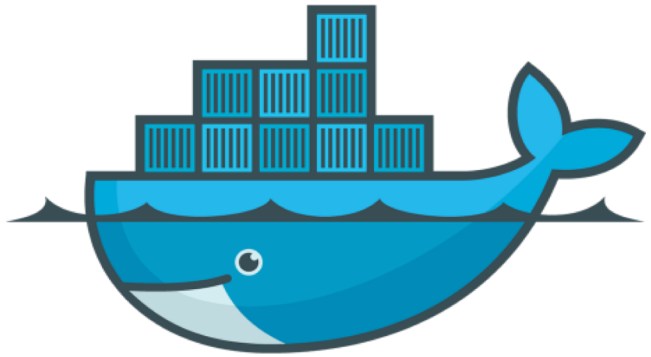


Docker JumpStart

Getting Started with Docker

Mike Pfeiffer

Blog

https://mikepfeiffer.io

Twitter

@Mike_Pfeiffer

Mike Pfeiffer is a twenty year IT industry veteran, published author, and international conference speaker. He's a former architect for Amazon Web Services and engineer for Microsoft. Today Pfeiffer serves as Chief Technologist at CloudSkills.io.

Dan Wahlin

Blog

https://blog.codewithdan.com

Twitter

@DanWahlin

Dan Wahlin founded Wahlin Consulting (https://codewithdan.com) which provides training and architecture services on front-end and back-end web technologies, Microservices, and Docker/Kubernetes. He's published multiple courses on Pluralsight.com and is a Docker Captain, Microsoft MVP/Regional Director, and Google GDE. Dan speaks at multiple conferences and runs the Code with Dan development newsletter.

Workshop Logistics

- 9 am – 4 pm PST
- Lunch from 12 – 1 pm PST (with breaks in the morning/afternoon)
- Recordings will be available by June 28th
- Please mute your microphone
- Ask questions using the chat
- For more details questions/answers we'll call on you to unmute your mic
- Hands-on labs

Pre-Reqs

Prior experience with one of the following:

- Command line
- Systems administration
- Software development

Course Expectations

- Learn Docker from the ground up
- Learn how to work with images and containers
- Learn different container orchestration techniques
- Learning is not effective unless you’re relaxed and having fun. Have fun and enjoy it! ☺

Workshop Content

Day 1

- Getting Started with Docker
- Setting Up Your Environment
- Building and Managing Docker Images
- Running and Managing Docker Containers

Workshop Content

Day 2

- Mastering Container Volumes
- Container Networking Demystified
- Docker Compose Orchestration In-Depth
- Container Orchestration with Kubernetes

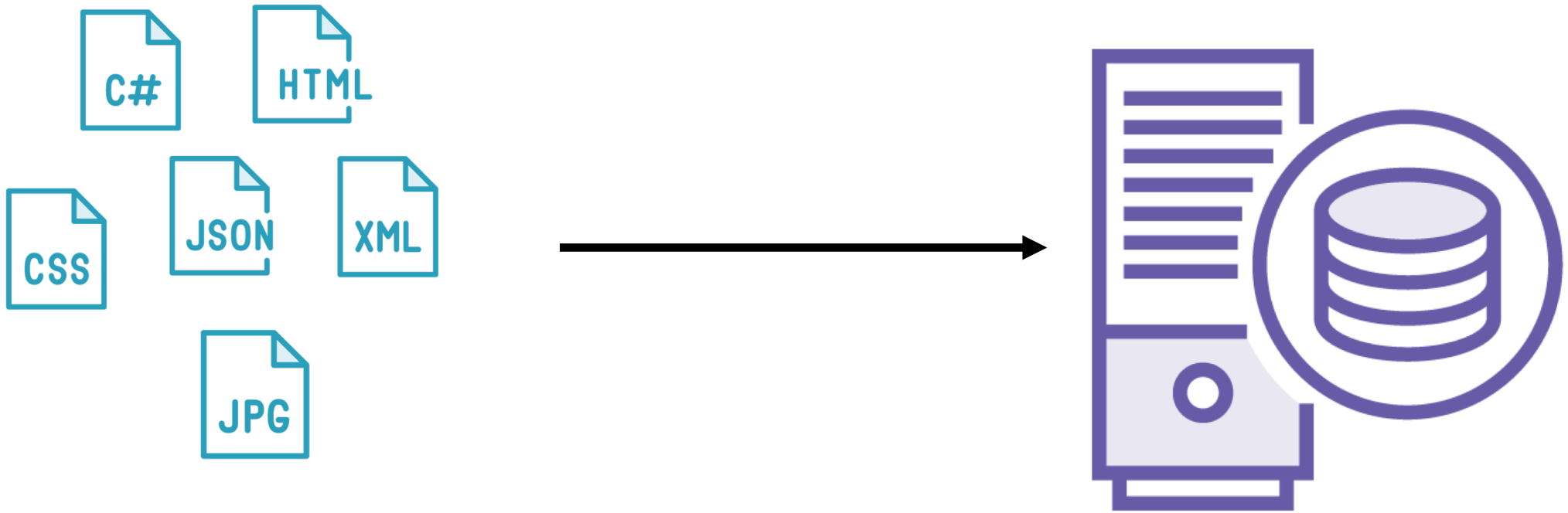
Agenda

- Why Docker?
- Getting Started with Docker
- Benefits to Developers and DevOps
- Containers vs. Virtual Machines
- Docker and Microservices

Why Docker?

