

# 一 环境介绍

zabbix\_server 版本:

```
root@iZuf61nha4v6rymngnvx79Z:/etc/zabbix/zabbix_agentd.d# zabbix_server -V
zabbix_server (Zabbix) 4.0.0alpha8
Revision 81985 18 June 2018, compilation time: Jun 18 2018 09:10:12
```

zabbix\_agent 版本:

```
root@iZuf61nha4v6rymngnvx79Z:/etc/zabbix/zabbix_agentd.d# zabbix_agentd -V
zabbix_agentd (daemon) (Zabbix) 4.0.0alpha8
Revision 81985 18 June 2018, compilation time: Jun 18 2018 09:10:12
```

OS 版本:

```
root@iZuf61nha4v6rymngnvx79Z:~# uname -a
Linux iZuf61nha4v6rymngnvx79Z 4.4.0-79-generic #100~14.04.1-Ubuntu SMP Fri May 19 18:36:51 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
```

mysql 版本:

```
mysql> select version();
+-----+
| version() |
+-----+
| 5.7.11    |
+-----+
1 row in set
```

## 二 环境配置

### 2.1 zabbix\_agent 自带配置

zabbix\_agentd 服务器中，在 /etc/zabbix/zabbix\_agentd.d/userparameter\_mysql.conf 文件中，定义了以下四个监控项

mysql.status[\*]

mysql.size[\*]

mysql.ping

mysql.version

### 2.2 zabbix\_server 自带配置

zabbix\_server 中自带模板 Template DB MySQL 中，有以下 14 个监控项，对应的是 2.1 中的三个 key，mysql.size[\*]后续我们自己添加。

名称 ▲	触发器	键值
MySQL begin operations per second		mysql.status[Com_begin]
MySQL bytes received per second		mysql.status[Bytes_received]
MySQL bytes sent per second		mysql.status[Bytes_sent]
MySQL commit operations per second		mysql.status[Com_commit]
MySQL delete operations per second		mysql.status[Com_delete]
MySQL insert operations per second		mysql.status[Com_insert]
MySQL queries per second		mysql.status[Questions]
MySQL rollback operations per second		mysql.status[Com_rollback]
MySQL select operations per second		mysql.status[Com_select]
MySQL slow queries		mysql.status[Slow_queries]
MySQL status	触发器 1	mysql.ping
MySQL update operations per second		mysql.status[Com_update]
MySQL uptime		mysql.status[Uptime]
MySQL version		mysql.version

还有一个触发器，2 个图形，一个聚合图形。

所有模板 / Template DB MySQL 应用集 1 监控项 14 触发器 1 图形 2 聚合图形 1 自动发现规则 Web 场景	
<input type="checkbox"/> 名称 ▲	宽
<input type="checkbox"/> MySQL bandwidth	900
<input type="checkbox"/> MySQL operations	900

## 2.3 新增配置文件

在 zabbix-agent 中，新建 my.cnf 配置文件(注意是隐藏文件)，此文件是为了 zabbix 执行 sql 时使用配置文件传入 IP、账号、密码等，一是方便，二是避免直接 `mysql -u** -p***` 时系统出现的警告。

文件内容如下：

```
vim /etc/zabbix/zabbix_agentd.d/.my.cnf
```

```
[client]
```

```
user=zabbix
```

```
host=127.0.0.1
```

```
password=zabbix
```

实际使用时要修改成自己的账号密码，账号权限一定要够，否则只能查到自己权限内的信息。

另外，此文件也可以建在其他路径，不过要下文增加 key 时，HOME 指向的路径一致，并保证 zabbix 用户可读。

## 2.4 修改配置文件

修改默认的配置文

```
cd /etc/zabbix/zabbix_agentd.d
```

```
sed -i 's#HOME=/var/lib/zabbix#HOME=/etc/zabbix/zabbix_agentd.d#g' userparameter_mysql.conf
```

