

Hints Lab Exercise – systemd 1

Plan: in the following exercises we will explore important systemd commands:

- 0.9 there are over 130 man pages.

How do you quickly determine the right manpage for an unknown directive?

A: Man systemd.directives

```
man -l /usr/share/man/man7/systemd.directives.7.gz
>systemd.directives.txt

wget -c http://man7.org/linux/man-pages/man7/systemd.directives.7.html
w3m -dump -T text/html systemd.directives.7.html
>systemd.directives.7.txt

# the man-page: "([0-9])"

grep "([0-9])" systemd.directives.7.txt|awk '{cnt[$1]++} END{for (item
in cnt) print item, cnt[item]}' |sort -n -k 2 >systemd.directives-
count.txt
```

- 1.0 is the journal persistent?

man journald.conf

Storage=

Controls where to store journal data. One of "volatile", "persistent", "auto" and "none".

If "persistent", data will be stored preferably on disk, i.e. below the /var/log/journal hierarchy (which is created if needed), with a fallback to /run/log/journal (which is created if needed), during early boot and if the disk is not writable. "auto" is similar to "persistent" but the directory /var/log/journal is not created if needed, so that its existence controls where log data goes. "none" turns off all storage, all log data received will be dropped. Forwarding to other targets, such as the console, the kernel log buffer or a syslog socket will still work however. Defaults to "auto".

- 1.1 In your KVM VM within virt-manager execute Ctrl+Alt+F10

Where does this output come from?

In SLES12SP2 log messages received by the journal daemon are forwarded to the console

```
# grep -ir tty /etc/systemd/
```

```
/etc/systemd/journald.conf:#TTYPath=/dev/tty10
```

Journal events messages are immediately forwarded to a socket(/run/systemd/journal/syslog), where the traditional syslog daemon can read them.

journald.conf(5)

- 1.2 installing lamp server without recommends

Hints Lab Exercise – systemd 1

```
zypper in --no-recommends -t pattern lamp_server
```

• 1.3 services disabled although installed

per default all services are disabled although installed

```
# cat /usr/lib/systemd/system-preset/99-default-  
disable.preset  
  
disable *
```

#exceptions:

```
/usr/lib/systemd/system-preset/90-default-SLE.preset
```

1.4 What files are changed when you execute the following commands?

```
systemctl enable apache2  
  
# ls -l /etc/systemd/system/multi-user.target.wants/ |grep apache2  
  
lrwxrwxrwx 1 root root 39 Oct  8 16:50 apache2.service ->  
/usr/lib/systemd/system/apache2.service  
  
systemctl disable apache2  
  
# ls -l /etc/systemd/system/multi-user.target.wants/ |grep apache2  
  
systemctl mask apache2  
  
# systemd-delta |grep apache2  
  
[MASKED]      /etc/systemd/system/apache2.service → /usr/lib/systemd/system/apache2.service  
  
# ls -l /etc/systemd/system/apache2.service  
  
lrwxrwxrwx 1 root root 9 Oct  8 16:37 /etc/systemd/system/apache2.service -> /dev/null  
  
systemctl unmask apache2  
  
ls -l /etc/systemd/system/apache2.service  
  
ls: cannot access /etc/systemd/system/apache2.service: No such file or directory
```

systemd.unit(5)

If a unit file is empty (i.e. has the file size 0) or is symlinked to /dev/null, its configuration will not be loaded and it appears with a load state of "masked", and cannot be activated. Use this as an effective way to fully disable a unit, making it impossible to start it even manually.

• 2.0 execute:

```
rccron status
```

• what does the command execute ?

Hint: softlink

```
# ls -l /usr/sbin/rccron  
  
/usr/sbin/rccron -> /sbin/service
```

Hints Lab Exercise – systemd 1

Change to /usr/lib/systemd/system

- 2.2 which services do not have their own .service file

hint:

```
ls -l /usr/sbin/rc*
```

- all services in this directory are linked to binary service

```
# grep systemctl /usr/sbin/service
```

- 2.5.1 What is the main difference this is causing?

