

SECURITY DAY06



# 服务安全与监控

NSD SECURITY

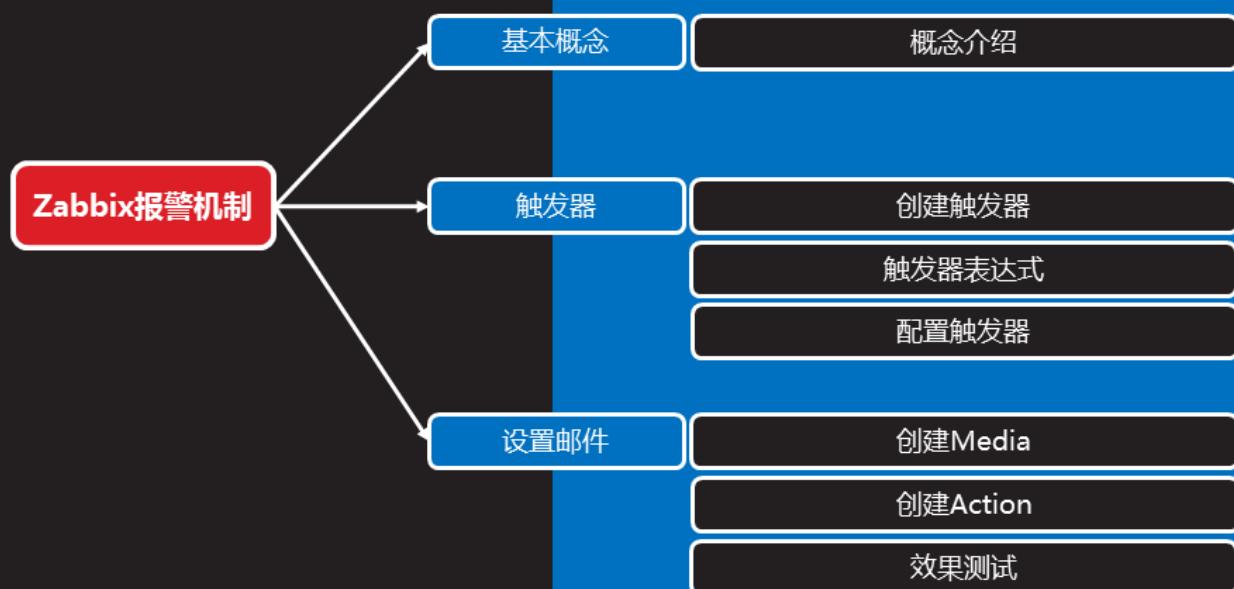
DAY06

# 内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	Zabbix报警机制
	10:30 ~ 11:20	
	11:30 ~ 12:00	Zabbix进阶操作
下午	14:00 ~ 14:50	
	15:00 ~ 15:50	监控案例
	16:10 ~ 17:00	
	17:10 ~ 18:00	总结和答疑



## Zabbix报警机制



# 基本概念

## 概念介绍

- 自定义的监控项默认不会自动报警
- 首页也不会提示错误
- 需要配置触发器与报警动作才可以自定报警

知识讲解

Problems						
Time ▾	Recovery time	Status	Info	Host	Problem • Severity	Duration
02/14/2018 09:20:07 PM	•	PROBLEM		<u>zabbix_client_web</u>	HTTP service is down on <u>zabbix_client_web</u>	2d 14h 51m



## 概念介绍（续1）

知识讲解

- 触发器 ( trigger )
  - 表达式，如内存不足300M，用户超过30个等
  - 当出发条件发生后，会导致一个触发事件
  - 触发事件会执行某个动作
- 动作 ( action )
  - 触发器的条件被触发后的行为
  - 可以是发送邮件、也可以是重启某个服务等

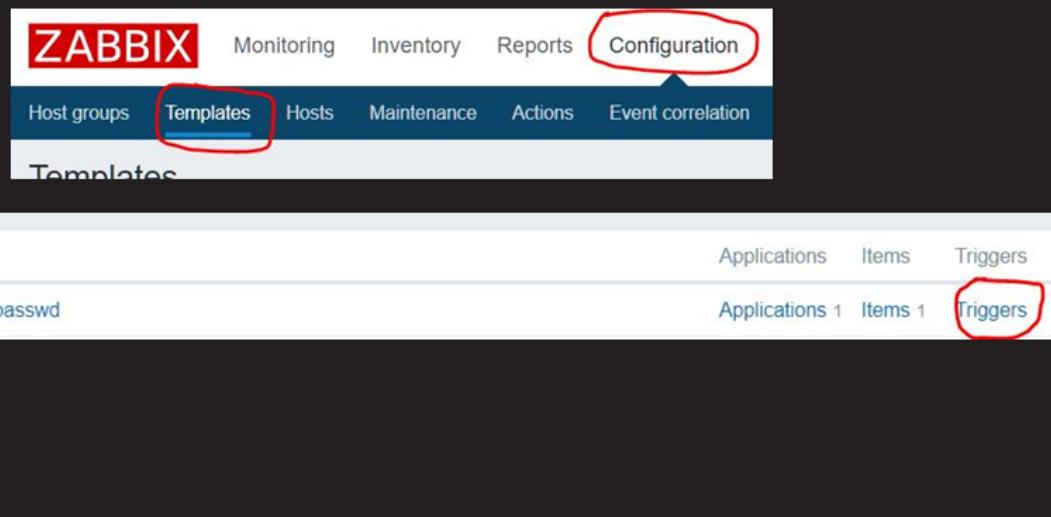


# 触发器

# 创建触发器

知识讲解

- 通过Configuration→Templates
- 选择模板点击后面的Triggers→Create trigger
  - 强烈建议使用英文创建（中文翻译不敢恭维）



