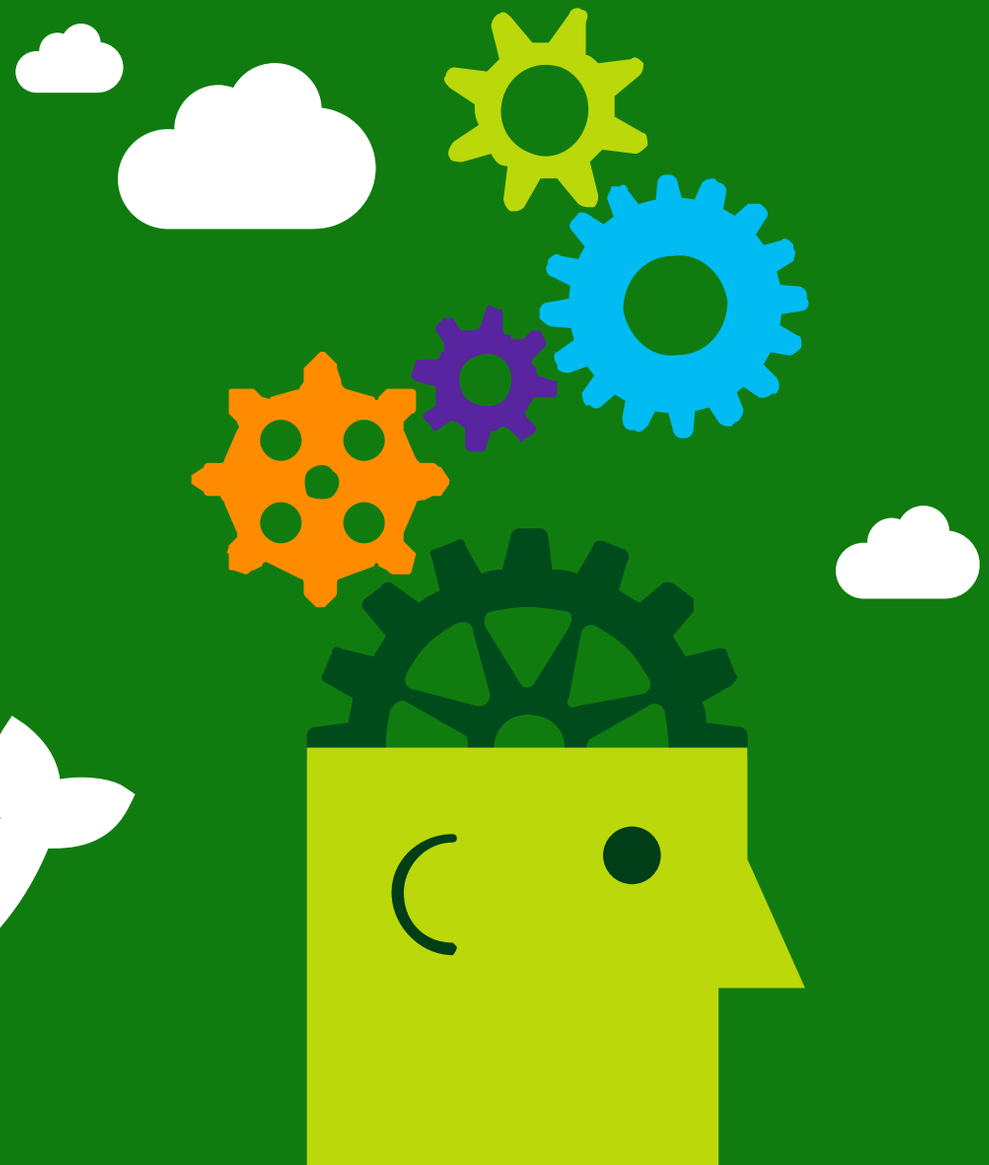
