Apr 12 2015

How to manage processes with cgroup on Systemd

Articles Add comments

Share Like 0 Tweet Share 4

systemd is a suite of system management daemons, libraries, and utilities designed as a central management and configuration platform for the GNU/Linux computer operating system.
It provides a system and service manager that runs as PID 1 and starts the rest of the system as alternative to the traditional sysVinit.
systemd provides aggressive parallelization capabilities, uses socket and D-Bus activation for starting services, offers on-demand starting of daemons,

It's becoming the standard of all the major GNU/Linux distributions and at the moment it's the default for Arch Linux, Red Hat Enterprise/Centos (version 7), Fedora, Mageia and Suse Enterprise, it's planned to be used on Debian 8 and Ubuntu 15.04.

There is a lot of people talking for and against systemd on the net as some see it as too intrusive, complex and against the Unix philosophy to keep things simple and make them do just one task.

Using Red Hat 7 at work and Arch Linux on my laptop I've started to use it and I must agree that it's not so simple in the start, but let's try to take the good thing from it and in this article I'd like to show you some commands that you can use with systemd to manage the processes on a GNU/Linux system and that I've found really useful.

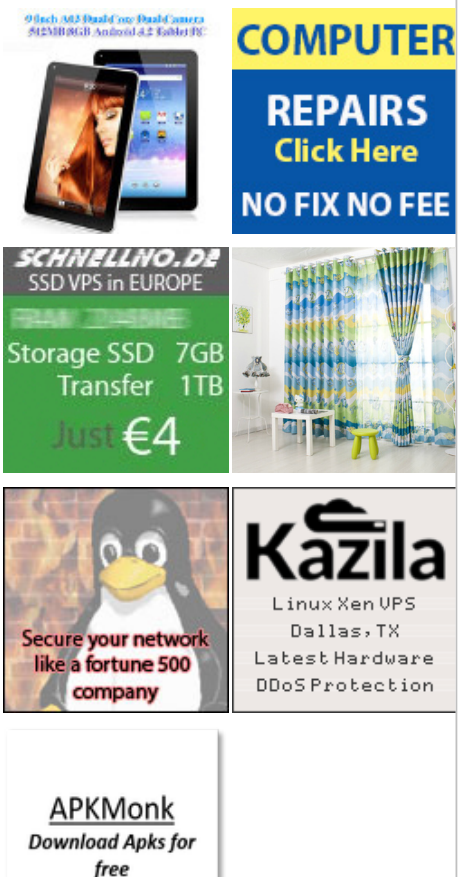
Processes and cgroups

systemd organizes processes with cgroups, this is a Linux kernel feature to limit, police and account the resource usage of certain processes (actually

Language:
English Italiano

Linuxaria
3K likes
Like Page

Sponsor



This website uses cookies to improve your experience. We'll assume you're ok with this, but you can opt-out if you wish. Accept
Read More

Control groups can be used in multiple ways:

- create and manage them on the fly using tools like `cgcreate`, `cgexec`, `cgclassify` etc
- the “rules engine daemon”, to automatically move certain users/groups/commands to groups (`/etc/cgrules.conf` and `/usr/lib/systemd/system/cgconfig.service`)
- through other software such as Linux Containers (LXC) virtualization

So Control Groups are two things: **(A)** *a way to hierarchally group and label processes*, and **(B)** *a way to then apply resource limits* to these groups. systemd only requires the former (A), and not the latter (B).