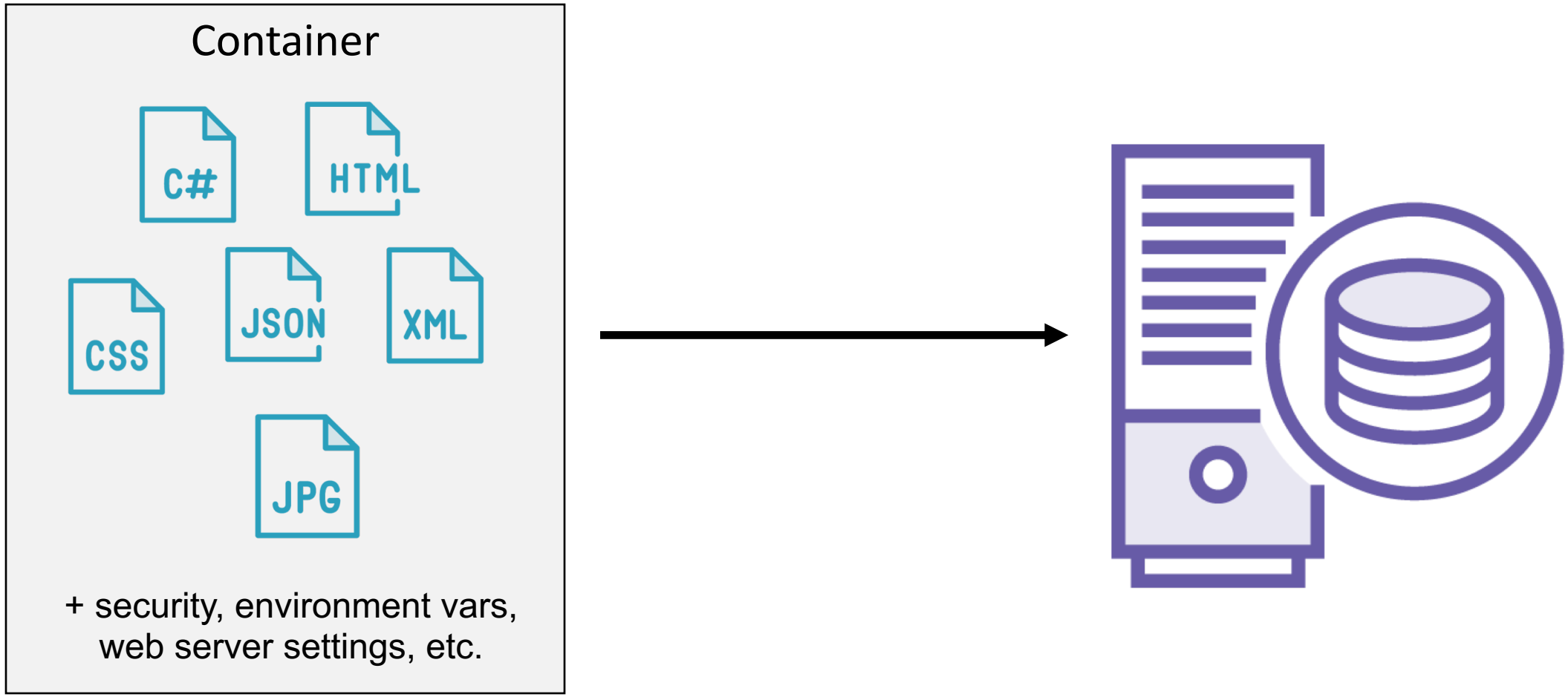

Shipping Applications the Traditional Way

Shipping Applications



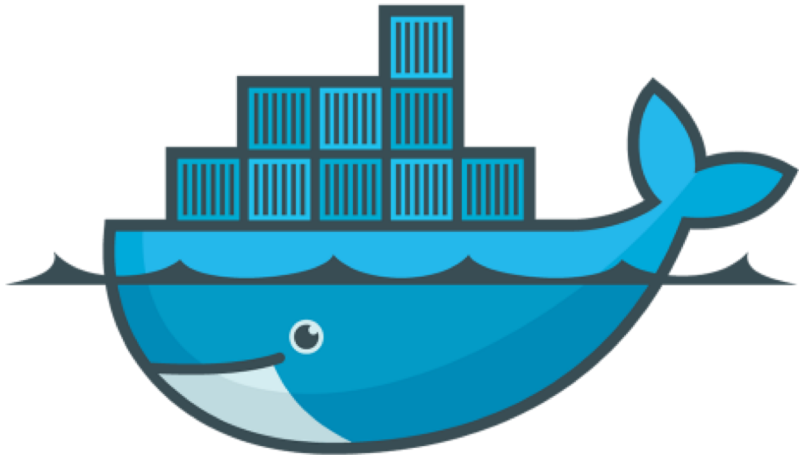
Shipping Applications with Containers

Shipping Applications using Containers



Getting Started with Docker

What is Docker?



