

# 了解5G SA应用上的show resources CLI

## 目录

[简介](#)

[运行原理](#)

[输出说明](#)

[Go programming语言的其他说明](#)

## 简介

本文档介绍如何了解Cisco 5G SA应用上show resources CLI的输出。

## 运行原理

show resources CLI列出集群中所有Pod ( 主要是协议和服务Pod ) 的资源信息。资源信息定期从每个Pod收集并呈现给CLI。

示例输出：

```
[unknown] smf# show resources
TOTAL USED DISK
NODE POD USAGE GO GC
CPU MEMORY MEMORY IN ROUTINES GC PAUSE
POD INSTANCE USAGE IN MB IN MB KBPS COUNT COUNT IN NS
-----
cache-pod-1 2 32011 85 0 172 29684 85
cache-pod-2 2 32011 83 0 172 29627 83
gtpc-ep-0 2 32011 70 0 69 29088 70
li-ep-0 5 32011 51 0 56 29095 51
oam-pod-1 2 32011 98 0 157 29095 98
smf-nodemgr-0 7 32011 94 0 213 29096 94
smf-protocol-0 3 32011 135 0 206 29092 135
smf-rest-ep-0 4 32011 125 0 203 29091 125
smf-service-0 3 32011 321 0 247 34958 321
smf-udp-proxy-0 1 32011 82 0 88 29083 82
```

## 输出说明

POD实例	Pod名称
CPU使用率CPU	Pod的CPU使用率
总节点内存(MB)	运行Pod的工作节点的总内存
已用POD内存(MB)	Pod使用的内存
磁盘使用率(KBPS)	磁盘I/O速率 ( 以Kbps为单位 )
GO例程计数	执行例程数
GC计数	垃圾收集例程计数

## Go programming语言的其他说明

Go例程是Go编程语言中的“线程”。它用于某些目的，如更好的性能等。在有问题的场景中，Go例程泄漏可以看到Go例程未正确完成且新Go例程不断创建。Go Routine泄漏的影响类似于内存泄漏。这可以通过查看show resources输出中的GO ROUTINES COUNT来识别。如果观察到剧增或持续增加，则有可能出现Go Routine泄漏。

垃圾回收是内置的内存管理功能。它会持续运行并清除不再用于保存内存使用情况的对象。NS中的GC计数和GC暂停可用于监控垃圾回收是否能正常工作而没有任何问题。例如，特定Pod的GC计数过多或GC暂停过长可能表示存在某些问题。