

# Container Fundamentals and Introducing Kubernetes

**Anthony E. Nocentino**  
[aen@centinosystems.com](mailto:aen@centinosystems.com)



# Course Overview

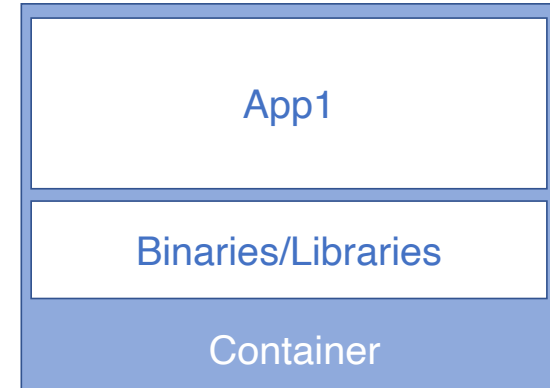
- **Module 0** - Introduction
- **Module 1** - Container Fundamentals
- **Module 2** - Kubernetes Architecture and API Objects
- **Lunch @ 12:00-12:45**
- **Module 3** - Interacting With Your Cluster
- **Module 4** - Deploying Applications in Kubernetes
- **Module 5** - Building and Deploying Container-based Applications in Kubernetes

# Agenda

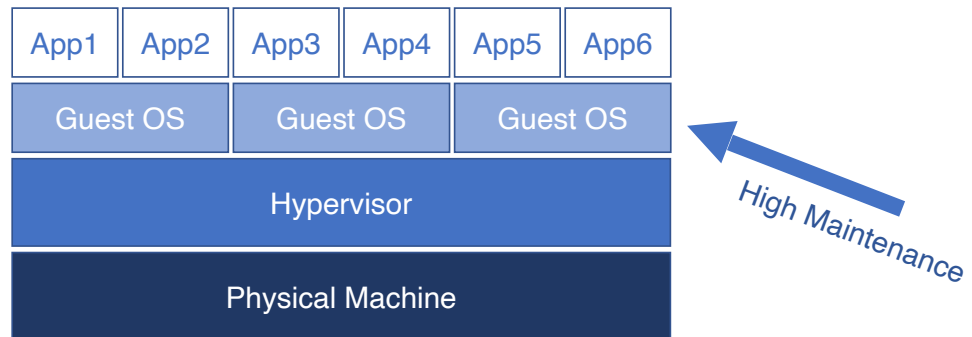
- **Container and Linux Fundamentals**
  - Container Fundamentals
  - Container Based Application Deployment
  - The Need for Container Orchestrators

# Container Fundamentals

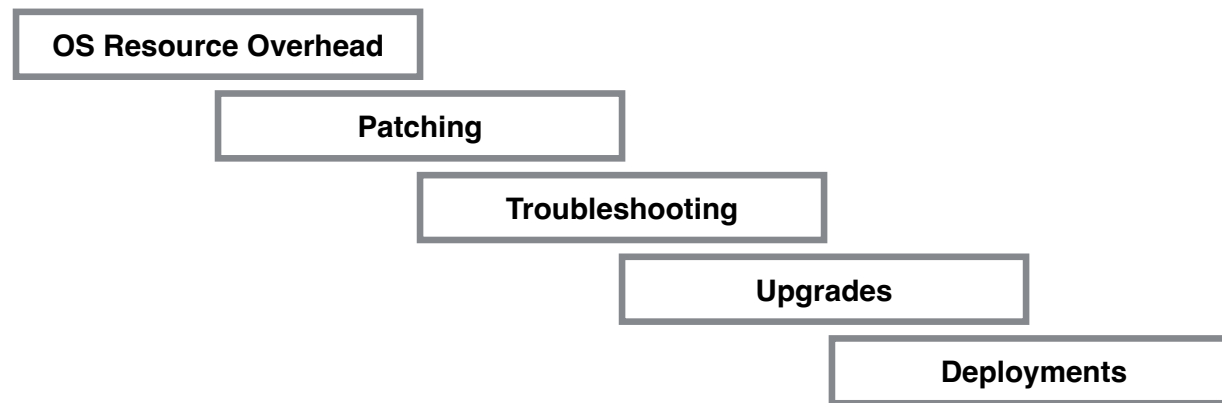
- Operating system virtualization
  - Shared kernel and system resources
- Container...contain...
  - Binaries, libraries and file system
- One app inside the container
  - This is the unit of work
- Containers are ephemeral
  - Let's start off with a comparison...