其实就是指定 my.cnf 的路径，这个路径就是 2.3 中 my.cnf 的路径。

## 2.5 新增监控模板

新增模板如下

模板

所有模板 / 6 mysql监控 应用集 1 监控项 14 触发器 1 图形 3 聚合图形 自动发现规则 Web 场景

模板 链接的模板 宏

\* 模版名称 mysql\_monitor

可见的名称 6 mysql监控

\* 群组 Templates x  
在此输入搜索

描述

并设置链接模板 Template DB MySQL:

所有模板 / 6 mysql监控 应用集 1 监控项 14 触发器 1 图形 3 聚合图形 自动发现规则 Web 场景

模板 链接的模板 宏

名称	动作
Template DB MySQL	<a href="#">取消链接</a>

链接指示器  
在此输入搜索 选择

[添加](#)

[更新](#) [克隆](#) [全克隆](#) [删除](#) [删除并清除](#) [取消](#)

链接完成后，取消链接，这一步很重要，否则没法进行下面的操作，当然你也可以直接修改原有模板，不建议这样做，进行先保留原有模板信息。

应用集 1 监控项 14 触发器 1 图形 2 聚合图形 自动发现规则 Web 场景

模板 宏

名称	动作
Template DB MySQL	<a href="#">取消链接</a> <a href="#">取消链接并清理</a>

链接指示器

这样就继承了 Template DB MySQL 模板的监控项、图形、触发器等。

- 1) 修改监控项、图形名称为中文，以利于观察
- 2) 取消 mysql.version 监控项，这项不需要
- 3) 添加 mysql.size[all,all,both]监控项，后面会介绍

名称 ▲	触发器	键值
MySQL删除语句条数/分钟		mysql.status[Com_delete]
MySQL 占用总磁盘大小		<u>mysql.size[all,all,both]</u>
MySQL发送的字节数/秒		mysql.status[Bytes_sent]
MySQL 回滚语句条数/分钟		mysql.status[Com_rollback]
MySQL慢查询语句条数/分钟		mysql.status[Slow_queries]
MySQL执行begin的语句条数/分钟		mysql.status[Com_begin]
MySQL执行commit语句条数/分钟		mysql.status[Com_commit]
MySQL执行的语句总条数/分钟		mysql.status[Questions]
MySQL接收的字节数/秒		mysql.status[Bytes_received]
MySQL插入语句条数/分钟		mysql.status[Com_insert]
MySQL更新语句条数/分钟		mysql.status[Com_update]
MySQL服务可用性	触发器 1	mysql.ping
MySQL服务已经运行时长(小时)		mysql.status[Uptime]
MySQL查询语句条数/分钟		mysql.status[Com_select]

## 三 启用默认监控项

本章节，只是介绍默认监控项的获取方式等，不牵涉到修改配置。

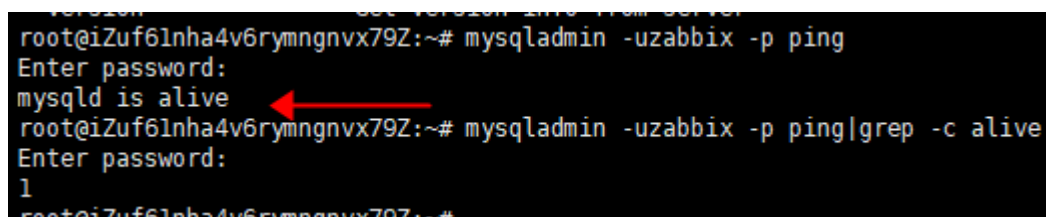
### 3.1. 启用 mysql.ping

不需要修改，使用默认配置即可，下面是对此 key 的理解：

```
cat /etc/zabbix/zabbix_agentd.d/userparameter_mysql.conf
```

```
UserParameter=mysql.ping,HOME=/etc/zabbix/zabbix_agentd.d mysqladmin ping | grep -c  
alive
```

