INTRODUCTION TO

# CONTAINERS

## OVERVIEW

1. History of Virtualization
2. Vagrant Overview
3. History of Containers
4. Docker Overview and Tutorial
5. Next Steps/Additional Resources

# THE MAINFRAME

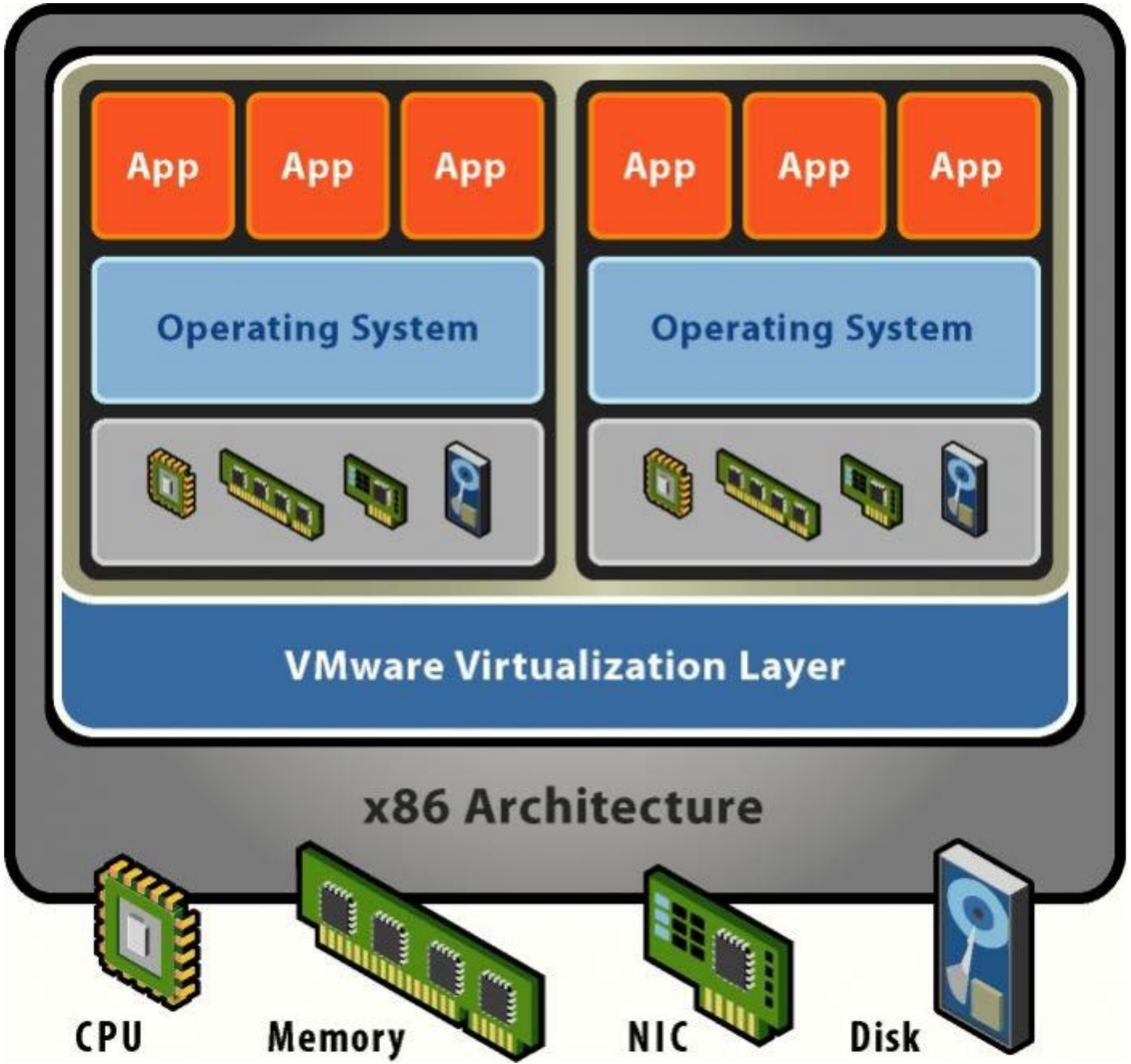
# HISTORY OF VIRTUALIZATION

# THE DATACENTER

# HISTORY OF VIRTUALIZATION

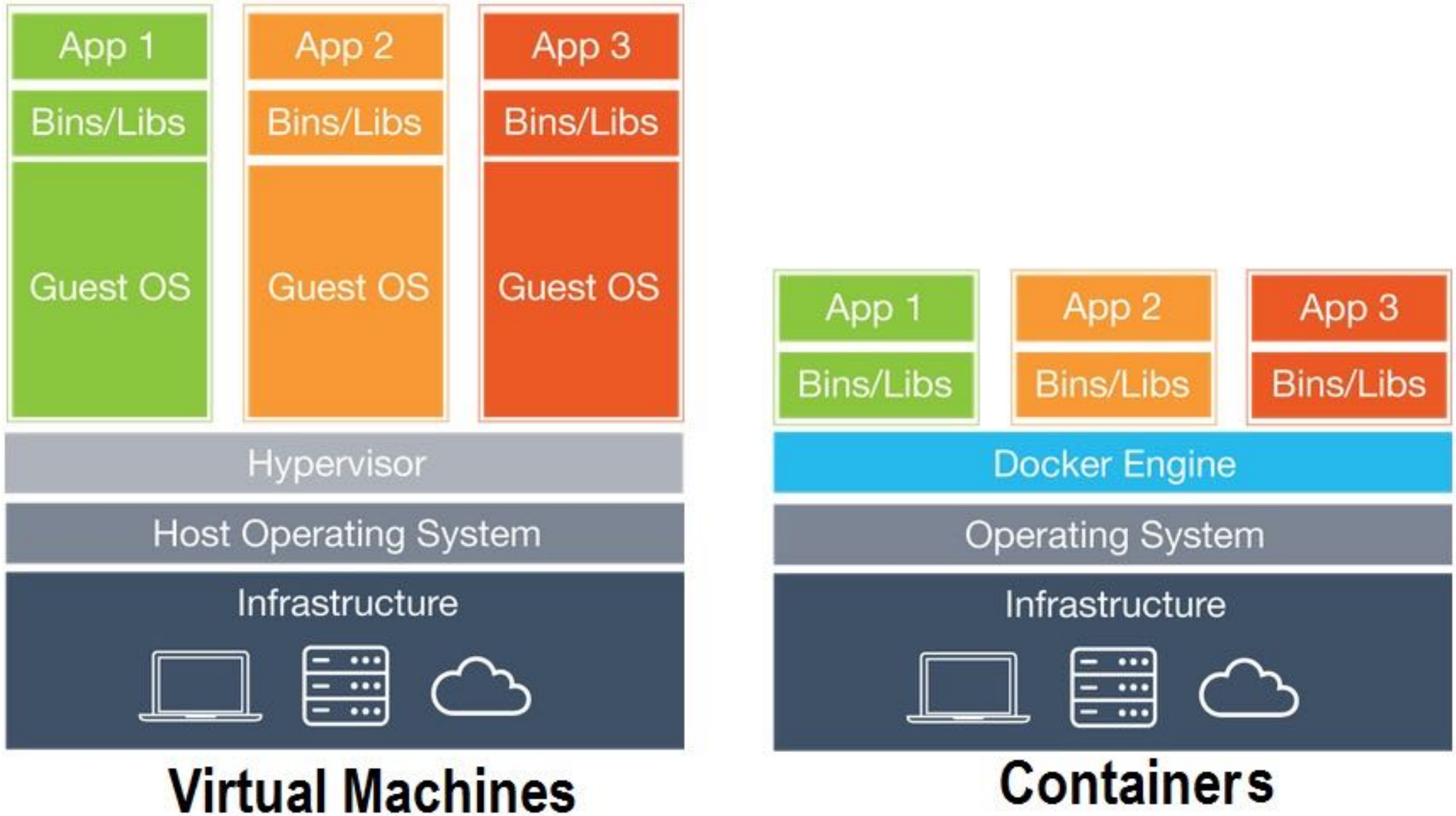
# VIRTUAL MACHINES

# HISTORY OF VIRTUALIZATION



# CONTAINERS

# HISTORY OF VIRTUALIZATION



# VAGRANT

- Build and manage virtual machines
- Isolate dependencies and configuration
- Disposable environments
- Consistency
- https://www.vagrantup.com/downloads.html
- Requires a “provider” (e.g. VMware, Virtualbox)

# IT IS ALL BECAUSE OF LINUX

- Linux-VServer
- Generic process containers (a.k.a control groups a.k.a. cgroups)
- Namespaces (user)
- LXC
- Docker
- Orchestration

# BUT WHAT ABOUT WINDOWS?

## SORCERY

# DOCKER

- Software Container Platform
- Community Edition and Enterprise Edition
- Toolchain

# KEY TERMINOLOGY

## IMAGE

A lightweight, stand-alone, executable package that includes everything needed to run a piece of software, including the code, a runtime, libraries, environment variables, and config files

# KEY TERMINOLOGY

## CONTAINER

A runtime instance of an image – what the image becomes in memory when actually executed

# TUTORIAL

https://docs.docker.com/get-started/

# NEXT STEPS

A. Finish the official Docker tutorial

B. Dive deeper into Docker via additional docs

    A. https://docs.docker.com/engine/userguide/

    B. https://docs.docker.com/engine/reference/commandline/cli/

C. Try Docker Codeschool course (https://www.codeschool.com/courses/try-docker)

D. Scalable Microservices with Kubernetes (https://www.udacity.com/course/scalable-microservices-with-kubernetes--ud615)