

systemd

Pierre Lemay

systemd est un gestionnaire
de système et de services

Survol

- Contrôle des unités (“units”)
- Gère les dépendances entre les unités
- Suit les processus et les services
 - Les services utilisent les cgroups
 - Limitations simple to configurer, CPU, mémoire et IO
- Termine (kill) les daemons adéquatement
- Debuggable – aucun message de démarrage perdu

Les unités

Unités

- Par convention: nom.type
 - httpd.service, sshd.socket, ou dev-hugepages.mount
 - Sans type, .service est assumé
- **Service** – Type de daemon, exécution, environnement et comment la surveiller
- **Socket** – Point de terminaison pour les communications entre les processus. Fichier, réseau ou socket Unix
- **Target** – Regroupement logique d'unités. Remplacement des runlevels
- **Device** – Créée par le kernel. Peuvent être identifiées comme dépendance pour les services.

Unités (suite)

- **Mounts, automounts, swap** – Gestion des points de montages
- **Snapshots** – Sauvegarde de l'état des services
- **Timers** – Activation via un chronomètre
- **Paths** – Utilise inotify pour surveiller une cible
- **Slices** – hiérarchie des cgroups pour la gestion des ressources et l'assignation de limites aux services
- **Scopes** – unité d'organisation des processus des services

Gestion des dépendances

- Dépendances gérées entre unités
 - Requiert d'autres unités
 - Identifie l'ordre d'exécution, avant ou après les autres unités
- Par exemple :
 - `nfs-lock.service`:
 - `Requires=rpcbind.service network.target`
 - `After=network.target named.service rpcbind.service`
 - `Before=remote-fs-pre.target`

Gestion des services

Gestions des services – Fichiers d'unités

- Fichiers
 - Distribution: `/usr/lib/systemd/system/`
 - Personnalisation: `/etc/systemd/system/`

Note : Les fichiers d'unités sous `/etc/` ont précedence sur les fichiers sous `/usr`

Gestion des services – Arrêts/départs

```
$ systemctl {start,stop,restart,reload,kill} service[.service]
```

- L'action est placé avant le nom du service
- Si le type d'unité n'est pas spécifié, .service est assumé
 - `systemctl start httpd == systemctl start httpd.service`
- La complétion « tab » fonctionne bien avec systemctl

Gestion des services – État

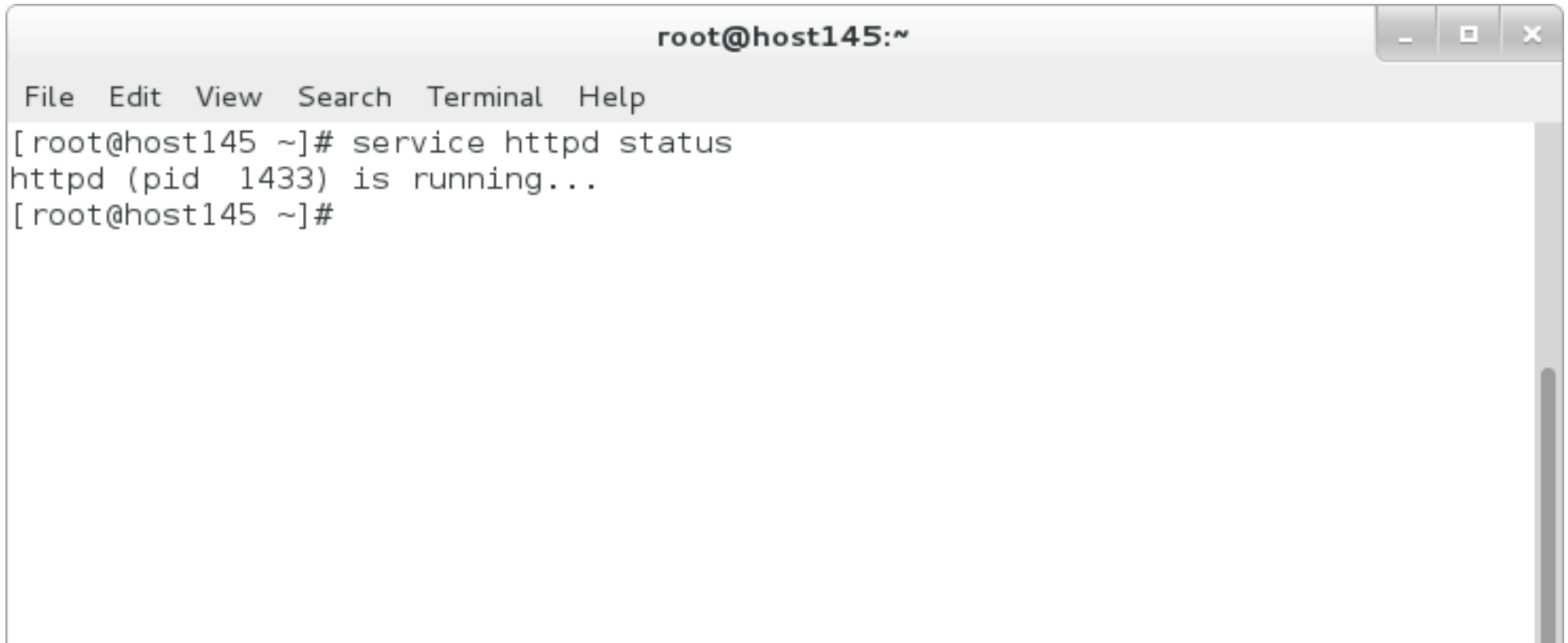
\$ systemctl status httpd.service

```
root@host158:~
File Edit View Search Terminal Help
[root@host158 ~]# systemctl status httpd
httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled)
  Active: active (running) since Fri 2013-08-09 09:22:25 CDT; 12s ago
  Process: 890 ExecStop=/usr/sbin/httpd $OPTIONS -k graceful-stop (code=exited, status
=0/SUCCESS)
  Main PID: 893 (httpd)
  Status: "Total requests: 0; Current requests/sec: 0; Current traffic:  0 B/sec"
  CGroup: name=systemd:/system/httpd.service
          └─893 /usr/sbin/httpd -DFOREGROUND
            └─894 /usr/sbin/httpd -DFOREGROUND
              └─895 /usr/sbin/httpd -DFOREGROUND
                └─896 /usr/sbin/httpd -DFOREGROUND
                  └─897 /usr/sbin/httpd -DFOREGROUND
                    └─898 /usr/sbin/httpd -DFOREGROUND

Aug 09 09:22:23 host158.local systemd[1]: Starting The Apache HTTP Server...
Aug 09 09:22:25 host158.local systemd[1]: Started The Apache HTTP Server.
[root@host158 ~]# █
```

Gestion des services – État

- Un peu plus pratique que...



```
root@host145:~  
File Edit View Search Terminal Help  
[root@host145 ~]# service httpd status  
httpd (pid 1433) is running...  
[root@host145 ~]#
```

Gestion des services – État

- Liste des services chargés:
 - `systemctl -t service`
- Liste des services installés:
 - `systemctl list-unit-files -t service` (similaire à `chkconfig --list`)
- Liste par état:
 - `systemctl --state failed`

Gestion des services – Activation

```
$ systemctl {enable, disable, mask, unmask} httpd.service
```

- Mask :
 - Empêche le démarrage manuel des services
 - “This will link these units to /dev/null, making it impossible to start them. This is a stronger version of disable, since it prohibits all kinds of activation of the unit, including manual activation. Use this option with care.”

Runlevels



Runlevels == Targets

- “Runlevels” sont remplacé par des unités du type “target”
- /etc/inittab n'est plus utilisé
- Les targets sont nommés et non numérotés:
 - multi-user.target au lieu de runlevel3
 - graphical.target au lieu de runlevel5
- Pour obtenir la configuration par défaut:
 - ``systemctl get-default``
- Établir la configuration par défaut:
 - ``systemctl set-default graphical.target``

Runlevels == Targets

- Changement de target interactif:
 - ``systemctl isolate [target]``
- Target au démarrage en ajoutant comme paramètre du kernel:
 - `systemd.unit=[target]`
 - `systemd.unit=rescue.target` (ou “1” ou “s”)
 - Système de base avec un shell, équivalent à `init=1`
 - `systemd.unit=emergency.target`
 - Équivalent de `init=/bin/sh`, permet toutefois un démarrage complet
 - `systemd.unit=multiuser.target`
 - `systemd.unit=graphical.target`

Runlevel nommés comme targets

Runlevel	Systemd Target	Description
0	poweroff.target, runlevel0.target	System halt
1	rescue.target, runlevel1.target	Single user mode
3 (2,4)	multi-user.target, runlevel3.target	Multi-user, non graphical
5	graphical.target, runlevel5.target	Multi-user, graphical
6	reboot.target, runlevel6.target	System reboot

```
ls /lib/systemd/system/runlevel*target -l
```

```
lrwxrwxrwx. 1 root root 15 Jul 3 21:37 /lib/systemd/system/runlevel0.target -> poweroff.target
lrwxrwxrwx. 1 root root 13 Jul 3 21:37 /lib/systemd/system/runlevel1.target -> rescue.target
lrwxrwxrwx. 1 root root 17 Jul 3 21:37 /lib/systemd/system/runlevel2.target -> multi-user.target
lrwxrwxrwx. 1 root root 17 Jul 3 21:37 /lib/systemd/system/runlevel3.target -> multi-user.target
lrwxrwxrwx. 1 root root 17 Jul 3 21:37 /lib/systemd/system/runlevel4.target -> multi-user.target
lrwxrwxrwx. 1 root root 16 Jul 3 21:37 /lib/systemd/system/runlevel5.target -> graphical.target
lrwxrwxrwx. 1 root root 13 Jul 3 21:37 /lib/systemd/system/runlevel6.target -> reboot.target
```

getty

getty

- N'utilise pas /etc/inittab
- Configuration par défaut
 - /etc/systemd/login.conf
- Arrêt / départ via:
 - `systemctl [start|stop] getty@tty[1-x]`

getty

- Configuration pour les terminaux série:
 - `cp /usr/lib/systemd/system/serial-getty@.service /etc/systemd/system/serial-getty@ttyS2.service`
 - Editer la config
 - `systemctl enable serial-getty@ttyS2.service`
 - `systemctl start serial-getty@ttyS2.service`
- Ajouter `console=ttyS0` à la ligne de commande du kernel pour un démarrage automatique d'une console série

Débogage au démarrage

Démarrage

- Interactif: `systemd.confirm_spawn=1`
- `/var/log/boot.log` – pas de changement
- Débogage depuis grub:
 - `systemd.log_level=debug systemd.log_target=kmsg log_buf_len=1M`
- Activation d'un « boot shell » sur `tty9`
 - `systemctl enable debug-shell.service`
- `systemctl list-jobs`

Démarrage

- rc.local
 - touch /etc/rc.d/rc.local ; chmod +x /etc/rc.d/rc.local
 - N'oublier pas #!/bin/bash
- systemd-analyze
 - blame
 - plot
 - critical-chain

Personnaliser les fichier d'unités de service

Personnaliser les fichier d'unités de service

- Fichiers “drop-ins” pour altérer la configuration par défaut d'un service sans impacter le fichier original :
`/etc/systemd/system/foobar.service.d/*.conf`

```
# cat /etc/systemd/system/httpd.service.d/50-httpd.conf
```

```
[Service]  
Restart=always  
StartLimitInterval=10  
StartLimitBurst=5  
StartLimitAction=reboot  
CPUShares=2048
```

Personnaliser les fichier d'unités de service

- Puis :
 - `systemctl daemon-reload`
- Personnalisation visible avec ``systemctl status``

```
# systemctl status httpd.service
```

```
httpd.service - The Apache HTTP Server
```

```
Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled)
```

```
Drop-In: /etc/systemd/system/httpd.service.d
```

```
└─50-httpd.conf
```

