

systemd 再入門

Kazuhiro NISHIYAMA

東京エリア・関西合同Debian勉強会

2021-04-17

株式会社Ruby開発

自己紹介

- 西山 和広
- Ruby のコミッター
- twitter, github など: @znz
- 株式会社Ruby開発 www.ruby-dev.jp

systemd のおすすめ資料



systemdエッセンシャル / systemd-
intro - Speaker Deck

[https://speakerdeck.com/moriwaka/
systemd-intro](https://speakerdeck.com/moriwaka/systemd-intro)

agenda

- crontab の代わりに systemd-timer
- user 権限での systemd

systemd-timer の利点

- `systemctl start` での動作確認と `timer` での実行の環境が同じ
- `journald` に自動でログが残る
- 他の `unit` との依存関係が設定できる (DB バックアップなら DB 起動必須など)
- 時刻指定が `crontab` より柔軟

systemd-timer の欠点

- 最低限の利用でも記述量が多い
 - service ファイルと timer ファイルが必要で 1 行だけでは出来ない
- 時刻指定が独自
 - kubernetes の cronjob のような新しいものでも crontab 形式の時刻指定が使われていることが多い

シンプルな使い方

- Type=oneshot の service ユニットを作成
- 同名の timer ユニットを作成して
systemctl enable --now
foo.timer のように enable と
start をする
 - 別の名前のユニットを start するなら
Unit= で指定

enable と start

- `foo.timer` から起動する
`foo.service` は `enable` しない
- `foo.timer` も `enable` を忘れるとマシンの再起動後に動いていない (`service unit` と同じ)
- `systemctl start foo.service` で `timer` を待たずに起動して動作確認可能

動作確認例

- `systemctl list-timers`
- `systemctl status systemd-tmpfiles-clean.timer`
- `systemctl status systemd-tmpfiles-clean.service`

ログの例

- ログ表示は `systemd-journal` グループに所属するか `sudo` を使う
- `journalctl -u systemd-tmpfiles-clean.timer`
- `journalctl -u systemd-tmpfiles-clean.service`
- `Starting` が実行開始時刻で `Started` が実行終了時刻

service 作成例

```
# /etc/systemd/system/gitlab-backup.service
[Unit]
Description=Backup gitlab
After=gitlab-runsvdir.service
Requires=gitlab-runsvdir.service

[Service]
Type=oneshot
ExecStart=/opt/gitlab/bin/gitlab-rake gitlab:backup:create CRON=1
```

service 作成例 (解説)

- ExecStart が crontab での実行内容に相当 (ARGV0 の部分はフルパス必須)
- crontab 代わりに最低限必要なのは Type と ExecStart
- After や Requires で gitlab の service unit が動いているときだけバックアップを実行 (runsvdir 経由なので gitlab 全体が正常に動いているかどうかは未確認)

timer 作成例

```
# /etc/systemd/system/gitlab-backup.timer
[Unit]
Description=Backup gitlab

[Timer]
OnCalendar=*-*-* 2,14:00
Persistent=true

[Install]
WantedBy=timers.target
```

timer 作成例 (解説)

- Persistent=true は anacron 相当
- crontab 代わりに最低限必要なのは OnCalendar と WantedBy

OnCalendar

- `systemd-analyze calendar '*-*-* 2,14:00'`などで OnCalendar の指定内容を確認可能
- 任意の時刻を基準にするには `faketime` コマンドと組み合わせて `faketime '2021-04-17 16:00' systemd-analyze calendar '*-*-* 2,14:00'`

at コマンド代わり

- `systemd-run --on-active=30 /bin/touch /tmp/foo` で30秒後
- `systemd-run --on-active="1h 30m" --unit foo.service` で1時間半後
- `systemd-run --on-calendar="2021-04-17 16:00" /bin/touch /tmp/foo` のように日時指定も可能

user 権限での systemd

- 特に設定していなければ
pam_systemd.so で /lib/systemd/
systemd --user が起動して、ログアウト
時に終了
- systemctl --user で操作
- loginctl enable-linger someuser
で常に起動 (loginctl disable-
linger someuser で戻す)

動作確認

- 自ユーザーなら `systemctl --user status`
- 別ユーザーなら `sudo -u someuser XDG_RUNTIME_DIR=/run/user/$(id -u someuser) systemctl --user status`
- `sudo -u someuser systemctl --user status` だけだと `Failed to connect to bus: No such file or directory`

unit ファイルの場所

- 普通は `~/.config/systemd/user/` に置く
- 全ユーザー共通なら `/etc/systemd/user/` に置くのも可能
- 全パスは `systemd.unit(5)` 参照

Install の WantedBy

- system の unit なら WantedBy=multi-user.target を使うことが多い
- user の unit は WantedBy=default.target を代わりに使う

unit ファイル例

```
# /home/chatuser/.config/systemd/user/weechat.service
[Unit]
Description=A WeeChat client and relay service using Tmux
After=network.target

[Service]
Type=forking
RemainAfterExit=yes
ExecStart=/usr/bin/tmux -L weechat new -d -s weechat weechat
ExecStop=/usr/bin/tmux -L weechat kill-session -t weechat

[Install]
WantedBy=default.target
```

unit ファイル例 (解説)

- マシン起動時に weechat を tmux の中で自動起動
- 手抜きで tmux を手動終了したときの処理は省略
 - 終了してしまったときはマシンを再起動している

動作確認例

- `sudo -u chatuser
XDG_RUNTIME_DIR=/run/user/$(id -
u chatuser) systemctl --user
status weechat`
- `sudo -u chatuser
XDG_RUNTIME_DIR=/run/user/$(id -
u chatuser) journalctl --user -
u weechat`

/etc にあった例

```
# ls /etc/systemd/user/sockets.target.wants/  
dirmngr.socket  gpg-agent-browser.socket  gpg-agent-extra.socket  
gpg-agent.socket  gpg-agent-ssh.socket  
# readlink /etc/systemd/user/sockets.target.wants/gpg-agent.socket  
/usr/lib/systemd/user/gpg-agent.socket  
# cat /etc/systemd/user/sockets.target.wants/gpg-agent.socket  
[Unit]  
Description=GnuPG cryptographic agent and passphrase cache  
Documentation=man:gpg-agent(1)  
  
[Socket]  
ListenStream=%t/gnupg/S.gpg-agent  
FileDescriptorName=std  
SocketMode=0600  
DirectoryMode=0700  
  
[Install]  
WantedBy=sockets.target
```


まとめ

- crontab の代わりに timer unit + service unit
- systemd-run は at の代わりになる
- user 権限での systemd は loginctl enable-linger で常時起動
- ~/.config/systemd/user/ に unit ファイル