Chapter: 06

CONTAINERIZATION AND VIRTUALIZATION

Mohd Naved Ul Haq*
Faculty, Glocal School of Science and Technology,
Glocal University, Saharanpur, U.P.
**Correspondence to:* naved@theglocaluniversity.in

Mohit Kumar Sharma
Faculty, Glocal School of Science and Technology,
Glocal University, Saharanpur, U.P.

DOI: ***https://doi.org/10.52458/9788196869434.2023.eb.grf.ch-06***
Ch.Id:-GU/GRF/EB/ETCSIA/2023/Ch-06

ABSTRACT

Virtualization and containerization are two revolutionary technologies that are thoroughly examined in this chapter. The ability to efficiently share processing, storage, and network resources is made feasible by virtualization, as it is detailed in this article. It talks about the function of hypervisors, the isolation of virtual machines, and the adaptability this technology provides. The lightweight virtualization method known as containerization, in contrast, is advertised as packing applications and their dependencies into transportable containers. The process of containerization is covered in detail in this chapter, along with container images, resource separation, orchestration, and the inherent portability of containers.

In addition, the chapter contrasts traditional and virtual architecture's architectural features, emphasizing the transition from physical to digital construction and the consequences of tangibility, materials, iteration, and applications in each strategy.

Both virtualization and containerization's effects on security are covered, with a focus on the necessity of routine patching, access control, and vulnerability testing to reduce any dangers.

The chapter ends by highlighting the importance of these technologies in contemporary computing. It draws attention to their contributions to resource usage optimization, increased flexibility, cost savings, and consistent application deployment from development to production environments. The information in this chapter acts as a thorough reference for comprehending these crucial technologies and the variety of uses to which they are put in the current IT environment.

Keywords: *Virtualization, Containerization, Resource Sharing, Hypervisor, Virtual Machines, Container Images.*

INTRODUCTION OF VIRTUALIZATION

A technology called virtualization makes it possible to create virtual replicas of actual resources including computers, servers, storage units, and networks. Despite using shared physical hardware, these virtual instances behave as though they are separate, self-contained entities. From data centers and cloud computing to desktop environments, virtualization is employed in a variety of settings and has several advantages, including effective resource usage, flexibility, and cost savings [1].

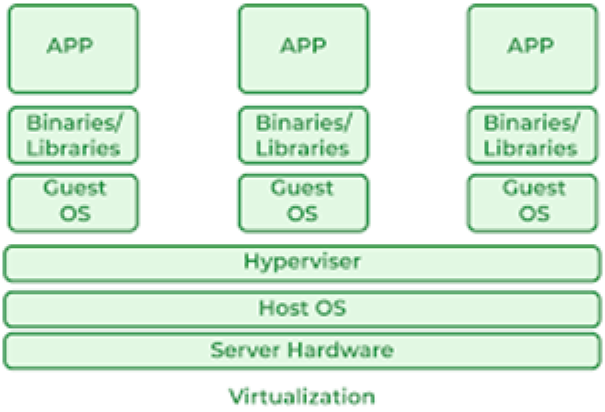

Figure-1: Virtualization

Virtualization's capacity to abstract and separate the underlying hardware from the software and applications running on it is one of its primary features. Numerous benefits result from this abstraction, including the ability to run different operating systems on a single physical server, simplified virtual machine deployment, and improved disaster recovery capabilities.

Jeff Smith, the author of "Virtualization Essentials," teaches the core ideas and procedures of virtualization. He explores the complexities of virtualization technologies, providing explanations of how they operate and their real-world applications for businesses and IT settings. Smith's thorough review makes it easier for people and companies to comprehend the relevance of virtualization in contemporary computing and how it contributes to resource optimization, increased agility, and lower operational costs.

ARCHITECTURE: TRADITIONAL VS. VIRTUAL

Two different methods of architectural design and construction are represented by traditional and virtual architecture. The main variations between the two are as follows:

i. **Digital vs. Physical**

- **Traditional Architecture:** Real-world elements like concrete, steel, wood, and glass are used to physically construct structures and spaces in traditional architecture. The material and observable parts of the architecture are covered.

- **Virtual Architecture:** It is concerned with the construction of digital or virtual spaces. It entails creating and visualizing architectural concepts in a digital manner using computer-aided design (CAD), 3D modeling, and other digital tools.

