的虚拟内存占用	離値			hwrds.py[r	ds080_vm_peak]				5m	90d	365d	外部检查	已启用	

六、验证

验证是否可以正常发现实例

新增主机, 链接华为云 RDS 自动扫描模板后, 点自动发现规则, 可以正常获取到数据

lue
N
701014
K(P)

验证监控数据是否获取正常

点击发现出来实例,测试监控项可以正常获取到数据

信息类型	Test item				×	
* 主机接口						
单位	Get value from host	▼				
* 更新间隔	Host address			端 口		
do alto di concessioni di	agent代理程序	(无agent代理程序) ~				
自定义的时间隔					Get value	
	值	1.0	2	时间	now	
		不支持的				
*历史数据保留时长	Previous value		1	Prev. time		
* 趋势存储时间	Fadation	15 0015				
填入主机资产纪录栏位	End of line sequence	LF CRLF				
描述	结果	Result converted to 浮点数			1	
				G	iet value and test 取消	
		4				
已启用	~					
	最新数据					

七十八、腾讯云 NAT 网关监控

1.NAT 网关介绍

NAT 网关 (NAT Gateway) 是一种支持 IP 地址转换服务,提供网络地址转换能力,主要包括 SNAT (Source Network Address Translation,源网络地址转换) 和 DNAT (Destination Network Address Translation,目的网络地址转换)能力,可为 私有网络 (VPC)内的资源提供安全、高性能的 Internet 访问服务。

NAT 网关分为公网 NAT 网关和私网 NAT 网关。公网 NAT 网关提供公网地址转换服务, 而私网 NAT 网关提供私网地址转换服务。

NAT 网关支持高达 99.99%的高可用性、5Gbps 的带宽以及 1000 万以上的并发 连接数,其典型应用场景如下:

1. 大带宽、高可用的公网出口服务,例如:网络爬虫,访问 Internet 公共服务等。

安全的公网出口服务,例如: 云服务器需要与公网通信,但出于安全性考虑,不希望云服务器绑定公网 IP。

2.监控前准备

本次监控方式通过 zabbix 外部检查方式自动发现实例,监控项数据通过 zabbix 采集 器方式生成。将下面脚本上传至外部检查路径下。

部署路径有差异,根据实际环境的外部检查路径放至脚本。

#!/usr/bin/python3
import json

from tencentcloud.common import credential from tencentcloud.common.profile.client profile import ClientProfile from tencentcloud.common.profile.http_profile import HttpProfile from tencentcloud.common.exception.tencent cloud sdk exception import TencentCloudSDKException from tencentcloud.vpc.v20170312 import vpc client, models import sys from os import popen from zbxsend import ZabbixMetric, ZabbixSender akey = sys.argv[1] ckey = sys.argv[2] region = sys.argv[3] ZABBIX_SEND_ADDR = '/itops/zabbix/bin/zabbix_sender' ZABBIX_SERVER_IP = '127.0.0.1' ZABBIX PORT = '10051' def get_data(offset): $params = {$ "Limit": 100, "Offset": offset req.from_json_string(json.dumps(params)) # 返回的 resp 是一个 DescribeNatGatewaysResponse 的实例,与请求对象对应 resp = client.DescribeNatGateways(req) # 输出 json 格式的字符串回包 res = resp.to_json_string() data = json.loads(res) disklist = data['NatGatewaySet'] # 获取记录总条目数 totalcount = int(data['TotalCount']) return totalcount, disklist try: # 实例化一个认证对象,入参需要传入腾讯云账户 SecretId 和 SecretKey # 密钥可前往官网控制台 https://console.cloud.tencent.com/cam/capi 进行获取 cred = credential.Credential(akey,ckey) # 实例化一个 http 选项,可选的,没有特殊需求可以跳过 httpProfile = HttpProfile() httpProfile.endpoint = "vpc.tencentcloudapi.com" # 实例化一个 client 选项,可选的,没有特殊需求可以跳过 clientProfile = ClientProfile() clientProfile.httpProfile = httpProfile # 实例化要请求产品的 client 对象, clientProfile 是可选的

```
# 实例化一个请求对象,每个接口都会对应一个 request 对象
   req = models.DescribeNatGatewaysRequest()
   totalcount,disklist = get data(0) ## 获取 100 条以内的 disk 数据
   if totalcount > 100: ## 如果 disk 记录大于 100, 则将 100 往后的数据拼接至 list
       if totalcount % 100 > 0:
           num = totalcount // 100 + 1
       else:
           num = totalcount // 100
       for i in range(num):
           if i == 0:
               continue
           else:
               disklist = disklist + get data(i*100)[1]
   list disk = []
   for i in disklist:
       nat id = i['NatGatewayId']
       nat name = i['NatGatewayName']
       nat name = nat name+nat id
       list disk.append({"{#NATID}":nat id,"{#NATNAME}":nat name})
       keys_list = list(i.keys())
       for key in keys list:
           result = ZabbixSender(ZABBIX_SERVER_IP).send([ZabbixMetric(nat_id,key,str(i[key]))])
   print(json.dumps(list disk))
except TencentCloudSDKException as err:
```

print(err)

注意以下三处的值根据实际环境调整

ZABBIX_SEND_ADDR: zabbix_sender 路径

ZABBIX SERVER IP: server 服务器 IP 地址

ZABBIX_PORT: server 服务器端口

```
10 from zbxsend import ZabbixMetric, ZabbixSender
11 akey = sys.argv[1]
12 ckey = sys.argv[2]
13 region = sys.argv[3]
14 ZABBIX_SEND_ADDR = '/itops/zabbix/bin/zabbix_sender'
15 ZABBIX_SERVER_IP = '127.0.0.1'
16 ZABBIX_PORT = '10051'
```

3.监控模板制作

3.1.新增自动发现规则用于自动发现 NAT 网关实例

ZABBIX « 🗈	自动发现规则
٩	所有機板/ 展讯云 NAT网关 自动出现横板 自动发现消息 / NAT网关农谷自动发现 监控调察型 酸洗液型 医洗原型 主机横板 1
 ● 监测 - 	自动投现现则 進程 LLD macros 过滤器 医盖
③服务 -	"名称 NAT网关名你自动发现
💷 资产记录 🗸	新聞 外部位置 ・
11 报表 ~	* 1818 tencent.NAT.py[[\$ACCESS_KEY],[\$BECRET_KEY],[\$REGION]]
く 配画	*更新问题 1d
主机器料组	自起义的间间得 类型 同语 期间 动作 动作 动作 动作 动作 网络
横板	
主机	· 街頭陶解不足 2d
able >	服 送
关联项事件	
自动发现	
管理 ~	
り 支持	使新 克隆 测试 删除 取消
Integrations	
? 帮助	
L User settings -	
し退出	Zabbix 6 0 5 @ 2001-2022, Zabbix SIA Wit

键值解释:

tencent.NAT.py: 脚本名称, 可自定义脚本名称, 需要与外部检查路径下的脚本名称一致

\$ACCESS_KEY}: SecretId

{\$SECRET_KEY}: SecretKey

可前往官网控制台 https://console.cloud.tencent.com/cam/capi 获取 SecretId 和

SecretKey

{\$REGION}:资源地域,表示操作的资源所属的地域

3.2.新增监控数据接收的监控模板

监控项名称入口:



文档中心 入门中心 API中心	SDK中心 我的反馈		文档提出大赛:大数据专题> 1101 提家本产品组》文档	Q	
API ΦΟ V	NatGateway NAT例关闭象。 彼如下使口可用: CreateNatGateway, Des	cribeNatGatewaya,		LirectLonnectGatewayUc nRouta DirectConnectSubnet	
世界码 *6.4000 ADM 2014 7	名称	关型	编述	EndPoint	
3.日内日にから2000 く 設置計算 3.0 し	NatGatewayid	String	NAT网经会00。 可够调整: "nat-12345678"	EndPointService Filter FilterObject	
₩1500× 30 0 API 阿关 30 0	NatGatewayName	String	NAT网关约名称。 示例编: "hos"	FlowLog FlowLogStorage	
对象存储 ~ · · · · · · · · · · · · · · · · · ·	CroatedTime	String	NATN93X包括#29931月。 开册编:"2021-05-24 19:28:31"	GatewayFlowMonitorDeta ii	
文件存储 30 · · · · · · · · · · · · · · · · · ·	State	String	NATRONOPUSS、 PENDING: 並严中, 'DELETING: 删除中, 'AVAILABLE: 這 行中, 'UPDATING: 并统中, 'TALED: "大利L 示例道: 'AVAILABLE'	Gatawaytuos Haktip IKEOptionsSpecification IPSECOptionsSpecificati on	•
云数编库 MySQL 30	InternetMaxBandwidthOut	Integer	网次载大外网出带宽(单位Mbps)。 示例编: 10	InstanceStatistic	
云数調準 SQL Server 3.8 · ·	MaxConcurrentConnection	Integer	网队并发生换上啊。 示例语:1000000	IpGRuleInfo IpGTranslator	四 文档 2
云政法库 Redis 3.0 ~	PublicipAddressSet	Array of NatGatewayAddress	振动NAT网关的公司IP对象数组。 示例值:无	tpAddressStates lpField	
内容分发网络 CDN 3.9 云数国际 Memcached 3.0 云数国际 MongoDB 3.0	NetworkState	String	NAT词头网络伏杰,"AVAILABLE"运行中,"UNAVAILABLE"不可用,"INSUFFICIENT"次要存职。 示例值:"AVAILABLE"	IpGeolocationInfo Ipv6Address Ipv6SubnetCidtBlock	立 文档活动
TDSQL MySQLI 30	DestinationIpPortTranslationNatRuleSet	Array of Destination/pPortTranslationNatRule	NATREXEMPLACEMENT.	ItemPrice LocalGateway Memberlefo	ø
金球应用加速 30	Vpcld	String	VPC卖的回。 可说明道: "vpc-12345678"	NatDirectConnectGatewa yRoute	
MINICOMPTE 30	7	Dalas	NAT网关所在的可用区。	NatGateway	<u>.</u>

ZABBIX « S	监控项								0682390
٩	所有统权 / 资	讯云 NAT网头 模板 盐拉顶 18 触发器 图形		Web 场景					itisa V
 ● 第時 ○ 服务 三 河へ行政 ○ 服务 三 河へ行政 ○ 服务 二 河へ行政 二 戸へ行政 	(11) Loop (1) 20	主が近くには <u>100</u> 100 100 主が詳細 使用 単規 単規 単規 の対		道择 送月章 送月章			 →) (@ 	95 3	12,918
	世滅器 日間4 倍息受型 美符 15 数字。	向け3月后が(数)留 (天正気) 3		键(直填写上图红框中的	名称	类型约	在一选择Zabbix采集器	
关联项事件 自地发现		名称▲ NAT网关创建的时间 NAT网关创建的时间	1	57 8	RestedTime	inite Inite	历史记录 植物 90d	米型 状态 Zabbo采集器 日用用 Turbur引きる 日本目	632
	····	NAT网大大服本标 NAT网关所在的可用区 NAT网关是否很刻禁			Zone RestrictState		90d 90d 90d	Zabbix梁建建 日日用 Zabbix梁建建 日日用	
	- ···	NAT网关的SNAT都是规则 NAT网关的状态			SourcelpTranslationNatRuleSet State		90d 90d	Zabbox深筆器 日和用 Zabbox深筆器 日和用 Zabbox深筆器 日和用	
		NAT和天的論L報发用。 NAT和天明是的安全组列表 NAT和天明描状态			DestinationipiPortTranslationNatRuleSe SecurityGroupSet NetworkState		90d 90d 90d	Zabbox梁建設 Zabbox梁建設 Zabbox梁建設 Zabbox梁建設 Zabbox梁建設	
0 3is	····	VPC5098D 新國子网ID			Vpcld SubnetId		90d 90d	Zabbox 梁集器 已回用 Zabbox 梁集器 日用用	
El Integrations ? 4651	0	是否他学型NAT 标签编编符			IsExclusive TagSet		90d 90d	Zabbox采集器 日间用 Zabbix采集器 日用用	
土 User settings 、 し 訳出		接手至NAT時在ISI時又素斟的带克 他ENAT時并的公開戶的最前級 構造的各項用关ID			ExclusiveGatewayBandwidth PublicipAddrossSat DirectConnectGatewayIds		90d 365d 90d 90d	Zabbix采集器 已回用 Zabbix采集器 已回用 Zabbix采集器 已回用	演成

设置完成后,等待实例自动发现即可。

大家好,我是乐乐,专注运维技术研究与分享,关注我学习更多 Zabbix 等使用技巧,更多运维问题还可以到乐维社区留言提问。乐维社区是运维监控领域的垂直社区,专注打造 Zabbix 等技术栈的线上交流与共享平台,每周三下午,社区关联答疑群(如下图)还提供免费专家在线答疑,欢迎小伙伴们加入。



七十九、使用 Zabbix 监控 openstack 的系统资源 概述

OpenStack 是一个开源的云计算管理平台项目,是一系列软件开源项目的组合。由 NASA(美国国家航空航天局)和 Rackspace 合作研发并发起,以 Apache 许可证(Apache 软件基金会发布的一个自由软件许可证)授权。

OpenStack 为私有云和公有云提供可扩展的弹性的云计算服务。项目目标是提供实施 简单、可大规模扩展、丰富、标准统一的云计算管理平台。

对于 Openstack 运维人员来说,需要掌握 Openstack 云平台系统资源的整体运行情况,包括域(AZ)的 cpu/menmory 等资源使用情况;本文将介绍使用 zabbix 结合 openstack python-api 对 openstack 系统资源进行监控,详情如下:

使用控制台获取域(AZ)监控信息

nova availability-zone	e-list	
+	-++	
Name	Status	I
+	-++	
internal	available	I
- computer03	I	I
- nova-storage	enabled :-) 2018-10-18T10:56:07.00000	00

使用如下命令可以获取当前 openstack 的所有可用域。

- computer04	I	Ι
- nova-storage	enabled :-) 2018-10-18T10:56:08.00000	0
- computer05	I	Ι
- nova-storage	enabled :-) 2018-10-18T10:56:13.00000)
- computer06	I	Ι
- nova-storage	enabled :-) 2018-10-18T10:56:05.00000	D
- computer07	I	Ι
- nova-storage	enabled :-) 2018-10-18T10:56:05.00000	D
- computer08	I	Ι
- nova-storage	enabled :-) 2018-10-18T10:56:07.00000	D
- controler02	I	I
- nova-conductor	enabled XXX 2017-10-09T09:46:27.000	000
- nova-consoleaut	n enabled XXX 2017-10-09T09:46:37.0000	000
- nova-monitor	enabled :-) 2018-10-18T10:56:11.00000	0
- nova-scheduler	enabled XXX 2017-10-09T09:46:37.0000	000
- nova-cert	enabled XXX 2017-10-09T09:46:37.0000	00
- controller01	I	
- nova-conductor	enabled :-) 2018-10-18T10:56:11.00000	0
- nova-consoleaut	n enabled :-) 2018-10-18T10:56:04.00000	0
- nova-monitor	enabled :-) 2018-10-18T10:56:07.00000	0
- nova-scheduler	enabled :-) 2018-10-18T10:56:09.00000)
1		

IMS	available	I
- computer03	I	I
- nova-compute	enabled :-) 2018-10-18T10:56:07.00000	0
- computer04	I	I
- nova-compute	enabled :-) 2018-10-18T10:56:05.00000	0
- computer08	I	I
- nova-compute	enabled :-) 2018-10-18T10:56:13.00000	0
paas	available	I
- computer05	I	I
- nova-compute	enabled :-) 2018-10-18T10:56:07.00000	0
- computer06	I	I
- nova-compute	enabled :-) 2018-10-18T10:56:13.00000	0
- computer07	I	I
- nova-compute	enabled :-) 2018-10-18T10:56:04.00000	0
+	++	

