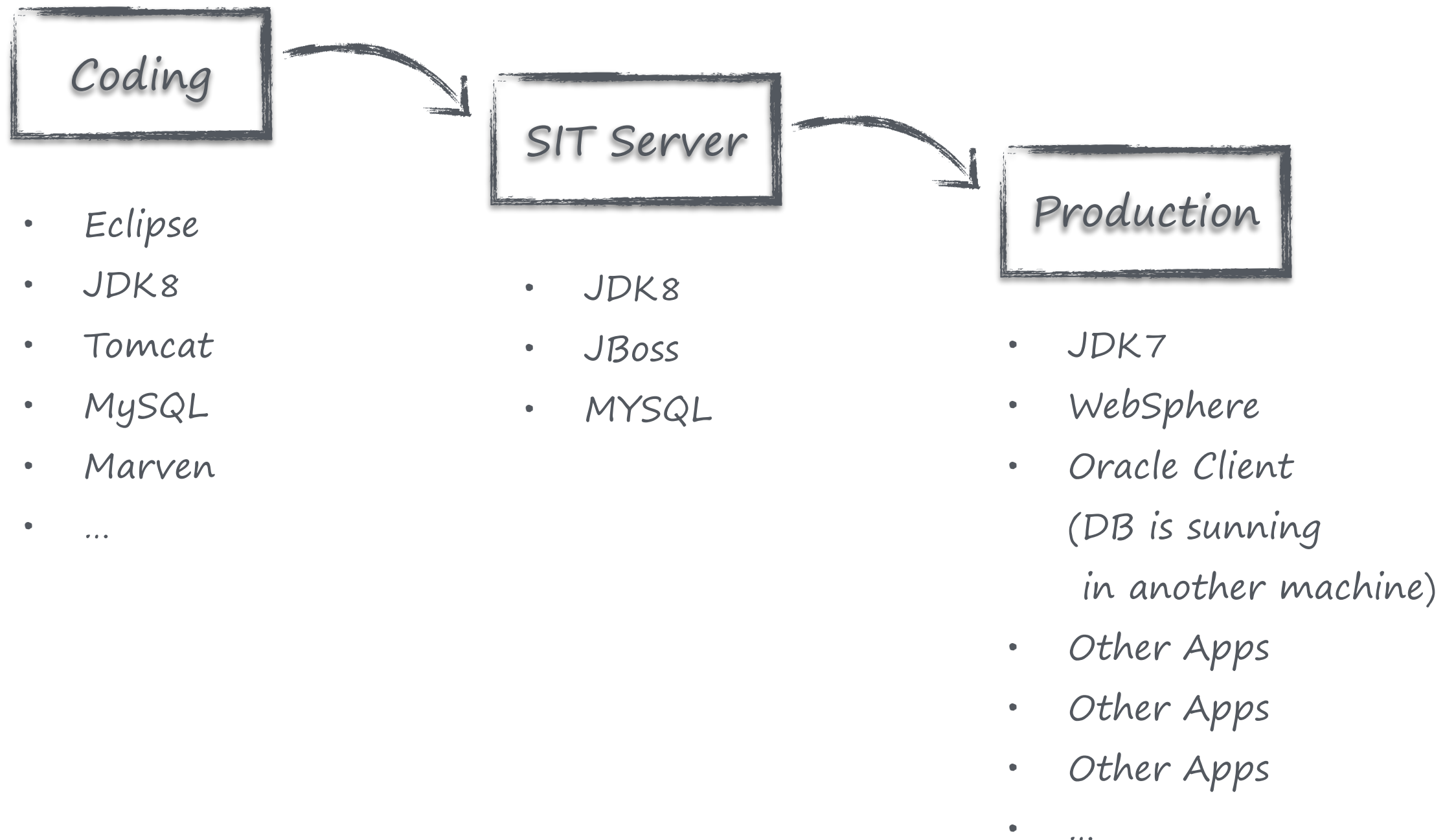


Introduction to Container & Docker

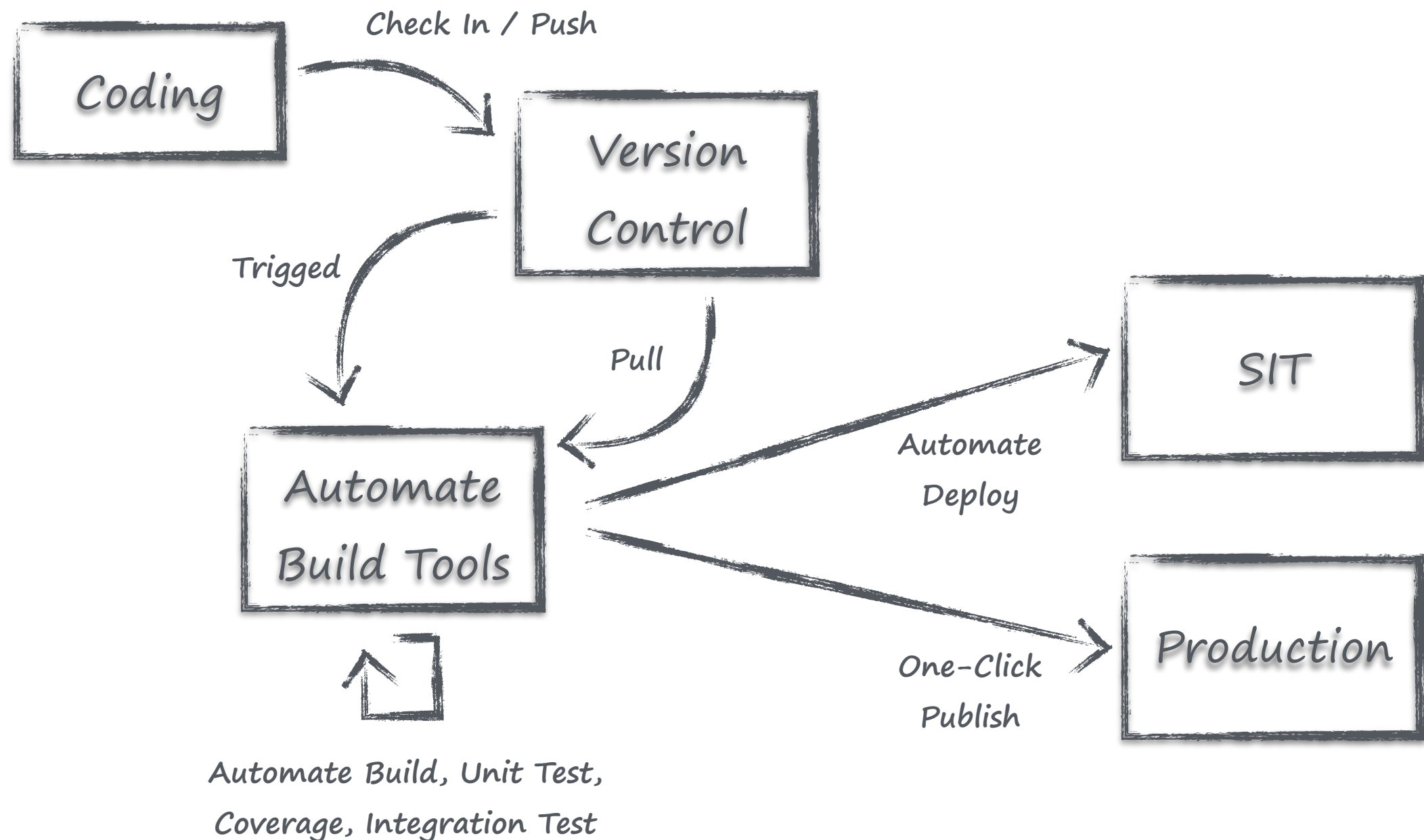
Container

Containers are a lightweight and portable store for an **application** and its **dependencies**.

Software Development



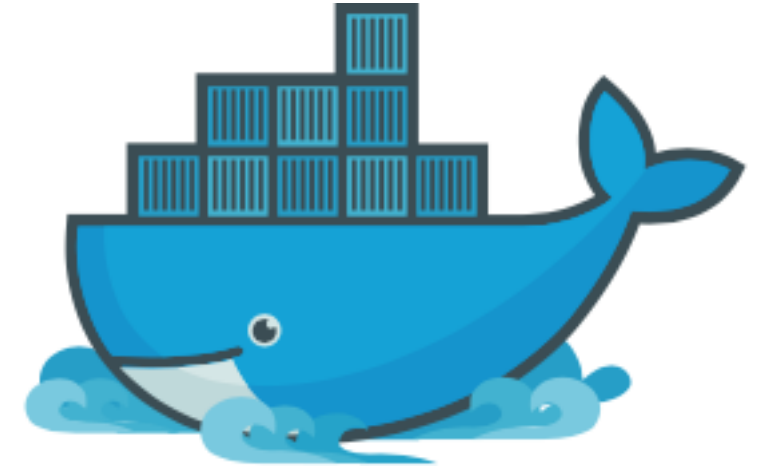
Automate CI / CD Environment



Virtualization

- Install Everything, from OS to application libraries
- My app not requires graphic drivers, printer drivers, Windows Services, COM+, etc... I just want to run my app.
- Prepare by operation. Most developer has no permission to do it.
- Solve issues by emails, screen captures, ... Lack of communication.
- On production server. Capacity is always full, but runtime usage only few.

Docker



- Docker is the original author and primary sponsor of the Docker open source project.
- Together with the community of maintainers and contributors, Docker aims to deliver open tools to help developers build applications with open APIs to help sysadmins better manage these applications.
- Services :- Docker Cloud, Docker Datacenter

