Introduction to Containers

Anton Sankov 05.06.23

About Me

- Senior Software Engineer at VMware Carbon Black
- Working on Containers Security

@a_sankov

Anton Sankov

https://asankov.dev/

Containers

*New** and modern way to **build**, **package** and **run** software applications

* Not much new - containers have been around since 2013**

** And even before that

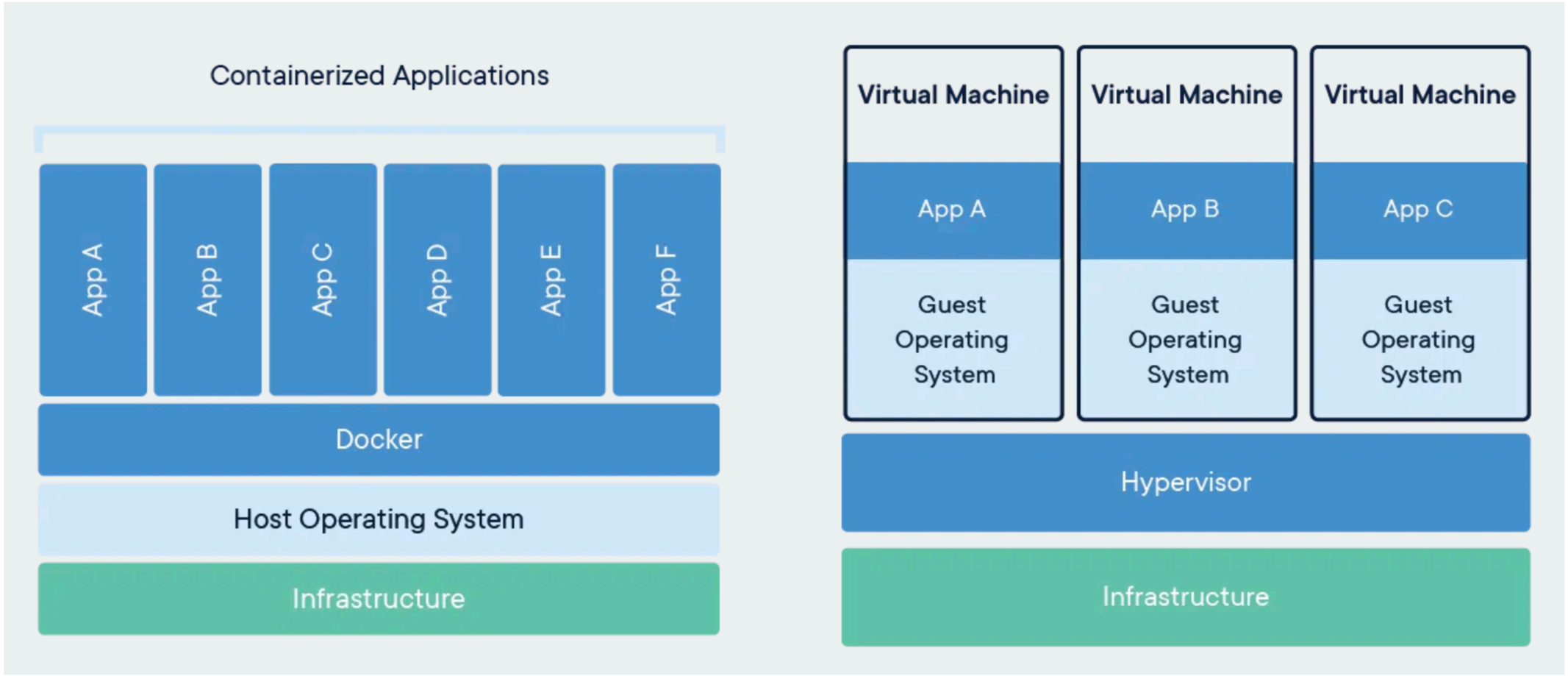
https://www.docker.com/resources/what-container/

Containers

An application container contains everything needed to run a given application - runtime, dependencies, application binary/ code, etc.

https://www.docker.com/resources/what-container/

Containers vs. Virtual Machines



https://www.docker.com/resources/what-container/

Benefits of using Containers

- **Interoperability** - you can run **any container** on **any machine** that has a container runtime installed
- **Efficiency** - containers don’t have their own OS/kernel and reuse the host OS
- **Security** - containers are isolated, and generally considered more secure than virtual machines*

* This depends on the configuration of the containers. The opposite can also be true.

Why learn containers?

- Containers are a standard way to run applications
- As a developer, it’s quite possible your application will be running in a container

- You can use containers in your local development environment for things like databases, caches, etc.
- With containers you can run applications, without having any language-specific tooling installed on your machine

Linux and Windows Containers

- 99% of the time when someone says “Containers” they mean “**Linux containers**”
- Windows containers also exist, but they are not widely used
- **In this presentations we are going to talk only about Linux containers**

A little bit of history

Pre-2013

Containers work by utilising Linux kernel features like **cgroups** and **namespaces**.