Personnaliser les fichier d'unités de service

- **NE PAS MODIFIER** les fichiers sous :
`/usr/lib/systemd/system/`, ces changements seraient perdus avec les mises à jour
- Les fichiers de service sous `/etc` en précédence sur les fichiers de service sous `/usr`
- Effacer simplement les « drop-in » pour revenir aux configurations par défaut. N'oublier pas ``systemctl daemon-reload``
- `systemd-delta` – permet de trouver les configurations personnalisées

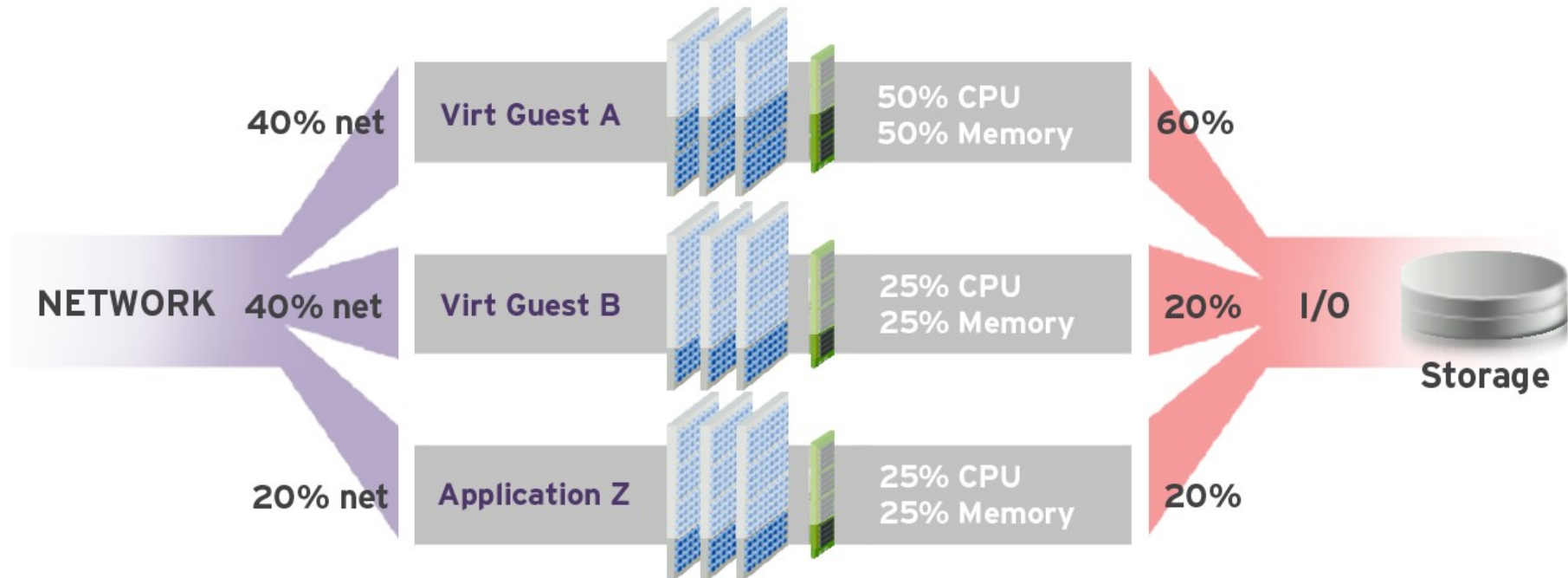
Personnaliser les fichier d'unités de service

- Nice, CPUAffinity, CPUSchedulingPolicy, CPUSchedulingPriority, LimitCPU, IOSchedulingPriority, OOMScoreAdjust, IOSchedulingClass, etc
- Pour plus de details:
 - man 5 systemd.service
 - man 5 systemd.exec

Gestion de ressources

Gestion de ressources

- La gestion des ressources avec les “cgroups” permet de contrôler les applications, de réduire la contention et de prévoir les comportements



