Lab Exercise – systemd-nspawn

Plan: systemd-nspawn is a powerful namespace container for debugging, testing and building. In these exercises we will get a much better understanding of the internal working of systemd itself

For the following exercise the SLES12 iso needs to be provided by the instructor, if not already done.

The SLES12SP2 iso file is provided in directory /home/iso.

```
mkdir -p /mnt/loop

mount -o loop /dev/disk/by-id/ata-QEMU_DVD-ROM_QM00003 /mnt/loop

mkdir -p /var/tmp/sles12minimal/etc/zypp/repos.d/

cat << EOF > /var/tmp/sles12minimal/etc/zypp/repos.d/repo_sles12.repo
            [repo_sles12]
            name=repo_sles12
            enabled=1
            autorefresh=0
            baseurl=/mnt/loop
            type=yast2
            EOF
```

#if a yum repo is used as baseurl, repository meta-data type needs to be changed to rpm-md

#baseurl=http://download.opensuse.org/distribution/13.2/repo/oss/

#install minimal SLES12SP2 distribution without kernel and grub

#Operating in root directory /var/tmp/sles12minimal

The --root option influences the location of the repos.d directory and the metadata cache directory and also causes rpm to be run with the --root option to do the actual installation or removal of packages

#test if zypp repos are available first

sudo zypper --root /var/tmp/sles12minimal ref

```
# sudo zypper --root /var/tmp/sles12minimal  install --no-recommends
systemd syslinux perl-Bootloader-YAML zypper
```

you will see a message:

The following 134 NEW packages are going to be installed:

[..]

the following NEW product is going to be installed:

SUSE Linux Enterprise Server 12

Don't run mkinitrd after installation!

```
# mkdir -p /var/tmp/sles12min
```

We copy the contents of the root directory into its own filesystem. Hint:with rsync always use the --dry-run option or -n first

```
sles12t1:~ # rsync -av  /var/tmp/sles12minimal/* /var/tmp/sles12min/
```

Only very little disk space is used in the newly created directory

```
du -s -m /var/tmp/sles12min
# find /var/tmp/sles12min|wc -l
10776
```

adapt /etc/shadow in container for root login, password: linux

```
# grep root /var/tmp/sles12min/etc/shadow
```

root:
6bWg4cZho5l2J$Ttce9S63ssTNHXpMch6DbcX3Z2v3jRtuxu7mWVZ60PVH7Sz10xh.gHzMsylrYS9xyfzVr0EFqvjmJY8D40t7N0:16458:0:::::

Hint: adapt the uid as well.

We want to be able to login to the container via ssh

```
zypper --root /var/tmp/sles12min  install --no-recommends iproute2 vim openssh
```

test, what you have done so far:

parameters used:

-b, --boot

Automatically search for an init binary and invoke it instead of a shell or a user supplied program.

-D, --directory=

Directory to use as file system root for the container

```
# /usr/bin/systemd-nspawn -bD /var/tmp/sles12min 3
```

you will not be able to login, read up on securetty before executing the following command from a new root shell to VM trouble

|Press ^] three times within 1s to kill container

check contents of file /etc/securetty first:

```
cat /var/tmp/sles12min/etc/securetty

# echo console >>/var/tmp/sles12min/etc/securetty
```

Or for testing purposes only delete securetty.

Start the container again and try to login ,this time it will work, we have just shown, that you are able to change files from the host OS for an nspawn container while it is running!

For troubleshooting purposes this allows to rapidly modify files.

Execute some commands in the container and look up their commands if not familiar with them

```
systemctl, journalctl, ip a, mount, systemd-cgls, systemd-
analyze critical-chain
```

in the host trouble execute:

```
ps -eo pid,machine,args
```

the second column shows the container a process belongs to

see man ps

„machine MACHINE displays machine name for processes assigned to VM or container.“

You will get an even better understanding of the boot process by observing the debugging output with the following systemd-nspawn command, that

```
/usr/bin/systemd-nspawn -bD /var/tmp/sles12min 3
systemd.log_level=debug
```

spawned /usr/lib/systemd/system-generators/systemd-fstab-generator

loading of unit files in priority

load configuration failures

Creation of private D-Bus server

Activating default.target

Ignoring failed dependency jobs

Installing targets, service, sockets, mounts,..

use the following command to log boot process to a textfile:

```
/usr/bin/systemd-nspawn -D <rootfs_dir> /bin/systemd
systemd.log_level=debug | tee /tmp/systemd-debug.log
```

Use the generated logfile to answer the following questions:

which services in the container SLES12SP2 are launched using a sysv init script ?

