NAME

systemd.nspawn − Container settings

SYNOPSIS

/etc/systemd/nspawn/*machine*.nspawn

/run/systemd/nspawn/*machine*.nspawn

/var/lib/machines/*machine*.nspawn

DESCRIPTION

An nspawn container settings file (suffix .nspawn) contains runtime configuration for a local container, and is used used by **systemd-nspawn**(1). Files of this type are named after the containers they define settings for. They are optional, and only required for containers whose execution environment shall differ from the defaults. Files of this type mostly contain settings that may also be set on the **systemd−nspawn** command line, and make it easier to persistently attach specific settings to specific containers. The syntax of these files is inspired by .desktop files, similarly to other configuration files supported by the systemd project. See **systemd.syntax**(7) for an overview.

.NSPAWN FILE DISCOVERY

Files are searched for by appending the .nspawn suffix to the machine name of the container, as specified with the **−−machine=** switch of **systemd−nspawn**, or derived from the directory or image file name. This file is first searched for in /etc/systemd/nspawn/ and /run/systemd/nspawn/. If found there, the settings are read and all of them take full effect (but may still be overridden by corresponding command line arguments). Otherwise, the file will then be searched for next to the image file or in the immediate parent of the root directory of the container. If the file is found there, only a subset of the settings will take effect however. All settings that possibly elevate privileges or grant additional access to resources of the host (such as files or directories) are ignored. To which options this applies is documented below.

Persistent settings files created and maintained by the administrator (and thus trusted) should be placed in /etc/systemd/nspawn/, while automatically downloaded (and thus potentially untrusted) settings files are placed in /var/lib/machines/ instead (next to the container images), where their security impact is limited. In order to add privileged settings to .nspawn files acquired from the image vendor, it is recommended to copy the settings files into /etc/systemd/nspawn/ and edit them there, so that the privileged options become available. The precise algorithm for how the files are searched and interpreted may be configured with **systemd−nspawn**'s **−−settings=** switch, see **systemd-nspawn**(1) for details.

[EXEC] SECTION OPTIONS

Settings files may include an [Exec] section, which carries various execution parameters:

Boot=

Takes a boolean argument, which defaults to off. If enabled, **systemd−nspawn** will automatically search for an init executable and invoke it. In this case, the specified parameters using *Parameters=* are passed as additional arguments to the init process. This setting corresponds to the **−−boot** switch on the **systemd−nspawn** command line. This option may not be combined with *ProcessTwo=yes*. This option is specified by default in the systemd−nspawn@.service template unit.

Ephemeral=

Takes a boolean argument, which defaults to off, If enabled, the container is run with a temporary snapshot of its file system that is removed immediately when the container terminates. This is equivalent to the **−−ephemeral** command line switch. See **systemd-nspawn**(1) for details about the specific options supported.

ProcessTwo=

Takes a boolean argument, which defaults to off. If enabled, the specified program is run as PID 2. A stub init process is run as PID 1. This setting corresponds to the **−−as−pid2** switch on the **systemd−nspawn** command line. This option may not be combined with *Boot=yes*.

Parameters=

Takes a whitespace−separated list of arguments. Single (""") and double (""") quotes may be used around arguments with whitespace. This is either a command line, beginning with the binary name to

execute, or – if *Boot=* is enabled – the list of arguments to pass to the init process. This setting corresponds to the command line parameters passed on the **systemd−nspawn** command line.

Note: **Boot=no**, **Parameters=a b "c c"** is the same as **systemd−nspawn a b "c c"**, and **Boot=yes**, **Parameters=b 'c c'** is the same as **systemd−nspawn −−boot b 'c c'**.

Environment=
Takes an environment variable assignment consisting of key and value, separated by "=". Sets an environment variable for the main process invoked in the container. This setting may be used multiple times to set multiple environment variables. It corresponds to the **−−setenv=** command line switch.

User=
Takes a UNIX user name. Specifies the user name to invoke the main process of the container as. This user must be known in the container's user database. This corresponds to the **−−user=** command line switch.

WorkingDirectory=
Selects the working directory for the process invoked in the container. Expects an absolute path in the container's file system namespace. This corresponds to the **−−chdir=** command line switch.

PivotRoot=
Selects a directory to pivot to / inside the container when starting up. Takes a single path, or a pair of two paths separated by a colon. Both paths must be absolute, and are resolved in the container's file system namespace. This corresponds to the **−−pivot−root=** command line switch.

