

 lieter_

 @lieter@mastodon.lieter.nl

pieterlexis 

RUNNING CONTAINERS AND OS IMAGES WITH **systemd-nspawn**

Pieter Lexis

May 4, 2019

Pieter Lexis

- SysAdmin by training, developer by accident
- Senior PowerDNS Engineer at PowerDNS
- Packaging and autotools wizard
- Build and test automation fanatic

systemd-nspawn

WHAT IS `systemd-nspawn`?

DESCRIPTION

`systemd-nspawn` may be used to run a command or OS in a light-weight namespace container. In many ways it is similar to `chroot(1)`, but more powerful since it fully virtualizes the file system hierarchy, as well as the process tree, the various IPC subsystems and the host and domain name.¹

- Included in `systemd`
- Fully integrated *with* `systemd`
- Everything is documented with manpages

¹`systemd-nspawn(8)`

- Image creation etc. considered out of scope
- Images are not layered
- Containers are considered persistent
- No multi-node orchestration
- Processes are directly managed by systemd

- *Fully* virtualizes the filesystem hierarchy
- Sandboxing with cgroups
- `chroot` does not use namespaces or cgroups

- LXC is more low-level

- No kernel is booted in nspawn
- VM introspection not possible from host using KVM

WORKING WITH `systemd-nspawn`

CREATING A CONTAINER

```
debootstrap stretch LOADays-nspawn
systemd-nspawn -D LOADays-nspawn --as-pid2 \
  systemctl enable systemd-networkd
systemd-nspawn -D LOADays-nspawn --as-pid2 \
  passwd root
systemd-nspawn -D LOADays-nspawn -b -n
```

```

A #320 sudo system-resolve 0 10Days-respan 0
Warning: container 10Days-respan on /home/leifer/str/10Days-respan.
Note: 7 Drive Lines either in to kill container.
extended 332 running in system mode. (+RAM-AUGIT +SELINUX +TNG +APPARMOR +GNOME +OVS/JA3T +UTMP +LIBCRYPTSETUP +CRYPT +GUTILS +ACL +XZ +LZA +SECCOMP +BLKID +EUFUTLS +KMOD +I9N)
selected virtualization system-respan.
selected architecture x86-64.

Welcome to Debian GNU/Linux 9 (stretch)!

set hostname to canamas.how.plexis.eu.
failed to install release report, ignoring: No such file or directory
OK | Listening on Journal Socket.
OK | Listening on Journal Socket (/dev/log).
OK | Created slice System Slice.
Mounting Huge Pages File System...
OK | Listening on Syslog Socket.
OK | Started Dispatch Password Requests to Console Directory Watch.
Reached target Remote File Systems.
OK | Created slice system-getty.slice.
Mounting FUSE Control File System...
OK | Reached target Samba.
OK | Started Forward Password Requests to Wall Directory Watch.
OK | Reached target Encrypted Volumes.
Starting Journal Service...
OK | Created slice User and Session Slices.
OK | Reached target Slices.
Starting Remount Root and Kernel File Systems...
OK | Listening on /dev/initctl Compatibility Named Pipe.
OK | Reached target Paths.
OK | Mounted FUSE Control File System.
OK | Mounted Huge Pages File System.
OK | Started Remount Root and Kernel File Systems.
OK | Reached target Local File System (Fds).
OK | Reached target Local File Systems.
Starting Raise Network Interfaces...
OK | Started Journal Service.
Starting Flush Journal to Persistent Storage...
OK | Started Flush Journal to Persistent Storage.
Starting Create Volatile Files and Directories...
OK | Started Raise network interfaces.
OK | Reached target Network.
OK | Started Create Volatile Files and Directories.
OK | Reached target System Time Synchronized.
Starting Update UTMP about System Boot/Shutdown...
OK | Started Update UTMP about System Boot/Shutdown.
OK | Reached target System Initialization.
OK | Listening on D-Bus System Message Bus Socket.
OK | Reached target Sockets.
OK | Started Daily Cleanup of Temporary Directories.
OK | Started Daily apt download activities.
OK | Started Daily apt upgrade and clean activities.
OK | Reached target Timers.
OK | Reached target Basic System.
Starting System Logging Service...
OK | Reached target Containers.
OK | Started Regular background program processing daemon.
OK | Started D-Bus System Message Bus.
Starting Permit User Sessions...
Starting Login Service...
OK | Started System Logging Service.
OK | Started Permit User Sessions.
OK | Started Console Getty.
OK | Reached target Login Prompts.
OK | Started Login Service.
OK | Reached target Multi-User System.
OK | Reached target Graphical Interface.
Starting Update UTMP about System Runlevel Changes...
OK | Started Update UTMP about System Runlevel Changes.

debian GNU/Linux 9 canamas.how.plexis.eu console
canamas login |

