Docker Desktop vs DIY with Docker Engine

Developers love using Docker Desktop for all the best reasons: it's easy to use, it accelerates productivity, and it eliminates the toil of setting up complex environments for building container applications.

And while Docker Engine is sometimes viewed as a drop-in DIY (do-it-yourself) replacement for Docker Desktop, it's important to understand that there are vast differences between the two, and that going it alone might not be as smooth - or economical - as it seems. Here are a few key considerations when evaluating any DIY approach:

Is it easy to set up and maintain?

Docker Desktop

With Docker Desktop, installation, configuration, and maintenance are as easy as one click. Starting from the top, Docker Desktop comes as one single package for Mac or Windows. By this, we have a single installer which, in one click, sets up everything you need to use Docker in seconds. Docker simplifies configuration under Docker Desktop, taking care of port mappings, file system concerns, and other default settings, making it seamless to develop on your local machine. Docker also maintains and regularly updates Docker Desktop with bug fixes and security updates.

DIY approach

A DIY approach requires developers to install, configure, patch, and maintain all the components and dependencies on their own.

Is it easy to learn and use?

Docker Desktop

Docker is the #1 most wanted and #2 most loved developer tool according to Stack Overflow's 2021 Developer survey. As such, there is an abundance of training courses available from formal certifications to youtube videos, and everything in between, including extensive documentation, to help you learn and stay up to date with all things Docker. Additionally, Docker Desktop comes with a simple user interface to help you control and manage all settings. The Docker Dashboard lets you easily manage your container images locally and in Docker Hub, manage local running containers, and gives you the ability to manage and explore your Docker volumes and more.

DIY approach

When you take a DIY approach, documentation and training are often scarce and out of date. Additionally, configuring safe and sane default settings for all the components, dependencies, access, and networking is a burden on developers and is managed via the command line.

Is it secure?

Docker Desktop

At the heart of Docker Desktop, we have a lightweight Linux VM that Docker manages for you. As well as setting up this VM, Docker Desktop will keep this VM up to date for you over time by applying kernel patches or other security fixes as are required. This gives you the peace of mind that you don't have another machine image you are managing in your estate and instead Docker will look after this for you. Similarly, for all of the components, Docker's updates keep them all in sync working together and secure with the latest fixes applied automatically for you. This keeps your team in sync, working securely with the same set of tools.

DIY approach

A DIY approach places the burden of keeping all components updated and all vulnerabilities patched on the development team.

Does it scale?

Docker Desktop

Docker Desktop is a proven, reliable, and trusted developer tool and is used in many organizations with hundreds or thousands of developers to build, ship, and run modern cloud-native applications in production, at scale.

DIY approach

DIY alternatives require engineering time to build, configure, and maintain. Instead of spending time on value add activities like developing new features, engineers end up spending time maintaining, updating, and patching the container development environment. DIY alternatives also create unnecessary learning curves for onboarding new developers.

Side-by-side Comparison

		Docker D for Mac/Wi
Cloud-na		
OCI-compatible e	✓	
Build immutable m source code and Dockerfiles	✓	✓
Command line i terface (CLI) for managing container lifecycle	✓	✓
Integrated Kubernetes runtime and Kubernetes load balancer	✗	✓
Integrated Docker Compose 2.0	✗	✓
Deploy images from desktop to AWS ECS or Azure ACI	✗	✓
Integrated Linux VM		
Supports integration with any Microsoft WSL2 Linux distro	✗	✓
Integrated with new Apple hypervisor	✗	✓
Built-in control of local host system resources	✗	✓
Security		
Automated security patches	✗	✓
Integrated container image vulnerability scanning	✗	✓
Local file system access controls with secure defaults	✗	✓
One-click installation, configuration, & maintenance updates		
All dev tools - Docker Engine, Kubernetes, CLI, Build, Compose...	✗	✓
Consistent across Mac & Windows	✗	✓
Local host network port mapping from VM to host	✗	✓
VPN integration for interoperating with remote IPs	✗	✓
Bind mount files into host VM	✗	✓
Connect to services on the localhost of the host machine	✗	✓

Consider the Total Cost of Ownership (TCO)

According to a recent Gartner note: "If you are looking at alternative solutions, you must include the opportunity cost of using this solution for your engineering resources. For example, a 100-seat annual subscription to Docker Business without any discounts is currently $25,200. Supporting 100 seats with an open-source alternative is likely to significantly exceed this cost due to the level of engineering resources required to maintain the solution. If you decide to pursue open-source alternatives, you must ensure doing so is a worthwhile use of your engineering resources."

Asking The Right Questions

Making technology decisions that could impact developers across an entire organization is never an easy task, but it's important to ask the right questions upfront. Docker Desktop delivers a valuable, reliable, and secure developer experience. Any alternative DIY approach should offer a solid YES to all the following questions, or it's simply not comparable with Docker Desktop:

- Does it have the tools my developers want to use?
- Does it support the architectures I need?
- Does it provide the instant feedback loop for changes when my developers refresh the page?
- Does it have a long-term roadmap and a future as a product/ commitment to support?
- Is the product growing/will it keep adding new features to make my developers more productive?
- Does it come with sane defaults/keep me secure by default?
- Am I able to keep my developers consistent on all the platforms they use?
- Does it have support?

Learn more about the magic behind Docker Desktop

docker.com/blog/the-magic-behind-the-scenes-of-docker-desktop

Learn More

Get Started With Docker Desktop Today

docker.com/products/docker-desktop

Download Now