

# Docker for Databases

# Audience

- People completely new to Docker
- Developers/DBAs not sure how Docker fits in with databases

# Agenda

- What is Docker?
- Why Docker?
- How does it work?
- Demo
- Docker + Databases = 😍

# Goals

- Understand what Docker is
- Understand when to use and when not to use Docker with your DB

# Who am I?

- Director of Engineering at Lean TECHniques
- Co-organizer of [Iowa .NET User Group](#)
- [Friend of Redgate](#)
- Built pipelines for dozens of databases
- Blog at [scottsauer.com](http://scottsauer.com)



# What is Docker?

- Tools to allow you to run applications in an isolated environment called containers (agnostic to Docker)
- Backed by Docker Inc.
- Synonymous with containers, like Kleenex

# What is an image?

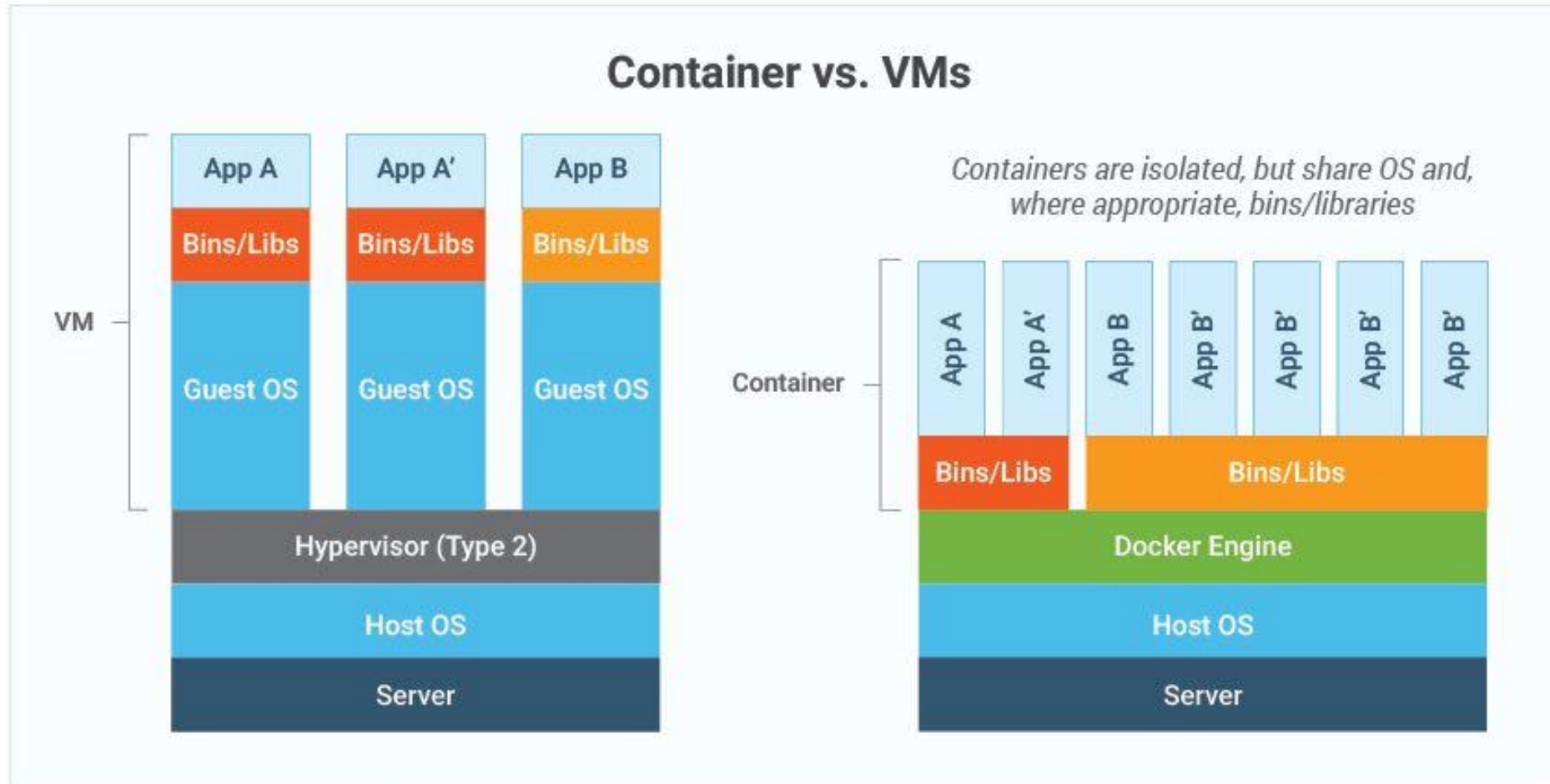
- Immutable file with instructions for how to create a container
- Made up of layers
- Starts with an OS
- Install any software

