

# docker

Alex Ovidiu Popa

Specialized Protocols in Computer Networks

# Presentation Agenda

1. Introduction
2. Docker Containers, Images vs Virtual Machines
3. Dockerfile, Useful commands
4. Docker Hub
5. Demo
6. Q&A



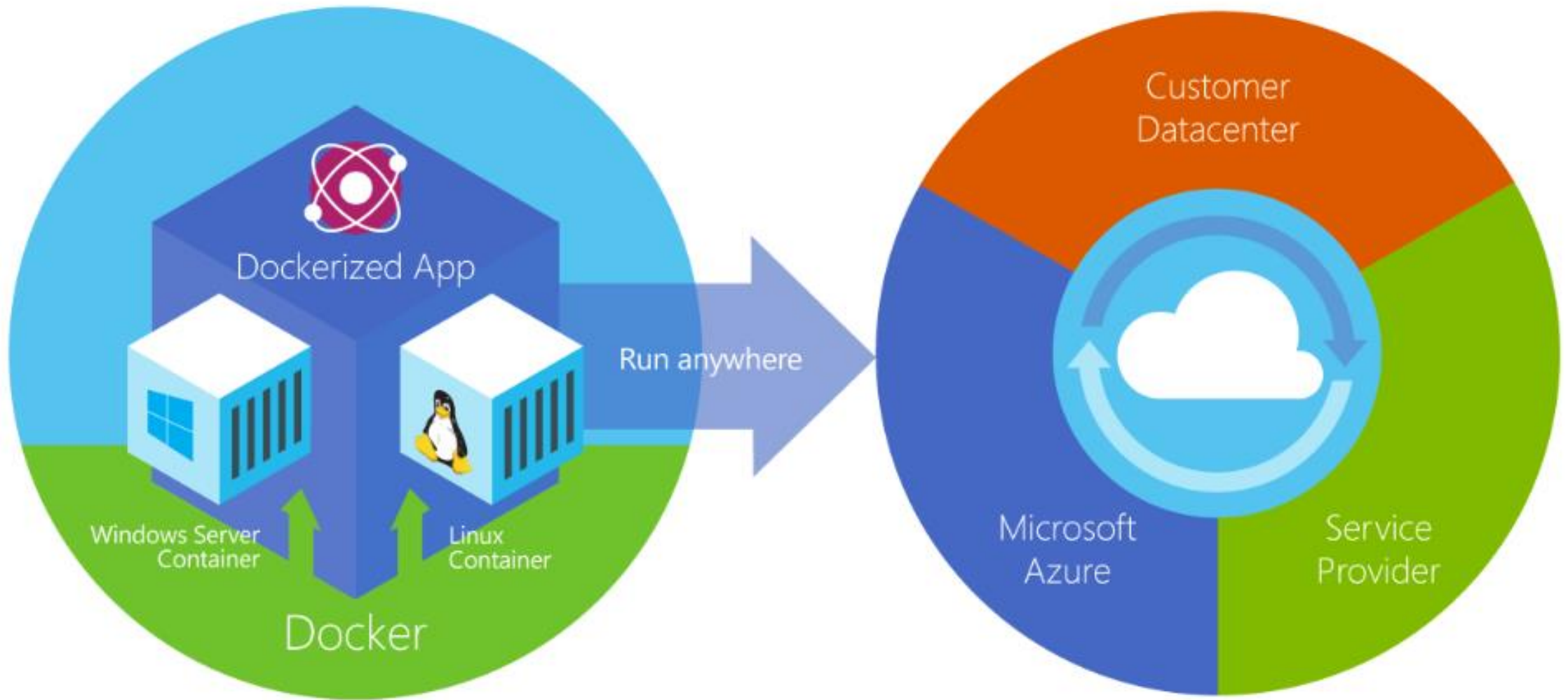
# What is Docker?

Docker is an open-source project for automating the deployment of applications as portable, self-sufficient containers that can run on the cloud or on-premises.

Available on all operating systems, using Virtualization.