由结果可以看出,该系统含有两个域: IMS 和 paas 。

但是该命令结果不太直观,不容易提取,我们也可以使用如下命令:

nova aggrega	te-list
++	-++
Id Name	Availability Zone
++	-++

1 IMS+RCS IMS		
4 paas paas		
++	+	

根据 aggregate-list ,我们可以查看域对应的计算节点:

nova aggregate-details paas
++++++
+
Id Name Availability Zone Hosts
Metadata
+++++
+
4 paas paas 'computer07', 'computer05', 'computer06'
'availability_zone=paas'
++++++
 +

由结果可知, paas 域, 一共包含三个 computer 节点, 对于每个节点的使用情况, 我

们一样可以通过命令获取:

nova hypervisor-show computer07			
++	+		
Property	Value	I	

+	+	F
free_disk_gb	55	
free_ram_mb	126156	I
host_ip	193.2.0.37	7
hypervisor_hostname	compute	er07
hypervisor_type	QEMU	Ι
id	9	I
local_gb	445	I
local_gb_used	390	I
memory_mb	25774	0
memory_mb_used	13158	34
npt_ept	ept	I
pci_pools	-	I
running_vms	8	I
service_disabled_reason	-	
service_host	computer	07
service_id	27	
state	up	
status	enabled	I
vcpus	48	
vcpus_used	32	I
+	+4	÷

#截取部分信息

我们可以得知,该节点的 vcpu 总数及其使用情况,memory_mb 总数及其使用情况。

将每个节点的使用情况均获取后,经过计算,就可以得出域的资源整体使用情况。对于 云平台的整体使用情况,我们也可以通过命令行获取:

nova hypervisor-stats			
+	+	+	
Property	Value		
+	+	+	
count	6	I	
current_workload	0	I	
disk_available_least	304		
free_disk_gb	518		
free_ram_mb	96173	6	
local_gb	2670	Ι	
local_gb_used	2152	Ι	
memory_mb	1540	5440	
memory_mb_used	584	704	
running_vms	42	Ι	
vcpus	288		

vcpus_used	172	I		
+	++			

上面我们讨论了使用命令行进行相关信息的获取,下面我们讨论使用 python-api 进行

相关信息的获取与计算。

使用 openstack python-api 获取域监控信息

程序代码如下:

#!/usr/bin/python			
# -*- coding: utf-8 -*-			
#imports			
import json			
from optparse import OptionParser			
from novaclient import client as noclient			
from novaclient import utils			
#登录及授权			
keystone = {}			
keystone['os_username']='admin'			
keystone['os_password']='keystone'			
keystone['os_auth_url']='http://lb-vip:5000/v2.0/'			
keystone['os_tenant_name']='admin'			

```
nova client
                            noclient.Client(2,
                                                    keystone['os_username'],
                  =
keystone['os password'], keystone['os tenant name'], keystone['os auth url'])
def main():
   #获取云平台整体信息并打印
   total_info = nova_client.hypervisor_stats.statistics()._info.copy()
   print "total_info_vcpus:", total_info["vcpus"]
   print "total info vcpus used:", total info["vcpus used"]
   print "total_info_memory_mb:", total_info["memory_mb"]
   print "total info memory mb used:", total info["memory mb used"]
    print "total info running vms:", total info["running vms"]
   #获取域列表信息
   aggregates = nova_client.aggregates.list()
   for aggregate in aggregates:
       #初始化每个域的资源统计变量
       vcpus = 0
       vcpus used = 0
       memory_mb = 0
       memory_mb_used = 0
       running_vms = 0
```



print "memory_mb_used:", memory_mb_used

print "running_vms:", running_vms

if __name__ == "__main__":

main()

执行该程序后,可以获取各个域节点的信息:

total_info_vcpus: 288

total_info_vcpus_used: 172

total_info_memory_mb: 1546440

total_info_memory_mb_used: 584704

total_info_running_vms: 42

1 IMS+RCS IMS [u'computer04', u'computer03', u'computer08']

vcpus: 144

vcpus_used: 84

memory_mb: 773220

memory_mb_used: 243200

running_vms: 20

4 paas paas [u'computer07', u'computer05', u'computer06']

vcpus: 144

vcpus used: 88

memory_mb: 773220

memory_mb_used: 341504

running_vms: 22

经过适当的计算,我们就可以获取各个域分配及使用比例等信息。

上面我们就使用 python-api 打印出了所有需要的信息,但是对于监控来说,我们需要提取的是各个监控项的信息,这样才能方便的搜索和做图表展示;下面我们讨论结合 zabbix 进行相关信息的监控。

结合 zabbix 获取域相关监控信息

获取可用域信息列表

上面我们已经获取了所有的可用域信息,但对于 zabbix 来说,我们还需要返回固定格 式的数据,供 zabbix 进行解析:

#!/usr/bin/python
-*- coding: utf-8 -*-
#imports
import json
from optparse import OptionParser
from novaclient import client as noclient
from novaclient import utils

```
#登录及授权
keystone = {}
keystone['os username']='admin'
keystone['os password']='keystone'
keystone['os_auth_url']='http://lb-vip:5000/v2.0/'
keystone['os_tenant_name']='admin'
nova client
                             noclient.Client(2,
                   =
                                                     keystone['os username'],
keystone['os_password'], keystone['os_tenant_name'], keystone['os_auth_url'])
def main():
    r = {"data":[]}
    aggregates = nova_client.aggregates.list()
    for aggregate in aggregates:
        aggregate info = aggregate. info.copy()
        r['data'].append( {"{#NAME}":aggregate_info["name"]} )
    print(json.dumps(r, indent=2, sort keys=True, encoding="utf-8"))
if __name__ == "__main__":
    main()
```

执行后,返回的结果如下,可以通过供 zabbix 自动发现模版使用:

zabbix, Item 及自动发现 LLD 配置

云平台整体信息, Item 设置如下:

openstack.total[vcpus]

openstack.total[vcpus_used]

openstack.total[memory_mb]

openstack.total[memory_mb_used]

openstack.total[running_vms]

获取{#NAME}后就可以根据监控项获取对应的监控内容了, zabbix 自动发现自动发现

key 设置如下:

openstack.system.discovery

discovery 下,设置监控 Item,key 设置如下:

openstack.zone[hosts,{#NAME}]

openstack.zone[vcpus,{#NAME}]

openstack.zone[vcpus_used,{#NAME}]

openstack.zone[memory_mb,{#NAME}]

openstack.zone[memory_mb_used,{#NAME}]

openstack.zone[running_vms,{#NAME}]

其中{#NAME}为第一步的脚本中返回的可用域对应的 aggregate 相关信息。

zabbix-agent 配置与对应的脚本

对应的 zabbix-agent.conf 配置如下:

/etc/zabbix/zabbix_agentd.d/userparameter_openstack-system.conf

UserParameter=openstack.system.discovery,python

/etc/zabbix/zabbix_agentd.d/openstack-system.py --item discovery

UserParameter=openstack.total[*],python

/etc/zabbix/zabbix_agentd.d/openstack-system.py --item total --moniter \$1

UserParameter=openstack.zone[*],python

/etc/zabbix/zabbix_agentd.d/openstack-system.py --item \$1 --aggregate \$2

对应的 zabbix 自动发现的监控脚本如下:					
# /etc/zabbix/zabbix_agentd.d/openstack-system.py					
#!/usr/bin/python					
# -*- coding: utf-8 -*-					
#imports					
import json					
from optparse import OptionParser					
from novaclient import client as noclient					
from novaclient import utils					
#登录及授权					
keystone = {}					
keystone['os_username']='admin'					
keystone['os_password']='keystone'					
keystone['os_auth_url']='http://lb-vip:5000/v2.0/'					
keystone['os_tenant_name']='admin'					
nova_client = noclient.Client(2, keystone['os_username'],					
keystone['os_password'], keystone['os_tenant_name'], keystone['os_auth_url'])					
def main():					

```
options = parse_args()
    if options.item = = "discovery":
        zone list()
    elif options.item = = "total":
       total moniter(options)
    else:
        zone_moniter(options)
#判断入参合法性
def parse args():
    parser = OptionParser()
   valid item = ["discovery", "total", "hosts", "vcpus", "vcpus used",
"memory_mb", "memory_mb_used", "running_vms"]
    parser.add_option("", "--item", dest="item", help="", action="store",
type="string", default=None)
    parser.add option("", "--moniter", dest="moniter", help="", action="store",
type="string", default=None)
    parser.add_option("", "--aggregate", dest="aggregate", help="",
action="store", type="string", default=None)
    (options, args) = parser.parse_args()
    if options.item not in valid_item:
        parser.error("Item has to be one of: "+", ".join(valid_item))
```

return options #获取可用域列表 def zone list(): r = {"data":[]} aggregates = nova_client.aggregates.list() for aggregate in aggregates: aggregate_info = aggregate._info.copy() r['data'].append({"{#NAME}":aggregate_info["name"], "{#AVAILABLE ZONE}":aggregate info["availability zone"]}) print(json.dumps(r, indent=2, sort_keys=True, encoding="utf-8")) #获取云平台整体监控信息 def total_moniter(options): total_info = nova_client.hypervisor_stats.statistics()._info.copy() print (total_info[options.moniter]) #获取可用域对应的监控信息 def zone_moniter(options): aggregate = utils.find_resource(nova_client.aggregates, options.aggregate) aggregate_info = aggregate._info.copy()

```
aggregate_hosts = aggregate_info["hosts"]
if options.item=="hosts":
    print (aggregate_hosts)
else:
    monitor_data = 0
    for aggregate_host in aggregate_hosts:
        hypervisor_info = utils.find_resource(nova_client.hypervisors,
aggregate_host)._info
        monitor_data = monitor_data + hypervisor_info[options.item]
    print (monitor_data)

if __name__ == "__main__":
    main()
```

在 zabbix 上进行对应的配置后重启,将模版应用于主机,此时应当监控获取所有的可 用域,并监控对应的信息。

参考资料

- 1. Openstack 中的 zone ,aggregates 和 host 及 其 应 用,https://blog.csdn.net/ztejiagn/article/details/8948688
- 2. nova 命 令 汇 总 四 — 计 算 相 关 命 令,http://blog.51cto.com/13788458/2129157
- 3. The novaclient Python

API,https://docs.openstack.org/python-novaclient/latest/reference/api/in dex.html

4. GitHub

larsks/openstack-api-samples,https://github.com/larsks/openstack-api-samples

_

以上就是这一期的 Zabbix 技术分享内容。

八十、更多……

Docker 容器

八十二、如何使用 docker 快速部署一个 zabbix 监控系统

使用 docker 拉取官方提供的 zabbix 镜像来快速部署一个 zabbix 监控系统

1.环境准备

Centos7 系统

服务器 IP: 192.168.75.31

2.安装 docker

2.1. Yum 安装

yum -y install docker

2.2. 二进制安装 (可离线)

Docker 二进制包下载,也可在 windows 下载后再上传至服务器

wget https://download.docker.com/linux/static/stable/x86_64/docker-20.10.14.tg

Ζ

 Solvard Solvard

tar -zxf docker-20.10.14.tgz

```
[System not activated][root@localhost ~]#
[System not activated][root@localhost ~]#
[System not activated][root@localhost ~]# tar -zxf docker-20.10.14.tgz
[System not activated][root@localhost ~]# ls
docker docker-20.10.14.tgz
[System not activated][root@localhost ~]#
```

复制文件

cp -ra docker/* /usr/bin

```
docker docker-20.10.14.tgz
[System not activated][root@localhost ~]# cp -ra docker/* /usr/bin/
[System not activated][root@localhost ~]# cd docker/
[System not activated][root@localhost docker]# ls
containerd containerd-shim containerd-shim-runc-v2 ctr docker dockerd docker-init docker-proxy runc
```

编写 system 文件

vim /etc/systemd/system/docker.service

[Unit]

Description=Docker Application Container Engine

Documentation=http://docs.docker.io

[Service]

OOMScoreAdjust=-1000

ExecStart=/usr/bin/dockerd

ExecStartPost=/sbin/iptables -I FORWARD -s 0.0.0.0/0 -j ACCEPT

ExecReload=/bin/kill -s HUP \\$MAINPID

Restart=on-failure

RestartSec=5


启动 docker

systemctl daemon-reload

systemctl start docker



3.拉取 docker 镜像

访问 zabbix 官方提高的 docker 镜像

https://www.zabbix.com/container_images



这里选择使用 MySQL 数据库的 zabbix-server, zabbix-agent 以及 zabbix 界面的三个镜像,

zabbix 版本为最新的 6.2



Zabbix6版本以上的 MySQL 数据库需为 MySQL 8以上,因此拉取 MySQL 官方提供的

MySQL 8 镜像。

拉取四个镜像命令: (linux 上执行)

docker pull mysql:latest

docker pull zabbix/zabbix-server-mysql:latest

docker pull zabbix/zabbix-web-nginx-mysql:latest

docker pull zabbix/zabbix-agent:latest

•	System nat activated[[reacted.ac:]hest .lt docker.null_mysel.latest
	Jatest Pulling from Jbray (mysg)
	astalodbodb Rull complete
	7675754909 Rull complete
	Jula 250/JTACA Pull complete
	aquission//wo.rell.complete
	solosantaussi, Rull complete
	allabeeside full complete
	1322/edub122: Pull complete
	D40430001903: Full complete
	120212810C32: Pull complete
	deb564215195: Full complete
	dSSS8a///d0/: Pull complete
	455689080009: Pull complete
	Digest: sna256:D953201edea/206Cee12d915a/854/003812ea5d0842566e1/18033291ed2921
	Status: Downloaded newer image for mysql:Latest
	docker.10/library/mysdi:latest
	lesster not activated [root@localnost ~]# docker pull ZabbiX/ZabbiX-Server-mysql:latest
	Latest: Putting from ZabbiX/ZabbiX-server-mysql
	213ec9aee270: Pull complete
	9484/6a4203: Pull complete
	ebbc2985f6d2: Pull complete
	76ac7d2089c9: Pull complete
	fe0137176975: Pull complete
	c408cc9e8da5: Pull complete
	1c55d56cfe7c: Pull complete
	4f4fb700ef54: Pull complete
	c731d2d69475: Pull complete
	Digest: sha256:8e616db62b75daeac9260d39f921d9ddf021eb613b32353a3b991ee17efcddd9
	Status: Downloaded newer image for zabbix/zabbix-server-mysql:latest
	docker.io/zabbix/zabbix-server-mysql:latest
	[System not activated][root@localhost ~]# docker pull zabbix/zabbix-web-nginx-mysql:latest
	latest: Pulling from zabbix/zabbix-web-nginx-mysql
	213ec9aee27d: Already exists
	3324412a7d2b: Pull complete
	3721c33b0791: Pull complete
	d790b14bebc9: Pull complete
	4f4fb700ef54: Pull complete
	0c31154cbbbd: Pull complete
	Digest: sha256:5c5c70d2529b34e6dfe0dac36f115da76f2ab3495fd8cea2193e8747c8916da1
	Status: Downloaded newer image for zabbix/zabbix-web-nginx-mysql:latest
	docker.io/zabbix/zabbix-web-nginx-mysgl:latest
	[System not activated][root@localhost ~]# docker pull zabbix/zabbix-agent:latest
	latest: Pulling from zabbix/zabbix-agent
	213ec9aee27d: Already exists
	1b601b806eee: Pull complete
	a20d3d7735e6: Pull complete
	176f3b1a0a49: Pull complete
	65f4014bf43c: Pull complete
	35afe5502e3a: Pull complete
	4f4fb700ef54 Pull complete
	JASSadocrán Pull complete
	Digest - 0/200/00/07/07/00/07/07/07/07/07/07/07/07/0
	Status Downloadal news image for zabiy/zabiy-zabiy/zabiy-latest
	dacker jo/ashbiy/ashbiy/ashbi/latest
	System not activated [rootal acabost _]#

命令执行完后查看镜像是否拉取成功

docker images [root@docker ~]# docker images REPOSITORY TAG IMAGE ID CREATED SIZE 43fcfca0776d 3 days ago 209092699757 9 days ago mysql latest 449MB zabbix/zabbix-web-nginx-mysql latest 211MB zabbix/zabbix-server-mysql latest 3e15d051ed63 9 days ago 82.2MB zabbix/zabbix-agent latest a9975d793793 9 days ago 15.4MB [root@docker ~]#

