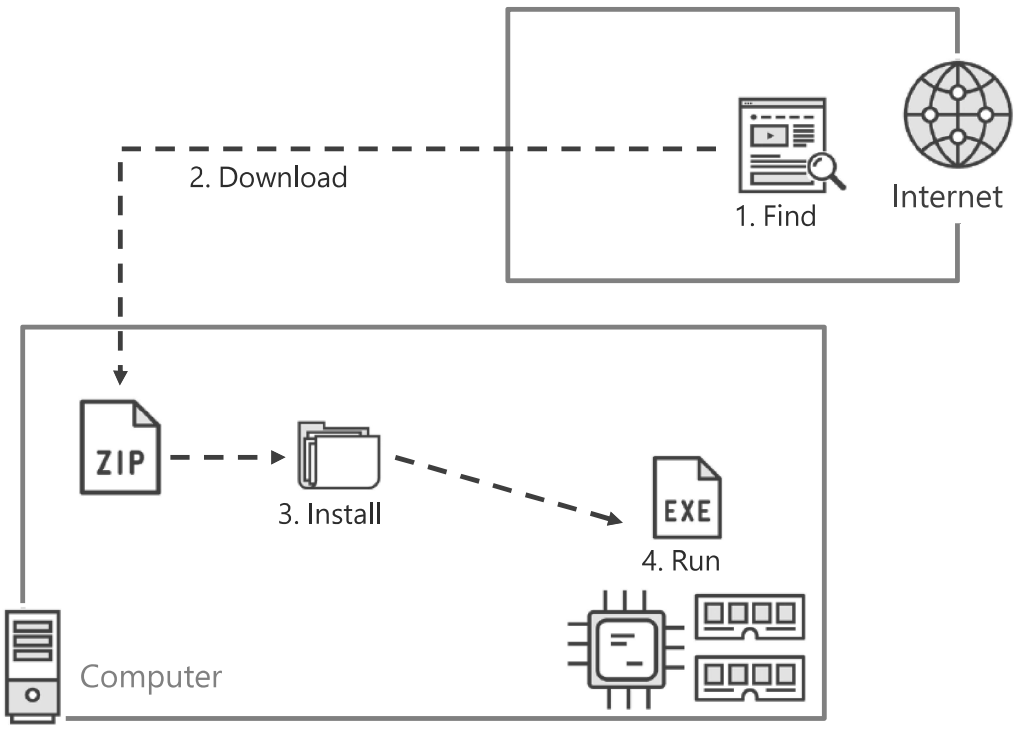
Getting Started with Docker

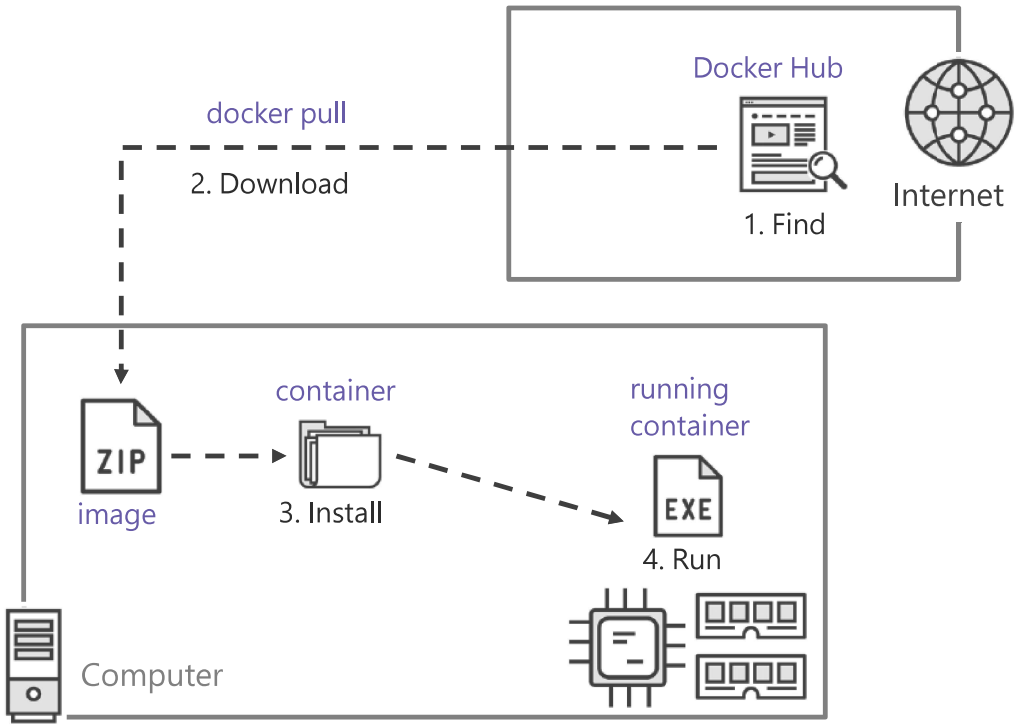
WHAT IS A CONTAINER?