# Virtual Machines

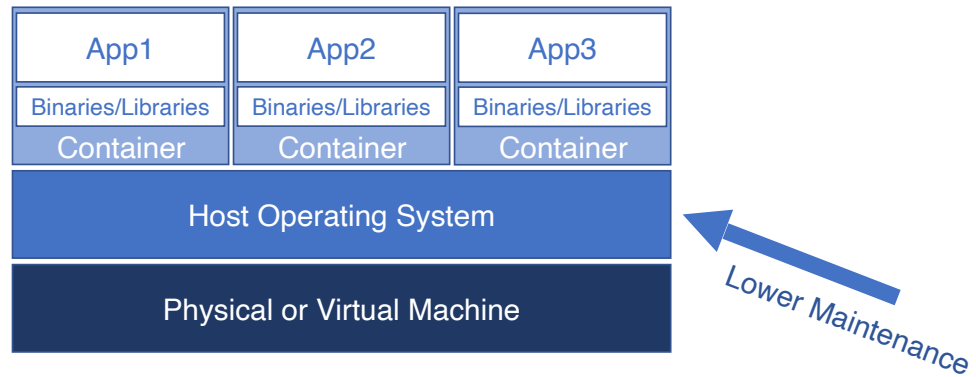


# What's so Hard About Virtual Machines?

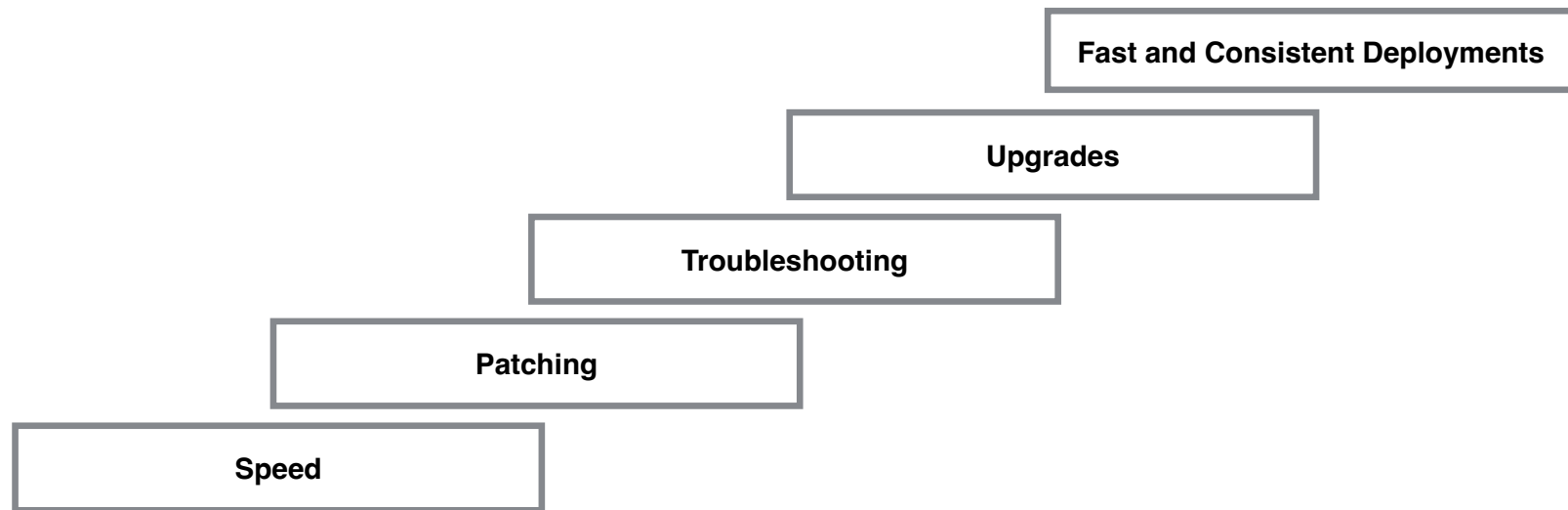


**Does any of this move your business forward?**

# Containers



# What do Containers Bring to the Table?

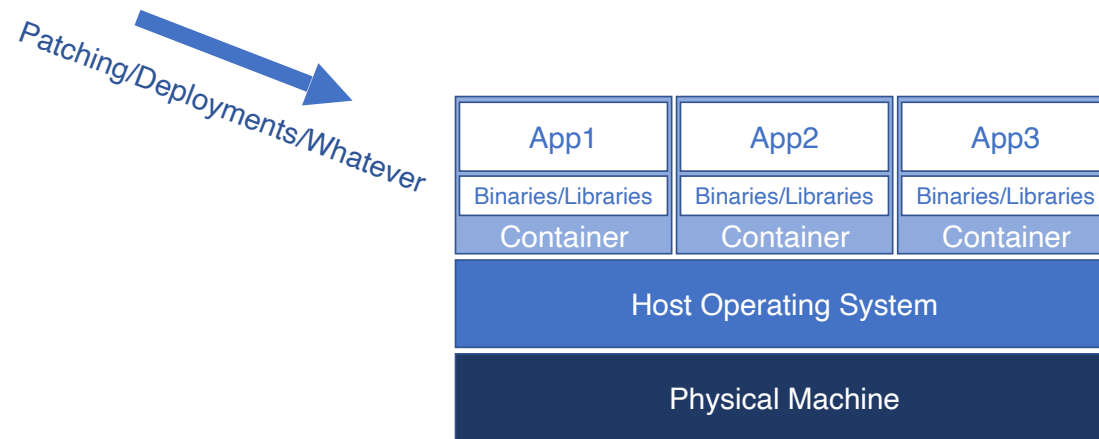


**Services, we care about getting work done!**





# Containers

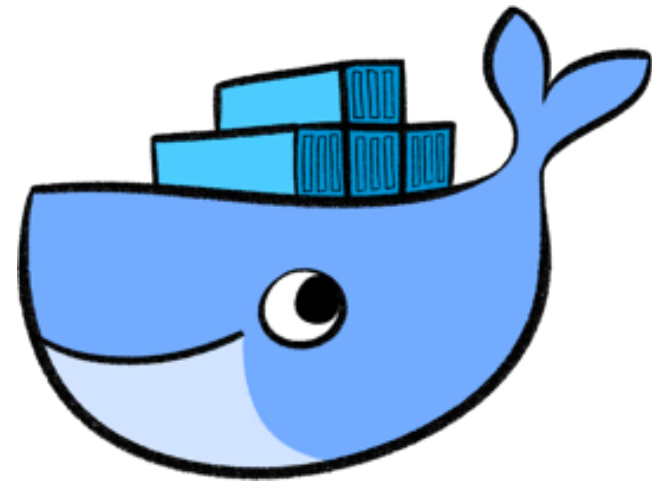