Gestion de ressources

- Visualiser la hiérarchie des cgroups via `systemd-cgls`
- Visualiser les statistique d'utilisation via `systemd-cgtop`
- Hierarchie par défaut
 - `system.slice` – contient les services systèmes
 - `user.slice` – contient les sessions des utilisateurs
 - `machine.slice` – contient les machines virtuelles et les conteneurs
- Les services peuvent utiliser leur propre slice au besoin

Gestion de ressources – systemd-cgls

```
File Edit View Search Terminal Help
├─1 /usr/lib/systemd/systemd --switched-root --system --deserialize 22
├─machine.slice
│   ├──machine-qemu\x2drhel7.scope
│   │   └─17307 /usr/bin/qemu-system-x86_64 -machine accel=kvm -name rhel7 -S -machi
│   └─machine-qemu\x2dEAP6.scope
│       └─15290 /usr/bin/qemu-system-x86_64 -machine accel=kvm -name EAP6 -S -machin
├─user.slice
│   ├──user-0.slice
│   │   └─user@0.service
│   │       ├──3289 /usr/lib/systemd/systemd --user
│   │       └─3299 (sd-pam)
│   └─user-1000.slice
│       ├──session-7.scope
│       │   ├──13655 gdm-session-worker [pam/gdm-password]
│       │   ├──13665 /usr/bin/gnome-keyring-daemon --daemonize --login
│       │   ├──13710 gnome-session
│       │   ├──13718 dbus-launch --sh-syntax --exit-with-session
│       │   ├──13719 /bin/dbus-daemon --fork --print-pid 4 --print-address 6 --session
│       │   ├──13784 /usr/libexec/gvfsd
│       │   ├──13788 /usr/libexec//gvfsd-fuse /run/user/1000/gvfs -f -o big_writes
│       │   ├──13879 /usr/libexec/at-spi-bus-launcher
│       │   ├──13883 /bin/dbus-daemon --config-file=/etc/at-spi2/accessibility.conf --n
│       │   └─13887 /usr/libexec/at-spi2-registryd --use-gnome-session
└─lines 1-23
```

Gestion de ressources – Configuration

- `systemctl` peut configurer les attributs des cgroups
 - `systemctl set-property httpd.service CPUShares=2048`
 - Configuration persistante
- Ajouter `--runtime` pour ne pas avoir de persistance:
 - `systemctl set-property --runtime httpd.service \ CPUShares=2048`
- La configuration peut être placée dans un fichier d'unité
 - [Service]
 - CPUShares=2048

Gestion de ressources - CPU

- CPUAccounting=1 pour activer
- CPUShares – défaut 1024.
- Faire varier pour assigner plus ou moins de CPU à un service
 - CPUShares=1600

<https://www.kernel.org/doc/Documentation/scheduler/sched-design-CFS.txt>

Gestion de ressources - Mémoire

- MemoryAccounting=1 pour activer
- MemoryLimit et MemorySoftLimit
- Utiliser les suffixes K, M, G, T
 - MemoryLimit=1G

The idea behind soft limits is to allow control groups to use as much of the memory as needed, provided:

- a. There is no memory contention*
- b. They do not exceed their hard limit*

<https://www.kernel.org/doc/Documentation/cgroups/memory.txt>

Gestion de ressources – Block IO

- `BlockIOAccounting=1`
- `BlockIOWeight=` assigne un « poid » IO à un service spécifique
 - Similaire à `CPUshares`
 - Défaut 1000
 - Étendu 10 – 1000
 - Peut-être défini par block device ou point de montage
- `BlockIOReadBandwidth & BlockIOWriteBandwidth`
 - `BlockIOWriteBandwidth=/var/log 5M`

<https://www.kernel.org/doc/Documentation/cgroups/blkio-controller.txt>

Conversion d'« Init Scripts »

