



Build, ship, and run any application, anywhere with Windows Server 2016 and Docker Enterprise Edition

Improve your apps faster

Industry trends such as cloud and containers have revolutionized how modern enterprises architect their application environments. For example, in a recent survey of CIOs, 44 percent say they want to adopt DevOps, with three out of every top four initiatives revolving around the need to modernize apps.¹ The cloud enables faster innovation and delivery of time-to value through cloud-native applications and microservices architectures. But most businesses are grappling with how to manage and update thousands of existing applications.

Customers expect 100 percent uptime and apps that work in multiple environments. Enterprises need to deliver just-in-time speed, portability, and continuous integration, without sacrificing flexibility. Yet, many solutions address only one app or pain point—or lock you in under the guise of control.

You need a solution that helps you invigorate existing applications and create new, cloud-native apps. Thanks to a partnership between Microsoft and Docker, Inc., Windows Server 2016 with Docker Enterprise Edition (EE) can do both.

One platform, one journey for all applications

Every version of Windows Server 2016 grants access to a Docker Enterprise Edition, enabling the use of containers in the Windows Server app development and management ecosystem. Organizations can now securely build, ship, and run any app, across any infrastructure—from desktop to datacenter to public cloud.

The partnership provides the agility, portability, and control of the Docker platform to Windows developers and IT pros. Together, Docker and Microsoft address 98 percent of enterprise app requirements. Windows Server containers help secure and modernize existing enterprise .NET and line-of-business server applications with little or no code changes. Package existing apps in containers to realize the benefit of a more agile DevOps model, then deploy on-premises, to any cloud, or in a hybrid model. And reduce infrastructure and management costs for those applications, as well.

¹ State of App Development Survey: Q1 2016 Morgan Stanley CIO Survey: June 30, 2016

A one-two punch

Security giant Tyco uses Windows Server containers built on the CS Docker Engine to keep life-safety monitoring applications running without interruption. Containers enable Tyco to update older applications in a modular fashion. Tyco can also scale applications to handle larger monitoring loads and run those applications in the cloud to reduce costs. Systems run smoothly so everyone can sleep better at night.

“By containerizing legacy applications using Windows Server containers, we gain better consistency and control between developers, testers, and deployment teams—a full DevOps environment—without changing the application.”

– Matthew Roberts, Principle Software Engineer, Tyco International

Microsoft + Docker = 98% of workloads

Any OS	 Linux	 Windows
Anywhere	 Physical	 Azure Cloud
Any App	 Traditional	 Micro Services
Any Language	.NET Microsoft	 Open Source

Operate at the speed of innovation

Using Windows Server containers, developers and IT pros can rely on a consistent application journey for Windows and Linux applications, without needing to worry about breaking systems or workflows. Developers can ship more software faster, and IT pros gain the control they require for a flexible and secure enterprise application environment.

Ship more software, faster

Accelerate your public and hybrid cloud strategy by containerizing Windows workloads on-premises or in the cloud, with the option for enhanced isolation in Hyper-V virtual machines (VMs). Get a reliable cloud-ready OS that supports current workloads, with new capabilities to build, ship, and run containerized applications and trusted enterprise-class support from Microsoft, including available service level agreements (SLAs) and hotfixes.

The partnership integrates the Microsoft and Docker portfolios of developer tools, operating systems, and cloud infrastructure. Businesses benefit from an agile container environment and standardized enterprise application workflows, with one journey for all applications.

Containerize existing apps or build cloud-native apps

Windows Server 2016 offers three ways to benefit from this partnership. Containerize existing Windows-based applications, incrementally deliver a microservices architecture, or use Nano Server as the container image for new development.

Developers can **containerize traditional applications** with few or no code changes. Gain portability and efficiency by enabling “write-once, run-anywhere” apps that can be deployed on-premises, to any cloud, or in a hybrid architecture across clouds. Enjoy cost savings over traditional app maintenance. For an additional layer of security in multitenant environments, add Hyper-V isolation, which packages the container image with its own dedicated kernel. And security is a real concern, with the average business cost of a security breach placed at \$4 million.²

You can also transform monolithic apps to microservices. Take advantage of a distributed approach to separate application functionality into separate, smaller services, which you can then deploy independently, using containers for greater agility and availability.

Developers can also build and test new apps in containers, in a development environment such as Visual Studio, or one from a third party.