Simplify building, shipping, running apps

Shipping container system for code

Runs natively on Linux or Windows Server 2016+

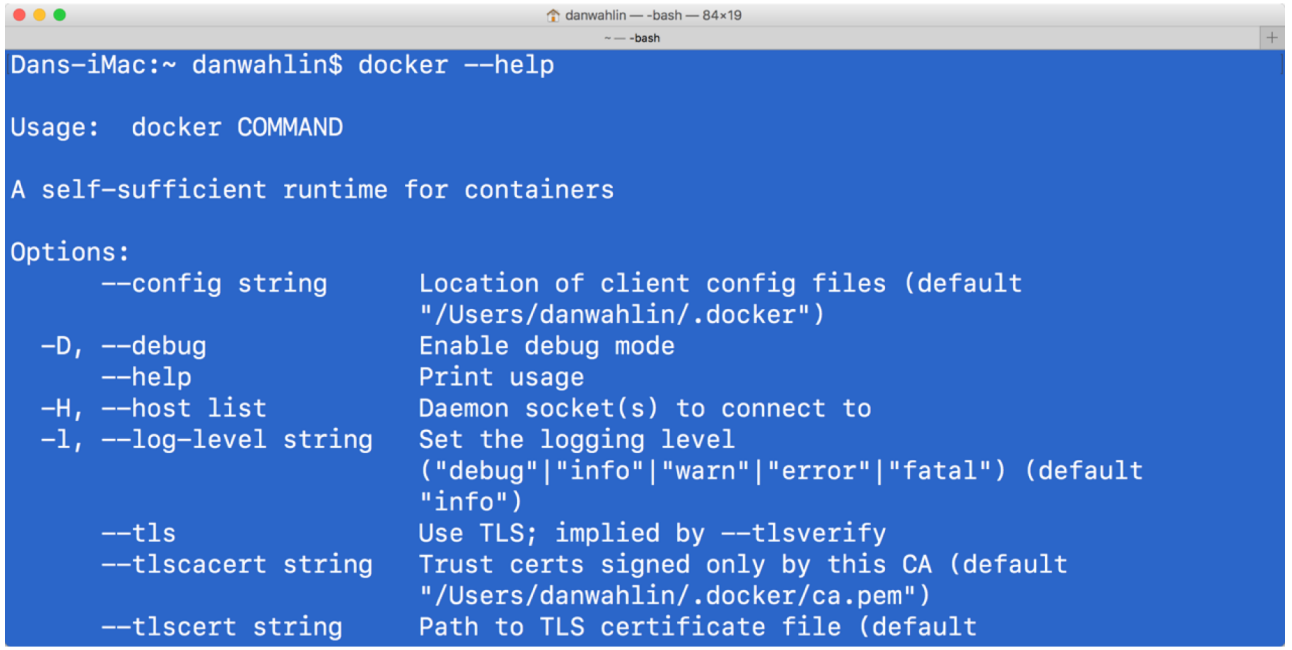
Runs on Windows or Mac Development machines (with a virtual machine)