Hint: Read systemd.service(5) and search for DefaultDependencies

Unless DefaultDependencies= is set to false, service units will implicitly have dependencies of type Requires= and After= on sysinit.target, a dependency of type After= on basic.target as well as dependencies of type Conflicts= and Before= on shutdown.target.

These ensure that normal service units pull in basic system initialization, and are terminated cleanly prior to system shutdown.

Only services involved with early boot or late system shutdown should disable this option.

- 2.6 what other types are available for systemctl command?

Hint: use bash completion

A:

```
systemctl -t <tab>
```

automount device mount path service snapshot socket swap target timer

See the respective man pages for more information: systemd.service(5), systemd.socket(5), systemd.device(5), systemd.mount(5), systemd.automount(5), systemd.swap(5), systemd.target(5), systemd.path(5), systemd.timer(5), systemd.snapshot(5) systemd.slice(5). systemd.scope(5).

- 2.7 Display the contents of the cron service:

A: systemctl cat cron

- 2.8

A:

After=ybind.service nscd.service network.target

After=postfix.service sendmail.service exim.service

- 2.9 what are the units that depend on it (Look for WantedBy and RequiredBy)

[Install]

Hints Lab Exercise – systemd 1

WantedBy=multi-user.target

- 2.10 when will the cron service be restarted?

A: on-abort

If set to on-abort, the service will be restarted only if the service process exits due to an uncaught signal not specified as a clean exit status

- 2.11 what other options are available for restart of a service?

no, on-success, on-failure, on-watchdog, on-abort, or always

- 2.12 which manpage describes this?

systemd.service(5)

- 2.14 what happens to the service?

A: restarted

- 2.16 Which commands help you to see log/error messages

A:

```
systemctl status cron, journalctl -u cron
```

- 2.17 stop the unit using systemctl stop cron<tab> .

- 2.18 What is its status now?

A: Loaded: loaded (/usr/lib/systemd/system/cron.service; enabled)

Active: inactive (dead)

- 2.19 Disable the unit. What does the output of systemctl show?

A: Loaded: loaded (/usr/lib/systemd/system/cron.service; disabled)

- 2.20 List the directory that the indicated symlink used to be in. Re-examine the unit file and see which directive created this symlink. This symlink was created in response to a WantedBy directive.

hint: use systemctl list-unit-files

A:

```
/etc/systemd/system/multi-user.target.wants/
```

is missing the cron file

- 2.21 Re-enable the unit. This will re-create the symlink. List the directory that the symlink was contained in.

A:

```
ls -ltr /etc/systemd/system/multi-user.target.wants/
```

Have a look at the timestamp of symlink

Hints Lab Exercise – systemd 1

A: it has the current timestamp

- 2.22 Now list the units that depend on this one using `systemctl --reverse list-dependencies`

A: `cron.service`

└─multi-user.target

└─graphical.target

- 2.23 Finally, list the units that this unit depends on using the previous command without the `--reverse` option. Is there a surprise?

A: yes, The units this unit depends on depend on a lot of other units, so this dependency list is large.

For a further explanation lookup “man systemctl” - target units are recursively expanded

- 2.24 is the cron unit started yet?

list its status [It should not be, assuming you only enabled it and did not start it.]

- 2.25 Start the unit.

- 3.0 Look at the symlink named `default.target`. What is it symlinked to?

`ls -l /usr/lib/systemd/system/default.target`

Note that the symlink appearing here is really just window-dressing. The real symlink appears in `/etc/systemd/system` as `default.target`. List it.

```
ls -l /etc/systemd/system/default.target
```

```
/etc/systemd/system/default.target ->  
/usr/lib/systemd/system/graphical.target
```

Remember, `/etc/systemd/system` contains all local configuration.

- 3.2 name another(shorter) way of doing:

`less /usr/lib/systemd/system/graphical.target`

A: `systemctl cat gra<tab>`

A: `systemctl cat graphical.target`

- 3.3 Change your default target to multi-user mode using

A: `systemctl set-default multi-user.target`

- 3.4 Did either of the `default.target` symlinks change? You just changed the

Hints Lab Exercise – systemd 1

system 'initdefault' state effectively from 5 to 3. Did you disable the GUI console?