These features have existed for a long time, but were considered complex and hard to configure and use.

That’s why not mean people used containers before 2013…

https://en.wikipedia.org/wiki/Cgroups
https://en.wikipedia.org/wiki/Linux_namespaces

A little bit of history

Post-2013

A company called dotCloud was using Linux containers and building tools to make running, building and distributing containers easier.

The company was later renamed to **Docker** and focused on developing these tools.

And the rest, they say is history…

The Future of Linux Containers, Solomon Hykes, PyCon 2013

https://youtu.be/9xciauwbsuo

But what exactly are these tools?

Runtimes

Images

Registries

Container Runtimes

The component that runs the containers

Popular container runtimes are **Docker**, **containerd** and **runc**

https://www.docker.com/
https://containerd.io/
https://github.com/opencontainers/runc

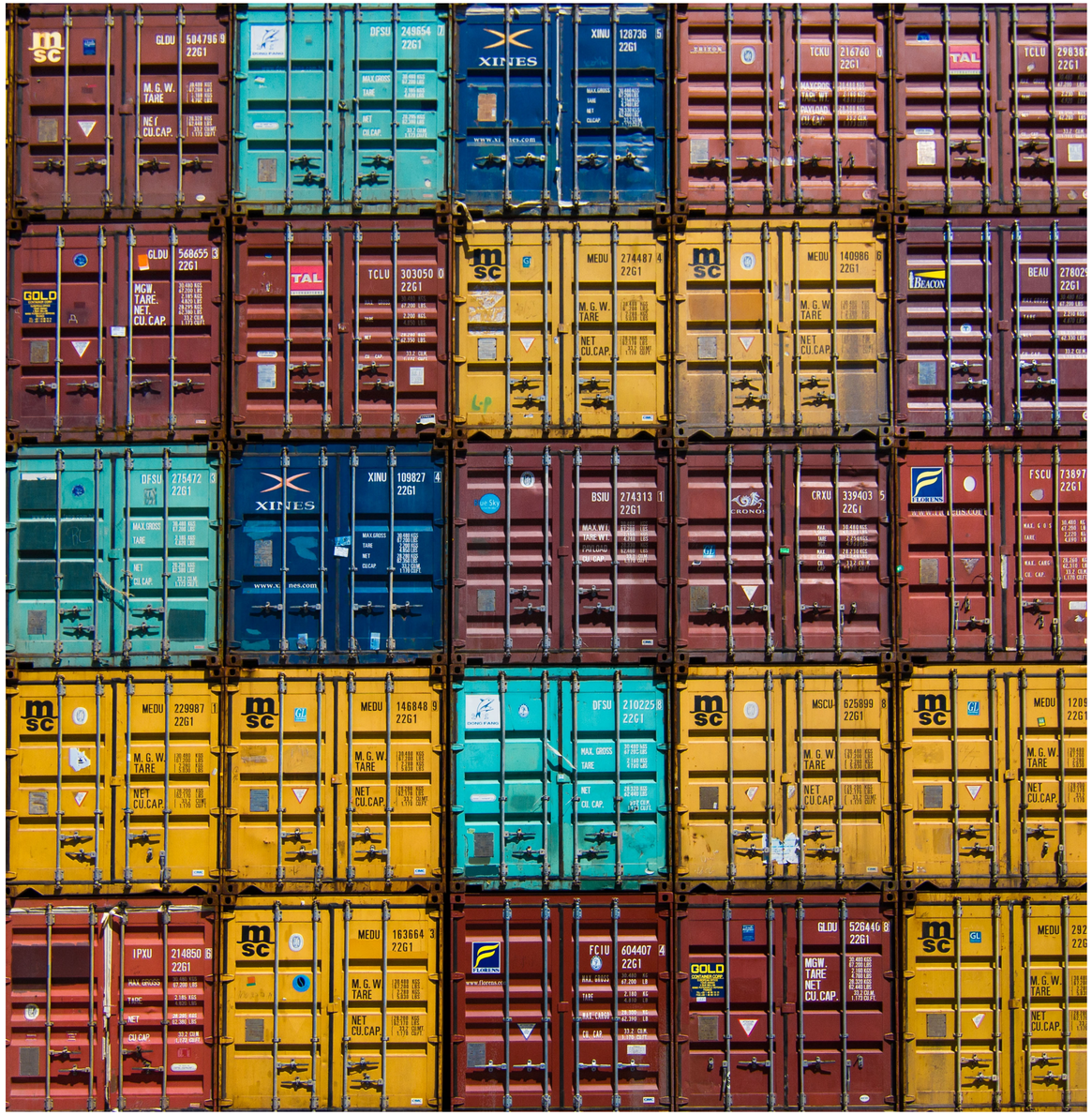
Container Images

The blueprint for a container

Container runtimes run containers from images