4.启动镜像

4.1. 首先要启动 MySQL 数据库

docker run --name zabbix-mysql -e MYSQL DATABASE=zabbix -e MYSQL ROO

T PASSWORD=zabbix -p 3306:3306 -d mysql:latest

参数解释:

--name zabbix-mysql #定义容器名称

-e MYSQL DATABASE=zabbix #初始数据库名 zabbix

-e MYSQL ROOT PASSWORD=zabbix #数据库 root 用户的密码 zabbix

-p 3306:3306 #将容器 3306 端口映射到主机 3306 端口

06:3306 -d mysql:lates 11325861d606ef11463d8 CREATED STATUS PORTS Up 1 second 0.0.0.3306->3306/tcp, :::3306->3306/tcp, 33060/tcp NAMES zabbix-mysql

4.2. 然后启动 zabbix-server

docker run --name zabbix-server -e MYSQL DATABASE=zabbix -e DB SERVER

HOST=192.168.75.31 -e DB SERVER PORT=3306 -e MYSQL USER=root -e MYS

QL PASSWORD=zabbix -p 10051:10051 -d zabbix/zabbix-server-mysql:latest

参数解释:

--name zabbix- server

#定义容器名称

-e DB SERVER HOST=192.168.75.31 #数据库连接地址

-e DB SERVER PORT=3306

#数据库连接端口

整理 by 乐维社区(https://forum.lwops.cn)



4.3. 最后启动 zabbix 界面

```
docker run --name zabbix-web -e ZBX_SERVER_HOST=192.168.75.31 -e ZBX_SE
RVER_PORT=10051 -e DB_SERVER_HOST=192.168.75.31 -e DB_SERVER_PORT=3
306
```

-e MYSQL_DATABASE=zabbix -e MYSQL_USER=root -e MYSQL_PASSWORD=za

bbix -p 8080:8080 -p 8443:8443 -d zabbix/zabbix-web-nginx-mysql:latest

参数解释:

```
--name zabbix- web #定义容器名称
-e ZBX_SERVER_HOST=192.168.75.31 #web 连接 server 的地址
-e ZBX_SERVER_PORT=10051 #web 连接 server 的端口
-e DB_SERVER_HOST=192.168.75.31 #数据库连接地址
```

整理 by 乐维社区(https://forum.lwops.cn)



5.访问 web

全部启动后等待 2-3 分钟,待数据库初始化完成。

然后访问 web 界面

http://IP:8080

	Cabbix docker
ZABBIX	
用户名称	
老弱	
☑ 30天内设任我	
£2	
· 如此	
© 2001-2022, Zabbu: SIA	

初始登录信息

账户:Admin

密码: zabbix

登录后的界面

ZABBIX « 🔊	Global view										? Edit deshlooerd 📃 👯
Zabbu docker	All dashboards / Glo	bal view									
Monitoring ~	Top hosts by CP	U utilization	50 200	Km avn Proce	PEDAS		System informa	ation	Value	Details	
Dashboard			No data found.			0.821	Zebbix server is ru	ning	Yes	192.168.164.133:10051	
Problems						Zabbix server Values per second	Number of hosts (e	nabled/disabled)	1	1/0	17.10
Hosts							Number of templati	15	310		17.13
Latest data							Number of items (e	nabled/disabled/not supp	orted) 99	91/0/8	Shanahai
Maps							Number of triggers	(enabled/disabled (proble	am/ok() 55	56 / 0 [0 / 56]	Shanghai
Discovery							Number of users (o	inlinė)	2	1	
🙆 Services 🐳							Required server pe	rformance, new values pr	er second 1.46		
:= Inventory -	Host availability			_	Problems by s	everity				Geomap	
Reports -										+	V
3	0 Available	1 Not available	0 Unknown	1 Total	0 Disester	0 High	O Average V	O C Arming Inform) () Not class	isified bo - to	12 Conto
Configuration -										Contes Demonster	
Administration ~	Current problem	s								o	P
	T	me 🔻 🛛 Info	Host	Problem • Severity		Duration	Ack	Actions	Tags		h the second
					No data	found.				52/2	0
										2	Rina
										SER C	M
										SP X	
										Surpristinge	ATO THE ATOM
O Support										Mar	upe /
E Integrations										Mar Ser Th	N. C.
7 Help										PIJZ	AA Balot
											Leaflet © OpenStreetMap contributors
🔟 User settings 🗸 🗸											
신 Sign out							ebix 5.2.2. © 2001-2022	Zabbix SIA			

修改成中文显示 _____

ZABBIX « 🗈	<u>GUI v</u>
٩	Default langua <mark>pe English (en_US) ~</mark>
Monitoring ✓	Default time zone System: (UTC+03:00) Europe/Riga 🗸
	Default theme Blue ✓
 Services - 	" Limit for search and filter results 1000
Ξ Inventory 🗸	* Max number of columns and rows in overview tables 50
🔥 Reports 🗸 🗸	* Max count of elements to show inside table cell 50
🔧 Configuration 🗸	Show warning if Zabbix server is down 🔽
Administration	* Working time 1-5,09:00-18:00
General <	GUI Show technical errors
Proxies	Autoregistration ax history display period 24h
Authentication	Housekeeping Time filter default period 1h
User groups	Audit log k period for time selector 2y
User roles	Icon mapping Update Reset defaults
Users	Regular expressions
Media types	Macros
Scripts	Trigger displaying options
Queue >	Geographical maps
	API tokens
	Other

6.添加 agent 监控

启动 agent

```
docker run --name zabbix-agent --link zabbix-server -e ZBX_HOSTNAME=192.
```

```
168.75.31 -e ZBX SERVER HOST=172.17.0.1 -e ZBX SERVER PORT=10051 -p 10
```

050:10050 --privileged -d zabbix/zabbix-agent:latest

参数解释:

--name zabbix-agent #定义容器名称

--link zabbix-server #连接到 zabbix-server 容器

-e ZBX_HOSTNAME=192.168.75.31 #zabbix-server 主机名

-e ZBX_SERVER_HOST=172.17.0.1 #zabbix-server 连接地址,这里固定这个 IP

-p 10050:10050 #将容器 10050 端口映射到主机 10050 端口

-d #后台运行

PS:这里 ZBX_SERVER_HOST 无须修改 IP

然后到 UI 界面将默认的 zabbix 监控修改一下 agent 的 IP 信息

主机							? ×				
主机	PMI 标记	宏 资产记录 • 加密 值映射									
	* 主机名称	Zabbix server									
	可见的名称	Zabbix server									
	模板	名称	动作				<u>1</u>				
		Linux by Zabbix agent	取消链接 取消链接并清理								
		Zabbix server health	取消链接 取消链接并清理								
		在此輸入搜索		选择							
	* 主机群组	Zabbix servers ×		选择							
		TTH AND / 13L PC	Contraction of the Contraction o					¥	志	可用性	а
	接口	兴型 IP地址	DNS名称	连接到	端口	默认		E	信用	ZBX	
		Agen 192.168.75.31		IP DN	S 10050	● 移除					
		添加									
	描述										
			1								
由agent	代理程序监测	(无agent) ×									
	日白田										
					再新	京勝 全京勝 新除	ED:H				
					Saderi	ADRA ADRA					

即可完成 agent 监控的添加

在监测>最新数据可看到监控正常获取数据

ø	112週 へ				另存为	应用 重没			
	仪表盘	过	波器 只影响过滤后的数据						
	问题	183	签值 monant application 1 cou 17 da	the collector 12 environment 2 internal process 20 memory 7 or 2 raw 1 storage	a 3 montant 35				
	主机		inponent. uppreaser i cpu i u	an concept to environment a memory occupated interiory i bard tarting	1 3 3 3 3 3 3 3				
	最新数据		王印	名称 *	報近程查记录	織新数据	更改	रेगर्ट	信息
	拓扑器		Zabbix server	Available memory	85	130.36 MB	+12 MB	component: memory	観形
	自动发现		Zabbix server	Available memory in % 📔	7s	8.9621 %	+0.8396 %	component: memory	题形
ō	服务 🗸		Zabbix server	Checksum of /eto/passwd	1m 36s	a87c9e04ee115c7ca6915		component: environ	历史记录
_	2010123		Zabbix server	Context switches per second	508	1336.1759	-2032.7045	component: cpu	國形
	an-illar v		Zabbix server	CPU guest nice time	485	0 %		component: cpu	國形
	报表 ~		Zabbix server	CPU guest time M	49s	0 %		component: cpu	開形
2	168 ·		Zabbix server	CPU idle time M	47s	97.358 %	+0.4617 %	component: cpu	開形
•	管理 、		Zabbix server	CPU interrupt time	46s	0.4712 %	+0.02543 %	component: cpu	觀形
_			Zabbix server	CPU lowalt time	53s	0.07572 %	-0.0168 %	component: cpu	觀形
			Zabbix server	CPU nice time	35	0 %		component: cpu	翻形
			Zabbix server	CPU softing time 12	45	0.2269 %	-0.05062 %	component: cpu	翻形
			Zabbix server	CPU steal time	58	0 %		component: cpu	图形
			Zabbix server	CPU system time	225	0.8154 %	-0.261 %	component: cpu	問形
			Zabbix server	CPU user time III	215	0.4792 %	-0.4544 %	component: cpu	图形

至此,整个系统的搭建到此结束。

要监控其他的服务器,可在服务器安装部署一个 agent,然后在配置>主机添加对应的监控 对象即可完成监控。

Agent 部署:将 Agent 打包成通用二进制部署包,以方便如何快速进行

Agent 客户端的安装,参考文章:如何快速部署 zabbix-agent 客户端

使用 docker-compose 工具进行部署(更快!):比单纯使用 docker 部 署 zabbix 更快拉起一个监控系统,而且管理起来更方便!参考:使用 docker-compose 快速部署 zabbix 监控系统

八十三、zabbix6 监控 k8s

1、创建 yaml 文件安装 metrics 监控组件,修改节点名称 (kubectl get nodes 可查看节点名)

vim metrics.yaml
apiVersion: v1
automountServiceAccountToken: false
kind: ServiceAccount
metadata:
labels:
app.kubernetes.io/component: exporter
app.kubernetes.io/name: kube-state-metrics
app.kubernetes.io/version: 2.4.2
name: kube-state-metrics
namespace: monitoring
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
labels:
app.kubernetes.io/component: exporter
app.kubernetes.io/name: kube-state-metrics
app.kubernetes.io/version: 2.4.2
name: kube-state-metrics
rules:
- apiGroups:
- **
resources:
- configmaps
- secrets
- nodes

- pods
- services
- resourcequotas
- replicationcontrollers
- limitranges
- persistentvolumeclaims
- persistentvolumes
- namespaces
- endpoints
verbs:
- list
- watch
- apiGroups:
- apps
resources:
- statefulsets
- daemonsets
- deployments
- replicasets
verbs:
- list
- watch
- apiGroups:
- batch
resources:
- cronjobs
- jobs
verbs:
- list
- watch
- apiGroups:
- autoscaling
resources:

- horizontalpodautoscalers verbs: - list - watch - apiGroups: - authentication.k8s.io resources: - tokenreviews verbs: - create - apiGroups: - authorization.k8s.io resources: - subjectaccessreviews verbs: - create - apiGroups: - policy resources: - poddisruptionbudgets verbs: - list - watch - apiGroups: - certificates.k8s.io resources: - certificatesigningrequests verbs: - list - watch - apiGroups: - storage.k8s.io resources:

- storageclasses	
- volumeattachments	
verbs:	
- list	
- watch	
- apiGroups:	
- admissionregistration.k8s.io	
resources:	
- mutatingwebhookconfigurations	
- validatingwebhookconfigurations	
verbs:	
- list	
- watch	
- apiGroups:	
- networking.k8s.io	
resources:	
- networkpolicies	
- ingresses	
verbs:	
- list	
- watch	
- apiGroups:	
- coordination.k8s.io	
resources:	
- leases	
verbs:	
- list	
- watch	
apiVersion: rbac.authorization.k8s.io/v1	
kind: ClusterRoleBinding	
metadata:	
labels:	

app.kubernetes.io/component: exporter
app.kubernetes.io/name: kube-state-metrics
app.kubernetes.io/version: 2.4.2
name: kube-state-metrics
roleRef:
apiGroup: rbac.authorization.k8s.io
kind: ClusterRole
name: kube-state-metrics
subjects:
- kind: ServiceAccount
name: kube-state-metrics
namespace: monitoring
apiVersion: v1
kind: Service
metadata:
annotations:
prometheus.io/scraped: "true" # 设置能被 prometheus 抓取到,因为不带这个 annotation prometheus-service-endpoints 不会
去抓这个 metrics
labels:
app.kubernetes.io/component: exporter
app.kubernetes.io/name: kube-state-metrics
app.kubernetes.io/version: 2.4.2
name: kube-state-metrics
namespace: monitoring
spec:
clusterIP: None # 允许通过 svc 来进行访问
ports:
- name: http-metrics
port: 8088
targetPort: http-metrics
- name: telemetry
- name: telemetry port: 8089

selector:
app.kubernetes.io/name: kube-state-metrics
apiVersion: apps/v1
kind: Deployment
metadata:
labels:
app.kubernetes.io/component: exporter
app.kubernetes.io/name: kube-state-metrics
app.kubernetes.io/version: 2.4.2
name: kube-state-metrics
namespace: monitoring
spec:
replicas: 1
selector:
matchLabels:
app.kubernetes.io/name: kube-state-metrics
template:
metadata:
labels:
app.kubernetes.io/component: exporter
app.kubernetes.io/name: kube-state-metrics
app.kubernetes.io/version: 2.4.2
spec:
nodeName: k8s-master-1 # 设置在 k8s-master-1 上运行
tolerations: # 设置能容忍在 master 节点运行
- key: "node-role.kubernetes.io/master"
operator: "Exists"
effect: "NoSchedule"
automountServiceAccountToken: true
containers:
- image: k8s.gcr.io/kube-state-metrics/kube-state-metrics:v2.4.2
- image: anjia0532/google-containers.kube-state-metrics.kube-state-metrics:v2.4.2

livenessProbe:
httpGet:
path: /healthz
port: 8088
initialDelaySeconds: 5
timeoutSeconds: 5
name: kube-state-metrics
ports:
- containerPort: 8088
name: http-metrics
- containerPort: 8089
name: telemetry
readinessProbe:
httpGet:
path: /
port: 8089
initialDelaySeconds: 5
timeoutSeconds: 5
securityContext:
allowPrivilegeEscalation: false
capabilities
drop:
- ALL
readOnlyRootFilesystem: true
runAsUser: 65534
runAsGroup: 65534
#fsGroup: 65534
serviceAccountName: kube-state-metrics



2、添加命名空间 monitoring

kubectl create namespace monitoring

3、执行 yaml 文件

kubectl apply -f metrics.yaml

注: 若在启动容器时提示拒绝连接,则在 yaml 文件中注释掉对应健康探针

如下为正常启动示例

[root@node1 ~]# kubectl get pods -n	kube-met	rics		
NAME	READY	STATUS	RESTARTS	AGE
kube-state-metrics-6b96bdf5fb-zkwc4	1/1	Running	0	45s

4、下载解压 helm-v3.9.4-linux-amd64.tar.gz 包,对 helm 命令进行软链

tar xvf helm-v3.9.4-linux-amd64.tar.gz

5、官网下载配置文件、模板进行 k8s agent 安装

https://git.zabbix.com/projects/ZT/repos/kubernetes-helm/browse?at=refs%2Fheads%2Frelease%2F6.0

(以下为官网配置步骤)

添加存储库:

helm repo add zabbix-chart-6.0 https://cdn.zabbix.com/zabbix/integrations/kubernetes-helm/6.0

导出图表到文件的默认值:helm-zabbix\$HOME/zabbix_values.yaml

helm show values zabbix-chart-6.0/zabbix-helm-chrt > \$HOME/zabbix_values.yaml

根据文件中的环境更改值。\$HOME/zabbix_values.yaml

列出群集的命名空间。

kubectl get namespaces

创建命名空间(如果群集中不存在)。monitoring

kubectl create namespace monitoring

在 Kubernetes 集群中部署 Zabbix。(如有必要,请更新 YAML 文件路径)。

helm install zabbix zabbix-chart-6.0/zabbix-helm-chrt --dependency-update -f \$HOME/zabbix_values.yaml -n monitoring