# What is a container?

- Runnable instance of an image
- Liked a self-contained VM
- Isolated from other containers
- Can run multiple containers from the same image (i.e. multiple 2019 MSSQL containers)
- Or from other images (i.e. 2019 MSSQL, 2017 MSSQL, etc)



# Containers vs Virtual Machines



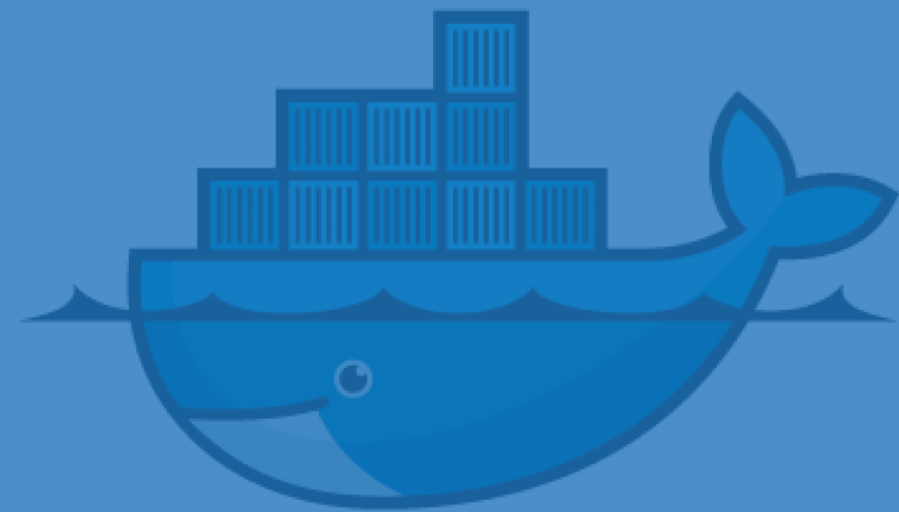
# Why Containers?

- “Works on my machine”
- Isolated from each other and host machine
- Repeatable
- Fast to spin up
- Easy to tear down
- Multiple instances and versions

# Container Registry

- Sharing images
- Images tagged with versions (i.e. MSSQL 2022, 2019, specific CUs, etc)
- Docker Hub is public example
- Can have an internal one just for your company
- Like NuGet or npm for sharing code but for Docker

# Demo



docker

# DevOps Challenges With Databases

- Development
- CI
- Isolated from other team members
- Applying migrations
- Version matching

# Which of these does Docker solve?

- Development
- CI
- Isolated from other team members
- Applying migrations
- Version matching

# Which of these does Docker solve?

- Development
- CI
- Isolated from other team members
- Applying migrations
- Version matching

# When do I NOT Docker my DB?

- Non-local/CI environments
- Definitely not for Production
  - Prefer PaaS over IaaS



# Docker Adoption Progression

- Start using Docker with local dev
- Start use Docker in CI for tests
- Start publishing Docker images with your schema
- Start including common datasets with your schema
- Consider thinking about backing up Production nightly and publishing a Docker image with the schema and data (after sanitizing)

# Takeaways

- Overview of what Docker is
- Where Docker fits (Local + CI) and where it doesn't (Production)

# Questions?

Contact: [ssauber@leantechniques.com](mailto:ssauber@leantechniques.com)

# Thanks!