```

SYNOPSIS

```
systemd-nspawn [OPTIONS...] [COMMAND [ARGS...]]  
systemd-nspawn --boot [OPTIONS...] [ARGS...] 2
```

²systemd-nspawn(8)

Image

- `-D, --directory=` – Use this directory as root filesystem
- `-i, --image=` – Use this disk-image as root file-system
- `--template=` – Use this directory or btrfs root as template for the container
- `-x, --ephemeral` – Run with a temporary snapshot of the file system

Execution

- `-b, --boot` – Find an `run init` of the container image
- `-a, --as-pid2` – Run the `COMMAND` as PID 2 with a stub `init` as PID 1
- `--notify-ready=` – The container's `init` will tell `systemd` when it has correctly booted

Networking

- `-p, --port=` – Map a port from the host to the container
- `--network-veth` – Create a virtual ethernet interface in the container
- `--network-bridge=` – Connect the virtual ethernet interface of the container to this bridge
- `--network-macvlan=` – Create a MAC-VLAN for the container on the specified interface
- `--network-interface` – Assign this interface from the host to the container

Filesystems and mounting

- `--bind=`, `--bind-ro=` – Bind a directory from the host in the container
- `--overlay=`, `--overlay-ro=` – Create an overlay of multiple directories on the host as a directory in the container

User namespacing

- `--private-users=` – Enable different user namespacing methods
- `--private-users-chown` – Chown the on-disk files to the mapped UIDs used in the container
- `-U` – Shortcut for `--private-users=pick`
`--private-users-chown`

CONTAINERIZED SERVICES

CONTAINERS AS SERVICES

```
[Unit]
```

```
Description=Foo Service
```

```
[Service]
```

```
Type=Simple
```

```
ExecStart=/usr/bin/systemd-nspawn --quiet \
```

```
-U -i /opt/foo.qcow2 /usr/bin/foo -D
```

```
Restart=on-failure
```

```
CapabilityBoundingSet=CAP_NET_BIND_SERVICE
```

```
NoNewPrivileges=true
```

```
RestrictAddressFamilies=AF_UNIX AF_INET AF_INET6
```

```
LimitNOFILE=4200
```

```
[Install]
```

```
WantedBy=multi-user.target
```

systemd-nspawn@.service

[Unit]

```
Description=Container %i
Documentation=man:systemd-nspawn(1)
PartOf=machines.target
Before=machines.target
After=network.target systemd-resolved.service
RequiresMountsFor=/var/lib/machines
```

[Service]

```
ExecStart=/usr/bin/systemd-nspawn --quiet --keep-unit --boot --link-journal=try-guest \
  --network-veth -U --settings=override --machine=%i
KillMode=mixed
Type=notify
RestartForceExitStatus=133
SuccessExitStatus=133
WatchdogSec=3min
Slice=machine.slice
Delegate=yes
TasksMax=16384
...
```

[Install]

```
WantedBy=machines.target
```

“drop-ins” used for configuration overrides³.

- `/etc/systemd/nspawn/$machine.nspawn`
- `/run/systemd/nspawn/$machine.nspawn`
- In the directory of the image
- In the parent directory of the directory used

³`systemd.nspawn(5)`

.nspawn FILES

[Exec]

Parameters=/usr/bin/foo --foreground

User=foo

ProcessTwo=true

Ephemeral=true

[Files]

Overlay=/srv/dir1:/srv/dir2:/srv/foo

[Network]

Interface=enp0s31 f6

`machinectl` AND WORKING WITH CONTAINERS

WHAT IS machinectl?