Capability=, *DropCapability=*
Takes a space−separated list of Linux process capabilities (see **capabilities**(7) for details). The *Capability=* setting specifies additional capabilities to pass on top of the default set of capabilities. The *DropCapability=* setting specifies capabilities to drop from the default set. These settings correspond to the **−−capability=** and **−−drop−capability=** command line switches. Note that *Capability=* is a privileged setting, and only takes effect in .nspawn files in /etc/systemd/nspawn/ and /run/system/nspawn/ (see above). On the other hand, *DropCapability=* takes effect in all cases. If the special value "all" is passed, all capabilities are retained (or dropped).

These settings change the bounding set of capabilities which also limits the ambient capabilities as given with the *AmbientCapability=*.

AmbientCapability=
Takes a space−separated list of Linux process capabilities (see **capabilities**(7) for details). The *AmbientCapability=* setting specifies capability which will be passed to the started program in the inheritable and ambient capability sets. This will grant these capabilities to this process. This setting correspond to the **−−ambient−capability=** command line switch.

The value "all" is not supported for this setting.

The setting of *AmbientCapability=* must be covered by the bounding set settings which were established by *Capability=* and *DropCapability=*.

Note that *AmbientCapability=* is a privileged setting (see above).

NoNewPrivileges=
Takes a boolean argument that controls the **PR_SET_NO_NEW_PRIVS** flag for the container payload. This is equivalent to the **−−no−new−privileges=** command line switch. See **systemd-nspawn**(1) for details.

KillSignal=
Specify the process signal to send to the container's PID 1 when nspawn itself receives SIGTERM, in order to trigger an orderly shutdown of the container. Defaults to SIGRTMIN+3 if **Boot=** is used (on systemd−compatible init systems SIGRTMIN+3 triggers an orderly shutdown). For a list of valid

signals, see **signal**(7).

Personality=
Configures the kernel personality for the container. This is equivalent to the **−−personality=** switch.

MachineID=
Configures the 128−bit machine ID (UUID) to pass to the container. This is equivalent to the **−−uuid=** command line switch. This option is privileged (see above).

PrivateUsers=
Configures support for usernamespacing. This is equivalent to the **−−private−users=** command line switch, and takes the same options. This option is privileged (see above). This option is the default if the systemd−nspawn@.service template unit file is used.

NotifyReady=
Configures support for notifications from the container's init process. This is equivalent to the **−−notify−ready=** command line switch, and takes the same parameters. See **systemd-nspawn**(1) for details about the specific options supported.

SystemCallFilter=
Configures the system call filter applied to containers. This is equivalent to the **−−system−call−filter=** command line switch, and takes the same list parameter. See **systemd-nspawn**(1) for details.

LimitCPU=, *LimitFSIZE=*, *LimitDATA=*, *LimitSTACK=*, *LimitCORE=*, *LimitRSS=*, *LimitNOFILE=*, *LimitAS=*, *LimitNPROC=*, *LimitMEMLOCK=*, *LimitLOCKS=*, *LimitSIGPENDING=*, *LimitMSGQUEUE=*, *LimitNICE=*, *LimitRTPRIO=*, *LimitRTTIME=*
Configures various types of resource limits applied to containers. This is equivalent to the **−−rlimit=** command line switch, and takes the same arguments. See **systemd-nspawn**(1) for details.

OOMScoreAdjust=
Configures the OOM score adjustment value. This is equivalent to the **−−oom−score−adjust=** command line switch, and takes the same argument. See **systemd-nspawn**(1) for details.

CPUAffinity=
Configures the CPU affinity. This is equivalent to the **−−cpu−affinity=** command line switch, and takes the same argument. See **systemd-nspawn**(1) for details.

Hostname=
Configures the kernel hostname set for the container. This is equivalent to the **−−hostname=** command line switch, and takes the same argument. See **systemd-nspawn**(1) for details.

ResolvConf=
Configures how /etc/resolv.conf in the container shall be handled. This is equivalent to the **−−resolv−conf=** command line switch, and takes the same argument. See **systemd-nspawn**(1) for details.

Timezone=
Configures how /etc/localtime in the container shall be handled. This is equivalent to the **−−timezone=** command line switch, and takes the same argument. See **systemd-nspawn**(1) for details.

LinkJournal=
Configures how to link host and container journal setups. This is equivalent to the **−−link−journal=** command line switch, and takes the same parameter. See **systemd-nspawn**(1) for details.

[FILES] SECTION OPTIONS

Settings files may include a [Files] section, which carries various parameters configuring the file system of the container:

ReadOnly=
Takes a boolean argument, which defaults to off. If specified, the container will be run with a read−only file system. This setting corresponds to the **−−read−only** command line switch.

