From sketch to Kubernetes

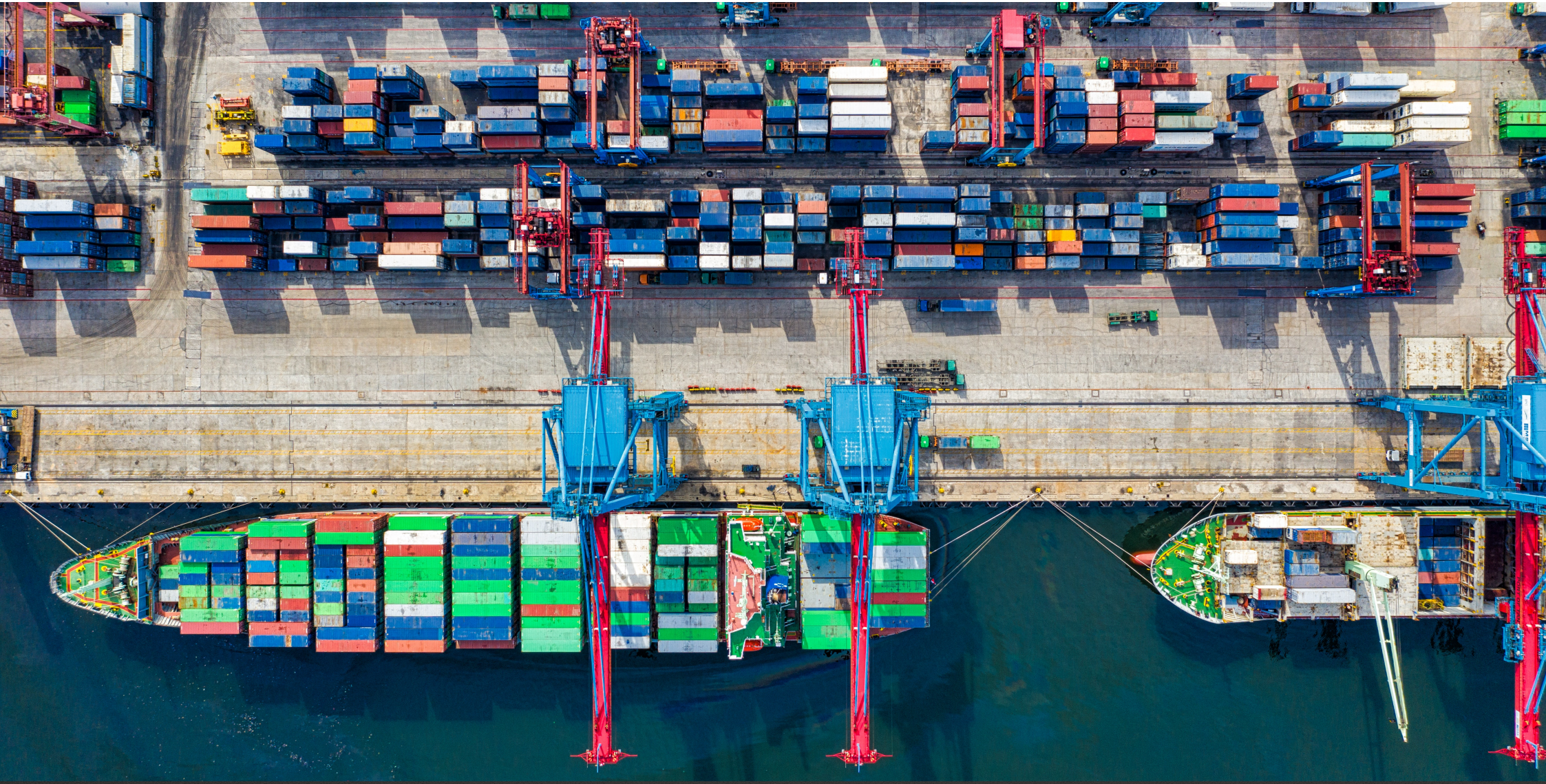
A developer’s journey into infrastructure