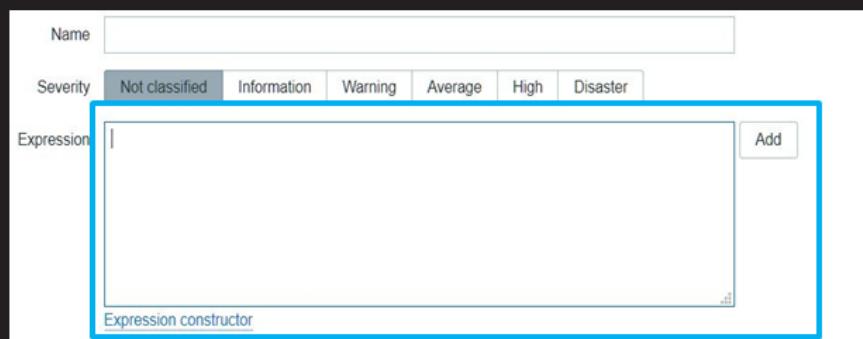
The screenshot shows the Zabbix configuration interface. At the top, there's a navigation bar with tabs: Monitoring, Inventory, Reports, Configuration (which is circled in red), Host groups, Templates (which is also circled in red), Hosts, Maintenance, Actions, and Event correlation. Below this, a sub-menu for 'Templates' is displayed, showing a list of templates: 'Name' (with a checkbox) and 'count.line.passwd'. To the right of the list, there are links for Applications (1), Items (1), and Triggers (1). The 'Triggers' link is also circled in red.

# 触发器表达式

知识讲解

- Expression表达式：触发异常的条件
 

```
{<server>:<key>.<function>(<parameter>){<operator><constant>
{主机 : key.函数(参数)}<表达式>常数}
```



The screenshot shows the 'Create trigger' dialog in Zabbix. It has fields for 'Name' (empty), 'Severity' (set to 'Not classified'), and an 'Expression' field which contains the placeholder '{<server>:<key>.<function>(<parameter>){<operator><constant>}' followed by an empty expression constructor area. There are tabs for 'Information', 'Warning', 'Average', 'High', and 'Disaster'.

## 触发器表达式（续1）

知识讲解

- Expression表达式案例

`{web1:system.cpu.load[all,avg1].last(0)}>5` //0为最新数据

如果web1主机最新的CPU平均负载值大于5，则触发器状态Problem

`{vfs.fs.size[/,free].max(5m)}<10G` //5m为最近5分钟

根分区，最近5分钟的最大容量小于10G，则状态进入Problem

`{vfs.file.cksum[/etc/passwd].diff(0)}>0` //0为最新数据

最新一次校验/etc/passwd如果与上一次有变化，则状态进入Problem



## 触发器表达式（续2）

知识讲解

- Expression表达式案例

- 大多数函数使用秒作为参数，使用#代表不同含义

- avg, count, last, min and max 函数支持额外的第二个参数`time_shift`（时间偏移量）

- 这个参数允许从过去一段时间内引用数据。

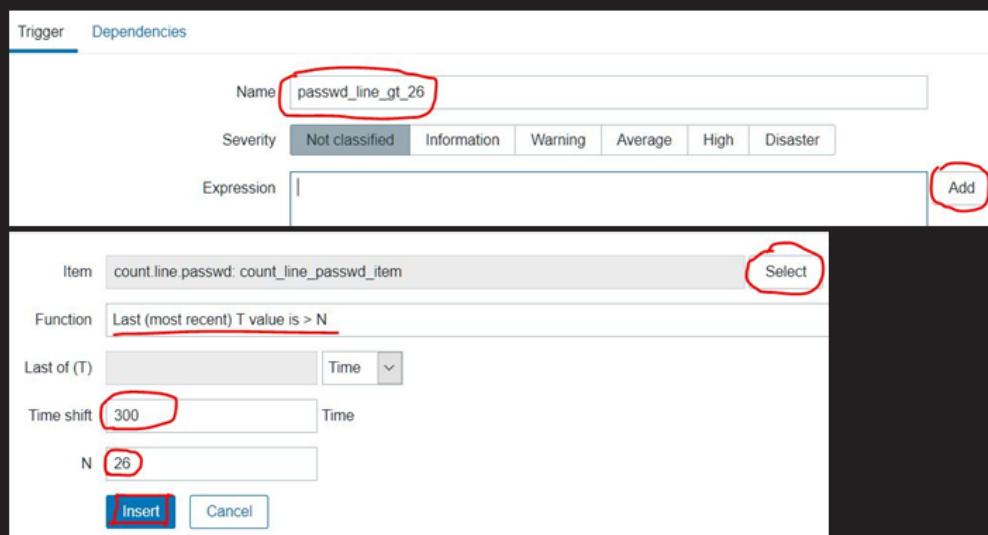
函数内容	描述
<code>sum(600)</code>	600秒内所有值的总和
<code>sum(#5)</code>	最后5个值的总和
<code>last(20)</code>	最后20秒的值
<code>last(#5)</code>	倒数第5个值
<code>avg(1h,1d)</code>	一天前的1小时的平均值



## 配置触发器

知识讲解

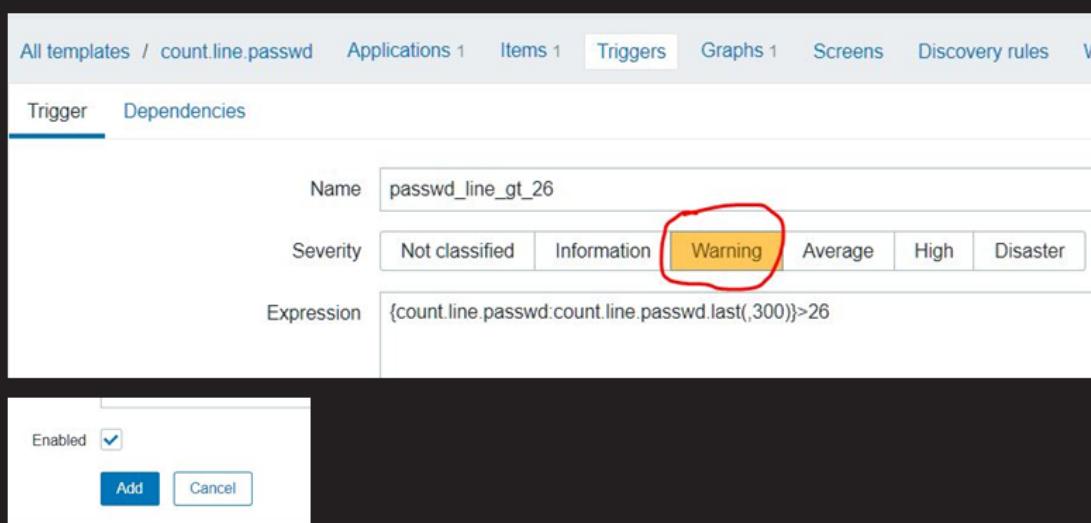
- 设置触发器名称，点击add添加表达式
- 填写表达式
  - 监控项为账户数量，最近300秒账户数量大于26