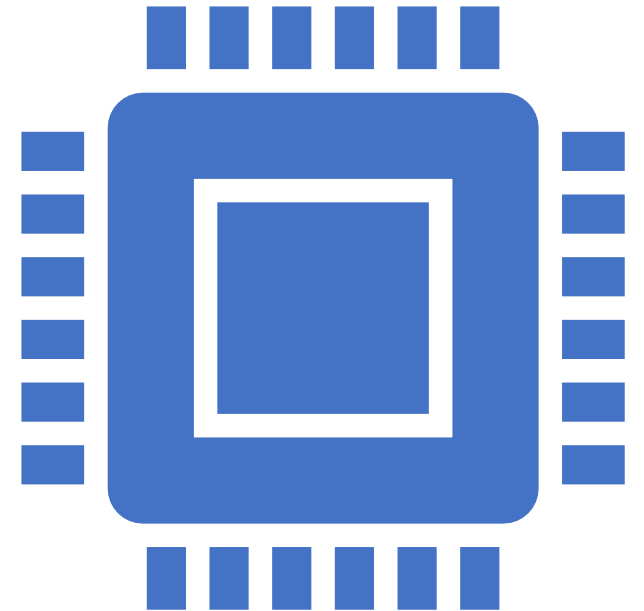
Hyper-V enables the use of virtualization and virtual machines.

# Why is Docker useful?

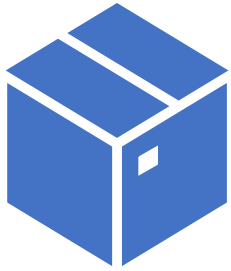


# Containers and Images

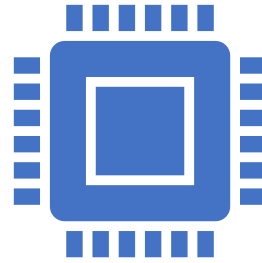
- **A container** is a standard unit of software that packages up code and all its dependencies, so the application runs quickly and reliably from one computing environment to another. (think of a server running an app)
- A Docker container **image** is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings. (think of the JAR for a Java program)



# Advantages of using Docker



**Standard:** Docker created the industry standard for containers, so they could be portable anywhere

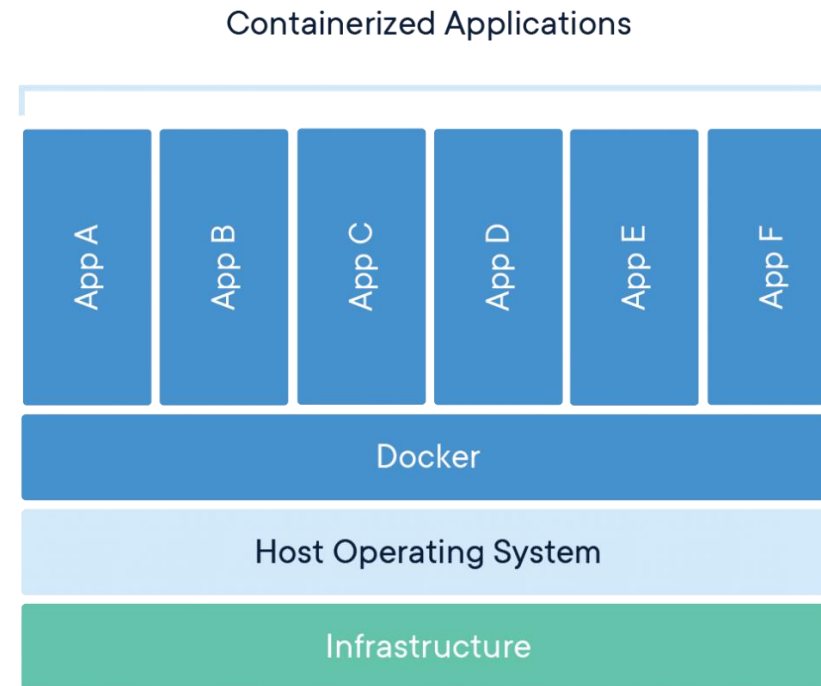
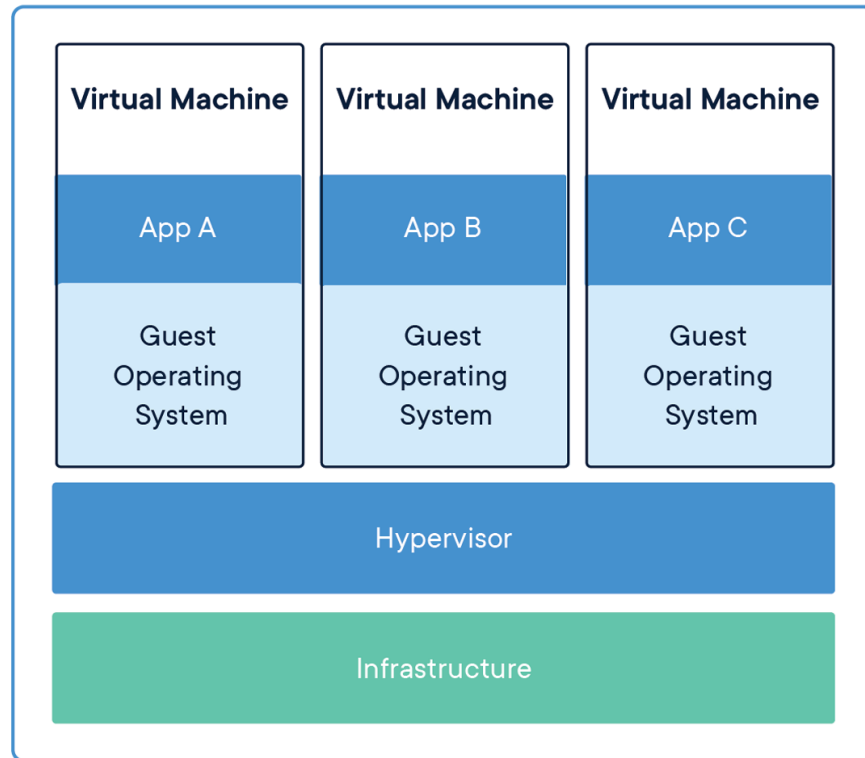