查看容器。

```
kubectl get pods -n monitoring
```

查看 Pod 的信息。

kubectl describe pods/POD_NAME -n monitoring

查看 Pod 的所有容器。

kubectl get pods POD_NAME -n monitoring -o jsonpath='{.spec.containers[*].name}*'

查看 Pod 的日志容器。

kubectl logs -f pods/POD_NAME -c CONTAINER_NAME -n monitoring

容器的访问提示。

kubectl exec -it pods/POD_NAME -c CONTAINER_NAME -n monitoring -- sh

卸载

要卸载/删除部署,请执行以下操作: zabbix

helm delete zabbix -n monitoring

6、修改模板所有与 IP、TOKEN 相关的宏 (IP 为 k8s 服务器 IP 地址,下附获取 TOKEN 方

式)

Kubernetes API componentstatuses endpoi	nt /api/v1/componentstatuses			
{\$KUBE.API.ENDPOINT.URL}	https://192.168.1.140.644 3/api	Т ~	更改	
Kubernetes API endpoint URL in the format	<scheme>://<host>:<port>/api</port></host></scheme>			
{\$KUBE.API.HTTP.CLIENT.ERROR}	2	T •	更改	₫ = Kubernetes API server by HTTP: "2"
用于触发器的HTTP客户端请求失败的最大数	対目			
{\$KUBE.API.HTTP.SERVER.ERROR}	2	T •	更改	⊈ = Kubernetes API server by HTTP: "2"
用于触发器的HTTP客户端请求失败的最大数	対目			
{\$KUBE.API.LIVEZ.ENDPOINT}	/livez	T •	更改	Kubernetes cluster state by HTTP: "/livez"
Kubernetes API livez endpoint /livez				
{\$KUBE.API.READYZ.ENDPOINT}	/readyz	T •	更改	X = Kubernetes cluster state by HTTP: "/readyz"
Kubernetes API readyz endpoint /readyz				
{\$KUBE.API.SERVER.URL}	https://192.168.1.140:644 3/metrics	Τ •	更	<u>x</u> = Kubernetes API server by HTTP: "https://192.168.1.14
实例URL				
(\$KUBE.API.TOKEN)	eyJhbGclölJSUz11NilsT mbZClölmlXd3FrdHFN b185V2pONHBPUjg1d XZoam1ueGLM0J4U1 9lTK5vMRGZCsifLey Jpc3MicJidWJLcmBidG VzL3MicnZpY2VhY2Nv dW50lwia3VIZXJuZXRI cy5pby9zZXJ22WNYW Njb3VudC9uYVHIc3Bh Y2UiGJirdWJLXK5c3R IbSIsImt1YmVybmV02 XMuaW8vc2VydmjZW FIY291bn0vc2VicmV0L	Τ~	更已	⊈ = Kubernetes API server by HTTP: "eyJhbGciOiJSUzI1 = "eyJhbGciOiJSUzI1NiIsImtp

TOKEN 获取方式:

k8s 服务器执行命令

kubectl describe secrets \$(kubectl get secrets -n monitoring grep zabbix-servi
ce-account grep -v zabbix-service-account-token-hctrs cut -f1 -d ' ') -n mo
nitoring grep -E '^token' cut -f2 -d':' tr -d '\t' tr -d ' '

注: 宏中调用的 API 请求地址为默认端口, 若接口无数据返回, 可执行 kubectl config view

|grep server|cut -f 2- -d ":" | tr -d " "获取当前环境的 API 地址

八十四、监控平台组件 docker 编译使用手册

一、docker 安装

1. 安装 docker

curl -fsSL https://get.docker.com | bash -s docker --mirror Aliyu

2.启动 docker 应用

systemctl start docker && systemctl enable docker

二、SQL 安装

MySQL 与 PostGreSQL 只选其一安装即可

1、mysql 安装

使用方法

1、先创建对应用户

useradd mysql -u20223 -s /sbin/nologin

2、上传 mysql5.7.36.tar 与 data.tar 文件

mysql5.7.36.tar 为镜像文件; data.tar 为数据库文件和配置文件

3、导入镜像

docker load -i mysql5.7.36.tar

3.解压 data.tar

mkdir -p /itops/

tar -xf data.tar -C /itops/

chown mysql.mysql /itops/mysql -R

4.运行

docker run -itd -p3306:3306 -v /itops/mysql/etc:/etc/mysql/conf.d

-v/itops/mysql/data:/var/lib/mysql lw_mysql:5.7.36

账号: root

密码: ITIM_p@ssw0rd

账号: zabbix

密码: zabbix

2、postgresql 安装

使用方法

1、先创建对应用户

useradd postgres -u20224 -s /sbin/nologin

2、上传 postgresql12.8.tar 与 data.tar 文件

postgresql12.8.tar 为镜像文件 ; data.tar 为数据库文件和配置文件

3、导入镜像

docker load -i postgresql12.8.tar

4、.解压 data.tar

mkdir -p /itops/

tar -xf data.tar -C /itops/

chown postgres.postgres /itops/postgres -R

5、.运行

docker run -itd -v/itops/postgres/data:/var/lib/postgresql/data -v

/itops/postgres/etc/postgres.conf:/etc/postgresql/postgresql.conf -p5432:5432

lw_postgres:12.8

三、server 安装

使用方法

1、先创建对应用户

useradd zabbix -u20222 -s /sbin/nologin

2、上传 zabbix_server.tar 与 conf.tar 文件

zabbix_server.tar 为镜像文件; conf.tar 为 server 配置文件和日志等目录

3、导入镜像

docker load -i zabbix_server.tar

4、解压 conf.tar

mkdir -p /itops/zabbix_server

tar -xf conf.tar -C /itops/zabbix_server

5、替换本机 IP 地址

sed -i 's/192.168.229.17/本机 ip/g' /itops/zabbix_server/etc/zabbix_server.conf

sed -i 's/192.168.229.17/本机 ip /g' /itops/zabbix_server/etc/zabbix_agentd.conf

6、修改目录属主

chown zabbix.zabbix /itops/zabbix_server -R

7、运行

docker run -itd -p10050:10050 -p10051:10051 -p10052:10052

-v/itops/zabbix_server/:/itops/zabbix/ zabbix_server:5.0.16

四、Nginx 安装

使用方法

1、上传压缩包 nginx-v1.3.tar.gz 与 itops_v1_4_x86_64.tar 文件

itops_v1_4_x86_64.tar 为镜像文件; nginx-v1.3.tar.gz 为 nginx 配置文件和日志等目录

2、导入镜像

docker load -i itops_v1_4_x86_64.tar

3、解压 nginx-v1.3.tar.gz

mkdir -p /itops/

tar -xf nginx-v1.3.tar.gz -C /itops/

4、修改配置文件

sed -i 's/192.168.126.151/本机

ip/g' /itops/nginx/html/zabbix/conf/zabbix.conf.php

sed -i 's/192.168.126.151/本机

ip/g' /itops/nginx/html/lwjk_v3/web/z/conf/zabbix.conf.php

sed -i 's/192.168.126.151/本机 ip/g' /itops/nginx/html/lwjk_v3/config/db.php

5、运行

docker run -d -v /itops/nginx/etc:/itops/etc -v

/itops/nginx/html:/itops/nginx/html -p 80:80 -p 8081:8081 itops:v1.3

/itops/php/sbin/php-fpm --fpm-config /itops/etc/php/php-fpm.conf

/itops/nginx/sbin/nginx -c /itops/etc/nginx/nginx.conf -g "daemon off;"

五、agent 安装

使用方法

1、上传 zabbix_agent.tar conf.tar

zabbix_agent.tar 为镜像文件; conf.tar 为 agent 配置文件和日志目录

2、导入镜像

docker load -i zabbix_agent.tar

3、解压 conf.tar

mkdir -p /itops/zabbix_agent

tar -xf conf.tar -C /itops/zabbix_agent

chown zabbix.zabbix /itops/zabbix_agent -R

4、修改配置文件

sed -i 's/127.0.0.1/本地 ip/g' /itops/zabbix_agent/etc/zabbix_agentd.conf

5、运行

docker run -itd -p10050:10050 -v/itops/zabbix_agent/etc/:/itops/zabbix/etc/ -v/itops/zabbix_agent/logs/:/itops/zabbix/logs/

-v/itops/zabbix_agent/scripts/:/itops/zabbix/scripts/ zabbix_agent:5.0.16

6、登录 web 界面

http://本机 ip

账号: Admin

密码: zabbix

八十五、更多……

可视化

八十八、Zabbix 聚合图形配置指引

作为一款企业级监控工具,Zabbix 支持提供图形化的报表和图形展示功能。用户可以 使用 Zabbix 聚合图形 (aggregate graph) 来汇总和展示多个数据项的图形,将多个监控 指标放在同一个图形中,以便更好地展示整体趋势和变化。

本文将详细介绍 Zabbix 6 版本聚合图形配置的过程:

1.创建一个新的仪表板:在 Zabbix 的 Web 界面中,选择"监测">"仪表板">"添加仪表盘">"创建仪表盘",填写仪表盘名称后点击"应用"按钮即可。

ZABBIX « 🔊	新仪表盘1			非 · 添加 · 保存设置 取消
٩	添加仪表盘 / 新仪表盘1	仪表板属性		×
● 监测 ^	Page 1 ····	* 所有者	Admin (超级管理员 Administrator) × 透择	
仪表板		* 名称	新仪表盘1	
问题		Default page display period	1 hour 🔗	
最新数据	o	Start slideshow automatically		
拓扑图	1年为日朝行构州中		成用 取消	
自动发现				
③ 服务 ~				
≔ 资产记录				
🖬 报表 🗸				
へ配置				
管理 ~				
9 支持				
Integrations				
? 帮助				
💄 User settings 🗸 🗸				
の簡単			Zabbar 6.0.6. @ 2001-2022, Zabbar SIA	

2.创建一个新的聚合图形:点击空白位置,"类型"选择"图形",填写构件"名称"。

ZABBIX « 🔊	新仪表盘1	✿ ◆添加 > 保存设置 取消
٩	/#anotexe / 1 添加的9件	
• 监测 •	Page 1 ···· 與型 更形 v Show header V	
仪表板	名称 监控服务器磁盘使用量	
问题	周期间隔 2分 >>>	
主机	1	
最新数据		
拓扑蜀		
自动发现		
Õ 1885 -		
≔ 资产记录 →		
🚺 报表 🗸 🗸		
く 配置 -	0	
管理 -	8-16 21 X1 8-16 21 X7 8-16 21 54 8-16 22 00 8-16 22 07 8-16 22 13 8-16 22 20 8-16 22 28 8-16 22 33 8-16 22 39	
	数据设置1 显示忠项 时间编词 坐标结 题例 • 问题 覆盖	
9 支持	数据设置 主机样式 遗择 监控领领式 遗释 ×	
Integrations	Base color 記述 是出版 视为 0	
? 帮助	55 F 2010 F 2010 F 350 F 350	
🔔 User settings 🗸		
ら 脂田	Zabbox 6.0.6. @ 2001-2022, Zabbox SJA	

3.添加监控项:在"数据设置"标签页下,分别单击 2 个"选择"按钮,分别设置监控 主机和监控项。可以选择多个监控项,并为数据集设置颜色。Aggregation function 设置 为 sum 并且 Aggregate 设置为数据设置,即表示将监控项数据进行汇总计算。





也可根据需要,调整展示的数据方式,可选最大值、最小值、平均值、最新值等。



4.配置显示选项:在"显示选项"标签页下,默认设置"选择历史数据"为"自动"即 可。"历史记录"表示每次采集到的实际数据,"趋势"表示监控项每1小时数据统计后 的最大值、最小值、平均值,这一个小时的时间段是无法自定义的。

Z	ABBIX « 🔊	新仪表盘1		券 →添加 ∨ 保存设置 取消
	٩	添加仪表盘 / 美 添加构件	×	
0	监测 ^	Page 1 ····	图形 Y Show header V	
	仪表版	名利	监控服务器磁盘使用量	
	问题	胞新间磷	2.37	
	主机	590.0 GB		
	最新数据			
	的小的	569/8 GB		
ă	ET N 10C-N0	589.6 GB		
0	HR093 V	589.4 GB		
=	资产记录			
	报表 🗸	589.2 GB		
٩	配置 -	589.0 GB		
	管理	8-16 sum(Dataset #1)	22.13 8-16 22.20 8-16 22.27 8-16 22.34 8-16 22.41 8-16 22.48 8-16 22.55 8-16 23.02	
			数据设置 1 显示选项 时间期间 ● 坐标轴 图例 ● 问题 爱盖	
9	支持	选择历史数据	自动 历史记录 趋势	
Z	Integrations			
2	25.05		16 Atta	
	TE 203			
-	User settings v			
ڻ ا	暖出		Zabbax 6.0.6. © 2001–2022, Zabbax SIA	

5.配置时间范围: 在"时间周期"标签页下, 勾选"设置自定义时间段", 然后选择要显示的时间范围, 默认是最近1小时。



6.配置坐标轴:在"时间周期"标签页下,可设置坐标轴的最大值、最小值,一般默认 不做调整,由系统计算即可。



7.配置图例:在"图例"标签页下,可设置是否展示图例,以及设置图例的行数。



配置完成后,点击添加即可。

通过以上步骤,用户可以轻松地配置 Zabbix 的聚合图形,以便更好地展示多个监控项的数据。

以上就是这一期的 Zabbix 技术知识分享。大家好,我是乐乐,专注运维技术研究与分享, 关注我学习 Zabbix 等使用技巧,更多运维问题还可以到<mark>乐维社区</mark>留言提问哦~



八十九、Zabbix6.0 仪表盘配置简易网络拓扑教程

一、新建拓扑图

1. 点击新建按钮

ZABBIX « 1	拓扑图				(制建拓計型) 导入
٩					THE A
• 盛商 •		8R []			
仅表板			100 (m/s)		
শিল্প			702745 BRC302		
主机	26¢.	12		动作	
報告数据	Local network	680	058	雇性 构造函数	
narren Eishiette					显示已再动发现的 1中的1
⑦ 服务 →	0 选择 新出 / 開除				
≔ 资产记录					
11 IST -					
ペ 和田					
2 88					
Q 支持					
E2 Integrations					
? 1883					
💄 User settings 🔍 🗸					
し 調出			Zabby 6.0.6. @ 2001-2022 Zabby SIA		
			Laure V.V.V. W LVV F-LVLA, 20000 OCA		

2. 配置网络拓扑图信息(PS:所有字段后续可以修改调整,所以这边可以先随意配

)

置

网络拓扎	ŀ	온
------	---	---

* 所有者	Admin (Zabbix Administrator) 🗙	选择										
* 名称	简易网络拓扑											
*宽	800											
* 高	800											
背裏图片	没有图片 >											
自动的图标映射	<手册> ~ 展示图标映射											
图标高亮												
触发器状态上的标记组件改变												
显示问题	展开单一问题 问题数量 问题的数量和扩大最关键的一个											
高级标签												
拓扑元素标签类型	标签 ~											
拓扑元素标签位置	底部 ~											
问题显示	所有 ~											
最小的严重级别	未分类 信息 警告 一般严重 严重 灾难											
显示处理的问题												
URLs	名称 URL 组件 动作											
	75-111											

