



Containers 101

A container is a lightweight, portable approach to running multiple applications on the same operating system kernel. Applications are isolated and packaged only with their unique dependencies, allowing for increased density because containers consume fewer resources than traditional virtual machines.

Why containers?



Developers

- Unlock ultimate productivity and freedom
- Deploy multitier distributed apps to any environment, on-premises to cloud
- Focus on standardized app infrastructure



Operations

- Provide standardized environments for development, QA, and production teams
- Achieve higher utilization and compute density
- Rapidly scale up or down to meet changing business needs



DevOps

- Integrate people, processes, and tools for optimized app development
- Focus on standardized infrastructure
- Allow developers to focus on building, deploying, and testing apps

Datacenter to cloud

On-premises

Service provider

Azure

Developer tools

The key for developers is the ability to choose the right tool at the right time.



Benefits containers enable:

- Rapid deployment
- Track changes / rollback
- Greater flexibility

Development framework and languages

PHP

Python

Go

Node

JavaScript

Perl

Win32

C++

.Net

Java

Ruby

Docker integration

Docker Hub:

Search and download thousands of public and curated images.

Docker Engine:

Docker Engine for Windows Server is part of the Docker open source project.

Docker Client:

One consistent CLI experience regardless of development environment.

Collaboration:

Bring Windows Server containers to the Docker ecosystem to expand the reach of both developer communities.

Windows

Linux

docker

The technology

Container

- No virtualized hardware components
- Self-contained instances of apps, dependencies and minimal OS components
- High resiliency due to abstraction
- Highly portable regardless of targeted host

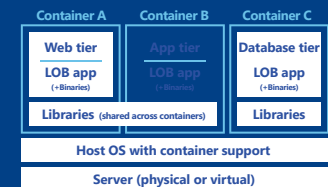
Virtual machine

- Fully virtualized set of abstracted hardware and drivers
- Full production OS with maintenance, patching, and security protocols per VM
- Higher resource consumption
- Portability requires moving the entire virtual machine

Deployment Types

Windows Server Containers

Developers can use Visual Studio and other tools to build modular apps that run within containers on shared kernels. Container capabilities are built into Windows Server, and they can be deployed with PowerShell or Docker.



Hyper-V Containers

Hyper-V Containers use the same APIs as Windows Server Containers and are built with Hyper-V virtualization technology on isolated kernels. The virtualization layer and OS are optimized for containers.