Docker simplifies software

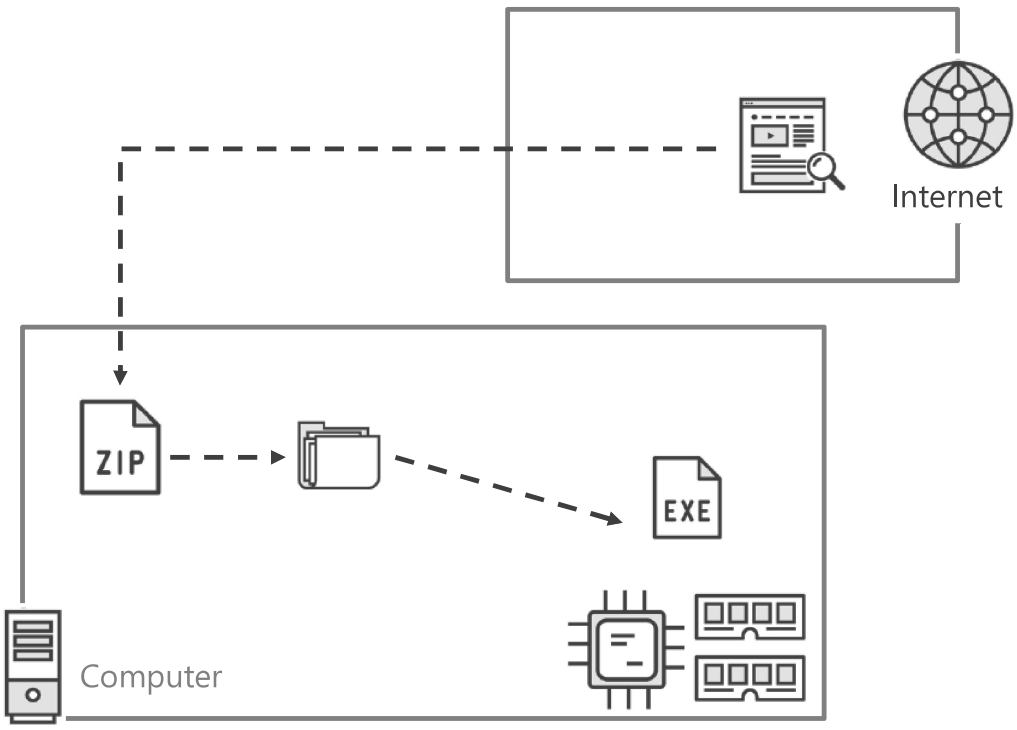
If you already know how to use software, then you already understand Docker



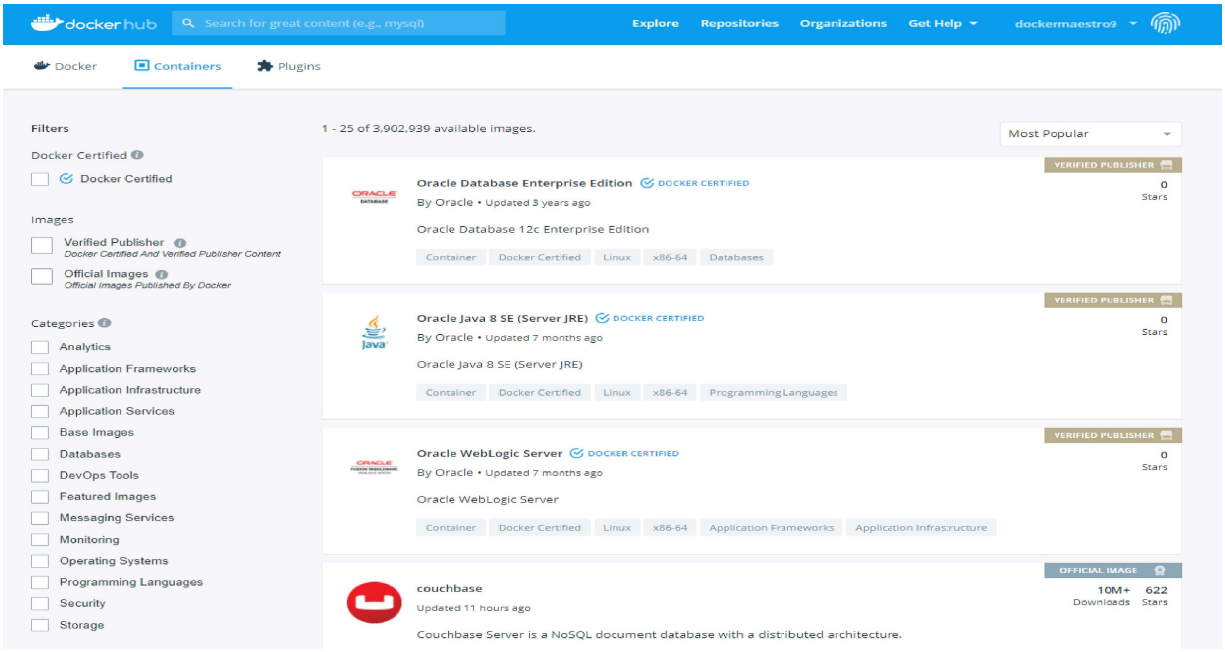
docker run
does all of this!



