La ejecución de comandos post-arranque en Debian 8 era tan fácil como agregar la ruta del script o del comando dentro de **/etc/rc.local**, antes de exit 0, pero si acabas de instalar Debian 9 encontrarás que no existe tal archivo. ¡Que no panda el cúnico!. Todavía podemos hacer uso de /etc/rc.local, sólo hay que hacer unos pequeños cambios en el sistema:

Para activarlo:

```
echo "[Unit]" > /etc/systemd/system/rc-local.service
echo "Description=/etc/rc.local Compatibility" >> /etc/systemd/system/rc-local.service
echo "ConditionPathExists=/etc/rc.local" >> /etc/systemd/system/rc-local.service
echo "" >> /etc/systemd/system/rc-local.service
echo "[Service]" >> /etc/systemd/system/rc-local.service
echo "Type=forking" >> /etc/systemd/system/rc-local.service
echo "ExecStart=/etc/rc.local start" >> /etc/systemd/system/rc-local.service
echo "TimeoutSec=0" >> /etc/systemd/system/rc-local.service
echo "StandardOutput=tty" >> /etc/systemd/system/rc-local.service
echo "RemainAfterExit=yes" >> /etc/systemd/system/rc-local.service
echo "SysVStartPriority=99" >> /etc/systemd/system/rc-local.service
echo "" >> /etc/systemd/system/rc-local.service
echo "[Install]" >> /etc/systemd/system/rc-local.service
echo "WantedBy=multi-user.target" >> /etc/systemd/system/rc-local.service
echo '#!/bin/bash' > /etc/rc.local
echo "" >> /etc/rc.local
echo "exit 0" >> /etc/rc.local
chmod +x /etc/rc.local
systemctl enable rc-local
systemctl start rc-local.service
systemctl status rc-local.service
```

Después de esos comandos ya podemos agregar comandos a /etc/rc.local en las líneas anteriores a «exit 0» y se ejecutarán correctamente en el próximo re-arranque sin tener que acudir al @reboot de cron.