# Digital technologies for creating a modern learning environment in higher education

Volodymyr Aleksiyev,

Professor of Cybersecurity and Information Technologies Department, S. Kuznets KhNEU

Introduction

Today, the speed of a new digital product to market is faster than ever.

For example, the creation of a new S. KUZNETS KhNUE [website](#) took 2 weeks with the active development of 3 people on the WordPress platform and development tools completely free.

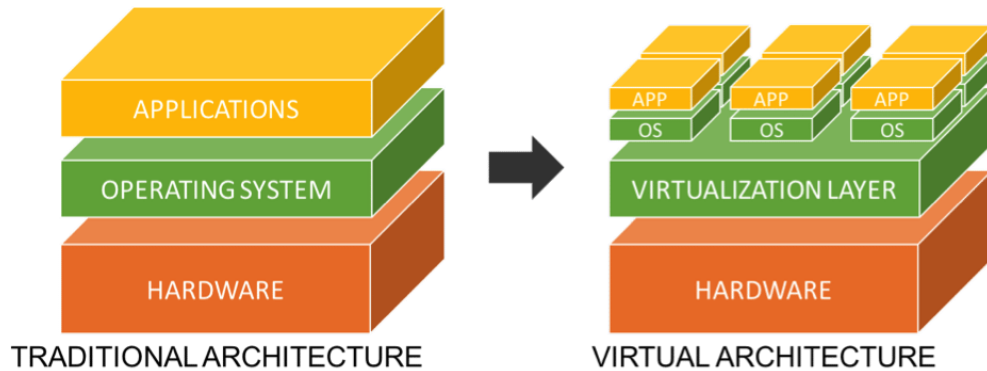It's interesting what's hidden behind it?
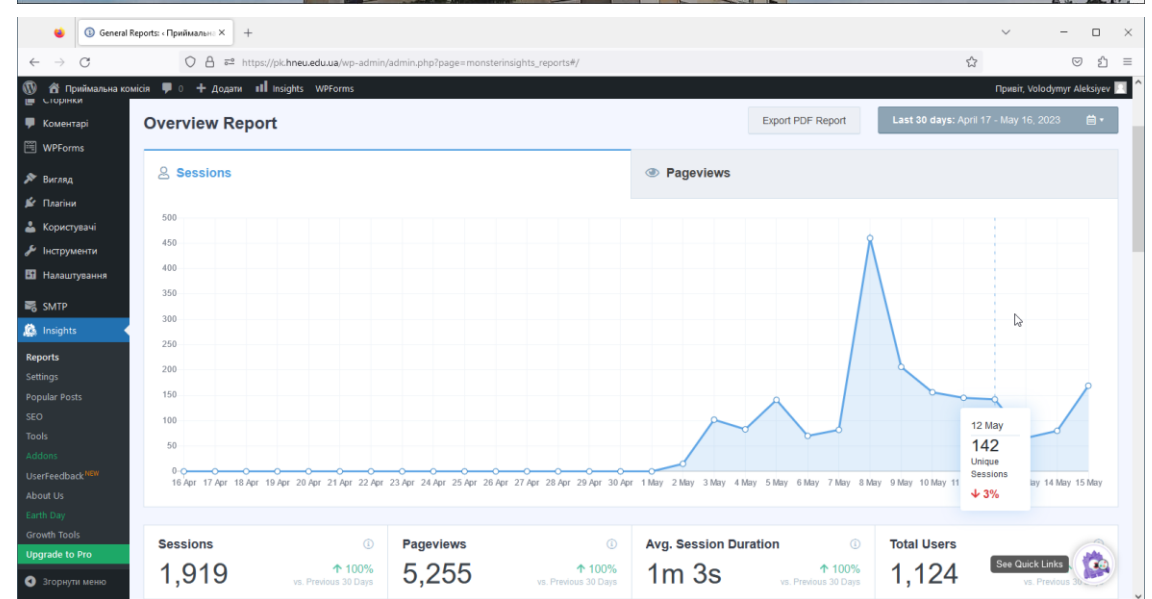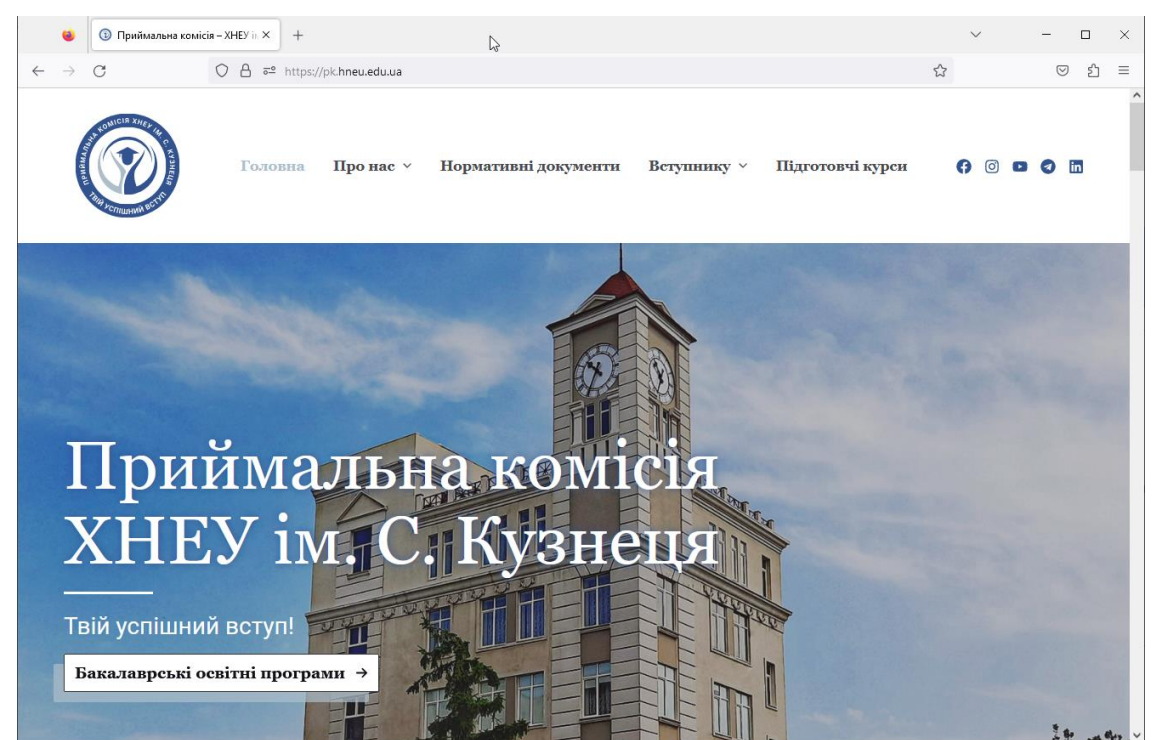
Site's structure
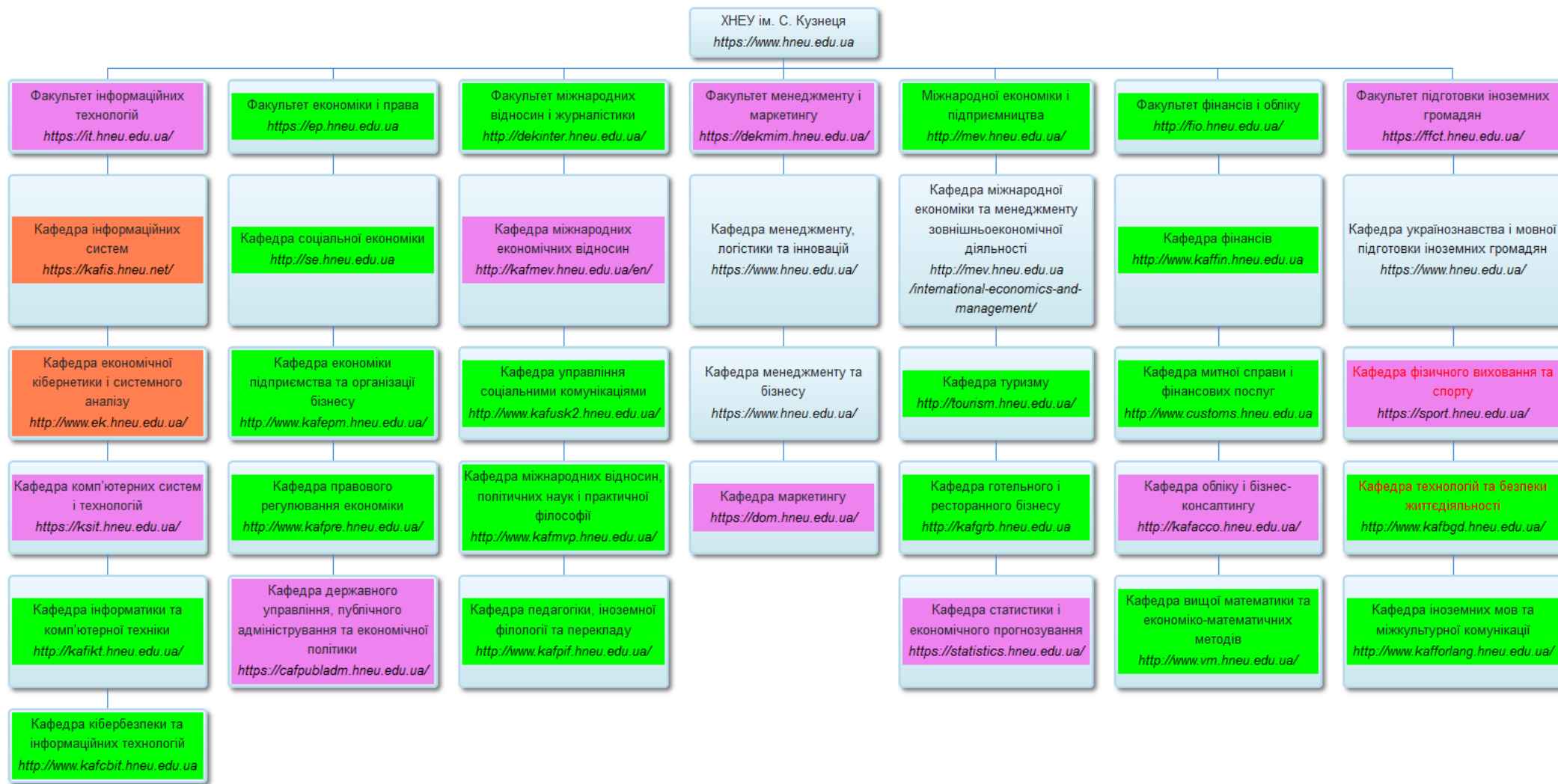
Data center (concept)