Relies on images and containers

Docker Client CLI

Help switch used to list all Docker CLI commands

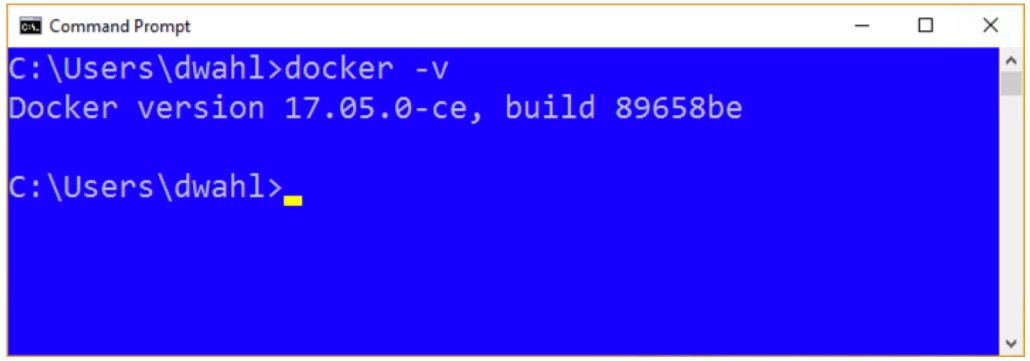
docker --help



Docker Client Version

Get Docker client CLI version

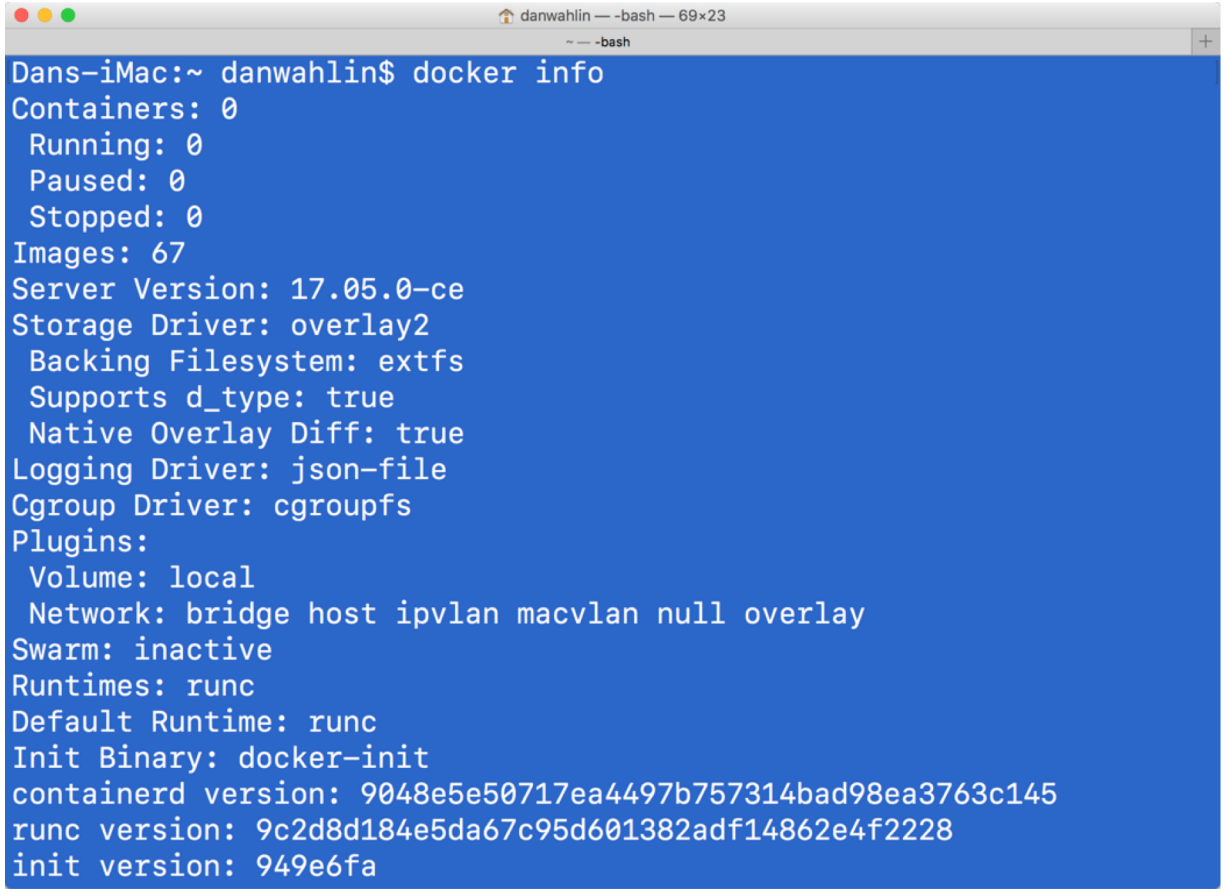
```
docker -v
```



Docker Client Information

Get information about host, images, containers and more