这个 key 其实就是使用 mysqladmin 命令监控服务可用性，如下



```
root@iZuf6lnha4v6rymngnvx79Z:~# mysqladmin -uzabbix -p ping  
Enter password:  
mysqld is alive  
root@iZuf6lnha4v6rymngnvx79Z:~# mysqladmin -uzabbix -p ping|grep -c alive  
Enter password:  
1  
root@iZuf6lnha4v6rymngnvx79Z:~#
```

返回 1 代表 mysql 存活，返回 0 表示服务未存活，所以在模板中会有一个触发器，当返回不是 1 时，进行告警。

### 3.2 启用 mysql.version

这个意义不大，后续会在自建模板中取消这个监控项。

### 3.3 启用 mysql.status[\*]

不需要修改，使用默认配置即可，下面是对此 key 的理解：

```
cat /etc/zabbix/zabbix_agentd.d/userparameter_mysql.conf
```

```
UserParameter=mysql.status[*],echo "show global status where Variable_name='$1';" |
```

```
HOME=/etc/zabbix/zabbix_agentd.d mysql -N | awk '{print $2}'
```

-N：不显示表头

这个 key 值其实就是对使用 show global status 得到的信息进行筛选和展示，比如：

```

root@iZuf61nha4v6rymngnvx79Z:~# echo "show global status where Variable_name='Com_update';"|mysql -uzabbix -p
Enter password:
Variable_name  Value
Com_update    297932050
root@iZuf61nha4v6rymngnvx79Z:~# echo "show global status where Variable_name='Com_update';"|mysql -N -uzabbix -p
Enter password:
Com_update    297932055

```

可选的\$1 有

Com\_update: mysql 执行的更新条数

Com\_select: mysql 执行的查询条数

Com\_insert: mysql 执行插入的条数

Com\_delete: 执行删除的条数

Com\_rollback: 执行回滚的操作条数

Com\_begin: 执行 begin 语句条数

Com\_commit: 执行 commit 语句条数

Questions : 执行的语句总数

Bytes\_received:接受的字节数

Bytes\_sent: 发送的字节数

Slow\_queries: 慢查询语句的条数

Uptime: 服务运行时长

以上 12 个即是模板自带的，可直接使用。

### 3.4 启用 mysql.size[\*]

默认模板 Template DB MySQL 中没有这个监控项，需要在 2.5 步中新增。

配置文件不需要修改，使用默认配置即可，下面是对此 key 的理解：

cat /etc/zabbix/zabbix\_agentd.d/userparameter\_mysql.conf

默认配置的有以下 key，语句比较长，不好理解，将语句整理显示如下，可能更好读。

UserParameter=mysql.size[\*],

bash -c

,

echo

"

select sum(

\$(

case "\$3" in

both|")

echo "data\_length+index\_length";;

data|index)

echo "\$3\_length";;

```

        free)
            echo "data_free";;
        esac)
    )
    from information_schema.tables
    $
    (
        [[ "$1" = "all" || ! "$1" ]] || echo " where table_schema=\"$1\""
    )
    $
    (
        [[ "$2" = "all" || ! "$2" ]] || echo "and table_name=\"$2\""
    )
    ;
"
| HOME=/var/lib/zabbix mysql -N

```

该 key 接受 3 个参数:

\$1 可选 all|数据库名|空, 表示获取所有库还是指定库,默认是所有

\$2 可选 all|表名|空,表示获取所有表还是指定表, 默认是所有

\$3 可选 both|data|index|free|空

both 或者空, 表示统计 data\_length 和 index\_length 所有

data,表示 data\_length

index,表示 index\_length

free,表示 data\_free

这个 key,其实就是执行如下命令, 再经过指定库、指定表、指定项等进行汇总得到的结果。

```

mysql> select table_schema,table_name,index_length,data_length,data_free from information_schema.tables;
+-----+-----+-----+-----+-----+
| table_schema | table_name | index_length | data_length | data_free |
+-----+-----+-----+-----+-----+
| information_schema | CHARACTER_SETS | 0 | 0 | 0 |
| information_schema | COLLATIONS | 0 | 0 | 0 |
| information_schema | COLLATION_CHARACTER_SET_APPLICABILITY | 0 | 0 | 0 |
| information_schema | COLUMNS | 1024 | 0 | 0 |
| information_schema | COLUMN_PRIVILEGES | 0 | 0 | 0 |
| information_schema | ENGINES | 0 | 0 | 0 |
| information_schema | EVENTS | 1024 | 0 | 0 |
| information_schema | FILES | 0 | 0 | 0 |
| information_schema | GLOBAL_STATUS | 0 | 0 | 0 |
| information_schema | GLOBAL_VARIABLES | 0 | 0 | 0 |
| information_schema | KEY_COLUMN_USAGE | 0 | 0 | 0 |
| information_schema | OPTIMIZER_TRACE | 1024 | 0 | 0 |
| information_schema | PARAMETERS | 1024 | 0 | 0 |
| information_schema | PARTITIONS | 1024 | 0 | 0 |

```

统计所有 mysql.size[all,all,both], 其中\$1,\$2,\$3 都可以为空, 若为空, 则取默认值



```

root@iZuf61nha4v6rymngnvx79Z:/etc/zabbix/zabbix_agentd.d# zabbix_get -s 127.0.0.1 -k mysql.size[all,all,both]
24307277963
root@iZuf61nha4v6rymngnvx79Z:/etc/zabbix/zabbix_agentd.d#
root@iZuf61nha4v6rymngnvx79Z:/etc/zabbix/zabbix_agentd.d#
root@iZuf61nha4v6rymngnvx79Z:/etc/zabbix/zabbix_agentd.d# zabbix_get -s 127.0.0.1 -k mysql.size[all,all,]
24307277963
root@iZuf61nha4v6rymngnvx79Z:/etc/zabbix/zabbix_agentd.d#
root@iZuf61nha4v6rymngnvx79Z:/etc/zabbix/zabbix_agentd.d#
root@iZuf61nha4v6rymngnvx79Z:/etc/zabbix/zabbix_agentd.d# zabbix_get -s 127.0.0.1 -k mysql.size[all,,]
24307277963
root@iZuf61nha4v6rymngnvx79Z:/etc/zabbix/zabbix_agentd.d#
root@iZuf61nha4v6rymngnvx79Z:/etc/zabbix/zabbix_agentd.d# zabbix_get -s 127.0.0.1 -k mysql.size[,,,]
24307277963
root@iZuf61nha4v6rymngnvx79Z:/etc/zabbix/zabbix_agentd.d#
root@iZuf61nha4v6rymngnvx79Z:/etc/zabbix/zabbix_agentd.d#
root@iZuf61nha4v6rymngnvx79Z:/etc/zabbix/zabbix_agentd.d# zabbix_get -s 127.0.0.1 -k mysql.size[,all,]
24307277963
root@iZuf61nha4v6rymngnvx79Z:/etc/zabbix/zabbix_agentd.d#

```

指定库 mysql.size[information\_schema,all,both]:

```

root@iZuf61nha4v6rymngnvx79Z:~#
root@iZuf61nha4v6rymngnvx79Z:~# zabbix_get -s 127.0.0.1 -k mysql.size[information_schema,all,both]
10240
root@iZuf61nha4v6rymngnvx79Z:~#
root@iZuf61nha4v6rymngnvx79Z:~#

```

指定表: mysql.size[information\_schema,tables,both]

```

root@iZuf61nha4v6rymngnvx79Z:/var/lib/mysql#
root@iZuf61nha4v6rymngnvx79Z:/var/lib/mysql# zabbix_get -s 127.0.0.1 -k mysql.size[zabbix,images,both]
1605632
root@iZuf61nha4v6rymngnvx79Z:/var/lib/mysql#

```

指定查 data\_free, mysql.size[zabbix,all,free]

```

root@iZuf61nha4v6rymngnvx79Z:/var/lib/mysql#
root@iZuf61nha4v6rymngnvx79Z:/var/lib/mysql# zabbix_get -s 127.0.0.1 -k mysql.size[zabbix,all,free]
53477376
root@iZuf61nha4v6rymngnvx79Z:/var/lib/mysql#
root@iZuf61nha4v6rymngnvx79Z:/var/lib/mysql#