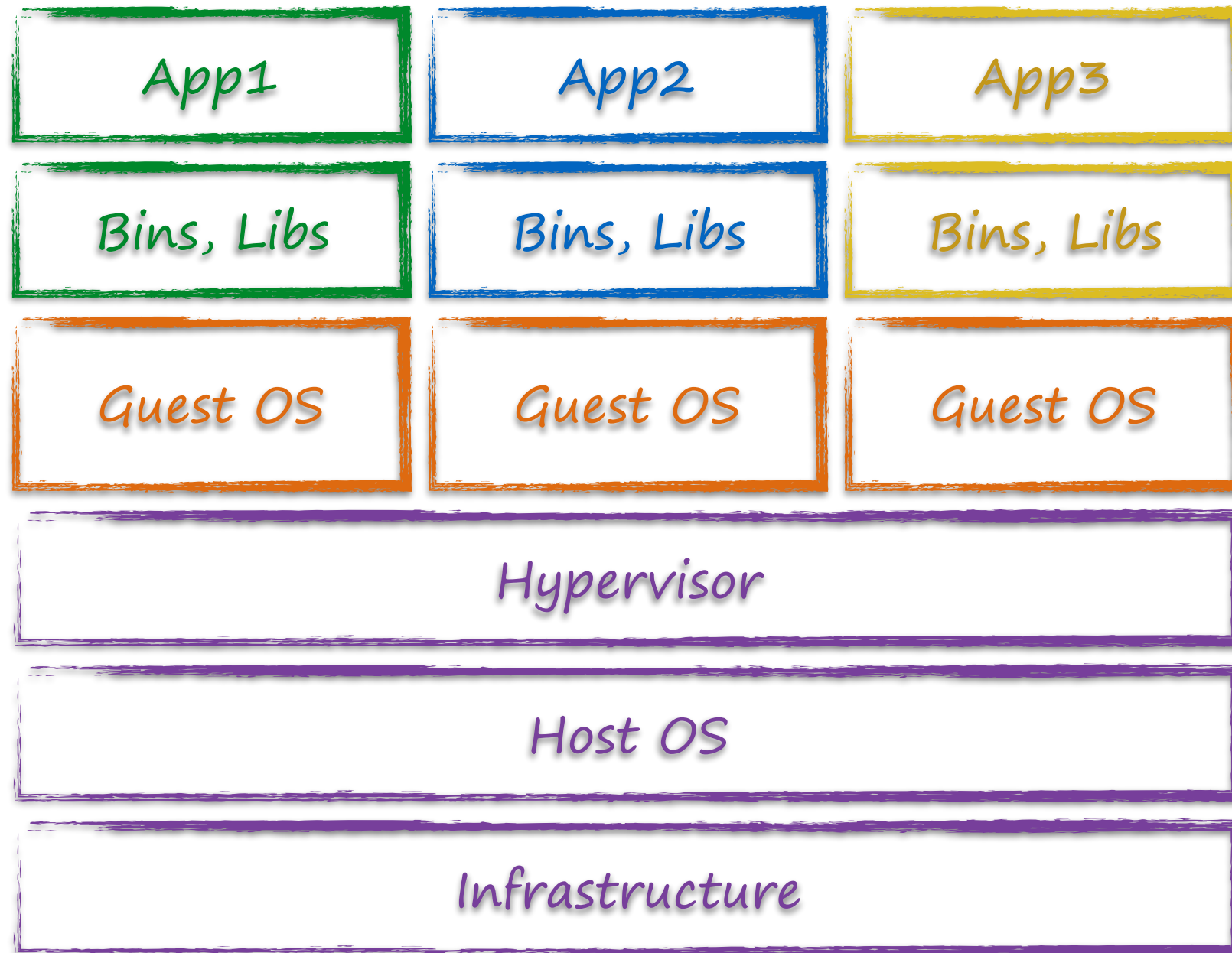
Container

- The docker project offers higher-level tools which work together, **built on top of some Linux kernel** features.
- Docker achieves this by creating safe, **LXC** (i.e. Linux Containers) based environments for applications called “**docker containers**”.
- Encapsulation of an application with its dependencies.
- At first glance, its appear to be just a lightweight form of VM. However, containers have several advantages.
- Container holds an isolated instance of an OS, which we can use to run applications.
- Containers are fundamentally changing the way we develop, distribute, and run software.

