

在CPS中修改因子引數值的過程

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[背景資訊](#)

[問題](#)

[修改因子引數值的過程](#)

簡介

本文檔介紹在思科策略套件(CPS)副本集中修改因子引數值的過程。

必要條件

需求

思科建議您瞭解以下主題：

- Linux
- CPS
- 木偶

思科建議您必須具有對CPS CLI的Root訪問許可權。

採用元件

本文中的資訊係根據以下軟體和硬體版本：

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

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

背景資訊

Facter是Puppet的跨平台系統配置檔案庫。它發現並報告每個節點的事實，這些事實在Puppet清單中作為變數提供。Puppet支援將多個值保持為環境變數。Puppet中通過使用更快的速度支援此功能。在Puppet中，facter是儲存環境級別變數的獨立工具。可以認為它類似於Bash或Linux的env變數。有時，儲存在事實中的資訊與機器的環境變數之間可能存在重疊。在Puppet中，key-value對稱為「fact」。每個資源都有自己的事實，在Puppet中，使用者可以利用這些事實構建自己的自定義事實。

命令因子可用於列出所有不同的環境變數及其相關值。

因子對於Puppet很重要的原因在於，因子和事實在整個Puppet代碼中作為一個全域性變數是可用的，這意味著它們可以在代碼中的任何時間點使用，而無需任何其他引用。

問題

假定CPS指令碼在某些用途上使用因子引數值，例如設定用於生成警報的閾值，如果要根據網路微調閾值，則必須相應地修改各自的因子引數值。

考慮示例CPS指令碼`/var/qps/install/20.2.0/scripts/bin/support/snmp-traps/process-traps/gen-gx-drop-trap.sh`和alert `Gx average Message CCR-X processing increased`。

您可以看到，如果進行了配置，指令碼首先會使用因子值。

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

因此，在這裡，指令碼不考慮提供的輸入`TRAP_AVG_LEVEL=200`，而是考慮因子值`20ms`。

從`pcrfclient`運行此命令以顯示相應的因子引數值。

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

修改因子引數值的過程

以下是將`facter`引數值修改為`200ms`的步驟。

1.在OpenStack中託管CPS的方法。

步驟1.備份當前工廠配置。

從群集管理器運行此命令(確保目錄`/mnt/backup`存在)。

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

步驟2.使用正確的閾值引數值準備YAML檔案。在ClusterManager中準備`yaml`檔案。

```
# vi gx_alarm_threshold.yaml
gxAlarmCcrIAvgThreshold: "200"
gxAlarmCcrUAvGThreshold: "200"
gxAlarmCcrTAvGThreshold: "200"
```

步驟3.從群集管理器運行此命令以更新閾值引數。

附註：此命令必須從放置`gx_alarm_threshold.yaml`檔案的同一目錄執行。這裡是`/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]
```

步驟4. 檢驗閾值引數值的變化。從pcrfclient運行此命令。

```
#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. 在VMWare中託管的CPS的方法。

步驟1。 登入到ClusterManager並修改/var/qps/config/deploy/csv/Configuration.csv中必需的引數的值。請考慮使用以下當前值為20的示例引數。

```
[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 ~]#
```

運行此命令並修改Configuration.csv檔案中的示例引數值。

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

運行此命令以檢查Configuration.csv檔案中是否更改了值。

```
[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 ~]#
```

步驟2. 運行此命令以將新配置匯入到ClusterManager。

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

步驟3. 運行此命令以驗證ClusterManager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 ~]#
```

步驟4. 運行此命令重建CPS包。

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

步驟5. 運行此命令可下載所有Puppet指令碼、CPS軟體/etc/hosts文件，並使用Cluster Manager中

的新軟體更新每台VM。

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

步驟6.登入到pcrfclient並運行此命令以驗證因子值的更改。

```
[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 ~]#
```