

## Zabbix 通过 api 监控 k8s

监控方式: python+zabbix\_sender

原理：python 的 request 库，请求 k8s 的 api 地址，对数据处理。

## 第一步：获取 k8s 的 api 地址

查看 K8S 的 api 地址的命令如下（最好在 K8s 集群的 master 上执行这条命令，因为如果在 node 节点上执行该命令有可能会获取旧的 api 地址）

```
[root@k8s-master ~]# cat /etc/kubernetes/kubelet.conf | grep server
server: https://192.168.1.140:6443
[root@k8s-master ~]#
```

第二步: K8S 集群的 Master 主机生成一个 token 用来认证, 请求数据 api 数据

生成令牌命令如下（在 K8S 的 Master 主机执行命令）

```
kubectl -n kube-system describe secret $(kubectl -n kube-system get secret | grep admin-  
user | awk '{print $1}')
```

把 token 的值保存在本地的一个文件即可，后期需要把 token 值添加到脚本中。

[illegible]

模板脚本下载地址:

### 第三步：上传脚本并修改脚本

cd /usr/local/zabbix/share/zabbix/externalscripts 目录下创建一个命名 k8s 的目录,把 get k8s.py 脚本放到此目录

```




root@23_233_ZBXmuban k8s]# cd /usr/local/zabbix/share/zabbix/externalscripts/k8s
root@23_233_ZBXmuban k8s]# pwd
/usr/local/zabbix/share/zabbix/externalscripts/k8s
root@23_233_ZBXmuban k8s]# ls
get_k8s.py
root@23_233_ZBXmuban k8s]#

```

[illegible]

需要修改脚本三个地方

上传外部检查 get\_k8s 脚本到/usr/local/zabbix/share/zabbix/externalscript 目录

	get_k8s	2020/6/12 13:11	文件	1 KB
	get_k8s.py	2020/6/12 14:04	PY 文件	6 KB
	zbx_export_templates(3).xml	2020/6/12 14:07	XML 文件	58 KB

#### 第四步创建主机并且套用脚本

注意：主机名称必须是 k8s\_master，否则无法获取数据。（因为数据是通过 zabbix\_sender 发送到该主机）

主机群组 模板 主机 维护 动作 关联项事件 自动发现 服务

### 主机

主机 模板 IPMI 标记 宏 资产记录 加密

\* 主机名称

可见的名称

\* 群组

\* 至少存在一个接口。

agent代理程序的接口	IP地址	DNS名称	连接到	端口	默认
	<input type="text" value="192.168.1.140"/>	<input type="text"/>	<input checked="" type="radio"/> IP地址 <input type="radio"/> DNS	<input type="text" value="10050"/>	<input checked="" type="radio"/> 默认 <input type="button" value="移除"/>
	<input type="button" value="添加"/>				