You can see the sue of cgroups with the ps command, which has been updated to show cgroups. Run this command to see which service owns which processes:

```
$ ps xawf -eo pid,user,cgroup,args
  PID USER     CGROUP                                  COMMAND
    2 root     -                                       [kthrea
    3 root     -                                        \_ [ks
[...]
 4281 root     -                                        \_ [fl
    1 root     name=systemd:/systemd-1                 /sbin/i
  455 root     name=systemd:/systemd-1/sysinit.service /sb
28188 root     name=systemd:/systemd-1/sysinit.service  \_
28191 root     name=systemd:/systemd-1/sysinit.service  \_
 1096 dbus     name=systemd:/systemd-1/dbus.service /bin/d
 1131 root     name=systemd:/systemd-1/auditd.service audi
 1133 root     name=systemd:/systemd-1/auditd.service  \_
 1135 root     name=systemd:/systemd-1/auditd.service
 1171 root     name=systemd:/systemd-1/NetworkManager.serv
 4028 root     name=systemd:/systemd-1/NetworkManager.serv
 1175 avahi    name=systemd:/systemd-1/avahi-daemon.servic
 1194 avahi    name=systemd:/systemd-1/avahi-daemon.servic
 1193 root     name=systemd:/systemd-1/rsyslog.service /sb
 1195 root     name=systemd:/systemd-1/cups.service cupsd
 1207 root     name=systemd:/systemd-1/mdmonitor.service m
 1210 root     name=systemd:/systemd-1/irqbalance.service
 1216 root     name=systemd:/systemd-1/dbus.service /usr/s
 1219 root     name=systemd:/systemd-1/dbus.service /usr/l
 1242 root     name=systemd:/systemd-1/dbus.service /usr/s
 1249 68       name=systemd:/systemd-1/haldaemon.service h
 1250 root     name=systemd:/systemd-1/haldaemon.service
 1273 root     name=systemd:/systemd-1/haldaemon.service
 1275 root     name=systemd:/systemd-1/haldaemon.service
 1284 root     name=systemd:/systemd-1/haldaemon.service
 1285 root     name=systemd:/systemd-1/haldaemon.service
 1287 68       name=systemd:/systemd-1/haldaemon.service
 1317 root     name=systemd:/systemd-1/abrtd.service /usr/
 1332 root     name=systemd:/systemd-1/getty@.service/tty2
 1339 root     name=systemd:/systemd-1/getty@.service/tty3
 1342 root     name=systemd:/systemd-1/getty@.service/tty5
 1343 root     name=systemd:/systemd-1/getty@.service/tty4
 1344 root     name=systemd:/systemd-1/crond.service cronc
.....
```

Follow Me

Enter email address

Subscribe

447 readers
BY FEEDBURNER

Popular posts

None Found

Subscribe by Email

Subscribe to RSS» English by mail

Iscriviti agli RSS» Italiano via Email

Recent Comments

Ashwin on What You Don’t Know About Linux Open Source Could Be Costing to More Than You Think

frann on gEdit, an easy to use text editor with many advanced features

greg125 on The best way to move data

Ashwin on Top Five Mobile Devices That Run Linux

Ashwin on Paas and continuos integration

My scoop.it feed

This website uses cookies to improve your experience. We'll assume you're ok with this, but you can opt-out if you wish. Accept

Read More

1/sysinit.service cgroup, which is where systemd places all processes started by the sysinit.service service, which covers early boot.

A different way to present the same information is the systemd-cgls tool that is shipped with systemd. It shows the cgroup hierarchy in a pretty tree. Its output looks like this:

```
$ systemd-cgls
+    2 [kthreadd]
[...]
+ 4281 [flush-8:0]
+ user
| \ lennart
|   \ 1
|     +  1495 pam: gdm-password
|     +  1521 gnome-session
|     +  1534 dbus-launch --sh-syntax --exit-with-session
|     +  1535 /bin/dbus-daemon --fork --print-pid 5 --prin
|     +  1603 /usr/libexec/gconfd-2
|     +  1612 /usr/libexec/gnome-settings-daemon
|     +  1615 /ushr/libexec/gvfsd
|     +  1621 metacity
|     +  1626 /usr/libexec//gvfs-fuse-daemon /home/lennart
|     +  1634 /usr/bin/pulseaudio --start --log-target=sys
|     +  1635 gnome-panel
|     +  1638 nautilus
|     +  1640 /usr/libexec/polkit-gnome-authentication-age
|     +  1641 /usr/bin/seapplet
|     +  1644 gnome-volume-control-applet
|     +  1645 /usr/libexec/bonobo-activation-server --ac-a
|     +  1646 /usr/sbin/restorecond -u
|     +  1649 /usr/libexec/pulse/gconf-helper
|     +  1652 /usr/bin/devilspie
|     +  1662 nm-applet --sm-disable
|     +  1664 gnome-power-manager
|     +  1665 /usr/libexec/gdu-notification-daemon
|     +  1668 /usr/libexec/im-settings-daemon
|     +  1670 /usr/libexec/evolution/2.32/evolution-alarm-
|     +  1672 /usr/bin/python /usr/share/system-config-pri
|     +  1674 /usr/lib64/deja-dup/deja-dup-monitor
|     +  1675 abrt-applet
|     +  1677 bluetooth-applet
|     +  1678 gpk-update-icon
|     +  1701 /usr/libexec/gvfs-gdu-volume-monitor
|     +  1707 /usr/bin/gnote --panel-applet --oaf-activate
|     +  1725 /usr/libexec/clock-applet
|     +  1727 /usr/libexec/wnck-applet
|     +  1729 /usr/libexec/notification-area-applet
|     +  1759 gnome-screensaver
|     +  1780 /usr/libexec/gvfsd-trash --spawner :1.9 /org
|     +  1864 /usr/libexec/gvfs-afc-volume-monitor
|     +  1874 /usr/libexec/gconf-im-settings-daemon
|     +  1882 /usr/libexec/gvfs-gphoto2-volume-monitor
|     +  1903 /usr/libexec/gvfsd-burn --spawner :1.9 /org/
|     +  1909 gnome-terminal
|     +  1913 gnome-pty-helper
|     +  1914 bash
|     +  1968 ssh-agent
|     +  1994 gpg-agent --daemon --write-env-file
|     +  2221 bash
|     +  2461 bash
|     +  4193 ssh tango
```