Volatile=

Takes a boolean argument, or the special value "state". This configures whether to run the container with volatile state and/or configuration. This option is equivalent to **−−volatile=**, see **systemd-nspawn**(1) for details about the specific options supported.

Bind=, *BindReadOnly=*
Adds a bind mount from the host into the container. Takes a single path, a pair of two paths separated by a colon, or a triplet of two paths plus an option string separated by colons. This option may be used multiple times to configure multiple bind mounts. This option is equivalent to the command line switches **−−bind=** and **−−bind−ro=**, see **systemd-nspawn**(1) for details about the specific options supported. This setting is privileged (see above).

BindUser=
Binds a user from the host into the container. This option is equivalent to the command line switch **−−bind−user=**, see **systemd-nspawn**(1) for details about the specific options supported. This setting is privileged (see above).

TemporaryFileSystem=
Adds a "tmpfs" mount to the container. Takes a path or a pair of path and option string, separated by a colon. This option may be used multiple times to configure multiple "tmpfs" mounts. This option is equivalent to the command line switch **−−tmpfs=**, see **systemd-nspawn**(1) for details about the specific options supported. This setting is privileged (see above).

Inaccessible=
Masks the specified file or directory in the container, by over−mounting it with an empty file node of the same type with the most restrictive access mode. Takes a file system path as argument. This option may be used multiple times to mask multiple files or directories. This option is equivalent to the command line switch **−−inaccessible=**, see **systemd-nspawn**(1) for details about the specific options supported. This setting is privileged (see above).

Overlay=, *OverlayReadOnly=*
Adds an overlay mount point. Takes a colon−separated list of paths. This option may be used multiple times to configure multiple overlay mounts. This option is equivalent to the command line switches **−−overlay=** and **−−overlay−ro=**, see **systemd-nspawn**(1) for details about the specific options supported. This setting is privileged (see above).

PrivateUsersOwnership=
Configures whether the ownership of the files and directories in the container tree shall be adjusted to the UID/GID range used, if necessary and user namespacing is enabled. This is equivalent to the **−−private−users−ownership=** command line switch. This option is privileged (see above).

[NETWORK] SECTION OPTIONS

Settings files may include a [Network] section, which carries various parameters configuring the network connectivity of the container:

Private=
Takes a boolean argument, which defaults to off. If enabled, the container will run in its own network namespace and not share network interfaces and configuration with the host. This setting corresponds to the **−−private−network** command line switch.

VirtualEthernet=
Takes a boolean argument. Configures whether to create a virtual Ethernet connection ("veth") between host and the container. This setting implies *Private=yes*. This setting corresponds to the **−−network−veth** command line switch. This option is privileged (see above). This option is the default if the systemd−nspawn@.service template unit file is used.

VirtualEthernetExtra=
Takes a colon−separated pair of interface names. Configures an additional virtual Ethernet connection ("veth") between host and the container. The first specified name is the interface name on the host, the second the interface name in the container. The latter may be omitted in which case it is set to the same name as the host side interface. This setting implies *Private=yes*. This setting corresponds to the

−−network−veth−extra= command line switch, and maybe be used multiple times. It is independent of *VirtualEthernet=*. Note that this option is unrelated to the *Bridge=* setting below, and thus any connections created this way are not automatically added to any bridge device on the host side. This option is privileged (see above).

Interface=
Takes a space−separated list of interfaces to add to the container. This option corresponds to the **−−network−interface=** command line switch and implies *Private=yes*. This option is privileged (see above).

MACVLAN=, *IPVLAN=*
Takes a space−separated list of interfaces to add MACLVAN or IPVLAN interfaces to, which are then added to the container. These options correspond to the **−−network−macvlan=** and **−−network−ipvlan=** command line switches and imply *Private=yes*. These options are privileged (see above).

Bridge=
Takes an interface name. This setting implies *VirtualEthernet=yes* and *Private=yes* and has the effect that the host side of the created virtual Ethernet link is connected to the specified bridge interface. This option corresponds to the **−−network−bridge=** command line switch. This option is privileged (see above).

Zone=
Takes a network zone name. This setting implies *VirtualEthernet=yes* and *Private=yes* and has the effect that the host side of the created virtual Ethernet link is connected to an automatically managed bridge interface named after the passed argument, prefixed with "vz−". This option corresponds to the **−−network−zone=** command line switch. This option is privileged (see above).

Port=
Exposes a TCP or UDP port of the container on the host. This option corresponds to the **−−port=** command line switch, see **systemd-nspawn**(1) for the precise syntax of the argument this option takes. This option is privileged (see above).

SEE ALSO

systemd(1), **systemd-nspawn**(1), **systemd.directives**(7)