/etc/init.d/httpd

```
. /etc/rc.d/init.d/functions
if [ -f /etc/sysconfig/httpd ]; then
    . /etc/sysconfig/httpd
fi
HTTPD_LANG=${HTTPD_LANG-"C"}
INITLOG_ARGS=""
apachectl=/usr/sbin/apachectl
httpd=${HTTPD-/usr/sbin/httpd}
prog=httpd
pidfile=${PIDFILE-/var/run/httpd/httpd.pid}
lockfile=${LOCKFILE-/var/lock/subsys/httpd}
RETVAL=0
STOP_TIMEOUT=${STOP_TIMEOUT-10}
start() {
    echo -n "Starting $prog: "
    LANG=$HTTPD_LANG daemon --pidfile=${pidfile} $httpd $OPTIONS
    RETVAL=$?
    echo
    [ $RETVAL = 0 ] && touch ${lockfile}
    return $RETVAL
}
stop() {
    echo -n "Stopping $prog: "
    killproc -p ${pidfile} -d ${STOP_TIMEOUT} $httpd
    RETVAL=$?
    echo
    [ $RETVAL = 0 ] && rm -f ${lockfile} ${pidfile}
}
}
```

De RHEL 6.4; commentaires retirés

/etc/init.d/httpd - Suite

```
reload() {
    echo -n $"Reloading $prog: "
    if ! LANG=$HTTPD_LANG $httpd $OPTIONS -t >&/dev/null; then
        RETVAL=6
        echo $"not reloading due to configuration syntax error"
        failure $"not reloading $httpd due to configuration syntax error"
    else
        LSB=1 killproc -p ${pidfile} $httpd -HUP
        RETVAL=$?
        if [ $RETVAL -eq 7 ]; then
            failure $"httpd shutdown"
        fi
    fi
    echo
}

case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    status)
        status -p ${pidfile} $httpd
        RETVAL=$?
        ;;

```

/etc/init.d/httpd – Suite - Suite

```
restart)
    stop
    start
    ;;
condrestart|try-restart)
    if status -p ${pidfile} $httpd >&/dev/null; then
        stop
        start
    fi
    ;;
force-reload|reload)
    reload
    ;;
graceful|help|configtest|fullstatus)
    $apachectl $@
    RETVAL=$?
    ;;
*)
    echo $"Usage: $prog {start|stop|restart|condrestart|try-restart|force-reload|reload|status|fullstatus|graceful|help|
configtest}"
    RETVAL=2
esac
exit $RETVAL
```

En contraste avec les fichiers d'unités de systemd

Fichier d'unité – httpd.service

[Unit]

Description=The Apache HTTP Server

After=network.target remote-fs.target nss-lookup.target

[Service]

Type=notify

EnvironmentFile=/etc/sysconfig/httpd

ExecStart=/usr/sbin/httpd \$OPTIONS -DFOREGROUND

ExecReload=/usr/sbin/httpd \$OPTIONS -k graceful

ExecStop=/usr/sbin/httpd \$OPTIONS -k graceful-stop

KillSignal=SIGCONT

PrivateTmp=true

[Install]

WantedBy=multi-user.target

*Sans les commentaires

Fichier d'unité – Application personnalisée

[Unit]

Description=Description du service, pour les humains
After=syslog.target network.target

[Service]

ExecStart=/usr/sbin/[myapp] -D
Type=forking
PIDFile=/var/run/myapp.pid

[Install]

WantedBy=multi-user.target

Example

[Unit]

Description=JBoss Enterprise Application Platform

After=syslog.target network.target

[Service]

User=jboss-as

Environment=JBOSS_USER=jboss-as

Environment=JBOSS_HOME=/usr/local/EAP-6.1.1/jboss-eap-6.1

Environment=JBOSS_CONSOLE_LOG=/var/log/jbossas/console.log

ExecStart=/usr/local/EAP-6.1.1/jboss-eap-6.1/bin/standalone.sh -b 0.0.0.0

PIDFile=/var/run/jboss-as/jboss-as-standalone.pid

SyslogIdentifier=jboss-as

LimitNOFILE=102642

CPUShares=1600

Restart=always

Slice=jboss.slice

[Install]