```

其他情况, 与此类似, 有需要自己添加即可

备注:

data\_length: 记录表的大小 (单位字节, 下同)

index\_length: 记录表的索引的大小

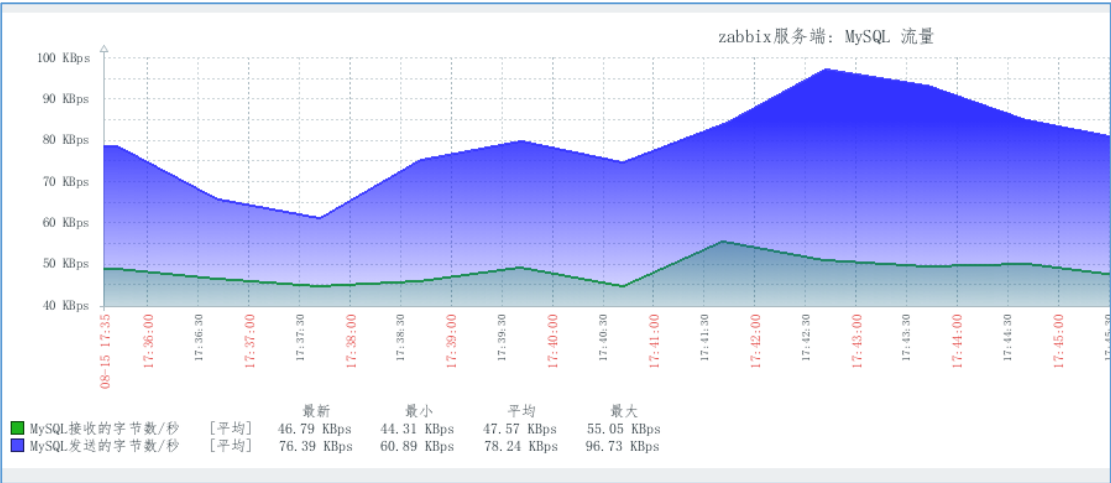
Data\_free: 空间碎片大小

# 四 观察

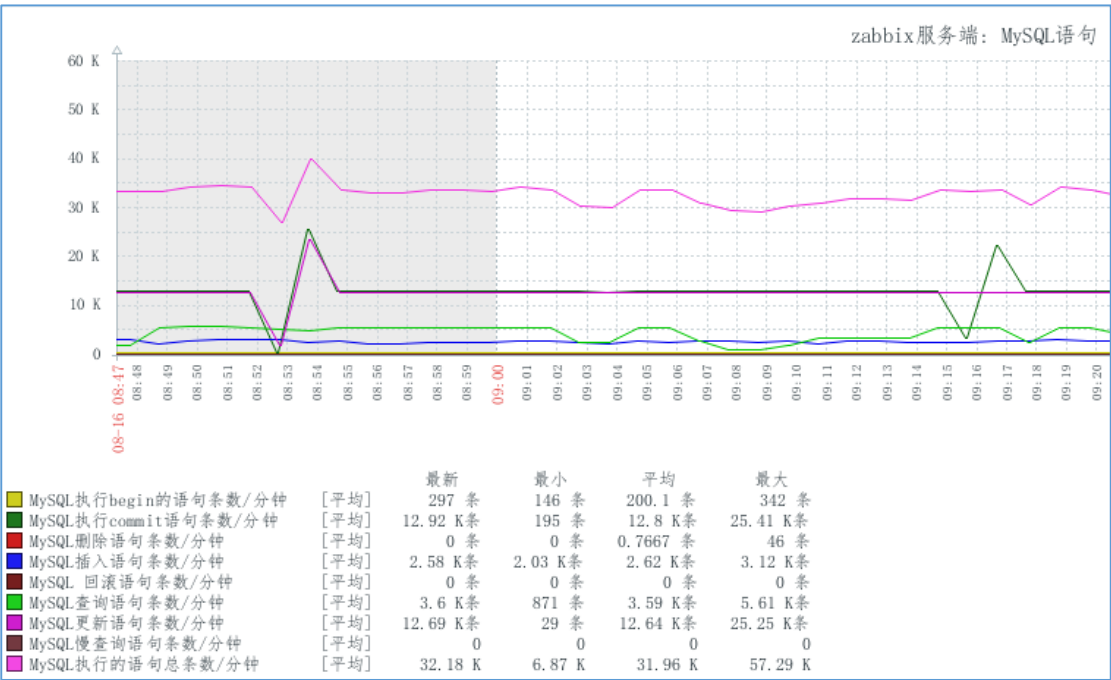
在 2.5 节中, 新增的监控模板一共有 14 个监控项, 除了 mysql.status[Uptime], mysql.ping 外还有 12 个监控项, 绘制成 3 张图, 如下。