3. 添加完成后, "拓扑图"菜单中点击新建的拓扑名称进入拓扑页

然后点击右上角"编辑拓扑图"按钮,进入到拓扑图编辑界面

二、拓扑图绘制

1.添加已经监控的网络设备主机

网络拓扑图	N.																				
四元第 添加 !	修除 形: 汤	加/修修機	接:添加	1/修除用	研索: ᄎ	网格: 已	讀/ <u>开</u>	50x50	× X	齐拓扑元素	更新										
	100	150	2.00		300	350	-400	450			600 6	50 700		地图	元素	++0					
			1												· · · · · · · · · · · · · · · · · · ·	土11 核心交换	<u>،</u>				
		#feb48/#													标签位置	默认	~		-		
200															* 主机	华三交换 与/或(机S5500-28C (默认) 或	-si ×		选择	
250																标记 添加			包含 ~ 位	移除	
300														自动的	的图标选择						
															图标	默认	Switch_(64)	~		
400																19302 1930	(数以) (数以)		~		
																停用的	2 53,		×		
															座标 URLs	X 139 名称	Y 77		URL		动的
600						ur	配置	置,	可以	人选持	圣性福	配置	, 非	必须	į	设备端口 添加]最新数据		http://192.168.3.70/z/zabbic.php?name=&eval	type=0&tags%5B0%5E	%5Btag%5D: 18%
																应用	移除	关闭]		

2.如图操作添加多个设备后,设备间两两选择添加"链接"

a) 安装 ctrl 选择两台网络设备, 然后点击链路"添加"按钮



b) 编辑链路配置绑定触发器及指标标签
批量更新组件	
所选的组件	类型 名称 主机 华三SR8802-X-S 主机 华三交换机S5500-28C-SI
标签	E J
- 标签位置	t 默认 ~
自动的图标选择	
图标(默认)) Cloud_(24) ~
图标(问题)) 默认 ~
图标(维修)) 默认 ~
图标(停用的)) 默认 ~
链接	应用 移除 关闭 自从 到 链接指示器 动作 华三交换机S5500-28C-SI 华三SR8802-X-S 編編
链接	自从 到 链接指示器 动作 华三交换机S5500-28C-SI 华三SR8802-X-S 编辑 带宽:1000Mbps 上行速率:{?last(/S5500-28C-SI/ifHCOutOctetsPersecond[GigabitEthernet1/0/23])} 下行速率:{?last(/S5500-28C-SI/ifHCInOctetsPersecond[GigabitEthernet1/0/23])}
类型(OK) Color (OK)	援 V
链接指示器	触发器 类型 Color 动作 华三交换机S5500-28C-SI: [网络设备]华三交换机S5500-28C-SI设备端口GigabitEthernet1/0/23处于DOWN 援 く 校志 华三交换机S5500-28C-SI: [网络设备]华三交换机S5500-28C-SI设备端口GigabitEthernet1/0/23带宽发送利 用率大于90%
	华三交换机S5500-28C-SI: [网络设备]华三交换机S5500-28C-SI设备端口GigabitEthernet1/0/23带宽接收利 用率大于90% 添加
	应用 移除 关闭

标签使用监控项数据,格式参考 zabbix 官方手册

1 配置网络地图 (zabbix.com)

链接属性:

参数	说明
标签	将在链接顶部呈现的标签。 此字段支持奏达式 宏,但仅限于 avg、last、 min和 max 函数,以时间为参数(例如,{?avg(/host/key,1h)})。

3.更新拓扑图, 查看效果









三、配置首页仪表盘使用此拓扑图

1.创建新仪表盘

ZABBIX « 🔊	Zabbix server health		编辑仪表表 😑 🛠
٩	添加仪表金 Zabbk server health		く 線小 > 2022-12-19 10.08.37 - now (5)
• • •	Values processed per second	Utilization of data collectors 🔅 …	Utilization of internal processes
	1	17	1
主机。			
最新政策			
格計圖			
· 秦5∺3章			
11 报表 -	0 2022-12-21 2022-12-24 2022-12-26 2022-12-29 2023-1-01	0 2022-12-21 2022-12-24 2022-12-26 2022-12-29 2023-1-01	0 2022-12-21 2022-12-24 2022-12-26 2022-12-29 2023-1-01
4 RE -	Cache usage	Value cache effectiveness	Queue size
	1	1	1
9 支持	0 2022-12-21 2022-12-24 2022-12-26 2022-12-29 2023-1-01	0 2022-12-21 2022-12-24 2022-12-26 2022-12-29 2023-1.01	0 2022-12-21 2022-12-24 2022-12-26 2022-12-29 2023-1-01

仪表盘	创建仪来自 🎦
	ilian V
□ 名称▲	
Global view	My Shared
Zabbix server	My
Zabbix server health	My Shared
新议表面	My.
	显示已自动发现的 4中的4
0 335 800	

仪表板属性		
* 所有者	Admin (Zabbix Administrator) 🗙	选择
* 名称	网络拓扑图	
Default page display period	30秒 ~	
Start slideshow automatically		
		应用 取消

2.添加卡片组件,选择拓扑图

添加仪表盘 / 网络拓扑图	添加构件	×
Page 1 ····	美型 拓扑图 🗸	Show header 🔽
	名称 网络拓扑图	
	刷新间隔 默认 (15分) >	
Release to create a	源类型 拓扑图 拓扑图导航树	
widget.	* 拓扑图 简易网络拓扑	选择
		添加 取消

3.效果查看

ZABBIX « 🔊	拓扑图
۹.	所有地图 / 简易网络拓扑
● 监测 ^	
仪表板	
问题	
主机	
最新数据	
拓扑图	路自当口 正常
自动发现	
⑦服务 ∽	一行波・1000mbps 上行演率:33.26 Kbps 下行速率:104 bps
ः 當一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一	
□ 报表 ✓	No. of Contract of
🔧 配置 🗸	核心交换机 正常
2 管理 🗸	

更多……

第三方平台对接

九十、Zabbix 与 Prometheus 区别

Zabbix 和 Prometheus 都是非常流行的监控系统。它们有许多相似之处,但也有一些不同之处。以下是 Zabbix 和 Prometheus 监控对比的一些关键点:

1、数据模型和查询语言

Prometheus使用一个称为 PromQL 的查询语言来查询和处理时间序列数据。PromQL 支持许多数据模型和查询功能,包括度量标准、标签和聚合函数。Zabbix 使用自己的数据模型和查询语言,包括项、触发器和动作等概念。

2、存储方式

Prometheus 使用一种称为 TSDB 的时间序列数据库来存储时间序列数据。TSDB 使用 一种称为 WAL 的写前日志,以确保数据的可靠性。Zabbix 使用关系型数据库来存储数据。

3、自动化和配置管理

Prometheus 具有自动化和自动配置的能力,它可以自动发现服务和指标,并对它们进行监控。Zabbix 也提供了类似的功能,但需要手动配置。

4、可视化和警报

Zabbix 和 Prometheus 都支持可视化和警报功能。 Zabbix 提供了一个基于 Web 的前端界面,可以查看监控数据和设置警报。 Prometheus 通常与 Grafana 等工具一起使用,以实现更高级的可视化和警报功能。

5、性能和扩展性

Prometheus 在性能和扩展性方面表现良好,能够处理大规模的时间序列数据。Zabbix

也具有良好的性能和扩展性,但在大规模监控方面可能需要更多的资源和配置。

综上所述, Zabbix 和 Prometheus 都是出色的监控系统, 具有各自的优势和适用场景。 zabbix 更加适合用于 本地计算机 的监控, 而 Prometheus 更适合在现在流行的 云计 算 监控上使用。

大家好,我是乐乐,专注运维技术研究与分享,关注我,了解更多运维知识。如有问题 也可以到乐维社区 (https://forum.lwops.cn/) 留言提问,与运维技术爱好者共同交流心 得。

九十一、Zabbix 对接 Prometheus 实操——基于 Prometheus pattern 监控

概述

得益于对云原生和容器监控的友好支持,如今, Prometheus 监控受到越来越多企业的 青睐。然而,对于已经部署了 Zabbix 监控系统的企业,想要用 Prometheus 完全替换 Zabbbix,可能既无必要,短期也不现实。实际上,Zabbix 4.2 及后续版本增加了对 Prometheus 数据源的接入,已经能够实现用 Zabbix 来对接 Prometheus 监控。

Prometheus 通过 Exporters 组件来收集数据。Exporters 是一类数据采集组件的总称, 它负责从目标处搜集数据,并将其转化为 Prometheus 支持的格式,并且暴露出一个 HTTP API 地址,等待 Prometheus Server 拉取数据并进行数据处理。

Zabbix 对接 Prometheus 也是通过 HTTP 代理, 拉取 Exporters 提供的大量 Prometheus 指标数据, 然后通过内置的 Prometheus pattern 进行数据的处理和筛选, 从 而获取监控值。

本文将介绍如何使用 Zabbix 的 Prometheus pattern 项来对接 Prometheus 数据源。 以 Prometheus 官方提供的 Node exporter 采集器为例。

Node_exporter 部署

下载 Node_exporter 部署包

Node_exporter 部署包可以从 Prometheus 官网进行下载。

地址: https://prometheus.io/download/

node_exporter				
Exporter for machine metrics O prometheus/node_exporter				
1.5.0 / 2022-11-29 Release notes				
File name	OS	Arch	Size	SHA256 Checksum
node_exporter-1.5.0.darwin-amd64.tar.gz	darwin	amd64	4.58 MiB	aa87d2553f1b194od883826ce19ee2700f4fe6a0bce4592b4c2884a3d2dc7cbb
node_exporter-1.5.0.linux-amd64.tar.gz	linux	amd64	9.71 MiB	af999fd31ab54ed3a34b9f0b10c28e9acee9ef5ac5a5d5edfdde85437db7acbb

可以选择所需的版本,然后复制相关链接在服务器上面使用 wget 或 curl 进行下载,抑或

直接下载后再上传到服务器。

解压安装

下载好后,直接进行解压:

```
[root@localhost prometheus]# ls
node_exporter-1.5.0.linux-amd64.tar.gz
[root@localhost prometheus]# pwd
/data/test/prometheus]# _
```

编写 systemd 启动服务

vim /usr/lib/systemd/system/node exporter.service

将以下内容复制到文件中:

ILInit1 Description=node-exporter service After=network.target

IServicel User=nrometheus Groun=prometheus KillMode=control-group Restart=on-failure RestartSec=60 FxecStart=/data/test/prometheus/node_exporter-1.5.0.linux-amd64/node_exporte r \ --web listen-address=:9100 \ --collector systemd \ --collector systemd \ --collector systemd \ --collector processes \ --collector.tcpstat

[Install]

WantedBy=multi-user.target

注意, 启动命令的路径要根据实际的路径进行修改。

创建普通用户 prometheus

useradd prometheus

启动 Node_exporter 服务

systemctl daemon-reload

systemctl start node_exporter.service

systemctl status node_exporter.service

TrongElocalhost inde apprier 1.5.0.lium-and601e systemti status node_exporter.service = node opporter service: Loaded (/usr/lib/systemd/systemd/sole.exporter service) &titue: attive: attive (mming) since = 2023-02.22 14:505 (ST: 4 days ago Main (FD) 885 (mode_opporter) = Namory: 26.44 (Group: /system.sluconode_exporter.service (Group: /system.sluconode_exporter.service)
2/1 22 14:50:53 localbst.localdmain node_opporter[5345]: ts=203:30:22705/053.0022 callerande_opporter_go:117 level-info collector-themal_zone 2/1 22 14:50:53 localbst.localdmain node_opporter[5345]: ts=203:30:22705/053.0022 callerande_opporter_go:117 level-info collector-time 2/1 22 14:50:53 localbst.localdmain node_opporter[5345]: ts=203:30:22705/053.0022 callerande_opporter_go:117 level-info collector-time 2/1 22 14:50:53 localbst.localdmain node_opporter[5345]: ts=203:30:22705/053.0022 callerande_opporter_go:117 level-info collector-time 2/1 22 14:50:53 localbst.localdmain node_opporter[5345]: ts=203:30:2705/053.0022 callerande_opporter_go:117 level-info collector-time 2/1 22 14:50:53 localbst.localdmain node_opporter[5345]: ts=203:30:2705/053.0022 callerande_opporter.go:117 level-info collector-matca 2/1 22 14:50:53 localbst.localdmain node_opporter[5345]: ts=203:30:2705/053.0022 callerande_opporter.go:117 level-info collector-matca 2/1 22 14:50:53 localbst.localdmain node_opporter[5345]: ts=203:30:2705/053:0022 callerande_opporter.go:117 level-info collector-matca 2/1 22 14:50:53

访问测试

然后访问 IP:9100 端口

★ → C
 ▲ 不安全 192.168.3.141.9100/metrics
 # THEP \$c_sc_duration_second & summary of the pause duration of garbage collection cycles.
 # TFF \$c_sc_duration_second [summile="0] 1.8408-05
 \$c_sc_duration_second [summile="0] 1.8408-05
 \$c_sc_duration_second [summile="0] 1.7.5408-05
 \$c_sc_duration_second [summile="0.7], 7.8408-05
 \$c_sc_duration_second [summile="0.7], 7.8408-05
 \$c_sc_duration_second [summile="1.7, 5408-05
 \$c_sc_duration_second [summile="1.7, 5408-05

如图所示表示部署成功,并且成功采集到数据。

Node_exporter 相关指标说明

Node_exporter 是 Prometheus 用于监控服务器的采集器,其相关的采集指标说明可以参

考以下地址,然后按需进行监控:

参考指标地址: https://blog.csdn.net/qq_33326449/article/details/126663517

Zabbix 对接 Node_exporter

创建监控模板

*模版名称	Prometheus-Node_exporter	
可见的名称	Prometheus监控-Node_exporter	
模板	在此輸入搜索	选择
* <mark>群组</mark>	服务器模板 × 在此輸入搜索	选择
描述		

添加一个宏值:用以灵活监控多个主机

模板标记 宏1 值映射				
	模板宏继承模板的宏			
	宏	值	描述	
	{\$PROMETHEUS_URL}	http://URL	T → Prometheus指标接	口地址 移除
	添加			

创建 HTTP 代理监控项,获取大量 Prometheus 指标

首先使用 HTTP 代理监控项, 获取 Prometheus 的指标数据。

所有模板 / Prometheus监控-Node_exporter 出	拉拉项 触发器 图形 仪表盘 自动发现规则 Web 场景	
监控项 标记 进程		类型选择HTTP代理
* 名称	Prometheus指标采集-{\$PROMETHEUS_URL}	
类型	HTTP 代理 V	目定义一个键值
	prometheus.node_export.metrics	信自悉刑选择文木
信息类型	文本 · ·	旧心夫主题并文本
	(SPROMETHEUS_URL)	N 白完V 的URI 宏值
重调子版		
	[11] 添加	1947 Febr
请求类型	GET Y	
* 超时	15s	请求类型使用get, 超时
请求 body 类型	原始数据 JSON 数据 XML 数据	给个15s
请求正文		
		2
쐇		
	····· ···· ···· ···· ···· ···· ···· ·	移除
	添加	
要求的状态码	200	
民間通知比率专	v	
恢复模式	内容 头 body 和 headers	

创建相关项目监控项,使用 Prometheus pattern 获取监控值

监控项 标记 进程 1 * 名称 服务器总内存 类型为相关项目 类型 相关项目 键值可从指标说明文档查 node_memory_MemTotal_bytes 键值 类型为浮点数 找 eus指标采集-{\$PROMETHEUS_U... × 洗择 * 主要项 单位 B 单位输入为B_{历史数据保留时长} Do not keep history Storage period 90d 主要项为前面添加的 *趋势存储时间 Do not keep trends Storage period 365d HTTP代理率 值映射 在此输入搜索 插入主机溶产纪录栏位 -无v

这里以监控服务器的内存大小为例:

然后添加处理步骤:



添加监控主机

然后添加主机进行测试

* 主机名称	192.168.3.141				
可见的名称	192.168.3.141-node-exporter				
模板	Prometheus监控-Node_exporter × 在此输入搜索		选择	添加主	机,关联刚创建
* 群组	Linux 操作系统 × 在此输入搜索		选择	的榠奴	
Interfaces	类型 IP地址	DNS名称	连接到	四被	默认
	客户端 127.0.0.1		IP	DNS 10050	 參
	添加				
描述					
			le		
ent代理程序监测	(无agent代理程序) V				

修改宏值:

	×	
主机 IPMI 标记 发 预计记录 加速 值录明 主机宏 继承以及主机宏		
幸 値 [\$PROMETHEUS_URL} http://192.168.3.141;9100/m	描述 retrics T > description 發驗	
		移
	现新 完整 一般 一般 一般 一般 一般 一般 一般 一	<u>1</u>