WantedBy=multi-user.target

Example

```
root@host204:~  
File Edit View Search Terminal Help  
[root@host204 ~]# systemctl status jboss-as  
jboss-as.service - JBoss Enterprise Application Platform  
  Loaded: loaded (/etc/systemd/system/jboss-as.service; enabled)  
  Active: active (running) since Fri 2014-01-10 11:31:20 CST; 45s ago  
  Main PID: 692 (standalone.sh)  
  CGroup: /jboss.slice/jboss-as.service  
          └─ 692 /bin/sh /usr/local/EAP-6.1.1/jboss-eap-6.1/bin/standalone.s...  
             └─ 1095 java -D[Standalone] -server -XX:+UseCompressedOops -Xms1303...  
  
Jan 10 11:31:30 host204.local jboss-as[692]: 11:31:30,580 INFO [org.jboss.w...7  
Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,005 INFO [org.apache....0  
Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,036 INFO [org.apache....0  
Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,647 INFO [org.jboss.a...9  
Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,674 INFO [org.jboss.a...s  
Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,675 INFO [org.jboss.a...]  
Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,679 INFO [org.jboss.a...7  
Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,954 INFO [org.jboss.a...t  
Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,954 INFO [org.jboss.a...0  
Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,955 INFO [org.jboss.a...)  
[root@host204 ~]# █
```

Example

```
root@host204:~  
File Edit View Search Terminal Help  
├─jboss.slice  
│   └─jboss-as.service  
│       ├── 692 /bin/sh /usr/local/EAP-6.1.1/jboss-eap-6.1/bin/standalone.sh -b 0.0.0  
│       └─1095 java -D[Standalone] -server -XX:+UseCompressedOops -Xms1303m -Xmx1303  
├─user.slice  
│   └─user-0.slice  
│       └─session-1.scope  
│           ├── 1179 sshd: root@pts/0  
│           ├── 1185 -bash  
│           ├── 1216 systemd-cgls  
│           └─1217 systemd-cgls  
└─system.slice  
    ├── 1 /usr/lib/systemd/systemd --switched-root --system --deserialize 20  
    ├── polkit.service  
    │   └─512 /usr/lib/polkit-1/polkitd --no-debug  
    ├── auditd.service  
    │   └─389 /sbin/auditd -n  
    ├── systemd-udevd.service  
    │   └─343 /usr/lib/systemd/systemd-udevd  
    ├── lvm2-lvmetad.service  
    │   └─314 /usr/sbin/lvmetad  
    ├── systemd-journald.service  
    │   └─311 /usr/lib/systemd/systemd-journald  
lines 1-23
```


Fichier d'unité – Application personnalisée

- Copier le fichier d'unité
 - `cp [myapp].service /etc/systemd/system/`
- Informer systemd du changement:
 - `systemctl daemon-reload`
- Démarrer le service
 - `systemctl start [myapp].service`
- Vérifier l'état du service
 - `systemctl status [myapp].service`

Compatibilité avec les "Init Scripts"

Compatibilité avec les "Init Scripts"

- systemd est compatible à 99% avec les « init scripts » et les exceptions sont documentés
- La conversion des scripts en fichiers d'unités est encouragée mais non obligatoire
- Liste des incompatibilités :
<http://www.freedesktop.org/wiki/Software/systemd/Incompatibilities/>
- Conversion d'Init Scripts de SysV:
<http://0pointer.de/blog/projects/systemd-for-admins-3.html>

Le journal

Journal

- Ne remplace pas rsyslog dans RHEL 7
 - rsyslog est activé par défaut
 - rsyslog est utilisé pour la journalisation traditionnelle avec des fonctions d'entreprise
- Le journal n'est pas persistant par défaut, il contient l'information actuelle du système, depuis le démarrage
- Indexé
- Formaté
 - Erreurs en **rouge**
 - Avertissements en **gras**

Journal

- Collecte les métadonnées des événements
- Entrepose des données sous un format clé-valeur
 - man page: `systemd.journal-fields(7)`
- `journalctl` – utilitaire pour voir le journal
 - Filtres
 - Contenu lu avec "less" si non redirigé