## 配置触发器（续1）

知识讲解

- 选择触发器报警级别
- Add创建该触发器



# 设置邮件

## 创建Media

知识讲解

- 设置邮件服务器
  - Administration → Media Type → 选择Email邮件
  - 设置邮件服务器信息

The screenshot shows the Zabbix Administration interface. On the left, there's a blue sidebar labeled "知识讲解". The main menu has tabs: 监测中, 资产记录, 报表, 配置, 管理 (highlighted with a red circle). Below the menu, there are sub-tabs: 一般, agent代理程序, 认证, 用户群组, 用户 (highlighted with a red circle), 报警媒介类型 (highlighted with a red circle), and 脚本. A large red circle highlights the "管理" tab. A smaller red circle highlights the "报警媒介类型" tab. In the center, a table lists media types:

名称	类型
Email	电子邮件
Jabber	Jabber
SMS	短信

On the right, a detailed configuration dialog for "报警媒介类型" is shown. It has tabs: 报警媒介类型 (selected) and 选项. The configuration fields include:

- 名称: Email
- 类型: 电子邮件 (selected)
- SMTP服务器: localhost (highlighted with a red circle)
- SMTP服务器端口: 25
- SMTP HELO: company.com
- SMTP电邮: root@localhost (highlighted with a red circle)
- 安全链接: 无 (selected)
- 认证: 无 (selected)
- 已启用: checked

At the bottom are buttons: 更新, 克隆, 删除, 取消.

## 创建Media (续2)

- 为账户添加Media
  - 在Administration→Users中找到选择admin账户

知识讲解



The screenshot shows the Zabbix administration interface. At the top, there's a navigation bar with tabs: 监测中 (Monitoring), 资产记录 (Assets), 报表 (Reports), 配置 (Config), and 管理 (Management). The Management tab is circled in red. Below it, there are sub-tabs: 一般 (General), agent代理程序 (Agent), 认证 (Authentication), 用户群组 (User Groups), and 用户 (User). The User tab is also circled in red. The main content area displays a list of users with columns: 别名 (Alias), 用户名第一部分 (First Part of Username), and 姓名 (Name). The 'Admin' user is listed and circled in red.



## 创建Media (续3)

- 选择Media菜单→点击Add添加报警媒介
  - 在Media Type中填写报警类型，收件人，时间等信息

知识讲解



The screenshot consists of two parts. On the left, the Zabbix configuration interface is shown with the 'Media' tab selected (circled in red). Below it, there's a 'Add' button highlighted with a red circle. On the right, a detailed configuration dialog for 'Media' is displayed. It includes fields for 'Type' (set to 'Email'), 'Recipient' (set to 'root@localhost'), and a section for '如果存在严重性则使用' (If severity exists) with several checkboxes checked: '未分类' (Unclassified), '信息' (Information), '警告' (Warning), '一般严重' (General Severity), '严重' (Severe), and '灾难' (Catastrophic). There's also a '已启用' (Enabled) checkbox and '添加' (Add) and '取消' (Cancel) buttons at the bottom.



# 创建Action

知识讲解

- Action ( 行为 )
  - 定义当触发器被触发时，执行什么Action
  - 通过Configuration→Actions→Create action创建

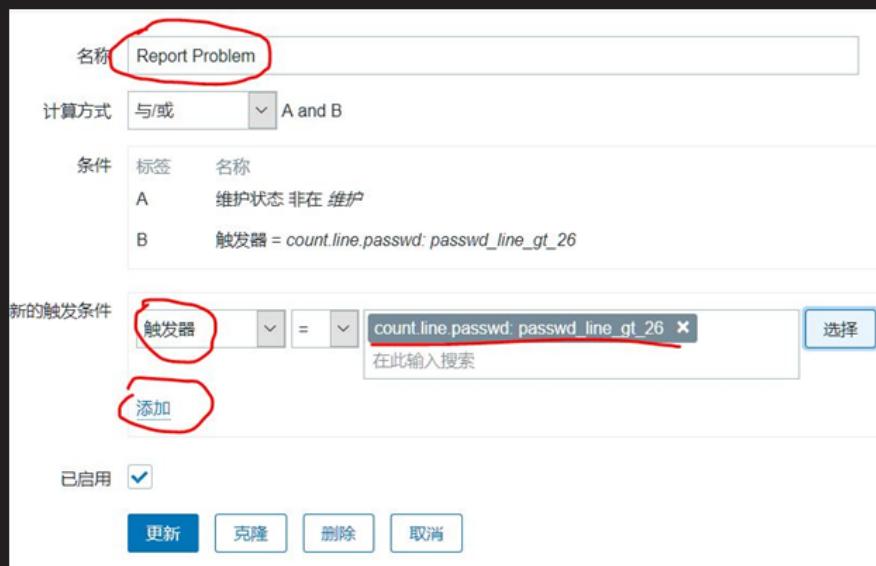


+

## 创建Action ( 续1 )

知识讲解

- 配置Action ( 填写名称 )
- 配置导致动作的触发条件 ( 账户大于26 )