`docker info`

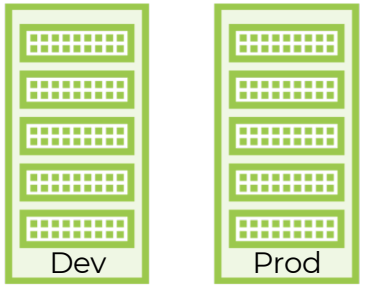


Benefits to Developers and DevOps

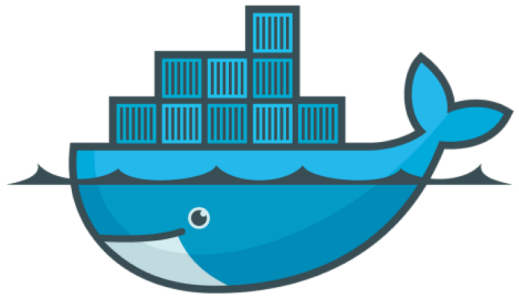
Docker Benefits

Accelerate Developer Onboarding

Eliminate App Conflicts

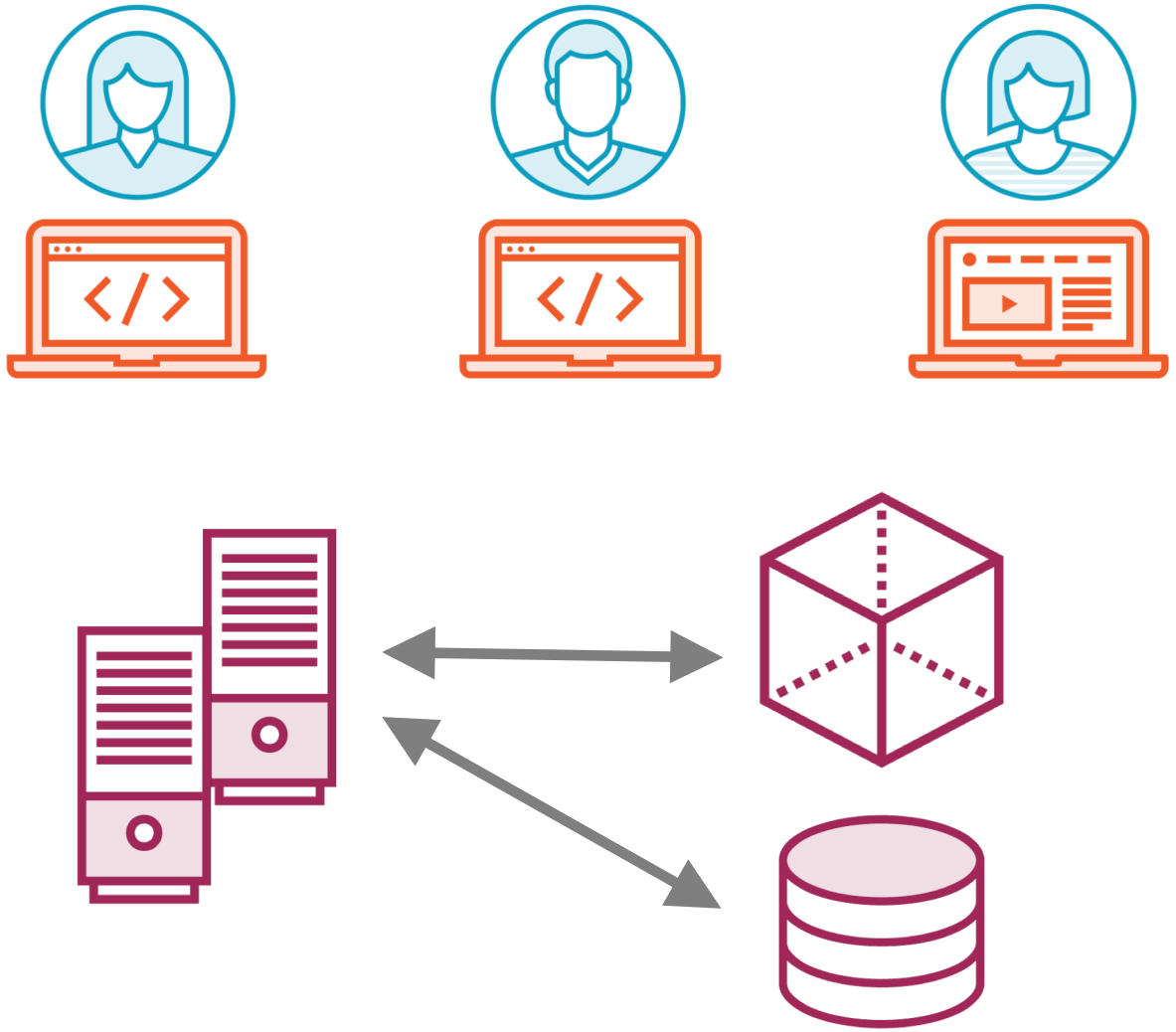


Environment Consistency

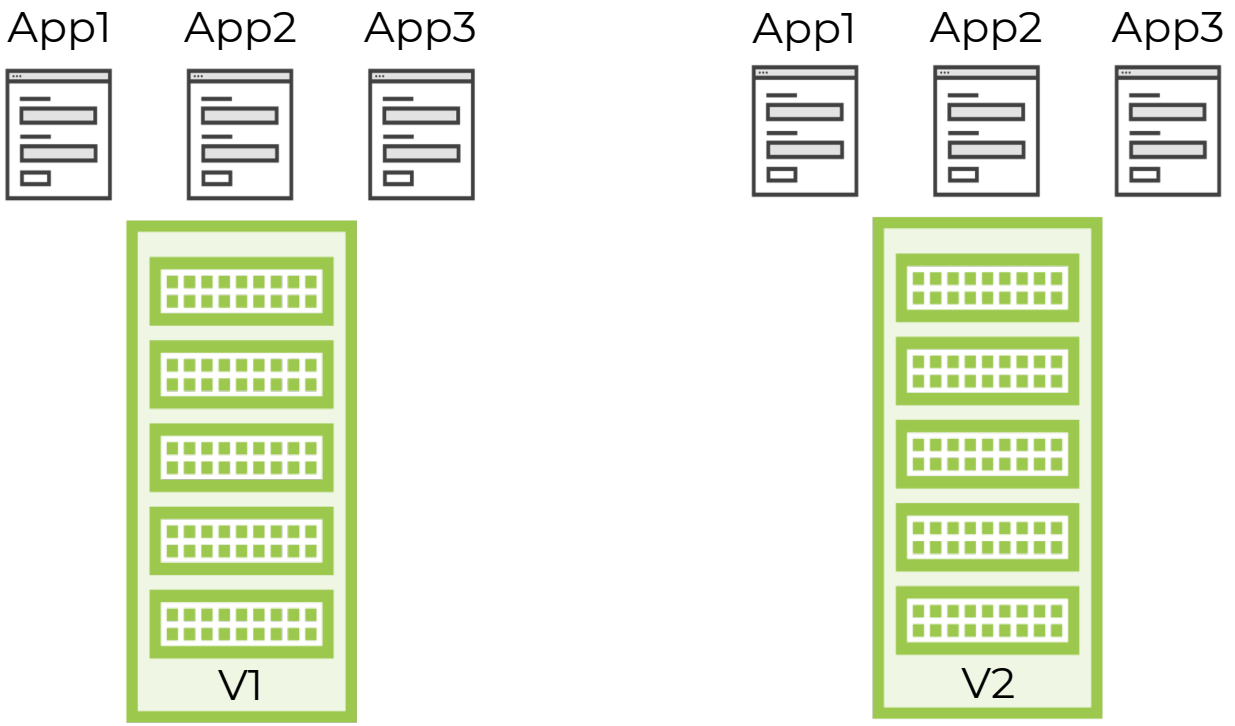


