

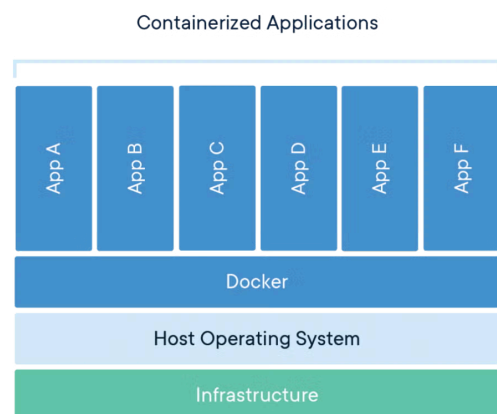


# Use containers to Build, Share and Run your applications

**Package  
Software into  
Standardized  
Units for  
Development,**

# Shipment and Deployment

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings. Container images become containers at runtime and in the case of Docker containers – images become containers when they run on [Docker Engine](#). Available for both Linux and Windows-based applications, containerized software will always run the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite



differences for instance  
between development and  
staging.

Docker containers that run on  
Docker Engine:

**Standard:** Docker created the  
industry standard for  
containers, so they could be  
portable anywhere

**Lightweight:** Containers  
share the machine's OS  
system kernel and therefore  
do not require an OS per  
application, driving higher  
server efficiencies and  
reducing server and licensing  
costs

**Secure:** Applications are  
safer in containers and  
Docker provides the  
strongest default isolation  
capabilities in the industry

[Learn more](#)

**Docker  
Containers Are  
Everywhere:  
Linux,**



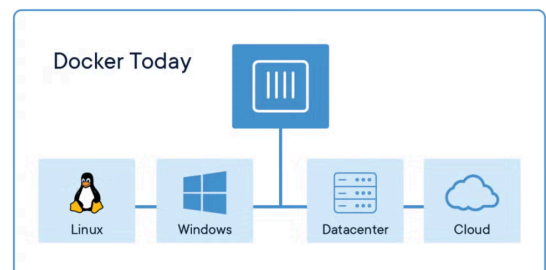
# Windows, Data center, Cloud, Serverless, etc.

Docker container technology was launched in 2013 as an open source [Docker Engine](#).

It leveraged existing computing concepts around containers and specifically in the Linux world, primitives known as cgroups and namespaces. Docker's technology is unique because it focuses on the requirements of developers and systems operators to separate application dependencies from infrastructure.

Success in the Linux world drove a partnership with Microsoft that brought Docker containers and its functionality to Windows Server.

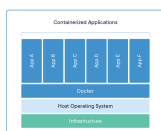
Technology available from Docker and its open source project, Moby has been



leveraged by all major data center vendors and cloud providers. Many of these providers are leveraging Docker for their container-native IaaS offerings. Additionally, the leading open source serverless frameworks utilize Docker container technology.

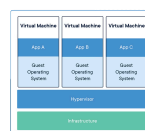
# Comparing Containers and Virtual Machines

Containers and virtual machines have similar resource isolation and allocation benefits, but function differently because containers virtualize the operating system instead of hardware. Containers are more portable and efficient.



## CONTAINERS

Containers are an abstraction at the app layer that packages code and dependencies together. Multiple containers



## VIRTUAL MACHINES

Virtual machines (VMs) are an abstraction of physical hardware turning one server into many servers. The hypervisor allows

can run on the same machine and share the OS kernel with other containers, each running as isolated processes in user space. Containers take up less space than VMs (container images are typically tens of MBs in size), can handle more applications and require fewer VMs and Operating systems.

multiple VMs to run on a single machine. Each VM includes a full copy of an operating system, the application, necessary binaries and libraries – taking up tens of GBs. VMs can also be slow to boot.

## **Containers and Virtual Machines Together**

Containers and VMs used together provide a great deal of flexibility in deploying and managing app

# **Container Standards and Industry Leadership**



The launch of Docker in 2013 jump started a revolution in application development – by democratizing software containers. Docker developed a Linux container technology – one that is portable, flexible and easy to deploy. Docker open sourced libcontainer and partnered with a worldwide community of contributors to further its development. In June 2015, Docker donated the container image specification and runtime code now known as runc, to the Open Container Initiative (OCI) to help establish standardization as the container ecosystem grows and matures.

Following this evolution, Docker continues to give back with the containerd project, which Docker donated to the Cloud Native Computing Foundation (CNCF) in 2017. containerd is an industry-standard container runtime that leverages runc and was created with an emphasis on



simplicity, robustness and portability. containerd is the core container runtime of the Docker Engine.

[Learn more about containerd](#)

## Choose a plan that's right for you

[See pricing](#)

Docker isn't just for personal projects. Discover the perfect plan to empower your team and streamline your workflow.

### Products

[Products Overview](#)

[Docker Desktop](#)

[Docker Hub](#)

[Docker Scout](#)

[Docker Build Cloud](#)

[Testcontainers Desktop](#)

[Testcontainers Cloud](#)

[Docker MCP Catalog and Toolkit](#)





## Features

Command Line Interface

IDE Extensions

Container Runtime

Docker Extensions

Trusted Open Source Content

Secure Software Supply Chain

## Developers

Documentation

Getting Started

Trainings

Extensions SDK

Community

Open Source

Preview Program

Newsletter

## Pricing

Personal

Pro

Team

Business

Pricing FAQ

Contact Sales

## Company

About Us

What is a Container

Blog

Why Docker

Trust

Customer Success

Partners

Events



[Docker System Status](#)

[Newsroom](#)

[Swag Store](#)

[Brand Guidelines](#)

[Trademark Guidelines](#)

[Careers](#)

[Contact Us](#)

**Languages**

[English](#)

[日本語](#)



© 2025 Docker Inc. All rights reserved  
[Terms of Service](#)

[Privacy](#)[Legal](#)

[Cookies](#)  
[Settings](#)

