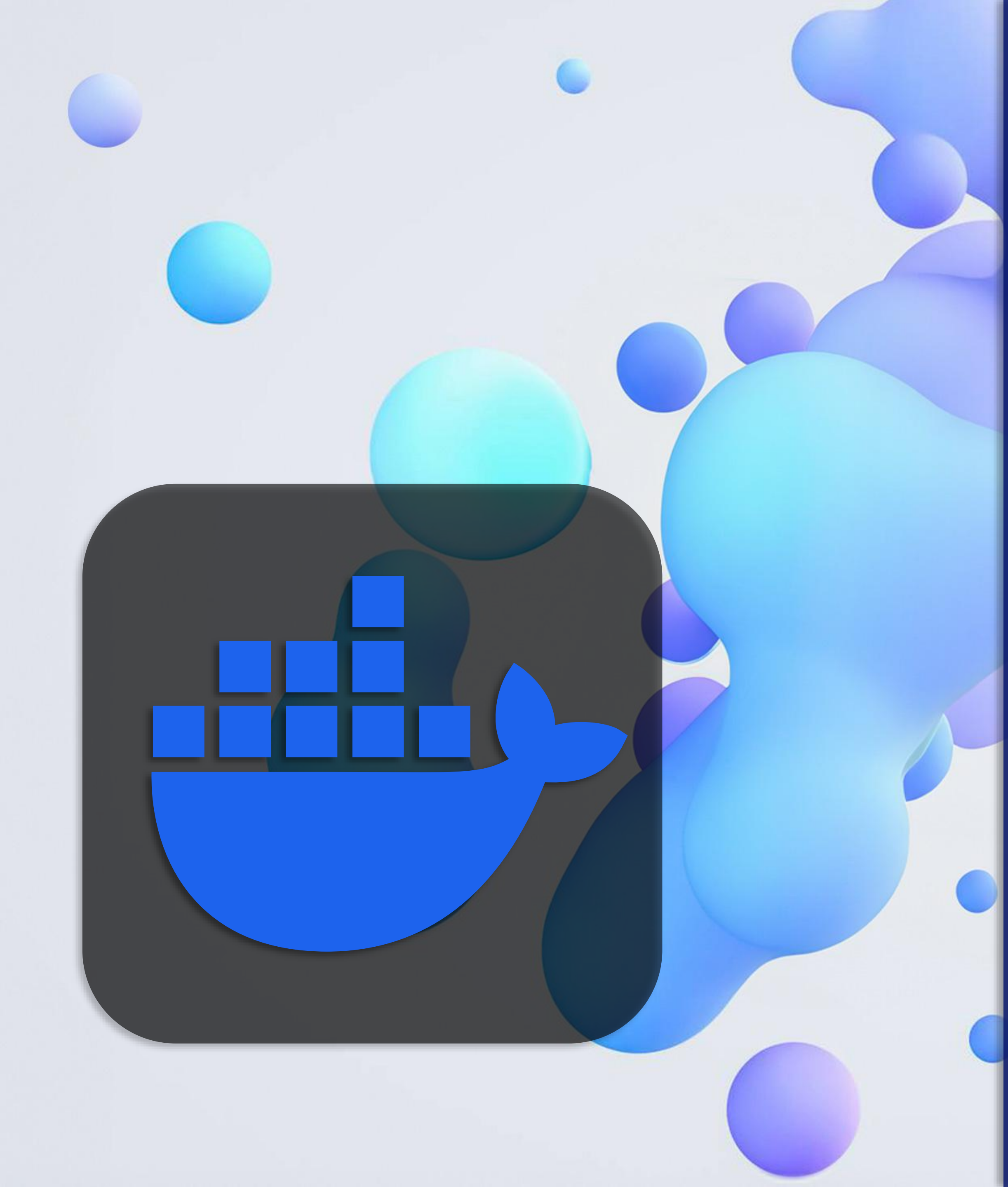


Diet **Docker**

CRAFTING LIGHTWEIGHT CONTAINERS WITH MULTI-STAGE BUILDS

Agenda

ABOUT

THE PLATFORM

THE BASE

THE FILE

THE CONTENTS

QUESTIONS

Hello from Grand Rapids!