**Lightweight:** Containers share the machine's OS system kernel and therefore do not require an OS per application, driving higher server efficiencies and reducing server and licensing costs



**Secure:** Applications are safer in containers and Docker provides the strongest default isolation capabilities in the industry

# Virtual Machines vs Containers



## Containers

- Abstraction at the app layer that packages code and dependencies together
- Multiple containers can run on the same machine
- Not resource demanding, typically in the order of MBs

## Virtual Machines

- Abstraction of physical hardware turning one server into many servers
- Hyper-V allows multiple VMs to run on the same machine
- Resource heavy, slow to boot, copy an entire OS

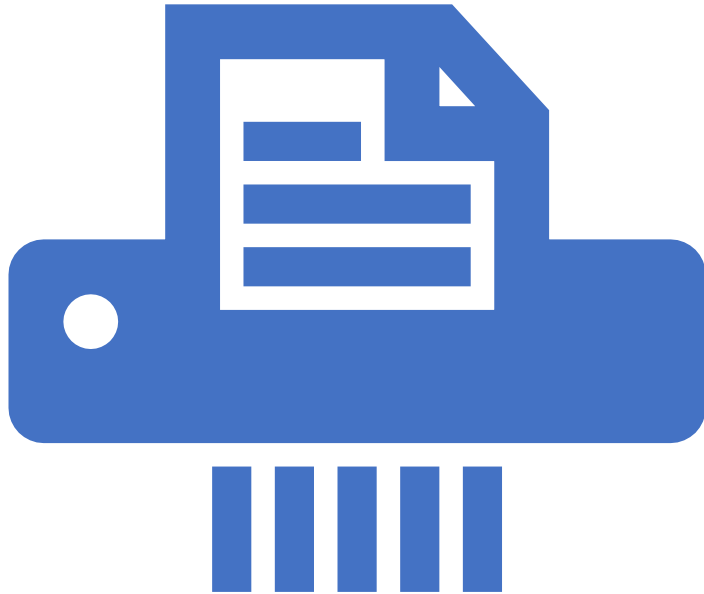


# Dockerfile

- Docker can build images automatically by reading the instructions from a **Dockerfile**.
- A **Dockerfile** is a text document that contains all the commands a user could call on the command line to assemble an image.
- Using **docker build** users can create an automated build that executes several command-line instructions in succession.

