Check memory pressure inside a Linux VPS

Linux uses spare RAM for caches, so a small free-memory figure is not enough to diagnose exhaustion. This guide checks memory visible inside a Debian or Ubuntu VPS. It cannot prove the physical hypervisor's available RAM. You need a shell in the affected guest, and administrative access may be required to read kernel logs or other users' processes.

Capture the current picture

```
date -Is
free -h
swapon --show
vmstat 1 10
```

In `free`, start with the available estimate rather than only the free column. Available memory includes an estimate of reclaimable capacity. Swap use by itself is historical state: pages can remain swapped out after pressure has passed. In `vmstat`, inspect repeated swap-in and swap-out activity alongside CPU and I/O counters. The first report can cover a different interval from subsequent samples.

Do not disable swap to make a dashboard look cleaner. Moving all swapped pages back into limited RAM can worsen pressure. Adding swap also does not replace a suitable application memory limit or fix an overloaded physical host. Changes to swap or kernel tuning need their own reviewed plan.

Find likely consumers

```
ps -eo pid,user,comm,rss --sort=-rss | head -n 15
systemctl status example.service --no-pager
journalctl -k --since "2 hours ago" --no-pager
```

RSS gives a process-level resident-memory view, but summing it can count shared pages more than once. Identify services and their expected workload before treating a large process as faulty. Databases and Java applications may intentionally reserve substantial memory; their heap or buffer

setting must still leave room for the operating system and other processes.

Look for a kernel out-of-memory event around the reported failure. A matching victim process and timestamp strengthen the case for guest memory exhaustion. Absence of a message does not prove adequate host capacity: abrupt VM power loss, limited log retention or unavailable kernel access can leave gaps.

Check limits below the VM size

Containers and systemd services can have their own memory ceilings. Inspect the relevant container configuration or `systemctl show example.service -p MemoryMax -p MemoryHigh`. A process may hit that boundary while the guest still has available RAM. Tool output and cgroup features vary between operating-system releases, so compare with the installed systemd documentation.

Collect an observation during the problem and a quiet period. Note player count, scheduled jobs, backups or a recent application update. A short-lived spike needs different treatment from steadily growing use. Avoid restarting every service together: doing so destroys useful evidence and can cause a larger startup peak.

Choose a measured response

Adjust only a documented application setting with a saved previous value and a maintenance window. Verify the service and memory trend afterward. If pressure is outside the guest or several unrelated VMs fail together, provide support with the UTC interval and logs for host correlation. Keep secrets out of process arguments and screenshots before sharing them. No host-level conclusion should rest on the guest's free-memory figure alone.

Related instructions

Investigate systemd services. Find disk-space use.

Reference: Official documentation. Check the documentation installed on your own operating-system release when an option differs.

Revision #1
Created 2026-11-09 10:42:42 UTC by CloudExa Hosting Admin
Updated 2026-11-09 10:42:42 UTC by CloudExa Hosting Admin