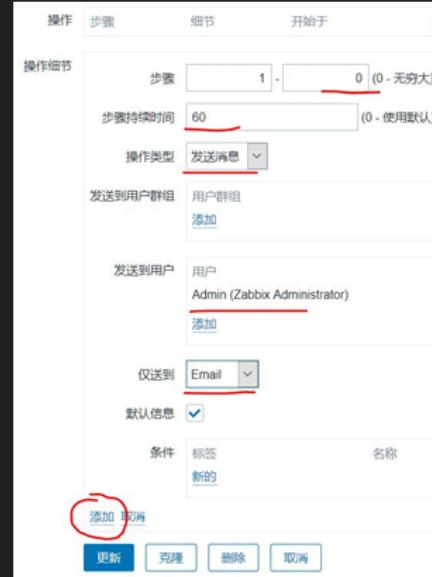


名称	Report Problem
计算方式	与/或
条件	标签 A 维护状态 非在 维护 B 触发器 = count.line.passwd: passwd_line_gt_26
新的触发条件	触发器 添加
已启用	<input checked="" type="checkbox"/>
<input type="button" value="更新"/> <input type="button" value="克隆"/> <input type="button" value="删除"/> <input type="button" value="取消"/>	

## 创建Action ( 续2 )

- 配置动作的具体操作行为 ( 发送信息或执行远程命令 )
  - 无限次数发送邮件 , 60秒1次 , 发送给Admin用户

知识讲解



+

## 效果测试

知识讲解

- 在被监控主机创建账户
- 登录监控端Web页面，在仪表盘中查看问题



问题						
时间	恢复时间	状态	信息	主机	问题·严重性	持续时间
23:13:39		问题		zabbix_client_web	passwd_line_gt_26	57s

+

## 效果测试（续1）

- 在监控服务器上使用mail命令查收报警邮件

```
>N 35 root@localhost.local Sat Feb 17 10:15 20/846 "Problem: passwd_line_gt_26"
N 36 root@localhost.local Sat Feb 17 10:15 21/923 "Problem: /etc/passwd has bee
& 35
Message 35:
From root@localhost.localdomain Sat Feb 17 10:15:41 2018
Return-Path: <root@localhost.localdomain>
X-Original-To: root@localhost
Delivered-To: root@localhost.localdomain
From: <root@localhost.localdomain>
To: <root@localhost.localdomain>
Date: Sat, 17 Feb 2018 10:15:41 -0500
Subject: Problem: passwd_line_gt_26
Content-Type: text/plain; charset="UTF-8"
Status: R

Problem started at 10:13:39 on 2018.02.17
Problem name: passwd_line_gt_26
Host: zabbix_client_web
Severity: Warning
```

知识讲解



## 案例1：实现Zabbix报警功能

- 使用Zabbix监控软件实现报警机制
  - 创建Media，设置邮件服务器及收件人邮箱
  - 设置触发器规则，当Linux账户数量超过26个时，自动发送邮件报警

课堂练习



# Zabbix进阶操作

## Zabbix进阶操作

自动发现

概述

自动发现规则

创建动作

创建新的主机

主被动监控

概述

创建新的主机

克隆模板

修改监控项模式

添加监控主机

验证效果

拓扑图与聚合图形

拓扑图

聚合图形

Tedu.cn  
达内教育

## 概述

知识讲解

- 自动发现 ( Discovery )
  - 当Zabbix需要监控的设备越来越多，手动添加监控设备越来越有挑战，此时，可以考虑使用自动发现功能
  - 需要批量一次性添加一组监控主机，也可以使用自动发现功能
- 自动发现可以实现：
  - 自动发现、添加主机，自动添加主机到组
  - 自动连接模板到主机，自动创建监控项目与图形等



## 概述（续1）

知识讲解

- 自动发现（Discovery）流程
  - 创建自动发现规则
  - 创建Action动作，说明发现主机后自动执行什么动作
  - 通过动作，执行添加主机，链接模板到主机等操作



## 自动发现规则

知识讲解

- 创建自动发现规则
  - Configuration→Discovery→Create discovery rule



# 自动发现规则（续1）

知识讲解

- 填写规则
  - 自动发现的IP范围（逗号隔开可以写多个）
  - 多久做一次自动发现（默认为1小时，仅实验修改为1m）
  - 检查的方式：HTTP、FTP、Agent的自定义key等检查



+

# 创建动作

知识讲解

- Configuration → Actions
- Event source(Discovery) → Create action
  - 注意：选择事件源为：自动发现

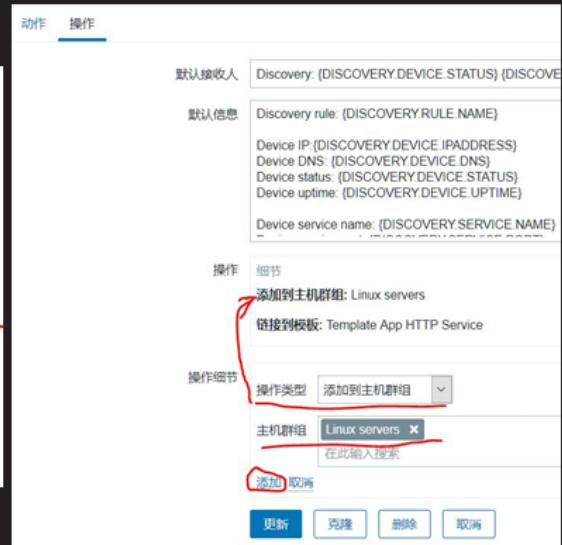
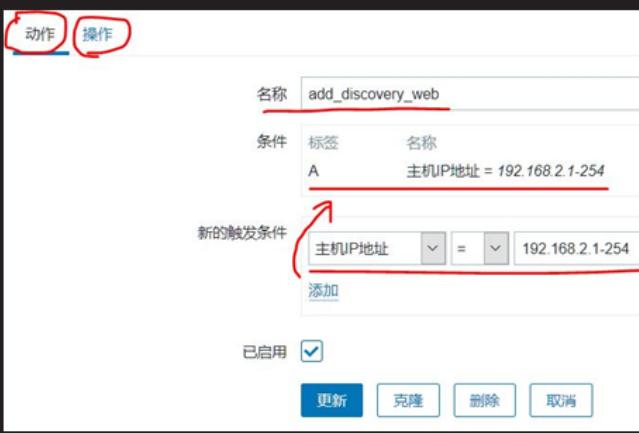