SNMP接口

JMX接口

IPMI接口

描述

ZABBIX 监测 资产记录 报表 配置 管理

主机群组 模板 主机 维护 动作 关联项事件 自动发现 服务

### 主机

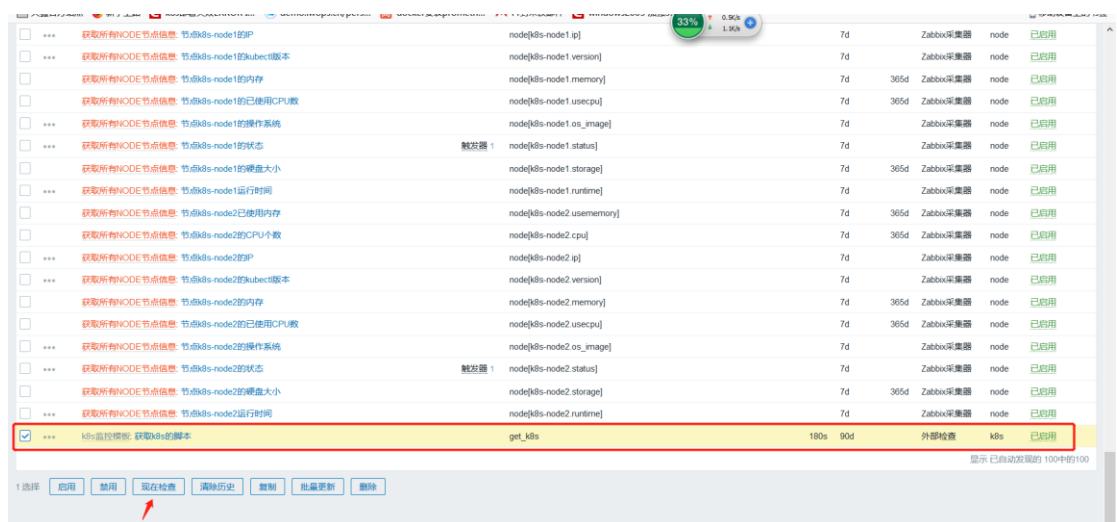
主机 模板 IPMI 标记 宏 资产记录 加密

链接的模板 名称 动作

链接指示器

第五步，手动触发脚本。

选中该监控项。点击立即检查



## 数据效果

Wizard	名称 ▲	触发器	键值	间隔	历史记录	趋势	类型	应用集	状态	依赖
***	获取k8s的健康信息: controller-manager的健康信息		health{controller-manager.message}	7d			Zabbix采集器	health	已启用	
***	获取k8s的健康信息: controller-manager的状态	触发器 1	health{controller-manager.status}	7d			Zabbix采集器	health	已启用	
***	获取k8s的健康信息: etcd-0的健康信息		health{etcd-0.message}	7d			Zabbix采集器	health	已启用	
***	获取k8s的健康信息: etcd-0的状态	触发器 1	health{etcd-0.status}	7d			Zabbix采集器	health	已启用	
	获取所有NODE节点信息: k8s-master的CPU使用率		usecpu{k8s-master}	60s	90d	365d	可计算的	node	已启用	
	获取所有NODE节点信息: k8s-master的内存使用率		usememory{k8s-master}	60s	90d	365d	可计算的	node	已启用	
	获取所有NODE节点信息: k8s-node1的CPU使用率		usecpu{k8s-node1}	60s	90d	365d	可计算的	node	已启用	
	获取所有NODE节点信息: k8s-node1的内存使用率		usememory{k8s-node1}	60s	90d	365d	可计算的	node	已启用	
	获取所有NODE节点信息: k8s-node2的CPU使用率		usecpu{k8s-node2}	60s	90d	365d	可计算的	node	已启用	
	获取所有NODE节点信息: k8s-node2的内存使用率		usememory{k8s-node2}	60s	90d	365d	可计算的	node	已启用	
***	获取所有pod信息: pod_coredns-fb8b8dcf-9fg2f的状态	触发器 1	pod{coredns-fb8b8dcf-9fg2f.status}	90d			Zabbix采集器	pod	已启用	
***	获取所有pod信息: pod_coredns-fb8b8dcf-9fg2f的运行时间		pod{coredns-fb8b8dcf-9fg2f.runtime}	90d			Zabbix采集器	pod	已启用	
	获取所有pod信息: pod_coredns-fb8b8dcf-9fg2f的重启次数		pod{coredns-fb8b8dcf-9fg2f.restartcount}	90d	365d		Zabbix采集器	pod	已启用	
***	获取所有pod信息: pod_coredns-fb8b8dcf-4lng的状态	触发器 1	pod{coredns-fb8b8dcf-4lng.status}	90d			Zabbix采集器	pod	已启用	
***	获取所有pod信息: pod_coredns-fb8b8dcf-4lng的运行时间		pod{coredns-fb8b8dcf-4lng.runtime}	90d			Zabbix采集器	pod	已启用	
	获取所有pod信息: pod_coredns-fb8b8dcf-4lng的重启次数		pod{coredns-fb8b8dcf-4lng.restartcount}	90d	365d		Zabbix采集器	pod	已启用	
***	获取所有pod信息: pod_dashboard-metrics-scraper-677768c755-m6gc的状态	触发器 1	pod{dashboard-metrics-scraper-677768c755-m6gc.status}	90d			Zabbix采集器	pod	已启用	
***	获取所有pod信息: pod_dashboard-metrics-scraper-677768c755-m6gc的运行时间		pod{dashboard-metrics-scraper-677768c755-m6gc.runtime}	90d			Zabbix采集器	pod	已启用	
	获取所有pod信息: pod_dashboard-metrics-scraper-677768c755-m6gc的重启次数		pod{dashboard-metrics-scraper-677768c755-m6gc.restartcount}	90d	365d		Zabbix采集器	pod	已启用	

## Zabbix 通过 api 获取 k8s 结果成功