O. Mahurin, Student
V. Shadura, Senior Lecturer, language advisor
Zhytomyr Polytechnic State University

TECHNICAL ASPECTS AND COMPARISON OF VIRTUALIZATION ANDCONTAINERIZATION

Virtualization and containerization are considered to be crucial technologies that provide users and organizations with flexible, scalable, and cost-effective infrastructure, allowing them to efficiently deploy, manage, and secure their applications, services, and data ina dynamic and rapidly changing digital landscape.

Seemingly, virtualization and containerization may appear to be similar technologies since they both involve creating virtual environments to run applications and services. However, there are significant differences between these technologies. The mentioned above technologies are usually combined to create a hybrid infrastructure that takes an advantage of the benefits to both virtualization and containerization. By using the both technologies together, even greater flexibility, scalability, and efficiency in managingtheir applications, services, and data can be achieved.

Virtualization is the process of creating a virtual version of something, such as a server, an operating system, a storage device, or network resources. This technology allows multiple virtual machines (VMs) to run on a single physical server, effectively utilizing the available resources and reducing hardware costs. Virtualization also enables organizations to create and manage isolated environments, ensuring that applications and services do not interfere with each other. Moreover, virtualization facilitates the migration of workloads between servers, balancing the workload and ensuring high availability.

Virtualization is a technology that utilizes a software layer known as a hypervisor to create virtual machines on a single physical server. The hypervisor is responsible for managing hardware resources, such as CPU, memory, storage, and network, and allocating them to the virtual machines. Each virtual machine runs its own operating system, applications, and services, and is isolated from other virtual machines running on the same physical server. The hypervisor provides a layer of abstraction that allows virtual machines to run on different hardware platforms without modification. This enables the creation of flexible, scalable, and cost-effective infrastructures that can quickly adapt to changing needs. Additionally, virtualization allows easy migration of workloads between different physical servers or cloud environments, providing high availability and fault tolerance.

Overall,virtualization has become an essential technology for modern data centers, enabling efficient resource utilization and enhancing IT infrastructure management.

Virtual machine hypervisors are categorized into two types: type 1 and type 2. Type 1 hypervisors, also known as bare-metal hypervisors, run directly on the physical server's hardware and are responsible for managing the hardware resources and allocating them to virtual machines. Type 1 hypervisors are designed for high-performance computing and are commonly used in data centers and cloud

environments. Examples of type 1 hypervisors include VMware ESXi, Microsoft Hyper-V, and Citrix Hypervisor.

Type 2 hypervisors, also known as hosted hypervisors, run on top of a host operating system and are more commonly used in desktop environments for running multiple operating systems on a single physical machine. Type 2 hypervisors provide a layer of abstraction between the virtual machines and the host operating system and hardware resources. Examples of type 2 hypervisors include Oracle VirtualBox, VMware Workstation, and Parallels Desktop.

Containerization is a technology that allows applications to run within isolated containers. A container is a lightweight and portable package that contains everything an application needs to run, including code, libraries, and dependencies. Containers enable applications to be deployed quickly and reliably across different environments, without being affected by variations in operating systems or hardware configurations. Containerization also allows for efficient use of resources, as multiple containers can run on a single physical machine, sharing the same operating system kernel.

Technically, containerization is based on the use of a container runtime engine, such as Docker or Kubernetes, which creates and manages containers. The container runtime engine uses a layered file system, where each layer represents a change made to the container image. This allows for efficient use of storage space and enables quick and reliable deployment of applications. Containers are also isolated from the host operating system and other containers, providing an additional layer of security. Additionally, containers can be easily orchestrated and managed using container orchestration tools, such as Kubernetes, which automate the deployment, scaling, and management of containers across multiple hosts.

Docker uses several tools to run containers, from low-level to high-level ones. At thelowest level a runc is a low-level container runtime that follows the OCI standard and uses native Linux features to create and run containers. Above runc is containerd, a high-level container runtime that adds features such as image transfer, storage, and networking, and fully supports the OCI specification. The Docker daemon, being dockered, sits above containers and provides a standard API for interacting with them. Finally, the Docker CLI tool, docker-cli, allows users to control containers through commands without a need to understand the lower levels. In practice, when running a container with Docker, it goes through the Docker daemon, which calls containerd, that then uses runc.

Virtualization and containerization are two distinct techniques with different outcomes. Virtualization offers fully isolated operating systems and virtual machine instances, while containerization isolates the host operating system and containers from one another, sharing a kernel. Additionally, virtualization can host multiple operating systems with their own kernels, while containerization runs all containers via a user mode on a single operating system. While virtualization allows for a range ofoperating systems to be used on the same server or machine, containerization is relianton the host OS, meaning Linux containers cannot run on Windows and vice-versa. Furthermore, virtualization uses failover clusters with load balancing support, whereas containerization uses orchestration via Docker or Kubernetes to start and stop containers, maximizing resource utilization. Finally, virtualization uses virtual network adaptors running through a master network interface card for networking, while

containerization splits the VNA into multiple isolated views for lightweight network virtualization.

Hence, virtualization and containerization are two distinct technologies that offer different advantages and disadvantages. While virtualization allows for the creation of multiple virtual machines with their own operating systems, containerization operates on a single host OS and can be much faster and more lightweight. Both technologies have their own benefits and are often used together in modern IT environments.

REFERENCES

1. Bigelow S. J. What's the difference between Type 1 vs. Type 2 hypervisor? | TechTarget. URL: https://www.techtarget.com/searchitoperations /tip/Whats-the-difference-between-Type-1-vs-Type-2-hypervisor (date of access: 09.04.2023).

2. Containerization vs. virtualization: 7 technical differences | trianz. Accelerating Digital Evolution - Trianz. URL: https://www.trianz.com/insights/c on tainerization - vs - virtualizatio n # : ~ : text = What % 2 0 i s % 2 0Containerization ?,isolate%20processes%20from%20one%20another. (date of access:09.04.2023).

3. The differences between Docker, containerd, CRI-O and runc. TutorialWorks. URL: https://www.tutorialworks.com/difference-docker-containerd-runc-crio- oci/ (date of access: 09.04.2023).