+

## 创建动作 (续1)

- 添加动作名称，添加触发动作的条件
- 操作（触发动作后要执行的操作指令），操作细节如下：
  - 添加主机到组，与模板链接（HTTP模板）

知识讲解



## 创建新的主机

- 创建一台新的主机，验证zabbix是否可以自动发现该主机
  - 可以重新部署一台新的虚拟机
  - 也可以将旧虚拟机的IP地址，临时修改为其他IP
- 登陆Zabbix服务器的Web页面，查看主机列表

知识讲解

## 案例2：Zabbix自动发现

课堂练习

- 配置Zabbix的自动发现机制
  - 创建自动发现规则
  - 创建自动发现后的动作，添加主机、为主机链接模板



## 主被动监控

## 概述

- 主动和被动都是对被监控端主机而言的
- 默认zabbix采用的是被动监控
  - 被动监控：Server向Agent发起连接，发送监控key，Agent接受请求，响应监控数据
  - 主动监控：Agent向Server发起连接，Agent请求需要检测的监控项目列表，Server响应Agent发送一个items列表，Agent确认收到监控列表，TCP连接完成，会话关闭，Agent开始周期性地收集数据
- 区别：
  - Server不用每次需要数据都连接Agent，Agent会自己收集数据并处理数据，Server仅需要保存数据即可



## 概述（续1）

- 当监控主机达到一定量级后，Zabbix服务器会越来越慢
- 此时，可以考虑使用主动监控，释放服务器的压力
- 另外，Zabbix也支持分布式监控，也是可以考虑的方案



# 创建新的主机

- 创建新的被监控主机（主动监控）
  - 给Web2 ( 192.168.2.200 ) 安装zabbix\_agent软件

知识讲解

```
[root@web2 ~]# yum -y install gcc pcre-devel
[root@web2 ~]# tar -xf zabbix-3.4.4.tar.gz
[root@web2 ~]# cd zabbix-3.4.4/
[root@web2 ~]# ./configure --enable-agent
[root@web2 ~]# make && make install
```



## 创建新的主机（续1）

知识讲解

- 修改配置文件

```
[root@web2 ~]# vim /usr/local/etc/zabbix_agentd.conf
#Server=127.0.0.1,172.25.0.10 //注释该行，允许谁监控本机
StartAgents=0
//被动监控时启动多个进程
//设置为0，则禁止被动监控，不启动zabbix_agentd服务
ServerActive=172.25.0.10
//允许哪些主机监控本机（主动模式），一定要取消127.0.0.1
Hostname=zabbix_client_web2
//告诉监控服务器，是谁发的数据信息
//一定要和zabbix服务器配置的监控主机名称一致（后面设置）
RefreshActiveChecks=120
//默认120秒检测一次
UnsafeUserParameters=1 //允许自定义key
Include=/usr/local/etc/zabbix_agentd.conf.d/
[root@web2 ~]# zabbix_agentd //启动服务
```



## 克隆模板

知识讲解

- 为了方便，克隆系统自带模板（在此基础上就该更方便）
- Configuration → Templates
  - 选择Template OS Linux
  - 全克隆该模板，新建一个新的模板
  - 新模板名称为：Template OS Linux ServerActive



## 修改监控项模式

知识讲解

- 将模板中的所有监控项目全部修改为主动监控模式
  - Configuration → Templates
  - 选择新克隆的模板，点击后面的Items（监控项）
  - 点击全选，选择所有监控项目，点击批量更新
  - 将类型修改为：Zabbix Agent ( Active主动模式 )



## 修改监控项模式 (续1)

知识讲解

- 批量修改监控项的监控模式后，并非所有监控项目都支持主动模式
  - 批量修改后，会发现有几个没有修改主动模式成功
  - 说明，这些监控项目不支持主动模式，关闭即可
  - 可以点击类型排序，方便操作，点击状态即可关闭



Wizard	名称	触发器	键值	间隔	历史记录	趋势	类型	应用集	状态
	Template App Zabbix Agent: Version of zabbix_agentd running	触发器 1	agent.version	1h	1w	Zabbix 客户端	Zabbix agent		停用的
	Template App Zabbix Agent: Host name of zabbix_agentd running	触发器 1	agent.hostname	1h	1w	Zabbix 客户端	Zabbix agent		停用的
	Template App Zabbix Agent: Agent ping	触发器 1	agent.ping	1m	1w	365d	Zabbix 客户端	Zabbix agent	停用的
	Maximum number of processes	触发器 1	kernel.maxproc	1h	1w	365d	Zabbix客户端(主动式)	OS	已启用