A:

```
ls -l /etc/systemd/system/default.target  
  
/etc/systemd/system/default.target ->  
/usr/lib/systemd/system/multi-user.target
```

- 3.5 Issue the command:

```
systemd-cgls | tee /tmp/graphical-target.txt
```

- We will now go immediately to multi-user.target. This will not be apparent unless you can see the GUI console.

- 3.6 Issue the command

```
systemctl isolate multi-user.target
```

If you can see the GUI console, verify that the console has changed to a text prompt. Otherwise, look in the output of `systemd-cgls` and verify that the `gdm` processes have exited

execute the following commands:

```
systemd-cgls | tee /tmp/multi-target.txt
```

- 3.7 Execute:

```
vimdiff <(cat /tmp/graphical-target.txt) <(cat /tmp/multi-target.txt)
```

- 3.8 which services have also been used in the graphical target but not in multi-user.target ?

hint: use

```
journalctl -xb
```

- to inquire `journalctl` options look up its man page

A: example output only:

```
Sep 02 16:23:37 sles12t1 dbus[812]: [system] Activating via systemd: service  
name='org.freedesktop.locale1' unit='dbus-org.freedesktop.locale1.ser
```

```
Sep 02 16:23:37 sles12t1 dbus[812]: [system] Successfully activated service 'org.freedesktop.locale1'
```

```
Sep 02 16:23:37 sles12t1 dbus[812]: [system] Rejected send message, 1 matched rules;  
type="method_call", sender=":1.104" (uid=0 pid=4850 comm="/us
```

Hints Lab Exercise – systemd 1

Sep 02 16:23:37 sles12t1 dbus[812]: [system] Activating via systemd: service name='org.freedesktop.Accounts' unit='accounts-daemon.service'

Sep 02 16:23:37 sles12t1 dbus[812]: [system] Activating via systemd: service name='org.freedesktop.PolicyKit1' unit='polkit.service'

- 3.9 what do above services have in common?

A: they are all activated by dbus

```
alias A1=' awk '\''{print $1}'\''  
  
sles12t1:~ # systemctl --type=service|A1|while read x; do  
systemctl --type=service --property=Type show $x|grep dbus  
&& printf $x'\n'; done
```

accounts-daemon.serviceType=dbus

polkit.serviceType=dbus

rtkit-daemon.serviceType=dbus

systemd-logind.serviceType=dbus

upower.serviceType=dbus

wickedd-auto4.serviceType=dbus

wickedd-dhcp4.serviceType=dbus

wickedd-dhcp6.serviceType=dbus

wickedd-nanny.serviceType=dbus

- 3.10 look up the accounts-daemon,i.e.

A:

```
systemctl cat accounts-daemon
```

Lab exercise 3 – systemd listing units

Lab exercise 3 – systemd listing units

Plan: In this exercise we will see which units are installed and enabled on your system.

In the previous exercise, you listed the unit files with the extension `.service` in the main systemd directory `/usr/lib/systemd/system`.

- Recreate that list.
- Ask systemd for a list of all the unit files using the option `--type=service` and the command `list-unit-files`. If you see any discrepancies, you should know that there is a second directory of systemd information `/etc/systemd/system`. This second directory is reserved for implementation-specific data.
- execute `systemd-delta`
- `list-unit-files` does not tell you what units are enabled. For that, use the `list-units` command (again, with `--type=service`). This indicates which units are enabled on your system. In the listing you will find some service units that are not enabled (disabled). Can you find any that you recognize?
- Repeat the above experiments changing the type of the units to `target`.

The `isolate` command for `systemctl` can be used on any target that can 'stand on its own', or, in other words, can trigger a valid change in system state. List all the target units in `/usr/lib/systemd/system` that allow the use of the `isolate` command (they have the keyword/value pair `AllowIsolate=yes` in their unit file).

A:

```
grep -l "AllowIsolate=yes" /usr/lib/systemd/system/*.target
```

Note your findings:

Lab exercise 3 – systemd listing units
