



Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'podman-stats.1' command

\$ man podman-stats.1

podman-stats(1) General Commands Manual podman-stats(1)

NAME

podman-stats - Display a live stream of one or more container's resource usage statistics

SYNOPSIS

podman stats [options] [container]
podman container stats [options] [container]

DESCRIPTION

Display a live stream of one or more containers' resource usage statistics

Note: Podman stats will not work in rootless environments that use CGroups V1. Podman stats relies on CGroup information for statistics, and CGroup v1 is not supported for rootless use cases.

Note: Rootless environments that use CGroups V2 are not able to report statistics about their networking usage.

OPTIONS

--all, -a

Show all containers. Only running containers are shown by default

--format=template

Pretty-print container statistics to JSON or using a Go template

Valid placeholders for the Go template are listed below:

??

?Placeholder	? Description	?
--------------	---------------	---

??

?.AvgCPU ? Average CPU, full preci? ?

? ? sion float ?

??

?.AVGCPU ? Average CPU, formatted as ?

? ? a percent ?

??

?.BlockInput ? Block Input ?

??

?.BlockIO ? Block IO ?

??

?.BlockOutput ? Block Output ?

??

?.ContainerID ? Container ID, full (un? ?

? ? truncated) hash ?

??

?.ContainerStats ... ? Nested structure, for ex? ?

? ? perts only ?

??

?.CPU ? Percent CPU, full preci? ?

? ? sion float ?

??

?.CPUNano ? CPU Usage, total, in ?

? ? nanoseconds ?

??

?.CPUPerc ? CPU percentage ?

??

?.CPUSystemNano ? CPU Usage, kernel, in ?

? ? nanoseconds ?

??

?.Duration ? Same as CPUNano ?

??

?.ID ? Container ID, truncated ?

??

?.MemLimit ? Memory limit, in bytes ?

??

?.MemPerc ? Memory percentage ?

??

?.MemUsage ? Memory usage ?

??

?.MemUsageBytes ? Memory usage (IEC) ?

??

?.Name ? Container Name ?

??

?.NetInput ? Network Input ?

??

?.NetIO ? Network IO ?

??

?.NetOutput ? Network Output ?

??

?.PerCPU ? CPU time consumed by all ?

? ? tasks [1] ?

??

?.PIDs ? Number of PIDs ?

??

?.PIDS ? Number of PIDs (yes, we ?

? ? know it's a dup) ?

??

?.SystemNano ? Current system datetime, ?

? ? nanoseconds since epoch ?

??

?.Up ? Duration (CPUNano), in hu? ?

? ? man-readable form ?

??

?.UpTime ? Same as UpTime ?

??

[1] Cgroups V1 only

When using a Go template, precede the format with table to print head?

ers.

`--interval, -i=seconds`

Time in seconds between stats reports, defaults to 5 seconds.

`--latest, -l`

Instead of providing the container name or ID, use the last created container. Note: the last started container could be from other users of Podman on the host machine. (This option is not available with the remote Podman client, including Mac and Windows (excluding WSL2 machines))

`--no-reset`

Do not clear the terminal/screen in between reporting intervals

`--no-stream`

Disable streaming stats and only pull the first result, default setting is false

`--no-trunc`

Do not truncate output

EXAMPLE

```
# podman stats -a --no-stream
```

```
ID      NAME      CPU % MEM USAGE / LIMIT MEM % NET IO  BLOCK IO PIDS
a9f807ffaacd frosty_hodgkin -- 3.092MB / 16.7GB 0.02% --/-- --/-- 2
3b33001239ee sleepy_stallman -- --/-- -- --/-- --
```

```
# podman stats --no-stream a9f80
```

```
ID      NAME      CPU % MEM USAGE / LIMIT MEM % NET IO  BLOCK IO PIDS
a9f807ffaacd frosty_hodgkin -- 3.092MB / 16.7GB 0.02% --/-- --/-- 2
```

```
$ podman stats --no-trunc 3667 --format 'table {{ .ID }} {{ .MemUsage }}'
```

```
ID      MEM USAGE / LIMIT
3667c6aacb06aac2eaffce914c01736420023d56ef9b0f4cfe58b6d6a78b7503 49.15kB / 67.17GB
```

```
# podman stats --no-stream --format=json a9f80
```

```
[
```

```
{
```

```
"id": "a9f807ffaacd",
```

```

    "name": "frosty_hodgkin",
    "cpu_percent": "--",
    "mem_usage": "3.092MB / 16.7GB",
    "mem_percent": "0.02%",
    "netio": "-- / --",
    "blocki": "-- / --",
    "pids": "2"
  }
]
# podman stats --no-stream --format "table {{.ID}} {{.Name}} {{.MemUsage}}" 6eae
ID      NAME      MEM USAGE / LIMIT
6eae9e25a564  clever_bassi  3.031MB / 16.7GB

```

Note: When using a slirp4netns network with the rootlesskit port handler, the traffic sent via the port forwarding will be accounted to the lo device. Traffic accounted to lo is not accounted in the stats output.

SEE ALSO

podman(1)

HISTORY

July 2017, Originally compiled by Ryan Cole rycole@redhat.com
[?mailto:rycole@redhat.com?](mailto:rycole@redhat.com)

podman-stats(1)