Docker Desktop vs Docker Machine

Docker

I've been reading about **Docker for Mac** recently, and realized that there was something I never quite understood – how does **Docker** run on the recent macOS versions without **VirtualBox** or Parallels that I remember it required just a few years ago.

Docker Machine

This is the original implementation of Docker for macOS and Wndows. You have a special VM image (Docker Machine) that needs a third party virtualization solution like VirtualBox or Parallels to run.

VM image acts as a Docker host – it's a minimal Linux distro optimised for minimal footprint and best performance. You also have a set of command line utilities that you can run from your native OS – macOS, for instance – that talk to the Docker Machine for starting/stopping or otherwise managing Docker containers.

The way I understand it, you must expand RAM and vCPU allocation for the Docker Machine manually to provide more compute and memory resources for your Docker containers.

Docker Desktop

Docker Desktop is a different kind of solution. It doesn't need a third party virtualization because it depends on native OS virtualization methods. Will be interesting to research this for Windows, but for macOS these technologies are called Hypervisor Framework.

Docker Desktop spins up a kind of VM using HyperKit, which is an open-source solution for embedding Hypervisor virtualization into your app.

You don't need to manage this VM and have a nice interface for controlling how many virtual CPUs and RAM is allocated to that Docker instance.

There's less overhead when it comes to managing Docker VM, but some functionality is limited by the OS implementation.

That's it for today! Will share more if/when I'm researching this topic again.

See Also

- Docker
- How To Tag Docker Images
- Migrating Docker Containers
- List All Docker Contianers
- Install Docker on Mac