## 4.1 MySQL 进出流量

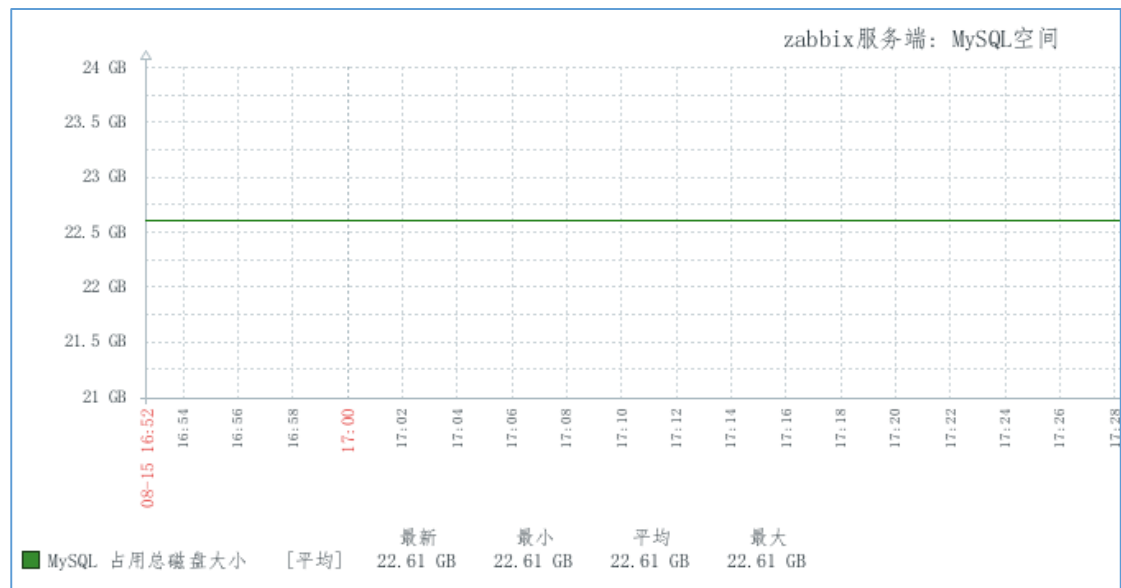
此处我是在 2.5 节修改了默认图形的名称、修改了 item 的名称, 才有如下显示, 下同。



## 4.2 mysql 语句统计



## 4.3 MySQL 占用空间



## 五 新增自定义监控项

### 5.1 介绍

zabbix 自带的一些 key 值，适用于单机情况，对于 mysql 主从复制的集群，也需要监控下主从同步的状态。这一章节，就是解决这个问题。

我们知道 mysql 的主从同步依靠三个进程：master 上的 Binlog Dump 线程，slave 上的 IO、SQL 线程，当然主从同步延迟也应该被监控。