Images consist of **layers**

We can build images via **Docker** or **Buildpacks**



https://www.docker.com/
https://buildpacks.io/

Container Registries

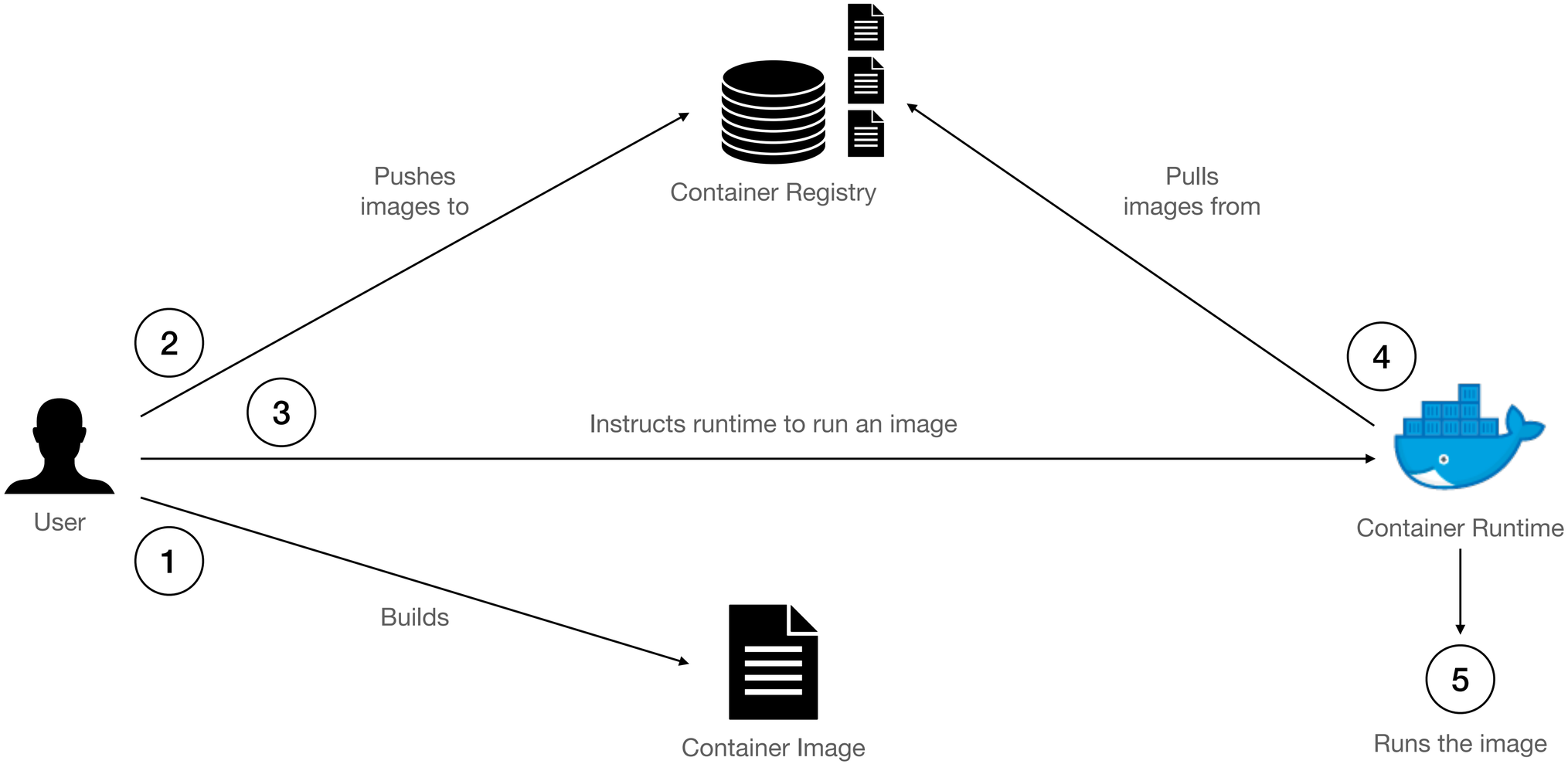
A place that **stores** container images

Container runtimes pull images from containers registries and run containers from them

Popular container registries are **Docker Hub**, **GitHub Container Registry**, **quay.io**, **Harbor** and others.

https://hub.docker.com/
https://github.com/
https://quay.io/
https://goharbor.io/

Let’s run some containers


Container Registry
Pushes
images to
Pulls
images from
2
3
4
Instructs runtime to run an image
User
Container Runtime
1
Builds
5
Container Image
Runs the image

Summary

- Containers are one of the most popular way to build and run applications
- Containers have been around for a long time, but went mainstream around 10 years ago
- **Docker** is one of the preferred ways to **build**, **run** and **store** containers

Thank you!

@a_sankov

Anton Sankov

https://asankov.dev/

Link to slides and examples