

[linuxbabe.com](https://www.linuxbabe.com)

How to Enable /etc/rc.local with Systemd - LinuxBabe

Xiao Guoan (Admin)

5-6 minutes

If you are running a Linux distro that uses Systemd, then you may find that your command in `/etc/rc.local` file would not run on system boot. This guide explains how to enable `/etc/rc.local` script to run on system startup.

Enable /etc/rc.local on Systemd

If you type the following command in terminal:

```
sudo systemctl status rc-local
```

You may get this output:

```
● rc-local.service - /etc/rc.local Compatibility
   Loaded: loaded (/lib/systemd/system/rc-local.service;
   static; vendor preset: enabled)
   Active: failed (Result: exit-code) since Thu
2015-11-26 23:54:58 CST; 59s ago
     Process: 1001 ExecStart=/etc/rc.local start
   (code=exited, status=1/FAILURE)
Nov 26 23:54:57 vivid rc.local[1001]: File "/usr/lib
/python2.7/dist-packages/pkg_resources/__init__.py",
line 920, in require
Nov 26 23:54:57 vivid rc.local[1001]: needed =
self.resolve(parse_requirements(requirements))
Nov 26 23:54:57 vivid rc.local[1001]: File "/usr/lib
/python2.7/dist-packages/pkg_resources/__init__.py",
line 807, in resolve
```

```
Nov 26 23:54:57 vivid rc.local[1001]: raise
DistributionNotFound(req)
Nov 26 23:54:57 vivid rc.local[1001]:
pkg_resources.DistributionNotFound: shadowsocks==2.8.2
Nov 26 23:54:58 vivid sudo[1008]:
pam_unix(sudo:session): session closed for user root
Nov 26 23:54:58 vivid systemd[1]: rc-local.service:
control process exited, code=exited status=1
Nov 26 23:54:58 vivid systemd[1]: Failed to start
/etc/rc.local Compatibility.
Nov 26 23:54:58 vivid systemd[1]: Unit rc-local.service
entered failed state.
Nov 26 23:54:58 vivid systemd[1]: rc-local.service
failed.
```

And if you try to enable /etc/rc.local to run on system boot with the command:

```
sudo systemctl enable rc-local
```

You may get:

The unit files have no [Install] section. They are not meant to be enabled

using systemctl.

Possible reasons for having this kind of units are:

1) A unit may be statically enabled by being symlinked from another unit's

.wants/ or .requires/ directory.

2) A unit's purpose may be to act as a helper for some other unit which has

a requirement dependency on it.

3) A unit may be started when needed via activation (socket, path, timer,

D-Bus, udev, scripted systemctl call, ...).

The solution

As you can see from above, The unit file have no [Install] section. As such Systemd can not enable it. First we need to create a file:

```
sudo nano /etc/systemd/system/rc-local.service
```

Then add the following content to it.

```
[Unit]
Description=/etc/rc.local Compatibility
ConditionPathExists=/etc/rc.local

[Service]
Type=forking
ExecStart=/etc/rc.local start
TimeoutSec=0
StandardOutput=tty
RemainAfterExit=yes
SysVStartPriority=99

[Install]
WantedBy=multi-user.target
```

Save and close the file. To save a file in Nano text editor, press `Ctrl+O`, then press `Enter` to confirm. To exit the file, Press `Ctrl+X`. Next, run the following command to make sure `/etc/rc.local` file is executable.

```
sudo chmod +x /etc/rc.local
```

Note: Starting with 16.10, Ubuntu doesn't ship with `/etc/rc.local` file anymore. You can create the file by executing this command.

```
printf '%s\n' '#!/bin/bash' 'exit 0' | sudo tee -a
/etc/rc.local
```

Then add execute permission to `/etc/rc.local` file.

```
sudo chmod +x /etc/rc.local
```

After that, enable the service on system boot:

```
sudo systemctl enable rc-local
```

Output:

```
Created symlink from /etc/systemd/system/multi-
user.target.wants/rc-local.service to /etc/systemd
/system/rc-local.service.
```

Now start the service and check its status:

```
sudo systemctl start rc-local.service
sudo systemctl status rc-local.service
```

Output:

```
● rc-local.service - /etc/rc.local Compatibility
   Loaded: loaded (/etc/systemd/system/rc-local.service;
   enabled; vendor preset: enabled)
   Active: active (running) since Fri 2015-11-27 00:32:56
   CST; 14min ago
     Process: 879 ExecStart=/etc/rc.local start
   (code=exited, status=0/SUCCESS)
    Main PID: 880 (watch)
     CGroup: /system.slice/rc-local.service
```

Cron @reboot

If the above method does not work for you, or you just want some simple commands to be executed on system boot, then you can also use the @reboot feature in cron to automatically execute command on system boot. For example, I want my [shadowsocks client](#) to auto start, so I open the root user's cron file:

```
sudo crontab -e
```

And put the following line at the end of it.

```
@reboot /usr/bin/sslocal -c /etc/shadowsocks.json -d
start
```

Save and close the file.

In some Linux distributions such as archlinux, the cron daemon is not enabled by default. So you have to manually enable it. To enable it on archlinux, enter the following command in the terminal.

```
sudo systemctl enable cronie
```

Shadowsocks is a socks5 proxy that can be used to bypass Internet firewalls, If you are interested, click the link below to learn how to setup your own shadowsocks server.

[Setup your own shadowsocks server on Debian, Ubuntu and Centos](#)

Rate this tutorial

[Total: 44 Average: 3.8]