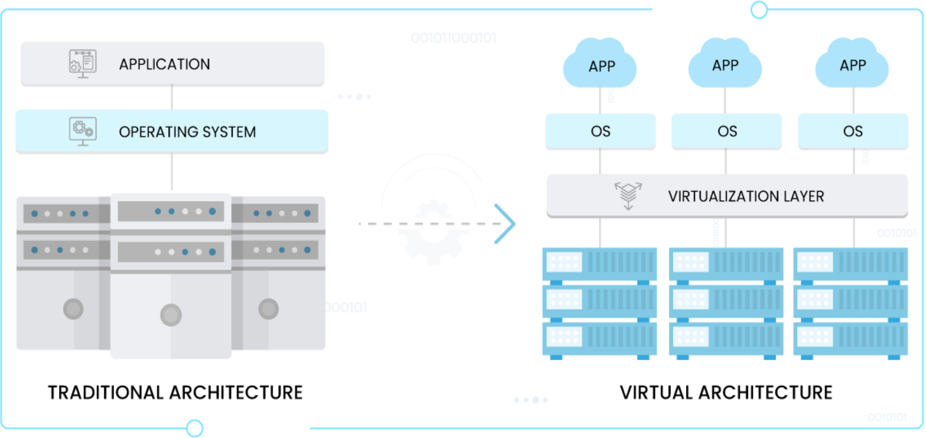

Figure-2: Traditional Architecture Vs Virtual Architecture

ii. Tangibility

- **Traditional Architecture:** Traditional architecture produces real-world physical structures. The physical senses can be used to view, feel, and experience these structures and areas.

- **Virtual Architecture:** Intangible virtual architecture. It is available in a digital format, and virtual architectural encounters are often mediated by screens, VR headsets, or other electronic devices.

iii. Building and Materiality

- **Traditional Architecture:** The design and construction processes in traditional architecture involve the use of real materials, specific construction methods, and adherence to set building norms and regulations.

- **Virtual Architecture:** Architecture that doesn't entail actual physical construction is known as virtual architecture. Instead, it focuses on how architectural plans are visually and interactively represented in a digital world.

iv. **Iteration and Flexibility**

- **Traditional Architecture:** Making changes to a traditional architectural design after work has begun can be difficult and expensive, and they frequently call for actual physical changes to the building itself.

- **Architecture in the virtual world:** Virtual architectural designs can be quickly changed and improved upon digitally without requiring significant physical adjustments. This adaptability enables experimentation and prompt design modifications

v. **Applications**

- **Traditional Architecture:** Traditional architecture is used to construct physical structures including homes, businesses, historical sites, and transportation infrastructure like bridges and highways.

- **Virtual Architecture:** Digital simulations for training or education, architectural visualization, virtual reality applications, augmented reality experiences, and video game design are all common uses for virtual architecture.

vi. **Interaction**

- **Traditional Architecture:** Interactions with traditional architecture occur in the physical world, involving activities like walking through doorways, climbing stairs, and opening windows.

- **Virtual Architecture:** Interactions with virtual architecture are mediated through digital interfaces, often involving navigation within virtual spaces, manipulating objects, or experiencing immersive environments in virtual or augmented reality.

vii. **Environmental Impact**

- **Traditional Architecture:** Sustainable design in traditional architecture aims to minimize the environmental impact of physical buildings, considering factors like energy efficiency, material selection, and long-term sustainability.

- **Virtual Architecture:** While virtual architecture does not involve physical materials, it is not without environmental considerations, such as the energy consumption associated with digital technologies and data centers.

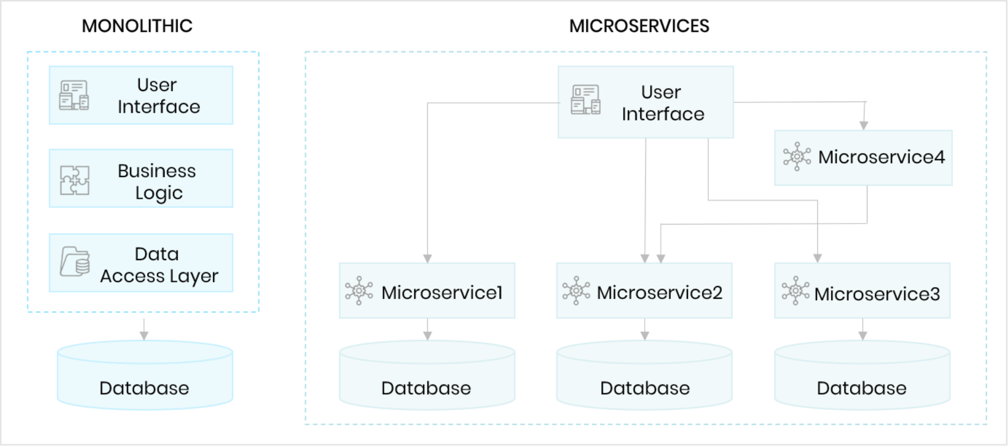

Figure-3: Monolithic Vs Microservices

VIRTUALIZATION'S BENEFITS AND DRAWBACKS

Consolidating physical servers is made possible by the technique of virtualization, which enables a physical computer with computing, network, and storage capabilities to share its resources with several virtual ones. On an underlying hypervisor, different operating systems are run by virtual machines. They are adaptable and simple to control. Because virtualization enables the independent operation of several operating systems, it isolates workloads and makes simple disaster recovery solutions possible. By using less physical hardware, it also conserves resources and money. Databases and important applications are ideal candidates for virtual machine operation.