Using Nano Server as the container image, developers can build offline, customized OS images optimized for your application. Nano Server offers both high performance and a reduced attack surface. Developers push the containers to a central registry, such as the Microsoft Azure Container Registry. IT operations then automates deployment of the containers from the repository to physical or virtual target machines. Operations monitors the containers to provide insight into app usage. If this insight drives an update—or rollback—to a container, developers can perform that update on their own boxes, iterate a version, and deploy the updated version to the central repository, which is then used to update existing deployed containers.

The right tools

This partnership between Microsoft and Docker spans the Microsoft portfolio.

Visual Studio Tools for Docker. Use a single integrated Visual Studio toolset to build, debug, and deploy apps in locally or Microsoft Azure-hosted containers. Developers also gain multi-project debugging for single and multi-container scenarios.

Docker for Azure. Get started building, assembling, and shipping containerized applications on Azure. The native Azure application provides an integrated, easy-to-deploy Docker environment, optimized to use the underlying Azure IaaS services. The environment is available in Community Edition (CE) for free and in Enterprise Edition (EE) with a subscription.

Docker Enterprise Edition for Azure Marketplace. Use prebuilt cloud templates for Docker EE Basic, Standard, and Advanced to develop and run containerized apps directly in the Azure Cloud. Docker EE delivers efficiency of computing and operations resources through Docker-supported container management and orchestration.

Azure Container Service. Start building, assembling, and shipping applications on Azure—no additional software installation required. This native Azure app provides an integrated, easy-to-deploy environment that uses the underlying Azure IaaS and Docker Enterprise Edition to deploy portable apps. Standard Docker tooling and API support are included.

.NET Core Tools. Enable a seamless experience for Windows, Linux, and Mac OS developers who want to build containerized .NET apps. Tools are optimized for high-scale, high-performance microservices.

Image2Docker. Point this PowerShell module at a virtual hard disk image, scan for common Windows components, and suggest a Dockerfile. The tool supports VHD, VHDK, and WIM, with a conversion tool for VMDK.

² IBM/Ponemon Institute: “2016 Ponemon Cost of Data Breach Study”



Containers offer shared infrastructure and higher density over VMs, much like the difference between condos and houses.

Deliver a win for developers and IT pros

Containers reduce the overall effort required to deploy applications. You can streamline the entire development and test cycle, ultimately reducing costs. Because containers can run on a physical or virtual host operating system, you gain the flexibility to increase server consolidation.

For developers, Windows Server 2016 containers with Docker unlock huge gains in productivity. Build an application, package the app within a container, and deploy the container—knowing that it will run without modification—on-premises, in a service provider’s datacenter, or in the public cloud, using services such as Azure. Distribute multitier apps across Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) models and deliver apps more rapidly than ever. At the same time, IT pros gain even higher levels of consolidation for apps and workloads. Secure and modernize existing enterprise .NET and line-of-business server apps with little or no code changes and a platform that can rapidly scale to meet changing business needs.

Getting started

Ready to learn how Windows Server containers can revitalize your DevOps efforts? Take the next step:

- [Explore Microsoft support for containers](#)
- [Learn more about the Microsoft and Docker partnership](#)
- [Get started with Windows Server 2016 and Docker Enterprise Edition](#)

A true triple threat

Windows Server Containers powered by Docker offer agility, portability, and control.

Agility. The Docker platform provides a 65 percent reduction in developer onboarding time and a 62 percent reduction in mean time to resolution.³ Quickly develop, build, test, and deploy applications—for up to 13x more software releases thanks to a continuous integration and delivery model.³ A standardized environment with a separation of infrastructure and applications makes new deployments, changes, and rollbacks fast and easy and helps deliver a 10x cost reduction in maintaining existing apps.³

Portability. Ship any application from dev to production with full-stack portability to any cloud and infrastructure, with open APIs for a flexible, pluggable architecture. Respondents were able to move 41 percent more workloads across private and public clouds and avoid typical “works on my machine” headaches.³

Control. Set the level of control and flexibility you need to maintain service levels, performance, and regulatory compliance. Securely provision, orchestrate, manage, and deploy container-based applications, at scale.

Modernize apps today

Ready to start containerizing traditional apps? Docker and Microsoft provide plenty of tools and best practices to help. Great choices for starter apps include Microsoft Windows Internet Information Server (IIS) websites and filesystem paths, mid-tier business logic apps, and Apache Web Server, or use the Nerd Dinner test app.

³ State of App Development Survey: Q1 2016, Cornell University case study