

УДК: 004.272.3

А. М. Луцків канд. техн. наук, доц., Р. О. Луцишин

Тернопільський національний технічний університет імені Івана Пулюя, Україна

ВИКОРИСТАННЯ КОНТЕЙНЕРИЗАЦІЇ ПРИ РОЗРОБЛЕННІ ЛАБОРАТОРНИХ ПРАКТИКУМІВ СТУДЕНТІВ

A. M. Lutskiv Ph.D., Assoc. Prof., R.O. Lutsyshyn

CONTAINERIZATION USAGE FOR THE STUDENT'S LABORATORY WORKSHOPS DEVELOPMENT

При розробленні лабораторних практикумів перед викладачем постають наступні завдання:

- скорочення часу для підготовки та розгортання лабораторного стенду;
- спрощення процесу розгортання лабораторного стенду (студент може не мати хороших навичок системного адміністрування для встановлення та конфігурування певних типів сервісів, або вони розглядаються у інших курсах);
- мінімізація використання системних ресурсів;
- у ряді випадків доцільною є максимальна автоматизація та спрощення розгортання лабораторного стенду, зокрема, якщо мова йде про дистанційне навчання.

Віртуалізація на рівні операційної системи — метод віртуалізації, при якому ядро операційної системи підтримує декілька ізольованих примірників простору користувача, замість одного. Ці примірники (часто звані контейнерами або зонами) з точки зору користувача повністю ідентичні реальному серверові. Ядро забезпечує повну ізольованість контейнерів, тому програми з різних контейнерів не можуть впливати одна на одну.

При організації навчального процесу доволі часто можна зустріти підхід до розгортання лабораторних стендів у вигляді образів готових віртуальних машин (Nadoor-кластери Cloudera CDH та Hortonworks HDP, MongoDB University[1]). Проте, такі образи віртуальних машин є доволі об'ємними й їх використання є доволі ресурсоємним. У даному випадку використовується повна віртуалізація з використанням віртуальних машин Virtual Box, VM Ware.

На думку авторів, доцільнішим є використання технологій контейнеризації, зокрема, використання платформи Docker. Такий підхід передбачає наявність у студента деяких базових уявлень про контейнеризацію, проте, запуск та робота з таким лабораторним стендом є доволі простою (кількість необхідних попередніх налаштувань та кількість необхідних дій для запуску/зупинки/зберігання стану) та ефективною (з точки зору використання пам'яті й процесорного часу), швидшою у розгортанні (необхідність завантаження даних меншого об'єму), може бути використана як на робочому комп'ютері та і в хмарному сервісі.

Даний програмний продукт має наступні властивості:

- динамічна інфраструктура;
- швидке впровадження змін;
- спрощене розгортання додатків та програм.

Значимо, що існує кілька технологій контейнеризації на базі операційної системи (ОС) Linux, основними з яких є LXC (LinuX Container), а також LXD, CGManager, LXCFS. Для спрощення роботи з контейнерами використовують спеціалізовані програмні продукти, які в свою чергу базуються на контейнерах і представляють зручний інтерфейс взаємодії з ними, надають різноманітні засоби автоматизації роботи з ними. До складу інструментарію LXC входить бібліотека liblxc,

набір утиліт (lxc-create, lxc-start, lxc-stop, lxc-ls і тому подібне), шаблони для побудови контейнерів і набір біндінгів для різних мов програмування. Для ізоляції процесів, мережевого стека ірс, uts і точок монтування використовується механізм просторів імен (namespaces). Для обмеження ресурсів застосовуються cgroups. Для пониження привілеїв і обмеження доступу задіяні такі можливості ядра, як профілі Apparmor і SELinux, політики Seccomp, Chroots (pivot_root) і capabilities.

У контексті контейнерів Linux управління ресурсами організовано через cgroups. Cgroups дозволяють користувачеві виділяти ресурси, такі як процесорний час, системна пам'ять, пропускна здатність мережі, блок введення-виведення або будь-яку комбінацію з цих ресурсів для установки обмеженою користувачем групи завдань або процесів, запущених в даній системі. Як приклад, можна навести одну з популярних платформ для роботи з контейнерами — Docker. У цілому, використання даного підходу дає змогу створювати віртуальні одиниці більш ефективно і менш ресурсозатратно, у порівнянні зі створенням віртуальної машини.

Дане вирішення є апробованим при підготовці лабораторних практикумів з курсів «Паралельні та розподілені обчислення»[2], з використанням технологій OpenMP, MPI, OpenCL, CUDA та «Інформаційні системи паралельної та розподіленої обробки даних» (використання веб-сервісів, систем опрацювання великих даних).

Використання даного підходу дає змогу викладачеві використовувати офіційні репозиторії компаній Nvidia, AMD, Intel та інших для отримання вже налаштованих та відлагоджених контейнерів з наборами системного програмного забезпечення для задач паралельного програмування та попередньоналаштованими хмарними сервісами. Використовуючи базові механізми багат шаровості Docker-контейнерів, викладач, може відносно просто інтегрувати в контейнер лабораторне завдання й необхідні елементи його виконання. Публікація в публічному репозиторії дає змогу студенту, відносно просто, завантажити контейнери та працювати з ними. При виконанні таких лабораторних практикумів, студенти здобувають додатково необхідні як розробникам так і системним адміністраторам навички DevOps-інженерів.

Варто відзначити, що операційна система Windows 10, також дає змогу запускати Linux-контейнери, що суттєво знижує поріг входження у предметну область для більшості студентів, які, на жаль, не мають досвіду роботи з Unix-подібними операційними системами.

Література

1. Turnbull J. The docker book containerization is the new virtualization / James Turnbull., 2014. – 321 с.
2. MongoDB University. M202: MongoDB Advanced Deployment and Operations. [Електронний ресурс] – Режим доступу до ресурсу: URL: <https://university.mongodb.com/courses/M202/about>
3. Docker Documents [Електронний ресурс] – Режим доступу до ресурсу: URL: <https://docs.docker.com/>.
4. Луцків А.М. Паралельні та розподілені обчислення : Навчальний посібник / Луцків А.М., Лупенко С.А., Пасічник В.В. — Львів : Магнолія 2006 , 2015 — 566 с. — ISBN 9786175741108.