Virtualization can have some disadvantages, though. Running many virtual machines requires a significant quantity of memory, central processing unit (CPU), and disk space. Additionally, each virtual computer needs its own

CONTAINERIZATION'S BENEFITS AND DRAWBACKS

The deployment of code inside a portable, safe, light-weight program that has all necessary files, configurations, and dependencies is made possible by containerization. Containers don't have an operating system; they just have the tools and libraries needed to run their applications. Developers have adopted containerization software packages like Docker over the past ten years because they make creating and deploying cloud-native applications easier

Containers have low resource and expense requirements. They are adaptable, portable, and speed up application deployment. Additionally, their integrated version control makes it simple to access earlier versions. The infrastructure needed to host containers is complicated, which is a drawback. To install and support it, you need a hybrid infrastructure and development team.

VIRTUALIZATION'S EFFECTS ON SECURITY

For an on-premises virtual environment to remain safe and secure, regular hardware and hypervisor patching is essential. The cloud service provider will take care of patching the hypervisor and hardware when using the cloud. Regular operating system patching is essential in both cases.

To prevent additional harm, a compromised virtual machine needs to be isolated and shut down. Attackers, however, can seize control of all linked virtual machines if they manage to infiltrate the underlying hypervisor.

It is essential to make sure that the appropriate rules and configurations are utilized during the deployment of a virtual machine and that default public access is disabled. Even if the virtual machine operates in the cloud, this is still true. Additionally, proper administration of virtual machine images necessitates regular image audits. The use of strict access controls, network segmentation, and encryption helps minimize the security threats brought on by virtualization. Because virtual computers frequently contain several installed apps, it's crucial to perform regular penetration and vulnerability testing.

EFFECTS OF CONTAINERIZATION ON SECURITY

Compared to virtual machines, containers have a reduced attack surface. However, there are security issues to take into account. The worst is container eruption, in which an attacker uses a flaw in the container to obtain access to the host operating system. The most recent security patches must be applied to the container image, engine, and orchestration systems, and they must be routinely checked for any vulnerabilities. Organizations should carry out regular audits and put in place security procedures and policies that are particular to containers. Vulnerabilities may be helped by routine releases and updates as part of the continuous integration and delivery (CI/CD) pipelines. Since the introduction of containerization, security has moved from being predominantly the domain of an operations (Ops) team to beginning with developers. By providing pod security policies to aid in securing container development, orchestration platforms like Kubernetes play a crucial role in lowering risk as container usage grows. Logins, code, and configurations can be examined by tools for container monitoring and scanning to identify vulnerabilities before they are deployed. It's also crucial to manage permissions to provide effective container security. Security during container running is very important and challenging to manage. By concentrating on application security and establishing container firewalls to track incoming and outgoing traffic, cybersecurity teams may meet this challenge. Due to the lack of cybersecurity experts, businesses should think about automating routine security operations to reduce the risk of attacks

HOW DOES VIRTUALIZATION OPERATE?

With the aid of the technology known as virtualization, you may construct virtual versions of physical resources like servers, storage units, and networks. It makes it possible for several virtual machines (VMs) to operate on a single physical computer while sharing its resources. Here is a description of how virtualization functions:

A sotware element called a hypervisor, commonly referred to as a Virtual Machine Monitor or VMM, lies at the core of virtualization. Between the virtual computers and the physical hardware is the hypervisor. The physical resources are managed and distributed to the virtual machines.

Virtual machine creation and management are handled by the hypervisor. Each virtual machine (VM) replicates the hardware of a physical system in a separate environment. Its virtual CPU, RAM, storage, and network are all independent.

- **Resource Allocation:** Virtual machines receive physical resources from the hypervisor. It makes sure that each VM receives an equitable allocation of CPU, memory, and other resources. In accordance with each VM's workload, it can also dynamically assign resources.
- **Isolation:** Virtualization effectively isolates VMs from one another. As a result, if one VM crashes or experiences other problems, it won't impact other VMs that are currently running on the same physical host. Every VM runs separately.
- Virtualization abstracts the underlying hardware through the use of virtualization. This implies that VMs are ignorant of the specifics of the underlying hardware. They communicate with the hypervisor-managed virtualized hardware.
- Virtualization enables you to take snapshots of VMs at specific points in time. This is known as cloning. VMs can also be copied.

