

Docker na Ubuntu Server

Czas: 2 × 45 min

praca indywidualna

Cel: Zainstalowanie Docker Engine z oficjalnego repozytorium, uruchomienie kontenerów z mapowaniem portów i wolumenami oraz porównanie kontenerów z maszynami wirtualnymi.

Po zajęciach potrafisz:

- instaluje Docker Engine i wtyczkę Compose z oficjalnego repozytorium Dockera
- uruchamia kontener z mapowaniem portu i katalogiem podpiętym z hosta
- wdraża aplikację z pliku `compose.yaml` z trwałym wolumenem
- diagnozuje kontenery poleceniami `docker ps`, `docker logs`, `docker stats`
- aktualizuje kontener bez utraty danych i porównuje kontener z maszyną wirtualną

Sprzęt i materiały

- komputer stacjonarny z VirtualBox i maszyną wirtualną Ubuntu Server LTS (z zadania 10): min. 2 rdzenie, 2 GB RAM, 20 GB dysku, karta sieciowa w trybie mostkowanym

Oprogramowanie

- **Docker Engine (instrukcja dla Ubuntu)** — metoda „Install using the apt repository” docs.docker.com/engine/install/ubuntu
- **Uptime Kuma** — obraz `louislam/uptime-kuma:2`, port 3001 github.com/louislam/uptime-kuma
- **Obraz nginx (Docker Hub)** — oficjalny obraz serwera WWW hub.docker.com/_/nginx
- **Klient SSH (Terminal Windows)** — praca na serwerze przez SSH ułatwia kopiowanie poleceń

Przygotowanie przed zajęciami

- Uruchom VM Ubuntu Server, zaktualizuj system (`sudo apt update && sudo apt full-upgrade`) i wykonaj migawkę „przed dockerem”.
- Przeczytaj na stronie Dockera akapit o usuwaniu pakietów konfliktowych (Uninstall old versions).

Bezpieczeństwo

- Instaluj Dockera tylko w maszynie wirtualnej. Polecenia kopiuj wyłącznie z oficjalnej dokumentacji docs.docker.com.
- Przynależność do grupy `docker` daje w praktyce uprawnienia administratora (root) — dodawaj do niej tylko zaufanych użytkowników.
- Używaj tylko obrazów oficjalnych lub od znanych autorów; sprawdź na Docker Hub, kto publikuje obraz.
- Nie wystawiaj paneli administracyjnych kontenerów poza sieć pracowni; po utworzeniu konta w Uptime Kuma ustaw silne hasło.

Przebieg zadania

1 Instalacja Docker Engine

Wykonuj polecenia z oficjalnej instrukcji i zapisuj je w sprawozdaniu.

1. Usuń ewentualne pakiety konfliktowe zgodnie z dokumentacją (np. `docker.io`, `docker-compose`, `podman-docker`).
2. Dodaj klucz GPG Dockera do `/etc/apt/keyrings/docker.asc` i plik repozytorium `/etc/apt/sources.list.d/docker.sources` dokładnie według instrukcji, następnie `sudo apt update`.
3. Zainstaluj pakiety: `sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin`.
4. Sprawdź `sudo systemctl status docker`, `docker --version`, `docker compose version` i uruchom `sudo docker run hello-world`. Przeczytaj, co wypisał kontener.
5. Dodaj się do grupy: `sudo usermod -aG docker $USER`, wyloguj się i zaloguj ponownie; `docker ps` działa bez `sudo`.

2 Serwer WWW w kontenerze

1. Utwórz katalog `~/www` z plikiem `index.html` zawierającym Twoje imię i nazwisko.
2. Uruchom: `docker run -d --name www -p 8080:80 -v ~/www:/usr/share/nginx/html:ro nginx:stable`.
3. Otwórz w przeglądarce na komputerze `http://<IP VM>:8080`. Zmień plik `index.html` na serwerze i odśwież stronę.
4. Sprawdź `docker ps`, `docker logs www` (widać Twoje żądania) i `docker exec -it www sh -c "cat /etc/os-release"` — porównaj system w kontenerze z systemem VM.

3 Aplikacja z docker compose

Uptime Kuma przyda się w zadaniu 28.

1. Utwórz katalog `~/kuma` i w nim plik `compose.yaml` z usługą: obraz `louislam/uptime-kuma:2`, `restart: unless-stopped`, port `3001:3001`, wolumen nazwany `kuma-data` zamontowany w `/app/data`.
2. Uruchom `docker compose up -d`, sprawdź `docker compose ps` i `docker compose logs --tail 20`.
3. Otwórz `http://<IP VM>:3001`, utwórz konto administratora i dodaj monitor HTTP strony nginx z kroku 2.
4. Sprawdź trwałość danych: `docker compose down`, potem `docker compose up -d` — monitor nadal istnieje. Odszukaj wolumen: `docker volume ls`, `docker volume inspect`.

4 Aktualizacja i porównanie z VM

1. Zaktualizuj aplikację: `docker compose pull`, `docker compose up -d`, a następnie `docker image prune`. Sprawdź, że dane zostały.
2. Uruchom `docker stats --no-stream` i zapisz zużycie RAM kontenerów. Porównaj z `free -h` całej VM i z rozmiarem obrazów (`docker images`).
3. Zatrzymaj i usuń kontener nginx: `docker rm -f www`. Kontener Uptime Kuma zostaw — będzie potrzebny w zadaniu 28.

W sprawozdaniu umieść

- ☐ lista wykonanych poleceń instalacji i wynik `docker run hello-world`
- ☐ zawartość pliku `compose.yaml`
- ☐ zrzuty: strona z nginx, panel Uptime Kuma, wynik `docker stats --no-stream`
- ☐ tabela porównania: kontener a maszyna wirtualna
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Czym różni się obraz od kontenera i co dzieje się z danymi w kontenerze po `docker rm`, jeśli nie użyto wolumenu?
2. Co oznacza zapis `-p 8080:80` i `-v ~/www:/usr/share/nginx/html:ro`?
3. Dlaczego kontener zużywa mniej zasobów niż maszyna wirtualna i czego kontener nie zapewnia (np. inny system operacyjny, izolacja jądra)?
4. Dlaczego nie instaluje się Dockera skryptem znalezionym na przypadkowym blogu?

Kryteria oceny

Kryterium	Punkty
Instalacja z oficjalnego repozytorium i weryfikacja	5
Kontener nginx z mapowaniem portu i katalogu	4
Uptime Kuma w docker compose z trwałym wolumenem	5
Aktualizacja i porównanie kontener — VM	3
Odpowiedzi na pytania kontrolne	3
Razem	20

Sprawozdanie przygotuj na wzorze dołączonym do zadania (plik .docx o tym samym numerze).