Lekcja: Uruchamianie usług sieciowych

Cel ogólny. Uczeń rozumie zasady działania usług sieciowych oraz potrafi zarządzać nimi w systemach Linux z wykorzystaniem systemd.

Cele szczegółowe Po lekcji uczeń:

- wyjaśnia pojęcia *usługa*, *demon*, *superdemon*,
- opisuje różnice między SysVinit a systemd,
- omawia rolę i znaczenie wybranych usług sieciowych (sshd, ntpd/chronyd, postfix, rsyncd),
- rozumie strukturę i przeznaczenie plików unitów systemd,
- opisuje podstawowe operacje wykonywane na usługach (start, stop, restart, enable, disable) - **bez wykonywania ich w praktyce**,
- zna współczesne metody uruchamiania usług (konteneryzacja, automatyzacja),
- rozumie znaczenie bezpieczeństwa usług w nowoczesnych systemach Linux.

1. Usługi i demony w systemach Linux

Usługi sieciowe (ang. *network services*) to programy działające w tle, których zadaniem jest realizacja określonych funkcji systemowych lub sieciowych. Procesy te uruchamiane są zwykle automatycznie przy starcie systemu i działają nieprzerwanie, oczekując na żądania klientów.

Cechy demonów (daemons):

- działają w tle i nie są powiązane z terminalem użytkownika,
- zużywają minimalne zasoby podczas oczekiwania,
- często są monitorowane przez system inicjalizacji (np. systemd), który automatycznie je restartuje w przypadku błędów,
- odpowiadają za kluczowe funkcje systemu: serwery WWW, SSH, DNS, NTP, SMTP, logowanie zdarzeń itd.

2. Superdemon – xinetd

Superdemony (np. *xinetd*) kontrolują wybrane porty i uruchamiają usługę **dopiero wtedy**, gdy nadejdzie pierwsze żądanie.

Dawniej stosowane częściej, dziś rzadziej spotykane, ponieważ większość usług działa w trybie **standalone** (np. sshd, httpd), co jest bardziej wydajne na współczesnych systemach.

3. Przykładowe usługi sieciowe

sshd

- demon protokołu SSH (szyfrowana komunikacja terminalowa),

- standard w administracji systemami Linux,
- zgodny z nowymi metodami autoryzacji: klucze FIDO2/WebAuthn, certyfikaty X.509.

ntpd / chronyd

- usługi synchronizacji czasu,
- we współczesnych systemach częściej stosuje się **chrony**, który jest wydajniejszy w środowiskach z niestabilną siecią (np. laptopach).

postfix

- agent transportu poczty (MTA),
- nadal wykorzystywany, ponieważ wiele procesów systemowych generuje komunikaty dla administratora.

rsyncd

- demon synchronizacji plików,
- obecnie często zastępowany przez transmisje SSH (rsync przez -e ssh) lub narzędzia typu *syncthing*, ale nadal kluczowy w automatyzacjach i backupach.

4. Systemy inicjalizacji – ewolucja

SysVinit (historyczny standard)

- oparty na skryptach w /etc/init.d/ i katalogach /etc/rc*.d/.
- zarządzanie usługami za pomocą: service start|stop|restart|status
- narzędzie chkconfig umożliwiało konfigurację uruchamiania usług w zależności od poziomu pracy systemu.

SysVinit nadal działa w systemach legacy, ale nowe dystrybucje korzystają z systemd.

5. Systemd – współczesny standard (od 2012 > dominuje obecnie)

Systemd to nowoczesny menedżer systemu i usług, stosowany w większości dystrybucji Linux (m.in. Ubuntu, Debian, RHEL, Fedora, Arch, openSUSE).

Najważniejsze elementy systemd:

- **unified unit files** – pliki .service, .socket, .timer, .mount, .target,
- bardzo szybki równoległy rozruch systemu,
- wbudowany mechanizm nadzoru usług (restart, restart-on-failure),
- obsługa zależności między usługami,
- możliwość zdalnego zarządzania usługami (systemctl -H).

Zarządzanie usługami w systemd

systemctl start <usługa>

systemctl stop <usługa>

systemctl restart <usługa>

systemctl status <usługa>

Automatyczne uruchamianie przy starcie

systemctl enable <usługa>

systemctl disable <usługa>

Sprawdzanie stanu

systemctl is-active <usługa>

systemctl is-enabled <usługa>

systemctl list-units --type=service

Lokalizacja plików unitów

- /usr/lib/systemd/system/ – pliki dostarczone przez system,
- /etc/systemd/system/ – pliki użytkownika (nadpisują powyższe).