- Virtual machines can be simply relocated from one physical host to another using migration and live migration. Particularly with live migration, VMs can be relocated without experiencing any downtime.
- **Management and orchestration:** Software tools that offer a central interface to create, configure, monitor, and operate virtual machines can be used to manage virtualization. Additionally, VM deployment and management may be automated and orchestrated with the help of these tools.
- Different types of virtualization exist, including network virtualization (for constructing virtual network segments), desktop virtualization (for running multiple desktop OS instances on a single client device), and server virtualization (for running multiple server OS instances on a single physical server).

WHAT IS THE PROCESS OF CONTAINERIZATION?

An application and all of its dependencies can be packaged into a single unit called a container using containerization, a simple type of virtualization. These containers are transportable and can function reliably in a range of settings, from development to production.

This is how containerization functions:

Container Engine: A container engine, like Docker or container, is necessary for containerization to function. A host operating system is used by the container engine, which makes it easier to create, deploy, and use containers.

Container Picture: A container is created from a picture. The application's code, runtime, system tools, libraries, and settings are all contained in this image, which is a compact, standalone executable package. Frequently, container images are created using a set of instructions included in a Docker file or similar configuration.

Image Repository: A container registry, such as Docker Hub or Google Container Registry, is often where container images are kept. You can publish, share, and version container images using these repositories.

Create a container by running an instance of a container image using the container engine. The image serves as a model for the container engine while building a running container. Containers are more lightweight than conventional virtual machines because they share the host OS kernel despite being segregated from the host and from one another.

Isolation: Because each container has its own file system and process space, containers offer process-level isolation. Containers are safe and secure because of this isolation, which prevents applications and their dependencies from interfering with one another predictable.

You can limit the CPU, memory, and network resources of containers by setting resource constraints. By doing this, you may effectively use the resources of the host system and stop one container from depleting the resources of another.

Connecting containers to networks will allow them to communicate with one another and with outside services. For instance, Docker offers networking features that let you create private networks and attach containers to them.

Orchestration: To automate the deployment, scaling, and administration of containerized applications in production environments, solutions like Kubernetes, Docker Swarm, or Amazon ECS are frequently employed. These technologies manage scalability, health checks, failover, load balancing, and complicated applications made up of numerous containers, which makes managing them simpler.

Containers can be moved around easily. You can construct a container image on one system and run it on any other system that supports the containerization technology since they package an application and its dependencies together. This guarantees consistency between the environments used for development, testing, and production.

Modern software development and deployment now rely heavily on containerization. It supports consistency throughout the many stages of the software development lifecycle, from local development environments to production deployments, speeds up the delivery of applications, and optimizes development workflows

CONCLUSION

Virtualization and containerization have been thoroughly covered in this chapter as two important technologies. The computing landscape has changed as a result of these advancements, which provide strong solutions for resource optimization, scalability, and application management. With its capacity to efficiently manage physical resources and abstract them, virtualization has emerged as a key component of data centers and cloud computing. It makes it possible to create virtual computers that are adaptable, affordable, and isolated to improve disaster recovery capabilities.

On the other side, the deployment of applications has been transformed by containerization, a more lightweight form of virtualization. Containers are a desirable option for contemporary software development and deployment because of their portability, security, and quick scalability.

The chapter also emphasized the differences in architecture between conventional and virtual techniques, demonstrating the transition from physical to digital realms. Additionally, it covered the critical security issues related to virtualization and containerization.

These innovations have not only transformed the IT sector but have also created new paths for innovation, enabling organizations and IT environments to function more effectively, quickly, and securely in the always changing digital environment.

REFERENCES

1. *Mastilak, L., Helebrandt, P., Galinski, M., & Kotuliak, I. (2022). Secure inter-domain routing based on blockchain: A comprehensive survey. Sensors, 22(4), 1437.*

2. *Thounthong, P., Sikkabut, S., Mungporn, P., Tricoli, P., Nahid-Mobarakeh, B., Pierfederici, S., ... & Piegari, L. (2014, June). Differential flatness control approach for fuel cell/solar cell power plant with Li-ion battery storage device for grid-independent applications. In 2014 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (pp. 261-266). IEEE.*

3. *Kaur, P., Josan, J. K., & Neeru, N. (2022, March). Performance analysis of docker containerization and virtualization. In Proceedings of Third International Conference on Communication, Computing and Electronics Systems: ICCCES 2021 (pp. 863-877). Singapore: Springer Singapore.*

4. *Bentaleb, O., Belloum, A. S., Sebaa, A., & El-Maouhab, A. (2022). Containerization technologies: Taxonomies, applications and challenges. The Journal of Supercomputing, 78(1), 1144-1181.*

5. *https://www.baeldung.com/cs/virtualization-vs-containerization*