Utilisation du journal

- journalctl (avec less)

```
root@host151:~  
File Edit View Search Terminal Help  
Oct 28 15:04:58 host151.local chronyd[329]: System clock wrong by -31.975399 seconds, adjustment  
Oct 28 15:04:26 host151.local chronyd[329]: System clock was stepped by -31.975 seconds  
Oct 28 15:04:26 host151.local systemd[1]: Time has been changed  
Oct 28 15:04:52 host151.local systemd[1]: Starting Stop Read-Ahead Data Collection...  
Oct 28 15:04:52 host151.local systemd[1]: Started Stop Read-Ahead Data Collection.  
Oct 28 15:05:32 host151.local chronyd[329]: Selected source 174.133.168.194  
Oct 28 15:06:08 host151.local sshd[2040]: Accepted password for root from 192.168.122.1 port 4512  
Oct 28 15:06:08 host151.local systemd[1]: Starting user-0.slice.  
Oct 28 15:06:08 host151.local systemd[1]: Created slice user-0.slice.  
Oct 28 15:06:08 host151.local systemd[1]: Starting User Manager for 0...  
Oct 28 15:06:08 host151.local systemd[1]: Starting Session 1 of user root.  
Oct 28 15:06:08 host151.local systemd[1]: Started Session 1 of user root.  
Oct 28 15:06:08 host151.local systemd-logind[322]: New session 1 of user root.  
Oct 28 15:06:08 host151.local sshd[2040]: pam_unix(sshd:session): session opened for user root by  
Oct 28 15:06:08 host151.local systemd[2044]: pam_unix(systemd-user:session): session opened for u  
Oct 28 15:06:08 host151.local systemd[2044]: Failed to open private bus connection: Failed to con  
Oct 28 15:06:08 host151.local systemd[2044]: Mounted /sys/kernel/config.  
Oct 28 15:06:08 host151.local systemd[2044]: Stopped target Sound Card.  
Oct 28 15:06:08 host151.local systemd[2044]: Starting Default.  
Oct 28 15:06:08 host151.local systemd[2044]: Reached target Default.  
Oct 28 15:06:08 host151.local systemd[2044]: Startup finished in 11ms.  
Oct 28 15:06:08 host151.local systemd[1]: Started User Manager for 0.  
lines 962-983/983 (END)
```

Utilisation du journal

- Activer la persistance: ``mkdir /var/log/journal``
 - Visualisé depuis un démarrage spécifique: ``journalctl -b``
- Tail -f et -n fonctionne tel qu'attendus:
 - `journalctl -f ; journalctl -n 50`
- Filtre par priorité: ``journalctl -p [level]``

0	emerg
1	alert
2	crit
3	err
4	warning
5	notice
6	debug

Utilisation du journal

- Autres filtres intéressants :
 - --since=yesterday ou YYYY-MM-DD (HH:MM:SS)
 - --until=YYYY-MM-DD
 - -u [unit]
 - Par exécutable: /usr/sbin/dnsmasq
 - Par fichier device : /dev/sda
- Voir le journal par champs :
 - journalctl [tab] [tab]
- Journal complet :
 - journalctl -o verbose

Avantages de systemd

- Meilleure standardisation entre les distributions
- Meilleure gestion des systèmes et des services
 - Gestion des dépendances
 - Surveillance des services (Autospawn, respawn)
 - Intégration aux cgroups pour la gestion des ressources
- systemd est simple à apprendre, plus simple pour les débutants
- Trucs : systemctl peut se connecter à des hôtes distants avec SSH "-H"
- systemd est rapide...

Ressources systemd

- Documentation RHEL 7
https://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/
- Page du projet systemd
<http://www.freedesktop.org/wiki/Software/systemd/>
- Le blog systemd de Lennart Poettering's (à lire)
<http://0pointer.de/blog/projects/systemd-for-admins-1.html>
- Red Hat System Administration II & III (RH134/RH254)
- FAQ
- Tips & Tricks

Sujets non couverts

- Snapshots
- Timer, paths
- Conteneurs
 - Systemd-nspawn