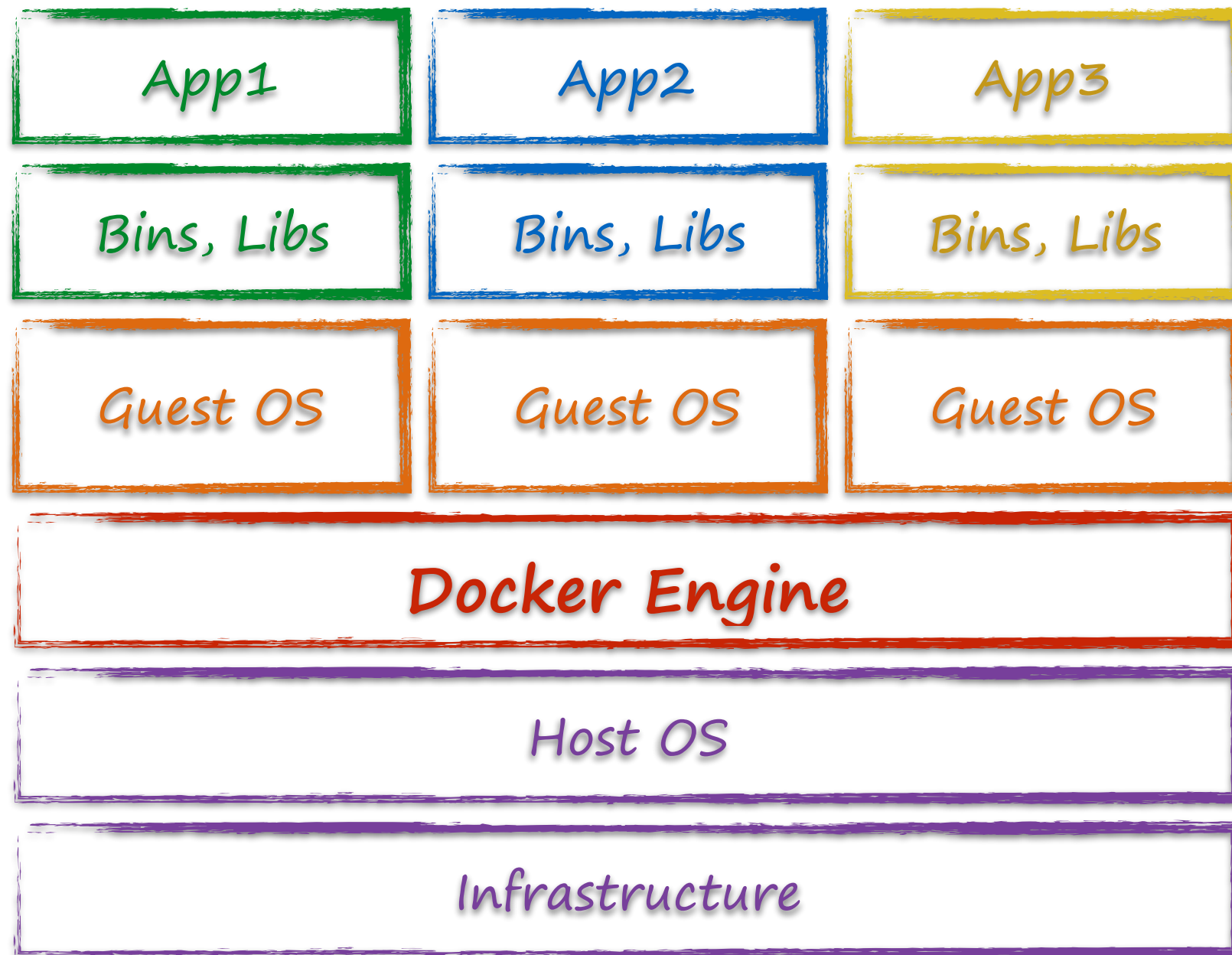
Container Benefits

- **Share resources** with **host OS**. That makes its an order more effective.
- **Start and Stop in a second**, incur **little** to no **overhead** compare to application running natively on the host OS.
- Containers has the potential to eliminate bugs cause by changed in the runtime environment. **“But it works on my machine”** stuff should be gone.
- Developers and operation can run dozens of containers at the same time. Making it **possible to emulate a production-ready** distributed system.
- Developers and operation can **avoid differences in user environments** and the availabilities of **dependencies**.
- Finally, you don't need to run container all the time to execute your tasks. Start-execute-stop in seconds, with large scale of worker nodes scenario, is possible.