DESCRIPTION

machinectl may be used to introspect and control the state of the systemd(1) virtual machine and container registration manager systemd-machined.service(8). machinectl may be used to execute operations on machines and images.⁴

⁴machinectl(1)

- Automatically started when `machine.slice` is started
- Keeps track of running containers and their processes
- Optionally allows local resolving of machine-names⁵
- Machines can be controlled with `machinectl`

⁵`nss-mymachines(8)`

⁶`systemd-machined(8)`

- Download images/directories
- Clone
- Rename
- Import/Export images/directories

IMAGE MANAGEMENT II

```
▲ ~ sudo machinectl pull-tar https://cloud-images.ubuntu.com/bionic/current/bionic-server-cloudimg-amd64-root.tar.xz
Enqueued transfer job 3. Press C-c to continue download in background.
Pulling 'https://cloud-images.ubuntu.com/bionic/current/bionic-server-cloudimg-amd64-root.tar.xz', saving as 'bionic-server-cloudimg-amd64-root'.
Downloading 141.4M for https://cloud-images.ubuntu.com/bionic/current/bionic-server-cloudimg-amd64-root.tar.xz.
Downloading 253B for https://cloud-images.ubuntu.com/bionic/current/bionic-server-cloudimg-amd64-root.nspawn.
HTTP request to https://cloud-images.ubuntu.com/bionic/current/bionic-server-cloudimg-amd64-root.nspawn failed with code 404.
Settings file could not be retrieved, proceeding without.
Downloading 836B for https://cloud-images.ubuntu.com/bionic/current/SHA256SUMS.gpg.
Download of https://cloud-images.ubuntu.com/bionic/current/SHA256SUMS.gpg complete.
Downloading 260B for https://cloud-images.ubuntu.com/bionic/current/bionic-server-cloudimg-amd64-root.tar.xz.sha256.
Downloading 3.3K for https://cloud-images.ubuntu.com/bionic/current/SHA256SUMS.
Download of https://cloud-images.ubuntu.com/bionic/current/SHA256SUMS complete.
Got 1% of https://cloud-images.ubuntu.com/bionic/current/bionic-server-cloudimg-amd64-root.tar.xz. 1min 40s left at 1.3M/s.
Got 2% of https://cloud-images.ubuntu.com/bionic/current/bionic-server-cloudimg-amd64-root.tar.xz. 1min 9s left at 1.9M/s.
Got 3% of https://cloud-images.ubuntu.com/bionic/current/bionic-server-cloudimg-amd64-root.tar.xz. 1min 22s left at 1.6M/s.
Got 6% of https://cloud-images.ubuntu.com/bionic/current/bionic-server-cloudimg-amd64-root.tar.xz. 1min left at 2.1M/s.
^CContinuing download in the background. Use "machinectl cancel-transfer 3" to abort transfer.
△ ~ sudo machinectl list-transfers
ID PERCENT TYPE LOCAL REMOTE
3 n/a pull-tar bionic-server-cloudimg-amd64-root https://cloud-images.ubuntu.com/bionic/current/bionic-server-cloudimg-amd64-root.tar.xz
1 transfers listed.
```

- Start/Stop
- Enable/Disable
- Login/Shell
- Reboot/Poweroff/Terminate
- Copy files from/to the container
- Bind mount a dir in the container

CONTAINER MANAGEMENT II

```
△ ~ sudo machinectl list-images
NAME                TYPE          RO  USAGE  CREATED  MODIFIED
lost+found           directory    no   n/a n/a   n/a
ubuntu-bionic-base  directory    no   n/a n/a   n/a

2 images listed.
▲ ~ sudo machinectl clone ubuntu-bionic-base my-machine
▲ ~ sudo machinectl start my-machine
▲ ~ sudo machinectl list
MACHINE  CLASS  SERVICE          OS    VERSION  ADDRESSES
my-machine  container  systemd-nspawn  ubuntu  18.04   -

1 machines listed.
▲ ~ sudo machinectl list
MACHINE  CLASS  SERVICE          OS    VERSION  ADDRESSES
my-machine  container  systemd-nspawn  ubuntu  18.04   -

1 machines listed.
```

