





Authored by: Lenesto Page (/author/lenesto.page)

 2  2  Share  PDF (<https://success.docker.com/api/articles/node-using-swap-memory-instead-of-host-memory/pdf>)

Node using swap memory instead of host memory

Article ID: KB000736

EE LINUX MANAGEMENT MONITORING EE-17.06.1-EE-2 UCP-2.2.3

Issue

By default, Docker recommends using a value of `vm.swappiness=0` for Docker environments, which prevents swapping except in the case of an OOM (OutOfMemory) condition. All nodes must set `vm.overcommit_memory=1`, which tells the kernel to always allow memory allocations until there is no truly memory. This article explains a situation that can occur whenever a value other `0` is used for `vm.swappiness`.

Root Cause

If `vm.swappiness` is set to a value higher than `0`, you might notice that only swap memory is being used on the node even though host memory was available.

Resolution

To successfully set `vm.swappiness=0` and `vm.overcommit_memory=1` use these steps:

1. Verify the current values for `vm.swappiness` and `vm.overcommit_memory`:

```
$ sudo sysctl -a |grep 'vm.swapp*'|vm.over*
```

2. Change the value of `vm.swappiness`:

```
$ sudo sysctl vm.swappiness=0
```

3. Change the value of `vm.overcommit_memory`:

```
$ sudo sysctl vm.overcommit_memory=1
```

4. Verify the settings have changed two ways:

```
$ sudo sysctl -a |grep 'vm.swapp*'|vm.over*
```

Also check the `/proc/sys/vm` file:

```
$ cat /proc/sys/vm/overcommit_memory
$ cat /proc/sys/vm/swappiness
```

What is Docker (<https://www.docker.com/what-docker>)

What is a Container (<https://www.docker.com/what-container>)

Use Cases (<https://www.docker.com/use-cases>)

Customers (<https://www.docker.com/customers>)

For Government (<https://www.docker.com/industry-government>)

For IT Pros (<https://www.docker.com/itpro>)

Find a Partner (<https://www.docker.com/find-partner>)

Become a Partner (<https://www.docker.com/partners/partner-program>)

About Docker (<https://www.docker.com/company>)

Management (<https://www.docker.com/company/management>)

Press & News (<https://www.docker.com/company/news-and-press>)

Careers (<https://www.docker.com/careers>)

Product (<https://www.docker.com/get-docker>)

Pricing (<https://www.docker.com/pricing>)

Community Edition (<https://www.docker.com/community-edition>)

Enterprise Edition (<https://www.docker.com/enterprise-edition>)

Docker Datacenter (https://www.docker.com/enterprise-edition#container_management)

- Docker Cloud (<https://cloud.docker.com/>)
- Docker Store (<https://store.docker.com/>)
- Get Docker (<https://www.docker.com/get-docker>)
- Docker for Mac (<https://www.docker.com/docker-mac>)
- Docker for Windows(PC) (<https://www.docker.com/docker-windows>)
- Docker for AWS (<https://www.docker.com/docker-aws>)
- Docker for Azure (<https://www.docker.com/docker-azure>)
- Docker for Windows Server (<https://www.docker.com/docker-windows-server>)
- Docker for Debian (<https://www.docker.com/docker-debian>)
- Docker for Fedora® (<https://www.docker.com/docker-fedora>)
- Docker for Oracle Linux (<https://www.docker.com/docker-oracle-linux>)
- Docker for RHEL (<https://www.docker.com/docker-red-hat-enterprise-linux-rhel>)
- Docker for SLES (<https://www.docker.com/docker-suse-linux-enterprise-server-sles>)
- Docker for Ubuntu (<https://www.docker.com/docker-ubuntu>)
- Documentation (<https://docs.docker.com/>)
- Blog (<https://blog.docker.com/>)
- RSS Feed (<https://blog.docker.com/feed/>)
- Training (<https://training.docker.com/>)
- Knowledge Base (<https://success.docker.com/kbase>)
- Resources (<https://www.docker.com/products/resources>)
- Community (<https://www.docker.com/docker-community>)
- Open Source (<https://www.docker.com/technologies/overview>)
- Events (<https://www.docker.com/community/events>)
- Forums (<https://forums.docker.com/>)
- Docker Captains (<https://www.docker.com/community/docker-captains>)
- Scholarships (<https://www.docker.com/community-partnerships>)
- Community News (<https://blog.docker.com/curated/>)