



马哥教育
最专业的Linux培训机构

管理systemd

- ❖ CentOS7启动
- ❖ Unit介绍
- ❖ 服务管理和查看
- ❖ 启动排错
- ❖ 破解口令
- ❖ 修复grub2

马哥教育

www.magedu.com

❖ POST --> Boot Sequence --> Bootloader --> kernel +
initramfs(initrd) --> rootfs --> /sbin/init

init: CentOS 5: SysV init

CentOS 6: Upstart

CentOS 7: Systemd

❖ **Systemd**: 系统启动和服务守护进程管理器，负责在系统启动或运行时，激活系统资源，服务器进程和其它进程

❖ **Systemd**新特性:

系统引导时实现服务并行启动

按需启动守护进程

自动化的服务依赖关系管理

同时采用socket式与D-Bus总线式激活服务

系统状态快照

❖ 核心概念：unit

unit表示不同类型的**systemd**对象，通过配置文件进行标识和配置；文件中主要包含了系统服务、监听**socket**、保存的系统快照以及其它与**init**相关的信息

❖ 配置文件：

/usr/lib/systemd/system: 每个服务最主要的启动脚本设置，类似于之前的**/etc/init.d/**

/run/systemd/system: 系统执行过程中所产生的服务脚本，比上面目录优先运行

/etc/systemd/system: 管理员建立的执行脚本，类似于**/etc/rc.d/rcN.d/Sxx**类的功能，比上面目录优先运行

- ❖ **Systemctl -t help** 查看unit类型
- ❖ **Service unit**: 文件扩展名为**.service**, 用于定义系统服务
- ❖ **Target unit**: 文件扩展名为**.target**, 用于模拟实现“运行级别”
- ❖ **Device unit**: **.device**, 用于定义内核识别的设备
- ❖ **Mount unit**: **.mount**, 定义文件系统挂载点
- ❖ **Socket unit**: **.socket**, 用于标识进程间通信用的**socket**文件, 也可在系统启动时, 延迟启动服务, 实现按需启动
- ❖ **Snapshot unit**: **.snapshot**, 管理系统快照
- ❖ **Swap unit**: **.swap**, 用于标识**swap**设备
- ❖ **Automount unit**: **.automount**, 文件系统的自动挂载点
- ❖ **Path unit**: **.path**, 用于定义文件系统中的**一个文件或目录使用**, 常用于当文件系统变化时, 延迟激活服务, 如: **spool** 目录

❖ 关键特性:

基于**socket**的激活机制: **socket**与服务程序分离

基于**d-bus**的激活机制:

基于**device**的激活机制:

基于**path**的激活机制:

系统快照: 保存各**unit**的当前状态信息于持久存储设备中
向后兼容**sysv init**脚本

❖ 不兼容:

systemctl命令固定不变, 不可扩展

非由**systemd**启动的服务, **systemctl**无法与之通信和控制

❖ 管理系统服务：

CentOS 7: service unit

注意：能兼容早期的服务脚本

- ❖ 命令: `systemctl COMMAND name.service`
- ❖ 启动: `service name start ==> systemctl start name.service`
- ❖ 停止: `service name stop ==> systemctl stop name.service`
- ❖ 重启: `service name restart ==> systemctl restart name.service`
- ❖ 状态: `service name status ==> systemctl status name.service`

- ❖ 条件式重启：已启动才重启，否则不做操作

`service name condrestart ==> systemctl try-restart name.service`

- ❖ 重载或重启服务：先加载，再启动

`systemctl reload-or-restart name.service`

- ❖ 重载或条件式重启服务：

`systemctl reload-or-try-restart name.service`

- ❖ 禁止自动和手动启动：

`systemctl mask name.service`

- ❖ 取消禁止：

`systemctl unmask name.service`

- ❖ 查看某服务当前激活与否的状态:

`systemctl is-active name.service`

- ❖ 查看所有已经激活的服务:

`systemctl list-units --type|-t service`

- ❖ 查看所有服务:

`systemctl list-units --type service -all|-a`

- ❖ `chkconfig`命令的对应关系:

- ❖ 设定某服务开机自启:

`chkconfig name on ==> systemctl enable name.service`

- ❖ 设定某服务开机禁止启动:

`chkconfig name off ==> systemctl disable name.service`

- ❖ 查看所有服务的开机自启状态：

```
chkconfig --list ==> systemctl list-unit-files --  
type service
```

- ❖ 用来列出该服务在哪些运行级别下启用和禁用

```
chkconfig sshd -list ==>  
ls /etc/systemd/system/*.wants/sshd.service
```

- ❖ 查看服务是否开机自启：

```
systemctl is-enabled name.service
```

- ❖ 其它命令：

查看服务的依赖关系：

```
systemctl list-dependencies name.service
```

- ❖ 杀掉进程：

```
systemctl kill 进程名
```

- ❖ `systemctl list-units --type service --all`显示状态
- ❖ `loaded`:Unit配置文件已处理
- ❖ `active(running)`:一次或多次持续处理的运行
- ❖ `active(exited)`:成功完成一次性的配置
- ❖ `active(waiting)`:运行中，等待一个事件
- ❖ `inactive`:不运行
- ❖ `enabled`:开机启动
- ❖ `disabled`:开机不启动
- ❖ `static`:开机不启动，但可被另一个启用的服务激活

- ❖ 显示所有单元状态

`systemctl` 或 `systemctl list-units`

- ❖ 只显示服务单元的状态

`systemctl --type=service`

- ❖ 显示sshd服务单元

`systemctl status sshd.service -l`

- ❖ 验证sshd服务当前是否活动

`systemctl is-active sshd`

- ❖ 启动，停止和重启sshd服务

`systemctl start sshd.service`

`systemctl stop sshd.service`

`systemctl restart sshd.service`

❖ 重新加载配置

```
systemctl reload sshd.service
```

❖ 列出活动状态的所有服务单元

```
systemctl list-units --type=service
```

❖ 列出所有服务单元

```
systemctl list-units --type=service --all
```

❖ 查看服务单元的启用和禁用状态。

```
systemctl list-unit-files --type=service
```

❖ 列出失败的服务

```
systemctl --failed --type=service
```

❖ 列出依赖的单元

`systemctl list-dependencies sshd`

❖ 验证sshd服务是否开机启动

`systemctl is-enabled sshd`

❖ 禁用network, 使之不能自动启动, 但手动可以

`systemctl disable network`

❖ 启用network

`systemctl enable network`

❖ 禁用network, 使之不能手动或自动启动

`systemctl mask network`

❖ 启用network

`systemctl unmask network`

❖ target units:

unit配置文件: .target

ls /usr/lib/systemd/system/*.target

systemctl list-unit-files --type target --all

❖ 运行级别:

0 ==> runlevel0.target, poweroff.target

1 ==> runlevel1.target, rescue.target

2 ==> runlevel2.target, multi-user.target

3 ==> runlevel3.target, multi-user.target

4 ==> runlevel4.target, multi-user.target

5 ==> runlevel5.target, graphical.target

6 ==> runlevel6.target, reboot.target

❖ 查看依赖性:

systemctl list-dependencies graphical.target

- ❖ 级别切换: `init N ==> systemctl isolate name.target`
`systemctl isolate multi-user.target`

注: 只有 `/lib/systemd/system/*.target` 文件中
`AllowIsolate=yes` 才能切换(修改文件需执行 `systemctl daemon-reload` 才能生效)

- ❖ 查看target:

`runlevel who -r`

`systemctl list-units --type target`

- ❖ 获取默认运行级别:

`/etc/inittab ==> systemctl get-default`

- ❖ 修改默认级别:

`/etc/inittab ==> systemctl set-default name.target`

`systemctl set-default multi-user.target`

`ls -l /etc/systemd/system/default.target`

❖ 切换至紧急救援模式:

`systemctl rescue`

❖ 切换至emergency模式:

`systemctl emergency`

❖ 其它常用命令:

传统命令init, poweroff, halt, reboot都成为
systemctl的软链接

关机: `systemctl halt`、`systemctl poweroff`

重启: `systemctl reboot`

挂起: `systemctl suspend`

休眠: `systemctl hibernate`

休眠并挂起: `systemctl hybrid-sleep`

- ❖ UEFI或BIOS初始化，运行POST开机自检
- ❖ 选择启动设备
- ❖ 引导装载程序，centos7是grub2
- ❖ 加载装载程序的配置文件：`/etc/grub.d/`
`/etc/default/grub` `/boot/grub2/grub.cfg`
- ❖ 加载initramfs驱动模块
- ❖ 加载内核选项
- ❖ 内核初始化，centos7使用systemd代替init
- ❖ 执行initrd.target所有单元，包括挂载`/etc/fstab`
- ❖ 从initramfs根文件系统切换到磁盘根目录
- ❖ systemd执行默认target配置，配置文件
`/etc/systemd/default.target` `/etc/systemd/system/`

- ❖ systemd执行sysinit.target初始化系统及basic.target准备操作系统
- ❖ systemd启动multi-user.target下的本机与服务器服务
- ❖ systemd执行multi-user.target下的/etc/rc.d/rc.local
- ❖ Systemd执行multi-user.target下的getty.target及登录服务
- ❖ systemd执行graphical需要的服务

马哥教育

www.magedu.com

- ❖ `/etc/systemd/system`: 系统管理员和用户使用
`/usr/lib/systemd/system`: 发行版打包者使用
- ❖ 以 “**#**” 开头的行后面的内容会被认为是注释
- ❖ 相关布尔值, **1**、**yes**、**on**、**true** 都是开启, **0**、**no**、**off**、**false** 都是关闭。
- ❖ 时间单位默认是秒, 所以要用毫秒 (**ms**) 分钟 (**m**) 等请显式说明
- ❖ **service unit file**文件通常由三部分组成:
 - **[Unit]**: 定义与Unit类型无关的通用选项; 用于提供unit的描述信息、unit行为及依赖关系等
 - **[Service]**: 与特定类型相关的专用选项; 此处为Service类型
 - **[Install]**: 定义由 “**systemctl enable**”以及“**systemctl disable**”命令在实现服务启用或禁用时用到的一些选项

- ❖ **Unit**段的常用选项:
- ❖ **Description:** 描述信息
- ❖ **After:** 定义**unit**的启动次序, 表示当前**unit**应该晚于哪些**unit**启动, 其功能与**Before**相反
- ❖ **Requires:** 依赖到的其它**units**, 强依赖, 被依赖的**units**无法激活时, 当前**unit**也无法激活
- ❖ **Wants:** 依赖到的其它**units**, 弱依赖
- ❖ **Conflicts:** 定义**units**间的冲突关系

马哥教育

www.magedu.com

- ❖ **Service**段的常用选项:
- ❖ **Type**: 定义影响**ExecStart**及相关参数的功能的**unit**进程启动类型
 - **simple**: 默认值, 这个**daemon**主要由**ExecStart**接的指令串来启动, 启动后常驻于内存中
 - **forking**: 由**ExecStart**启动的程序透过**spawns**延伸出其他子程序来作为此**daemon**的主要服务。原生父程序在启动结束后就会终止
 - **oneshot**: 与**simple**类似, 不过这个程序在工作完毕后就结束了, 不会常驻在内存中
 - **dbus**: 与**simple**类似, 但这个**daemon**必须要在取得一个**D-Bus**的名称后, 才会继续运作. 因此通常也要同时设定**BusName=** 才行
 - **notify**: 在启动完成后会发送一个通知消息。还需要配合**NotifyAccess** 来让 **Systemd** 接收消息
 - **idle**: 与**simple**类似, 要执行这个**daemon**必须要所有的工作都顺利执行完毕后才执行。这类的**daemon**通常是开机到最后才执行即可的服务