查看指标数据

过滤器只影响过滤后的数据		Save as	应用重设	
□ 主机	名称▲	最近检查记录	晶体物层	■2 に 已
192.168.3.141-node-exporter	Prometheus描标录集-{\$PROMETHEUS_URL}	42s	# HELP go_gc_du	ration_s
192.193.3.141-node-exporter 0.138年 国元中教育な法部 回 138年 国元中教育な法部	<u>服務構成改</u>	426	11.58 GB	# HELP pg_gdetailing_second. A summary of the pause duration of gattage collection cycles. # TYPE pg_gdetailing_second.commany pg_gdetailingsecond.guantiler*[0] 1.681 ho 5. pg_gdetailingsecond.guantiler*[0] 1.682 ho 5. pg_gdetailingsecond.guantiler*[0] 1.683 ho 5. pg_gdetailingsecond.guantiler*[0] 1.793 ho 5. pg_gdetailingsecond.guantiler*[0] 3.1275 ho 6. pg_gdetailingsecond.guantiler*[0] 3.1285 ho 6. pg_gdetailingsecond.guantiler*[1] 6.4128 ho 6. pg_gdetailingsecond.guantiler*[1] 6.4128 ho 6. pg_g_gdetailingsecond.guantiler*[1] 6.4128 ho 6. pg_g_getailingsecond.guantiler*[1] 6.4128 ho 6. pg_g_getailingsecond.guantiler*[1] 6.4128 ho 6. pg_g_getailing_second.guantiler*[1] 6.4128 ho 6. pg_ggtailing_second.guantiler*[1] 6.4128 ho 6. pg_ggtaili

如图所示,成功对接,并且获取到监控数据。

以上就是这一期的内容。大家好,我是乐乐,专注运维技术研究与分享,关注我,了解更多

运维知识。更多 Zabbix、Prometheus 等技术知识,还可关注<u>乐维社区</u>,如有问题也可以 到社区进行留言提问,与广大运维技术爱好者一起探讨。

九十二、开源监控 zabbix 对接可视化工具 grafana 教程



今天要给大家介绍的是开源监控工具 zabbix 对接可视化工具 grafana 问题。

有一定运维经验的小伙伴大抵都或多或少使用过、至少也听说过开源监控工具 zabbix, 更进一步的小伙伴可能知道 zabbix 在数据呈现方面有着明显的短板,因此需要搭配第三方 的可视化工具使用。

grafana 就是这样一款第三方可视化工具,它主要用于大规模指标数据的可视化展现, 是网络架构和应用分析中最流行的时序数据展示工具。grafana 官方库提供丰富的图表类型, 如热图、直方图、折线图和地图等,以适应不同的数据展示需求。下面我们直接进入 zabbix 对接 grafana 的操作:

一、对接前提

1.1 已经安装好 zabbix 且有监控数据

二、安装 grafana

2.1 安装 grafana

yum install -y https://dl.grafana.com/enterprise/release/grafana-enterprise-10.4.

1-1.x86_64.rpm

2.2 启动 grafana 服务

service grafana-server start

应用端口默认为 3000

用户名:admin

密码:admin

如果需要修改服务端口,修改/etc/grafana/grafana.ini 配置文件中的 http_port

;http_addr =

The http port to use
http_port = 3001

因本地服务端口冲突。本次实现修改 grafana 端口为 3001

用浏览器访问 http://ip/3001

6 G	rafana		× +							
← -	⇒ C	▲ 不安全	192.168.74.134:3001/	/login						
€ 4	🥖 Mysc	〕L监控_PMM	3、安装和使用PMM (C (81条满思) GPT4F	⑦ 想要免费、无需Ch	✤ GPT4free 免费使用	👲 Docker容器技术与	O Code search resul	中大型电脑单机游	ChatMindAl
							Welcome	to Grafan	a	
							VCICOINC		a	
						Ema	all or username			
						ad	Imin			
						Pas	sword			
									0	
								Log in		
								Forgot your pa	issword?	

成功登录 grafana 的首页

← → C ▲ 不安全 192.168.74.134	k3001/?orgId=1				
🕑 🍓 MySQL监控_PMM 🕅 安装行使用PM	IM (🧧 (81条演星) GPT4F 🚱 想要免謝	8. 无弯Ch 🛊 GPT4free 免费使用 🛓 Docker容器技术与 🔿 Code search re	sul 🎁 中大型电脑单机波 🌀 ChatMindAI-智慧	響 🔞 点击→注册→进 🚱 All ROMs ROMs	O Repository search
Ø		Q Search or jump to			+ ~ 💿 🔉
 A Home D Starred B Dashboards €xolore 	Welcome to Graf	ana		Need help? <u>Documentation</u> Tutorials	i <u>Community</u> Public Slack
A Alering @ Connections @ Administration	Basic The steps below will guide you to quickly finish sorting up your Grafana installation.	TUTORIAL DATA SOURCE AND DASHBOARDS Grafana fundamentals Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.	DATA SOURCES Add your first data source Learn how in the docs of	DASHEGARDS Create your first dashboard Learn how in the docs @	Remove this panel
	Dashboards Starred dashboards Recently viewed dashboards		Latest from the blog	4月 06 How ComplyAdvantage built a more strategy with OpenTelemetry and GF In the finitek word, companies are cont fraudsters and bad actors. ComplyAdva buinesses with compliance and risk ma one of the serve to uncoverbing money las	flexible observability afana Cloud stanty on the lookout for ntage, which provides nagement tools, knows that indering and other crimes is

2.3 汉化配置界面

打开参数设置界面

・ ・	- ^
② 参 MyGLEER PARAL ③ 安相の思想でMM () ④ 前原語音(GFUF)、③ 古茶品素、完全()、● GFR44+ 名意光素、金 Dasker@BEt#年。① Cade search mull、前 本式電磁形分晶、④ ChatkingAl-YEEL ④ All ROMA () ROML ① Grandware when line () FR44-EEL ④ All ROMA () ROML ① Grandware when line () FR44-EEL ④ All ROMA () ROML ① Grandware when line () FR44-EEL ④ All ROMA () ROML ① Grandware when line () FR44-EEL ④ All ROMA () ROML ① Grandware when line () FR44-EEL ④ All ROMA () ROML ① Grandware when line () FR44-EEL ④ All ROMA () ROML ① Grandware when line () FR44-EEL ④ All ROMA () ROML ① Grandware when line () FR44-EEL ④ All ROMA () ROML ① Grandware when line () FR44-EEL ④ All ROMA () ROML ① Grandware when line () FR44-EEL ④ All ROMA () ROML ① Grandware when line () FR44-EEL ④ All ROMA () ROML ① Grandware when line () FR44-EEL ④ All ROMA () ROML ① Grandware when line () FR44-EEL ④ All ROMA () ROML ③ Grandware when line () FR44-EEL ④ All ROMA () ROML ④ All ROML ④ All ROMA () ROML ④ All ROML ④ All ROML ● All RO	奥新 :
Image: Control of the state	39
E Home → admin > Profile admin → Home D Profile 0 + Home → dmin + Home	а 📀
Home U Profile Profile Overlage	
Name Change pass Ø Explore Name Ø Explore Name Ø Allering Email Image pass Image pass Image pass Image pass <td>history sword</td>	history sword
Interface those	

选择语言为中文(简体)

点击保存

🔞 Profile - admin - Grafana 🛛 🗙 🕂									~
← → C ▲ 不安全 192.168.74.134:3001/profile									N 18 A
● MySQL监控_PMM 以 安装和使用PMM (区 (81振得思) GPT4	F ③ 想要免费、无霜Ch	✿ GPT4free 免费使用	▲ Docker容器技术与	O Code search resul	前中大型电脑单机游	⑥ ChatMindAI-智慧	🕲 点击→注册→进	All ROMs ROMs	O Repository
0		Q Sear	ch or jump to		🖾 ctrl+k				+ ~
BE Dashboards Preferences									
Explore Interface theme									
> & Alerting Default									
> O Connections Home Dashboard									
O Administration Default dashboar									
Timezone									
Default									
Week start									
Default									
Language @Beta									
中文 (施体)			~) ~						
Save									
Organization	s								
Name	Role								
Main Org.	Admin								

三、对接 zabbix 数据

3.1 安装 zabbix 插件

grafana-cli plugins install alexanderzobnin-zabbix-app

重启 grafana-server 服务

systemctl restart grafana-server.service

C ▲ 不安全 192.168/24.134.3001/plugins/filterBy-installed&IIIIterBy-installed&IIIterBy-installed&IIIterBy-installed&III

上述步骤操作后,可再图片中对应的菜单位置看到 zabbix 插件已经安装

启用 zabbix 插件

	001/plugins/alexanderzobnin-zabbix-app				년 ☆ 🛛 🌲 👼
) 😺 MySQL监控_PMM 🔾 安徽和使用PMM	(🦲 (81条)は思) GPT4F 😵 想要免票、无能Ch 🔶 GPT	T4free 免審使用 🛓 Docker容器技术与	O Code search resul 前中大型	电枢单机语 🙆 ChatMindAI-智慧 🧔 点击 -	- IE → IE ③ All ROMs ROMs 〇 Repository search
<u>6</u>		Q 搜索或器转至	63 4	tri+k	at ⊙ ×+
■ 前页> 管理> 統件和政語> 統件 > 7					
 □ 当页 □ とい思想系 ○ とい思想系 ② 反素板 ● 原素 ● 新藤 	Zabbix Zabix piget for Grafina Gittuda Decs Lienese @ Overview © Version Pissory		Version From 4.4.6 Alexander Zobrin	Dependencies Signature S Gratara >+93.0 C Signed Levet: O C	oldens Late: Signed by Guidens Late:
 電気 転転 統計信息和許可 就計信息和許可 	Zabbix plugin for Grafana version (LLC) (door lot of a door later) () field except				
设置 田田 - 新作れ取38 新作	Protect Protect Prot	The DT C THE	nar ann grafan- zakhis zada shrig 13	12 The first sector of the	

3.2 创建 zabbix 数据源



6 Add data source - 助尿语 - 当 × +					- 0
← → C ▲ 不安全 192.168.74.1	:3001/connections/datasources/new			ළ ර	
● MySQL宣控_PMM 名 安装和使用P	M ((81 盐)) GPT4F 🚱 想要免票,无暇Ch 🛊 GPT4free 免察使用;	Docker容器技术与 Code search resul	第 中大型电脑单机游… 🥥 ChatMindAJ-智慧… 🕲	原曲→连册→进_	tory search
Ø	Q (880)	現時至			+ · O A
■ 首页 > 连接 > 数束液 > Add data					
山田 山田 山田 山田 ・ 公 三山田田県 - <	Add data source Choose a data source type C zabb Z žabbix © Signad			× Clear	+ Cancel
、 ② 福道					

如图片的位置,填写 zabbix 服务端 ip

5KTOP	-P7H 👖 20ms 🖽	20 × (+
o ale	xanderzobnin-zabbix-datas X	+
← →	C ▲ 不安全 192.16	74.134:3001/connections/datasources/edit/ddi0gkz26sjk0e
€ ø	MySQL监控_PMM 以 安装	更用PMM(🧧 (81条满意) GPT4F 📀 想要免费、无案Ch 🛊 GPT4free 免费使用 🛓 Docker容器技术与 🕥 Code search resul 購 中大型电脑单机游 🎯 ChatMind.
\$		Q 搜索或期转至 回 ctrl+k
	首页 > 连接 > 数据源 > ale	nderzobnin-zabbix-datasource
ය > ය > ස @ > අ) 首页 7 已加里标 引 仪表板) 探索 警报	C alexanderzobnin-zabbix-datasource Type: Zabbix H Settings B Dashboards
~ @	 连接 添加新连接 数据源 	Name 🔘 alexanderzobnin-zabbix-datasource Default 💽
› @	》 管理	Before you can use the Zabbix data source, you must configure it below or in the config file. For detailed instructions, view the documentation. Fields marked with * are required
		URL * O http://localhost/zabbik/api_jsonrpc.php Please enter a valid URL
		Authentication Authentication methods Choose an authentication method to access the data source No Authentication

填写 zabbix_server 的登陆账号和密码

点击保存测试,出现下面提示,说明已经连接成功

Enable Trends
after O
7d
Banna ()
4d
Direct DB Connection
Enable Direct DB Connection
Disable acknowledges for read-only users
Disable data alignment O
✓ Zabbix API version: 6.0.6
Next, you can start to visualize data by building a dashboard, or by querying data in the Explore view.
Delete Save & test

3.3 使用 grafana 官方的模版

选择导入模版

MySQL虚控_PMM	3) 安嶺和使用PMM (.	. 🦲 (81祭)高島) GPT4F	Ø 想要免费、无需Ch	♦ GPT4free 免费使用	🛕 Docker容器技术与	O Code search resul	中大型电脑单机道	○ ChatMindAI-智慧	. 🕲 点击→注册→进.	All ROMs ROMs.	O Repository	iearch »
				Q 疲素	或跳转至							0 a O
15頁 > 仪表板 > In											新建仪表板	
首页 已加层标 仪表板 编放列表 快照 序面板 公共仪表板 探索 容摄		Import dash Import dashboard fror	nboard n file or Grafana.com 上特化来板 脂肪等助地 Accepted file fy など意体用令人来の成用機用	L JSON X214 aturiatus aturiatus presu (son, tot senouses							母入议表 底 创建图积规则	
连接 管理		5583 ittoRife Joon H119A ("title": "Example "uid": "_OHnEON42 "panels": [] }	Repeating Dictionary va		25 9 8							

选择模版进行 5363 是一个 zabbix 的模版。输入模版 id, 点击加载

WySQL监控_PMM 以 安装和使用 PMM	PMM (🧧 (81条消息) GPT4F 🚱 想要免费、光器Ch 🎓 GPT4free 免费使用 🚖 Docker容疑技术与 🎧 Code search resul	第 中大型电视单机路 ③ ChatMindAI-智慧。	. 🗿 点击→注册→进 📀 All ROMs ROMs	. O Repositor
Ø	Q 理要成用结束	C ctri+k		+
■ 首页 > 仪表板 > Import dashboar				
○ 部内 □ ○ 日知思称 > 23 (24) (24) (24) (24) (24) (24) (24) (25) (24) (25) (24) (25) (24) (25) (24) (25) (24) (25) (24) (25) (24) (25) (25) (25) (24) (25) (25) (25)	Import dashboard Import dashboard from Rie or Grafana.com 			

选择配置好的 zabbix 的数据源。点击 import 按钮进行模版导入

=	1113		
	 88	首页 II 已加星标 (仪実版 満放列表 快照	Import dashboard Import dashboard from file or Grafana.com Importing dashboard from Grafana.com
			Published by PauloPaim
			Updated on 2023-04-13 12:49:08
		\$2:R	Options
		连接	Name
			Zabbix - Full Server Status
			Folder
			Dashboards v
			Unique identifier (UID) The unique identifier (UID) of a dashboard can be used for uniquely identify a dashboard between multiples Grafman Installs. The UID allows having consistent UIEs for accessing dashboards so changing the title of a dashboard will not break any bodymarked miss that dashboard.
			41URQF7mz Change úld
			2900x-1
			Z alexanderzobnin-zabbix-datasource v
			Import Cancel

查看导入的模版

🛜 Zabbix - Full Server Status - 6 🗙 🖬 配置监控项	× +								×	- 0	×
← → C ▲ 不安全 192.168.74.134:3001/d	d/41URQF7mz/zabbix-full-server-st	atus							6 4 1	🛛 😩 (🗷	新 :)
● MySQL监控_PMM 公 安排和使用PMM (🥝 (81张) GPT4F 📀 想要免费.	元驾Ch 會 GPT4free 先壽使用	▲ Docker容器技术相	O Code search resul	第一中大型电脑单机游…	○ ChatMindAl-智慧	Ø 点击→注册→进	All ROMs ROMs	O Repository s	earch	39
6		Q提案d	和时至								۰
) ištu - 🕬	 A Last 6 hours 			
<u>ක ස</u> තු 🖸											
→ ☆ 已加量核 ~ N	letwork										
→ 器 仪表版 (IN)) All							(OU	T) All		
編放列表											
快照											
岸海駅											
(i) 100					data					A	
 A 5期 											
> ② i脂酸											
~ @ 管理											
 一 概況 	roblems										
统计信息和许可 Pro	blems										
默认前选项											
699	High 💔 (操作系统] 监控系统-Server主采集服	务器 Agent 无法采集	到数据				1个月 06	3月 2024 19	:59 @	
組 织											
编开											