# Containerizing Apps and Data Centers

- Reducing development time
- Deployment automation – speed and consistency
- Enables DevOps and CI/CD scenarios
- Orchestration
- Rethink how you deploy - it's the service, not the server

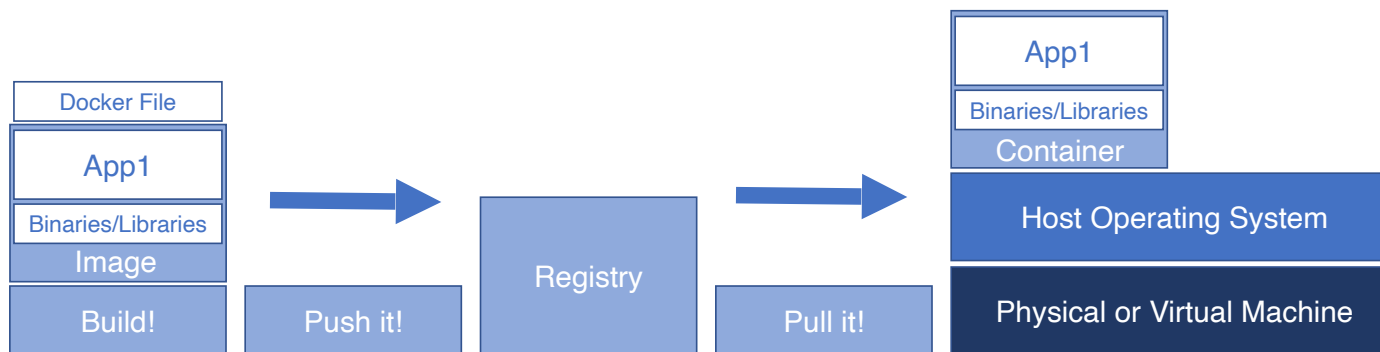
# The Container Universe

- Docker
  - Linux
  - Windows
  - Mac
- Docker Inc.
- Other Container Runtimes
  - containerd
  - CoreOS
  - Windows
  - chroot...chwhat?