CONTAINER MANAGEMENT III

```
▲ sudo machinectl status my-machine
my-machine(2547282dd1147f8e39745a3add1a7d)
  Since: Thu 2019-06-02 20:16:56 CEST; 39min ago
  Leader: 19955 (systemd)
  Service: systemd-nspawn; class container
  Root: /var/lib/machines/my-machine
  Iface: vx.my-machine
  Address: 192.168.138.194
          169.254.249.120
          fe80::a080:aaff:fe12:2cc6%4
  OS: Ubuntu 18.04.2 LTS
  UID Shift: 105512960
  Unit: systemd-nspawn@my-machine.service
  └─payload
     └─init.scope
        └─19955 /lib/systemd/systemd
           └─system.slice
              └─accounts-daemon.service
                 └─20059 /usr/lib/accounts-service/accounts-daemon
                    └─atd.service
                       └─20077 /usr/sbin/atd -f
                          └─console-getty.service
                             └─20825 /sbin/agetty -o -p -- \u --noclear --keep-baud console 115200,38400,9600 vt220
                                └─cron.service
                                   └─20063 /usr/sbin/cron -f
                                      └─dbus.service
                                         └─20065 /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
                                            └─networkd-dispatcher.service
                                               └─20062 /usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-triggers
                                                  └─polkit.service
                                                     └─20080 /usr/lib/policykit-1/polkitd --no-debug
                                                        └─rsyslog.service
                                                           └─20061 /usr/sbin/rsyslogd -n
                                                              └─systemd-journald.service
                                                                 └─20008 /lib/systemd/systemd-journald
                                                                    └─systemd-logind.service
                                                                       └─20078 /lib/systemd/systemd-logind
                                                                           └─systemd-networkd.service
                                                                              └─20054 /lib/systemd/systemd-networkd
                                                                                 └─systemd-resolved.service
                                                                                    └─20081 /lib/systemd/systemd-resolved
                                                                                       └─unattended-upgrades.service
                                                                                          └─20091 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
                                                                                             └─systemd-journald
                                                                                                └─19953 /usr/bin/systemd-nspawn --quiet --keep-unit --boot --link-journal-try-guest --network-veth -U --settings=override --machine=my-machine
May 02 20:18:56 perzik systemd-nspawn[19953]: [ OK ] Reached target Login Prompts.
May 02 20:18:56 perzik systemd-nspawn[19953]: [ OK ] Started Tezminate Plymouth Boot Screen.
May 02 20:18:56 perzik systemd-nspawn[19953]: [ OK ] Started LSB: Automatic crash report generation.
May 02 20:18:56 perzik systemd-nspawn[19953]: [ OK ] Reached target Multi-User System.
May 02 20:18:56 perzik systemd-nspawn[19953]: [ OK ] Reached target Graphical Interface.
May 02 20:18:56 perzik systemd-nspawn[19953]: Starting Update UTMP about System Runlevel Changes...
May 02 20:18:56 perzik systemd-nspawn[19953]: [ OK ] Started Update UTMP about System Runlevel Changes.
May 02 20:18:57 perzik systemd-nspawn[19953]: [2B blob data]
May 02 20:18:57 perzik systemd-nspawn[19953]: Ubuntu 18.04.2 LTS ubuntu console
May 02 20:18:57 perzik systemd-nspawn[19953]: [1B blob data]
```

CONTAINER MANAGEMENT IV

```
△ - sudo machinectl shell my-machine
Connected to machine my-machine. Press ^] three times within 1s to exit session.
root@ubuntu:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: host0@if4: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state LOWERLAYERDOWN group default qlen 1000
    link/ether a2:80:aa:12:2c:ca brd ff:ff:ff:ff:ff:ff link-netnsid 0
root@ubuntu:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: host0@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether a2:80:aa:12:2c:ca brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 169.254.249.120/16 brd 169.254.255.255 scope link host0
        valid_lft forever preferred_lft forever
    inet 192.168.138.194/28 brd 192.168.138.207 scope global dynamic host0
        valid_lft 3597sec preferred_lft 3597sec
    inet6 fe80:a080:aaff:fe12:2cca/64 scope link
        valid_lft forever preferred_lft forever
root@ubuntu:~# logout
Connection to machine my-machine terminated.
△ - sudo machinectl list
MACHINE   CLASS   SERVICE   OS   VERSION   ADDRESSES
my-machine container systemd-nspawn ubuntu 18.04 192.168.138.194...

1 machines listed.
△ - sudo systemd-nspawn -M my-machine -a ls /
Directory tree /var/lib/machines/my-machine is currently busy.
```