This website uses cookies to improve your experience. We'll assume you're ok with this, but you can opt-out if you wish. Accept

Read More

```
|     + 27251 empathy
|     + 27262 /usr/libexec/mission-control-5
|     + 27265 /usr/libexec/telepathy-haze
|     + 27268 /usr/libexec/telepathy-logger
|     + 27270 /usr/libexec/dconf-service
|     + 27280 /usr/libexec/notification-daemon
|     + 27284 /usr/libexec/telepathy-gabble
|     + 27285 /usr/libexec/telepathy-salut
|     + 27297 /usr/libexec/geoclue-yahoo
|     + 28900 /usr/lib64/nspluginwrapper/npviewer.bin --pl
|     + 29219 emacs systemd-for-admins-1.txt
|     + 29231 ssh tango
|     \ 29519 systemd-cgls
\ systemd-1
  + 1 /sbin/init
  + ntpd.service
  | \ 4112 /usr/sbin/ntpd -n -u ntp:ntp -g
  + systemd-logger.service
  | \ 1499 /lib/systemd/systemd-logger
  + accounts-daemon.service
  | \ 1496 /usr/libexec/accounts-daemon
  + rtkit-daemon.service
  | \ 1473 /usr/libexec/rtkit-daemon
  + console-kit-daemon.service
  | \ 1408 /usr/sbin/console-kit-daemon --no-daemon
  + prefdm.service
  | + 1376 /usr/sbin/gdm-binary -nodaemon
  | + 1391 /usr/libexec/gdm-simple-slave --display-id /org
  | + 1394 /usr/bin/Xorg :0 -nr -verbose -auth /var/run/gd
  | + 1419 /usr/bin/dbus-launch --exit-with-session
  | \ 1511 /usr/bin/gnome-keyring-daemon --daemonize --log
  + getty@.service
  | + tty6
  | | \ 1346 /sbin/mingetty tty6
  | + tty4
  | | \ 1343 /sbin/mingetty tty4
  | + tty5
  | | \ 1342 /sbin/mingetty tty5
  | + tty3
  | | \ 1339 /sbin/mingetty tty3
  | \ tty2
  |   \ 1332 /sbin/mingetty tty2
  + abrtd.service
  | \ 1317 /usr/sbin/abrtd -d -s
  + crond.service
  | \ 1344 crond
  + sshd.service
  | \ 1362 /usr/sbin/sshd
  + sendmail.service
  | + 4094 sendmail: Queue runner@01:00:00 for /var/spool/
  | \ 4096 sendmail: accepting connections
  + auditd.service
  | + 1131 auditd
  | + 1133 /sbin/audispd
  | \ 1135 /usr/sbin/sedispatch
....
```

this command shows the processes by their cgroup and hence service, as systemd labels the cgroups after the services. For example, you can easily see that the auditing service auditd.service spawns three individual processes, auditd, audisp and sedispatch.