1. Find
2. Download
3. Install
4. Run



Docker Hub



Traditional Software	Docker Equivalent	Docker Command
Find Software	Docker Hub	
Download software, i.e. a zip file or MSI	Pull an image	docker pull
Install software	Create a container from an image	docker create
Start software	Run the container	docker start
Stop software	Stop the container	docker stop
Uninstall software	Remove the container	docker rm
Not Possible	Do all of this with one command!	docker run

Inverted learning

Use software without knowing how to set it up.

When ready, everything is consistently documented for you to learn how to set it up (Dockerfile).

Challenges with Software Discovery

Where?

- App store
- Package manager
- Standalone website

Trust

- http
- Code itself

Download availability

Challenges with Software Installation

Compatible? Cross-Platform?
- OS, OS version/build, CPU arch
- macOS Sierra 10.12 x64
- Win10 Anniversary Edition x64

Format
- Source
- Standalone executable
- Executable + bundled libraries
- Executable + shared libraries / runtime

Installers, package managers, manual…
- What did it install?!

Updates (auto updaters) & uninstall

Configuration Management

Challenges with Running Software

Helpful documentation

Where is it? PATH?

Starting and stopping

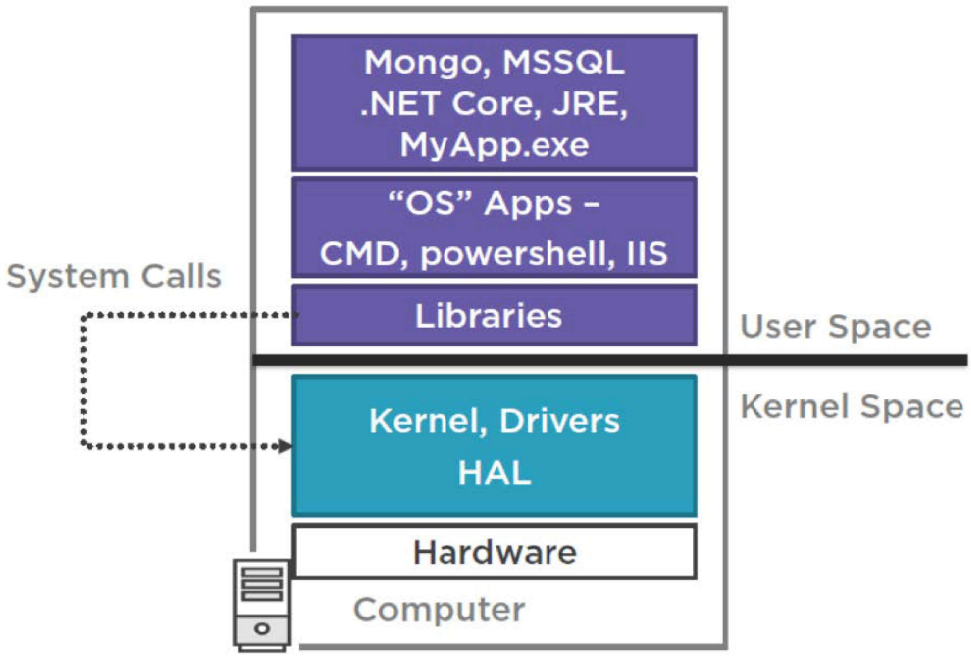
Service registration

Installing and running dependencies

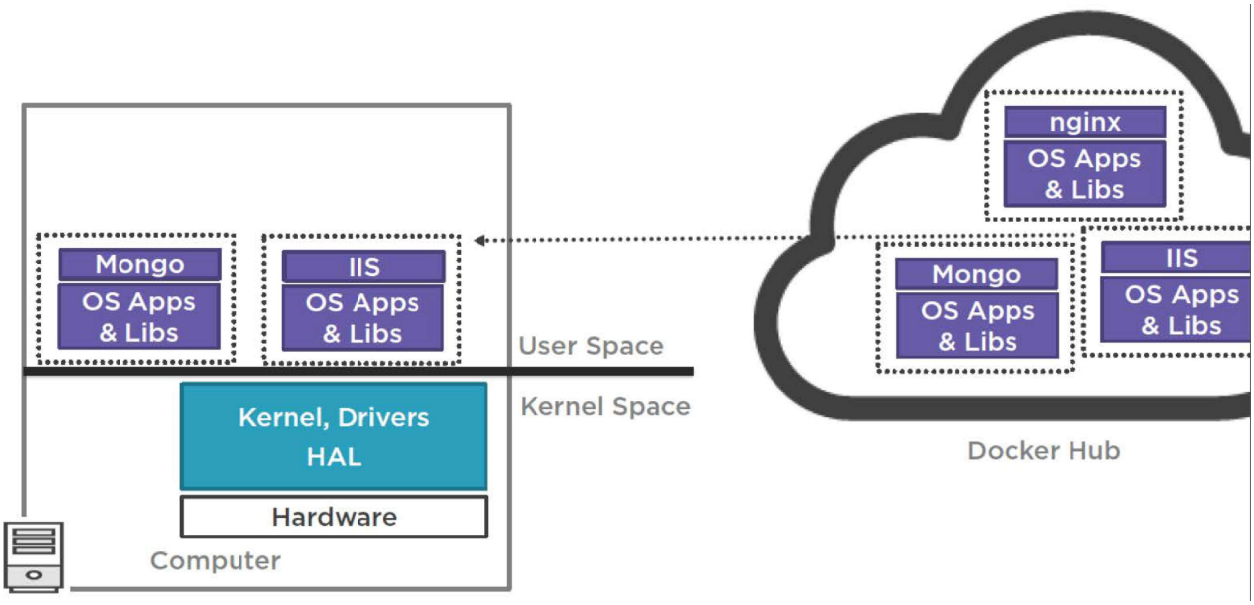
Security & sandboxing

Breaking changes
- OS updates
- Shared library updates

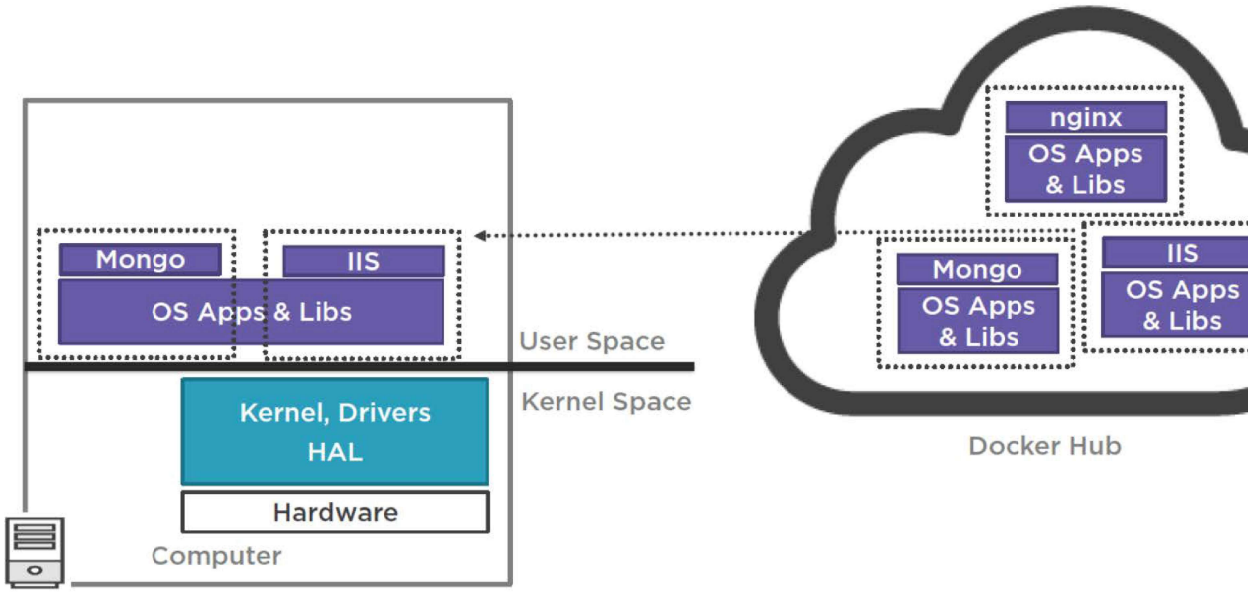
Traditional Software Layer



Container Software Layer



Layer Reuse



Containers are lightning fast compared to VMs