## 添加监控主机

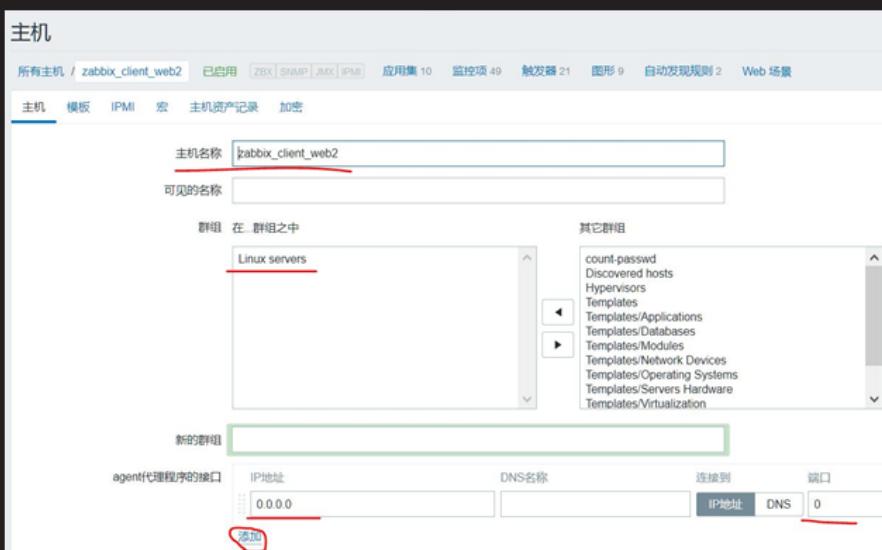
知识讲解

- 在Zabbix监控服务器，添加被监控的主机（主动模式）



## 添加监控主机（续1）

- 名称：zabbix\_client\_web2 //必须与被监控端的配置文件Hostname一致  
将主机添加到Linux servers组
- IP地址为0.0.0.0，端口为0 //不填写IP无法创建成功



## 添加监控主机（续2）

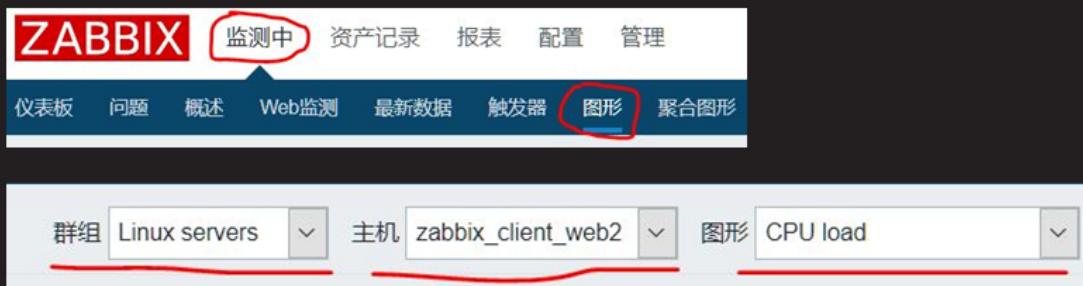
- 为主机添加监控模板
  - 选择刚刚创建的模板（主动模式）
  - 添加链接模板到主机



## 验证效果

知识讲解

- 查看数据图表
  - Monitoring → Graphs
  - 选择需要查看的主机组、主机以及图形



## 验证效果（续1）

知识讲解

- 但是，查看分区图表时并无数据
- 因为分区数据采用的是自动发现监控，与普通监控项一样，修改为主动模式即可
  - 选择Template OS Linux ServerActive模板
  - 修改Discovery自动发现为主动模式

名称	监控项	触发器	图形	主机	键值	间隔	类型
Mounted filesystem discovery	监控项原型 5	触发器原型 2	图形原型 1	主机模板	vfs.fs.discovery	1h	Zabbix客户端(主动式)
Network interface discovery	监控项原型 2	触发器原型	图形原型 1	主机模板	net.if.discovery	1h	Zabbix客户端(主动式)



## 案例3：Zabbix主动监控

课堂练习

- 配置Zabbix主动监控，实现如下功能
  - 修改被监控主机agent为主动监控模式
  - 克隆模板，修改模板为主动监控模板
  - 添加监控主机，并链接主动监控模板



## 拓扑图与聚合图形

# 拓扑图

- 绘制拓扑图可以快速了解服务器架构
- Monitoring → Maps ( 拓扑图 )
- 选择默认的Local network拓扑图，编辑即可

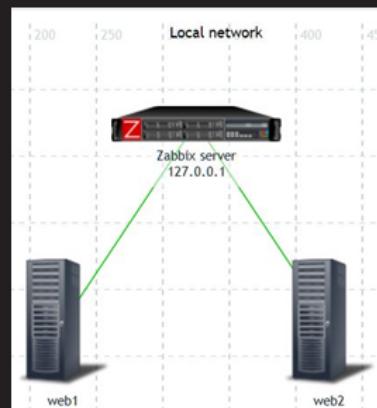


知识讲解



## 拓扑图（续1）

- 操作说明
  - Icon ( 图标 ) , 添加新的设备后可以点击图标修改属性
  - Shape ( 形状 )
  - Link ( 连线 ) , 先选择两个图标，再选择连线
  - 完成后，点击Update ( 更新 )



知识讲解



# 聚合图形

知识讲解

- 在一个页面显示多个数据图表，方便了解多组数据
- Monitoring → Screens ( 聚合图形 ) → Create screen
  - Owner : 使用默认的Admin用户
  - Name : 名称设置为Web2\_host
  - Columns : 列数设置为2列
  - Rows : 行数设置为4行



## 聚合图形 ( 续1 )

知识讲解

- 选择刚刚创建的聚合图形 ( web2\_host )
- 点击后面的构造函数 ( constructor )
  - 点击 Change(更改) , 设置每行每列需要显示的数据图表



