

# Verständnis der Kommandozeile für Ressourcen auf 5G SA-Anwendung

## Inhalt

[Einleitung](#)

[So funktioniert es](#)

[Beschreibung der Ausgabe](#)

[Zusätzliche Erklärung für Go-Programming-Sprache](#)

## Einleitung

In diesem Dokument wird beschrieben, wie die Ausgabe der CLI für **Anzeigeressourcen** in der Anwendung Cisco 5G SA verstanden wird.

## So funktioniert es

Die CLI **show resources** listet Ressourceninformationen für alle PODs (hauptsächlich Protokolle und Service-PODs) im Cluster auf. Die Ressourceninformationen werden regelmäßig von jedem POD erfasst und der CLI angezeigt.

Beispiel für das Ergebnis:

```
[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
```

## Beschreibung der Ausgabe

POD-INSTANZ	Pod-Name
CPU-NUTZUNG	CPU-Auslastung des POD
KNOTENSPEICHER INSGESAMT IN MB	Gesamtspeicher des Arbeitsknotens, in dem der ausgeführt wird
VERWENDETER POD-SPEICHER IN MB	Von POD verwendeter Arbeitsspeicher
DATENTRÄGERNUTZUNG IN KBPS	Festplatten-E/A-Rate in Kbit/s

ROUTINENANZAHL

Anzahl der Go-Routine

GC-ANZAHL

Garbage Collection Routinganzahl

GC PAUSE IN NS

Garbage Collection Routine-Pause in NanoSecor

## Zusätzliche Erklärung für Go-Programming-Sprache

Go Routine ist eine Art "Thread" in Go-Programmiersprache. Es wird für einen bestimmten Zweck verwendet, z. B. für eine bessere Leistung usw. In einem problematischen Szenario kann man Go Routine-Leck sehen, wo Go Routines nicht richtig fertig sind und neue Go Routines weiter erstellen. Die Auswirkungen von Go Routine-Lecks sind ähnlich wie Speicherlecks. Dies lässt sich ermitteln, indem Sie die ROUTINES COUNT von GO in der Ausgabe der Ressourcen betrachten. Wenn eine dramatische oder konstante Zunahme beobachtet wird, besteht die Möglichkeit eines Go Routine-Lecks.

Garbage Collection ist eine integrierte Speicherverwaltungsfunktion für Go. Es läuft kontinuierlich und bereinigt Objekte, die nicht mehr zum Speichern der Speichernutzung verwendet werden. GC COUNT und GC PAUSE IN NS können zur Überwachung verwendet werden, ob Garbage Collection ohne Probleme funktioniert. Beispielsweise kann eine zu hohe Anzahl an GCs für bestimmte PoD oder eine zu lange Pause bei GC auf ein Problem hinweisen.