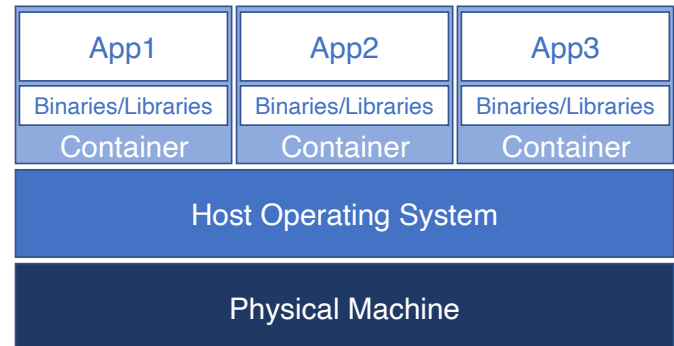
# Getting Containers

- **Images** – code, runtimes, libraries, environment variables
- **Registries** – where images live. Docker Hub, Azure Container Registry, internal
- **Docker Files** – defines the container image



# Container Internals

- Shared OS
- Resource isolation
  - Namespaces
    - Process Isolation - PID
    - File System – MNT
    - Network – NET
    - Interprocess Communication - IPC
    - Kernel Isolation - UTS
- Resource governing
  - cgroups
- Union file system



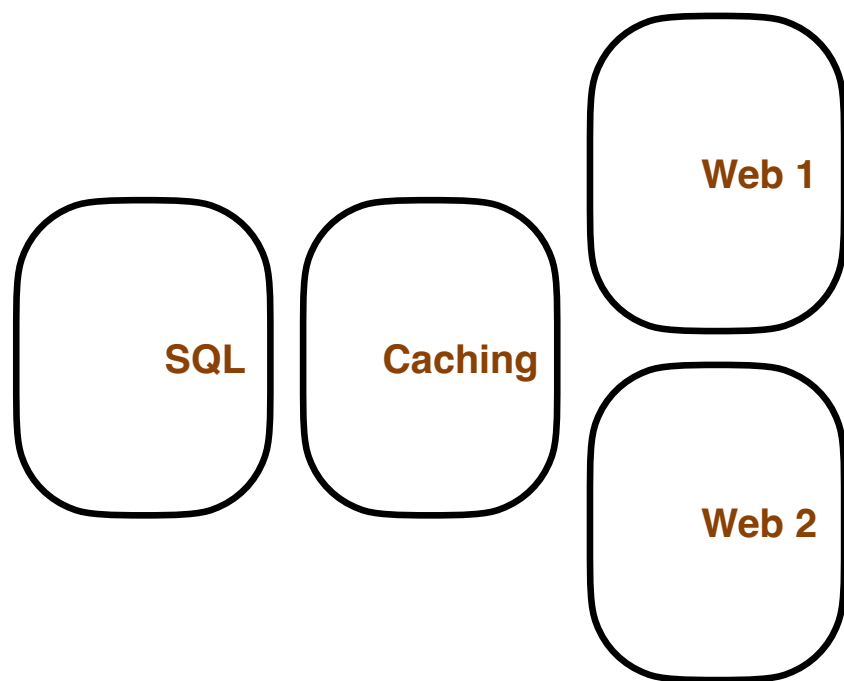
# Running SQL Server in Containers

- Why run SQL Server on a Container?
- Same reasons...
  - Deployments, upgrades, patching, speed...agility
  - What if the unit of persistency IS the database...NOT the Server!
- Only Linux is available
- Windows is no longer available
- Active Directory authentication available now
- What about Windows application containers?

# Hands on lab time...

- Pull an Image
- Run a Container
- Access our application
- Connect to the Container
- Persisting data with a Container

# Modern Application Deployment



- Where do I run the application?
- How do I scale the application?
- How do I consistently deploy?
- How do I or my applications access the services?



# Review

- **Container and Linux Fundamentals**
  - Container Fundamentals
  - Container Based Application Deployment
  - The Need for Container Orchestrators