Docker: All the Cool Things

Introduction

- **Docker** is an open platform for developing, shipping, and running applications.
- **Docker** enables you to separate your applications from your infrastructure so you can deliver software quickly.
- With **Docker**, you can manage your infrastructure in the same ways you manage your applications.

Why learn it?

- It’s cool.
  - Distributing working software (especially complex software) is difficult, because there’s tons of factors involved.
    - Hardware architecture, software packages (including versions), etc.

- You understand how your application works outside of the application logic.
  - What does my application need to work?
  - How can I upgrade my application in the future?
    - Will that be easy?
    - How much will it cost? (in development time + effort)

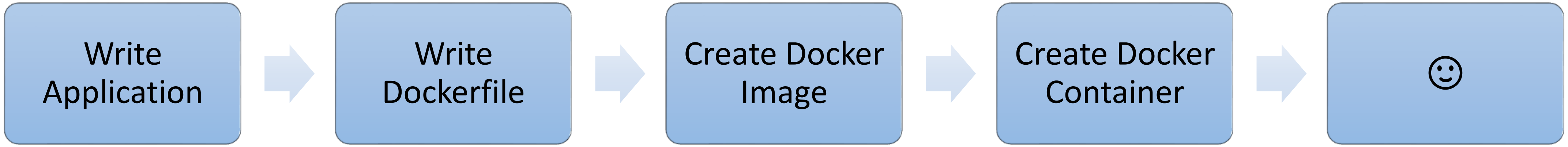
- It’s a bit future-proof.
  - Virtualization allows the environment to be emulated; hardware changes are less important.

Major Details

- Docker containers aren’t the same thing as a virtual machine.
    - Docker runs as a process on your machine (or someone else’s) containing the application code within the image.
    - A virtual machine emulates an entire operating system on top of your host OS.
- Overall, they’re similar concepts and tools similar to Docker are available for distributing virtual machine images with application code.
    - Vagrant by HashiCorp is cool, but that’s not what we’re learning today.

Major Details: Part 2

- We’ll take an application’s source code, write a Dockerfile, build a Docker image, and create a Docker Container.
  - A Docker **image** is a template that has instructions for creating a **container**.
  - A **Dockerfile** will define how we package our application into an image.
    - We share the image with other developers, not the container itself.



Write Application

Write Dockerfile

Create Docker Image

Create Docker Container

☺

Demo Time!

We’ll be running through an example available here:

https://github.com/mcdonagj/hoohacks-presentation-2021/tree/master

Questions?

Hope you enjoyed the talk!