到这一步默认很多数据没有,这是因为导入的模版的 item 项和 zabbix 那边的无法对应,

Wysql监控_PMM. 身機和使用P	*MM (🤮 (81条)補思) GPT4F 📀 想要が	免算、无雨Ch 會 GPT4free 免費使用	★ Docker容器技术标 〇 Code search resul 第 中大型电脑单机游 🛞 ChatMindAl-智慧	. 🕲 #击→注册→进 🔇 All RC	DMs ROMs O Repository search
		Q 提家:	就被至 回 tt/i+k		
■ 首页 > 佼実板 > Zabbix - Full Ser				() (100 - 100 () () () () () () () () () () () () ()	
	Group 全部 - Host 部校表	Kild-Server -			
	~ Info				
~ 器 仪表板	Total memory	Lintime ①			Total CPUs
構成列表					
成田板			OFFLINE		No data
	<分享 回 p 1	O Last 10 years			Problems
					Provenis
	0 628 GI		No data		No data
	○更多				
* © 監護	8 BB 00 pr				
修计信用和许可	Memory Constitution	Mainbry J CPU			CPO utilization
			No data		
	N/A				N/A
> 用户和访问权限	CPU lowait time ③	Number of processes	Filesystem (Space utilization %)	Disk drives ③	Logged in Users ①

自行修改面板的监控项即可

修改仪表盘的监控项

		Q. 搜索或用缺至		
首页 > 仅表板 > Zabbix - Full Se	erver Status > Edit panel			Discard Save Ap
 首页 日加星际 	D Group 全部 ~ Host 监控系统-Server ~		Table view Fill Actual	Ma Stat
	Total nemory	84	%	Q. Search options All Override Panel dinks Panel links
	Query 1 3 Transform data 0			Repeat options
 组织 指件和政策 指件 用户和边问权限 	Data source Z alexanderzobnin-zabbio-datar • A (sitesanderzobnin-zabbio-datarource) Overry type Metrics •			Value options show Calculate a single value per column or series or rev Calculate All values Origination
	Group <u>此时能</u> 分器 H Rem tag R Functions I → Options count = true	iost <u>這地來</u> 統-Server em 物理均分析用率		Lastuation Choose a reducer function / calculation Last * Fields Behrict the fields that should be included in the p Numeric Fields

Wysqu监控_PMM 以 安徽和使用P	MM (📴 (81条法思) GPT4F 🚱 想要先蒙	、无務Ch 🐓 GPT4free 免費使用 🛓 Docker容器技术与	O Code search resul	③ ChatMindAl-智慧 ④ 点击→注册→进.	🔇 All ROMs ROMs 🔘 Repository search
G		Q 喪弊或用時至			
				🛱 🕲 🖾 🖉	💴 📀 Last 6 hours 🕤 🔍 💭 30s 🕤
	Group 全部 - Host 蓝拉美统 ~Info	-Server -			
- 88 totalex 相談列表	Total memory	Uptime 💿			Total CPUs
	84%		OFFLINE		No data
	and the second second	O Last 10 years			Problems
 	OFFLINE		No data		No data
- © 1233	~ Stats				
 ₩DR ADL/WINDOW FI 	Memory Utilization	Memory / CPU			CPU utilization
统计微微的许可					

完成对接 zabbix,已经可以正常获取数据。

四、grafana 仪表板配置

DESKTOP-P7H 👖 26ms 🔟 🔍 🗡	× +	
🙆 探索 · alexanderzobnin-zabbi 🗙 🖬 配置	≣ant x + +	- 0 ×
← → C ▲ 不安全 192.168.74.134: → のいいのに定た PMM の 安美和使用PMM	43001/explore/schemal/ersion=1&panere=%78191%JAA781datasource%3A4dd0g2266jde%2C queriet=%3A4%558/H3 "schema %3A12%2C queryfype%3A4 (%3C group)%3A4781@ther%3A7%3D92C h 🖗 🏚	Search
0	Q 現款回期的至	0 a 0
■ 首页 > 探索 ペ ~		
	1 2 Outline Z steranderzobnin-zabbix-datar - Zabbix可以查看到刚才的仪表盘 Ш #59 器 WatH92### ③ Last Thour ~ Q G	运行查询 •
・ い Children・ 品 仮表版		
播放列表 快艇	Clumy type Metrics Clump Hest	
市画板		
C RE	Functions + Options	
	+ 我心面有 5 面积历史记录 0 声响论面离	
● 電光 统计信息和许可 默认首选项		

自定义面板

DESKTOP-P7H 👖 22ms 🔟 🔘 🗡	< +					
🙃 🖗 🖫 - alexanderzobnin-zabbo 🗙 🛃 🖽	Easestő × +				~	- 0
← → C ▲ 不安全 192.168.74.134	3001/explore?schemaVersion=1&panes=%78"191"%3A%78"	datasource*%3A*ddi0gkz26sjk0e*%	62C*querles*%3A%5B%7B*schema*%3A12%2	C*queryType*%3A*0*%2C*group*%3A%78*filter*%3	A‴%7D%2C″h 🖻 ✿	0 🛓 (🕬
📀 😸 мубоц <u>аю</u> рмм 🕄 安莱和克用PMI	M (🔃 (81銀陽島) GPT4F 🕲 想要免責,无獨Ch 🛊 GPT4F	free 免费使用 🛓 Docker容器技术年	〇 Code search resul 第 中大型电脑单机游	ChatMindAl-智慧 (3) 点击→注册→进 ③ All 5	ROMs ROMs O Repository	r search
Ø		Q 表示成期转至	EB ctri ik		+ -	0 1
☰ 前页→探索 ≪ -					新建仪表板	
() 해외 (미)) 수 Phottag	🕾 Outline 🛛 🗾 alexanderzobnin-zabbix-datar 🗸			🛄 評分 器 淋加到仪表板 🥝)Last1hc 创建智规规则	នកតព
 → 田 (公本板) 						
減放利率 快感	Group Host					
库加板 公共仅25版						
@ F##	Options +					
、 ¹ 9 昭和 、 19 昭和						
 ● 管理 ● 備況 						
統计信息和許可 默认肖选项						

选择添加可视化

Q	Q. 搜索或期结至				
■ 首页 > 仪表版 > New dashboard					
→ 器 仪表板					
	通过添加可能	同化开始您的新仪表板			
	造绎一个数据器,然后用距表、统计信息和表格查告	認的数据以及将其可视化,或创建列率、Markde	own 和其他小的件。		
		+ 添加可视化			
			-1-		
	漆加一个库面板	导入仪者	反权		
	漆加与其他反素使共享的可能化。	从这种能, grafana.com	V号入仪录载。		
	+ 凍加库園板		家板		
→ 16件和政策					

选择要展示的指标

	QI	教派就转至 回 ctrl+k		
页) 仅表版) New das	hboard > Edit panel			Discard Save Apply
		Table view		Time series
已加型标	Nginx問题			
totak BK				C Search options
magky932				
LCH				 Panel options
APPENDIX				Title
2234023088				Nginx画板
				Description
2252				Description
				Transparent background
	0 15:30 15:45 16:00 16:15 16:30 16:45 17:00 17:15 1	7:30 17:45 18:00 18:15 18:30 18:45 19:00 19:15 19:30 19:45	20:00 20:15 20:30 20:45 21:00 21:15	
统计信息和许可	- 每些处理请求数			> Panel links
默认首选项				> Repeat options
	Query 1 53 Transform data 0 Q Alert 0			r Toolfin
	Data source Zalexanderzobnin-zabbix-datar - 0 - Our		Query inspector	Toolio made
				Single All Hidden
	A (alexanderzobnin-zabbl/r-datasource)		0 @ A .:	Here we
用户和访问权限	Query type Metrics ~			How close the cursor must be to a point to trigger t tooltip, in pixels
	Group 监控服务器 Host 监治	空系统-Nginx		
	item tag	対い現実な数		Max width

修改面板标题

ହ					Q 搜索或就转至	⊕ ctri+k
= 1	i页 > 仪表板	New dashboard >				
6						Save dashboard
83						Details Trile
						Nginxin a
۲						Uescripaon
• 0						Folder
. 0						Dashboards ~
			0 - 48912/84(44			
			B Query (§)			
3			Data sporce			
i în						
				EIDERS B		
					NO DE	

整理 by 乐维社区(https://forum.lwops.cn)

查看已存在的配置面板

DESKTOP-P7H II 20ms 🔟	© × +										
🔞 仮原板 - Grafana 🗙	ⓒ 0283 - Galera X 🛛 KEL1923 X +										
← → C ▲ 不安全 192.168	← → C ▲ 不完全 192.168.74.134.3001/dishboards ()										
🕃 🗶 MySQL##PMM 🕅 👳	週用PMM (📴 (81条)高思) GPT4F 🤡 想要免票,无器Ch 🌻 GPT4free 免费使用 🛓 Docker容器技术与 📿 (Lode search resul)) 무大型电脑单标识 🜖 ChatMindAl-智慧 🚯 点击 → 注册 → 进 📀 All ROMs.) ROMs 📿 Repository search 😕									
Ø	Q 搜索或原转至	📾 etri-k + - O a 🕻									
■ 構页 > 化浓板											
 ○ 前页 → ○ 已加農标 → ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○											
席副板 公共仪物板 ④ 野寮	◎ 加速局法用法 ~ 2 加量 16	D = 1988 -									
 ・ 品< 昭元 ・ ゆ ・ 価 ・ 価 	IS Nojnušije: IS Zabbir - Full Server Status	(and the second s									
修计(前面相称)가 하 取べ(前面) () (2) () () () () () () () () () () () () () (

修改后的效果图如下



至此, zabbix 对接 grafana 完成。



常见问题

1、Zabbix 部署过程可能遇到的问题

这几天忙了一个项目,为顾客自定义安装一个他们公司专有的功能和界面,一堆的工具包, php 包,客户没有这么专业弄的。所以我的老同事就让我,弄一个可以为客户一键部署的 zabbix 安装包。

1, zabbix 界面忘记密码的操作, root 用户进入数据库, 使用 zabbix 库 update users

set passwd='\$2y\$10\$IEpFhXA/cXywW4t9MHI9J.2kyI23m7WGceQcy2tN144weF.z/l

nie' where alias="Admin"

这样登录的密码就是 Etx@2019 了,当然这个只是参考,原理就是进数据库改密码。

2, mysql_config not found, configure: error: Please reinstall the mysql distribution

报错,这种的数据库问题,可以执行:

配置文件 --with-mysqli=/usr/local/mysql/bin/mysql_config

具体对于自己 mysql 安装路径

3, 安装 sqlsrv, pdo_sqlsrv 的 php 扩展的时候报错,可下面这样做:



整理 by 乐维社区(https://forum.lwops.cn)

[size=12.0000pt]4,还有就是安装完了以后,界面报这种错,状态码是 500 的,这个报错 我花了很多时间去解决的,以为是配置文件的问题或者是 php-fpm 的问题,搜索了很多方 法,最后是老同事说的授权的问题。。。。。。。

可以去到 html 文件夹下,对 php 页面相关的所有文件,执行 chown -R zabbix:zabbix */nginx/html,具体看自己的 nginx 安装位置。



An internal server error occurred.

5, 部署过程出现 mysql 未找到命令, mysqladmin, python3 未找到命令的错误, 这些常见的就是软链接的问题了, 你可以直接再脚本上加上 ln -s /usr/local/mysql/bin/mysql /usr/bin/mysql 类似的命令, 具体看自己的软件安装位置。

6, nginx 启动过程中出现:

Nginx 错误: [emerg] getpwnam("www") failed

这种情况是缺少了 www 用户,属于脚本的用户添加失败,这时也可以手动添加用户和用户 组,命令是:

/usr/sbin/groupadd -f www

/usr/sbin/useradd -g www www

或者,也可以直接修改 nginx 的 nginx.conf 文件里的用户为目前已创建的用户,如 zabbix。

7, 在安装 mysql 还是啥的时候, 出了以下的错误:

./boost/python/detail/wrap_python.hpp:50:23: fatal error: pyconfig.h: No such file or directory

compilation terminated.

"g++" -ftemplate-depth-128 -O3 -finline-functions -Wno-inline -Wall -march=i686

-pthread -fPIC -m32 -DBOOST_ALL_NO_LIB=1 -DBOOST_PYTHON_SOURCE

-DNDEBUG -I"." -I"/usr/include/python2.7" -c -o

"bin.v2/libs/python/build/gcc-5.4.0/release/threading-multi/object/function_doc_s

ignature.o" "libs/python/src/object/function_doc_signature.cpp"

...failed gcc.compile.c++

bin.v2/libs/python/build/gcc-5.4.0/release/threading-multi/object/function_doc_si gnature.o...

.. failed updating 58 targets...

...skipped 12 targets...

...updated 11810 targets...

起初看到这么一大堆东西的时候很烦恼,不知啥原因,找了百度很多条,才找到一条博客是 解决问题的

解决方案是: yum -y install python-dev

8,



Stack trace 类的报错

这类问题我也是找开发人员才可以知道怎么解决的。

这种报错可能就 php 某个模块或版本不对应的,这时要更新替换某个 php 文件了。

9, 使用 1G 内存 1 个核的 CPU 的腾讯云主机源码安装 mysql 时, 在编译 make 命令时遇

到错误。

如下:

c++: internal compiler error: Killed (program cc1plus)

Please submit a full bug report,

with preprocessed source if appropriate.