Ship Software Faster

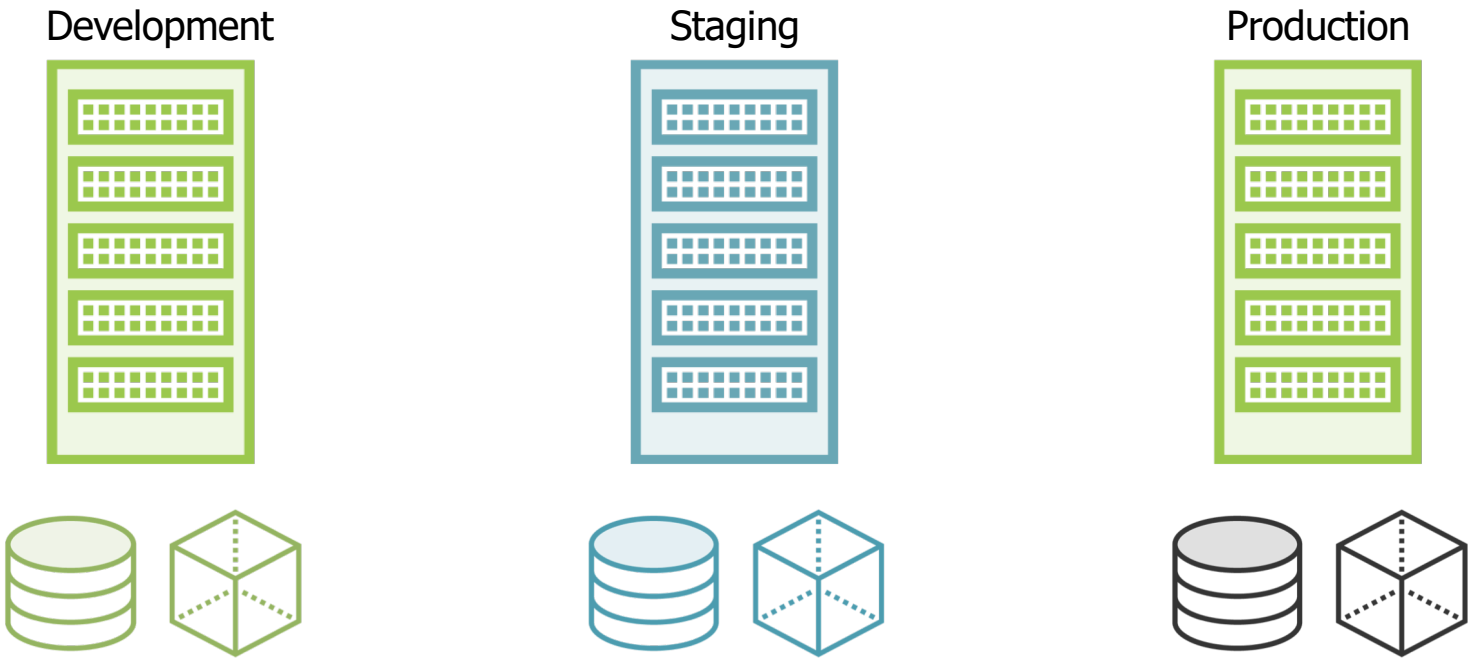
Accelerate Developer Onboarding



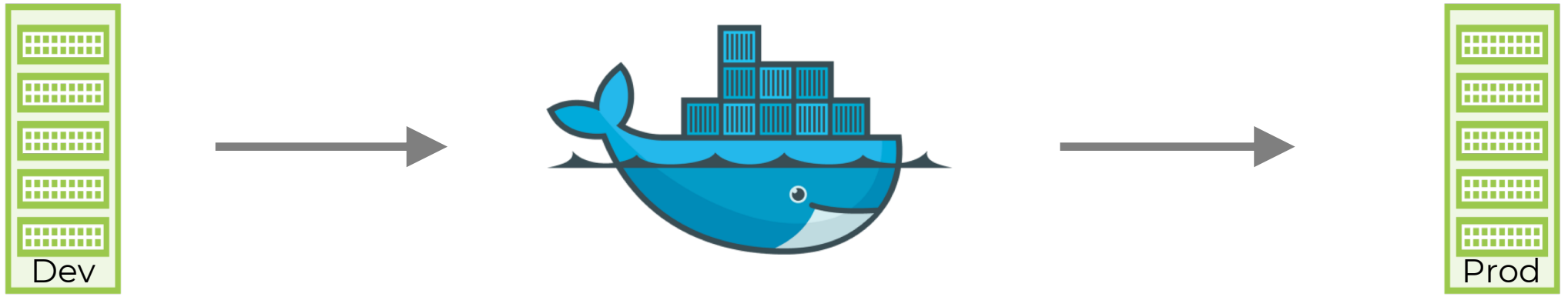
Eliminate App Conflicts



Environment Consistency



Ship Software Faster

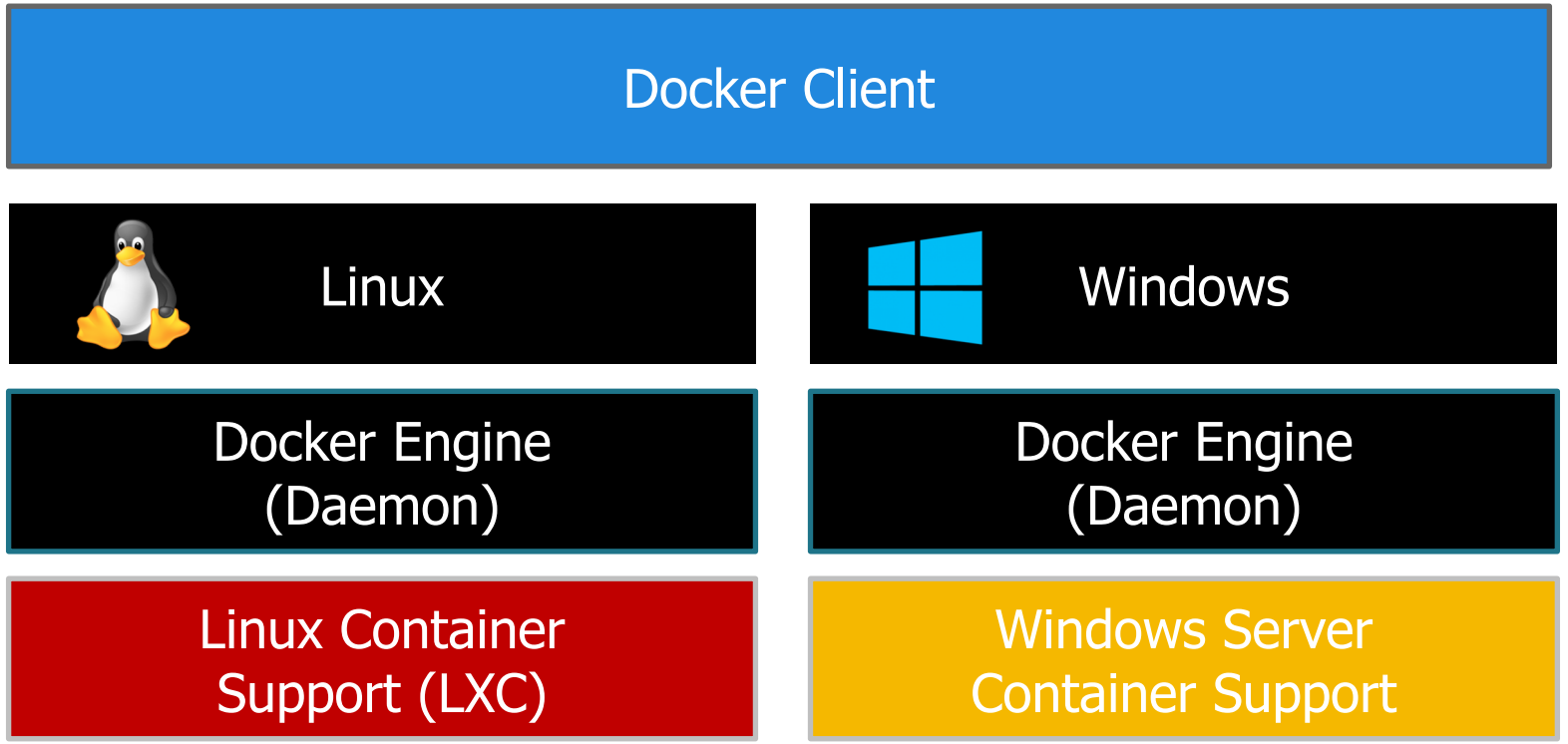


Container Benefits Review

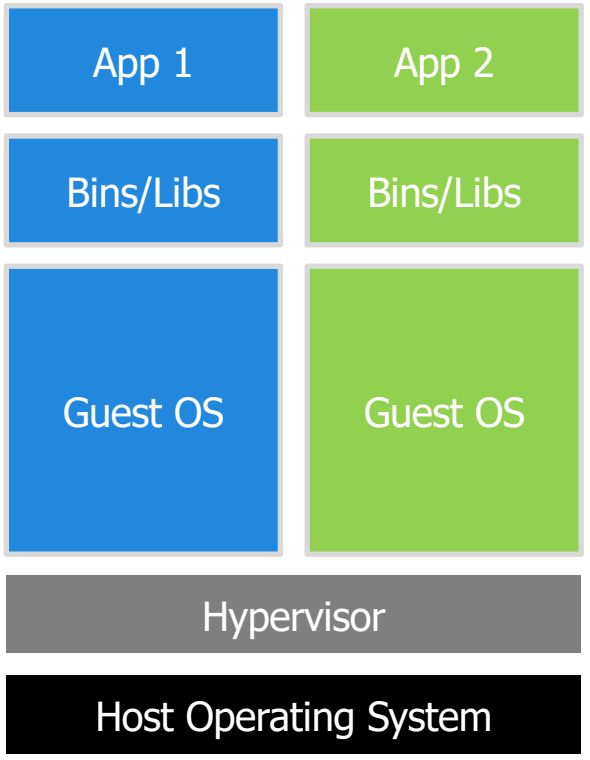
- Onboard developers more quickly to new/existing projects
- "Containerize" existing applications
- Convert monolithic applications to microservices
- Provide environment consistency
- Deploy more reliably
- Ship software faster
- Reduce infrastructure costs

Containers vs. Virtual Machines

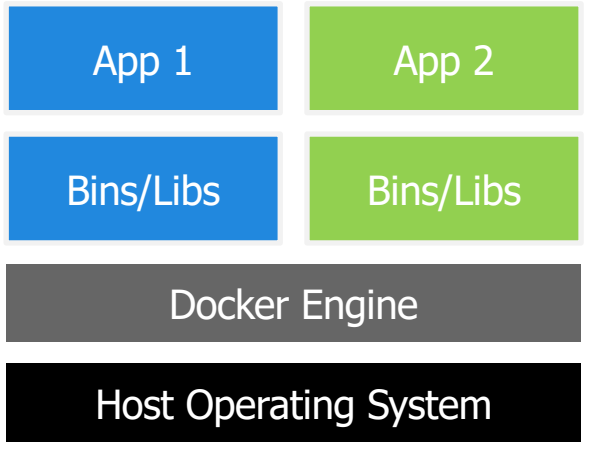
Where Does Docker Run?



