

# 在CPS中修改参数值的过程

## 目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[问题](#)

[修改参数值的过程](#)

## 简介

本文档介绍在思科策略套件(CPS)副本集中修改参数值的过程。

## 先决条件

### 要求

Cisco 建议您了解以下主题：

- Linux
- CPS
- 木偶

思科建议您必须拥有对CPS CLI的根访问权限。

### 使用的组件

本文档中的信息基于以下软件和硬件版本：

- CPS 20.2
- MongoDB v3.6.17
- UCS-B

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

## 背景信息

Facter是Puppet的跨平台系统配置文件库。它会发现并报告每个节点的事实，这些事实在Puppet清单中作为变量可用。Puppet支持将多个值作为环境变量进行保留。Puppet使用更快速度支持此功能。在Puppet中，facter是保存环境级变量的独立工具。它可视为类似于Bash或Linux的env变量。有时，存储在事实中的信息与机器的环境变量之间可能存在重叠。在Puppet中，密钥值对称为“事实”。每个资源都有自己的事实，在Puppet中，用户有能力构建自己的自定义事实。

命令因子可用于列出所有不同的环境变量及其关联值。

因素对Puppet很重要的原因是，因素和事实在整个Puppet代码中都可用为全局变量，这意味着它们可以在任何时间点代码中使用，而不需要任何其他引用。

## 问题

假设CPS脚本使用因子参数值以设置警报生成的阈值，如果要根据网络微调阈值，则必须相应地修改各自的因子参数值。

考虑示例CPS脚本`/var/qps/install/20.2.0/scripts/bin/support/snmp-traps/process-traps/gen-gx-drop-trap.sh`和警报Gx平均消息CCR-X处理增加。

如您所见，如果已配置，脚本首先使用`facter`值。

```
# If threshold configured in facter, get that value, else use default 20ms
TRAP_AVG_LEVEL=200
```

因此，此处脚本不考虑`TRAP_AVG_LEVEL=200`提供的输入，而是考虑因子值`20ms`。

从`pcrfclient`运行此命令以显示相应的`facter`参数值。

```
#facter | grep ccr
gx_alarm_ccr_i_avg_threshold => 20
gx_alarm_ccr_t_avg_threshold => 20
gx_alarm_ccr_u_avg_threshold => 20
```

## 修改参数值的过程

以下是将参数值修改为200毫秒的步骤。

1.在OpenStack中托管的CPS方法。

步骤1.备份当前因素配置。

从群集管理器运行此命令(确保目录`/mnt/backup`存在)。

```
# curl -X GET http://installer:8458/api/system/config/config/ -o /mnt/backup/ facter-
config_$(date +%Y-%m-%d).yaml
```

步骤2.使用适当的阈值参数值准备YAML文件。在ClusterManager中准备`.yaml`文件。

```
# vi gx_alarm_threshold.yaml
gxAlarmCcrIAvgThreshold: "200"
gxAlarmCcrUAvgThreshold: "200"
gxAlarmCcrTAvgThreshold: "200"
```

步骤3.从集群管理器运行此命令以更新阈值参数。

注意：此命令必须从`gx_alarm_threshold.yaml`文件所在的目录执行。这里是`/tmp`。

```
[root@installer tmp]# curl -i -X PATCH http://installer:8458/api/system/config/config -H
"Content-Type: application/yaml" --data-binary "@gx_alarm_threshold.yaml"
HTTP/1.1 200 OK
Date: Fri, 21 Jan 2022 06:45:20 GMT
Content-Length: 0
```

```
[root@installer tmp]
```

步骤4.检验阈值参数值的更改。从pcrfclient运行此命令。

```
#factor | grep ccr
```

```
Expected Output:
```

```
gx_alarm_ccr_i_avg_threshold => 200
gx_alarm_ccr_t_avg_threshold => 200
gx_alarm_ccr_u_avg_threshold => 200
```

2. VMWare中托管的CPS方法。

步骤1:登录ClusterManager并修改/var/qps/config/deploy/csv/Configuration.csv中必需参数的值。请考虑这些示例参数，当前值为20。

```
[root@installer ~]# cat /var/qps/config/deploy/csv/Configuration.csv | grep ccr
gx_alarm_ccr_i_avg_threshold,20,
gx_alarm_ccr_t_avg_threshold,20,
gx_alarm_ccr_u_avg_threshold,20,
[root@installer ~]#
```

运行此命令并修改Configuration.csv文件中的示例参数值。

```
[root@installer ~]#vi /var/qps/config/deploy/csv/Configuration.csv
```

运行此命令以检查Configuration.csv文件中的值是否更改。

```
[root@installer ~]# cat /var/qps/config/deploy/csv/Configuration.csv|grep ccr
gx_alarm_ccr_i_avg_threshold,200,
gx_alarm_ccr_t_avg_threshold,200,
gx_alarm_ccr_u_avg_threshold,200,
[root@installer ~]#
```

步骤2.运行此命令将新配置导入ClusterManager。

```
[root@installer ~]# /var/qps/install/current/scripts/import/import_deploy.sh
```

步骤3.运行此命令以验证ClustManager因子值的更改。

```
[root@installer ~]# factor | grep -i ccr
gx_alarm_ccr_i_avg_threshold => 200
gx_alarm_ccr_t_avg_threshold => 200
gx_alarm_ccr_u_avg_threshold => 200
[root@installer ~]#
```

步骤4.运行此命令以重建CPS包。

```
[root@installer ~]# /var/qps/install/current/scripts/build_all.sh
```

步骤5.运行此命令以下载所有Puppet脚本、CPS软件和/etc/hosts文件，并使用群集管理器中的新软

件更新每个虚拟机。

```
[root@installer ~]# /var/qps/install/current/scripts/upgrade/reinit.sh
```

步骤6. 登录pcrfclient并运行此命令以验证因子值的更改。

```
[root@dc1-pcrfclient01 ~]# factor | grep ccr
gx_alarm_ccr_i_avg_threshold => 200
gx_alarm_ccr_t_avg_threshold => 200
gx_alarm_ccr_u_avg_threshold => 200
[root@dc1-pcrfclient01 ~]#
```