CONTAINER MANAGEMENT V

```
▲ ~ sudo systemctl -M my-machine status ssh.service
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Thu 2019-05-02 20:33:00 CEST; 28min ago
     Process: 140 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
     Process: 146 ExecStart=/usr/sbin/sshd -D $SSH_OPTS (code=exited, status=0/SUCCESS)
    Main PID: 146 (code=exited, status=0/SUCCESS)
▲ ~ sudo systemctl -M my-machine start ssh.service
▲ ~ sudo systemctl -M my-machine status ssh.service
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2019-05-02 21:01:43 CEST; 6s ago
     Process: 260 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
    Main PID: 261
      CGroup: /system.slice/ssh.service
              └─261 /usr/sbin/sshd -D
```


CONTAINER MANAGEMENT VI

```
△ ~ sudo journalctl -M mastodon -u mastodon-web.service -n 10
-- Logs begin at Mon 2019-02-11 21:35:01 CET, end at Mon 2019-04-29 22:59:05 CEST. --
Apr 29 22:55:48 mastodon.lieter.nl bundle[65]: [c547b214-8914-4a5b-ba07-f536f8daa7dc] Rendered text template (0.0ms)
Apr 29 22:55:48 mastodon.lieter.nl bundle[65]: [c547b214-8914-4a5b-ba07-f536f8daa7dc] method=POST path=/inbox format
Apr 29 22:56:23 mastodon.lieter.nl bundle[65]: [544b55e6-f8fb-47f3-a70a-a6808a2511c8] method=POST path=/inbox format
Apr 29 22:56:26 mastodon.lieter.nl bundle[65]: [b529de63-5ea0-4d5d-adea-db3f04e16b6a] method=GET path=/api/v1/instanc
Apr 29 22:56:27 mastodon.lieter.nl bundle[65]: [36747567-470a-4f70-a9ea-28578d2438fd] method=GET path=/api/v1/instanc
Apr 29 22:58:27 mastodon.lieter.nl bundle[65]: [8bf95669-aa7a-45c7-a4ad-2ada688c3061] method=GET path=/api/v1/instanc
Apr 29 22:58:27 mastodon.lieter.nl bundle[65]: [4b70d147-b1f8-4f6c-931f-a973e9ab2052] method=GET path=/api/v1/instanc
Apr 29 22:59:05 mastodon.lieter.nl bundle[65]: [3e733587e-f41f-4deb-b6ee-45eb0d8e5c20] Rendering text template
Apr 29 22:59:05 mastodon.lieter.nl bundle[65]: [3e733587e-f41f-4deb-b6ee-45eb0d8e5c20] Rendered text template (0.0ms)
Apr 29 22:59:05 mastodon.lieter.nl bundle[65]: [3e733587e-f41f-4deb-b6ee-45eb0d8e5c20] method=POST path=/inbox format
```

BUILDING IMAGES

- `debootstrap`, `dnf`, `pacstrap`, `zypper`
- Use `systemd-nspawn -a [...] passwd root` to set a root password
- Do ensure that `systemd-container` is installed

“Build Legacy-Free OS Images”⁷

- From the systemd team⁸
- Different image targets (possibly with LUKS)
- RAW images with GPT disks that can be booted on bare-metal, in a VM or via `systemd-nspawn`
- Supports checksumming and signing images

`mkosi.default`

[Distribution]

Distribution=debian

Release=buster

[Output]

Format=tar

[Packages]

Packages=systemd-container

mkosi (MAKE OPERATING SYSTEM IMAGE) III

```
$ sudo mkosi
[..]
$ ls
ls -lsh
total 57M
 57M -rw-r--r-- 1 root  root  57M May  3 12:13 image.tar.xz
4.0K drwxr-xr-x 3 lieter users 4.0K May  3 12:10 mkosi.cache
4.0K -rw-r--r-- 1 lieter users  110 May  3 12:10 mkosi.default
```

`mkosi.postinst`

```
#!/bin/sh  
echo 'hello'  
# whatever else you need
```

```
sudo machinectl import-tar image.tar.xz debian-buster-base
```

⁷  systemd/mkosi

⁸ [0pointer.net](https://0pointer.net/blog/) blog


```
docker build -t nspawn-test:latest
cd /var/lib/machines
mkdir nspawn-from-docker
docker export \
    $(docker create nspawn-test:latest true) | \
    tar -x -C nspawn-from-docker
```

CONCLUSION

systemd-nspawn offers

- long-lived containers
- full integration in the systemd “ecosystem”
- a simple solution for sysadmins that don’t need
 - Image layering
 - Orchestration frameworks
- building blocks for advanced usage

Thanks

Questions?

