

Monitor自定义脚本设计

🔖 2.monitor

概述

本文解释如何在优云Monitor上扩展自定义采集能力，运维人员可以按文中定义的脚本规范，编写各类脚本，快速实现指标与事件的采集。

基本规范

支持脚本类型

优云Monitor系统支持以下脚本，可以根据自己的需要选择适合的语言来编写采集脚本。

脚本语法类型	操作平台
shell	Linux
python	Windows/Linux
vbs	Windows
bat	Windows

脚本存储路径

编写脚本后，可以通过Monitor控制台下发，也可以直接保存到agent/scripts目录，来安装扩展脚本，安装后，agent即可主动调用以采集数据并上报。

脚本内容结构

无论采用哪种脚本类型，脚本的结构分为以下三个部份：

元信息

根据需要，可以在采集脚本头部增加一段注释，提供一些脚本运行的默认配置信息，如默认采样周期等。

元信息可包含以下字段：

编码	名称	说明	举例
	脚本	代替脚本文件名提供一个友好的名字，默认	name=采集CPU

name 编码	名称	说明	为脚本文件名	使用率 举例
interval	采样 周期		单位秒，默认为60s	interval=15
version	脚本 版本	用于判断脚本相关文件名的版本差异，默认 为1.0.0		version=1.0.0

如果想全部使用默认值，则可以不提供元信息部份。

采集过程

这里根据脚本的采集目标，编写具体的采集过程、解析采集结果并保存到变量中，准备输出为采集结果。

输出过程

将采集解析的结果，用简单的格式作标准输出，即可作为采集结果上报。

根据不同的采集结果类型，我们需要使用不同的输出格式。

输出性能指标

性能指标输出内容为一个字符串，使用 “|” 分隔的字段组成，字段内容如下：

字段	可选	说明	举例
metric	必选	性能指标编码	metric=system.cpu.usage
value	必选	性能指标值	value=23
type	可选	指标值类型	type=gauge
tags	可选	标签集合	tags=app:Biz1

如下所示：

```
metric=system.logfile.number|value=75|type=count|tag=app:WebPortal
```

输出事件

事件输出内容为一个字符串，使用 “|” 分隔的字段组成，字段内容如下：

字段	可选	说明	举例
event	必选	事件标识	event=PortShutdownEvent
msg	必选	事件内容	msg=端口eth0下线
severity	可选	事件状态，默认级别warning	severity=error

tags 字段	可选	说明	标签集合	tags=app:Biz1 举例
------------	----	----	------	---------------------

如下所示：

```
event=PortShutdownEvent|msg=端口eth0下线|severity=error|tags=app:Biz1
```

组合输出

当一个脚本需要输出多个结果，如2个性能指标、1个事件时，可以使用换行作为分隔符，如下所示：

脚本样例

Linux日志文件数量与大小指标采集

本样例要采集/var/log目录文件数与大小，分别作为system.logfile.number与system.logfile.size指标上报。

以下是采集脚本（shell格式）：

```
#!/bin/bash

# 脚本元信息
# name=采集日志文件数量与大小
# interval=600

# 采集过程
NUM=`ls -lR /var/log | grep "^-" | wc -l`
SIZE=`du -s /var/log | awk '{print $1}'`

# 输出结果
echo "metric=system.logfile.number|value=$NUM|type=count|tag=app:WebPortal"
echo "metric=system.logfile.size|value=$SIZE|tag=app:WebPortal"
```

针对上述脚本，执行输出后，可以得到结果如下：

```
metric=system.logfile.number|value=76|type=count|tag=app:WebPortal
metric=system.logfile.size|value=22196|tag=app:WebPortal
```

这表示采集到了两个指标，并且用指定的指标类型与标签上报。

Linux启动错误事件采集

本样例将访问系统的启动文件，如果日志文件中出现FAILED行，则将其所在行作为事件上

报。

以下是采集脚本（shell格式）：

```
#!/bin/bash

# 脚本元信息
# name=采集错误事件
# interval=600

# 采集过程
cat /var/log/boot.log | grep FAILED | while read line
do
    # 输出结果
    echo "event=SystemBootFailed|msg=$line|severity=error|tag=app:WebPortal"
done
```

针对上述脚本，执行输出后，可以得到结果如下：

```
event=SystemBootFailed|msg=[FAILED] Failed to start Docker Storage Setup.|severity=error|tag=app:WebPortal
event=SystemBootFailed|msg=[FAILED] Failed to start Network Manager Script Dispatcher Service.|severity=error|tag=app:WebPortal
```

Windows进程性能指标采集

本样例将通过WMI访问Windows本地的进程列表，采集进程的线程数、句柄数等指标。

```
rem 脚本元信息
rem name=采集进程线程数与句柄数
rem interval=60

rem 准备wmi连接与执行采集操作
Set objWMIService = GetObject("winmgmts:\\.\root\cimv2")
Set collItems = objWMIService.ExecQuery("Select * from Win32_Process",,48)

For Each objItem in collItems
    rem 输出结果
    wscript.echo "metric=system.process.thread.count|value=" & objItem.ThreadCount & "|tag=process:" & objItem.Name
    wscript.echo "metric=system.process.handle.count|value=" & objItem.HandleCount & "|tag=process:" & objItem.Name
Next
```

针对上述脚本，执行输出后，可以得到结果如下：

```
metric=system.process.thread.count|value=49|tag=process:OUTLOOK.EXE  
metric=system.process.handle.count|value=1930|tag=process:OUTLOOK.EXE  
metric=system.process.thread.count|value=30|tag=process:chrome.exe  
metric=system.process.handle.count|value=2294|tag=process:chrome.exe  
...
```