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

```
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.

• 1.5 how would you list all masked services?

systemctl list-unit-files | grep masked

•2.0 execute:

```
rccron status
```

• what does the command execute ?

Hint: softlink

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

```
/usr/sbin/rccron -> /sbin/service
```

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=ypbind.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]

WantedBy=multi-user.target

• 2.10 the cron service will be restarted on-abort

```
systemctl cat cron.service
```

Difference between sending SIGTERM signal or SIGKILL

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

systemd.service(5)

Table 1. Exit causes and the effect of the Restart= settings

Note your findings: