Pid Eins

レナート
لينارت

POSTED ON MI 02 MÄRZ 2011

systemd for Administrators, Part V

It has been a while since the last installment of my systemd for Administrators series, but now with the release of Fedora 15 based on systemd looming, here's a new episode:

The Three Levels of "Off"

In systemd, there are three levels of turning off a service (or other unit). Let's have a look which those are:

1. You can **stop** a service. That simply terminates the running instance of the service and does little else. If due to some form of activation (such as manual activation, socket activation, bus activation, activation by system boot or activation by hardware plug) the service is requested again afterwards it will be started. Stopping a service is hence a very simple, temporary and superficial operation. Here's an example how to do this for the NTP service:

   ```
   $ systemctl stop ntpd.service
   ```

   This is roughly equivalent to the following traditional command which is available on most SysV inspired systems:

   ```
   $ service ntpd stop
   ```

   In fact, on Fedora 15, if you execute the latter command it will be transparently converted to the former.

2. You can **disable** a service. This unhooks a service from its activation triggers. That means, that depending on your service it will no longer be activated on boot, by socket or bus activation or by hardware plug (or any other trigger that applies to it). However, you can still start it manually if you wish. If there

is already a started instance disabling a service will *not* have the effect of stopping it. Here's an example how to disable a service:

```
$ systemctl disable ntpd.service
```

On traditional Fedora systems, this is roughly equivalent to the following command:

```
$ chkconfig ntpd off
```

And here too, on Fedora 15, the latter command will be transparently converted to the former, if necessary.

Often you want to combine stopping and disabling a service, to get rid of the current instance and make sure it is not started again (except when manually triggered):

```
$ systemctl disable ntpd.service
$ systemctl stop ntpd.service
```

Commands like this are for example used during package deinstallation of systemd services on Fedora.

Disabling a service is a permanent change; until you undo it it will be kept, even across reboots.

3. You can **mask** a service. This is like disabling a service, but on steroids. It not only makes sure that service is not started automatically anymore, but even ensures that a service cannot even be started manually anymore. This is a bit of a hidden feature in systemd, since it is not commonly useful and might be

confusing the user. But here's how you do it:

```
$ ln -s /dev/null /etc/systemd/system/ntpd.service
$ systemctl daemon-reload
```

By symlinking a service file to `/dev/null` you tell systemd to never start the service in question and completely block its execution. Unit files stored in `/etc/systemd/system` override those from `/lib/systemd/system` that carry the same name. The former directory is administrator territory, the latter terroritory of your package manager. By installing your symlink in `/etc/systemd/system/ntpd.service` you hence make sure that systemd will never read the upstream shipped service file `/lib/systemd/system/ntpd.service`.

systemd will recognize units symlinked to `/dev/null` and show them as *masked*. If you try to start such a service manually (via `systemctl start` for example) this will fail with an error.

A similar trick on SysV systems does not (officially) exist. However, there are a few unofficial hacks, such as editing the init script and placing an `exit 0` at the top, or removing its execution bit. However, these solutions have various drawbacks, for example they interfere with the package manager.

Masking a service is a permanent change, much like disabling a service.

Now that we learned how to turn off services on three levels, there's only one question left: how do we turn them on again? Well, it's quite symmetric. use `systemctl start` to undo `systemctl stop`. Use `systemctl enable` to undo `systemctl disable` and use `rm` to undo `ln`.

And that's all for now. Thank you for your attention!

Category: projects

← BACK TO INDEX