EXTRA BITS AND PIECES

USER NAMESPACING IN ACTION I

```
▲ ~ sudo ls /var/lib/machines/my-machine/home -la
total 12
drwxr-xr-x  3 vu-my-machine-0    vg-my-machine-0    4096 May  2 20:40 .
drwxr-xr-x 22 vu-my-machine-0    vg-my-machine-0    4096 May  2 20:16 ..
drwxr-xr-x  2 vu-my-machine-1000 vg-my-machine-1000 4096 May  2 20:40 loadays
▲ ~ sudo ls /var/lib/machines/my-machine/home -lan
total 12
drwxr-xr-x  3 105512960 105512960 4096 May  2 20:40 .
drwxr-xr-x 22 105512960 105512960 4096 May  2 20:16 ..
drwxr-xr-x  2 105513960 105513960 4096 May  2 20:40 loadays
□
```

USER NAMESPACING IN ACTION II

```
▲ ~ ps auxww | grep --color=never systemd
root      1  0.0  0.0 189384 10304 ?        Ss   17:46   0:02 /usr/lib/systemd/systemd --switched-root
root      561  0.0  0.2 144952 75920 ?        Ss   17:46   0:01 /usr/lib/systemd/systemd-journald
root      584  0.0  0.0  41364  8332 ?        Ss   17:46   0:00 /usr/lib/systemd/systemd-udev
root      821  0.0  0.0  28280  7280 ?        Ss   17:46   0:01 /usr/lib/systemd/systemd-logind
dbus      825  0.0  0.0  11436  4780 ?        Ss   17:46   0:04 /usr/bin/dbus-daemon --system --address=
lieter    953  0.0  0.0  34228  9308 ?        Ss   17:46   0:00 /usr/lib/systemd/systemd --user
lieter    963  0.0  0.0  10880  4352 ?        Ss   17:46   0:00 /usr/bin/dbus-daemon --session --address=
lieter   4863  1.2  0.1  82100 38208 pts/0    S+   17:51   2:36 vim LOADdays-systemd-nspawn.tex
lieter   6504  0.0  0.0   6272   2248 pts/5    S+   21:16   0:00 grep --color=auto --exclude-dir=.bzr --e
systemd
lieter   6603  0.3  0.0  19672 14452 pts/3    S+   17:56   0:45 perl /usr/bin/latexmk -lualatex -shell-c
lieter   6616  0.4  1.0 1413312 336072 pts/3    Sl   17:56   0:53 evince LOADdays-systemd-nspawn.pdf
root    12750  0.0  0.0   27912  6888 ?        Ss   19:53   0:00 /usr/lib/systemd/systemd-machined
root    19953  0.0  0.0   28772   7532 ?        Ss   20:16   0:00 /usr/bin/systemd-nspawn --quiet --keep-t
-machiné
vu-my-m+ 19955  0.0  0.0  224952  8976 ?        Ss   20:16   0:00 /lib/systemd/systemd
vu-my-m+ 20008  0.0  0.0   78440  9588 ?        Ss   20:16   0:00 /lib/systemd/systemd-journald
vu-my-m+ 20054  0.0  0.0   80036  5336 ?        Ss   20:16   0:00 /lib/systemd/systemd-networkd
vu-my-m+ 20065  0.0  0.0   50116  4620 ?        Ss   20:16   0:00 /usr/bin/dbus-daemon --system --address=
vu-my-m+ 20078  0.0  0.0   70552  6032 ?        Ss   20:16   0:00 /lib/systemd/systemd-logind
vu-my-m+ 20081  0.0  0.0   70620  5140 ?        Ss   20:16   0:00 /lib/systemd/systemd-resolved
systemd+ 21136  0.0  0.0   28684  6588 ?        Ss   20:20   0:00 /usr/lib/systemd/systemd-networkd
```