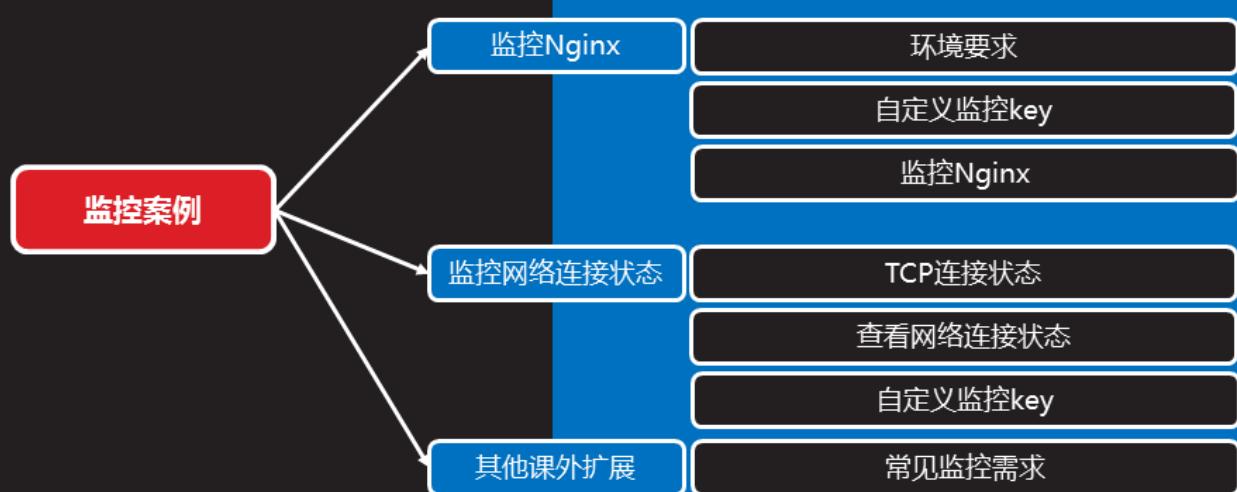
## 案例4：拓扑图与聚合图形

- 创建拓扑图
- 创建聚合图形

课堂练习



### 监控案例



# 监控Nginx

## 环境要求

- 一台Nginx服务器，部署Nginx时要加载status模块

```
[root@web1 nginx-1.12.2]# ./configure \
> --with-http_stub_status_module
[root@web1 nginx-1.12.2]# make && make install
```

```
[root@web1 ~]# cat /usr/local/nginx/conf/nginx.conf
```

....

```
location /status {
    stub_status on;
}
```

....

```
[root@web1 ~]# curl http://192.168.4.5/status
Active connections: 1
server accepts handled requests
10 10 3
Reading: 0 Writing: 1 Waiting: 0
```

知识讲解



# 自定义监控key

知识讲解

- 语法格式
  - UserParameter=key,command
  - UserParameter=key[\*],<command>
  - key里的所有参数，都会传递给后面命令的位置变量

如：

UserParameter=ping[\*],echo \$1  
ping[0] ,      返回的结果都是0  
ping[aaa] ,    返回的结果都是aaa



## 自定义监控key ( 续1 )

知识讲解

- 被监控端修改配置文件
  - 注意要允许自定义key并设置Include

```
[root@web1 ~]# vim /usr/local/etc/zabbix_agentd.conf.d/nginx.status  
UserParameter=nginx.status[*],/usr/local/bin/nginx_status.sh $1
```

```
[root@web1 ~]# killall zabbix_agentd  
[root@web1 ~]# zabbix_agentd
```



## 自定义key ( 续2 )

- 编写脚本（仅供参考，未检测完整状态）

```
[root@web1 ~]# vim /usr/local/bin/nginx_status.sh
#!/bin/bash
case $1 in
active)
    curl -s http://127.0.0.1/status |awk '/Active/{print $NF}';;
waiting)
    curl -s http://127.0.0.1/status |awk '/Waiting/{print $NF}';;
accepts)
    curl -s http://127.0.0.1/status |awk 'NR==3{print $2 }';;
esac
[root@web1 ~]# chmod +x /usr/local/bin/nginx_status.sh
测试效果：
[root@web1 ~]# zabbix_get -s 127.0.0.1-k 'nginx.status[accepts]'
```