This website uses cookies to improve your experience. We'll assume you're ok with this, but you can opt-out if you wish. Accept

Read More

all the processes of the service auditd you can use the command:

```
# systemctl kill auditd.service
```

This will ensure that SIGTERM is delivered to all processes of the auditd service, not just the main process. Of course, you can also send a different signal if you wish.

```
# systemctl kill -s SIGKILL auditd.service
```

Sometimes all you need is to send a specific signal to the main process of a service, maybe because you want to trigger a reload via SIGHUP. Instead of going via the PID file, here's an easier way to do this:

```
# systemctl kill -s HUP –kill-who=main crond.service
```

And in my point of view this is great !
No more ps -ef |grep something| awk something |kill the result of the awk 🙂

Analyzing Resource usage

To understand the resource usage of all services, the systemd developers created the tool systemd-cgtop, that will enumerate all cgroups of the system, determine their resource usage (CPU, Memory, and IO) and present them in a top-like fashion. Building on the fact that systemd services are managed in cgroups this tool hence can present to you for services what top shows you for processes.

Unfortunately, by default `cgtop` will only be able to chart CPU usage per-service for you, IO and Memory are only tracked as total for the entire machine. The reason for this is simply that by default there are no per-service cgroups in the `blkio` and `memory` controller hierarchies but that's what is needed to determine the resource usage.

If resource monitoring for these resources is required it is recommended to add blkio and memory to the DefaultControllers= setting in /etc/systemd/system.conf (see systemd.conf(5) for details). Alternatively, it is possible to enable resource accounting individually for services, by making use of the ControlGroup= option in the unit files (See systemd.exec(5) for details).

To emphasize this: unless **blkio** and **memory** are enabled for the services in question with either of the options suggested above no resource accounting will be available for system services and the data shown by systemd-cgtop will be

This website uses cookies to improve your experience. We'll assume you're ok with this, but you can opt-out if you wish. Accept
Read More

This recording has been archived

All unclaimed recordings (the ones not linked to any user account) are automatically archived 7 days after upload.

Resources

Managing Services on Linux with systemd

From the creator of systemd:

- Rethinking PID 1
- systemd for Administrators, Part 1
- systemd for Administrators, Part II
- systemd for Administrators, Part III
- systemd for Administrators, Part IV
- systemd for Administrators, Part V
- systemd for Administrators, Part VI
- systemd for Administrators, Part VII
- systemd for Administrators, Part VIII
- systemd for Administrators, Part IX
- systemd for Administrators, Part X
- systemd for Administrators, Part XI

Popular Posts:

None Found

Share Like 0 Tweet Share 4

Flattr this!

Posted by Riccardo at 22:05 Tagged with: arch linux, cgroups, operating system, pid, systemd, sysvinit

2 Responses to “How to manage processes with cgroup on Systemd”

1. **Heraclius** says:
   Monday April 13th, 2015 at 09:01 PM

   What? No comments in three days? And we were told that systemd was soooo popular an very much appreciated by Linux users worldwide!

This website uses cookies to improve your experience. We'll assume you're ok with this, but you can opt-out if you wish. Accept

Read More

2. **Felipe Azevedo** says:
Wednesday February 15th, 2017 at 02:36 AM

Great article!
One question, though: when I create a service through systemd, does it automatically generates its own cgroup? I am curious about it because I configured microservices (2 standalone jars) to run as services (using systemd). When I execute systemctl -l status service1 It outputs some description showing a main PID, and a cgroup tree, describing the service file path I configured, a sh file path that I also created and finally the jar file path I am running as a microservices.
Does this microservice, in this cenario, have isolated resources as in a container approach (such as Docker)?
Thanks!!

Reply

Leave a Reply

Your Comment

You may use these HTML tags and attributes: `<a href="" title=""> <abbr title=""> <acronym title=""> <b> <blockquote cite=""> <cite> <code> <del datetime=""> <em> <i> <q cite=""> <s> <strike> <strong>`

Name (required)

E-mail (required)

URI

CAPTCHA Code
*

Submit Comment

Introduction to gnome maps The ultime guide of linux logging

This website uses cookies to improve your experience. We'll assume you're ok with this, but you can opt-out if you wish. Accept
Read More

© 2013 Linuxaria

All site content, except where otherwise noted, is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License.

Suffusion theme by Sayontan Sinha

This website uses cookies to improve your experience. We'll assume you're ok with this, but you can opt-out if you wish. Accept

Read More