# Useful Commands

- `docker build`
- `docker run` – start an image
- `docker compose` – services, logging, jobs
- `docker images` – list all images on the host
- `docker container`

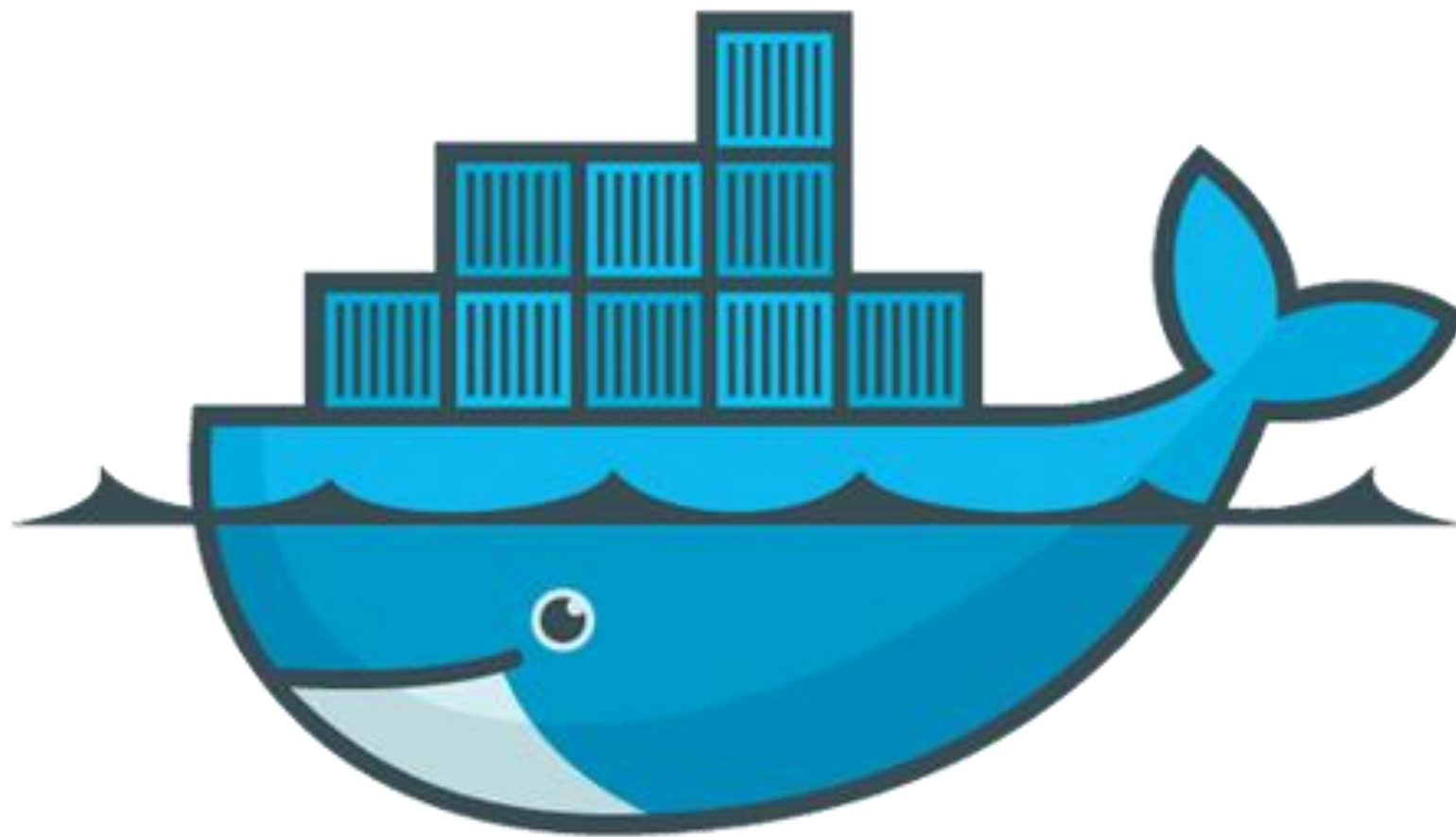


# Docker Hub

- [Docker Hub](#) is the default **registry** provided by Docker for finding, sharing and saving container images with anyone.
- It is the world's largest repository of container images.
- Images can be easily fetched from the Hub and loaded onto the desired cloud/server to be deployed.
- Alternative: Docker Cloud

# Working with Docker Hub

- `docker push`
- `docker pull`
- `docker tag`



Demo Time

Questions?

Thank you!

# References

- <https://docs.docker.com/reference/>
- <https://docs.docker.com/docker-hub/>
- <https://docs.microsoft.com/en-us/dotnet/architecture/microservices/container-docker-introduction/docker-defined>