A modern data center is built on the basis of the use of virtualization technologies
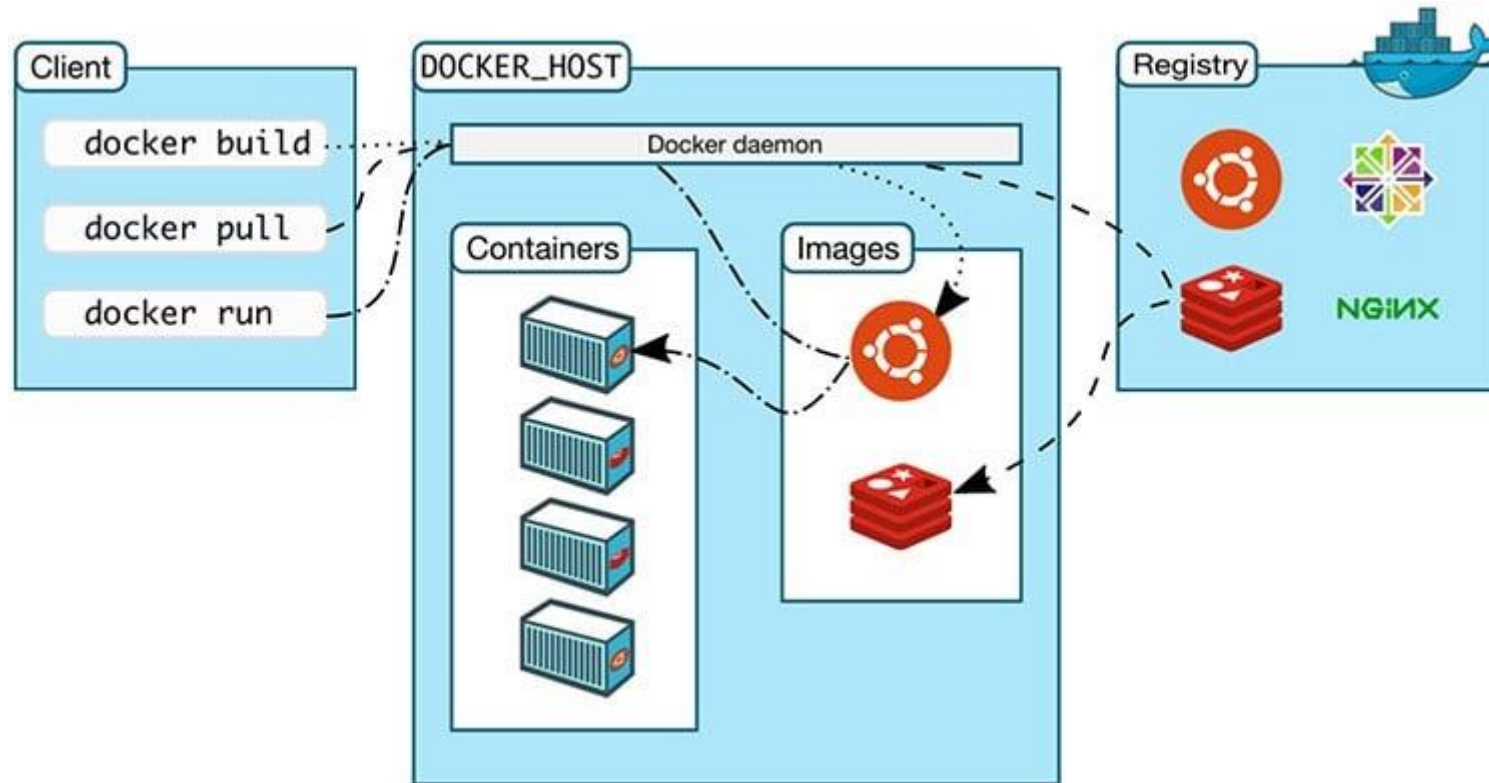