William Mendes

Proud Performance Engineer @ Farfetch
Solutions & Software architect

DESIGN AND CODE

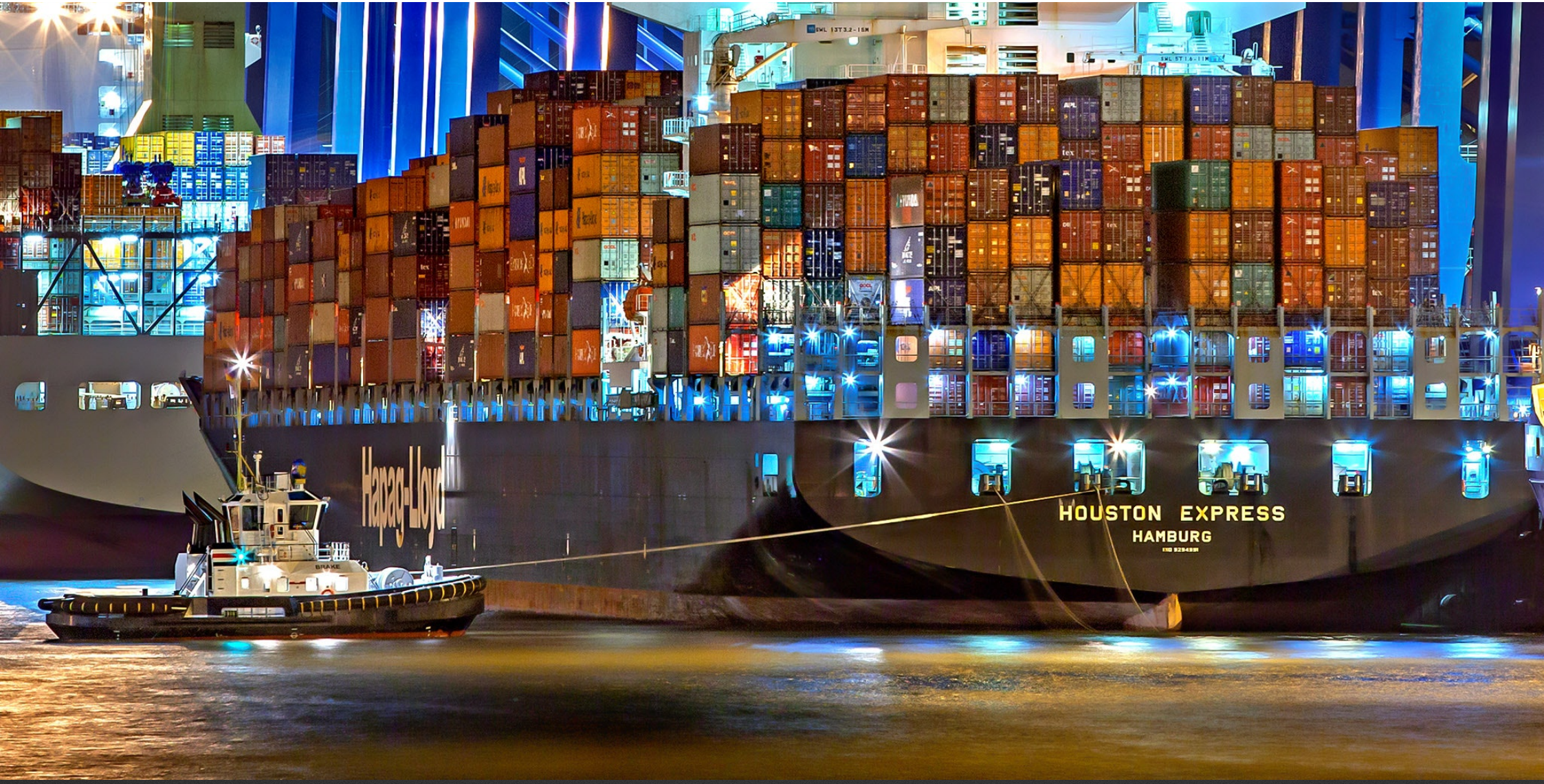
- Code first;
- Run locally
- Test locally
- Deploy directly, moving files.



LOAD TO LAUNCH

- Same code, with a DOCKERFILE or Docker-Compose
- Find and fix
  - Dependencies, files, ports, accesses
- Test in multiple environments
- Deploy packages/images, manually or automated

DO	DO NOT
DOCKERIZE	DOCKERIZE
Faster configurations	Faster and tuned application/OS
Faster deployments	Built-in security
If you can relay on external security	If you can relay on external security
Low Storage	High storage/file usage
Command/Web Applications	Easy, simple and fast maintainability



LOAD TO LAUNCH

- Think first, map and measure your needs
- Provisions
  - Azure Kubernets Cluster
  - Azure Container Registry
  - Azure DevOps Pipelines (Github/Bitbucket integration)
- Automated deploy

Stay attached

Simple/Lightweight Applications (Monolithic)

Security Concerns

Replication hell

Culture and Maturity

SAIL FAST

QUESTIONS

Bibliography

https://docs.microsoft.com/en-us/dotnet/core/docker/build-container

https://docs.microsoft.com/en-us/azure/devops/pipelines/apps/cd/deploy-aks?view=azure-devops&tabs=dotnet-core

https://docs.microsoft.com/en-us/azure/aks/cluster-container-registry-integration

https://docs.microsoft.com/pt-pt/azure/aks/kubernetes-walkthrough-portal

https://github.com/MicrosoftDocs/pipelines-dotnet-core-docker

https://www.freecodecamp.org/news/7-cases-when-not-to-use-docker/

https://techbeacon.com/devops/one-year-using-kubernetes-production-lessons-learned


YOU
GOT
THIS