- ❖ **EnvironmentFile:** 环境配置文件
- ❖ **ExecStart:** 指明启动unit要运行命令或脚本的绝对路径
- ❖ **ExecStartPre:** **ExecStart**前运行
- ❖ **ExecStartPost:** **ExecStart**后运行
- ❖ **ExecStop:** 指明停止unit要运行的命令或脚本
- ❖ **Restart:** 当设定**Restart=1** 时, 则当次**daemon**服务意外终止后, 会再次自动启动此服务

马哥教育

www.magedu.com

❖ Install段的常用选项:

- **Alias:** 别名, 可使用 `systemctl command Alias.service`
- **RequiredBy:** 被哪些units所依赖, 强依赖
- **WantedBy:** 被哪些units所依赖, 弱依赖
- **Also:** 安装本服务的时候还要安装别的相关服务

- ❖ 注意: 对于新创建的unit文件, 或者修改了的unit文件, 要通知systemd重载此配置文件, 而后可以选择重启

```
# systemctl daemon-reload
```

www.magedu.com

服务Unit文件示例:

❖ `vim /etc/systemd/system/bak.service`

`[Unit]`

`Description=backup my etc`

`Requires=atd.service`

`[Service]`

`Type=simple`

`ExecStart=/bin/bash -c "echo /testdir/bak.sh|at now"`

`[Install]`

`WantedBy=multi-user.target`

❖ `systemctl daemon-reload`

❖ `systemctl start bak`

设置内核参数

- ❖ 设置内核参数，只影响当次启动
- ❖ 启动时，在linux16行后添加`systemd.unit=desired.target`
- ❖ `systemd.unit=emergency.target`
- ❖ `systemd.unit=recure.target`
- ❖ `recure.target` 比`emergency` 支持更多的功能，例如日志等

马哥教育

www.magedu.com

- ❖ 文件系统损坏
先尝试自动修复，失败则进入**emergency shell**，提示用户修复
- ❖ 在**/etc/fstab**不存在对应的设备和UUID
等一段时间，如不可用，进入**emergency shell**
- ❖ 在**/etc/fstab**不存在对应挂载点
systemd 尝试创建挂载点，否则提示进入**emergency shell**.
- ❖ 在**/etc/fstab**不正确的挂载选项
提示进入**emergency shell**

www.magedu.com

- ❖ 启动时任意键暂停启动
- ❖ 按e键进入编辑模式
- ❖ 将光标移动linux16开始的行，添加内核参数rd.break
- ❖ 按ctrl-x启动
- ❖ mount -o remount,rw /sysroot
- ❖ chroot /sysroot
- ❖ passwd root
- ❖ touch /.autorelabel

马哥教育

www.magedu.com

❖ GRUB “the Grand Unified Bootloader”

引导提示时可以使用命令行界面

可从文件系统引导

❖ 主要配置文件 `/boot/grub2/grub.cfg`

❖ 修复配置文件

```
grub2-mkconfig > /boot/grub2/grub.cfg
```

❖ 修复grub

```
grub2-install /dev/sda BIOS环境
```

```
grub2-install UEFI环境
```

www.magedu.com

- ❖ 为编译安装的httpd服务，实现service unit文件
- ❖ 破解centos7 口令
- ❖ 修改默认的启动内核为新编译内核
- ❖ 启动时临时禁用SELinux
- ❖ 启动时进入emergency模式
- ❖ 卸载编译安装的新内核

马哥教育

www.magedu.com

- ❖ 博客: <http://magedu.blog.51cto.com>
- ❖ 主页: <http://www.magedu.com>
- ❖ QQ: 1661815153, 113228115
- ❖ QQ群: 203585050, 279599283

马哥教育
www.magedu.com



马哥教育
最专业的Linux培训机构

Thank You!