[11 Points to Consider When Virtualizing Security](#)

[Proxmox Virtual Environment](#)

Container Virtualization

[Docker Architecture, Life Cycle of Docker Containers and Data Management](#)

[Containers vs. Virtual Machines (VMs): What's the Difference?](#)

Traditional virtualization or Containers ?

Please note that the above table provides a general overview, and the choice between traditional virtualization and containerization depends on factors such as the nature of the application, scalability requirements, resource utilization goals, deployment needs, and team preferences.

Introducing ChatGPT

Certainly! Here's a comparison table highlighting some key differences between traditional virtualization and containerization:

Aspect	Traditional Virtualization	Containerization
Isolation	Full isolation of hardware and OS	OS-level isolation
Resource Overhead	Higher resource overhead (hardware, OS)	Lower resource overhead (shared OS kernel)
Performance	Slightly lower performance due to hypervisor	Higher performance due to direct access to host resources
Deployment	Slower deployment (virtual machine boot time)	Faster deployment (instant start time)
Scalability	Vertical scaling (adding more resources to VM)	Horizontal scaling (replicating containers)
Portability	Less portable (VMs tied to specific hypervisor)	Highly portable (containers can run on any host)
Resource Utilization	Requires more resources for each virtual machine	Efficient utilization of resources (shared kernel)
Application Density	Lower application density per physical machine	Higher application density per physical machine
Dependency Management	Independent OS and software dependencies per VM	Shared OS and dependencies (reduced duplication)
Development and Testability	Replicating production environment can be complex	Easier to replicate production environment for testing

How To Install Linux, Apache, MySQL, PHP (LAMP) Stack

Ubuntu 22.04

- sudo apt update
- sudo apt install apache2
- sudo ufw app list
- sudo ufw allow in "Apache"
- sudo apt install mysql-server
- sudo mysql
- ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'password';
- exit
- sudo mysql_secure_installation
- sudo apt install php libapache2-mod-php php-mysql
- php –v …

Vesta Control Panel

- \# Connect to your server as root via SSH ssh root@your.server
- \# Download installation script curl -O http://vestacp.com/pub/vst-install.sh
- \# Run it bash vst-install.sh

[How To Install Linux, Apache, MySQL, PHP (LAMP) Stack on Ubuntu 22.04](#)

[Vesta Control Panel](#)

[Docker Hub](#)

Docker

- docker run -d -p 80:80 --name my-apache-php-app -v "$PWD":/var/www/html php:7.2-apache

Microservices or Monolithic ?

Please note that the above table provides a general overview, and the suitability of microservices or monolithic architecture depends on various factors such as the size and complexity of the application, team dynamics, scalability requirements, and more.

Introducing ChatGPT

Scaling up the Prime Video audio/video monitoring service and reducing costs by 90%

May 25, 2023

Certainly! Here's a comparison table highlighting some key differences between microservices and monolithic architecture:

Aspect	Microservices	Monolithic Architecture
Size of Application	Smaller, divided into loosely coupled, independent services	Larger, single codebase/application
Modularity	Highly modular with independent components	Tightly coupled components
Development	Can be developed and deployed independently	Requires coordinated development and deployment
Scalability	Scalability of individual services	Scaling the entire application
Technology Diversity	Allows for the use of different technologies per service	Common technology stack for the entire application
Fault Isolation	Failure in one service doesn't affect others	A single failure can impact the entire application
Deployment	Easier deployment of individual services	Requires deployment of the entire application
Flexibility	Easier to make changes or introduce new features	Changes can be complex and impact the entire system
Team Independence	Enables independent development teams	Collaboration required for shared codebase
Complexity	Distributed system complexity	Simplicity in terms of structure and operations

Opinion

A digital transformation at the data center level has taken place - this is the implementation of container virtualization.

The right choice for new projects is Docker.

It is difficult for educational institution to change the infrastructure to containers and the Kubernetes Cluster architecture. A possible solution is Cloud Computing.

Container virtualization security is ensured by implementing DevSecOps practices.

Thank you

Volodymyr Aleksiyev

Email address: vlax@hneu.edu.ua

Simon Kuznets Kharkiv National University of Economics