Docker Containers Versus Virtual Machines



Virtual Machines

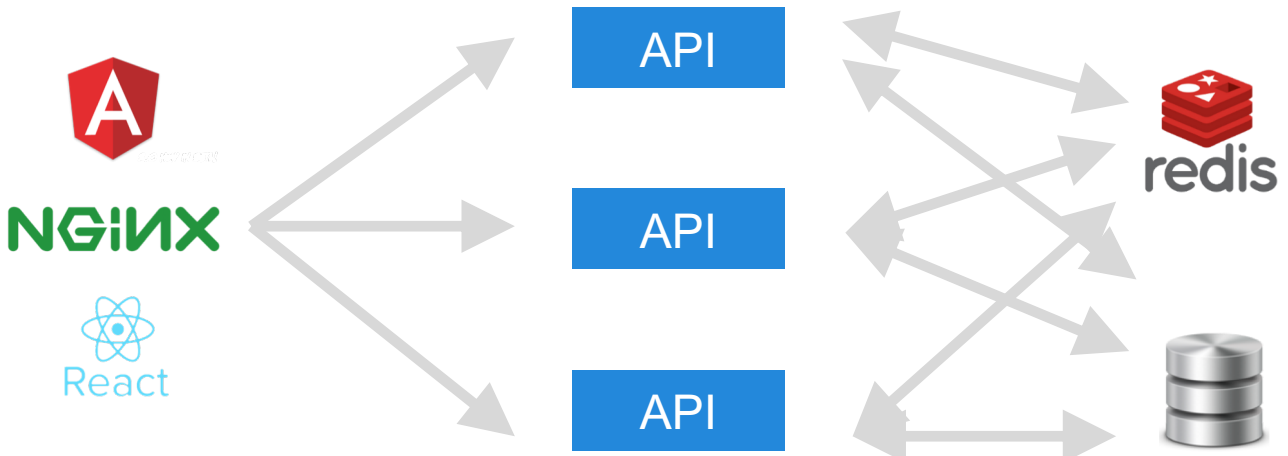


Docker Containers

Docker and Microservices

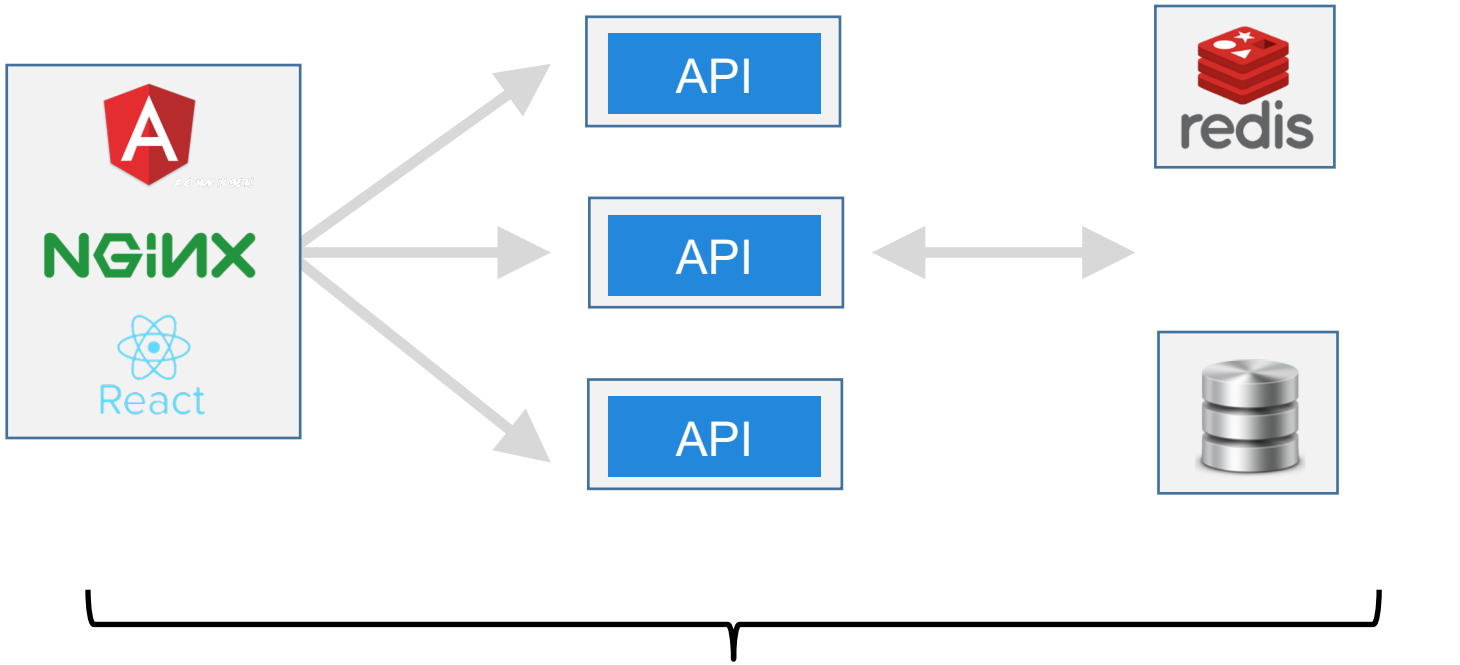
Containers simplify shipping microservices

Running Microservices



How do you make all of these services work **exactly** the same in development/staging/production?

Docker and Microservices



Containers can provide consistency between environments

Summary

- Deploying applications isn't as simple as moving code between environments
- Containers provide a way to consistently move applications between environments
- Images are read-only templates that can be used to create containers
- Docker containers are very different from virtual machines
- Docker can be used to implement microservices