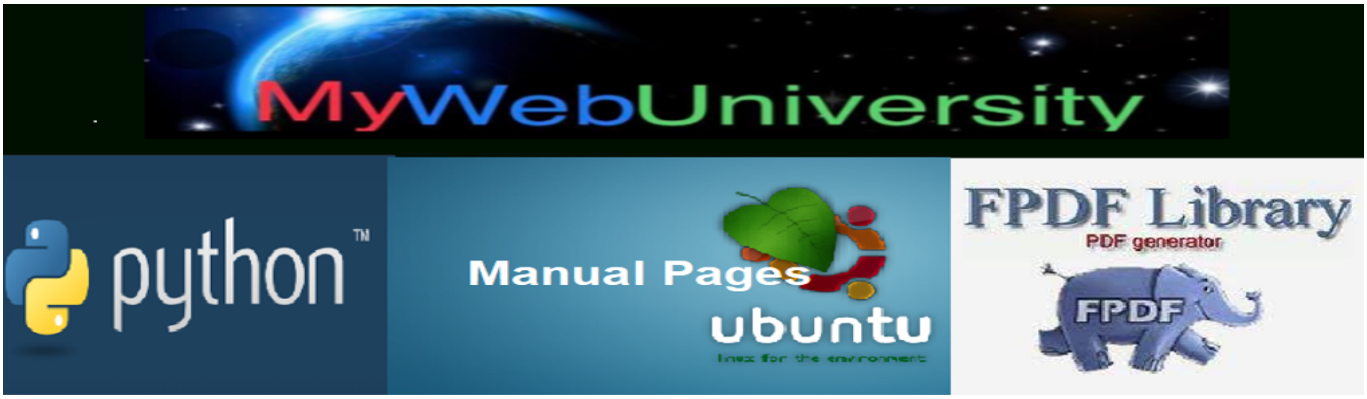

Rocky Enterprise Linux 9.2 Manual Pages on command 'systemd-socket-proxyd.8'

$ man systemd-socket-proxyd.8

SYSTEMD-SOCKET-PROXYD(8) systemd-socket-proxyd SYSTEMD-SOCKET-PROXYD(8)

NAME

systemd-socket-proxyd - Bidirectionally proxy local sockets to another (possibly remote) socket

SYNOPSIS

systemd-socket-proxyd [OPTIONS...] HOST:PORT

systemd-socket-proxyd [OPTIONS...] UNIX-DOMAIN-SOCKET-PATH

DESCRIPTION

systemd-socket-proxyd is a generic socket-activated network socket forwarder proxy daemon for IPv4, IPv6 and UNIX stream sockets. It may be used to bi-directionally forward traffic from a local listening socket to a local or remote destination socket.

One use of this tool is to provide socket activation support for services that do not natively support socket activation. On behalf of the service to activate, the proxy inherits the socket from systemd, accepts each client connection, opens a connection to a configured server for each client, and then bidirectionally forwards data between the two.

This utility's behavior is similar to socat(1). The main differences for systemd-socket-proxyd are support for socket activation with "Accept=no" and an

event-driven design that scales better with the number of connections.

OPTIONS

The following options are understood:

-h, --help

Print a short help text and exit.

--version

Print a short version string and exit.

--connections-max=, -c

Sets the maximum number of simultaneous connections, defaults to 256. If the limit of concurrent connections is reached further connections will be refused.

--exit-idle-time=

Sets the time before exiting when there are no connections, defaults to infinity. Takes a unit-less value in seconds, or a time span value such as "5min 20s".

EXIT STATUS

On success, 0 is returned, a non-zero failure code otherwise.

EXAMPLES

Simple Example

Use two services with a dependency and no namespace isolation.

Example 1. proxy-to-nginx.socket

```
[Socket]
ListenStream=80

[Install]
WantedBy=sockets.target
```

Example 2. proxy-to-nginx.service

```
[Unit]
Requires=nginx.service
After=nginx.service
Requires=proxy-to-nginx.socket
After=proxy-to-nginx.socket

[Service]
ExecStart=/lib/systemd/systemd-socket-proxyd /run/nginx/socket
PrivateTmp=yes
PrivateNetwork=yes
```

Example 3. nginx.conf

```
[...]
server {
    listen       unix:/run/nginx/socket;
    [...]
```

Example 4. Enabling the proxy

```
# systemctl enable --now proxy-to-nginx.socket
$ curl http://localhost:80/
```

If nginx.service has StopWhenUnneeded= set, then passing --exit-idle-time= to systemd-socket-proxyd allows both services to stop during idle periods.

Namespace Example

Similar as above, but runs the socket proxy and the main service in the same private namespace, assuming that nginx.service has PrivateTmp= and PrivateNetwork= set, too.

Example 5. proxy-to-nginx.socket

```
[Socket]
ListenStream=80

[Install]
WantedBy=sockets.target
```

Example 6. proxy-to-nginx.service

```
[Unit]
Requires=nginx.service
After=nginx.service
Requires=proxy-to-nginx.socket
After=proxy-to-nginx.socket
JoinsNamespaceOf=nginx.service

[Service]
ExecStart=/lib/systemd/systemd-socket-proxyd 127.0.0.1:8080
PrivateTmp=yes
PrivateNetwork=yes
```

Example 7. nginx.conf

```
[...]
server {
    listen       8080;
    [...]
```

Example 8. Enabling the proxy

```
# systemctl enable --now proxy-to-nginx.socket
$ curl http://localhost:80/
```

SEE ALSO

systemd(1), systemd.socket(5), systemd.service(5), systemctl(1), socat(1), nginx(1), curl(1)