NAME

systemd-oomd.service, systemd-oomd − A userspace out−of−memory (OOM) killer

SYNOPSIS

systemd−oomd.service

/lib/systemd/systemd−oomd

DESCRIPTION

systemd−oomd is a system service that uses cgroups−v2 and pressure stall information (PSI) to monitor and take action on processes before an OOM occurs in kernel space.

You can enable monitoring and actions on units by setting *ManagedOOMSwap=* and/or *ManagedOOMMemoryPressure=* to the appropriate value. **systemd−oomd** will periodically poll enabled units' cgroup data to detect when corrective action needs to occur. When an action needs to happen, it will only be performed on the descendant cgroups of the enabled units. More precisely, only cgroups with memory.oom.group set to **1** and leaf cgroup nodes are eligible candidates. Action will be taken recursively on all of the processes under the chosen candidate.

See **oomd.conf**(5) for more information about the configuration of this service.

SETUP INFORMATION

The system must be running systemd with a full unified cgroup hierarchy for the expected cgroups−v2 features. Furthermore, memory accounting must be turned on for all units monitored by **systemd−oomd**. The easiest way to turn on memory accounting is by ensuring the value for *DefaultMemoryAccounting=* is set to **true** in **systemd-system.conf**(5).

You will need a kernel compiled with PSI support. This is available in Linux 4.20 and above.

It is highly recommended for the system to have swap enabled for **systemd−oomd** to function optimally. With swap enabled, the system spends enough time swapping pages to let **systemd−oomd** react. Without swap, the system enters a livelocked state much more quickly and may prevent **systemd−oomd** from responding in a reasonable amount of time. See **"In defence of swap: common misconceptions"**[1] for more details on swap. Any swap−based actions on systems without swap will be ignored. While **systemd−oomd** can perform pressure−based actions on a system without swap, the pressure increases will be more abrupt and may require more tuning to get the desired thresholds and behavior.

Be aware that if you intend to enable monitoring and actions on user.slice, user−$UID.slice, or their ancestor cgroups, it is highly recommended that your programs be managed by the systemd user manager to prevent running too many processes under the same session scope (and thus avoid a situation where memory intensive tasks trigger **systemd−oomd** to kill everything under the cgroup). If you're using a desktop environment like GNOME, it already spawns many session components with the systemd user manager.

USAGE RECOMMENDATIONS

ManagedOOMSwap= works with the system−wide swap values, so setting it on the root slice −.slice, and allowing all descendant cgroups to be eligible candidates may make the most sense.

ManagedOOMMemoryPressure= tends to work better on the cgroups below the root slice −.slice. For units which tend to have processes that are less latency sensitive (e.g. system.slice), a higher limit like the default of 60% may be acceptable, as those processes can usually ride out slowdowns caused by lack of memory without serious consequences. However, something like user@$UID.service may prefer a much lower value like 40%.

SEE ALSO

systemd(1), **systemd-system.conf**(5), **systemd.resource-control**(5), **oomd.conf**(5), **oomctl**(1)

NOTES

1. "In defence of swap: common misconceptions"
   https://chrisdown.name/2018/01/02/in-defence-of-swap.html