监控的值其实就是在 mysql slave 中使用 show slave status 查出的三个值，  
Slave\_IO\_Running: IO 线程状态  
Slave\_SQL\_Running: SQL 线程状态  
Seconds\_Behind\_Master: 从同步延迟时间

```
mysql> show slave status \G
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
      Master_Host: 10.81.130.117
      Master_User: rpl
      Master_Port: 3306
      Connect_Retry: 60
      Master_Log_File: mysql-bin.000106
      Read_Master_Log_Pos: 1290323
      Relay_Log_File: iZuf6giyyobukf6nsmcw7fZ-relay-bin.000246
      Relay_Log_Pos: 1290536
      Relay_Master_Log_File: mysql-bin.000106
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes
      Replicate_Do_DB:
      Replicate_Ignore_DB: mysql_information_schema performance_schema sys
      Master_SSL_CA_File:
      Master_SSL_CA_Path:
      Master_SSL_Cert:
      Master_SSL_Cipher:
      Master_SSL_Key:
      Seconds_Behind_Master: 0
      Master_SSL_Verify_Server_Cert: No
```

### 5.2 新增自定义 key

在启用默认监控项的配置文件基础上，新增如下 key

```
UserParameter=mysql.slave.status[*],echo "show slave status \G" | HOME=/etc/zabbix/zabbix_agentd.d mysql |grep -m1 $1|awk -F: '{print $NF}'
```

```
root@iZuf6giyyobukf6nsmcw7fZ:/etc/zabbix/zabbix_agentd.d# cat mysql.slave.status.conf
UserParameter=mysql.slave.status[*],echo "show slave status \G" | HOME=/etc/zabbix/zabbix_agentd.d mysql |grep -m1 $1|awk -F: '{print $NF}'
root@iZuf6giyyobukf6nsmcw7fZ:/etc/zabbix/zabbix_agentd.d#
```

### 5.3 新增模板

新建模板，添加的监控项如下：

所有模板 / 7 mysql slave服务状态

应用集 1

监控项 3

触发器 3

图形

聚合图形

自动发现规则

Web 场景

主机群组

在此输入搜索

选择

类型

所有

信息类型

主机

7 mysql slave服务状...

选择

更新间隔

历史记录

应用集

选择

名称

键值

应用

重设

过滤器 只影响过滤后的数据

信息类型

数字 (无正负) 1 文本 2

<input type="checkbox"/>	Wizard	名称 ▲	触发器	键值
<input type="checkbox"/>	...	IO线程状态	触发器 1	mysql.slave.status[Slave_IO_Running]
<input type="checkbox"/>	...	SQL线程状态	触发器 1	mysql.slave.status[Slave_SQL_Running]
<input type="checkbox"/>	...	主从同步延迟	触发器 1	mysql.slave.status[Seconds_Behind_Master]

对应的可以建 3 个触发器

所有模板 / 7 mysql slave服务状态

应用集 1

监控项 3

触发器 3

图形

聚合图形

自动发现规则

Web 场景

严重性

所有

未分类

信息

警告

一般严重

严重

灾难

状态

所有

正常

未知的

状态

所有

已启用

停用的

应用

重设

<input type="checkbox"/>	严重性	名称 ▲	表达式
<input type="checkbox"/>	严重	IO线程状态	{mysql slave status monitor:mysql.slave.status[Slave_IO_Running].str(Yes)}=0
<input type="checkbox"/>	严重	SQL线程状态	{mysql slave status monitor:mysql.slave.status[Slave_SQL_Running].str(Yes)}=0
<input type="checkbox"/>	严重	从同步延迟	{mysql slave status monitor:mysql.slave.status[Seconds_Behind_Master].last()}>10

## 5.4 效果观察

ZABBIX

监测中

资产记录

报表

配置

仪表板

问题

概览

Web监测

最新数据

图形

聚合器

-mysql-slave: IO线程状态

时间戳	值
2018-08-16 10:37:34	Yes
2018-08-16 10:36:34	Yes
2018-08-16 10:35:34	Yes
2018-08-16 10:34:34	Yes
2018-08-16 10:33:34	Yes
2018-08-16 10:32:34	Yes
2018-08-16 10:31:34	Yes
2018-08-16 10:30:34	Yes
2018-08-16 10:29:34	Yes