Containers

examlabpractice.com

What are containers?

- Containers are a technology for packaging and running Windows and Linux applications across diverse environments on-premises and in the cloud.
- Containers provide a lightweight, isolated environment that makes apps easier to develop, deploy, and manage. Containers start and stop quickly, making them ideal for apps that need to rapidly adapt to changing demand.
- The lightweight nature of containers also make them a useful tool for increasing the density and utilization of your infrastructure.

Diversity of Containers

Anywhere

On-premises

Cloud

Any app

Monolith

Microservice

Any language

Java

.Net

Python

Node

The Microsoft Container Ecosystem

The Microsoft container ecosystem

Microsoft provides a number of tools and platforms to help you develop and deploy apps in containers:

- Run Windows-based or Linux-based containers on Windows for development and testing using Docker Desktop, which makes use of containers functionality built-in to Windows. You can also run containers natively on Windows Server.
- Develop, test, publish, and deploy Windows-based containers using the powerful container support in Visual Studio and Visual Studio Code, which include support for Docker, Docker Compose, Kubernetes, Helm, and other useful technologies
- Publish your apps as container images to the public DockerHub for others to use, or to a private Azure Container Registry for your org's own development and deployment, pushing and pulling directly from within Visual Studio and Visual Studio Code.

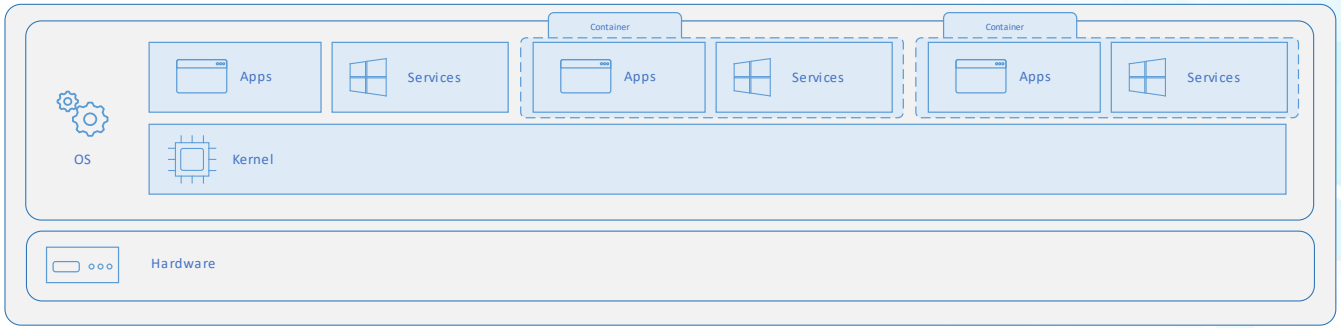
Azure support for containers

Deploy containers at scale on Azure or other clouds:

- Pull your app (container image) from a container registry, such as the Azure Container Registry, and then deploy and manage it at scale using an orchestrator such as Azure Kubernetes Service (AKS) or Azure Service Fabric.
- Azure Kubernetes Service deploys containers to Azure virtual machines and manages them at scale, whether that's dozens of containers, hundreds, or even thousands
- Deploy containers on-premises by using AKS on Azure Stack HCI, Azure Stack with the AKS Engine, or Azure Stack with OpenShift. You can also set up Kubernetes yourself on Windows Server (see Kubernetes on Windows

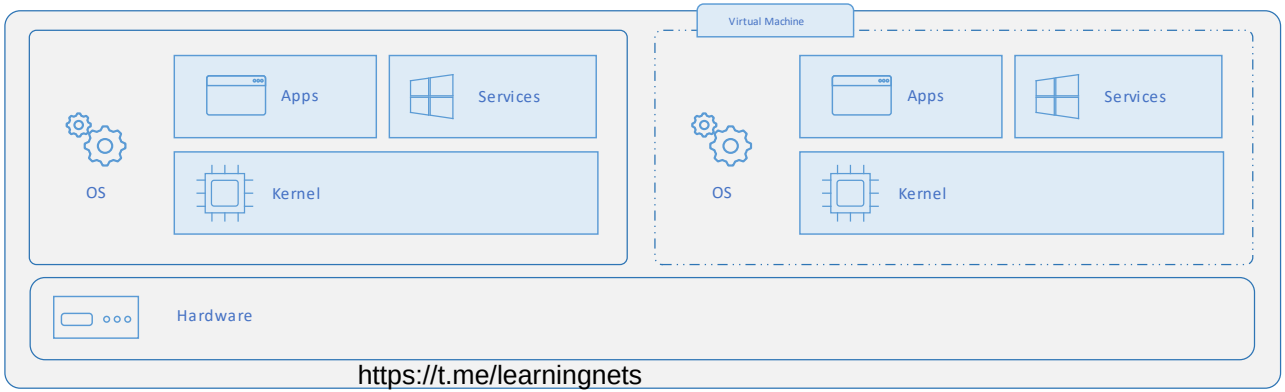
How containers work

- A container is an isolated, lightweight package for running an application on the host operating system.
- Containers build on top of the host operating system's kernel (which can be thought of as the buried plumbing of the operating system)



Containers vs. virtual machines

In contrast to a container, a virtual machine (VMs) runs a complete operating system–including its own kernel–as shown in this diagram.



Container images

All containers are created from container images. A container image is a bundle of files organized into a stack of layers that resides on your local machine or in a remote container registry. The container image consists of the user mode operating system files needed to support your app, any runtimes or dependencies of your app, and any other miscellaneous configuration file your app needs to run properly.

Microsoft offers several images (called base images) that you can use as a starting point to build your own container image:

- Windows - contains the full set of Windows APIs and system services (minus server roles).
- Windows Server - contains the full set of Windows APIs and system services.
- Windows Server Core - a smaller image that contains a subset of the Windows Server APIs–namely the full .NET framework. It also includes most but not all server roles (for example Fax Server is not included).
- Nano Server - the smallest Windows Server image and includes support for the .NET Core APIs and some server roles.