I’m Victor Frye

Your friendly neighborhood developer

Unchallenged #1 Clippy fan

Love internet culture and content
- victorfrye.com/blog
- microsoftgraveyard.com
- shrugman.com

Passion for .NET, React, Azure, Xbox

DevOps specialist at Leading EDJE

Building with Microsoft solutions

About **Leading EDJE**

Technology consultancy and services

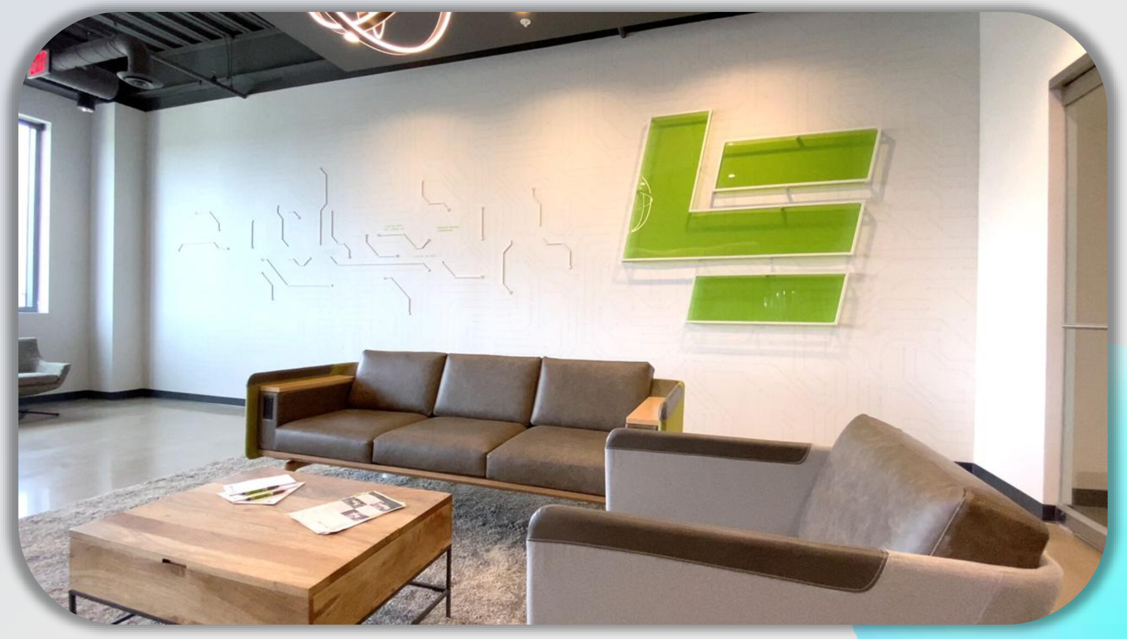
- Strategists, innovators, partners
- Craft bespoke solutions
- Positive disruption of business norm

Real. Fun. Geeks.

FREE lunch and learns

Expert talent

- Cloud infrastructure
- Digital and app innovation
- Data and AI
- Project management



The Platform

DOCKER AND CONTAINERIZATION

Abstracting the machine

Virtualization	Containerization
Virtualize the hardware	Virtualize the OS
Abstract compute and storage	Abstract OS and infrastructure
Runs multiple apps	Bundles single app and deps
Can only run one OS at a time	Lightweight solution
Slow to spin up	Can spin up quickly*
Best for complete control	Best for portability and perf

