# [Centos7 自定义systemctl服务脚本](https://www.cnblogs.com/wang-yc/p/8876155.html)

**序言篇：**

　　之前工作环境一直使用Centos6版本，脚本一直在使用/etc/init.d/xxx；系统升级到Cento7后，虽然之前的启动脚本也可以使用，但一直没有使用systemctl 的自定义脚本。

本篇文章用于总结下，具体的使用方式。Centos7 开机第一程序从init完全换成了systemd的启动方式，而systemd依靠unit的方式来控制开机服务，开机级别等功能。

**应用篇：**

　　Centos7的服务systemctl 脚本一般存放在：/usr/lib/systemd , 目录下又有user和system之分

* + /usr/lib/systemd/system   **# 系统服务，开机不需要登录就能运行的程序（相当于开机自启）**
  + /usr/lib/systemd/user       **# 用户服务，需要登录后才能运行的程序**

　　目录下又存在两种类型的文件：

* + \*.service   # 服务unit文件
  + \*.target     # 开机级别unit

　　CentOS7的每一个服务以.service结尾，一般会分为3部分：[Unit]、[Service]和[Install]

[复制代码](javascript:void(0);)

vim /usr/lib/systemd/system/xxx.service

[Unit] # 主要是服务说明

Description=test # 简单描述服务

After=network.target # 描述服务类别，表示本服务需要在network服务启动后在启动

Before=xxx.service # 表示需要在某些服务启动之前启动，After和Before字段只涉及启动顺序，不涉及依赖关系。

[Service] # 核心区域

Type=forking # 表示后台运行模式。

User=user # 设置服务运行的用户

Group=user # 设置服务运行的用户组

KillMode=control-group # 定义systemd如何停止服务

PIDFile=/usr/local/test/test.pid # 存放PID的绝对路径

Restart=no # 定义服务进程退出后，systemd的重启方式，默认是不重启

ExecStart=/usr/local/test/bin/startup.sh # 服务启动命令，命令需要绝对路径

PrivateTmp=true # 表示给服务分配独立的临时空间

[Install]

WantedBy=multi-user.target # 多用户

[复制代码](javascript:void(0);)

**字段说明：**

[复制代码](javascript:void(0);)

Type的类型有：

simple(默认）：# 以ExecStart字段启动的进程为主进程

forking: # ExecStart字段以fork()方式启动，此时父进程将退出，子进程将成为主进程（后台运行）。一般都设置为forking

oneshot: # 类似于simple，但只执行一次，systemd会等它执行完，才启动其他服务

dbus： # 类似于simple, 但会等待D-Bus信号后启动

notify: # 类似于simple, 启动结束后会发出通知信号，然后systemd再启动其他服务

idle： # 类似于simple，但是要等到其他任务都执行完，才会启动该服务。

EnvironmentFile:

指定配置文件，和连词号组合使用，可以避免配置文件不存在的异常。

Environment:

后面接多个不同的shell变量。

例如：

Environment=DATA\_DIR=/data/elk

Environment=LOG\_DIR=/var/log/elasticsearch

Environment=PID\_DIR=/var/run/elasticsearch

EnvironmentFile=-/etc/sysconfig/elasticsearch

连词号（-）：在所有启动设置之前，添加的变量字段，都可以加上连词号

表示抑制错误，即发生错误时，不影响其他命令的执行。

比如`EnviromentFile=-/etc/sysconfig/xxx` 表示即使文件不存在，也不会抛异常

KillMode的类型：

control-group(默认)：# 当前控制组里的所有子进程，都会被杀掉

process: # 只杀主进程

mixed: # 主进程将收到SIGTERM信号，子进程收到SIGKILL信号

none: # 没有进程会被杀掉，只是执行服务的stop命令

Restart的类型：

no(默认值)： # 退出后无操作

on-success: # 只有正常退出时（退出状态码为0）,才会重启

on-failure: # 非正常退出时，重启，包括被信号终止和超时等

on-abnormal: # 只有被信号终止或超时，才会重启

on-abort: # 只有在收到没有捕捉到的信号终止时，才会重启

on-watchdog: # 超时退出时，才会重启

always: # 不管什么退出原因，都会重启

# 对于守护进程，推荐用on-failure

RestartSec字段：

表示systemd重启服务之前，需要等待的秒数：RestartSec: 30

各种Exec\*字段：

# Exec\* 后面接的命令，仅接受“指令 参数 参数..”格式，不能接受<>|&等特殊字符，很多bash语法也不支持。如果想支持bash语法，需要设置Tyep=oneshot

ExecStart： # 启动服务时执行的命令

ExecReload： # 重启服务时执行的命令

ExecStop： # 停止服务时执行的命令

ExecStartPre： # 启动服务前执行的命令

ExecStartPost：# 启动服务后执行的命令

ExecStopPost： # 停止服务后执行的命令

WantedBy字段：

multi-user.target: # 表示多用户命令行状态，这个设置很重要

graphical.target: # 表示图形用户状体，它依赖于multi-user.target

[复制代码](javascript:void(0);)

**systemctl 命令**

systemctl daemon-reload # 重载系统服务

systemctl enable \*.service # 设置某服务开机启动

systemctl start \*.service # 启动某服务

systemctl stop \*.service # 停止某服务

systemctl reload \*.service # 重启某服务