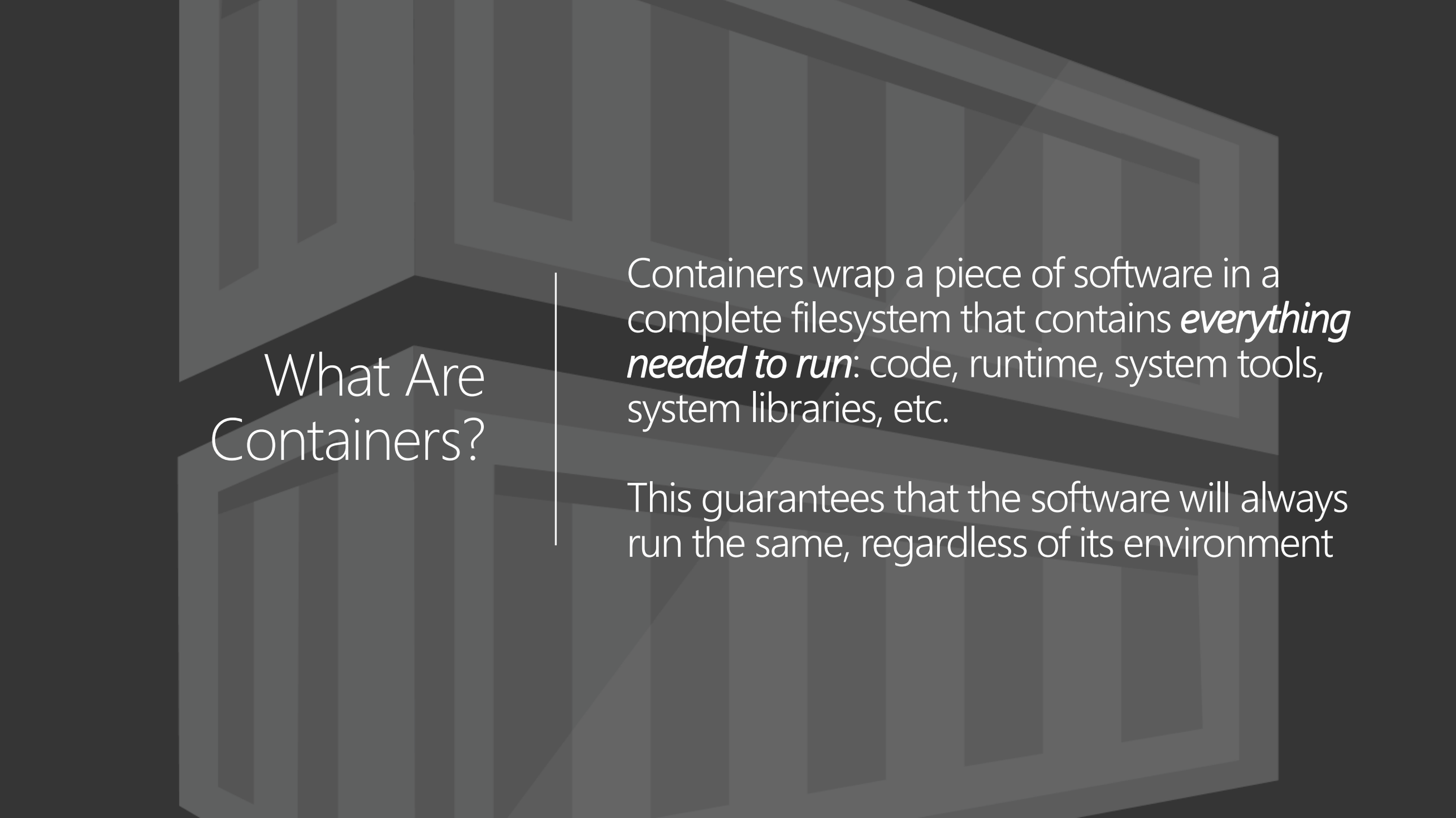
Microsoft