6. Nowoczesne podejścia (2024–2026)

Współczesna administracja usługami sieciowymi często wykracza poza klasyczne mechanizmy.

Konteneryzacja

- Docker, Podman, LXC/LXD, systemd-nspawn

  > przenoszenie usług do kontenerów zamiast uruchamiania ich w systemie hosta.

Co to jest konteneryzacja?

Konteneryzacja to sposób uruchamiania aplikacji (np. usług sieciowych) w **izolowanych środowiskach**, zwanych **kontenerami**.

Kontener to:

- osobny, odizolowany system użytkowy,
- współdzielący jądro hosta,
- ale mający własne procesy, sieć, system plików, biblioteki.

Czyli coś pomiędzy:

- „procesem + izolacja”

  a

- „lekką wirtualizacją”.

Najważniejsze:

usługa działa „w pudełku”, niezależnie od reszty systemu.

Co oznaczy ten zapis?

przenoszenie usług do kontenerów zamiast uruchamiania ich w systemie hosta.

To znaczy: Zamiast instalować usługę **bezpośrednio na systemie Linux**, instaluje się ją **w kontenerze**.

Np. zamiast instalować nginx na Ubuntu:

apt install nginx

systemctl start nginx

możesz go uruchomić w kontenerze:

docker run -p 80:80 nginx

Zalety:

- oddzielone konfiguracje,
- brak konfliktów pakietów,
- łatwa aktualizacja,
- łatwe przenoszenie między serwerami,
- dużo większe bezpieczeństwo.

Co to jest Docker, Podman, LXC, LXD, systemd-nspawn?

Pokrótce:

Docker - Najpopularniejsza platforma kontenerowa.

Cechy:

- bardzo duży ekosystem,
- gotowe obrazy na Docker Hub,
- kontenery aplikacyjne,
- świetne do DevOps.

Podman - Nowsza, bezpieczniejsza alternatywa dla Dockera.

Najważniejsze różnice:

- nie używa demona (jest *daemonless*),
- może być uruchamiany jako zwykły użytkownik,
- kompatybilny z Dockerfile.

W wielu dystrybucjach **zastępuje Dockera** (np. Fedora, RHEL).

LXC / LXD - Kontenery „systemowe”, czyli **całe środowisko systemowe**, a nie tylko jedna aplikacja.

- LXC = Linux Containers (niski poziom).
- LXD = zarządzanie LXC (wysoki poziom, bardziej user-friendly).

Zastosowanie:

- lekka alternatywa dla VM,
- tworzenie „małych Linuxów" w środku Linuxa.

systemd-nspawn - Wbudowana w systemd technologia uruchamiania bardzo lekkich kontenerów.

Możesz stworzyć mini-system komendą:

systemd-nspawn -D /path/do/katalogu

Używane w:

- testowaniu systemów,
- debugowaniu,
- tworzeniu chrootów nowej generacji.

Czyli podsumowując jednym zdaniem:

To technologie do uruchamiania usług i aplikacji w odizolowanych kontenerach, zamiast instalowania ich bezpośrednio w systemie operacyjnym hosta.

Automatyzacja i orkiestracja

- Ansible (YAML) – zarządzanie usługami: - name: restart ssh service service: name: ssh state: restarted
- Kubernetes – uruchamianie usług jako *deployments*, *pods*, *daemonsets*.

Obsługa zdarzeń i harmonogram zadań

Systemd zastępuje cron dzięki mechanizmowi *systemd timers*:

systemctl enable backup.timer

systemctl start backup.timer

Bezpieczeństwo (hardening)

Coraz częściej stosuje się:

- systemd-analyze security <usługa> – analiza twardości usługi,
- ograniczenia przestrzeni nazw (namespace),
- sandboxing w systemd: ProtectSystem=full PrivateTmp=true NoNewPrivileges=true

7. Podsumowanie

Współczesna administracja usługami sieciowymi opiera się głównie na **systemd**, który zapewnia:

- większą stabilność,

- automatyczny restart usług,
- bezpieczeństwo,
- zdalne zarządzanie,
- integrację z nowoczesnymi środowiskami (kontenery, automatyzacja).

Klasyczne narzędzia (service, init.d, xinetd) nadal mogą występować, ale pełnią funkcję kompatybilnościową.