See <http://bugzilla.redhat.com/bugzilla> for instructions.

make[2]: *** [sql/CMakeFiles/sql.dir/item_geofunc.cc.o] Error 4

make[1]: *** [sql/CMakeFiles/sql.dir/all] Error 2

make: *** [all] Error 2

解决办法:

[size=12.0000pt]1、增加虚拟内存,也就是增加 swap。

[size=12.0000pt]2、或者把跑的线程核数降低,可修改为跟主机一样 cpu 核数

[size=12.0000pt]10,还有一些就是 shell 脚本编写的问题了,很多细节需要注意,多练脚本就可以减少出错了。修改一些配置文件的时候,可以使用 sed -i 的命令,来批量修改文件中某部分的内容了,非常省时

如:

sed -i 就是直接对文本文件进行操作的。

sed -i 's/原字符串/新字符串/' /home/1.txt

sed -i 's/原字符串/新字符串/g' /home/1.txt

2、Zabbix 定制-MIB 库与 MIB Browser 用法

对于一个 Zabbix 监控玩的很溜小伙伴来说,MIB 这个词肯定有所耳闻。本文主要讲解 MIB的含义、作用以及 MIB 的查看方法。

先看下百度百科的说法:



个人理解:含义总的来说 MIB 就是一个有结构的数据库,里面存储了一些 OID 数据记录,OID (对象标识符)。一般各大硬件、服务器厂商都会维护有自己产品专属的 MIB 库,里面存储了对应系列、型号硬件的监控指标,例如风扇、电源等相关部件的信息。MIB 库它实际上就是一个.MIB 或者.txt 结尾的文件,如下图:

名称 ^	修改日期	类型	大小
STORAGE-MIB.mib	2020/11/6 16:55	MIB 文件	51 KB

上述图中的文件其实就是一个宏杉 MS3000 G2 型号存储设备 MIB 库文件,里面记 录了该型号设备的一些硬件信息,如果我们想通过 Zabbix 或者其他类似工具的 SNMP 协 议监控该存储设备,那就要对 MIB 库内容进行一个分析,这个地方我就为大家推荐一个好 用的 MIB 库查看工具 MIB Browser,这是一款收费工具,有 30 天的使用期限,不过可以 通过重复安装的方式实现永久使用,下载链

接 http://www.ireasoning.com/download.shtml 如下图:

$\leftrightarrow \ \rightarrow$	C (▲ 不安全 www.ireasoning.com/download.shtml	■ ☆ (世話待) : · · · · · · · · · · · · · · · · · · ·							
Software Download									
We offer 30-day trial versions of every product, so you can test-drive the software before you purchase it. Every trial version is completely free, fully functional, and runs for 30 days. We also offer support for your download versions. Test-drive one today!									
	MIB browser								
	MIB Browser Free Personal Edition	* DOWNLOAD							
	MIB Browser Professional Edition 30-day trial edition 4 MACOS 4 LINUX 64-BIT 4 OT	THER PLATFORMS							
	MIB Browser Enterprise Edition ± WINDOWS ± LINUX 64-BIT	THER PLATFORMS							

使用教程:

1.安装完成后打开 MIB Browser,如下图:

🚳 iReasoning MIB Browser			– o ×
File Edit Operations Tools Bookmarks Help			
Address: • 172.10.128.11 Advanced 0ID: .1.3		Operations: Get Next	~ 🜈 Go
SIMP MIBs	Result Table		
🕈 MIB Tree	Hame/OID	Value Tyr	pe IP:Port 👩
			ਣ
			Γ
			ā
			1
			1
			a
Nane	^		
MIB			
Syntax			
Access			
DefVal			
Indexes			
h			

2.点击左上角 File - Load MIBs 选取刚才的 MIB 库文件,加载 MIB 信息,如下图:

🚳 iReasoning MIB Browser										100	0	\times
File Edit Operations Tool	ls Bookmarks I	Help										
Adiress: • 172.10.128.11	1								×		~	Go
SIMP M.Bs	查找范围(I):	宏杉ms3000	0\$2	~ 0	🎓 📂 🛄 -							
T an In	快速防河 桌面 原 定示在 度 此电脑 砂塔	名称 〕STORAGE 填同上的文件、文	~ -MIB.mib 2件夹、程序快速方式和具f	也项目。	橡改日期 2020/11/6 16:55	樊型 MIB文件	大小 51 KB			pe	IP:Port	
Name	-											
MIB												
Syntax												
Access		文件名(N):						~	打开(0)			
Status		☆/計##用(7):	111 W11 (r. a)						Tracili			
DefVal		XH×201:	ALL FILES (#. #)					~	405月			
Indexes												
D										-	_	

3.MIB 库加载成功后,可以看到 MIB Tree 可以展开了,如下图:

🚳 iReasoning MIB Browser			– 6 ×						
File Edit Operations Tools Bookmarks Help									
Address: • 172.10.128.11 Advanced 0ID: .1.3		✓ Operations: Get Next	~ 🔗 Go						
SIMP MIBs	Result Table								
RE Tree B- iso. org. død. internet. private. enterprises. macrosum. macrosumStorage	Heat/012	Value 177	12:Part 😜						
Nene All All All All All All All All All Al									
Syntax Access Status Deff'al									
Indexes v									

4.展开后可以看到宏杉 MS3000 G2 存储的 MIB 库按照 SNMP、SNMP Trap 协议分为两

大块。

5.一级一级展开,可以看到里面记录下该型号存储的磁盘、风扇、Lun、FC等信息,如下图:

🚳 iReasoning MIB Browser	- 5 ×
File Edit Operations Tools Bookmarks Help	
Address: • 172.10.128.11 · Advanced 01D: .1.3	✓ Operations: Get Next ✓ 🜈 Go
SIMP MIBs Result Table	
In Tree Name/OlD In sour gefod internet private enterprises marcosuntersent Name/OlD In sour gefor internet private enterprises marcosuntersent Name/OlD Internet private enterprises marcosuntersent	Value Type IP:Port
Jiane A 01D M1B Syntax Access	
Stetus Defial Indexes v	
Decen	

6.这个地方我们随便点开一个 OID, 例如风扇模块的信息,就可以看到该存储风扇的具体情

况,结合监控工具和 SNMP 就可以监控上述设备的一些指标,如下图

🚳 iReasonin	g MIB Browser								- 0	×
File Edit	Operations Tools Bookmark	ks Help								
Address:	▼ 172. 10. 128. 11 ∨	Advanced	0ID: .1.3.6.1.4.	. 1. 35904. 1. 2 <mark>.</mark> 2. 3. 2. 1. 2			∨ Opez	ations: Get Next	~ @	Go
SIMP MIBs				Result Table						
	toragoSystem ansStoragaDeviceInfo asStoragaDeviceInfo asStoragaDeviceInfo fasState spuldState fasState fasState fasState fasState fasState fasState fasState fasState fasState presMalling fasState fasState fasState presMalling fasStatefasStatefasStatefasStatefasStatefasStatefasStatefasStatefasStatefasStatefasStatefasStatefasStatefasSta		~ ~ ~		Jane-/010		Value	Тура	IP:Port	
GID MIE Syntax Access Status LefVal	<pre>tantodState .1.3.6.1.4.1.35904.1.2.2.3.2. MACROSAL-STORAGE-MIE INTEGER (020) read-only mandatory</pre>	1.2								
Indexes	fanModIndex		~							
. iso. org. dod	. internet. private. enterprises. m	acrosan nacros	anStorage.msStorag	geSystem.msStorageStatu	s. fanState. fanModStatTable. i	anModStatEntry.fanModState				

3、如何使用 Zabbix 进行 IPMI 监控?

使用 Zabbix 进行 IPMI 监控所需要包含的步骤,包括安装相关软件包,配置 Zabbix,获取 传感信息,定义模板和配置 IPMI 用户密码等

1. 安装 IPMItool 软件包

yum -y install OpenIPMI OpenIPMI-devel ipmitoolfreeipmi

2. 配置 Zabbix

服务器端配置 zabbix IPMI pollers

cd /usr/local/zabbix/etc/

sed -i '/# StartIPMIPollers=0/aStartIPMIPollers=5'zabbix_server.conf

service zabbix-server restart

3. 获取传感器信息

登录 Zabbix 服务器,通过 ipmitool 远程访问服务器传感器信息
[root@localhost	shell]#	ipmitool -I lanplus	-H 192.1	68.1.	-U admin -P	sens	sor list		
UID Light	0x0	discrete	0x0080	na	na	na	na	na	na
Int. Health LED	0x0	discrete	0x0080	na	na	na	na	na	na
Ext. Health LED	0x0	discrete	0x0080	na	na	na	na	na	na
Power Supply 1	35	Watts	ok	na	na	na	na	na	na
Power Supply 2	95	Watts	ok	na	na	na	na	na	na
Power Supply 3	150	Watts	ok	na	na	na	na	na	na
Power Supply 4	110	Watts	ok	na	na	na	na	na	na
Power Supplies	0x0	discrete	0x0180	na	na	na	na	na	na
Fan 1	50.56	i8 percent	ok	na	na	na	na	na	na
Fan 2	50.56	i8 percent	ok	na	na	na	na	na	na
Fan 3	50.56	8 percent	ok	na	na	na	na	na	na
Fan 4	50.56	8 percent	ok	na	na	na	na	na	na
Fans	0	percent	ok	na	na	na	na	na	na
Temp 1	25.00	0 degrees C	ok	na	na	na	na	42.000	46.000
Temp 2	68.00	0 degrees C	ok	na	na	na	na	126.000	127.000
Temp 3	40.00	0 degrees C	ok	na	na	na	na	126.000	127.000
Temp 4	40.00	0 degrees C	ok	na	na	na	na	126.000	127.000
Temp 5	40.00	0 degrees C	ok	na	na	na	na	126.000	127.000
Temp 6	34.00	0 degrees C	ok	na	na	na	na	87.000	92.000
Temp 7	na		na	na	na	na	na	87.000	92.000 - 44:0-
Temp 8	34.00	0 degrees C	ok	na	na	na	na	87.000	92.000 开维化场

ipmitool -I lanplus -H 12.168.1.10 -U admin -P password-L user sensor list

ipmitool -I lanplus -H 12.168.1.10 -U admin -P password-L user sensor get "

Power Supply 1"

root@localhost Locating sensor record... Sensor ID : Power Supply 1 (0x4) 填写的key值 Sensor Type (Discrete): Power Supply Sensor Reading : 35 Watts 获取到的数据 States Asserted : Power Supply [Presence detected] 乐维论坛

4. 定义模板

创建一个空模板

・ 分 乐维监控											91lewei -
88. 首页	首页 /	系统管理	/ 模版管理								
4. 告警	全部	β	x .	请输入名称			٩			(1	运 看所有主机
「「「」」」」と注注していた。	_		415-				_				
<u>打</u> 视图 >									la l	第三步 新增	批量删除
■ 统计报表 >		序号	名称	应用集	监控项	触发器	图形	自动发现	链接的模板	已连接到	操作
 系統管理第一步 ▶ 自动发现规则 	•	1	HP Server Health Template	应用集 6	监控项 9	触发器 3	图形	自动发现 3	Template ICMP Ping		â
#11 模板管理 第二步		2	IBM DSA模板[通用]	应用集 1	监控项 16	触发器 16	图形	自动发现			â
 通知配置 维护模式 		3	HP IPMI模板[通用]	应用集 8	监控项 80	触发器 6	图形 1	自动发现			â
▲ 用户管理		4	Linux磁盘IO被动模板	应用集 2	监控项	触发器	图形	自动发现 1			â
▲ 权限配置	•	5	Linux磁盘IO主动模板	应用集 2	监控项	触发器	图形	自动发现 1			â
系统配置 >		6	中间件zookeeper模板 [通用]	应用集 4	监控项 31	触发器 5	图形 1	自动发现			Ê
		7	LW Template ZJ SNMP Device	应用集 4	监控项 6	触发器 3	图形	自动发现 1	LW_Template SNMP Interfaces / LW_Template ZJ SNMP		1
68.1.102/template/creat	e								Generic		赤细亚亚



模板建好之后,我们来添加一个 key

监控项													enze:	监控项
所有模板 / DELL-IF	PMI-模板 应用	集 监控项 触发	支器 图形	聚合图形 自动发现	現规则	Web 场景 过滤器 ▲								
Ŧ	机群组 在此输	入搜索	选择	类型	所有	•	信息类型	所有	Ŧ	状态	所有	¥		
	主机 DELL-	IPMI-模板 ×	选择	数据更新间隔(秒)			历史记录(天)			状态	所有	Ŧ		
3	应用集		选择				趋势(日)			触发器	所有	۲		
	名称]							模板	所有	23	,	
	键值		37											
						应用 重设								
过滤器只影响过滤	想后的数据													
Wizard	名称▲	触发	188	键值	间隔	历史记录	趋势	8	类型	应	用集		状态	
						未发现数据								
												显示已	自动发现的(0中的0
0选择 启用	禁用 清除	历史复制	批量更新	删除										
													乐省	论远

我们先确定 key,我们通过 sensor 的方式进行监控,在命令行执行

获取 Power Supply 1 的具体信息



之后,我们编写 Key,有空格的地方,我们使用_下划线连接,注意 IPMI sensor (IPMI 传

感器)不要加下划线,其他的就跟配置 agent 的方式一样就可以了。



添加完成。

接着,添加一条乐维监控(用于前端显示)

监控项	
所有模板 / Dell IPMI模板[通用] 应用集 4 监控项 10 触发器	10 图形 1 聚合图形 自动发现规则 Web 场景
名称	LW_HARDWARE_DELL 固定格式为 "LIT_(类型)_(产品型号)"
类型	外部检查 ▼
键值	LW_HARDWARE_DELL 同上 选择
信息类型	数字 (无正负) ▼
数据类型	+进位数字 ▼
单位	
使用自定义倍数	
数据更新间隔(秒)	30
白定义时间间隔	米型 间隔 即间 动作
	実主 Polyme Polyme
	添加
历史数据保留时长 (单位大)	90
趋势数据存储周期(单位天)	365
储存值	不变
查看值	不变 ▼ 展示值映射
新的应用集	
应用集	-无- Electric current / Voltage Fans Sensors Power Supply Temperature Sensors
填入主机资产纪录栏位	- . ▼
描述	{"agent":1}
已启用	
	更新 克隆 删除 取消 振维论损

模板制作完成。

新增监控,过程略。

5. 配置 IPMI 用户密码

系统管理—模板管理

找到新增模板 "DELL-IPMI-模板" ——点击已连接到的主机

主机					
所有主机 / DELL_VNX5300	已启用 ZBX SNMP JMX IPMI	应用集 7 监控项 127	触发器 3 图形	自动发现规则 5 Web 均	汤景
主机模板 IPMI 宏 音	主机资产记录 加密				
	认证算法	默认 无 MD2 MD5 Straight OEM RMCP+			
	优先权层级	回调 用户 操作者 管理者 OEM			
	用户名称				
	密码	更新 克隆 :	全克隆 删除 [取消	
					为家道部的方案

点击 IPMI,对优先权层级进行选择,填写用户名称和密码,点击更新。

主机配置完成。

稍等片刻,监控就会出现数据了。





4、Zabbix alerter processes more than 75% busy 告 警处理

一、告警现象

告警主机:	172.18.20.15
告警时间:	2021.03.08 02:23:43
告警等级:	Average
告警信息:	Zabbix alerter processes more than 75% busy
告警项目:	zabbix[process,alerter,avg,busy]
问题详情:	Zabbix busy alerter processes, in %:100 %
当前状态:	PROBLEM:100 %
事件 ID:	408627698

二、排查思路

1. 告警分析

告警显示的 zabbix 内部 alerter process 超过 75%, 已经达到 100%。 alerter process 是用于在触发告警时发送通知的进程,当产生大量的告警时, 该进程繁忙就会告 警。 但结合我们自己的情况, 每次出现告警时, 伴随的其他告警并多。 所以先是查询了 一番,有如下的回复,另外根据以下的回复, 并没有看出异常。

(1) Zabbix 的后端数据库卡了

- (2) Zabbix 服务器的 IO 卡顿
- (3) Zabbix 进程分配到内存不足

2. 进一步在查询, 看到可以通过修改 StartAlerters 的数量来调整, 紧接在

zabbix_server.conf 中修改。

Option: StartAlerters
Number of pre-forked instances of alerters.
Alerters send the notifications created by action operations.
#
Mandatory: no
Range: 0-100
Default:
StartAlerters=10

3. 然而在 zabbix_server.conf 的配置文件中,并没有看到 StartAlerters 这个参数,于 是手动加了以下配置,并重启 zabbix 服务,重启之后发现 zabbix 服务无法启动。

StartAlerters = 10

4. 在 zabbix 上的官网上看到,只有 zabbix 4.0 以上的版本才支持修改 StartAlerters 的数量,当前使用的是 3.0 的版本

 到这一步,已经没有招了,难道要准备升级版本 ? 这动作太大了,而且没有找到根本 原因,根据以往对网络设备升级经验来看,升级版本很多时候并没有用,只是作为尝试 的方法。所以升级版本暂时放弃,那到底是啥原因呢.....

6. 重新整理了一下告警思路, 下图可做参考



整理 by 乐维社区(https://forum.lwops.cn)

7. 周末在家继续排查,想起最近没有收到 zabbix 每天早上发送的定时邮件,所以判断是因为在出现告警时,大量的邮件无法发出,导致 alerter process 高。那么接下来就是查看发送邮件的脚本。

通过查看发送邮件的脚本,才知道之前更换了告警邮箱,原邮箱支持免密登录,新的邮
 箱不支持免密登录,开启使用用户名和密码登录后,可以正常发邮件。

smtp = smtplib.SMTP()

smtp.connect(mail_host)

smtp.login(mail_user, mail_pwd)

9. 观察两天, 未收到告警, 判断修复

5、更多.....

监控公开模板 (lwjk_template) 下载

乐维监控公开模板是乐维制作与推出的、面向市面主流厂商及型号设备的开源的监控模板 (含部分脚本),任何个人、团队、或企业,都可以免费使用对应模板实现基础设施监控纳 管,异常告警等。



监控模板下载

本文档为样章,完整版文档请添加乐乐 (lerwee)获取