Lab Exercise – mail.service

cat /etc/systemd/system/mail1.service

```
[Unit]
Description=Test
After=sshd.service

[Service]
Type=oneshot
ExecStart=/bin/sh -c "/usr/bin/mail -s test root@localhost </dev/null"
Restart=no
User=root
SuccessExitStatus=0

[Install]
WantedBy=multi-user.target
```

systemd should send a mail. If you run the cmd manually all is ok.

/usr/bin/mail -s test root@localhost </dev/null

verify with by inspecting /var/mail/root (assuming postfix service was started, which is default on provided demo system)

If it is started by systemd it will get a SIGTERM immediately after the start and terminate.

What is the reason, that it was terminated, what needs to be adjusted in the unit file, to get the unit working?

Hint: systemd.kill(5

Lab Exercise – resource control

Change resource control settings at runtime via cgroups

```
cat /etc/systemd/system/read-vda.service
[Unit]
Description=test resource control setting at runtime

[Service]
Type=oneshot
ExecStart=/usr/bin/head /dev/vda -c 1
DevicePolicy=strict
```

systemctl restart read-vda.service

Starting test...

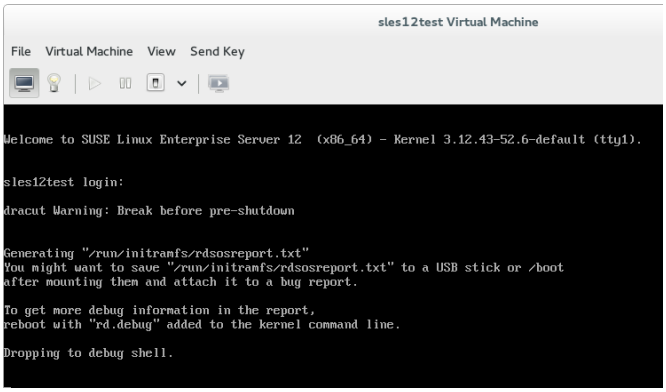
/usr/bin/head: cannot open '/dev/vda' for reading: Operation not permitted

How do you fix this service ?

Hint: systemd.resource-control(5)

Lab Exercise – dracut

In this exercise you will learn how to debug the shutdown process using dracut



In your VM “trouble” execute:

```
mkdir -p /run/initramfs/etc/cmdline.d

echo "rd.break=pre-shutdown rd.shell" >
/run/initramfs/etc/cmdline.d/debug.conf

touch /run/initramfs/.need_shutdown

systemctl start dracut-shutdown.service
```

and reboot

Provide the output of the following commands:

```
systemctl status dracut-shutdown.service

journalctl -u dracut-shutdown.service

bash -x /usr/lib/dracut/dracut-initramfs-restore
```

when you enter the pre-shutdown shell make note of the following files:

oldsys, shutdown and rdsosreport

If on your machine during shutdown process the pre-shutdown shell is not displayed, change the kernel cmdline to exclude "splash=silent quiet".

Early Debug Shell during boot

You can enable shell access to be available very early in the startup process to fall back on and diagnose systemd related boot up issues with various systemctl commands.

```
systemctl cat debug-shell.service
```

Enable it using:

```
systemctl enable debug-shell.service
```

Reboot

make the debug shell appear only on tty11

Lab exercise – boot when root account is locked - sulogin

No warranty on this exercise – you are on your own!

exercise:
if the root account is locked and --force is used - the prompt for the root password (which will not be useful as root is locked) will be skipped.

```
rpm -qf /usr/sbin/sulogin
  util-linux-2.25-12.2.x86_64
```

sulogin man page

```
 -e, --force
              If the default method of obtaining the root  password
from the system  via  getpwnam(3)  fails,  then  examine  /etc/passwd
and /etc/shadow to get the password.  If these files are damaged or
nonexistent,  sulogin will start a root shell without asking for a
password.
              Only use the -e option if you are sure the console is
physically protected against unauthorized access.
```

/etc/systemd/system/emergency.service.d/sulogin-force.conf

```
[Service]
ExecStart=
ExecStart=-/bin/sh -c "/usr/sbin/sulogin --force; /usr/bin/systemctl \
--fail --no-block default"
```

```
systemctl daemon-reload
mkinitrd
```

Show that 'root' is _locked_

```
sudo grep root /etc/shadow
root:!$6$e7c0RU672R7c$QYh4IW1LcyWu4b7us
reboot
```

append 'single' to grub after reboot ctl-x