Лекція 3: Віртуалізація та контейнеризація

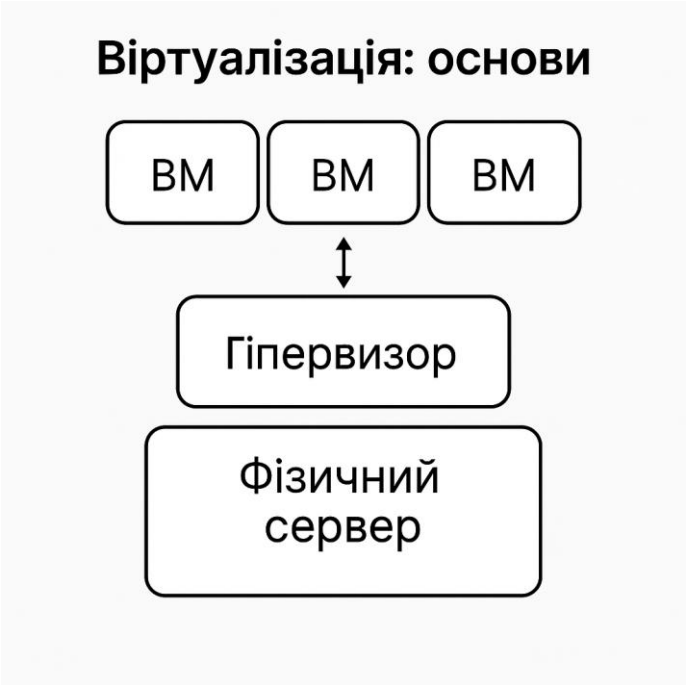
Гіпервізори, контейнери, приклади, порівняння, безпека

План лекції 3

1. Основи віртуалізації
2. Типи гіпервізорів
3. Контейнеризація vs віртуалізація
4. Інструменти контейнеризації
5. Docker vs Podman vs LXC
6. Кейс: Google та Kubernetes
7. Безпека контейнерів
8. Висновки

Віртуалізація: основи

- Дозволяє запускати кілька ОС на одному сервері
- Економія ресурсів, тестування різних систем
- Приклад: запуск Windows і Linux одночасно



Типи гіпервізорів

- Тип 1: bare-metal (ESXi, Hyper-V)
- Тип 2: hosted (VirtualBox, VMware Workstation)

Приклад: VirtualBox для розробників, ESXi для дата-центрів

Контейнеризація проти віртуалізації

- Контейнери легші за ВМ, працюють на одному ядрі ОС
- Швидкий запуск (секунди проти хвилин)
- Приклад: запуск 10 веб-серверів у Docker

Інструменти контейнеризації

- Docker — стандарт де-факто
- Podman — альтернатива без демона
- LXC — ближчий до ВМ

Приклад: використання Docker для Flask API

Docker vs Podman vs LXC

- Docker: простий, має Docker Hub
- Podman: більш безпечний, сумісний із Docker CLI
- LXC: низькорівневий контроль

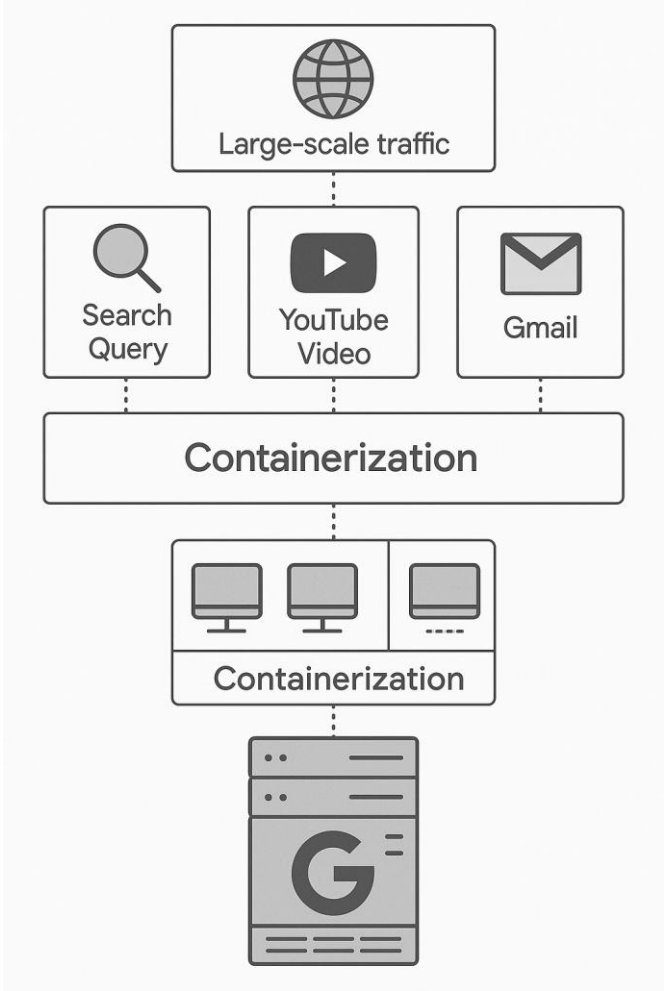
Docker vs Podman vs LXC

Порівняльна таблиця

Критерій	Docker	Podman	LXC (Linux Containers)
Призначення	Платформа контейнеризації, стандарт де-факто	Альтернатива Docker з акцентом на безпеку	Легка віртуалізація, ближче до повноцінних ВМ
Архітектура	Працює через постійний Docker Daemon	Працює без демона (rootless режим)	Використовує ізоляцію на рівні ядра Linux
Сумісність	Має власний CLI та екосистему	CLI сумісний із Docker (можна використовувати ті самі команди)	Менш сумісний з DevOps-інструментами
Популярність	Дуже висока, величезна екосистема	Зростає, але значно менше користувачів	Обмежена, переважно у Linux-середовищах
Простота	Простий у використанні, багато документації	Трохи складніший, особливо в адмініструванні	Потребує глибших знань Linux
Безпека	Має ризики через daemon з правами root	Більш безпечний, rootless-контейнери	Вища ізоляція, але складніше у налаштуванні
Інтеграція з DevOps	Легко інтегрується з Kubernetes, CI/CD	Також інтегрується з Kubernetes, підтримує OCI-образи	Обмежена інтеграція, не стандарт у DevOps
Приклад використання	Розробка та деплой веб-додатків, мікросервіси	Альтернатива Docker у середовищах із підвищеними вимогами до безпеки	Сценарії, де потрібна повна ізоляція середовища, але легша за ВМ

Кейс: Google та Kubernetes

- Google запускає мільярди контейнерів щодня
- Використання Borg → Kubernetes
- Приклад: YouTube, Gmail



Безпека контейнерів

- Використання ізоляції namespace, cgroups
- Ризики: втеча з контейнера, некоректні образи
- Захист: мінімальні образи, підписані образи

Висновки

- Контейнеризація = швидкість і гнучкість
- Використовується у всіх великих IT-компаніях
- Важливий аспект — безпека образів і контейнерів