Containers Basics



Ross Smith
Technical Evangelist
@ross_p_smith



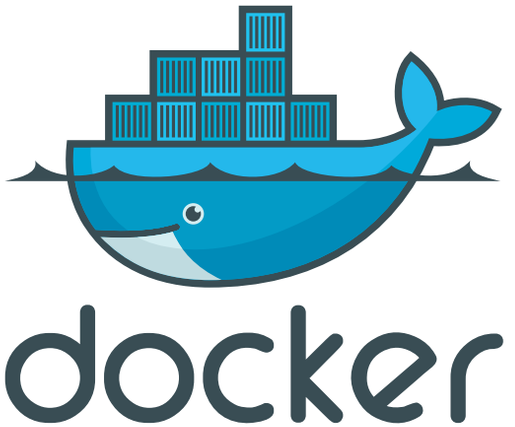
What Are Containers?

Containers wrap a piece of software in a complete filesystem that contains ***everything needed to run***: code, runtime, system tools, system libraries, etc.

This guarantees that the software will always run the same, regardless of its environment

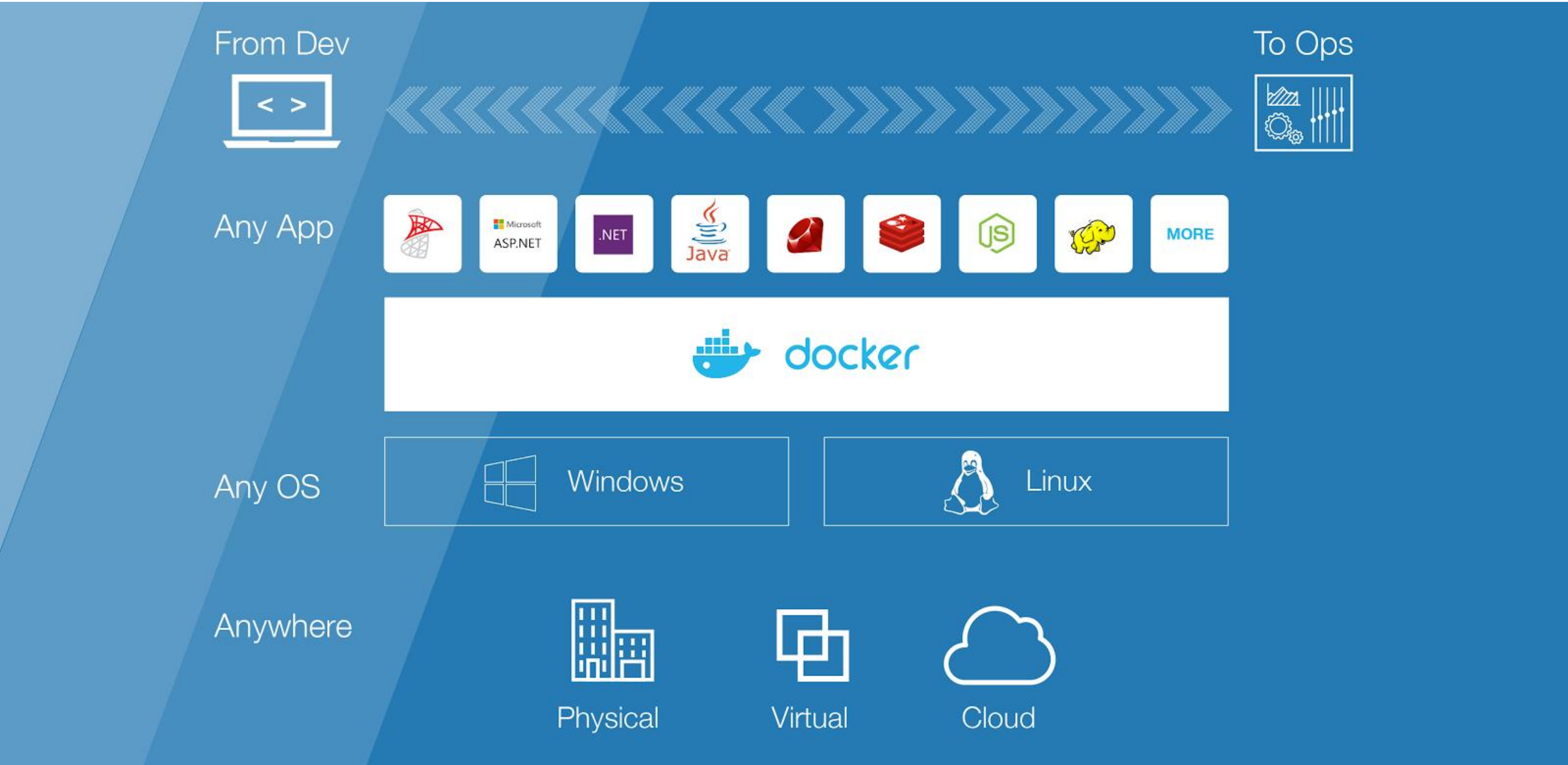
Docker and Containers

- Containers have been around for **many years** (80s)
  - Developed on top of the Linux Kernel (cgroups)
- **Docker Inc.** did not invent them
  - They created open source software to build and manage containers
- Docker made containers **easy**
- Docker is the de facto standard **container format** and set of **tools**
  - Docker CLI, Docker Engine, Docker Swarm, Docker Compose, Docker Machine etc. (more from Ben!)



Docker != Containers

Build, Ship, Run … Any App … Anywhere



Containers are not magic!

- You can't run a Linux process natively on Windows OS, so you can't run Linux processes in Windows containers
- And vice-versa…

Remember - processes in containers are running on the host kernel/OS

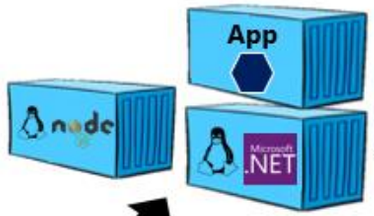
Terminology - Basics

- Container
  - A runtime instance of an image. Represents the execution of a single application, process, or service
- Image
  - A package with all the files, data, dependencies & information needed to create a running container
- Docker Engine
  - Runtime installed on a host machine (aka 'Docker host' or 'container host').