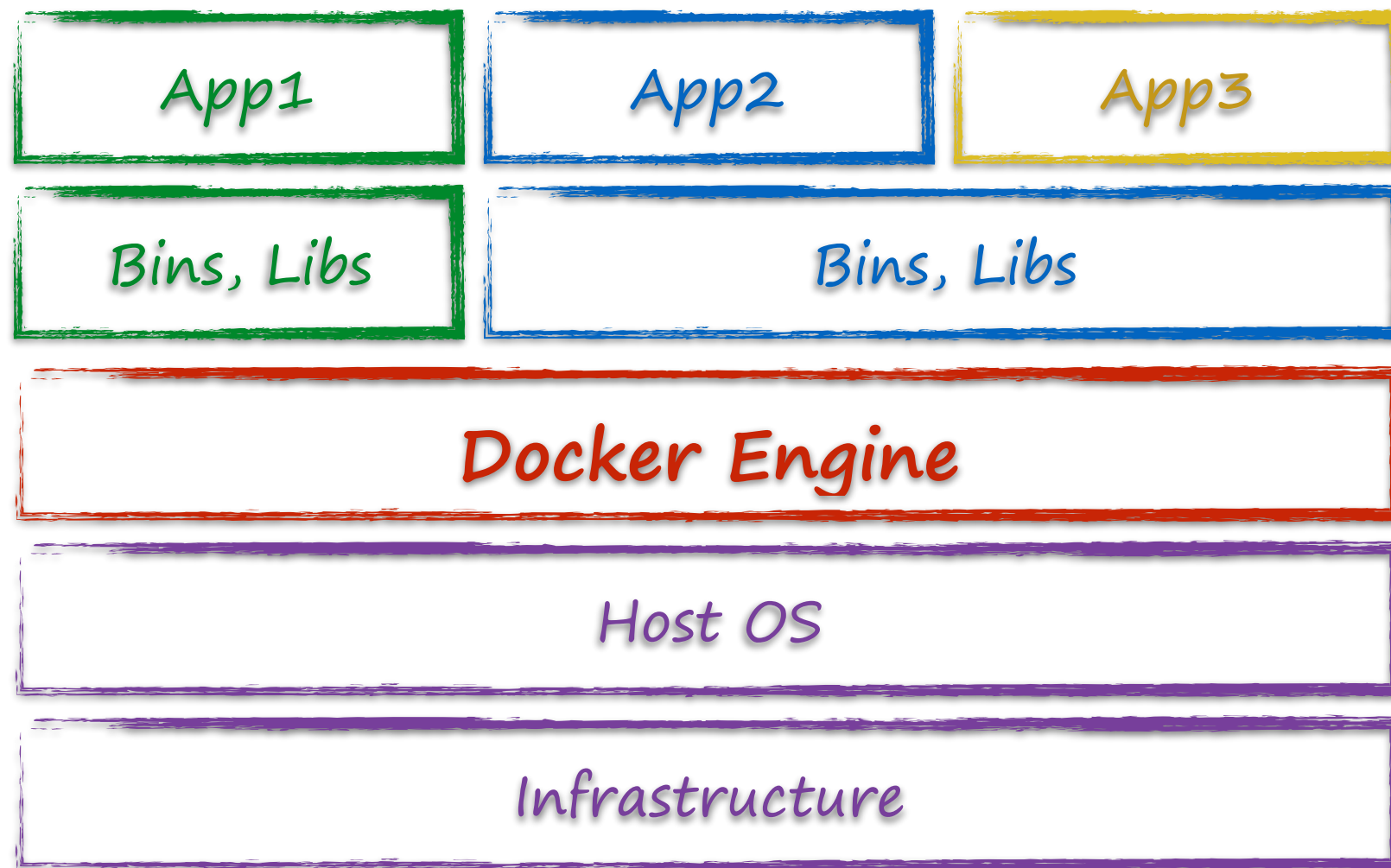
Virtual Machine



Container



Container



Docker Components



Docker Engine :-

Create Docker images and run Docker containers.
Including **Swarm** mode from 1.12.0-rc1



Docker Compose :-

Defines applications built using multiple containers.



Docker Hub :-

A hosted registry service for managing and building images.



Docker Cloud :-

A hosted service for building, testing, and deploying Docker images to your hosts.



Docker Trusted Registry :-

(DTR) stores and signs your images.



Docker Universal Control Panel :-

(UCP) Manage a cluster of on-premises Docker hosts as if they were a single machines.



Docker Machine :-

Automate container provisioning on your network or in the cloud. Available for Windows, Mac OS X, or Linux

Platforms Supported



Docker for Mac

A native application using the OS X sandbox security model which delivers all Docker tools to your Mac.

- OS X 10.10.3 Yosemite or newer
- Mac 2010 or newer
- Intel's hardware support for MMU virtualization
- At least 4GB Memory
- VirtualBox prior to version 4.3.30 must NOT be installed



Docker for Windows

A native Windows application which delivers all Docker tools to your Windows computer.

- 64bit Windows 10 Pro, Enterprise or Education (Build 10586 or later)
- Support Microsoft Hypervisor
- VirtualBox will no longer work



Docker for Linux

Install Docker on a computer which already has a Linux distribution installed.

- Arch Linux, CentOS, CRUX, Debian, Fedora, FrugalWare, Gentoo, Oracle Linux, RHEL, openSUSE / SUSE Ent., Ubuntu
- 64bit
- Kernel version 3.10 or later

Lab Resources

- 3 Laptops per team.
4GB Memory, 8GB recommend. 2.5GB - 4GB free memory remains after boot.
16GB free storage space.
- VMWare Workstation or VMWare Player.
<https://goo.gl/rZTnVn>
- Visual Studio Code with Dockerfile plugins
<https://code.visualstudio.com/download>
Atom or Sublime should be ok.
- Ubuntu 16.04.1 LTS VMWare Guest Image (1.13GB)
download: <http://download.somewhere.com/img.zip>
- Handouts
download: <http://download.somewhere.com/handouts>

References

- Docker Official Website :-
<https://www.docker.com>
- Documents :- <https://docs.docker.com>
Overview :- <https://goo.gl/vnVJGT>
Getting Start :- <https://goo.gl/pbfSjq>
Learn by Example :- <https://goo.gl/F5ZTQS>
- Github :-
<https://github.com/docker/docker>
<https://github.com/docker>
- Some Interesting Books :-
Using Docker :- <http://amzn.to/2aYflh5>
Docker Up & Running :- <http://amzn.to/2aYeBIC>