

**NAME**

systemd-machined.service, systemd-machined – Virtual machine and container registration manager

**SYNOPSIS**

systemd-machined.service

/lib/systemd/systemd-machined

**DESCRIPTION**

**systemd-machined** is a system service that keeps track of locally running virtual machines and containers.

**systemd-machined** is useful for registering and keeping track of both OS containers (containers that share the host kernel but run a full init system of their own and behave in most regards like a full virtual operating system rather than just one virtualized app) and full virtual machines (virtualized hardware running normal operating systems and possibly different kernels).

**systemd-machined** should *not* be used for registering/keeping track of application sandbox containers. A *machine* in the context of **systemd-machined** is supposed to be an abstract term covering both OS containers and full virtual machines, but not application sandboxes.

Machines registered with machined are exposed in various ways in the system. For example:

- Tools like **ps**(1) will show to which machine a specific process belongs in a column of its own, and so will **gnome-system-monitor**<sup>[1]</sup> or **systemd-cgls**(1).
- systemd's various tools (**systemctl**(1), **journaltctl**(1), **loginctl**(1), **hostnamectl**(1), **timedatectl**(1), **localectl**(1), **machinectl**(1), ...) support the **-M** switch to operate on local containers instead of the host system.
- **systemctl list-machines** will show the system state of all local containers, connecting to the container's init system for that.
- systemctl's **—recursive** switch has the effect of not only showing the locally running services, but recursively showing the services of all registered containers.
- The **machinectl** command provides access to a number of useful operations on registered containers, such as introspecting them, rebooting, shutting them down, and getting a login prompt on them.
- The **sd-bus**(3) library exposes the **sd\_bus\_open\_system\_machine**(3) call to connect to the system bus of any registered container.
- The **nss-mymachines**(8) module makes sure all registered containers can be resolved via normal glibc **gethostbyname**(3) or **getaddrinfo**(3) calls.

See **systemd-nspawn**(1) for some examples on how to run containers with OS tools.

If you are interested in writing a VM or container manager that makes use of machined, please have look at **Writing Virtual Machine or Container Managers**<sup>[2]</sup>. Also see the **New Control Group Interfaces**<sup>[3]</sup>.

The daemon provides both a C library interface (which is shared with **systemd-logind.service**(8)) as well as a D-Bus interface. The library interface may be used to introspect and watch the state of virtual machines/containers. The bus interface provides the same but in addition may also be used to register or terminate machines. For more information please consult **sd-login**(3) and **org.freedesktop.machine1**(5) and **org.freedesktop.LogControl1**(5).

A small companion daemon **systemd-importd.service**(8) is also available, which implements importing, exporting, and downloading of container and VM images.

For each container registered with systemd-machined.service that employs user namespacing, users/groups are synthesized for the used UIDs/GIDs. These are made available to the system using the **User/Group Record Lookup API via Varlink**<sup>[4]</sup>, and thus may be resolved with **userdbctl**(1) or the usual glibc NSS calls.

## SEE ALSO

**systemd(1)**, **machinectl(1)**, **systemd-nspawn(1)**, **nss-mymachines(8)**, **systemd.special(7)**

## NOTES

1. **gnome-system-monitor**  
<https://help.gnome.org/users/gnome-system-monitor/>
2. **Writing Virtual Machine or Container Managers**  
<https://www.freedesktop.org/wiki/Software/systemd/writing-vm-managers>
3. **New Control Group Interfaces**  
<https://www.freedesktop.org/wiki/Software/systemd/ControlGroupInterface/>
4. **User/Group Record Lookup API via Varlink**  
[https://systemd.io/USER\\_GROUP\\_API](https://systemd.io/USER_GROUP_API)