    Can be Linux, Windows Server or Desktop OS (e.g. Windows 10)
- Registry
  - A service that provides access to image repositories, typically a server or cloud service
- Repository
  - A collection of related Docker images, labelled with tags to indicate version or other metadata
- Dockerfile
  - A text file that contains instructions for how to build a Docker image

https://docs.microsoft.com/en-us/dotnet/standard/microservices-architecture/container-docker-introduction/docker-terminology

Basic taxonomy in Docker

A **Registry**
Stores many static images



Images
Static, persisted container image

Container
Image-instance running an app process (service/web)

On-premises
('n' private organizations)

- **Hosted Docker Registry**
- **Docker Trusted Registry on-prem.**

Default public registry

- **Docker Hub Registry**

Public Cloud
(specific vendors)

- Docker Trusted Registry on-cloud
- **Azure Container Registry**
- AWS Container Registry
- Google Container Registry
- Quay Registry
- Other Cloud

Docker File

```
1   FROM stefanscherer/node-windows:8.9.4-nanoserver-2016
2   ARG basedir="service-data-api"
3
4   # Node.js setup for the data-api
5   ENV NODE_ENV production
6   WORKDIR /home/app
7
8   # For efficient layer caching with NPM, this *really* speeds things up
9   COPY ${basedir}/package.json .
10
11  # NPM install for the server packages
12  RUN npm install --production --silent
13
14  # NPM is done, now copy in the the whole project to the workdir
15  COPY ${basedir}/ .
16
17  EXPOSE 4000
18  CMD npm start
```

Show me it!