The Docker platform

A containerization tool

A daemon and a client

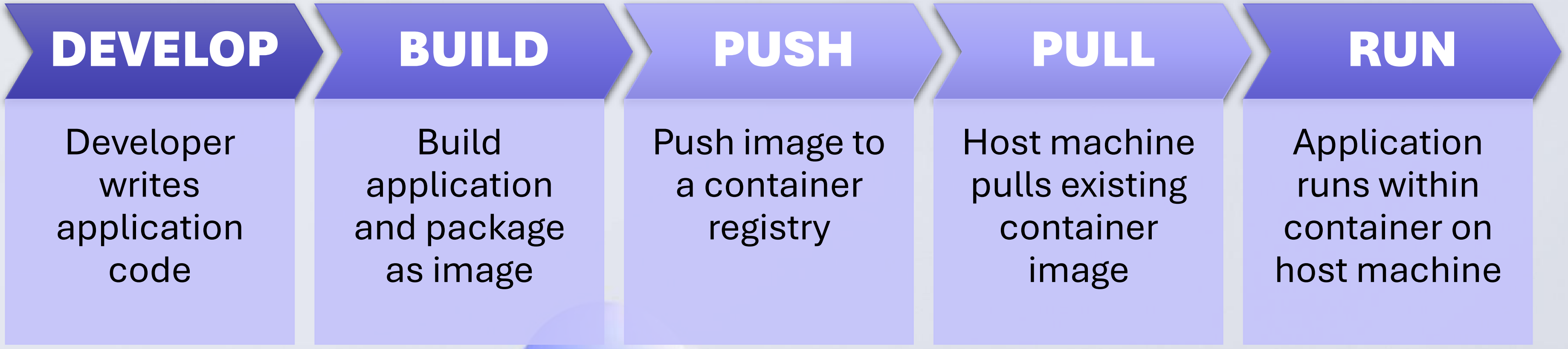
Objects include:

- Images: read-only templates with instructions for creating a container
- Containers: a runnable instance that can start, stop, or move
- Registries: A package store for container images

Alternatives and compliments

- Podman
- containerd + nerdctl
- Buildah
- Runc
- Kubernetes
- Red Hat OpenShift
- Azure Container Apps
- Amazon ECS

Container app lifecycle



DEVELOP

Developer writes application code

BUILD

Build application and package as image

PUSH

Push image to a container registry

PULL

Host machine pulls existing container image

RUN

Application runs within container on host machine

```
FROM mcr.microsoft.com/dotnet/sdk:10.0
WORKDIR /app

COPY . .
RUN dotnet publish ./MyApp.csproj -c Release -o /out

ENTRYPOINT ["dotnet", "/out/MyApp.dll"]
```

EXAMPLE DOCKERFILE FOR A .NET APPLICATION (1.25 GB)

The Base

WHERE ARE YOU FROM?

All about **Linux**

Choose your **distro**

Debian-based

Debian or Ubuntu
- Debian: Toy Story (bookworm, trixie)
- Ubuntu: alphabetical (jammy, noble)

Uses **glibc** for C library

Includes common utils

Large base image sizes

Alpine

Just Alpine

Uses **musl** instead of glibc

Includes **BusyBox** for utils

No **tzdata**

Very minimal out-of-the-box

Small base image size

Why does size matter?

The File

READ THE INSTRUCTIONS ALREADY

Dockerfiles

DEMONSTRATION

The Contents

SIZE MATTERS, BUT SO DOES HOW YOU USE IT

Go **distroless**

GOOGLE PIONEERED OPTION

HARDEN IMAGE BY:
- REMOVING UTILITIES
- REMOVING PACKAGE MANAGERS
- REMOVING THE DISTRO ITSELF*

GOOGLE DISTROLESS

AZURE LINUX

DOCKER HARDENED IMAGES

Distroless

DEMONSTRATION

Syft images

DEMONSTRATION

Thank you

Provide feedback

VICTOR FRYE
616-706-7407
VICTORFRYE@OUTLOOK.COM
VICTOR.FRYE@LEADINGEDJE.COM
LEADINGEDJE.COM

VICTORFRYE.COM/BLOG
LINKEDIN.COM/IN/VICTORFRYE
GITHUB.COM/VICTORFRYE/PRESENTATIONS
GITHUB.COM/VICTORFRYE/HELLODOCKER