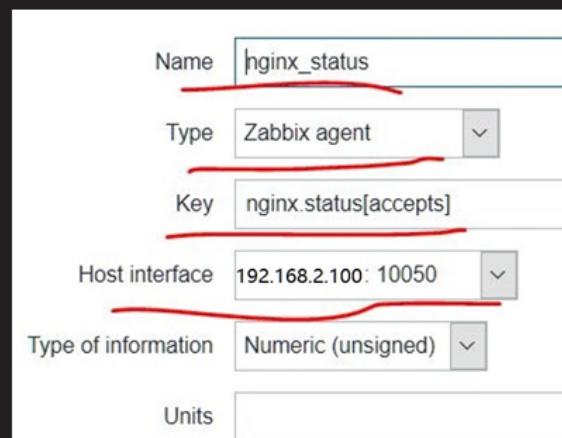
知识讲解



## 监控nginx

- 在监控服务器，添加监控项目item
  - Configuration→Hosts→点击主机后面的items
  - 点击Create item

知识讲解

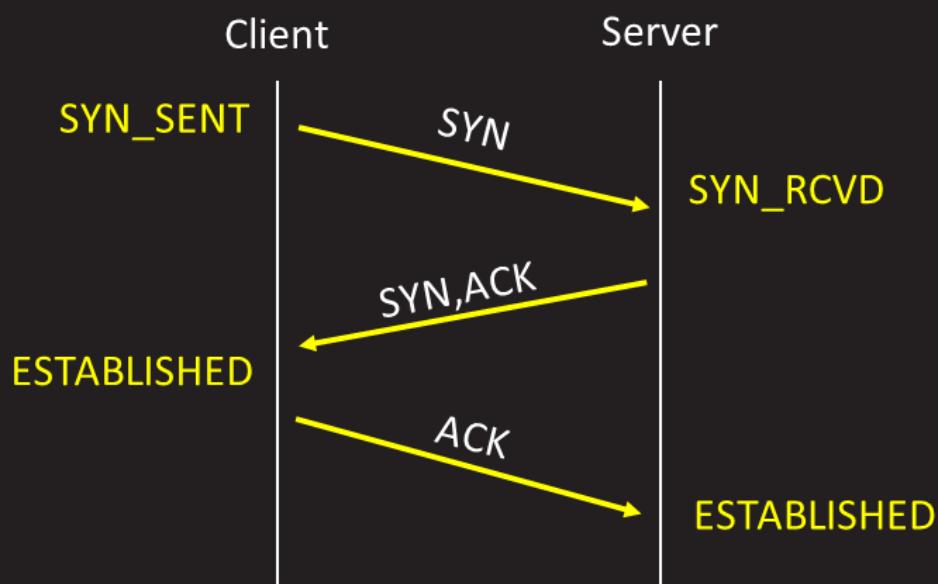


# 监控网络连接状态

## TCP连接状态

- 建立连接的3次握手

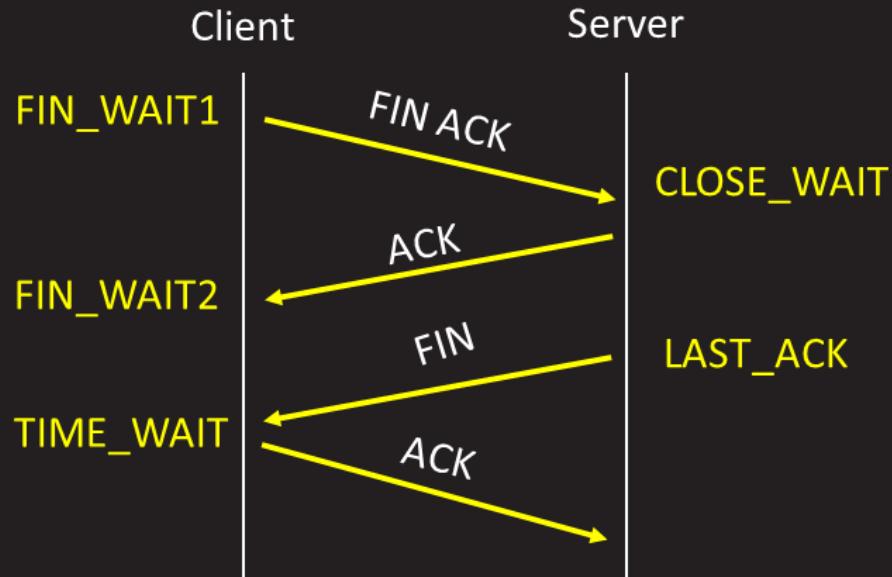
知识讲解



## TCP连接状态 (续1)

- 断开连接的4次握手

知识讲解



## 查看网络连接状态

知识讲解



- 模拟多人并发连接

```
[root@web1 ~]# ab -c 1000 -n 100000 http://192.168.2.100/
```

- 查看网络连接状态
  - 仔细观察、分析第二列的数据

```
[root@web1 ~]# ss -antup
// -a 显示所有
// -t 显示TCP连接状态
// -u 显示UDP连接状态
// -n 以数字形式显示端口号和IP地址
// -p 显示连接对应的进程名称
```

# 自定义监控key

知识讲解

- 被监控端修改配置文件
  - 注意要允许自定义key并设置Include

```
[root@web1 ~]# vim /usr/local/etc/zabbix_agentd.conf.d/net.status  
UserParameter=net.status[*],/usr/local/bin/net_status.sh $1
```

```
[root@web1 ~]# killall zabbix_agentd  
[root@web1 ~]# zabbix_agentd
```



## 自定义key ( 续1 )

知识讲解

- 编写脚本（仅供参考，未检测完整状态）

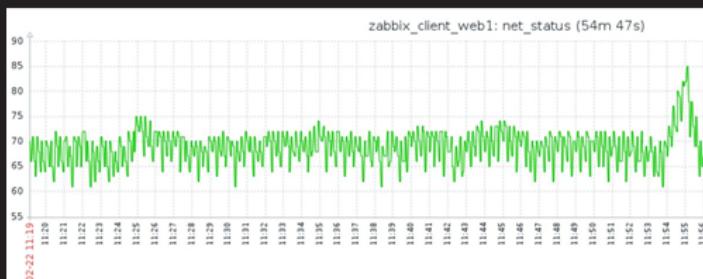
```
[root@web1 ~]# vim /usr/local/bin/net_status.sh  
#!/bin/bash  
case $1 in  
estab)  
    ss -antp | awk '/^TIME-WAIT/{x++} END{print x};;  
close_wait)  
    ss -antp | awk '/^CLOSE-WAIT/{x++} END{print x};;  
time_wait)  
    ss -antp | awk '/^TIME-WAIT/{x++} END{print x};;  
esac  
[root@web1 ~]# chmod +x /usr/local/bin/net_status.sh  
测试效果：  
[root@web1 ~]# zabbix_get -s 127.0.0.1 -k 'net.status[time_wait]'
```



# 监控netstatus

知识讲解

- 在监控服务器，添加监控项目item
  - Configuration→Hosts→点击主机后面的items
  - 点击Create item



名称   

类型  ▼

键值   

主机接口  ▼

信息类型  ▼

单位

更新间隔

## 其他课外扩展

# 常见监控需求

知识讲解

- mysql ( mysqladmin命令 )
  - 并发连接数
  - 慢查询数量
  - 增、删、改、查数量等
- NoSQL数据库 ( 数据库状态 )
- php-fpm ( 生成status页面 )
  - 并发、队列、进程数量等
- tomcat ( 服务器状态 )
- 硬件设备 ( 交换机、路由器等 )
  - 一般通过SNMP监控



## 案例5：自定义监控案例

- 监控Nginx状态
- 监控网络连接状态

课堂练习



## 总结和答疑

总结和答疑

自定义监控错误

问题现象

故障分析及排除



# 自定义监控错误

## 问题现象

- 创建自定义监控项后，提示not supported
- 提示不支持的监控项

知识讲解



## 故障分析及排除

- 原因分析
  - 查看zabbix日志：  
"nginx.status[accepts]" is not supported: Unsupported item key  
not supported: Value "" of type "string" is not suitable for value  
type "Numeric (unsigned)"
- 解决办法
  - 检查被监控端配置文件是否开启允许自定义key
  - 检查被监控端配置文件是否加载了正确的脚本目录Include
  - 检查监控服务器，定义的监控项，数据类型是否正确

知识讲解



